Parallel Radiation Transport Algorithms and Associated Architectural Requirements

Jim Morel, Randal Baker, and James Warsa
Computer and Computational Sciences Division
Los Alamos National Laboratory

jim@lanl.gov, rsb@lanl.gov, warsa@lanl.gov

Overview

Introduction

The Transport Equation

Source Iteration

Convergence of Source Iteration

Convergence Acceleration

Krylov Methods and Preconditioners

Overview

Rectangular-Mesh Parallel Sweep Algorithms

Rectangular-Mesh Sweep Layouts

Parallel Sweeps on Unstructured Meshes

Parallel Sweeps on Structured AMR Meshes

Domain Decomposition Solution Algorithms

Computer Architecture Requirements

Introduction

- Transport calculations are of fundamental importance to ASCI.
- Transport calculations require far more memory and CPU-time than any other type of physics calculation in an ASCI code.
- There are three relevant types of transport calculations: neutron, thermal radiation, and charged-particle.
- The numerical methods for neutron transport calculations are very mature and generally adequate.
- The numerical methods for charged-particle transport and thermal radiation transport are far less mature and their adequacy for multidimensional calculations is not well established.
- From a numerical methods standpoint, thermal radiation transport is significantly more demanding than charged-particle transport, and charged-particle transport is significantly more demanding than neutron transport.

Introduction

- For the most part, the same basic parallel solution techniques can be used for all three types of calculations.
- Parallel solution techniques are quite mature for rectangular spatial meshes, but they remain a research topic for structured AMR meshes and fully unstructured meshes.

The Transport Equation

- The following neutron transport equation is representative of all three types of transport equations:

$$\frac{1}{v}\frac{\partial\psi}{\partial t} + \overrightarrow{\Omega}\cdot\overrightarrow{\nabla}\psi + \sigma_t\psi =$$

$$\int_0^\infty \int_{4\pi} \sigma_s(E' \to E, \overrightarrow{\Omega}' \cdot \overrightarrow{\Omega})\psi(\overrightarrow{\Omega}', E')\, d\Omega'\, dE' + Q\,.$$

- We use operator notation to express this equation as follows:

$$L\psi = S\psi + Q\,.$$

- The primary unknown is the angular flux, ψ, which is seven-dimensional: $\psi = \psi(t, \overrightarrow{r}, \overrightarrow{\Omega}, E)$.
- t is time, $\overrightarrow{r}$ is the particle position, $\overrightarrow{\Omega}$ is the particle direction, and E is the particle energy.

The Transport Equation

- This equation is just a statement of particle conservation in a differential phase-space volume, $dP = dV\, d\Omega\, dE$.
- The differential phase space volume consists of the differential spatial volume about $\overrightarrow{r}$, the differential solid angle about $\overrightarrow{\Omega}$, and the differential energy interval about E.
- The equation states that the rate of change of the number of particles in the differential volume dP is equal to the particle source rates minus the particle sink rates.
- Sinks for dP include
  - streaming out of the differential spatial volume about $\overrightarrow{r}$.
  - being either absorbed or scattered out of the differential solid angle about $\overrightarrow{\Omega}$ and the differential energy interval about the energy E.

The Transport Equation

- Sources include
  - streaming into the differential spatial volume about $\overrightarrow{r}$.
  - being scattered into the differential solid angle about $\overrightarrow{\Omega}$ and the differential energy interval about the energy E.
  - Being explicitly created within dP.

Source Iteration

- The standard technique for solving the transport equation is the source iteration technique.
- Since the source term contains all the angle-energy coupling, it is iteratively lagged:

$$\psi^{\ell+1} = L^{-1} S \psi^{\ell} + L^{-1} Q \, .$$

- On rectangular meshes, the operator L represents a block lower-triangular matrix with each block corresponding to the unknowns in a single cell.
- Such matrices are very easy to invert on a scalar machine.
- In particular, one sequentially solves for the unknowns in each mesh cell, moving across the mesh in the direction of particle flow.
- Because of this movement, the inversion process is called a sweep.

Source Iteration

- The standard technique for solving the transport equation is the source iteration technique.
- Since the source term contains all the angle-energy coupling, it is iteratively lagged:

$$\psi^{\ell+1} = L^{-1} S \psi^{\ell} + L^{-1} Q \, .$$

- On rectangular meshes, the operator L represents a block lower-triangular matrix with each block corresponding to the unknowns in a single cell.
- Such matrices are very easy to invert on a scalar machine.
- In particular, one sequentially solves for the unknowns in each mesh cell, moving across the mesh in the direction of particle flow.
- Because of this movement, the inversion process is called a sweep.

