SAND2012-2253P

Enabling Extreme-Scale Computation for Emerging Discretizations

Michael Parks
Sandia National Laboratories

SAND2012-????P

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Collaborators

- **Burak Aksoylu** (One-year sabbatical at Sandia Labs)
  - Mathematical analysis, domain decomposition, peridynamics
- **Laurie Frink** (Colder Insights, Inc.)
  - Computational modeling of inhomogeneous fluids
- **Deaglan Halligan** (Ph.D. Student, Purdue, supervised by Ahmed Sameh)
  - Multiprecision computation
- **Mike Heroux** (Sandia Labs)
  - Scalable computation, numerical linear algebra, preconditioners
- **Dave Littlewood** (Sandia Labs)
  - Peridynamics, computational solid mechanics, material failure modeling
- **Mike Parks** (Sandia Labs)
  - PI
  - peridynamics, domain decomposition, numerical linear algebra, solvers, preconditioners

Part I
Enabling Science

Part II
Summary of Previous Work

Part III
Proposed Work

Sandia National Laboratories

Background

- Descriptive Modeling: Partial Differential Equations vs. Integral Equations
  - PDEs not suitable for everything (fundamentally local)
  - **Integral equations (IEs) more descriptive in many cases (fundamentally nonlocal)**

- IEs have different linear system properties

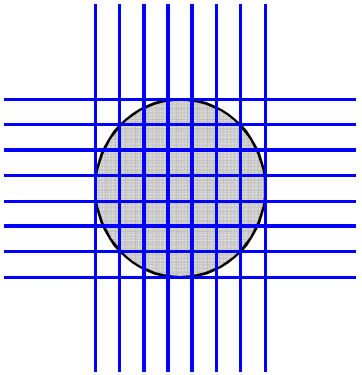
- PDEs - Matrix density independent of system size
- **IEs - Matrix density dependent upon system size**

- PDEs - Inter-nodal coupling dominates
- **IEs - Inter-physics coupling dominates**

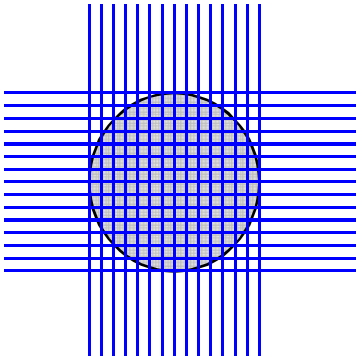
- PDE - Stencils based on nearest neighbors (mesh & geometry)
- **IEs - Stencils based on physical constants (physics)**

- PDEs - Usually a few DOFs per node
- **IEs - May have large numbers of DOF per node**

- **Two Important Applications of IEs**
  - **Modeling Fracture & Failure**
  - **Modeling Nanostructured Fluids**



Coarse Mesh
(Few nonzero per row)



Fine Mesh
(Many nonzero per row)

Fracture & Failure Modeling

- Equation for classical elastodynamics

$$\rho\ddot{\mathbf{u}}(\mathbf{x},t) = \nabla\cdot\sigma + \mathbf{b}(\mathbf{x},t) \qquad \sigma = \sigma(\nabla\mathbf{u})$$

  - Fracture solutions treated as a pathology
  - Special techniques (XFEM, cohesive zone) needed at discrete level to support desired solutions

Fracture & Failure Modeling

- Equation for classical elastodynamics

$$\rho\ddot{\mathbf{u}}(\mathbf{x},t) = \nabla\cdot\sigma + \mathbf{b}(\mathbf{x},t) \qquad \sigma = \sigma(\nabla\mathbf{u})$$

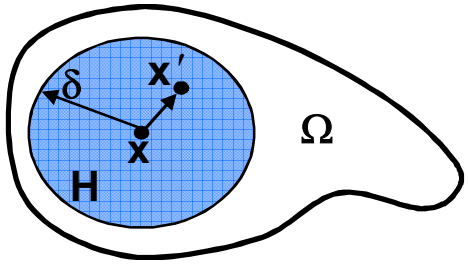
  - Fracture solutions treated as a pathology
  - Special techniques (XFEM, cohesive zone) needed at discrete level to support desired solutions

