LA-UR-13-20961
Approved for public release; distribution is unlimited.

Title: Multidimensional Discrete Ordinates Transport on Massively Parallel Architectures

Author(s): Azmy, Yousry
Zerr, Robert J.

Intended for: Seminars: U of Florida (2/14/13), Rennselaer Polytech Inst. (3/28/13, U. So. Carolina (4/13)

Multidimensional Discrete Ordinates Transport on Massively Parallel Architectures

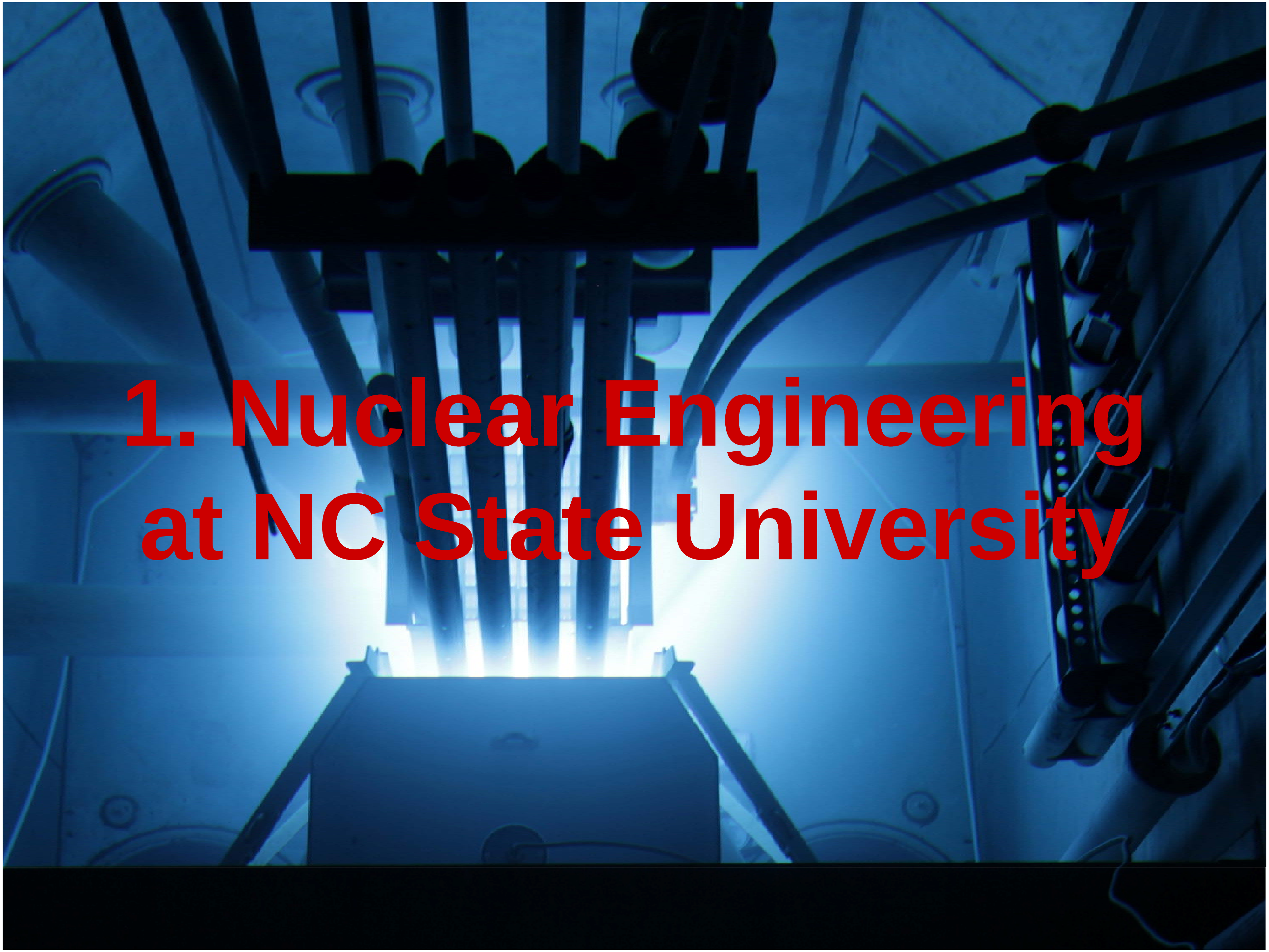
Yousry Azmy

Professor & Head
Department of Nuclear Eng
NC State University

Joe Zerr

Research Staff
Los Alamos National Laboratory

0. Outline



1. Nuclear Engineering at NC State University

1. Department's Brief History

1950 Established as graduate program in Physics Dept

~1950 First non-governmental university-based research reactor

1955 Two PhDs awarded

1962 Department of Nuclear Engineering established

1965 Rapid growth from 4 to 9 faculty; thrust areas: (1) Fission power reactors; (2) Radiation applications

1973 1MW PULSTAR operational (4th on-campus reactor)

1983 Added Plasma/fusion graduate track

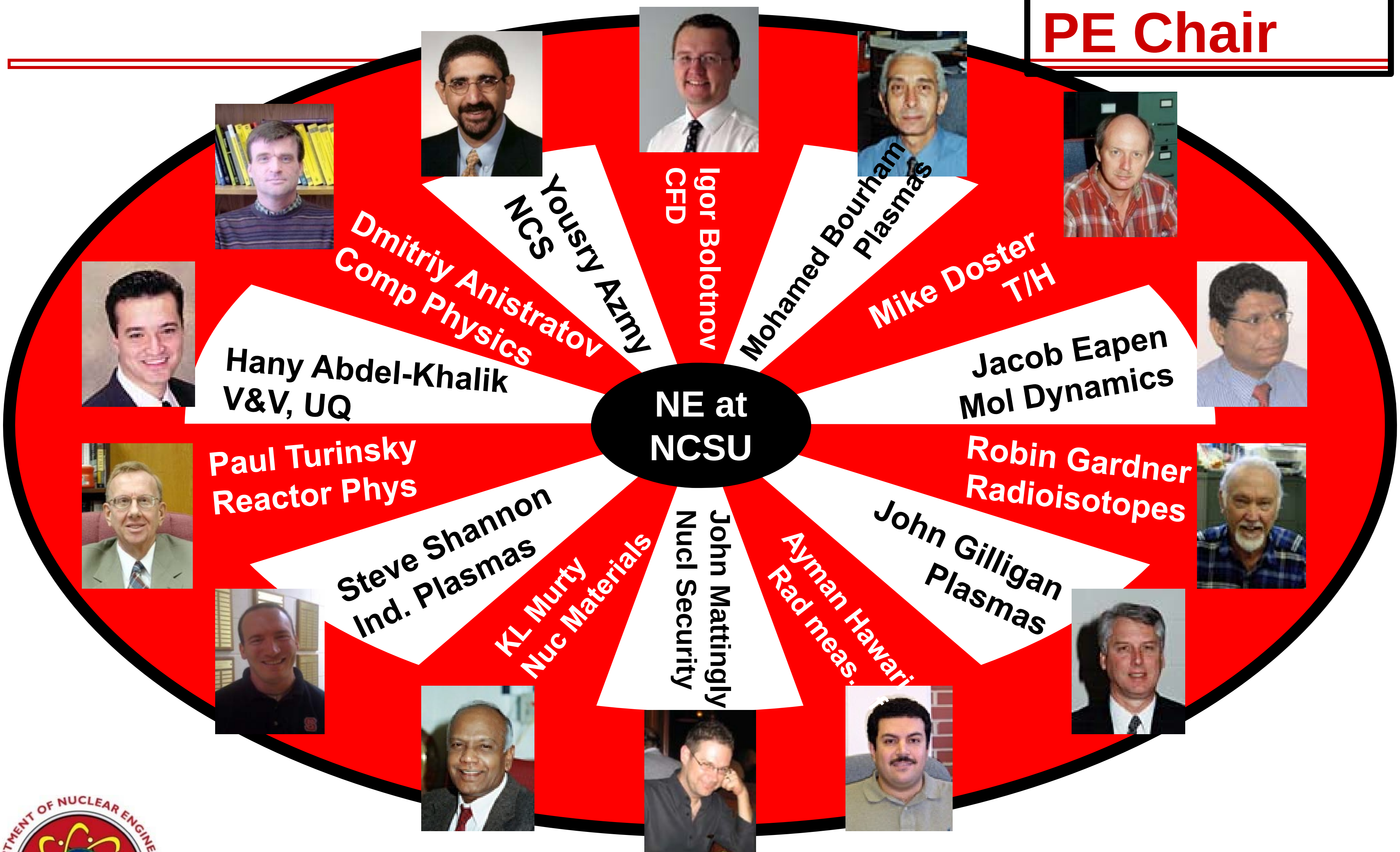
1994 Combined five-year BS/MNE degree established

2008 *Master of Nuclear Engineering* degree via Distance Ed

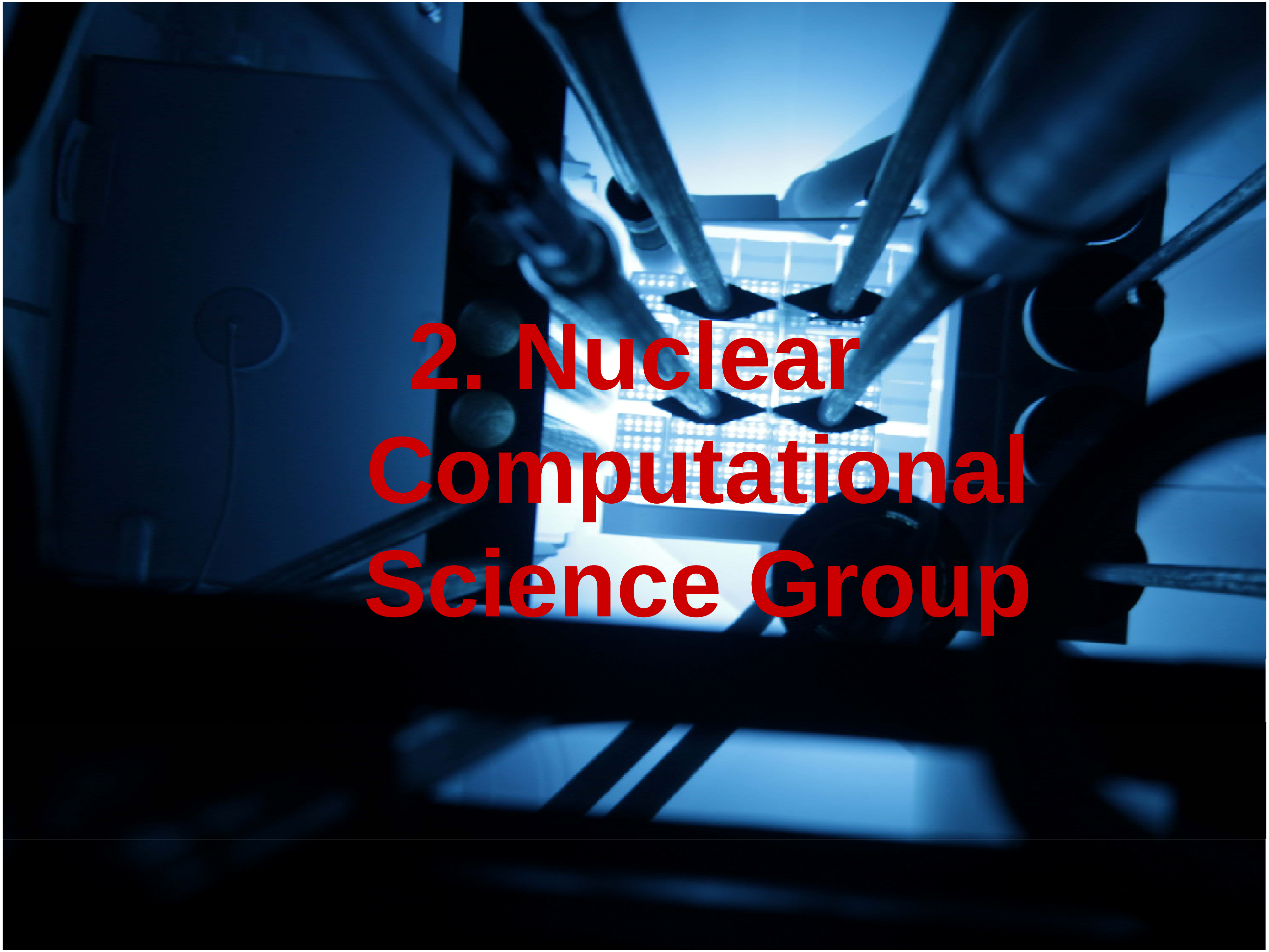
1. NCSU's Nuclear Engineering Today

- **Our Faculty:**
  - 8 active faculty in 2007 ⇨ 15 today
  - 2 open positions currently in search
  - 2 endowed chairs (*Progress Energy* & *Duke Energy*): PE Chair in search
  - Multiple Joint Faculty Appointments with ORNL and INL
  - Pivotal role in CASL: Turinsky Chief Scientist, Doster Ed Programs
  - Gilligan: Director of NEUP
- **Our Students:**
  - Enrolments surpassed 200 UGs & 100 Grads
  - Won Mark Mills Award (best PhD) 9 times in Award's 53 years
  - ~10% win one or more award, scholarship, or fellowship
- **Space:**
  - Increased by more than 50% since 2008
  - Future move to new building on Centennial Campus

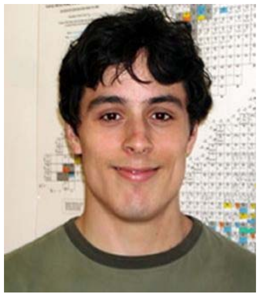
Add Nam & PE Chair



2. Nuclear Computational Science Group



2. NCSG at NCSU



Sebastian Schunert: PhD Student
Thesis: ***Comparing Various Spatial Discretization Schemes Based on a Method of Manufactured Solutions Benchmark Suite***

Sean O'Brien: PhD Student
MS Thesis: A Posteriori ***Error Estimators for the Discrete Ordinates Approximation of the One-Speed Neutron Transport Equation***

Sameer Vhora: MNE Student
Thesis: ***Spectral Analysis of Parallel Block Jacobi Iterations for Solving the Discrete Ordinates Equations with the ITMM Approach***

2. NCSG at NCSU

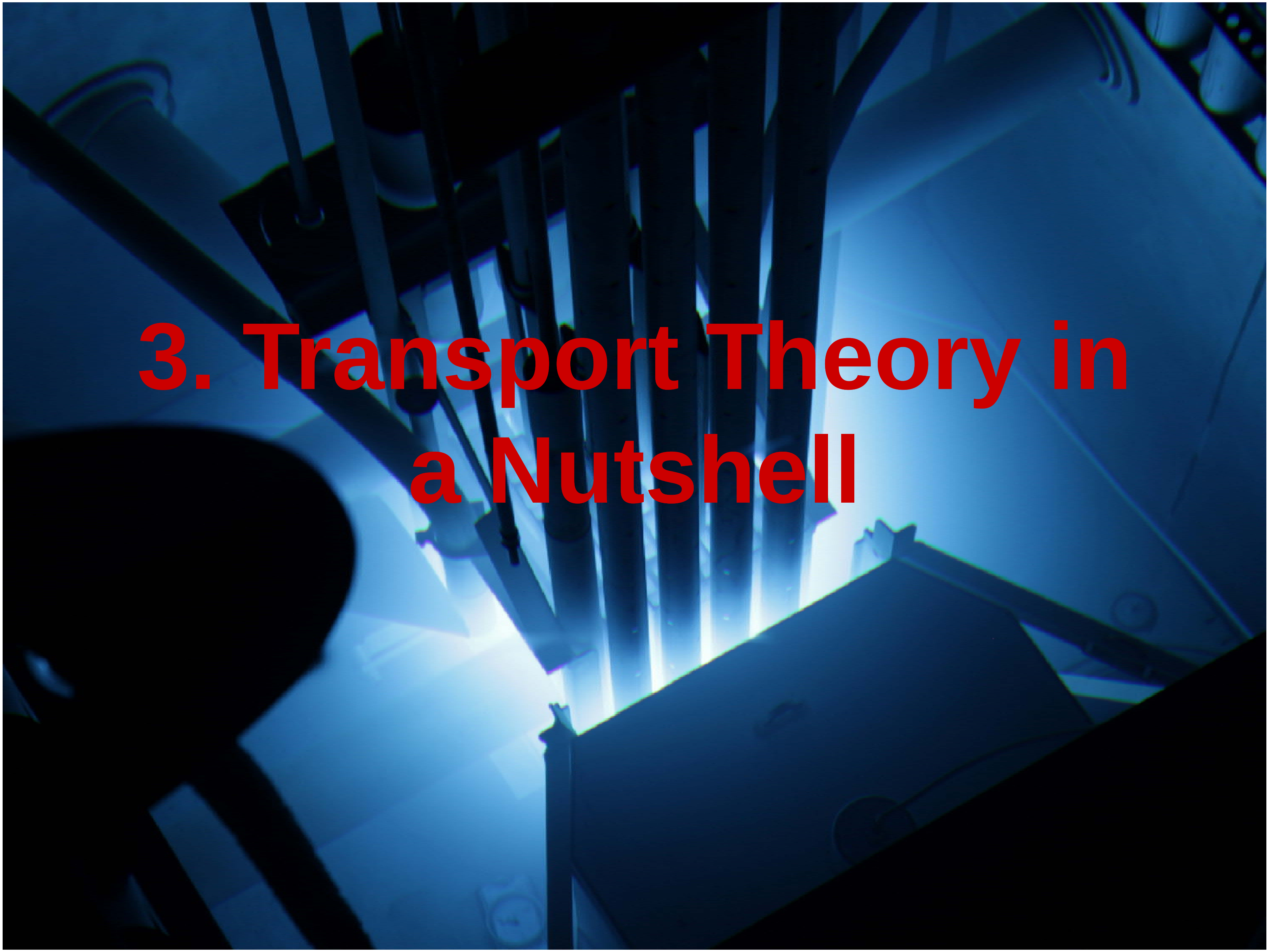
Brian Powell: MS Student
Thesis: ***Efficient Computation of Subdomain Operators Employed in the Integral Transport Matrix Method (ITMM)***

