Parallel S_N Methods

Peter N. Brown

Center for Applied Scientific Computing

May 2001

3-D time-dependent Boltzmann transport equation

$$\frac{1}{v(E)}\frac{\partial \psi}{\partial t} + \Omega \bullet \nabla \psi + \sigma(r, E)\psi =$$

$$\int_0^\infty \int_{S^2} \sigma_s(r, \Omega' \bullet \Omega, E' \rightarrow E)\psi(r, \Omega', E', t)\, d\Omega' dE' + q$$

where

$\psi(r, \Omega, E, t)$	= particle flux or intensity
r	= (x, y, z)
E, E'	= energies
Ω, Ω'	= directions
$q(r, \Omega, E, t)$	= source
$v(E)$	= particle speed
σ	= total cross section
σ_s	= scattering cross section

Multi-group energy discretization

$$\frac{1}{v_g}\frac{\partial \psi_g}{\partial t} + \Omega \bullet \nabla \psi_g + \sigma_g(r)\psi_g =$$

$$\sum_{g'=1}^{G} \int_{S^2} \sigma_{s,g,g'}(r, \Omega' \bullet \Omega)\psi_{g'}(r, \Omega')\, d\Omega' + q_g, \quad g = 1, \cdots, G$$

where $0 \le E_G < \cdots < E_g < E_{g-1} < \cdots < E_0$

$\psi_g(r, \Omega, t)$	= group g particle flux or intensity
r	= (x, y, z)
Ω, Ω'	= directions
$q_g(r, \Omega, t)$	= group g source
v_g	= group g particle speed
σ_g	= group g total cross section
$\sigma_{s,g,g'}$	= group g' to g scattering cross section

Transport solver combines traditional & multilevel methods

- **Based on Petrov-Galerkin finite element in space, multigroup in energy, S_N or finite element in direction**
- **Differential/algebraic system:** $f(t, \Psi, \dot{\Psi}) = 0$
  - —**implicit time integration via *IDA***
  - —**GMRES linear solver:** $T\Psi = F$
- **The matrix structure for** T **is** $T = H - S$ **where**

$$H = \begin{bmatrix} H_1 & & \\ & \ddots & \\ & & H_G \end{bmatrix}, \quad S = \begin{bmatrix} S_{11} & \cdots & S_{1G} \\ \vdots & \ddots & \vdots \\ S_{G1} & \cdots & S_{GG} \end{bmatrix} \text{ and}$$

$$H_g = \mathrm{diag}(H_{g1}, \cdots, H_{gL})$$

Matrix formulation

- **The components in the matrix** $T = H - S$ **have the form**

$$H_{gd} = \begin{bmatrix} \Omega_d \cdot C + (\Sigma_g + (v_g \Delta t)^{-1} I) M \\ B_d \end{bmatrix}$$

and

$$S_{gg'} = \begin{bmatrix} \sum_{n=0}^{N} L_n^{+} \Sigma_{s,n,gg'} L_n M \\ 0 \cdot B_d \end{bmatrix}$$

- M **= mass matrix,** C **= discrete gradient,** B_d **= boundary operator, and** Σ_g**,** $\Sigma_{s,n,gg'}$ **= cross section matrices**
- $\Phi_n \equiv L_n \Psi$ **= all** n^{th} **spherical harmonic moments of** Ψ
- $\Psi \equiv L_n^{+} \Phi_n$ **= flux vector from** n^{th} **order moments in** Φ_n

PVODE software suite

User Problem-Defining Code

PVODE
ODE
Integrator
$\dot{u} = f(t,u)$

IDA
DAE
Integrator
$f(t,u,\dot{u}) = 0$

KINSOL
Nonlinear
Eqn. Solver
$f(u) = 0$

Band Solver

Preconditioned GMRES Solver

...

