

Massively Parallel Monte Carlo Methods for Discrete Linear and Nonlinear Systems

Stuart R. Slattery
Engineering Physics Department
University of Wisconsin - Madison

October 28, 2012





- Predictive modeling and simulation enhances engineering capability
- Modern work focused on this task leverages multiple physics simulation (CASL, NEAMS)
- New hardware drives algorithm development (petascale and exascale)
- Monte Carlo methods have the potential to provide great improvements that permit finer simulations and better mapping to future hardware
- A set of massively parallel Monte Carlo methods is proposed to advance multiple physics simulation on contemporary and future leadership class machines



Predictive nuclear reactor analysis enables...

- Tighter design tolerance for improved thermal performance and efficiency
- Higher fuel burn-up
- High confidence in accident scenario models

Predictive nuclear reactor analysis enables...

- Tighter design tolerance for improved thermal performance and efficiency
- Higher fuel burn-up
- High confidence in accident scenario models

Multiple physics simulations are complicated...

- Neutronics, thermal hydraulics, computational fluid dynamics, structural mechanics, and many other physics
- Consistent models yield nonlinearities in the variables through feedback effects
- Tremendous computational resources are required with $O(1 \times 10^9)$ element meshes and $O(100,000)+$ cores used in today's simulations.

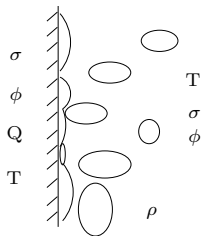
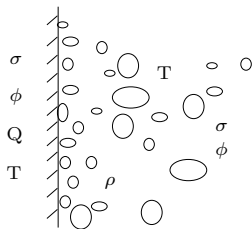


Figure: Departure from nucleate boiling scenario.

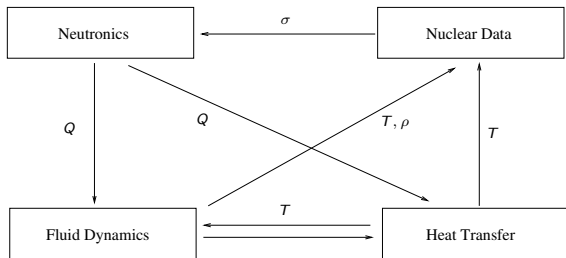
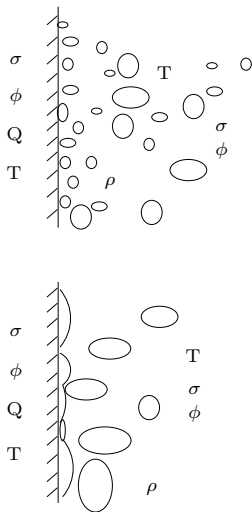


Figure: Multiphysics dependency analysis of departure from nucleate boiling.

Figure: Departure from nucleate boiling scenario.



- Modern hardware is moving in two directions:
 - Lightweight machines
 - Heterogeneous machines
 - Both characterized by low power and high concurrency
- Some issues:
 - Higher potential for both soft and hard failures
 - Memory restrictions are expected with a continued decrease in memory/FLOPS
- Potential resolution from Monte Carlo:
 - Soft failures buried within the tally variance
 - Hard failures are high variance events
 - Memory savings over conventional methods

- Parallelization of Monte Carlo methods for discrete systems
 - Parallel strategies taken from modern reactor physics methods
 - Research is required to explore varying parallel strategies
 - Scalability is of concern
- Development of a nonlinear solver leveraging Monte Carlo
 - Application to nonlinear problems of interest
 - Memory benefits
 - Performance benefits

- We seek solutions of the general linear operator equation

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{A} \in \mathbb{R}^{N \times N}, \mathbf{A} : \mathbb{R}^N \rightarrow \mathbb{R}^N, \mathbf{x} \in \mathbb{R}^N, \mathbf{b} \in \mathbb{R}^N$$

$$\mathbf{r} = \mathbf{b} - \mathbf{Ax}$$

- $\mathbf{r} = \mathbf{0}$ when an exact solution is found.

- We seek solutions of the general linear operator equation

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{A} \in \mathbb{R}^{N \times N}, \mathbf{A} : \mathbb{R}^N \rightarrow \mathbb{R}^N, \mathbf{x} \in \mathbb{R}^N, \mathbf{b} \in \mathbb{R}^N$$

$$\mathbf{r} = \mathbf{b} - \mathbf{Ax}$$

- $\mathbf{r} = \mathbf{0}$ when an exact solution is found.