Noel Nelson: MS Student
MS Thesis: ***Accurate Holdup Calculations with Predictive Modeling & Data Integration***

2. Work Presented Today

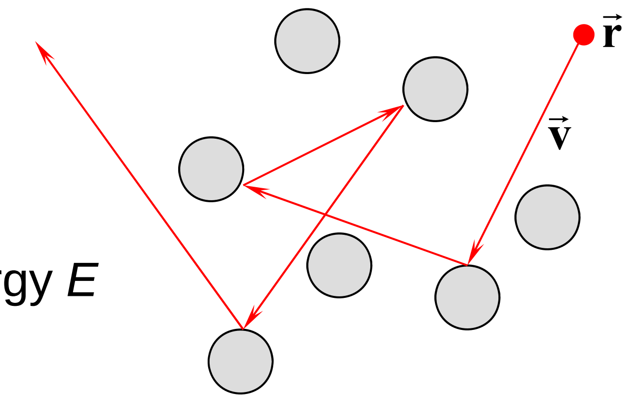
- **Dr. Joe Zerr is a former member of my research group**
- **Presently Research staff at Los Alamos National Laboratory**
- **Received PhD 2010, Penn State University**
- **Recipient of 2010 Mark Mills Award**

3. Transport Theory in a Nutshell



3. Fundamentals

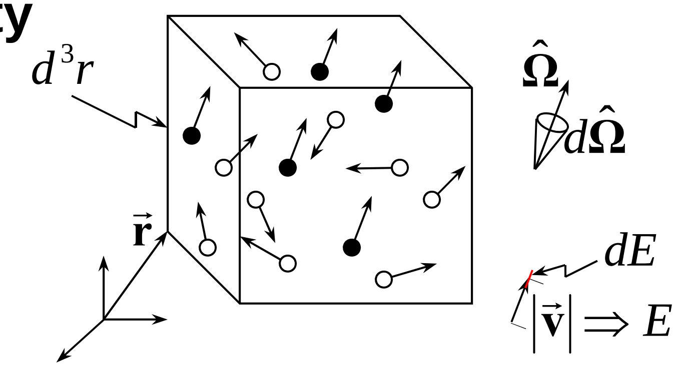
- **Classical point particle is fully described by independent variables:**
  - Time: t
  - Space: $\vec{\mathbf{r}}$
  - Direction of motion: $\hat{\mathbf{\Omega}}$
  - Energy: E
  - **Note**: the unit vector $\hat{\mathbf{\Omega}}$ & energy E are equivalent to velocity $\vec{\mathbf{v}}$



- **Observable quantities (e.g. heating) depend on reaction rate:**
  - Large number of interacting particles (neutrons/photons)
  - Much larger number of host targets (nuclei/electrons)
  - Impractical to solve dynamic system for individual particles/targets ⇨ statistical model: ***mean*** collision density rate
  - Proportional to density of interacting particles & host targets

3. Statistical Model

- **Particle Angular Density:**
  - $n(\vec{\mathbf{r}},\hat{\mathbf{\Omega}},E,t)\,d^3r\,dE\,d\hat{\mathbf{\Omega}} \equiv$ mean number of neutrons at time *t* in d^3r at $\vec{\mathbf{r}}$, with energy in [*E*, *E*+*dE*] traveling in the directional cone $d\hat{\mathbf{\Omega}}$ at $\hat{\mathbf{\Omega}}$
- **Particle Density:**
  - $N(\vec{\mathbf{r}},E,t)\,d^3r\,dE \equiv$ mean number of neutrons at time *t* in d^3r at $\vec{\mathbf{r}}$, with energy in [*E*, *E*+*dE*] ⇨ $N(\vec{\mathbf{r}},E,t) \equiv \int_{4\pi} d\hat{\mathbf{\Omega}}\, n(\vec{\mathbf{r}},\hat{\mathbf{\Omega}},E,t)$
- **Particle Flux: Speed × particle density**
  - Angular flux: $\psi(\vec{\mathbf{r}},\hat{\mathbf{\Omega}},E,t) \equiv vn(\vec{\mathbf{r}},\hat{\mathbf{\Omega}},E,t)$
  - Scalar flux: $\varphi(\vec{\mathbf{r}},E,t) \equiv vN(\vec{\mathbf{r}},E,t)$
  - Leakage rate: $\psi(\vec{\mathbf{r}},\hat{\mathbf{\Omega}},E,t)\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{n}}dA$
  - Reaction rate density: $\Sigma_j\,\varphi(\vec{\mathbf{r}},E,t)$
    - $\Sigma_j \equiv$ Probability reaction *j* / path length



Need to compute $\varphi(\vec{\mathbf{r}},E,t) = \int_{4\pi} \psi(\vec{\mathbf{r}},\hat{\mathbf{\Omega}},E,t)\,d\hat{\mathbf{\Omega}}$

3. Neutron Transport Equation

- **Special case of Boltzmann equation: First-order integro-differential**
  - Neutral particles ⇨ no electro-magnetic forces
  - Low particle densities ⇨ ignore neutron-neutron collisions ⇨ linear
- **Balance over infinitesimal element in phase space:** $(\vec{\mathbf{r}}, \hat{\mathbf{\Omega}}, E)$
- **Dependent variable: Angular flux** $\psi(\vec{\mathbf{r}}, \hat{\mathbf{\Omega}}, E, t)$

$$\underbrace{\frac{1}{v}\frac{\partial \psi(\vec{\mathbf{r}}, E, \hat{\mathbf{\Omega}}, t)}{\partial t}}_{\text{Transient}} + \underbrace{\hat{\mathbf{\Omega}} \cdot \nabla \psi(\vec{\mathbf{r}}, E, \hat{\mathbf{\Omega}}, t)}_{\text{Streaming}} + \underbrace{\Sigma_t(\vec{\mathbf{r}}, E)\psi(\vec{\mathbf{r}}, E, \hat{\mathbf{\Omega}}, t)}_{\text{Total collision}}$$

$$\text{Scattering}\left\{ = \int_{4\pi} d\hat{\mathbf{\Omega}}' \int_0^\infty dE' \Sigma_s(E', \hat{\mathbf{\Omega}}' \to E, \hat{\mathbf{\Omega}}) \psi(\vec{\mathbf{r}}, E', \hat{\mathbf{\Omega}}', t) \right.$$

$$\text{Fission}\left\{ + \frac{\chi(E)}{4\pi}\left[\int_0^\infty dE' \nu(E') \Sigma_f(\vec{\mathbf{r}}, E') \int_{4\pi} d\hat{\mathbf{\Omega}}' \psi(\vec{\mathbf{r}}, E', \hat{\mathbf{\Omega}}', t)\right] \right. + \underbrace{s(\vec{\mathbf{r}}, E, \hat{\mathbf{\Omega}}, t)}_{\text{Fixed source}}$$

3. Interface & Boundary Conditions

- **Steady state: Time derivative vanishes**
- **Interface condition: Angular flux continuous along direction of motion, $\hat{\mathbf{\Omega}}$, across material boundaries**
- **Physical intuition: Can specify what goes into a system**
  - What comes out is a consequence of the transport process inside
  - Example: shining light into crystal
    - Can choose color/intensity of incoming light
    - Can't choose color/intensity of outgoing light: depends on what happens inside
- **Typical Boundary Condition (BC):**
  - Set ***incoming*** flux $\psi(\vec{\mathbf{r}}_s, E, \hat{\mathbf{\Omega}}, t) = \psi_{in}(\vec{\mathbf{r}}_s, E, \hat{\mathbf{\Omega}}, t)$ for:
    - All times *t*
    - All energies: $E \in [0, \infty]$
    - Each $\vec{\mathbf{r}}_s$ on the boundary *S*
    - Each incoming angle: $\hat{\mathbf{\Omega}} \cdot \hat{\mathbf{e}}_s < 0, \hat{\mathbf{e}}_s$ is the normal unit vector pointing out
  - The function $\psi_{in}(\vec{\mathbf{r}}_s, E, \hat{\mathbf{\Omega}}, t)$ can be specified explicitly or implicitly
    - Vacuum BC: $\psi_{in}(\vec{\mathbf{r}}_s, E, \hat{\mathbf{\Omega}}, t) = 0$

3. Discretization of Transport Equation

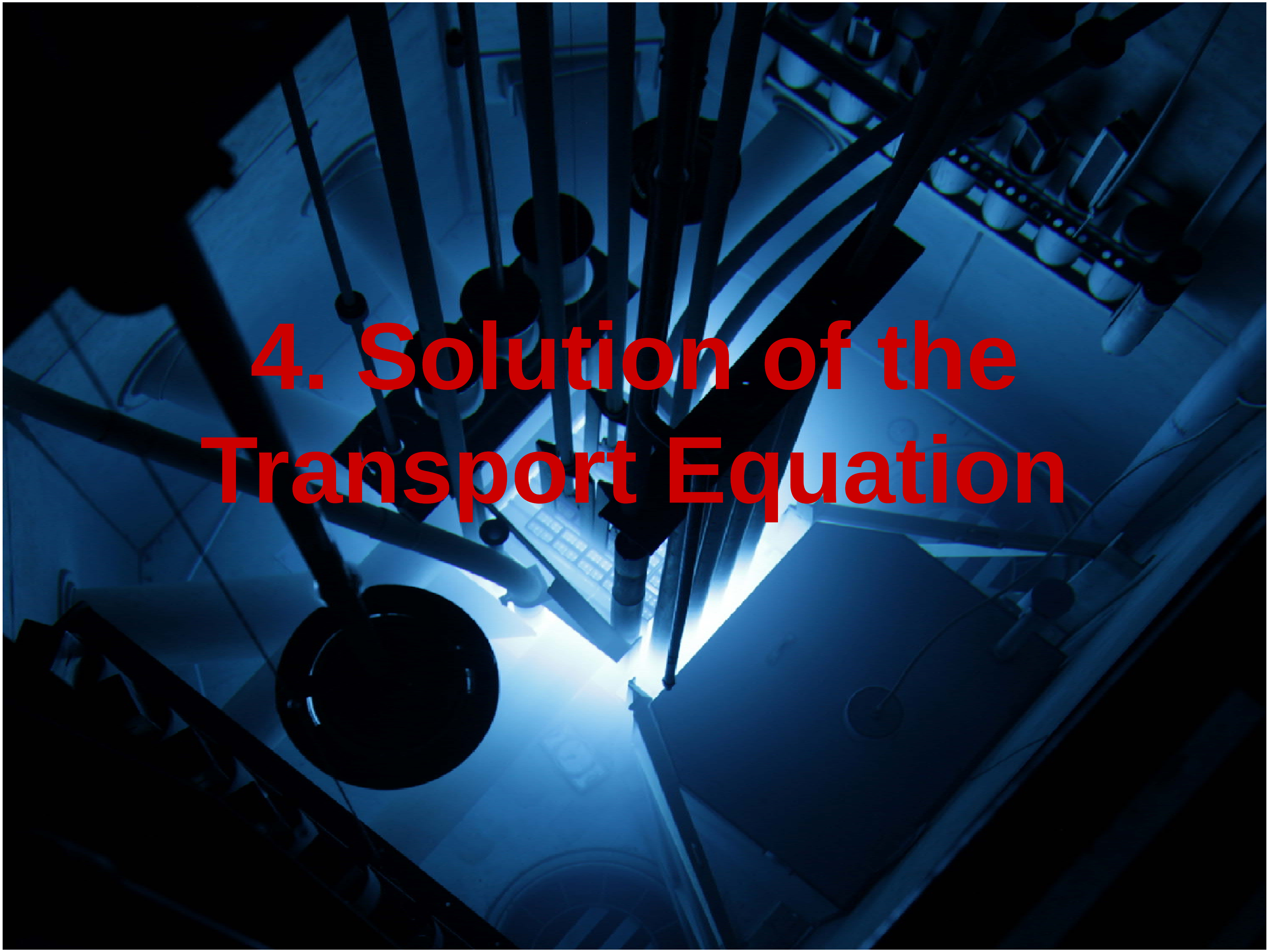
- ❑ **Implementation on digital computer ⇨ discretize independent variables & consequently dependent variables**
- ❑ **Energy: Multigroup ⇨ discretization into *bins* (E_g , E_{g-1})**
  - ❖ Victory, 1985: Total & scattering cross section fluctuations diminish with refinement of energy group structure

    ⇨ Multigroup solution → exact solution
- ❑ **Angle: Discrete-ordinates ⇨ discretization along discrete $\hat{\Omega}_n$**
  - ❖ Madsen, 1971: Quadrature formula converges with increasing order

    ⇨ Discrete Ordinates solution → exact one-speed solution
- ❑ **Space: Multitude of methods discretize $\nabla\psi$ on spatial mesh**
  - ❖ Madsen, 1972: Exact solution has bounded 3rd derivatives

    ⇨ Diamond Difference solution → exact Discrete Ordinates solution
  - ❖ Smoothness hypothesis unrealistic for most applications

4. Solution of the Transport Equation



4. Traditional Solution Algorithms

- **Difficulty of solving the transport equation (partial list):**
  - Steady-state 3-D problems: phase space is 6-D ⇨ 100 discrete variables per phase space dimension yields 10^{12} unknowns
  - Neutron cross sections sensitive to energy & nuclide composition
  - Source (fission & scattering) depends on solution ψ ⇨ iterate
- **Nested loops:**
  - Outer Iteration: converge fission/scattering source
  - Loop over energy groups: from highest to lowest *E*
  - In each group sum fission + inscattering source guess into *q*
  - Inner (or Source) Iteration: reconcile within group source & flux
    - Starting with guess for group φ within-group scattering source
    - Invert 1^{st} order PDE on source distribution ⇨ group ψ
    - Integrate over angle ⇨ new φ: test if too different from starting guess
    - Yes: Repeat unless already used too many iterations
    - No: solution converged successfully to group flux for into-group sources

4. Cell Equations

- **Kernel operation: Solving 1st order PDE for given source**
  - Conducted via *mesh sweep* algorithm: 1 cell, 1 angle at a time
- **Discretized equations per energy group/angle/cell:**
  - Balance: Include fission & inscattering from other groups in q_n

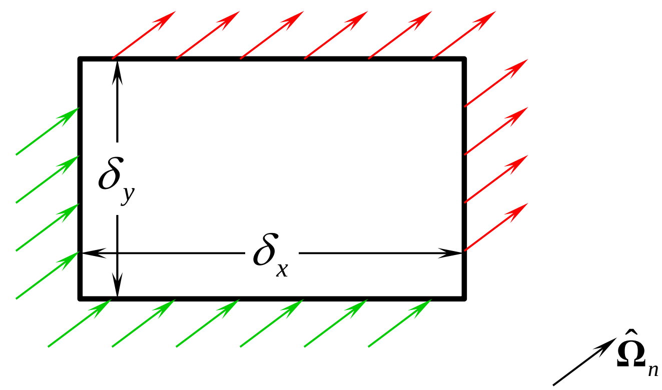
$$\frac{\mu_n}{\delta_x}\left(\psi^x_{n,out}-\psi^x_{n,in}\right)+\frac{\eta_n}{\delta_y}\left(\psi^y_{n,out}-\psi^y_{n,in}\right)+\frac{\xi_n}{\delta_z}\left(\psi^z_{n,out}-\psi^z_{n,in}\right)+\sigma_t\,\overline{\psi}_n=\sigma_s\,\overline{\varphi}+q_n$$

  - Auxiliary: Method-dependent, simplest is Diamond Difference

$$\overline{\psi}_n=\frac{1}{2}\left(\psi^u_{n,out}+\psi^u_{n,in}\right),\quad u=x,y,z$$

  - Quadrature formula:

$$\overline{\varphi}=\sum_{n=1}^{N}\omega_n\overline{\psi}_n$$



4. Cell Equations

- **Kernel operation: Solving 1st order PDE for given source**
  - Conducted via *mesh sweep* algorithm: 1 cell, 1 angle at a time
- **Discretized equations per energy group/angle/cell:**
  - Balance: Include fission & inscattering from other groups in q_n