- **Peridynamics is a nonlocal extension of classical solid mechanics that permits discontinuous solutions**

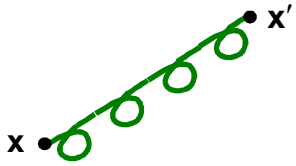
- **Peridynamic equation of motion (integral, nonlocal)**

$$\rho\ddot{\mathbf{u}}(\mathbf{x},t) = \int_{H} \mathbf{f}(\mathbf{u}'-\mathbf{u},\mathbf{x}'-\mathbf{x})dV' + \mathbf{b}(\mathbf{x},t)$$

  - Replace PDEs with integral equations
  - No obstacle to integrating nonsmooth functions (fracture)
  - Utilize same equation everywhere; cracks not "special"
  - When bonds stretch too much, they break
  - f(·, ·) is "force" function; contains constitutive model
  - f = 0 for particles x,x' more than δ apart (analogous to cutoff radius in molecular dynamics!)
  - Peridynamics is "continuum form of molecular dynamics"

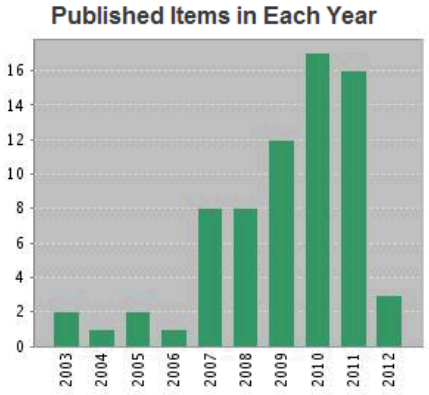


Peridynamic Domain



Peridynamic "bond"

Fracture & Failure Modeling



- Increasing adoption of peridynamics by academia, labs, industry
- Applications: Aerospace, mining, natural gas, etc.
- Several Ph.D. dissertations on peridynamics

- Use Sandia's ***Peridigm*** package for peridynamics
  - Sandia's primary open-source peridynamics code
  - Built upon Sandia's ***Trilinos Project***, algorithms and enabling technologies for the solution of large-scale, complex multi-physics engineering and scientific problems. **(trilinos.sandia.gov)**
  - Notable features: Massively parallel, Exodus input/output, multiple material blocks, explicit & implicit time integration linear elastic, elastic-plastic, viscoelastic models,
  - DAKOTA interface for UQ/optimization/calibration, etc. **(dakota.sandia.gov)**