A Requirement

Assert that \mathbf{A} is *nonsingular*. The solution is then:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

- General stationary methods are formed by splitting the linear operator

$$\mathbf{A} = \mathbf{M} - \mathbf{N} .$$

$$\mathbf{x} = \mathbf{M}^{-1}\mathbf{N}\mathbf{x} + \mathbf{M}^{-1}\mathbf{b} .$$

- We identify $\mathbf{H} = \mathbf{M}^{-1}\mathbf{N}$ as the *iteration matrix*

$$\mathbf{x}^{k+1} = \mathbf{H}\mathbf{x}^k + \mathbf{c} .$$

- The qualities of the iteration matrix dictate convergence
- Define $\mathbf{e}^k = \mathbf{x}^k - \mathbf{x}$ as the error at the k^{th} iterate

$$\mathbf{e}^{k+1} = \mathbf{H}\mathbf{e}^k$$

- We diagonalize \mathbf{H} to extract its Eigenvalues

$$\|\mathbf{e}^k\|_2 = \rho(\mathbf{H})^k \|\mathbf{e}^0\|_2,$$

- We bound \mathbf{H} by $\rho(\mathbf{H}) < 1$ for convergence



- Powerful class of iterative methods
- Provides theory that encapsulates most other iterative methods
- Leveraged in many modern physics codes at the petascale

- Powerful class of iterative methods
- Provides theory that encapsulates most other iterative methods
- Leveraged in many modern physics codes at the petascale

Search Subspace \mathcal{K}

Extract the solution from the search subspace:

$$\tilde{\mathbf{x}} = \mathbf{x}_0 + \boldsymbol{\delta}, \boldsymbol{\delta} \in \mathcal{K}$$

- Powerful class of iterative methods
- Provides theory that encapsulates most other iterative methods
- Leveraged in many modern physics codes at the petascale

Search Subspace \mathcal{K}

Extract the solution from the search subspace:

$$\tilde{\mathbf{x}} = \mathbf{x}_0 + \boldsymbol{\delta}, \quad \boldsymbol{\delta} \in \mathcal{K}$$

Constraint Subspace \mathcal{L}

Constrain the extraction with the constraint subspace by asserting orthogonality with the residual:

$$\langle \tilde{\mathbf{r}}, \mathbf{w} \rangle = 0, \quad \forall \mathbf{w} \in \mathcal{L}$$

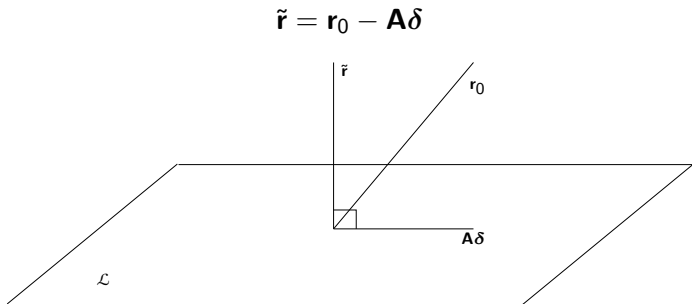


Figure: Orthogonality constraint of the new residual with respect to \mathcal{L} .

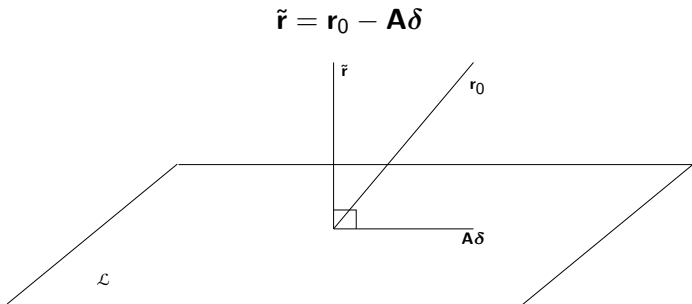


Figure: Orthogonality constraint of the new residual with respect to \mathcal{L} .

Minimization Property

The residual of the system will always be *minimized* with respect to the constraints

$$\|\tilde{\mathbf{r}}\|_2 \leq \|\mathbf{r}_0\|_2, \quad \forall \mathbf{r}_0 \in \mathbb{R}^N,$$

- Choose \mathbf{V} as a basis of \mathcal{K} and \mathbf{W} as a basis of \mathcal{L}

$$\delta = \mathbf{V}\mathbf{y}, \quad \forall \mathbf{y} \in \mathbb{R}^N$$

$$\mathbf{y} = (\mathbf{W}^T \mathbf{A} \mathbf{V})^{-1} \mathbf{W}^T \mathbf{r}_0$$