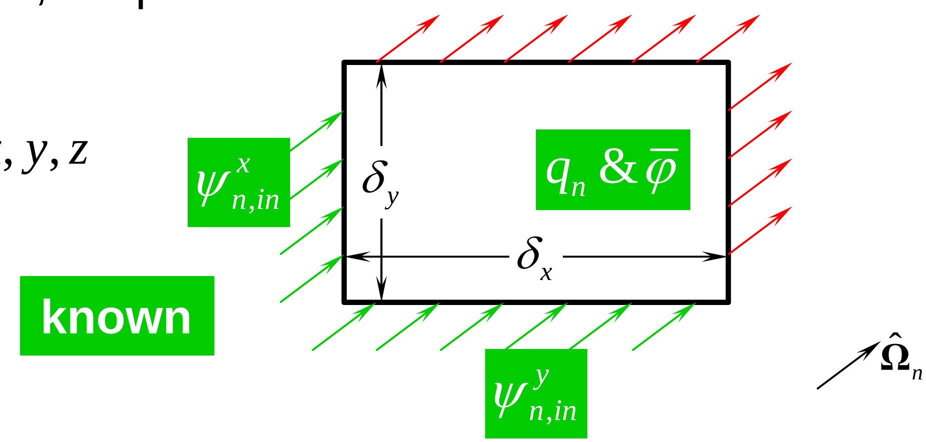
$$\frac{\mu_n}{\delta_x}\left(\psi^x_{n,out}-\psi^x_{n,in}\right)+\frac{\eta_n}{\delta_y}\left(\psi^y_{n,out}-\psi^y_{n,in}\right)+\frac{\xi_n}{\delta_z}\left(\psi^z_{n,out}-\psi^z_{n,in}\right)+\sigma_t\,\overline{\psi}_n=\sigma_s\,\overline{\varphi}+q_n$$

  - Auxiliary: Method-dependent, simplest is Diamond Difference

$$\overline{\psi}_n=\frac{1}{2}\left(\psi^u_{n,out}+\psi^u_{n,in}\right),\quad u=x,y,z$$

  - Quadrature formula:

$$\overline{\varphi}=\sum_{n=1}^{N}\omega_n\overline{\psi}_n$$



4. Cell Equations

- **Kernel operation: Solving 1st order PDE for given source**
  - Conducted via *mesh sweep* algorithm: 1 cell, 1 angle at a time
- **Discretized equations per energy group/angle/cell:**
  - Balance: Include fission & inscattering from other groups in q_n

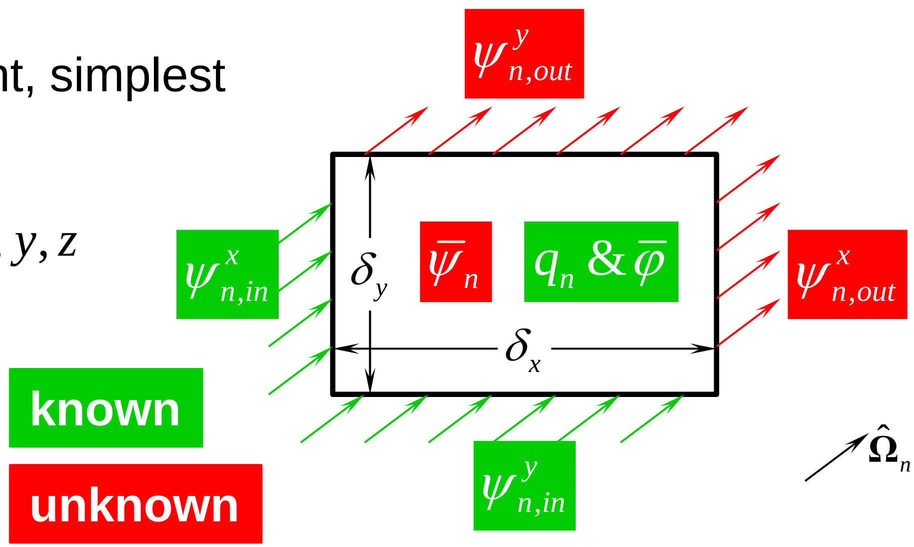
$$\frac{\mu_n}{\delta_x}\left(\psi^x_{n,out}-\psi^x_{n,in}\right)+\frac{\eta_n}{\delta_y}\left(\psi^y_{n,out}-\psi^y_{n,in}\right)+\frac{\xi_n}{\delta_z}\left(\psi^z_{n,out}-\psi^z_{n,in}\right)+\sigma_t\,\overline{\psi}_n=\sigma_s\,\overline{\varphi}+q_n$$

  - Auxiliary: Method-dependent, simplest is Diamond Difference

$$\overline{\psi}_n=\frac{1}{2}\left(\psi^u_{n,out}+\psi^u_{n,in}\right),\quad u=x,y,z$$

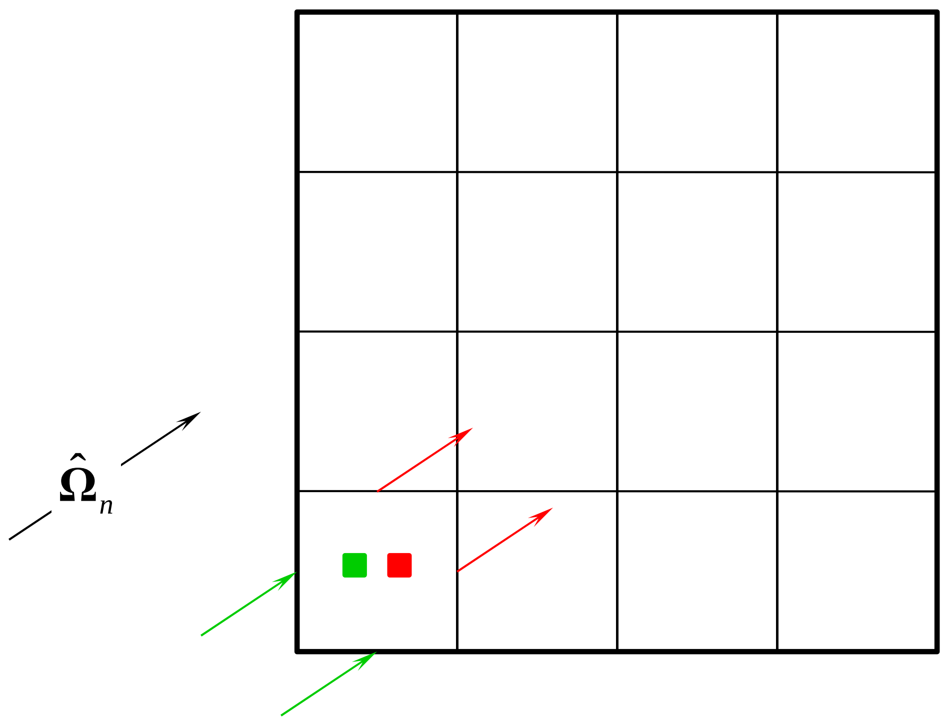
  - Quadrature formula:

$$\overline{\varphi}=\sum_{n=1}^{N}\omega_n\,\overline{\psi}_n$$



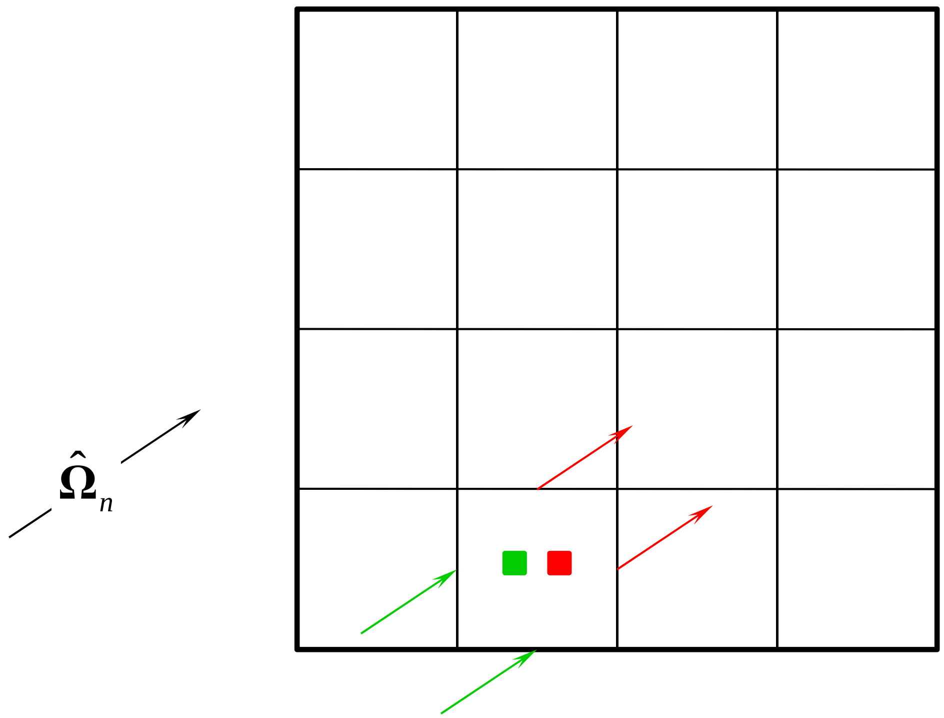
4. Mesh Sweep

- **For each discrete ordinate sweep cells along $\hat{\Omega}_n$**

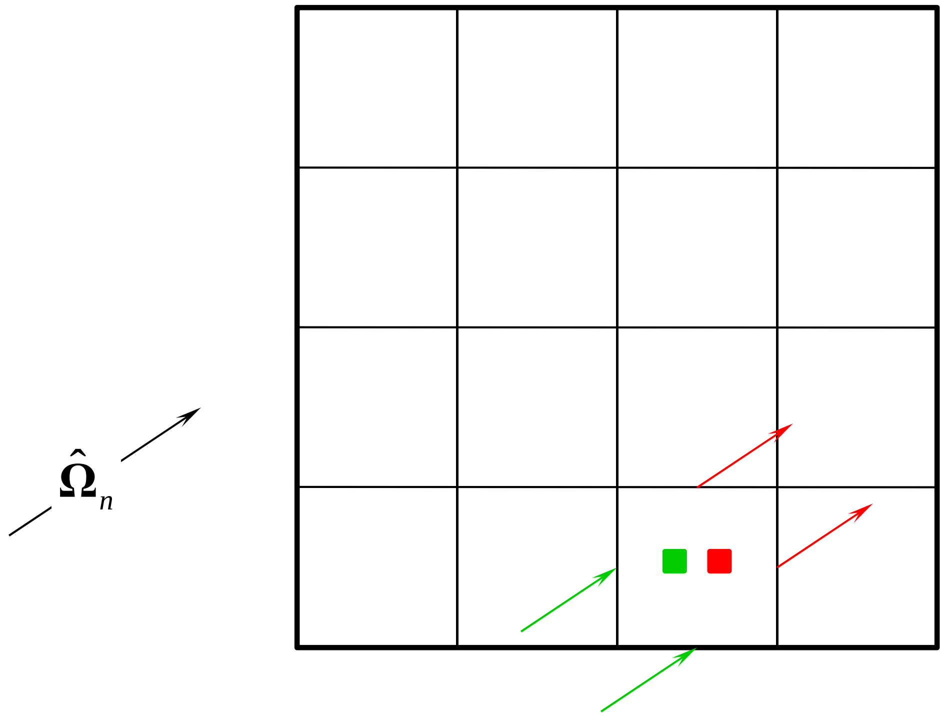


4. Mesh Sweep

- Interface angular fluxes couple neighboring cells

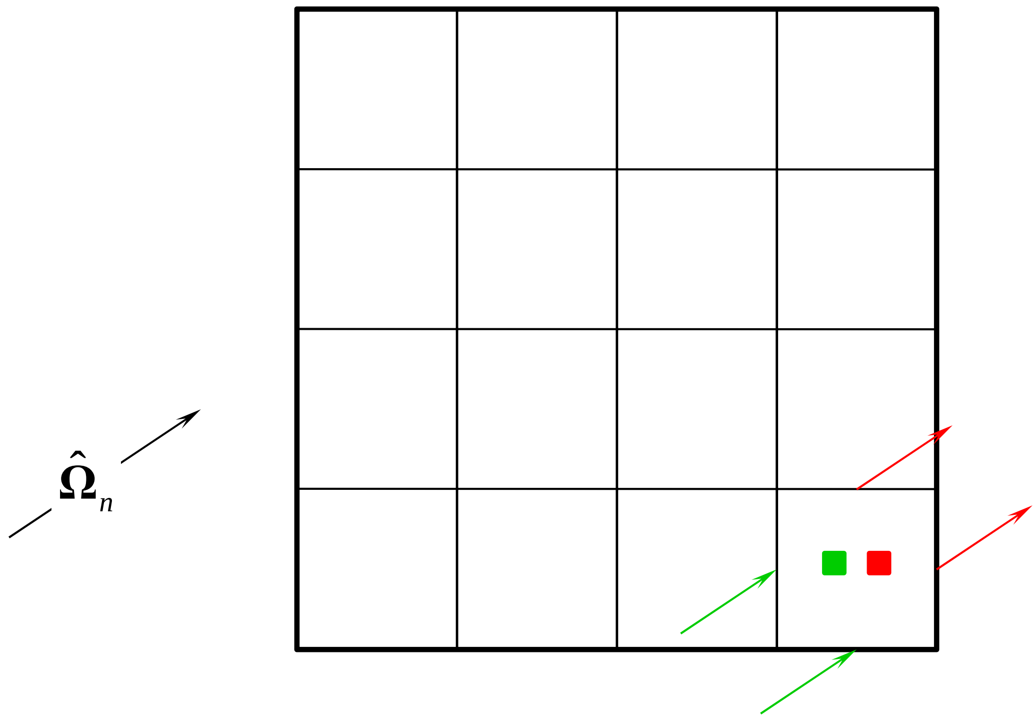


4. Mesh Sweep



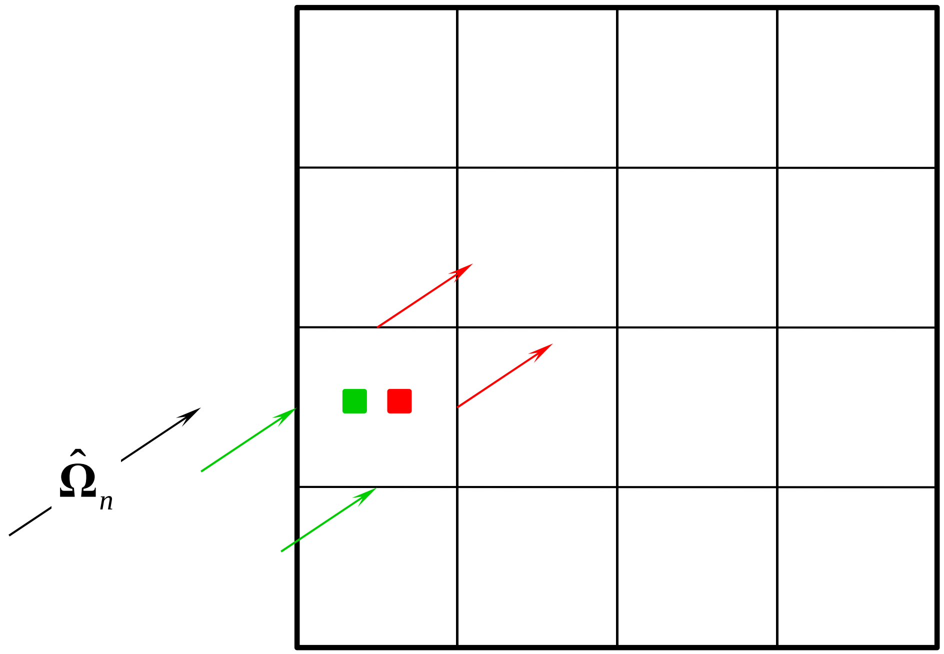
4. Mesh Sweep

- **Upon reaching end of row go to next row along $\hat{\Omega}_n$**



4. Mesh Sweep

- ❑ **Note sequential nature: must compute upstream cell first**



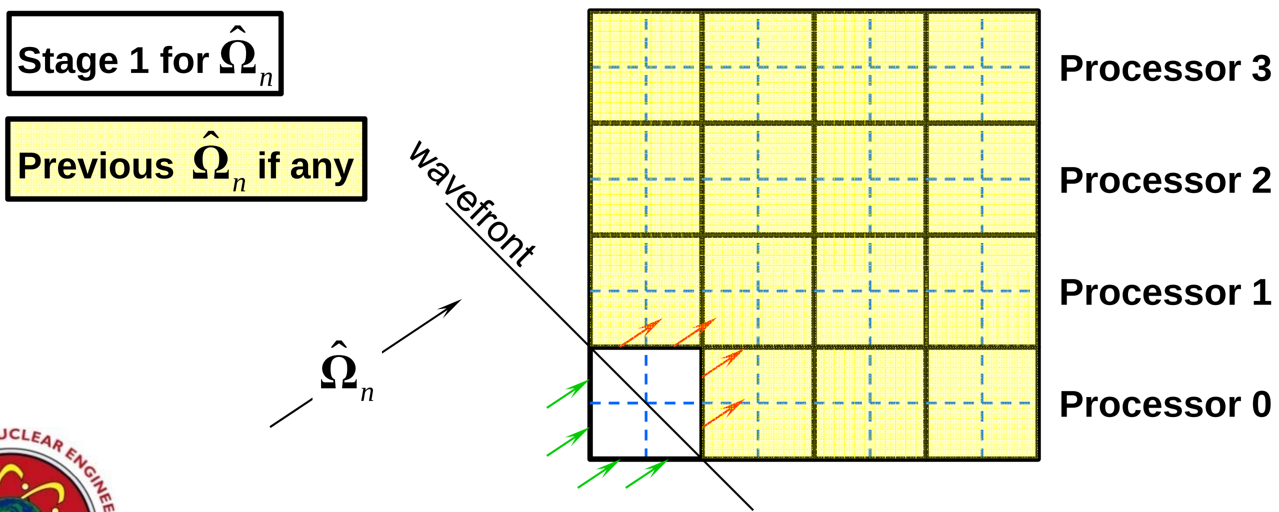
- ❑ **Slow convergence if SI often demands acceleration (DSA)**