Peridigm

Trilinos

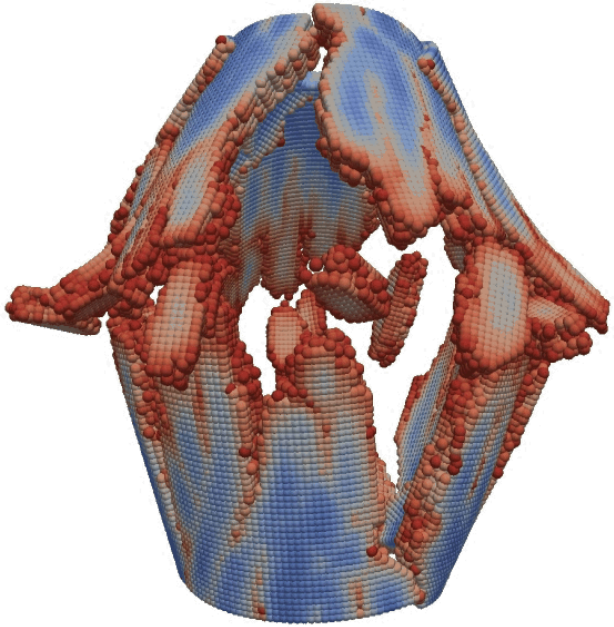
DAKOTA

Sandia National Laboratories

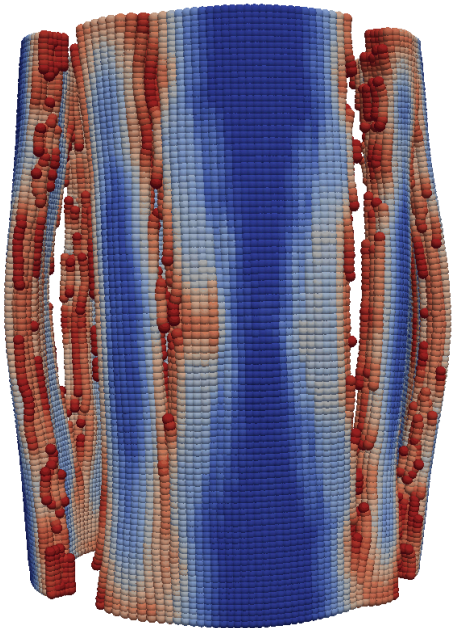
Fracture & Failure Modeling

- **Fragmenting Brittle Cylinder**
  - Motivated by tube fragmentation experiments of Winter (1979), Vogler (2003)*

Before



After
(brittle model)



After
(plastic model)

Color indicates damage

* D. Grady, Fragmentation of Rings And Shells: The Legacy of N.F. Mott, Springer, 2006.

Sandia National Laboratories

Fracture & Failure Modeling

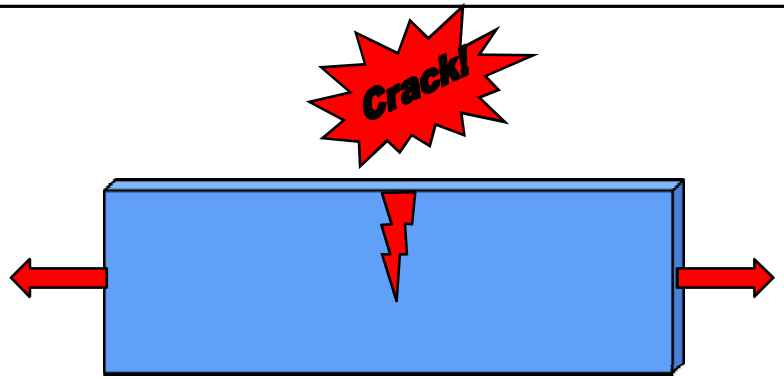
- **Example simulation: Dynamic brittle fracture in glass**
  - Joint with Florin Bobaru, Youn-Doh Ha (Nebraska), & Stewart Silling (SNL)

- **Soda-lime glass plate (microscope slide)**
  - Dimensions: 3” x 1” x 0.05”
  - Density: 2.44 g/cm3
  - Elastic Modulus: 79.0 Gpa

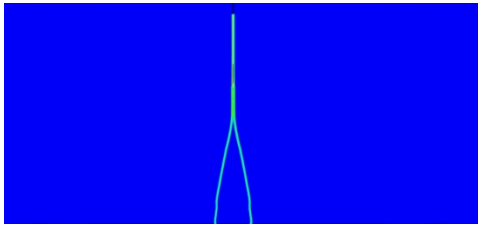
- **Discretization (finest)**
  - Mesh spacing: 35 microns
  - Approx. 82 million particles
  - Time: 50 microseconds (20k timesteps)

Setup

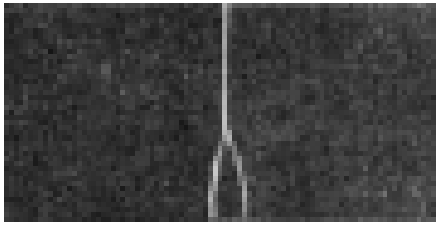
- Glass microscope slide
- Dimensions: 3” x 1” x 0.05”
- Notch at top, pull on ends



Results



Peridynamics



Physical Experiment*

Strain Energy Density

*S F. Bowden, J. Brunton, J. Field, and A. Heyes, *Controlled fracture of brittle solids and interruption of electrical current*, Nature, 216, 42, pp.38-42, 1967.