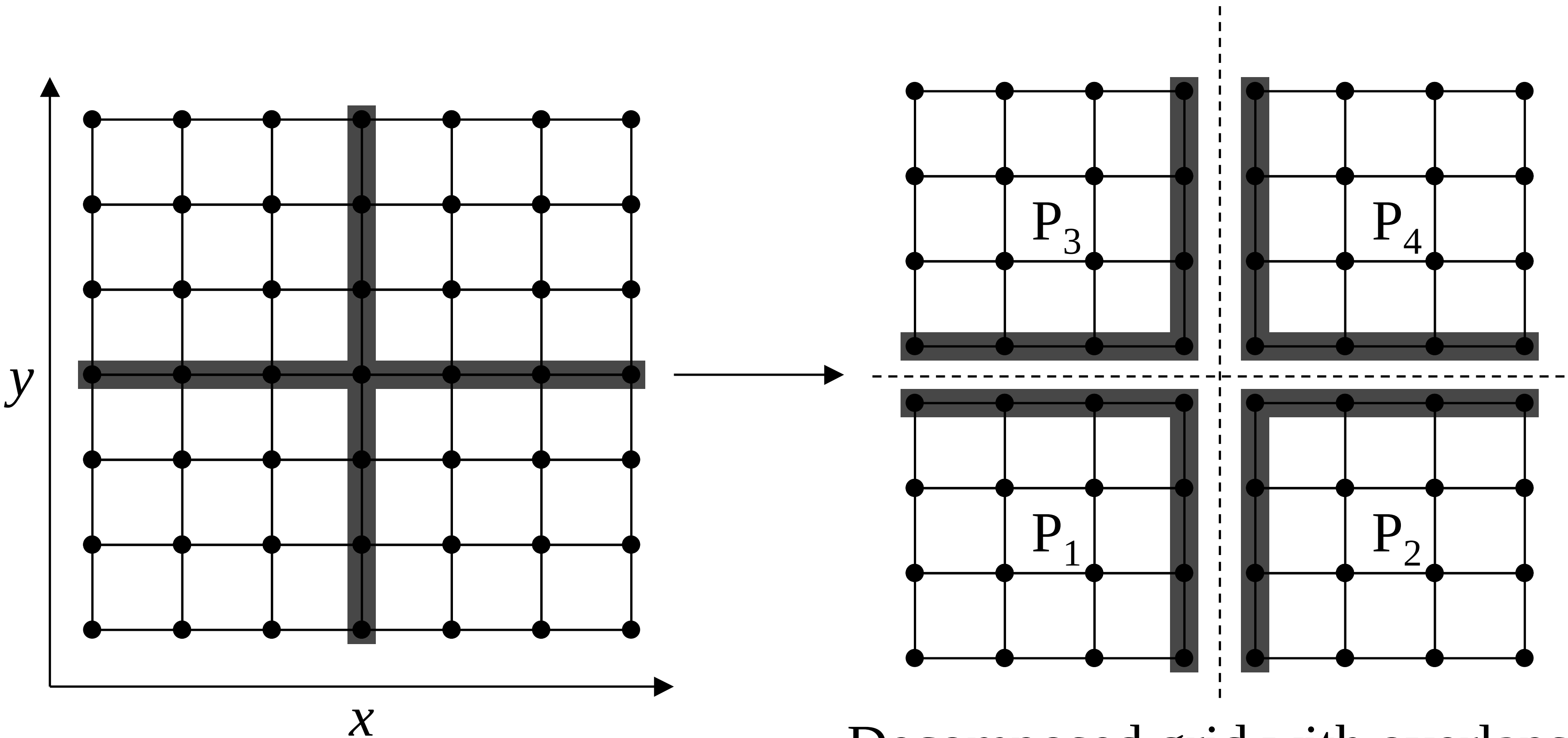
General Preconditioner Modules

Vector Kernels

Parallel $T\Psi$ evaluation

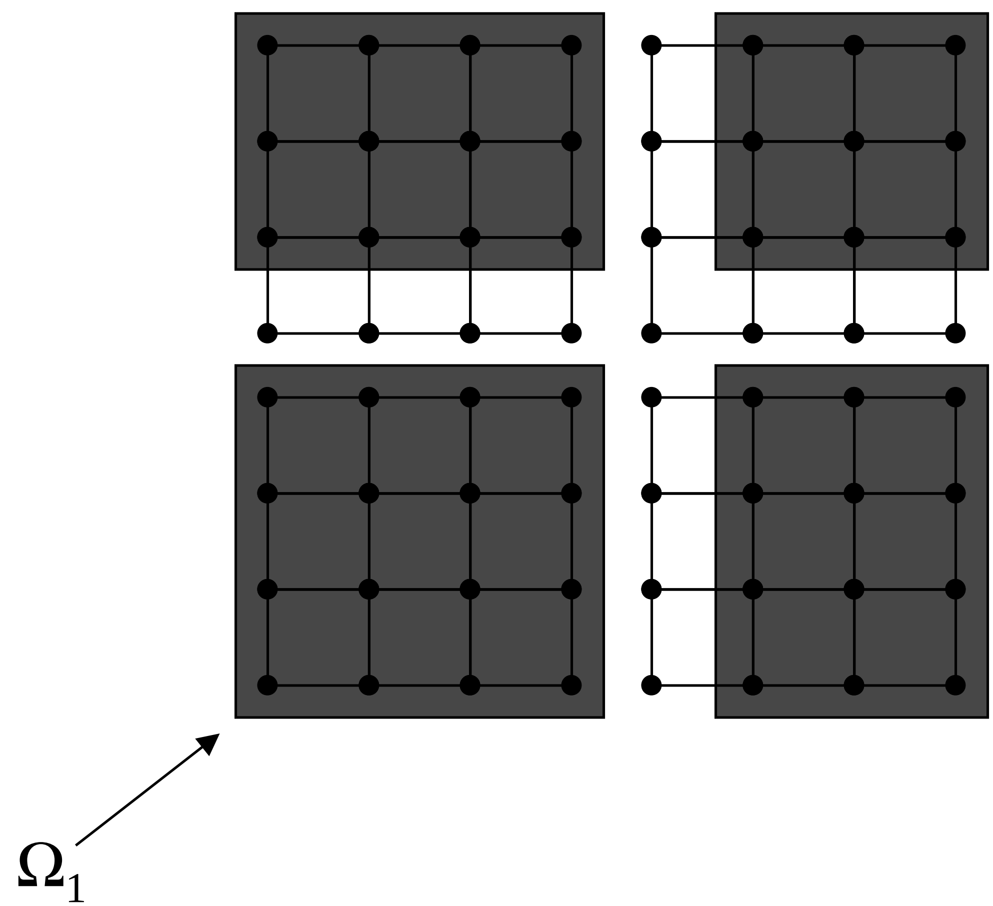
- **Parallelism achieved via: spatial domain & energy group decomposition, OpenMP threads (directions and vectors)**
- **Use MPI for spatial and energy group decomposition**
- **IBM ASCI White architecture**
  - **—512 SMP nodes, with 16 processors per node**
  - **—max of 2048 MPI tasks**
  - **—remaining procs/node used via OpenMP threads**