4. Parallelization Strategies

- ***Domain decomposition*: split range of phase space variable in *P* subdomain & assign each to different process**
  - Perfect parallelization ⇨ reduce execution time by factor *P*
  - ***Synchronous*** DD: processes fully independent ⇨ computation (e.g. number of iterations) independent of *P*
  - ***Asynchronous*** DD: coupled processes ⇨ work *P* dependent
- **Possible DD for S_N algorithms:**
  - Energy: speedup limited by # groups (hundreds at most)
    - Coarse-grain ⇨ easy to implement & high parallel efficiency
    - Asynchronous & pointless unless there is fission &/or upscattering
  - Angle: speedup limited by # angles (few thousands at most)
    - Medium-grain ⇨ easy to implement & good parallel efficiency
    - Synchronous in Cartesian coordinate system
  - Spatial: speedup limited by # cells (potentially many millions)
    - Fine-grain ⇨ difficult to implement & OK parallel efficiency
    - Synchronous implies retaining sequential order among subdomains

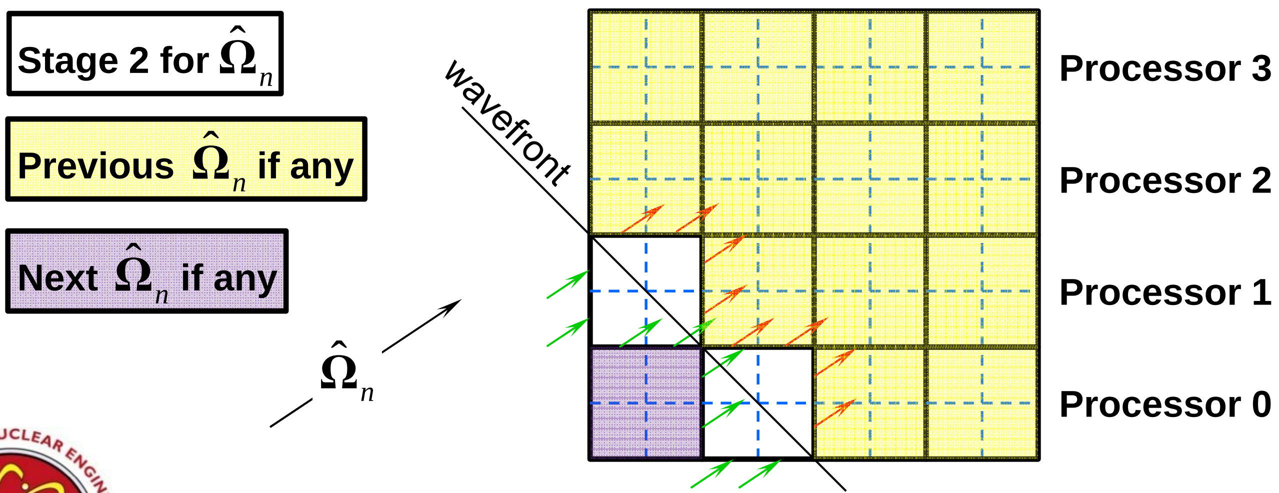
4. Spatial Domain Decomposition

- **Koche- Baker-Alcouffe SDD: synchronous**
  - Maps 3D mesh onto 2D processor topology
  - Sweep mesh by subdomain in natural sequence per $\hat{\mathbf{\Omega}}_n$
  - Concurrently sweep ready subdomains (on waverfront) ➪ SDD
  - Pipeline angles to reduce processor idleness
  - Communicate outgoing interface ψ to neighbors across wavefront



4. Spatial Domain Decomposition

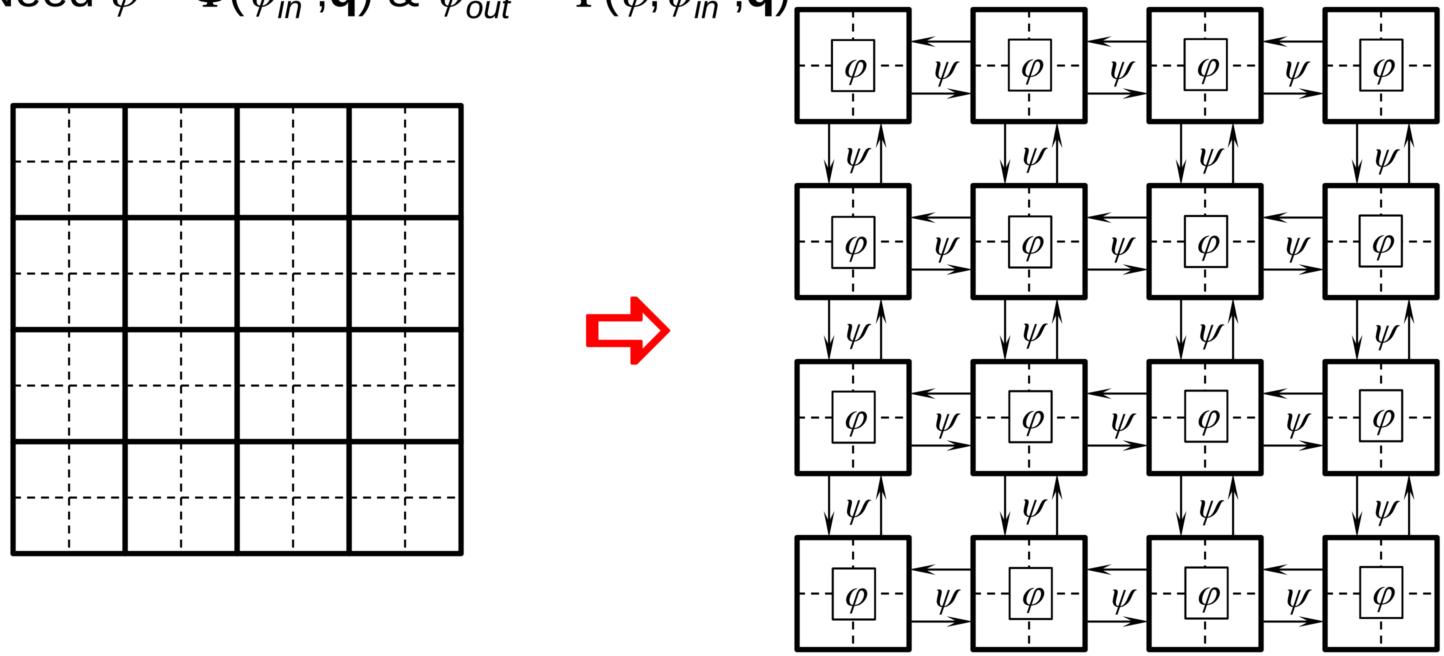
- **Koche- Baker-Alcouffe SDD: synchronous**
  - Maps 3D mesh onto 2D processor topology
  - Sweep mesh by subdomain in natural sequence per $\hat{\Omega}_n$
  - Concurrently sweep ready subdomains (on waverfront) ⇨ SDD
  - Pipeline angles to reduce processor idleness
  - Communicate outgoing interface ψ to neighbors across wavefront



5. Alternative SDD Parallelizations

5. Parallel Block Jacobi (PBJ)

- **Asynchronous SDD alternative to KBA:**
  - Eliminate processor idleness & increase concurrent processes
  - Combine all angles computations in subdomain via ITMM operators
  - Replace SI with iterations on subdomain interface fluxes: Communicate ψ between iterations
  - Need $\varphi = \Phi(\psi_{in}, \mathbf{q})$ & $\psi_{out} = \Psi(\varphi, \psi_{in}, \mathbf{q})$



5. Outline of ITMM

- **Express SI as a mapping of flux ℓ iterate into ℓ+1 iterate:**

$$\varphi^{(\ell+1)} = \mathbf{J}_\varphi \left(\varphi^{(\ell)} + \Sigma_s^{-1} \mathbf{q} \right) + \mathbf{K}_\varphi \psi_{in} \quad \Rightarrow \quad \mathbf{J}_\varphi = \partial \varphi^{(\ell+1)} \big/ \partial \varphi^{(\ell)}$$

  - Upon iterative convergence:

$$\left(\mathbf{I} - \mathbf{J}_\varphi \right) \varphi^\infty = \mathbf{J}_\varphi \Sigma_s^{-1} \mathbf{q} + \mathbf{K}_\varphi \psi_{in} \quad \rightarrow \quad \mathbf{\Phi}\left(\psi_{in}, \mathbf{q} \right)$$

  - For full domain where ψ_{in} is known from BCs:

$$\psi_{out} = \mathbf{J}_\psi \left(\varphi^\infty + \Sigma_s^{-1} \mathbf{q} \right) + \mathbf{K}_\psi \psi_{in} \quad \rightarrow \quad \mathbf{\Psi}\left(\varphi, \psi_{in}, \mathbf{q} \right)$$

- **Apply to subdomain: ψ_{in} is not known requires iteration**

- **ITMM operators are response matrices:**
  - $\mathbf{J}_\varphi$: cell-averaged scalar flux due to unit distributed source
  - $\mathbf{K}_\varphi$: cell-averaged scalar flux due to unit incident angular flux
  - $\mathbf{J}_\psi$: outgoing angular flux due to unit distributed source
  - $\mathbf{K}_\psi$: outgoing angular flux due to unit incident angular flux

5. Construction of ITMM Operators

- **Differential Mesh Sweep (DMS): concurrently in all subdomains**
  - Perform single sweep to compute elements of $\mathbf{J}_\varphi$ via $\mathbf{J}_\varphi = \partial\varphi^{(\ell+1)} / \partial\varphi^{(\ell)}$
  - Compute elements of other operators along the way

- **Dense operators: memory limitations as size grows**
  - Operators sizes grow superlinear with # cells, linear with # angles
  - Full coupling of cell- and face-fluxes
  - Expensive to invert for large subdomains

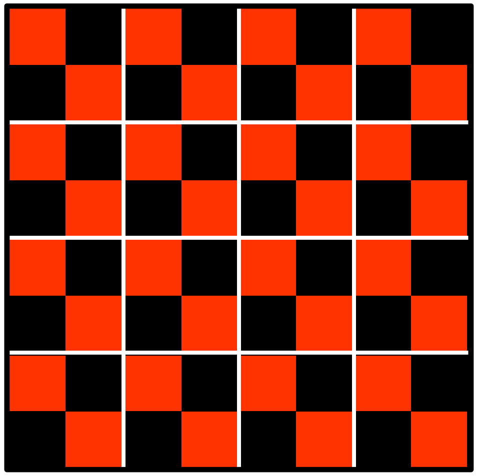
- **Applicable to high-order spatial discretizations & anisotropic scattering (φ ⇨ angular moments)**

5. PBJ Algorithm

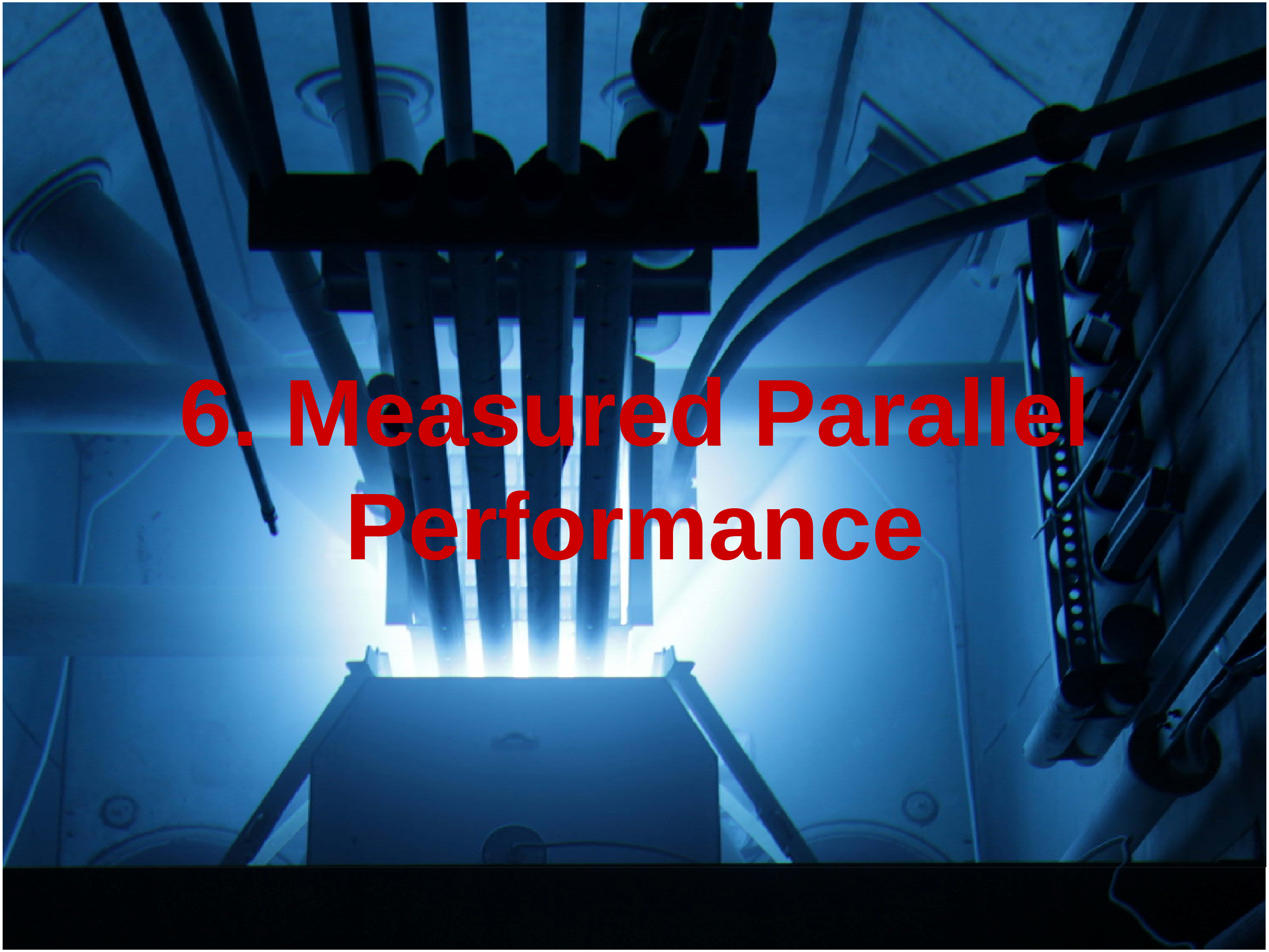
- **For each energy group, fully concurrent:**
  - Perform DMS per subdomain/processor ⇨ 4 ITMM operators
  - Need to effect $(\mathbf{I}-\mathbf{J}_{\varphi})^{-1}$ ⇨ LU factorization only once then store
  - Execute PBJ iterations on subdomain interface angular fluxes:
    - Given $\psi_{in}^{(\lambda)}$ compute for each subdomain $\varphi^{(\lambda+1)} = \mathbf{\Phi}\left(\psi_{in}^{(\lambda)}, \mathbf{q}\right)$
    - Test convergence of scalar flux: $\left|1-\varphi^{(\lambda+1)}/\varphi^{(\lambda)}\right|$ small?
      - If converged go to next group (if any)
      - Otherwise start new iteration with: $\psi_{out}^{(\lambda+1)} = \mathbf{\Psi}\left(\varphi^{(\lambda+1)}, \psi_{in}^{(\lambda)}, \mathbf{q}\right)$
      - Communicate $\psi_{out}^{(\lambda+1)}$ as $\psi_{in}^{(\lambda+1)}$ to neighboring subdomains
- **Observations:**
  - Reduced local computations to matrix-vector multiplies & adds
  - Sources of parallel inefficiency:
    - Increasing # iterations with *P* & tighter subdomain coupling: *c*↑ & Σ_t *h*↓
    - Network contention: Communicate subdomain interface angular fluxes