Sandia National Laboratories

Fracture & Failure Modeling

- Dawn (LLNL): IBM BG/P System
  - 500 teraflops; 147,456 cores
- Part of Sequoia procurement
  - 20 petaflops; 1.6 million cores
- Discretization (finest)
  - Mesh spacing: 35 microns
  - Approx. 82 million particles
  - Time: 50 microseconds (20k timesteps)
  - 6 hours on 65k cores
- Largest peridynamic simulations in history

***Dawn* at LLNL**

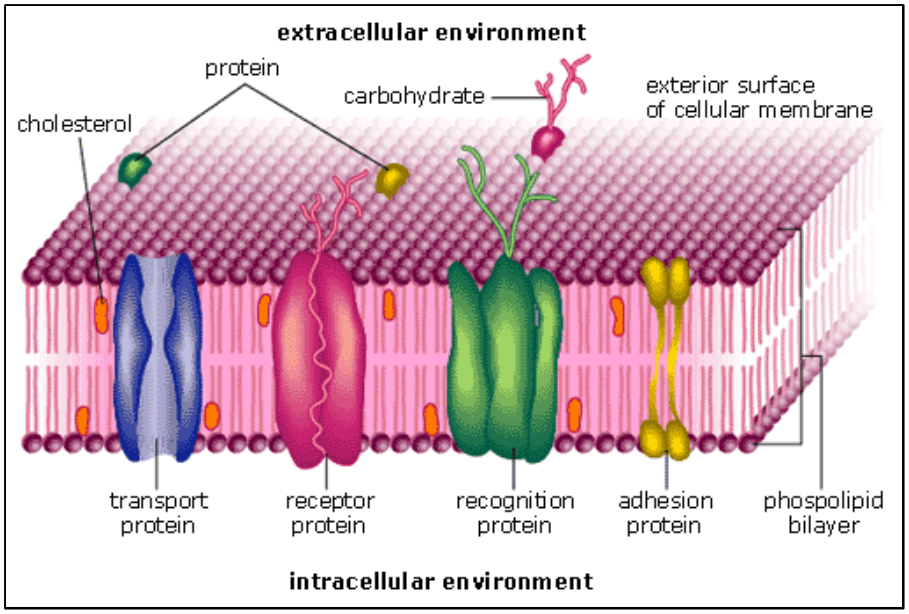
Weak Scaling Results

# Cores	# Particles	Particles/Core	Runtime (sec)	T(P)/T(P=512)
512	262,144	4096	14.417	1.000
4,096	2,097,152	4096	14.708	0.980
32,768	16,777,216	4096	15.275	0.963