Zonal based spatial decomposition leads to nodal values on overlapped grids

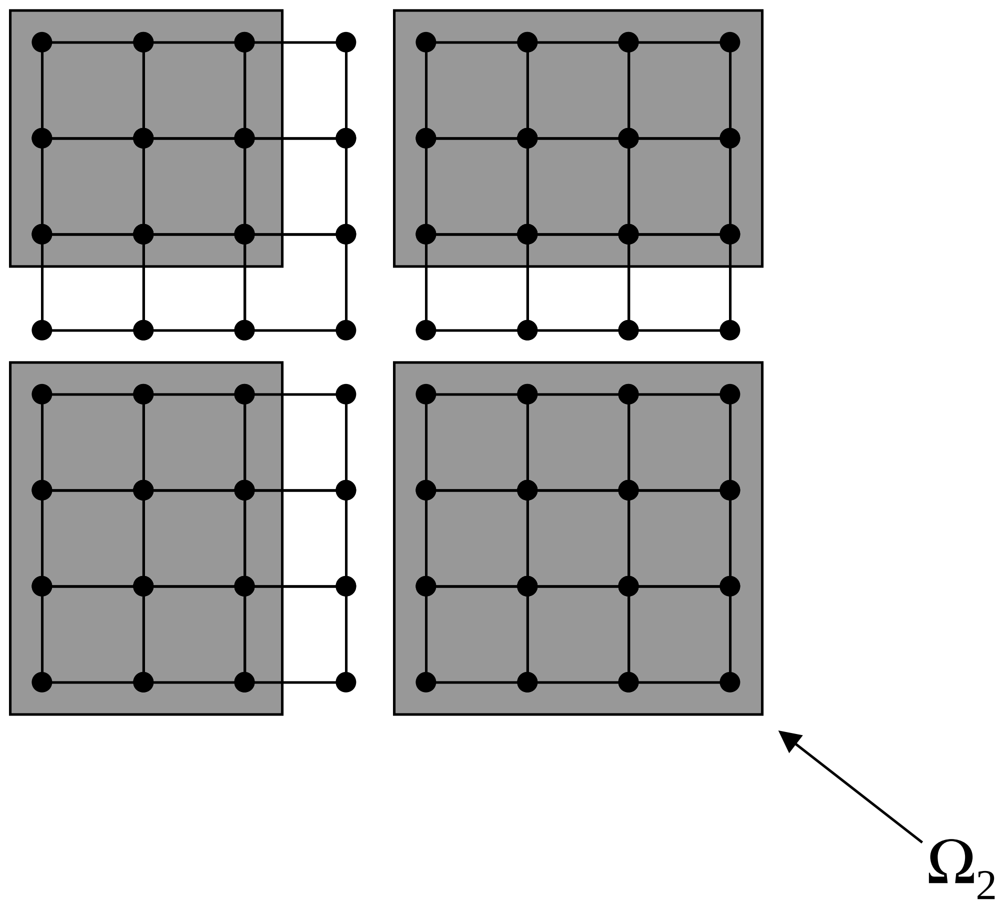


Decomposed grid with overlaps

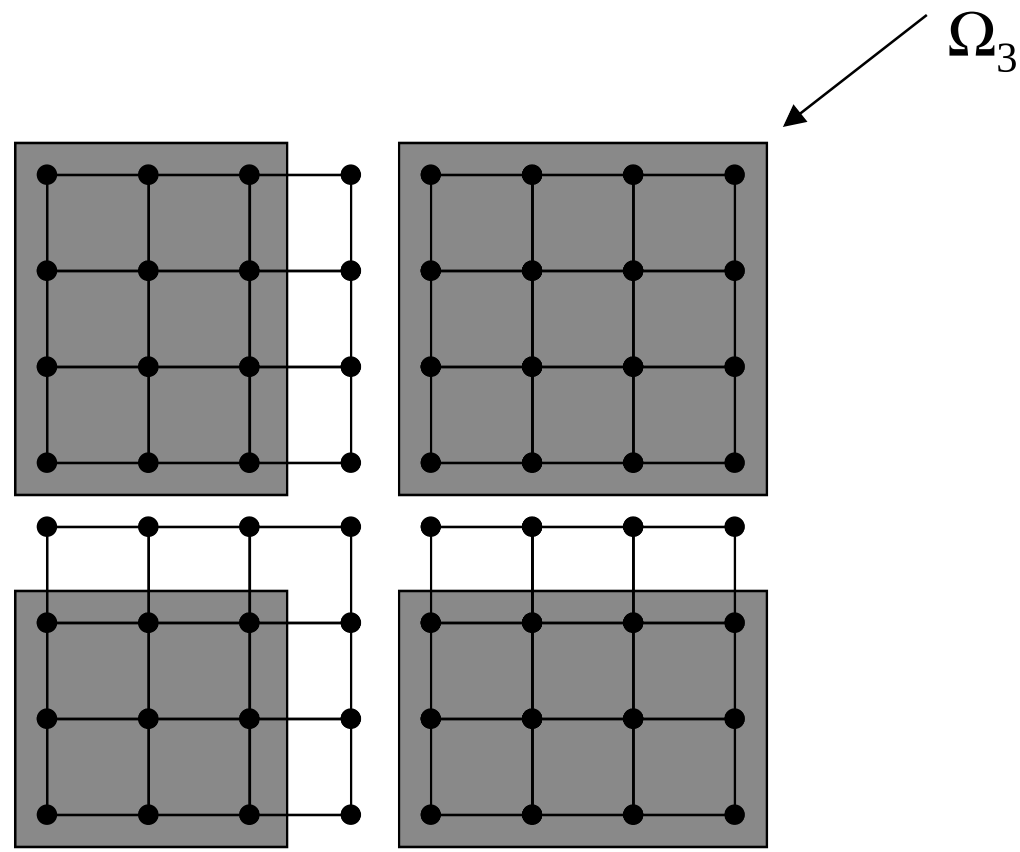
Nodal ownership is direction dependent



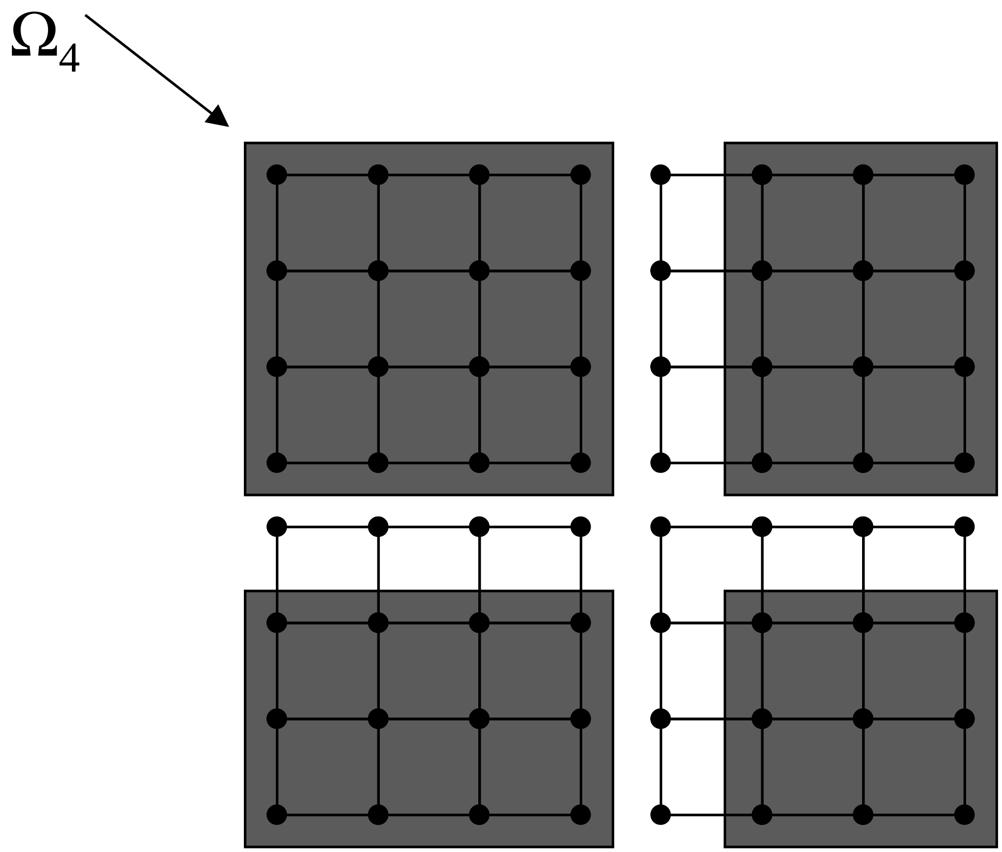
Nodal ownership is direction dependent



Nodal ownership is direction dependent



Nodal ownership is direction dependent



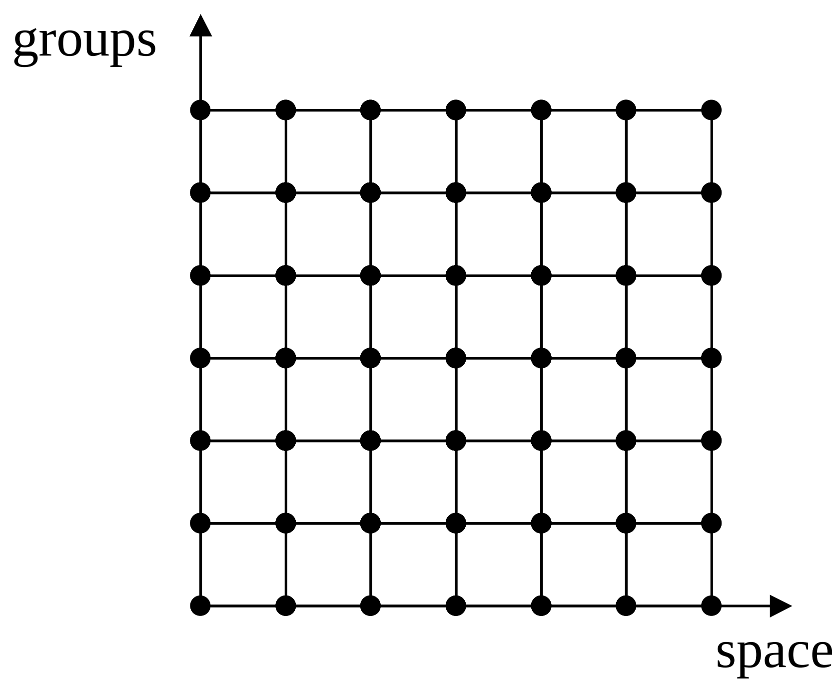
Energy group decomposition

- Assume 1 group per processor (G processors)
- Broadcast operations needed to calculate $S\,\Psi$ where

$$S = \begin{bmatrix} S_{11} & \cdots & \cdots & S_{1G} \\ \vdots & \ddots & & \vdots \\ S_{g1} & & \ddots & S_{gG} \\ \vdots & & \ddots & \vdots \\ S_{G1} & \cdots & \cdots & S_{GG} \end{bmatrix} \quad \text{and} \quad \Psi = \begin{bmatrix} \Psi_1 \\ \vdots \\ \Psi_g \\ \vdots \\ \Psi_G \end{bmatrix}$$

Communication involved in residual evaluations

- Communication of nodal data to overlapped mesh for each direction
- Broadcast operations for scattering operator
- 2-dimensional processor topology