5. Parallel Gauss-Seidel (PGS)

- **Red/Black splitting of each subdomain:**
  - Each sub-subdomain is either **red** or **black**
  - Operations per global iteration over interface angular flux:
    - Solve local ITMM system for φ & ψ_{out}
    - Copy/send $\psi_{out} \rightarrow \psi_{in}$ to intra-/iter-subdomain neighbor
    - Solve local ITMM system for φ & ψ_{out}
    - Copy/send $\psi_{out} \rightarrow \psi_{in}$ to intra-/iter-subdomain neighbor
  - Pros:
    - Memory requirement ↓ super-linearly with ↓ number of cells
    - Typically Gauss-Seidel convergence rate better than Block Jacobi
  - Cons:
    - Smaller ITMM subdomains ⇨ slower convergence of global iterations



6. Measured Parallel Performance

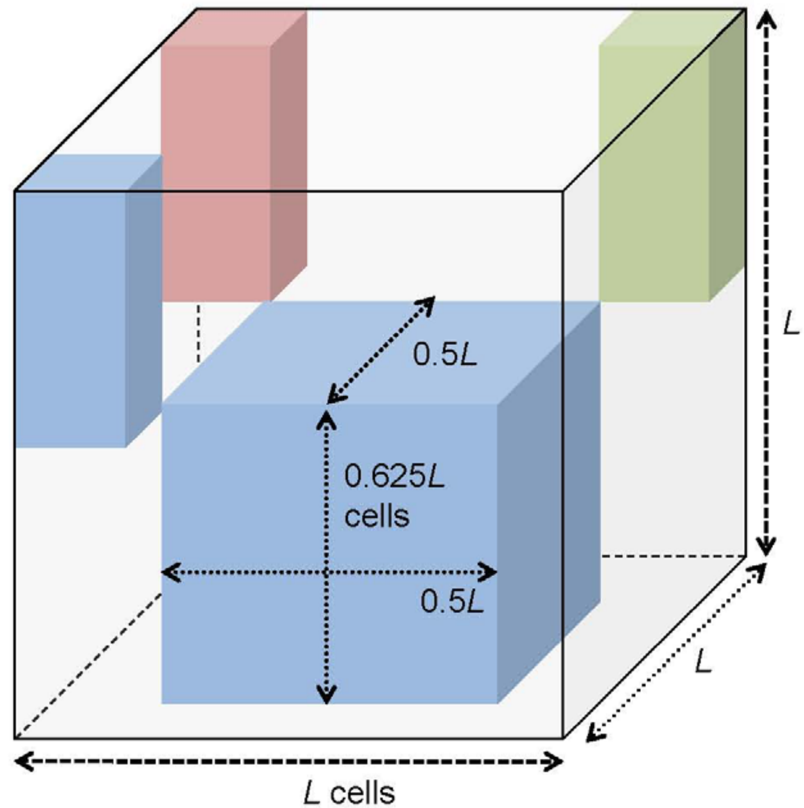


6. Implementation in PIDOTS

- **Implemented PBJ & PGS in *Parallel Integral Discrete Ordinates Transport Solver* (PIDOTS)**
- **All tests performed with one-group, DD, with isotropic scattering**
- **Preliminary testing on:**
  - LANL's Yellowrail:
    - 139 compute nodes each with 8 Processing Elements (PE) & 16 GB memory
    - Runs up to P = 256 processes
  - LANL's Redtail:
    - 1,834 compute nodes each with 8 PEs & 32 GB memory
    - Runs up to P = 1,024
  - ORNL's JaguarPF: All results here are for this platform
    - 18,688 compute nodes each with 12 PEs & 16 GB memory
    - Runs up to P = 32,768

6. Weak Scaling Tests

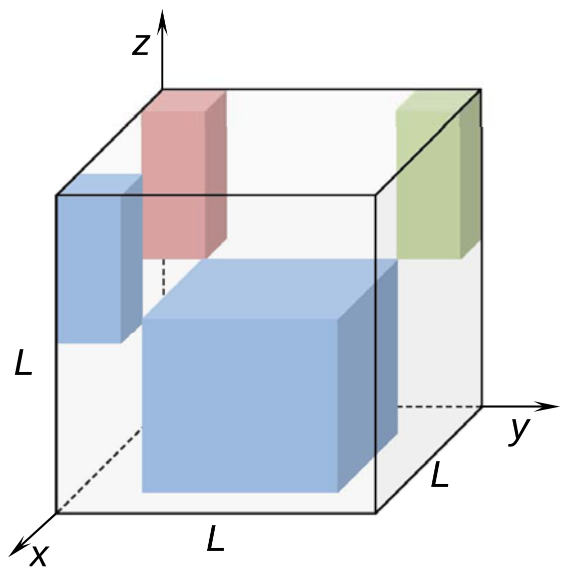
- **Evaluate parallel performance as problem size grows with *P***
- **Weak scaling ⇨ Number of cells per processor fixed:**
  - Start with *L* x *L* x *L* cubic-cells domain
  - Comprised of 4 materials: no symmetries
  - Examine effect on number of iterations ⇨ execution time:
    - Cell size *h* set to {0.1, 1.0, 10.0} cm
    - Scattering ratio *c* set to {0.9, 0.99}
  - Choice of *L* & S_N order is memory-limited



Dept of Nuclear Engineering
NC State University

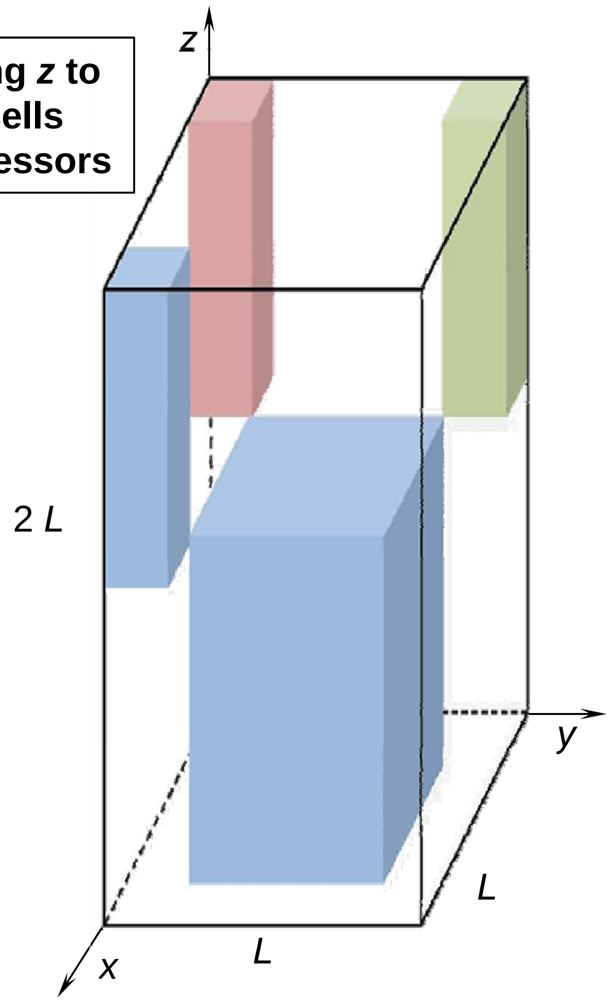
6. Growing Problem Size

Starting with the base case:
***L* x *L* x *L* cells**
on *P* processors



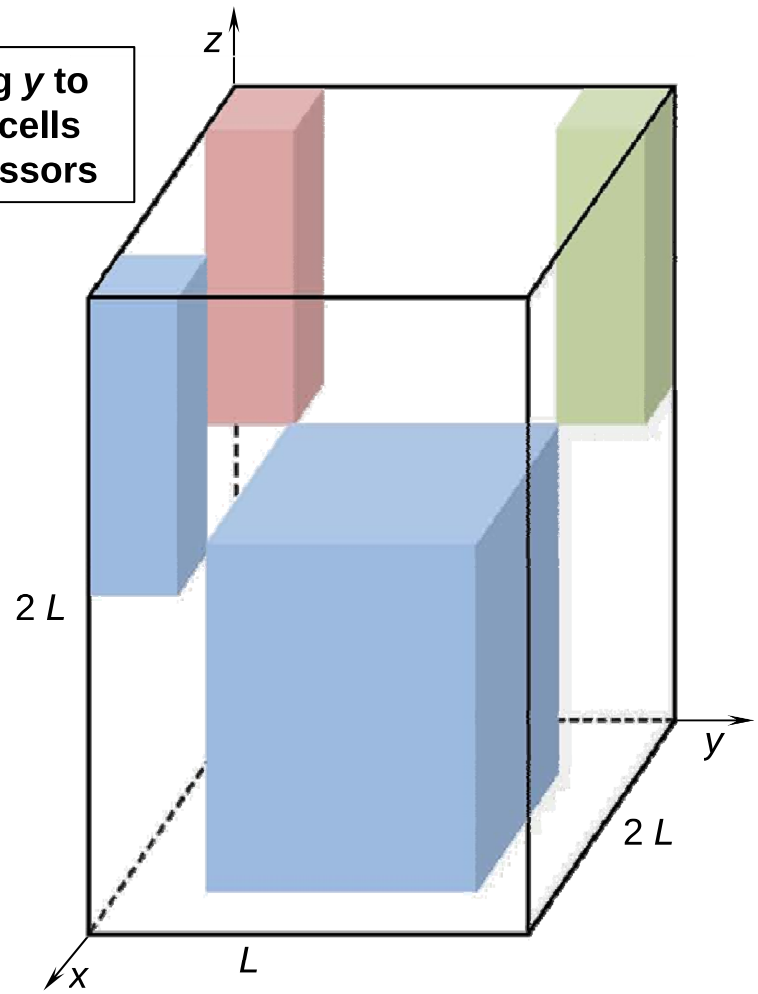
6. Growing Problem Size

Stretch along *z* to
***L* x *L* x 2 *L* cells**
on 2 *P* processors



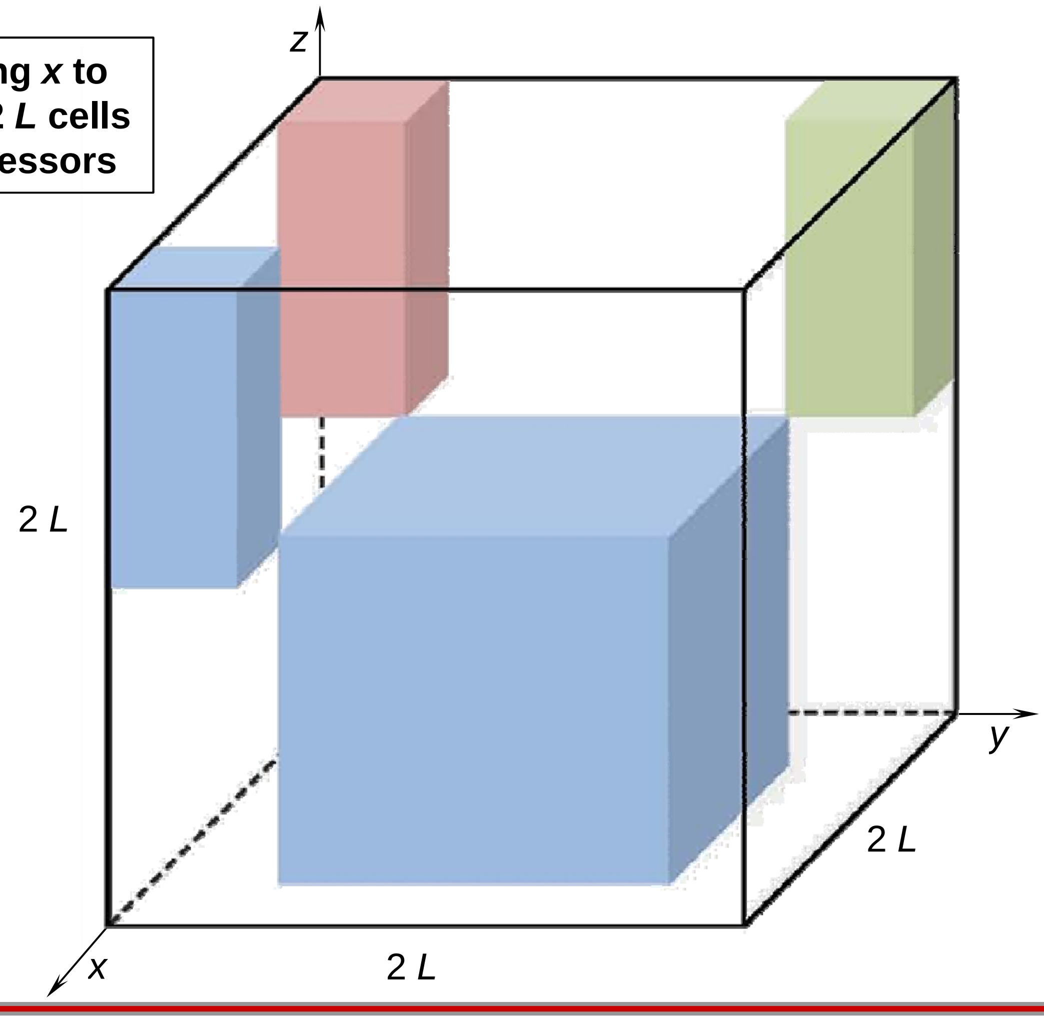
6. Growing Problem Size

Stretch along *y* to *L* x 2 *L* x 2 *L* cells on 4 *P* processors



6. Growing Problem Size

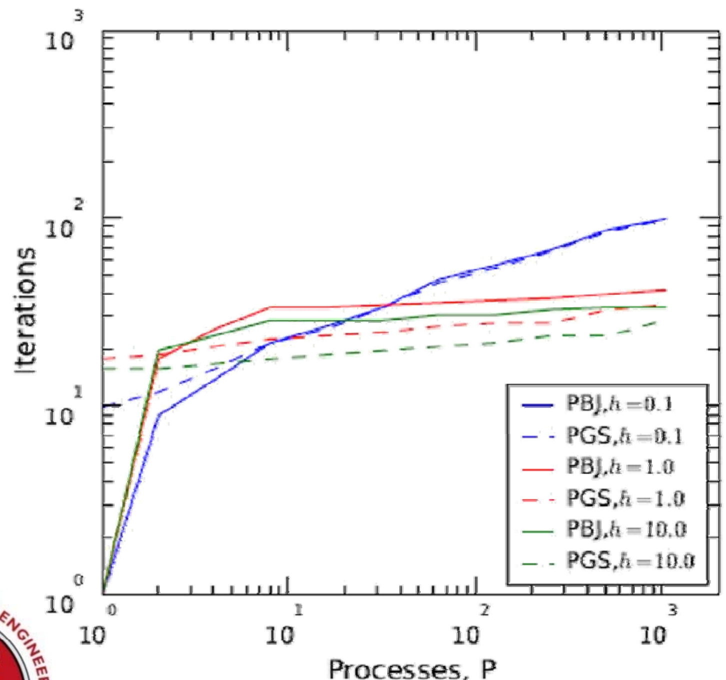
Stretch along *x* to
2 *L* x 2 *L* x 2 *L* cells
on 8 *P* processors

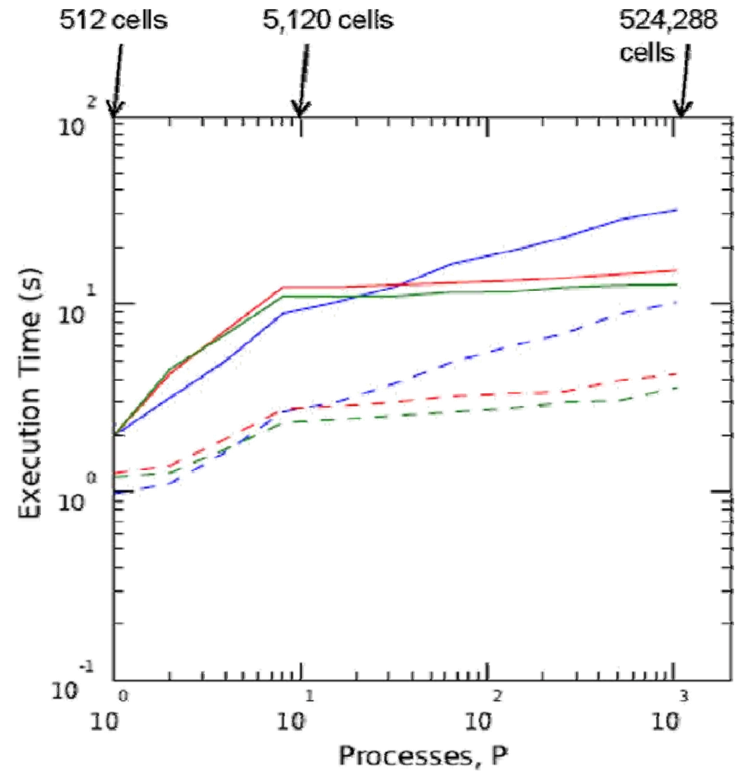


6. PBJ *vs* PGS Performance

- **8x8x8-cell model per *P*, S_{16}, *P* up to 1,024 on JaguarPF**
- **PGS: eight 4x4x4-cell sub-subdomains per *P* ⇨ shorter construction & per-iteration times**