- Choose \mathbf{V} as a basis of \mathcal{K} and \mathbf{W} as a basis of \mathcal{L}

$$\delta = \mathbf{V}\mathbf{y}, \quad \forall \mathbf{y} \in \mathbb{R}^N$$

$$\mathbf{y} = (\mathbf{W}^T \mathbf{A} \mathbf{V})^{-1} \mathbf{W}^T \mathbf{r}_0$$

Projection Method Iteration

$$\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k$$

$$\mathbf{y}^k = (\mathbf{W}^T \mathbf{A} \mathbf{V})^{-1} \mathbf{W}^T \mathbf{r}^k$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{V}\mathbf{y}^k$$

Update \mathbf{V} and \mathbf{W}

$$\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \dots, \mathbf{A}^{m-1}\mathbf{r}_0\}$$

$$\mathcal{L} = \mathbf{A}\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$$

- Yields the normal system $\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}$
- Must generate an orthonormal basis $\mathbf{V}_m \in \mathbb{R}^{N \times m}$ for $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$
- $\mathbf{W}_m = \mathbf{A} \mathbf{V}_m$
- Typically choose a Gram-Schmidt-like procedure such as Arnoldi or Lanczos

Algorithm 1 GMRES Iteration

```
1:  $\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$ 
2:  $\beta := \|\mathbf{r}_0\|_2$ 
3:  $\mathbf{v}_1 := \mathbf{r}_0/\beta$  {Create the orthonormal basis for the Krylov subspace}
4: for  $j = 1, 2, \dots, m$  do
5:    $h_{ij} \leftarrow \langle \mathbf{w}_j, \mathbf{v}_i \rangle$ 
6:    $\mathbf{w}_j \leftarrow \mathbf{w}_j - h_{ij}\mathbf{v}_i$ 
7: end for
8:  $h_{j+1,j} \leftarrow \|\mathbf{w}_j\|_2$ 
9:  $\mathbf{v}_{j+1} \leftarrow \mathbf{w}_j/h_{j+1,j}$  {Apply the orthogonality constraints}
10:  $\mathbf{y}_m \leftarrow \operatorname{argmin}_y \|\beta \mathbf{e}_1 - \mathbf{H}_m \mathbf{y}\|_2$ 
11:  $\mathbf{x}_m \leftarrow \mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m$ 
```

- Parallel vector update

$$\mathbf{y}[n] \leftarrow \mathbf{y}[n] + a * \mathbf{x}[n], \quad \forall n \in [1, N_g]$$

$$\mathbf{y}[n] \leftarrow \mathbf{y}[n] + a * \mathbf{x}[n], \quad \forall n \in [1, N_l]$$

- Parallel dot product

$$d_l = \mathbf{y}_l \cdot \mathbf{x}_l, \quad d_g = \sum_p d_l$$

- Parallel vector norm

$$\|\mathbf{x}\|_{\infty, l} = \max_n \mathbf{y}[n], \quad \forall n \in [1, N_l]$$

$$\|\mathbf{x}\|_{\infty, g} = \max_p \|\mathbf{x}\|_{\infty, l}$$

Parallel Matrix-Vector Multiplication

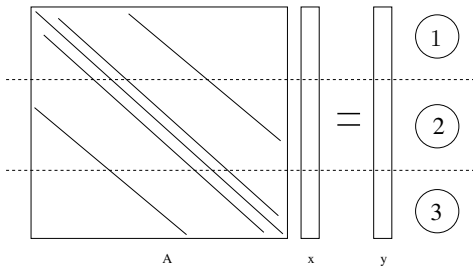


Figure: Matrix-vector multiply $Ax = y$ operation on 3 processors.

Parallel Matrix-Vector Multiplication

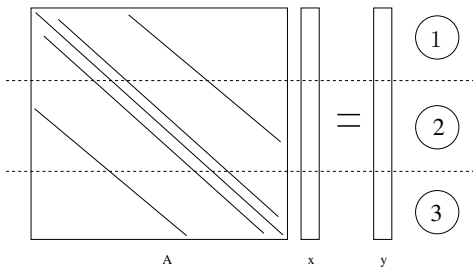


Figure: Matrix-vector multiply $Ax = y$ operation on 3 processors.

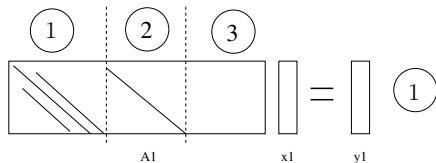


Figure: Components of multiply operation owned by process 1.













Monte Carlo Solution Methods for Nonlinear Problems

















Proposed Challenge Problem