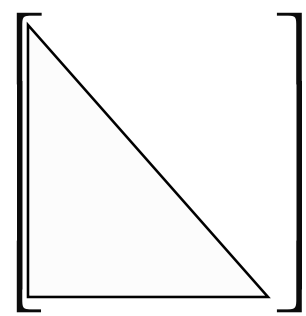
Preconditioning strategies

- **GMRES iterative solver used for** $T\Psi = F$
- **Exploit matrix structure and physics**
- **Matrix structure**

$$T = \begin{bmatrix} T_{11} & \cdots & T_{1G} \\ \vdots & \ddots & \vdots \\ T_{G1} & \cdots & T_{GG} \end{bmatrix}$$

and

$$H_g = \text{diag}(H_{g1}, \cdots, H_{gL})$$

$H_{gd} =$ 

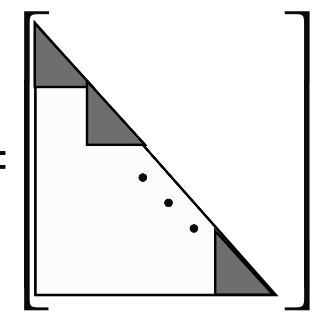
where each $T_{gg} = H_g - S_{gg}$

Preconditioners using matrix structure

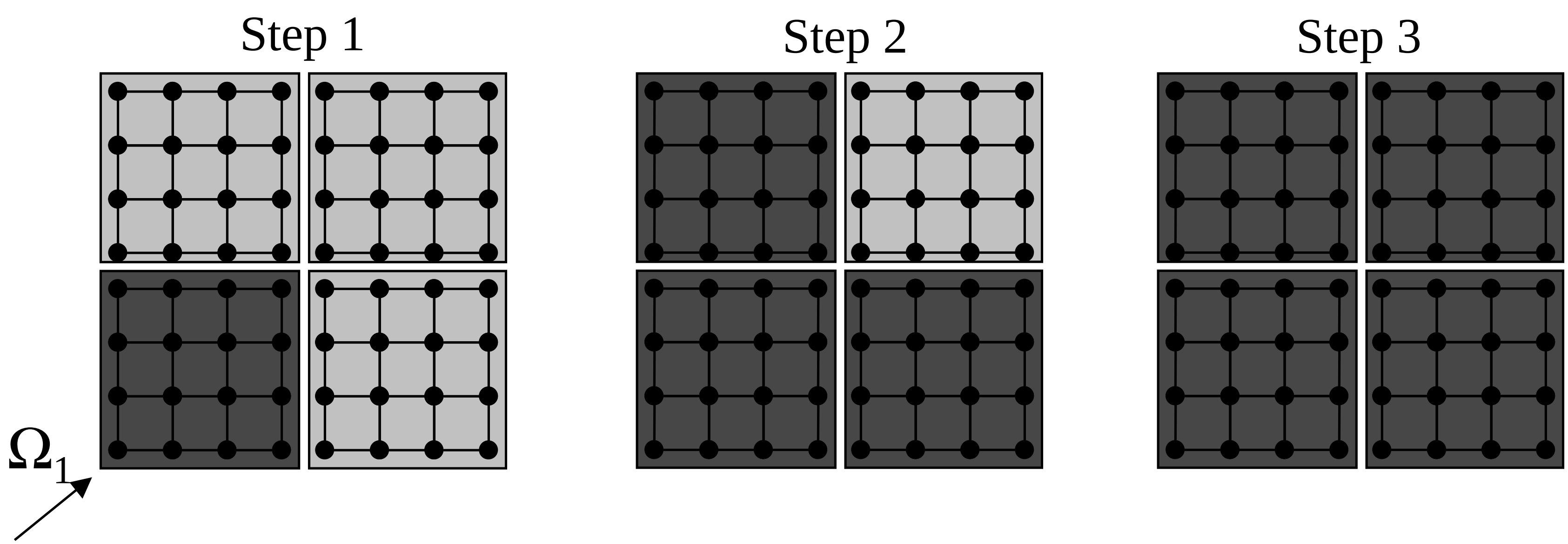
- Use lower triangular part of T in energy

$$P = \begin{bmatrix} T_{11} & & 0 \\ \vdots & \ddots & \\ T_{G1} & \cdots & T_{GG} \end{bmatrix}$$ **with blocks** $T_{gg} = H_g (I - H_g^{-1} S_{gg})$

- H_{gd}^{-1} **approximated by block Jacobi iteration, where**

$H_{gd} =$ 

Block Jacobi convergence



- Convergence in npx + npy + npz - 2 steps in 3D
- The smaller Δt the more "effective" absorption, and hence should get convergence in fewer steps
- Norm-based stopping test needs a global reduction, so use a fixed # of steps for time dependent problems
- Steady state problems require full # of steps