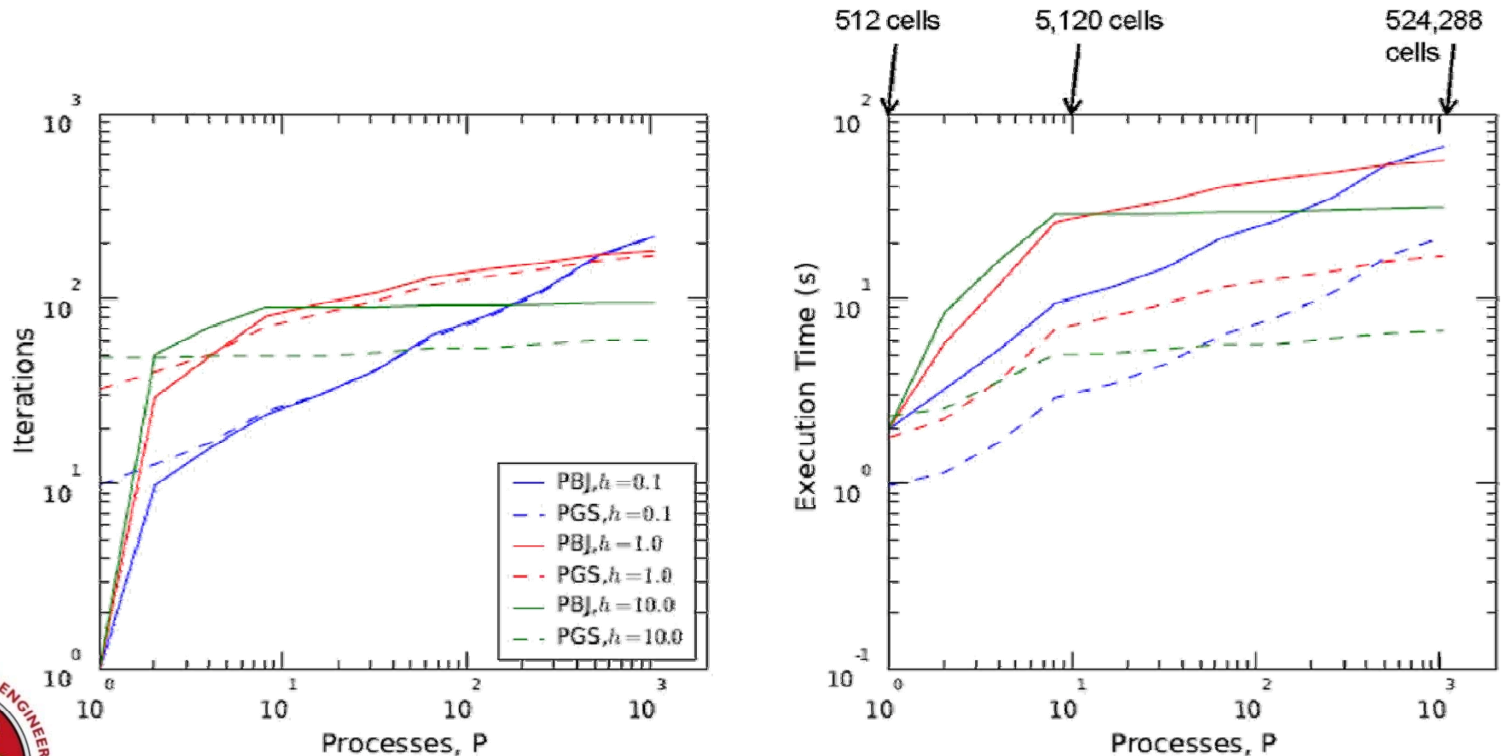
c = 0.9 results:





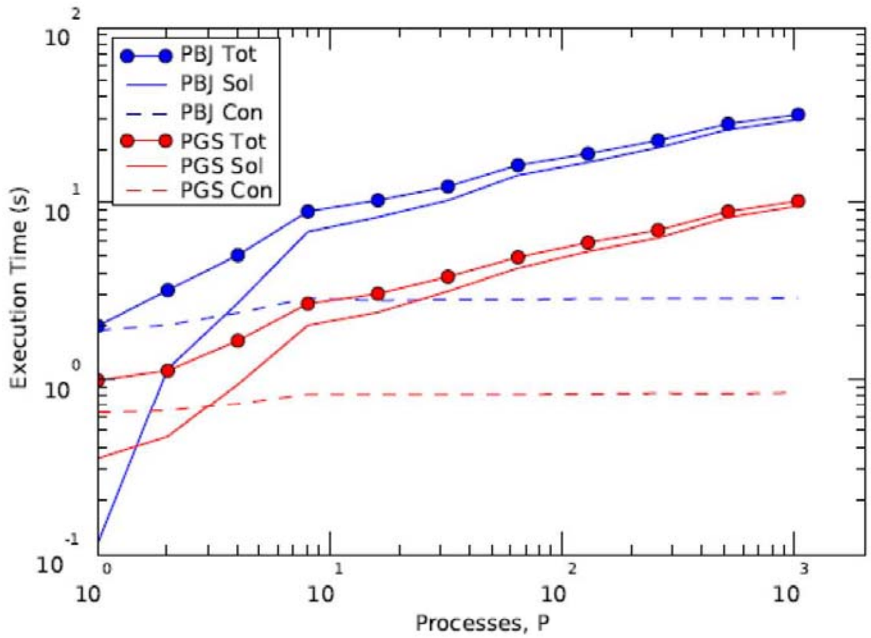
6. PBJ *vs* PGS Performance

c = 0.99 results:



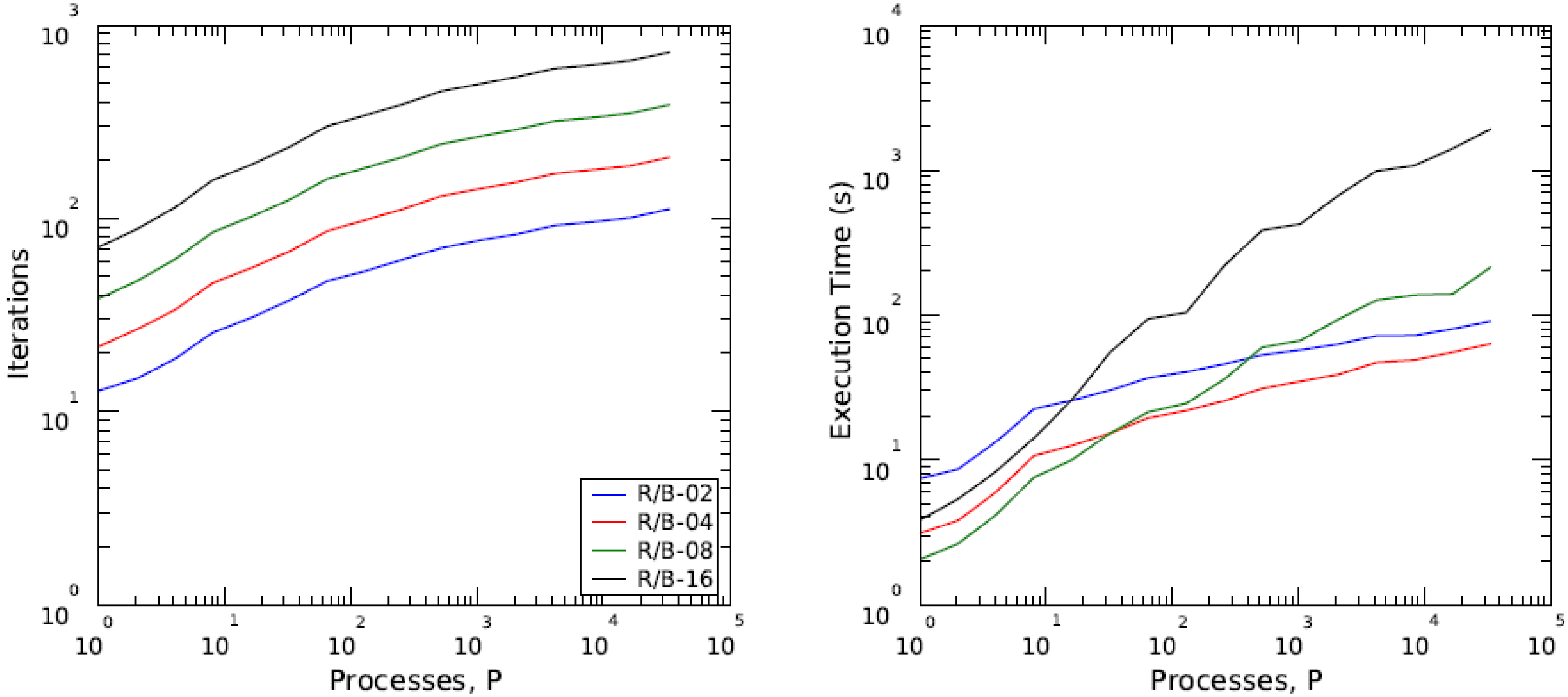
6. Construction *vs* Solution Time

- **Total execution time = construction time + iterative solution time:**
  - Construction time: independent of *c* & *h* ⇨ average over all cases
  - Iterative solution depicted for *c* = 0.9 & *h* = 0.1 cm as example



