Massively Parallel Monte Carlo Methods for Discrete Linear and Nonlinear Systems

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

October 28, 2012

Introduction

- 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

Physics-Based Motivation

Predictive nuclear reactor analysis enables...

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

Physics-Based Motivation

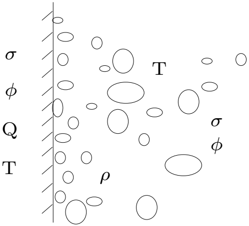
Predictive nuclear reactor analysis enables...

- 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.

Physics-Based Motivation: DNB



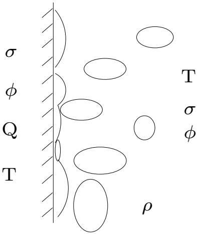


Figure: **Departure from nucleate boiling scenario.**

Physics-Based Motivation: DNB

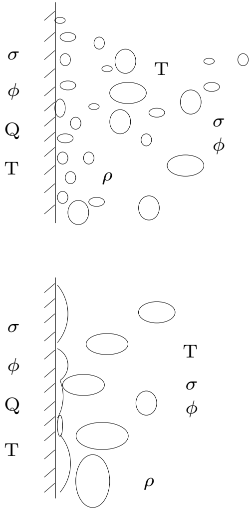


Figure: **Departure from nucleate boiling scenario.**

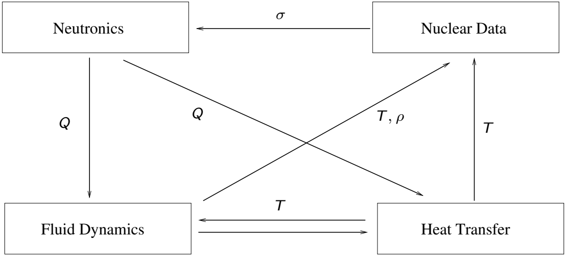


Figure: **Multiphysics dependency analysis of departure from nucleate boiling.**

Hardware-Based Motivation

- 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

Research Outline

- 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

Linear Operator Equations

- We seek solutions of the general linear operator equation

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

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

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

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

Linear Operator Equations

- We seek solutions of the general linear operator equation

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

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

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

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

A Requirement

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

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

Stationary Methods

- 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*

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

Stationary Methods Convergence

- 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

Projection Methods

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

Projection Methods

- 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:

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

Projection Methods

- 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:

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

Constraint Subspace $\mathcal{L}$

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

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

The Orthogonality Constraint

$$\tilde{\mathbf{r}} = \mathbf{r}_0 - \mathbf{A}\delta$$

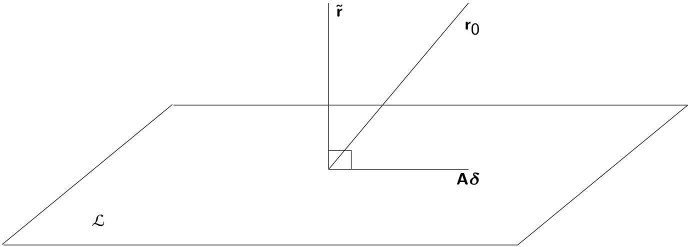


Figure: **Orthogonality constraint of the new residual with respect to $\mathcal{L}$.**

The Orthogonality Constraint

$$\tilde{\mathbf{r}} = \mathbf{r}_0 - \mathbf{A}\delta$$

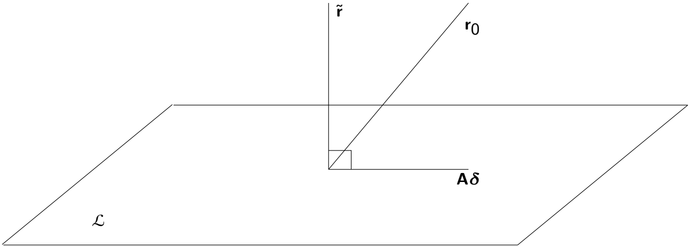


Figure: **Orthogonality constraint of the new residual with respect to $\mathcal{L}$.**