Preconditioners based on physics

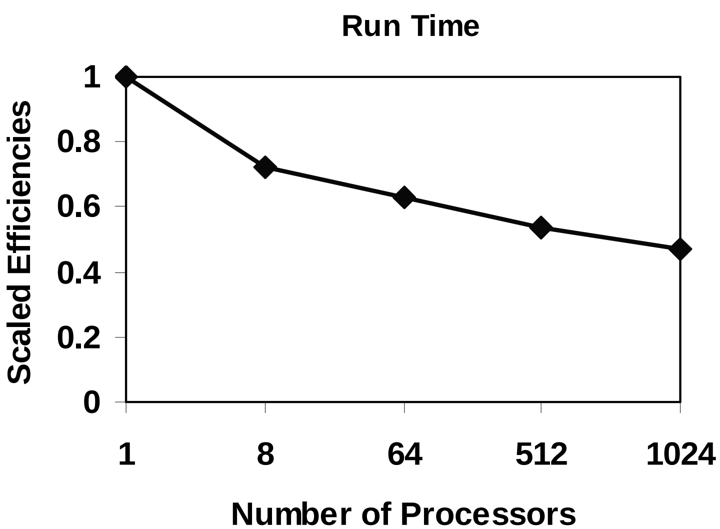
- $(I - H_g^{-1} S_{gg})^{-1}$ **approximated using**
  - **matrix based on *DSA***
  - **source iteration (with/without *DSA*)**
  - **BiCGSTAB iteration (with/without *DSA*)**
- **DSA solution**
  - **27-point stencil defined on overlapped nodal mesh**
  - **singular diffusion matrix**
  - **approximate solution given by CASC *hypre* SMG multigrid solver**

- ***Operator-split preconditioners use full matrix***

Scalability studies on ASCI Blue and White

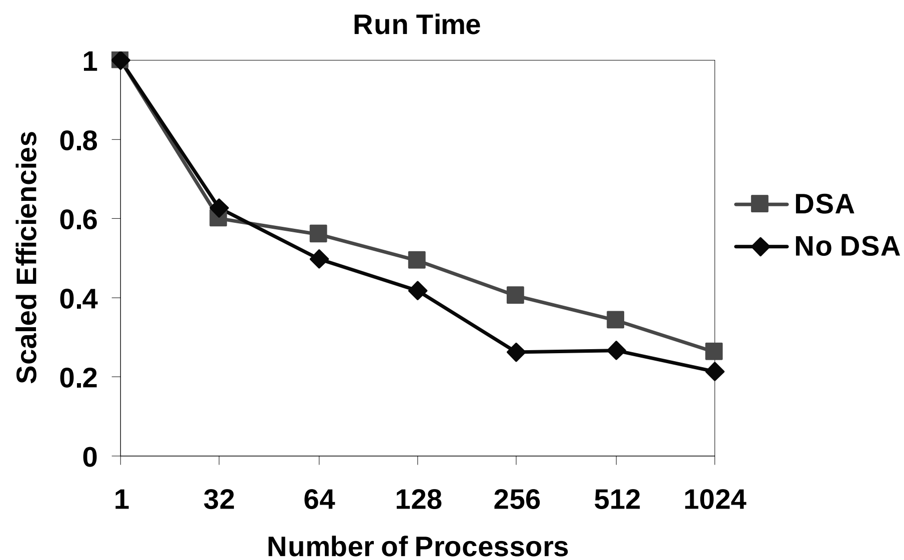
Time dependent study(Blue)

- 1 block Jacobi iteration
- 30x30x30 zones & 24 directions/processor
- Simple box problem



Two steady state studies(White)

- *DSA* + full block Jacobi iter.
  4 GMRES iter-s for all sizes
- Full block Jacobi iter.
  2 GMRES iter-s for all sizes