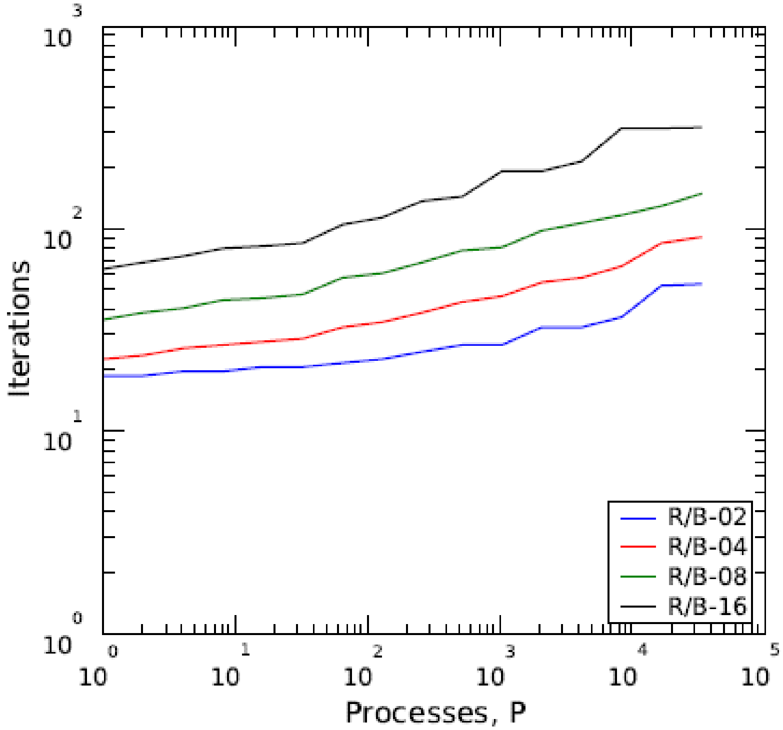
Weak Scaling Results

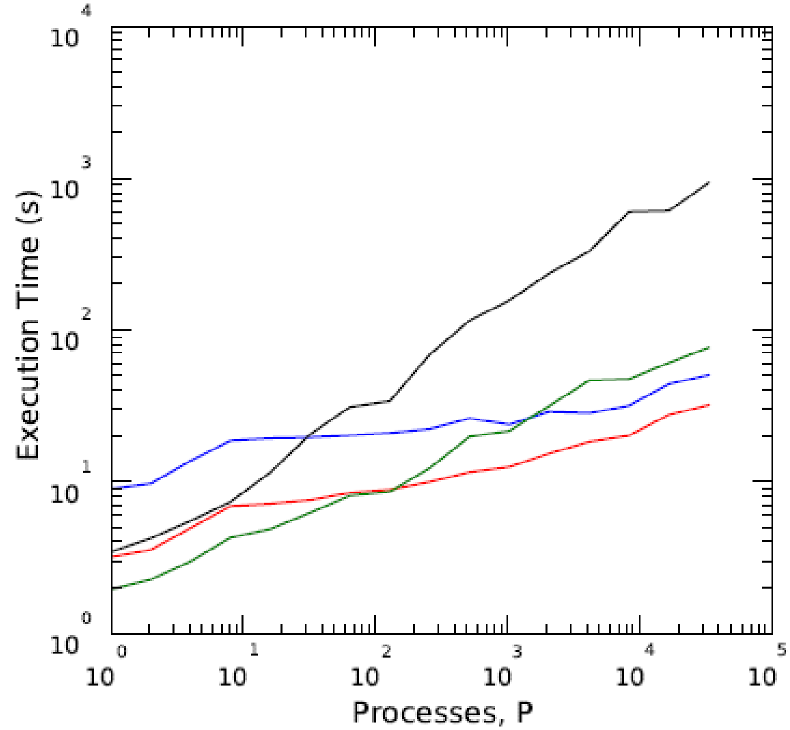
- c=0.9, h=0.1



Weak Scaling Results

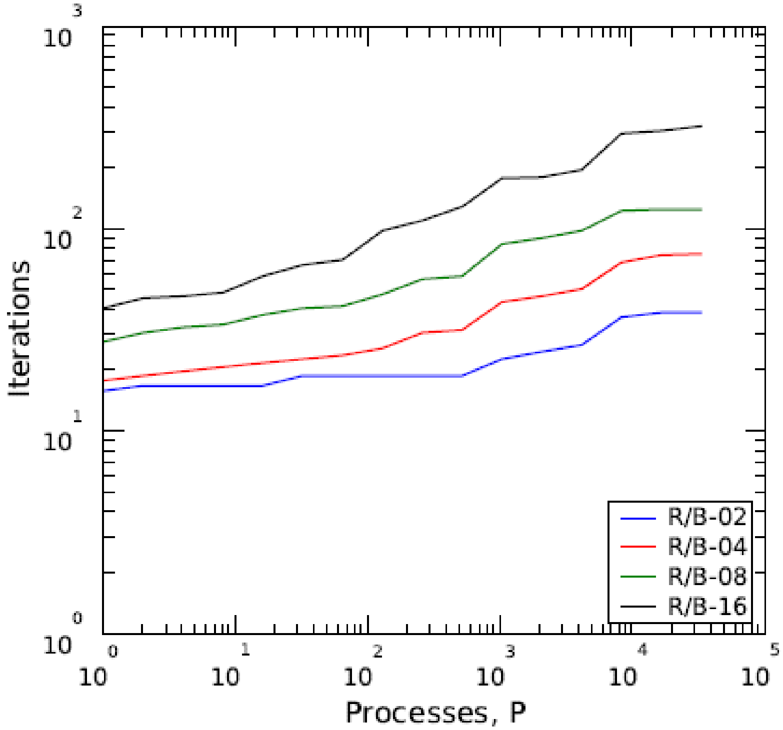
- **c=0.9, h=1.0**

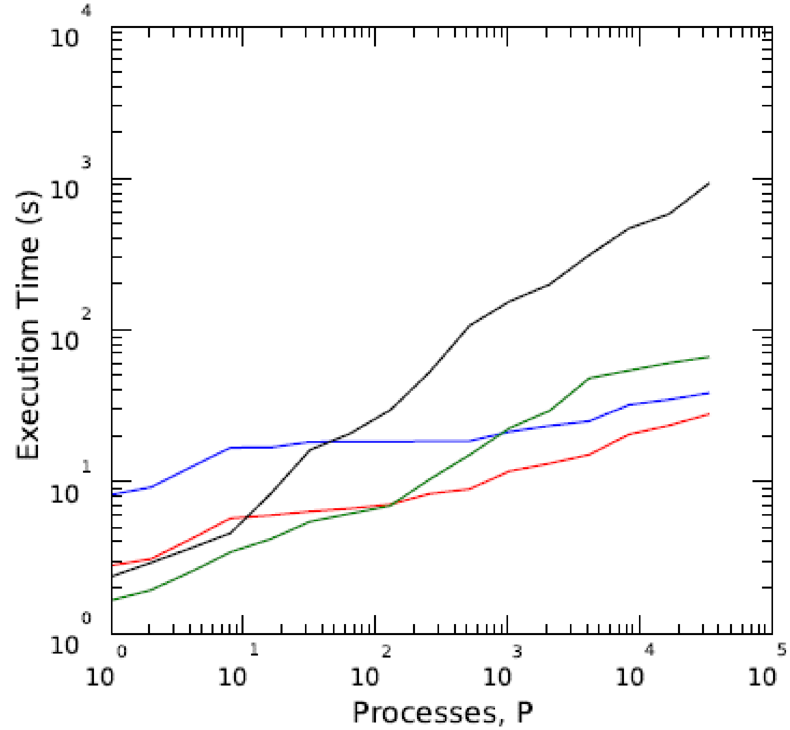




Weak Scaling Results

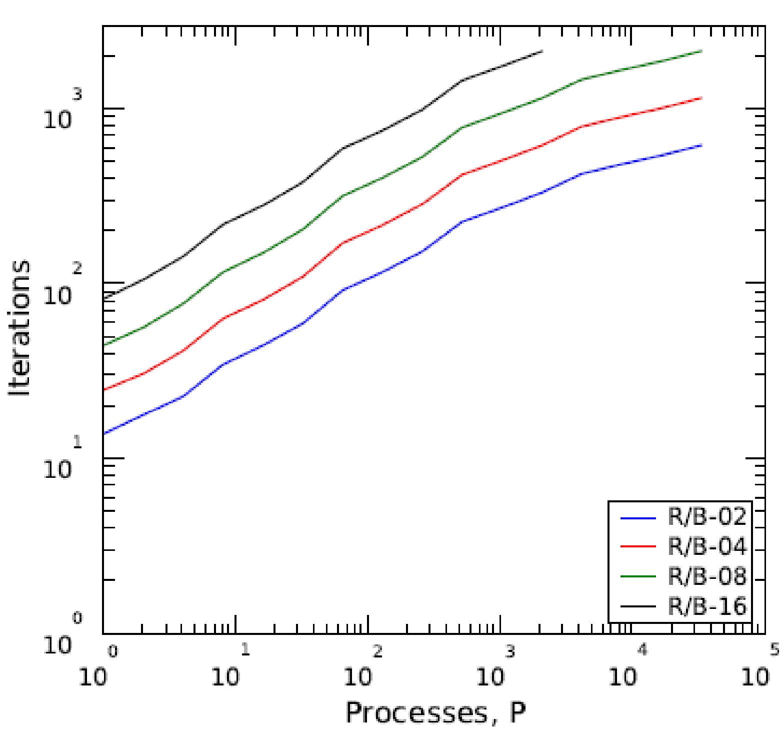
- c=0.9, h=10.0

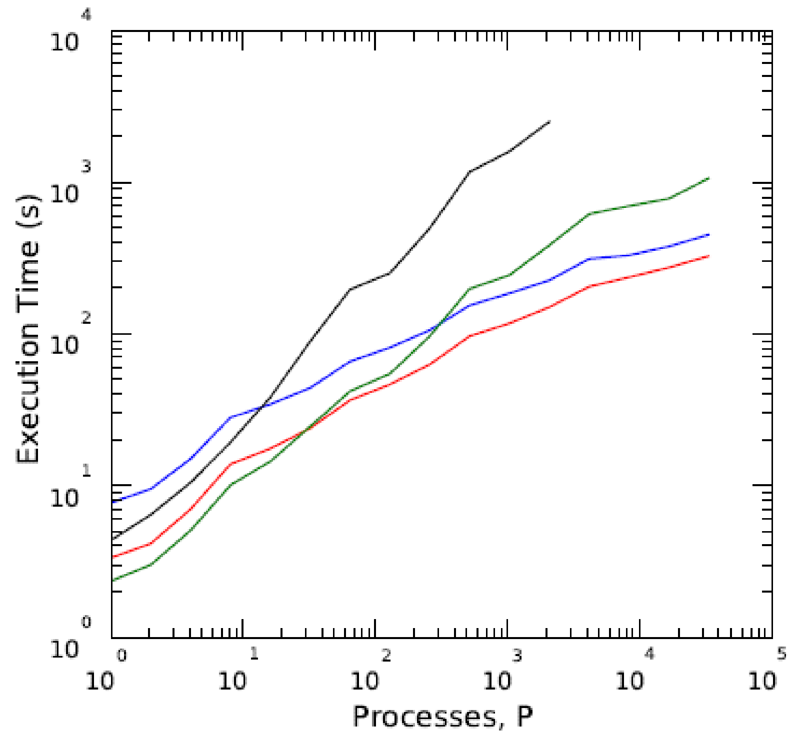




Weak Scaling Results

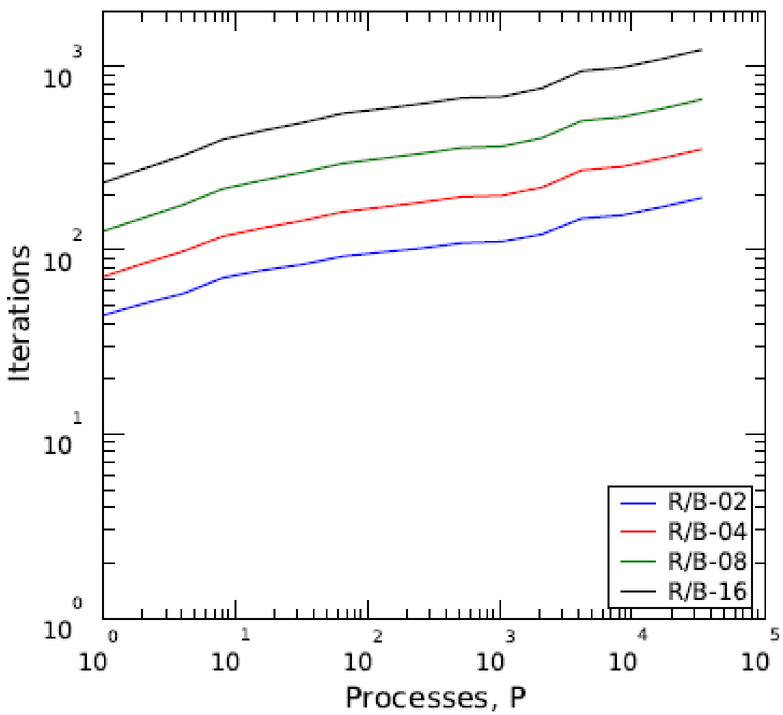
- c=0.99, h=0.1

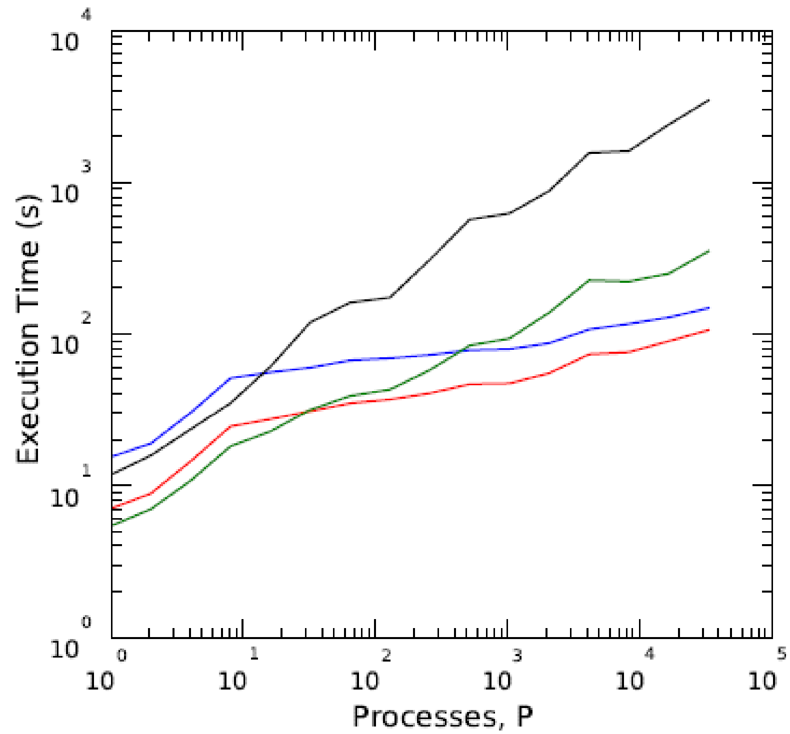




Weak Scaling Results

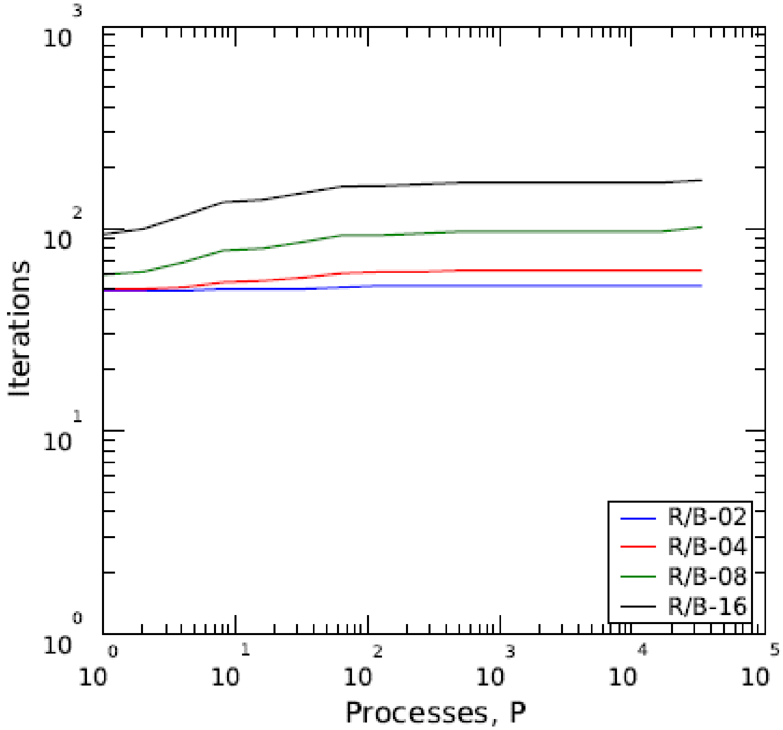
- c=0.99, h=1.0

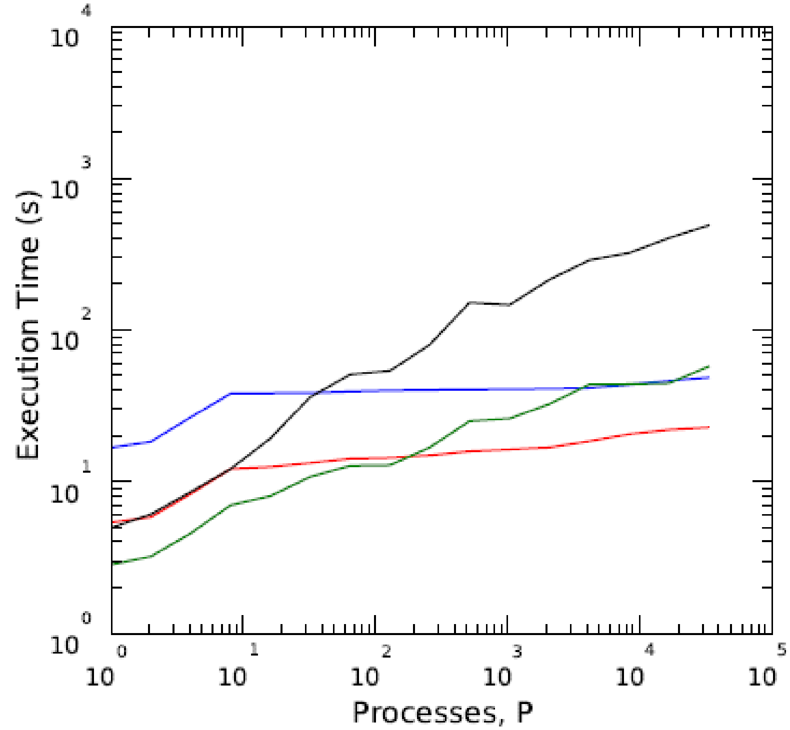




Weak Scaling Results

- c=0.99, h=10.0



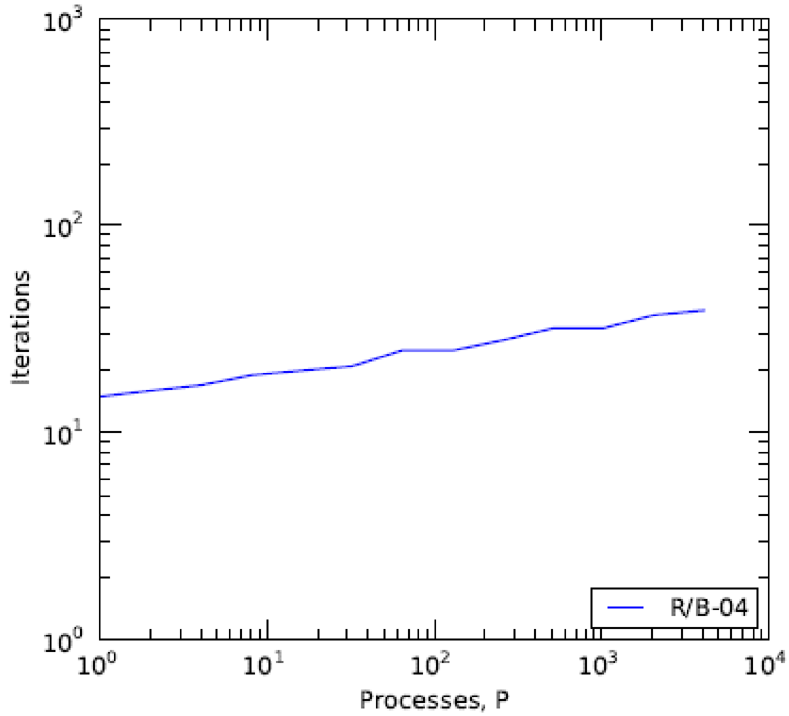


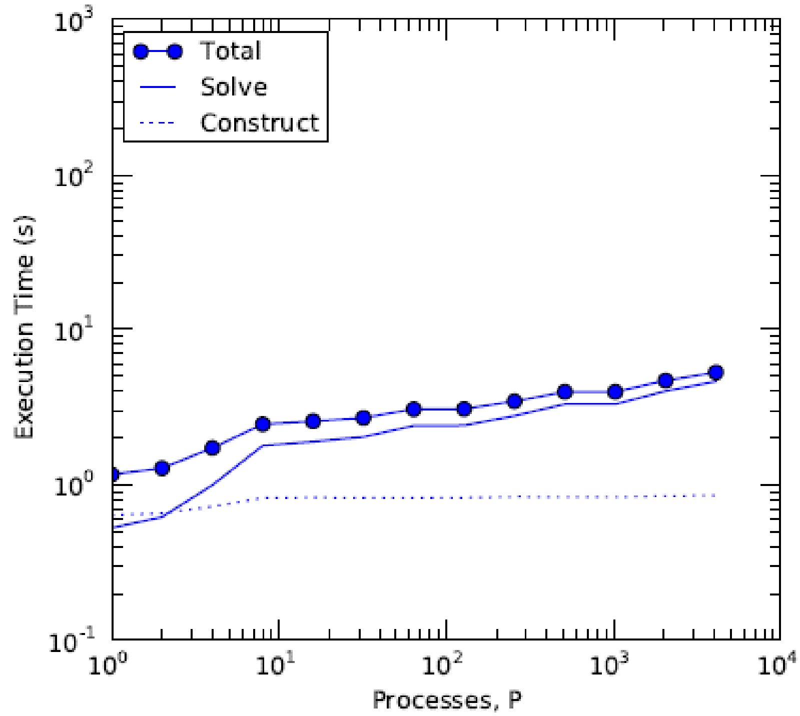
Periodic Heterogeneous Layers

- Alternating layers of optically thin and thick materials
- Known challenge for SI-DSA convergence
- ITMM explicitly couples thick and thin materials
- Starting with h=1, increase every other layer by a factor of ‘a’ and decrease the other layers by a factor of ‘a’