Minimization Property

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

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

Putting it All Together

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

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

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

Putting it All Together

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

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

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

Projection Method Iteration

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

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

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

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

Krylov Subspace Methods

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

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

- 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 Lanzcos

GMRES

Algorithm 1 GMRES Iteration

$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\beta := ||\mathbf{r}_0||_2$
$\mathbf{v}_1 := \mathbf{r}_0/\beta$ {Create the orthonormal basis for the Krylov subspace}
for $j = 1, 2, \cdots, m$ **do**
    $h_{ij} \leftarrow \langle w_j, v_j \rangle$
    $w_j \leftarrow w_j - h_{ij} v_i$
end for
$h_{j+1,j} \leftarrow ||w_j||_2$
$v_{j+1} \leftarrow w_j / h_{j+1,j}$ {Apply the orthogonality constraints}
$\mathbf{y}_m \leftarrow argmin_y ||\beta \mathbf{e}_1 - \mathbf{H}_m\ \mathbf{y}||_2$
$\mathbf{x}_m \leftarrow \mathbf{x}_0 + \mathbf{V}_\mathbf{m} \mathbf{y}_m$

Parallel Projection Methods

- Parallel vector update

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

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

- Parallel dot product

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

- Parallel vector norm

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

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

Parallel Matrix-Vector Multiplication

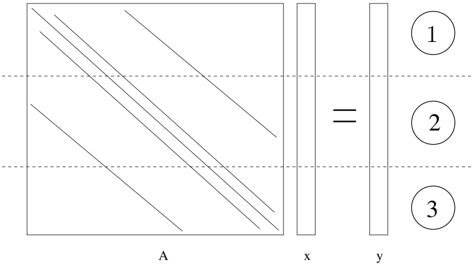


Figure: **Matrix-vector multiply $\mathbf{Ax} = \mathbf{y}$ operation on 3 processors.**

Parallel Matrix-Vector Multiplication

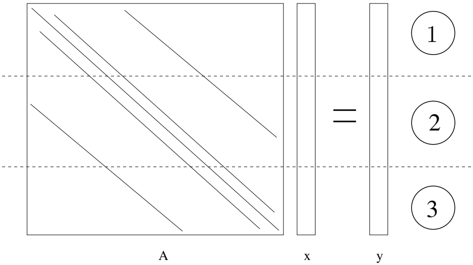


Figure: **Matrix-vector multiply $\mathbf{Ax} = \mathbf{y}$ operation on 3 processors.**

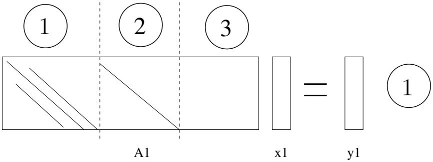


Figure: **Components of multiply operation owned by process 1.**

Projection Method Notes

- Global reduction operations observed not to impede scalability
  - Dot product
  - Vector norms
- Nearest neighbor computations have poor algorithmic strong scaling
  - Matrix-vector multiply
  - Weak scaling is better

Monte Carlo Solution Methods for Discrete Linear Systems

- First proposed by J. Von Neumann and S.M. Ulam in the 1940's
- Earliest published reference in 1950
- General lack of published work
- Modern work by Evans and others has yielded new applications
- No indication of parallel methods in the literature

Monte Carlo Linear Solver Preliminaries

- Split the operator

$$\mathbf{H} = \mathbf{I} - \mathbf{A}$$

$$\mathbf{x} = \mathbf{H}\mathbf{x} + \mathbf{b}$$

- Generate the *Neumann series*

$$\mathbf{A}^{-1} = (\mathbf{I} - \mathbf{H})^{-1} = \sum_{k=0}^{\infty} \mathbf{H}^k$$

- Require $\rho(\mathbf{H}) < 1$ for convergence

$$\mathbf{A}^{-1}\mathbf{b} = \sum_{k=0}^{\infty} \mathbf{H}^k \mathbf{b} = \mathbf{x}$$

Monte Carlo Linear Solver Preliminaries

- Expand the Nuemann series

$$x_i = \sum_{k=0}^{\infty} \sum_{i_1}^{N} \sum_{i_2}^{N} \dots \sum_{i_k}^{N} h_{i,i_1} h_{i_1,i_2} \dots h_{i_{k-1},i_k} b_{i_k}$$

- Define a sequence of state transitions

$$\nu = i \rightarrow i_1 \rightarrow \cdots \rightarrow i_{k-1} \rightarrow i_k$$

- Define the *Neumann-Ulam decomposition*[1]

$$\mathbf{H} = \mathbf{P} \circ \mathbf{W}$$

[1] The Hadamard product $\mathbf{A} = \mathbf{B} \circ \mathbf{C}$ is defined element-wise as $a_{ij} = b_{ij} c_{ij}$.

Direct Method

Adjoint Method

Monte Carlo Nonlinear Solver Preliminaries

Inexact Newton Methods

The FANM Method

Research Proposal

Experimental Framework

Progress to Date

Monte Carlo Methods Verification

Proposed Numerical Experiments

Proposed Challenge Problem

Conclusion