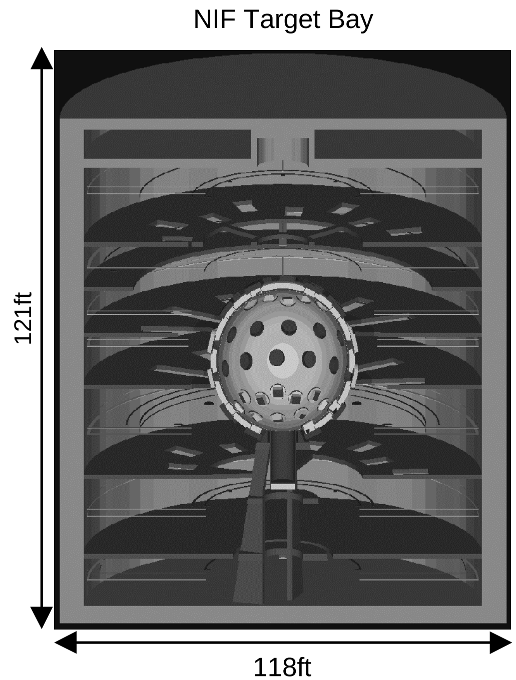


Hybrid MPI/OpenMP implementation

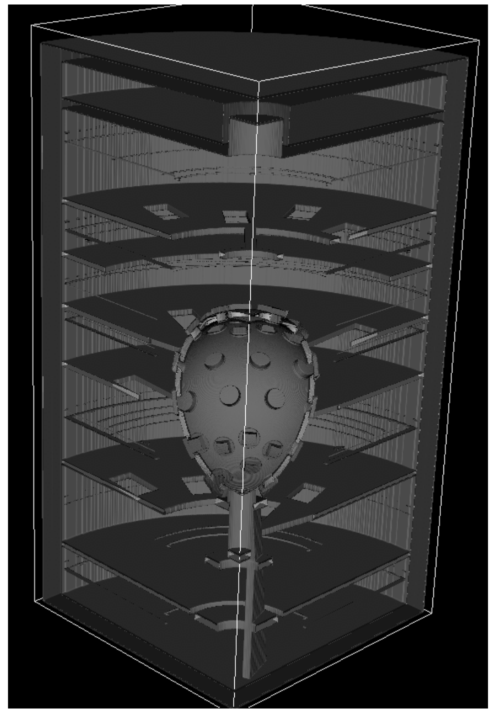
- Threaded implementation outperforms pure MPI implementation
- Test problem with 60x60x60 spatial zones and 24 directions per processor

Nodes	Pure MPI N nodes; N MPI tasks 1 processor per node	Hybrid MPI/OpenMP N nodes; N MPI tasks 4 processors per node	Pure MPI N nodes; 4N MPI tasks 4 processors per node
1	674.36 sec	222.15 sec speedup: 3.04 efficiency: 76%	203.01 sec speedup: 3.02 efficiency: 76%
8	804.99 sec	320.17 sec speedup: 2.52 efficiency: 63%	292.82 sec speedup: 2.75 efficiency: 69%
64	1437.27 sec	695.42 sec speedup: 2.07 efficiency: 52%	767.10 sec speedup: 1.87 efficiency: 47%
128	1596.25 sec	800.69 sec speedup: 1.99 efficiency: 50%	935.64 sec speedup: 1.71 efficiency: 43%

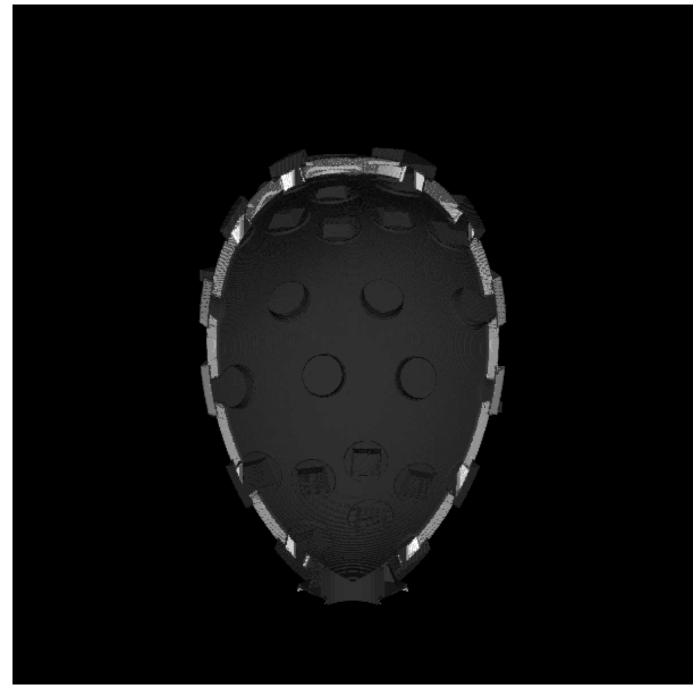
Large neutron time dependent run on ASCI White (NIF Target Bay)



Target bay geometry



Isosurfaces depicting spatial geometry



Run on ASCI White:
9 billion unknowns
(400x400x800 spatial zones,
24 directions, 3 energy groups)
Pulsed neutron point source

Hybrid MPI/OpenMP Implementation used 4,096 processors (1,024 MPI tasks with 4 threads per task)

Future work

- Development of adaptive techniques using structured adaptive mesh refinement
  - full finite element discretization approach
  - adaptive refinement in all of phase space
  - multilevel solution
- Better scattering kernel representation when using finite elements
- Implement corner balance in our code
- Implement operator-split preconditioner

UCRL-VG-143574

Work performed under the auspices of the U. S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48