Convergence of Source Iteration

- Assuming a zero initial guess for the scattering source, each source iteration adds the contribution to the solution from another generation of scattering.
- In other words, the first source iteration yields the solution due to particles that have not yet scattered. The second source iteration adds the contribution to the solution from particles that have scattered once. The third source iteration adds the contribution to the solution from particles that have scattered twice, and so on.
- Thus the source iteration process converges rapidly in problems in which particles only scatter a few times on average before either being absorbed or streaming out of the system.
- Arbitrarily slow convergence rates can be obtained in diffusive problems, i.e., problems with little absorption that appear very thick to the particles.

Convergence Acceleration

- In diffusive problems, the source iteration process must be accelerated. This is absolutely critical in thermal radiation transport calculations.
- In general, the resulting solution techniqes can be thought of as a two-grid method, with a diffusion equation playing the role of the “coarse-grid” transport operator.
- The diffusion equation must be discretized in accordance with the discretization of the transport equation to avoid instabilities.
- Because the transport equation is generally discretized using discontinuous finite-element methods, this leads to non-standard diffusion discretizations that are very difficult to solve.
- Furthermore, the acceleration technique can become increasingly ineffective with severe discontinuities in material properties.

Krylov Methods and Preconditioners

- An alternative to the two-grid approach is to use a Krylov method to solve the transport equation, and to recast the diffusion-based acceleration technique as a diffusion-based preconditioner.
- This approach relaxes the consistency requirements on the diffusion discretization, and is easily implemented in existing codes.
- Highly effective preconditioning is achieved via a standard diffusion discretization rather than a discontinuous diffusion discretization, and the material discontinutity problem is eliminated.
- The preconditioned Krylov approach is having a revolutionary impact upon solution techniques for the transport equation.

Krylov Methods and Preconditioners

- The equation which the Krylov method solves is not

$$L\psi = S\psi + Q\,,$$

but rather

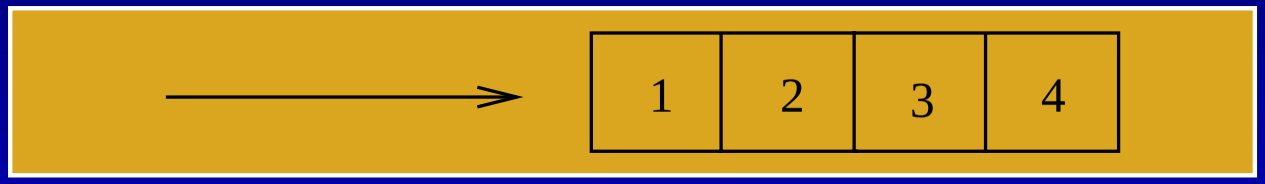
$$(I - L^{-1}S)\psi = L^{-1}Q\,.$$

- The latter equation has far better spectral properties, but requires a sweep to evaluate its action.
- Thus sweeps and diffusion solves are required with both traditional and advanced transport solution techniques.
- Our parallel methods for solving diffusion equations are taken from the applied mathematics community.
- Thus we do not discuss them further.

Rectangular-Mesh Parallel Sweep Algorithms

- On a N-cell 1-D mesh, the sweep starts at the boundary cell for which the particle direction is incident.
- Solving for the unknowns in the first cell provides the incident solution values required to solve for the unknowns in the second cell, and so on, as illustrated in Fig. 1.