PHL Weak Scaling Results

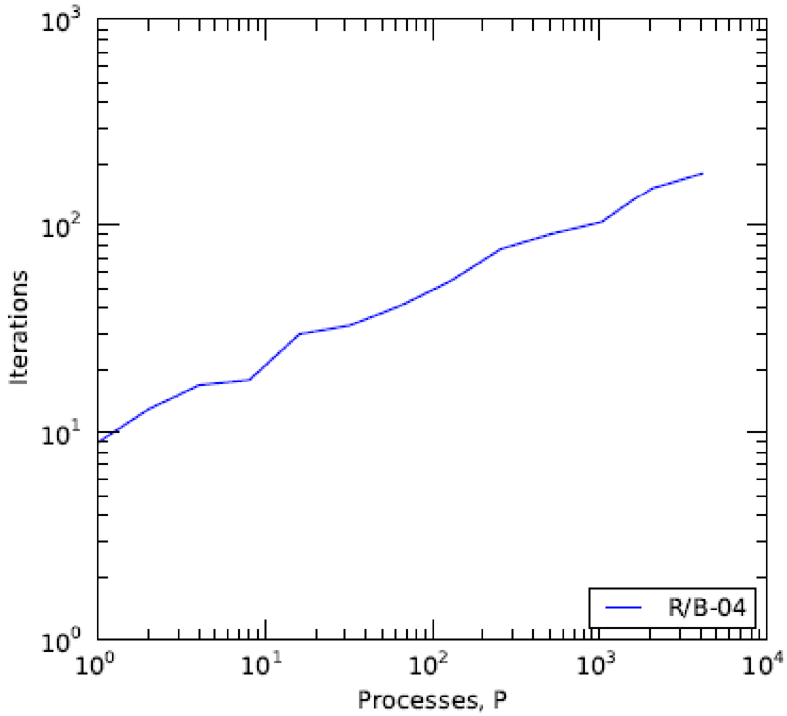
- c=0.9, a=10

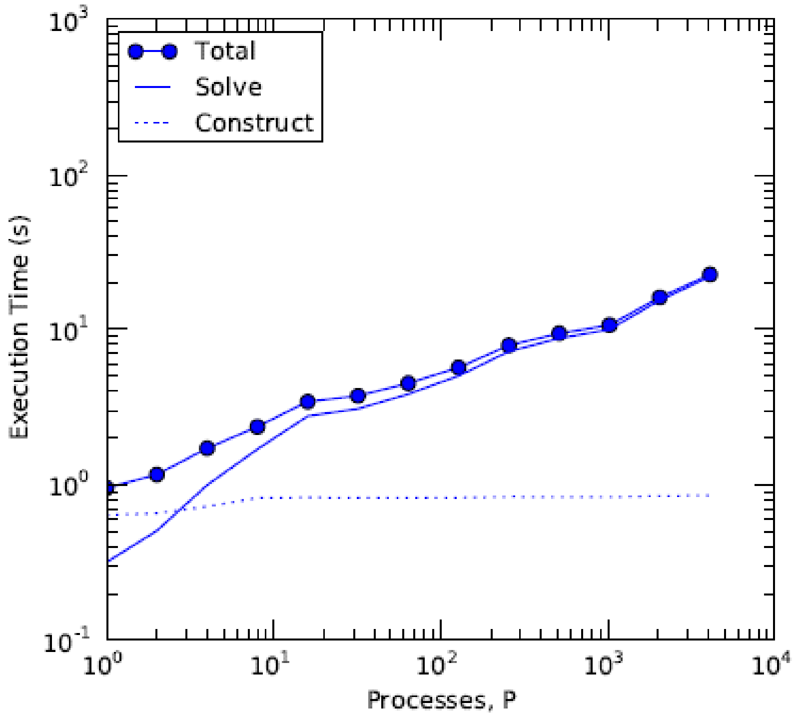




PHL Weak Scaling Results

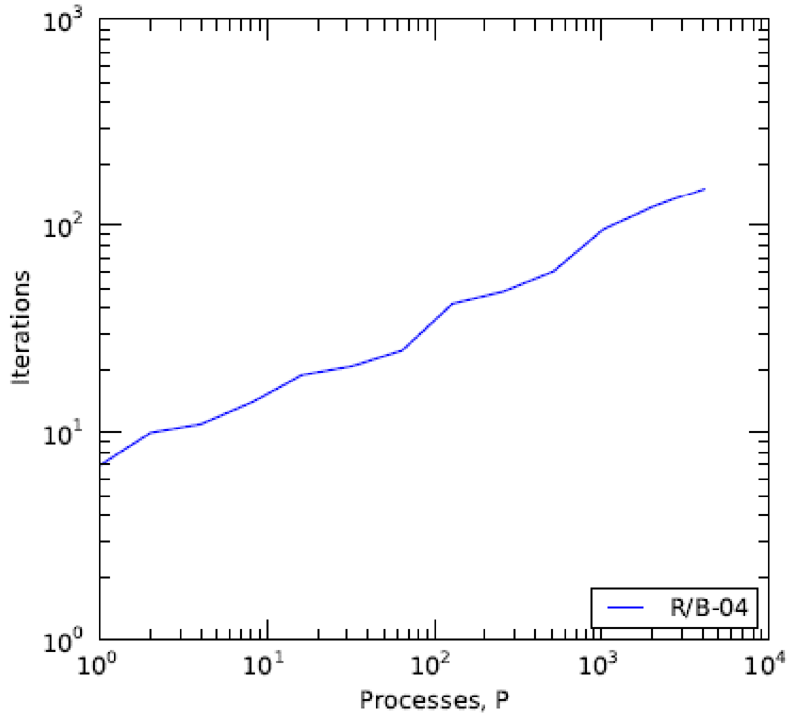
- c=0.9, a=100

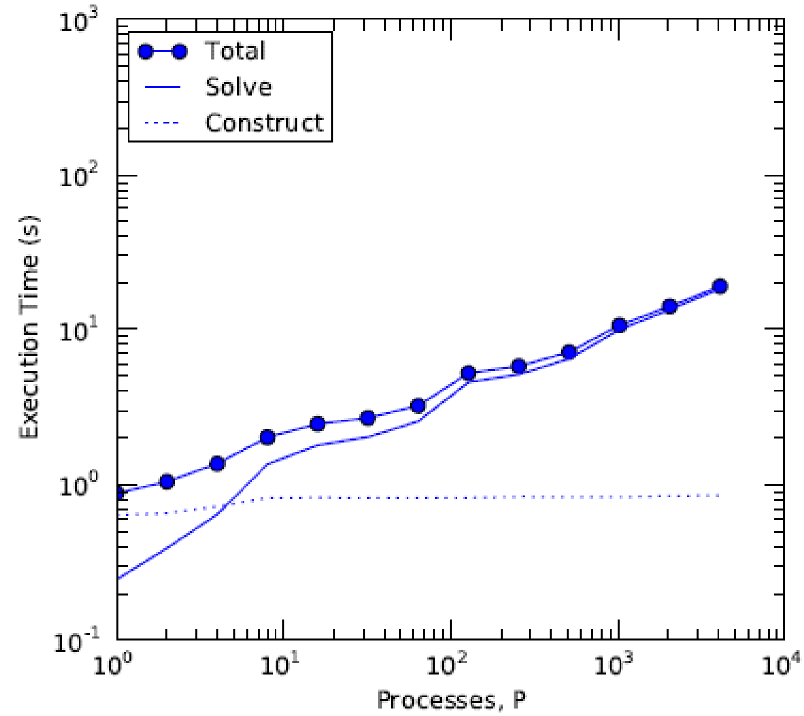




PHL Weak Scaling Results

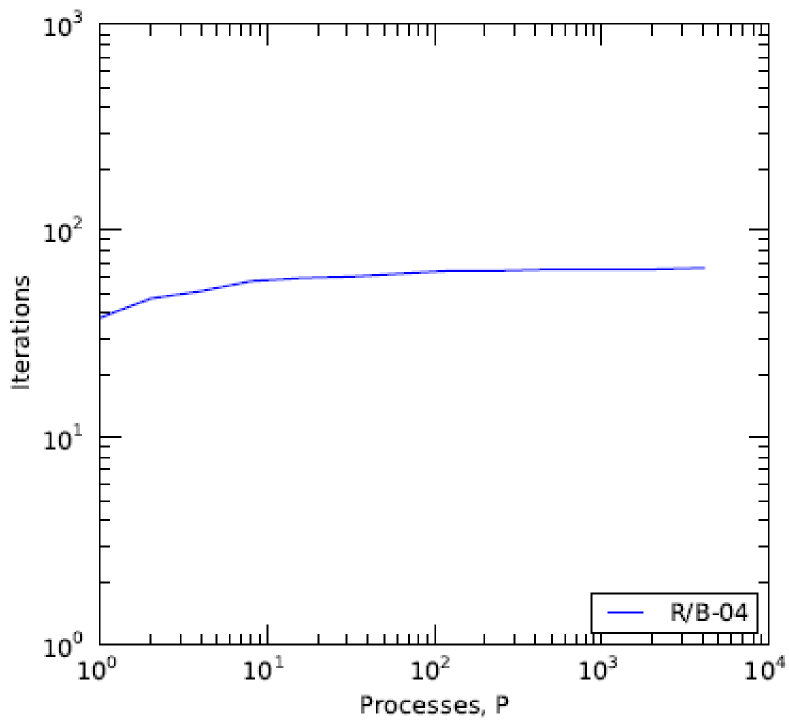
- c=0.9, a=1000

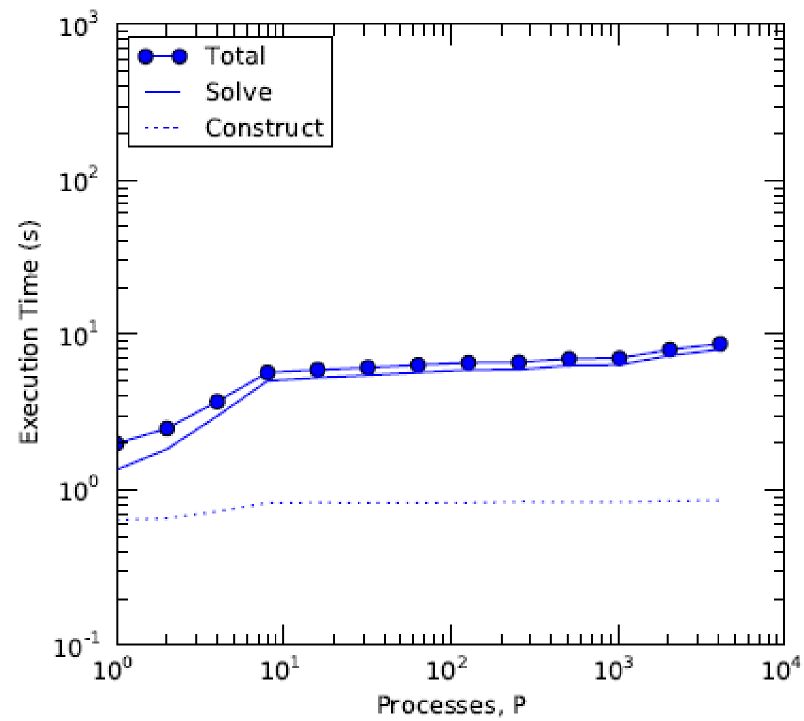




PHL Weak Scaling Results

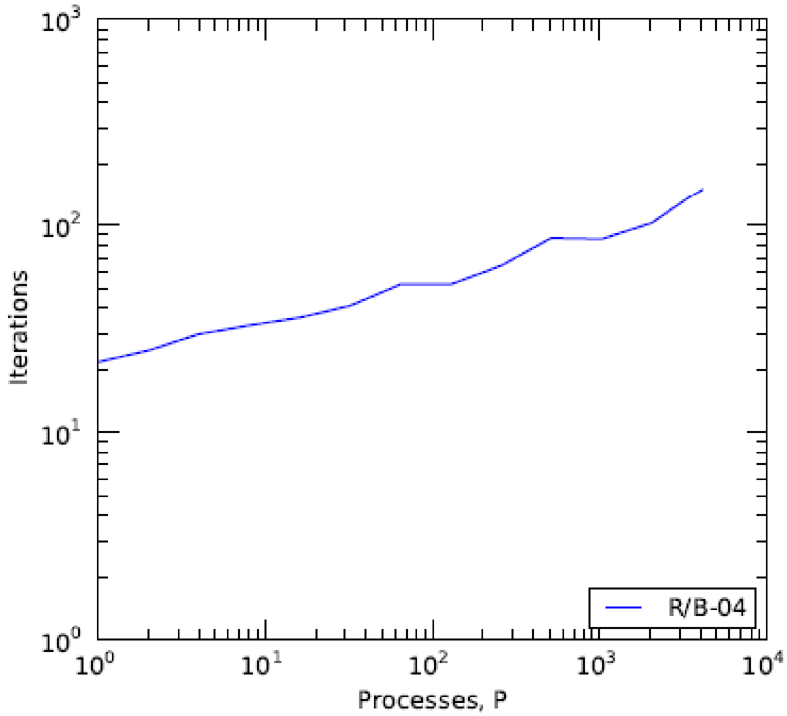
- c=0.99, a=10

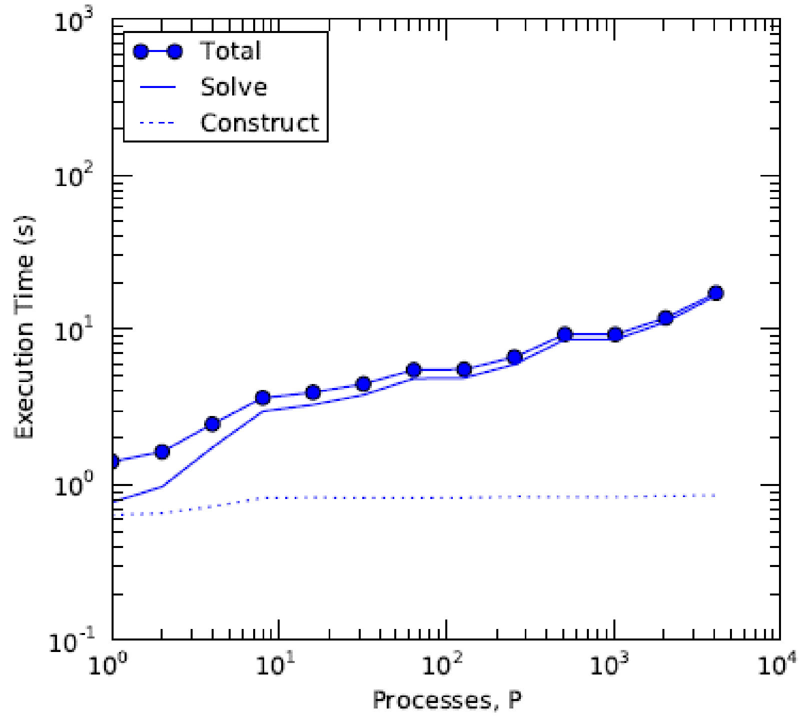




PHL Weak Scaling Results

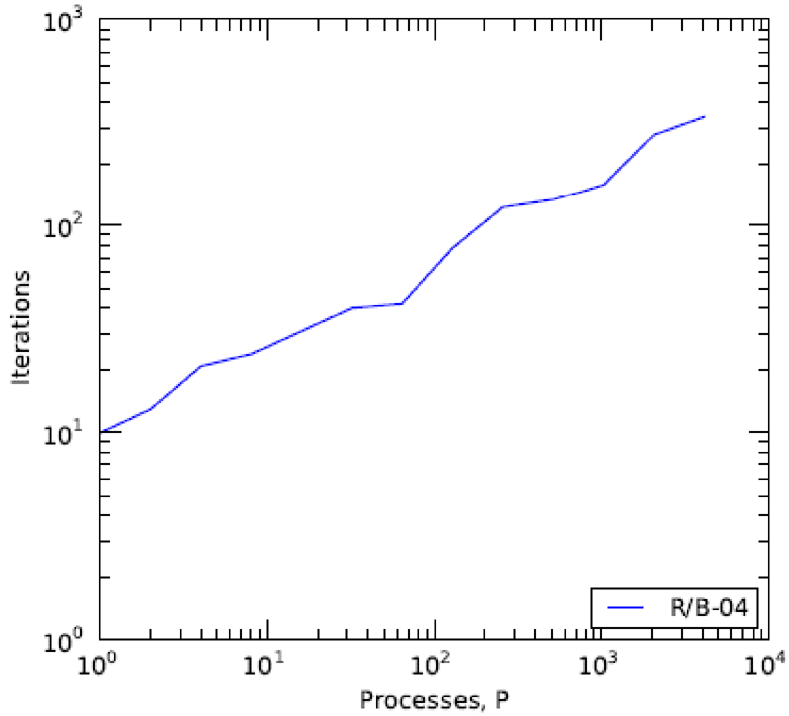
- c=0.99, a=100

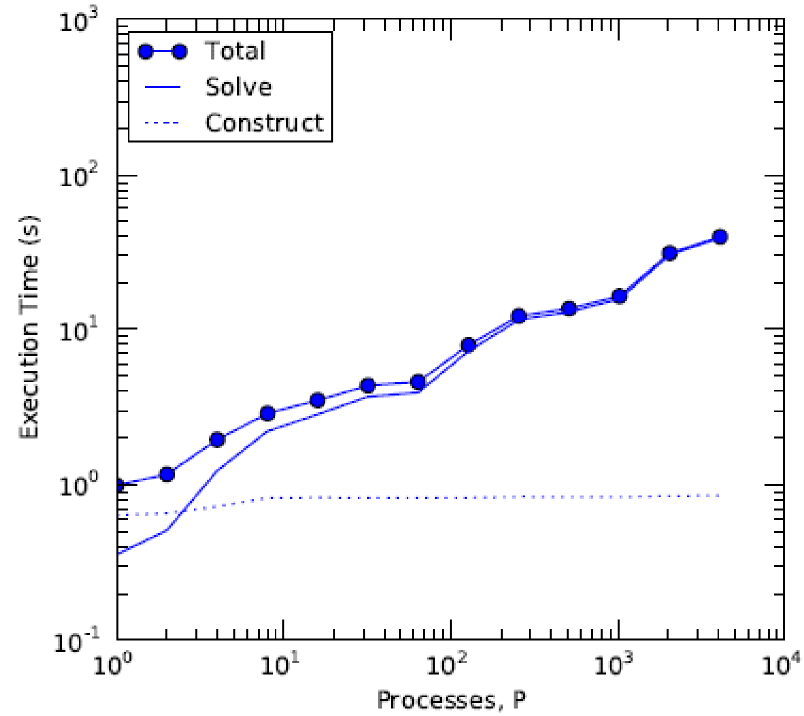




PHL Weak Scaling Results

- c=0.99, a=1000





7. Conclusions

7. Conclusions

- **The Nuclear Computational Science Group at NC State is engaged in broad span of topics**
- **Topic illustrated today: Multiprocessing strategies particularly suited for massively parallel architectures**
  - ITMM SDD avoids sequential mesh-sweeps
  - Considered BJ & GS parallelizations: PGS bests PBJ
  - Compared PGS to traditional KBS in PARTISN:
    - Very large differences when SI is accelerated with DSA
    - Gap closes as optical thickness and scattering ratio are increased ⇨ most difficult SI problems
    - SI & SI-DSA demonstrate larger growth in execution time as *P* ↑
    - Conclusions must be validated for *P* > 1,024
- **PGS performance should improve with suitable preconditioner: difficult**