Figure 1: 1-D Sweep

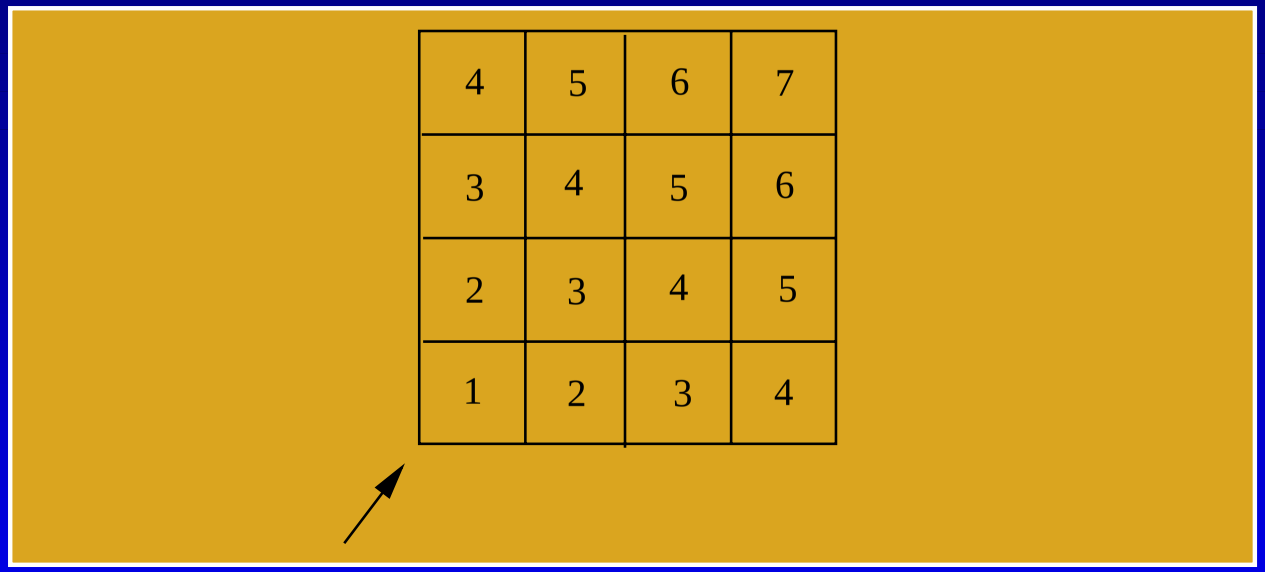


- The solution process is inherently sequential and parallelism is not possible.

Rectangular-Mesh Parallel Sweep Algorithms

- On a $N \times N$ 2-D mesh, the sweep starts at the corner mesh cell for which the particle direction is incident on two cell faces.
- Solving for the unknowns in the corner mesh cell provides the incident solution values required to solve for the unknowns in the next two cells, and so on, as illustrated in Fig. 2.

Figure 2: 2-D Sweep



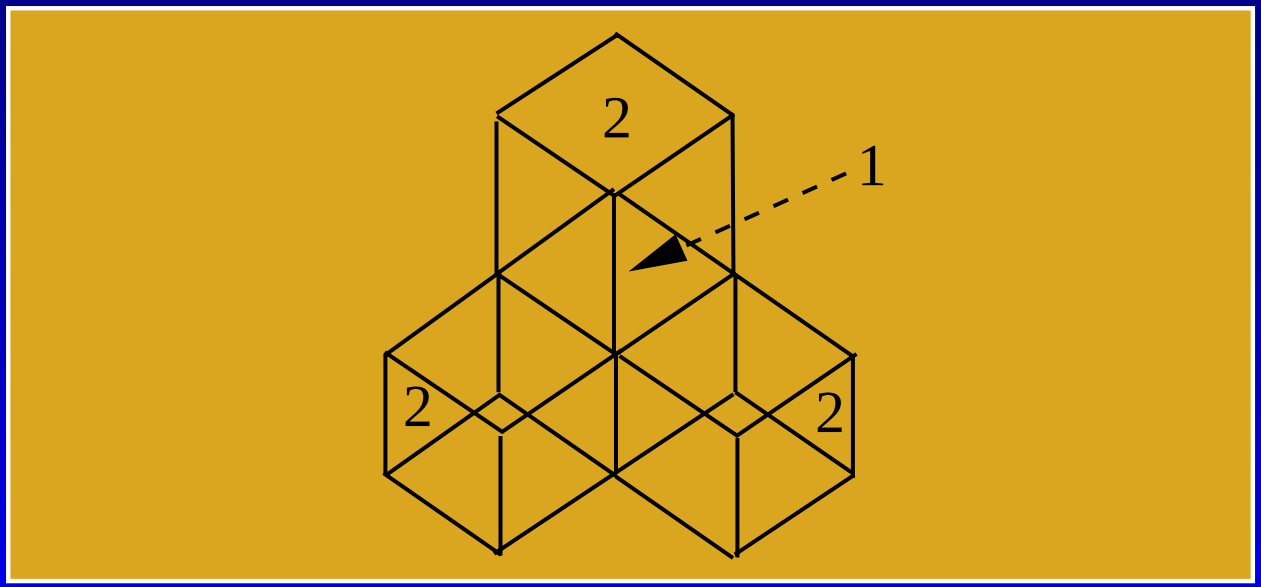
Rectangular-Mesh Parallel Sweep Algorithms

- $2N - 1$ sequential solution steps are required with N^2 cells, so one can simultaneously solve for O($N/2$) unknowns per step on the average.
- Thus parallelization is possible.

Rectangular-Mesh Parallel Sweep Algorithms

- On a $N \times N \times N$ 3-D mesh, the sweep starts at the corner mesh cell for which the particle direction is incident on three cell faces.
- Solving for the unknowns in the corner mesh cell provides the incident solution values required to solve for the unknowns in the next three cells, and so on, as illustrated in Fig. 3.

Figure 3: 3-D Sweep



Rectangular-Mesh Parallel Sweep Algorithms

- $3N - 2$ sequential solution steps are required with N^3 cells, so one can simultaneously solve for O($N^2/3$) unknowns per step on the average.
- Thus parallelization is possible.

Rectangular-Mesh Sweep Layouts

- The data layout for a 2-D transport calculation is 1-D and the data layout for a 3-D calculation is 2-D.
- For instance, consider a 4×4 mesh.
- The cell data for all directions and energies would be mapped to four processors as follows:

Table1: Layout for 4×4 mesh.

P1	P2	P3	P4
4	5	6	7
3	4	5	6
2	3	4	5
1	2	3	4

Rectangular-Mesh Sweep Layouts

- This is a non-standard layout.
- There is a load balancing problem that is addressed by starting the solution process for additional directions before any one direction finishes.
- On rectangular 3-D meshes, parallel efficiencies on the order of 50-80 percent (with weak scaling) have been achieved with thousands of processors.

Parallel Sweeps on Unstructured Meshes

- Matters become much more complicated on unstructured meshes.
- An ordering of the unknowns that leads to a block lower-triangular structure for the sweep equations always exists on “reasonable” 2-D unstructured meshes, but this is not necessarily the case for 3-D meshes.
- A block lower-triangular ordering essentially always exists for “reasonable” tetrahedral meshes, but it almost never exists for distorted hexahedral meshes.
- Even when a block lower-triangular ordering exists, the number of steps associated with a mesh of N cells varies along with the number of cells in each step.
- Optimal partitioning is an NP-complete problem.
- Parallel unstructured-mesh sweep algorithms are still a research topic.
- Nonetheless, very good parallel efficiencies have been obtained with thousands of processors on tetrahedral-meshes.

Parallel Sweeps on Structured AMR Meshes

- The difficulty of defining an efficient sweep algorithm for a structured AMR mesh can be similar to that for either unstructured or rectangular meshes.
- The difficulty of defining an efficient sweep algorithm for a structured AMR mesh usually increases with the flexibility of the mesh generation process.

Domain Decomposition Solution Algorithms

- Parallel transport solution algorithms based upon standard spatial domain decomposition are being investigated.
- In effect, source iterations are performed within each sub-domain and the solution values on the sub-domain boundaries are iterated upon.
- This approach appears to have considerable promise as long as each spatial sub-domain appears thick to the particles.
- When the sub-domains become thin to the particles, the algorithms become inefficient due to slow convergence of the sub-domain boundary unknowns.
- These approaches can be combined with Krylov methods.
- While Krylov methods can improve efficiency in the thick case, they are ineffective for the thin case.

Computer Architecture Requirements

- A lightweight OS that minimizes the number of interrupts generated.
  - The sweep algorithm is tightly coupled in the sense that the equations for the downwind cells cannot be solved until those for the upwind cells have been solved.
  - If a downwind processor has not finished it's computation-communication cycle due to an OS interrupt of some sort, then everything halts until it is freed up.
- A high performance memory sub-system.
  - Very few floating point operations are performed per memory load, so data quickly moves through the cache.

Computer Architecture Requirements

- A low latency, high bandwidth interconnect to minimize communication costs. Note that this also requires a high performance MPI library since on modern hardware most of the latency is due to software, not hardware.
  - Data is communicated between processors after each sequential step in the sweep.
  - The relative amount of data communicated is large enough to require significant bandwidth, but small enough to be sensitive to latency.

Computer Architecture Requirements

- These requirements are best met by a system composed of a few thousand high performance processors with a large amount of memory per processor.
- Minimizing the number of processors while meeting the requirements for speed and memory makes the design and implementation of a high performance interconnect easier.
- Although modern codes have been extensively parallelized, Amdahl's Law still comes into effect as the number of processors increases.
- Processor reliability becomes problematic with more than a few thousand processors.
- Existing ASCI software is designed for RISC-based CPU's. Transitioning to vector-type CPU's could result in significant redesign costs.