Sandia National Laboratories

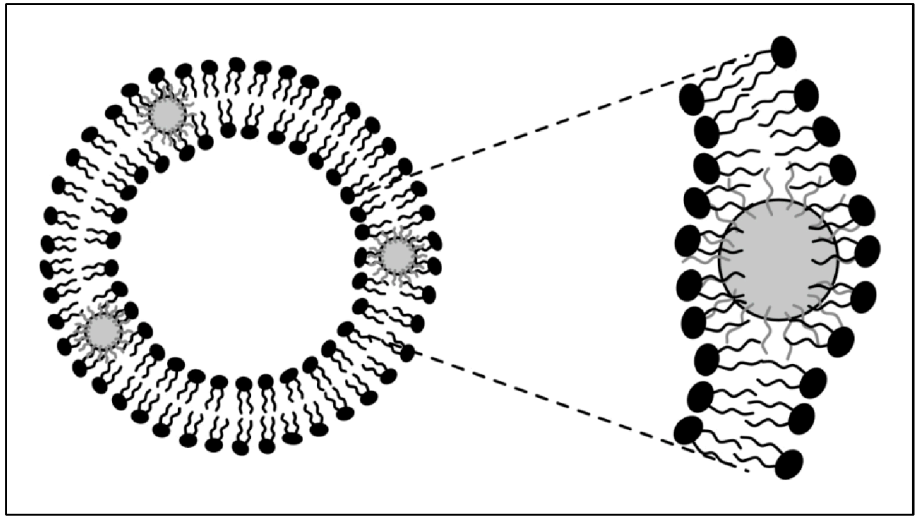
Nanostructured Fluids

- ❑ Structure arises from surfaces, fields, self-assembly
- ❑ Density, diffusion, and viscosity different from bulk fluid properties
- ❑ Rich phase behavior: wetting, capillary condensation, layering



Biological Membranes
Self-assembled fluid bilayer packed with proteins, peptides, etc.

Engineered Systems
Lipid vesicle/nanoparticle assemblies for drug delivery)



* G.D. Bothun, Hydrophobic silver nanoparticles trapped in lipid bilayers: Size distribution, bilayer phase behavior, and optical properties, *J. Nanobiotechnology*, 6, (2008).

Sandia National Laboratories

Density Functional Theory for Fluids

- Enable modeling and simulation of a wide range of applications, including fluids at interfaces, colloidal fluids, wetting, porous media, and biological mechanisms at the cellular level
- Given external field **V(r)**, determine structure of inhomogeneous fluid as captured by density distribution $\rho(\mathbf{r})$ via minimization of free energy functional $\Omega(\rho(\mathbf{r}))$

Coulomb interactions; Associations (H-bonding); [Applied field]

$$\Omega[\rho(r)] = F_{id} + F_{hs} + F_{vdW} + F_c + F_{assoc} + \int \rho(r)[V(r)-\mu]$$

Ideal gas; Hard sphere; Dispersion attractions

Legendre transform from Canonical to Grand Canonical ensemble

- Solve $\left(\frac{\delta\Omega}{\delta\rho(\mathbf{r})}\right)_{T,\mu} = \mathbf{0}$ with Newton-Krylov.

- Use Sandia's ***Tramonto*** package for complex fluid systems
  - Built upon ***Trilinos*** software components: *trilinos.sandia.gov*
  - Open source: *software.sandia.gov/tramonto/*

Sandia National Laboratories

Part I
Enabling Science

Part II
Summary of Previous Work

Part III
Proposed Work

Sandia National Laboratories

Summary of Previous Work

- Deployed and matured in Trilinos several new algorithmic capabilities consumed by the Tramonto Fluid DFT code
  - **Reduce node-level memory bandwidth and size usage**

- **Ability to solve fluid-DFT governing equations in 3D and at large scales crucial to continued scientific progress**

- **Schur-complement solvers are state-of-the-art in Fluid DFTs**

- **Mixed-precision and precision-neutral algorithms**
  - Leverage Trilinos/Tpetra (templated C++) solver stack
  - Performance and storage advantage of float over double
  - Utilize high-precision arithmetic if double inadequate

- **Least-squares methods (LSQR)**
  - Achieve robustness by dynamically adapting precision
  - Shield user from details of mixed-precision computation

- **Block Krylov recycling methods**
  - Recycling subspace information from previous solves to reduce iteration count
  - Block methods have superior convergence properties and computation to bandwidth requirements, improving processor utilization

Segregated Schur Complement Solvers*

- Resulting linear systems take the form

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

Each $\mathbf{A}_{ij}$ has own physics-based block structure

- Careful ordering of unknowns makes it advantageous to solve Schur complement

$$\mathbf{S}\mathbf{x}_2 = \mathbf{f}$$

where

$$\mathbf{S} = \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12} \qquad \mathbf{f} = \mathbf{b}_2 - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{b}_1$$

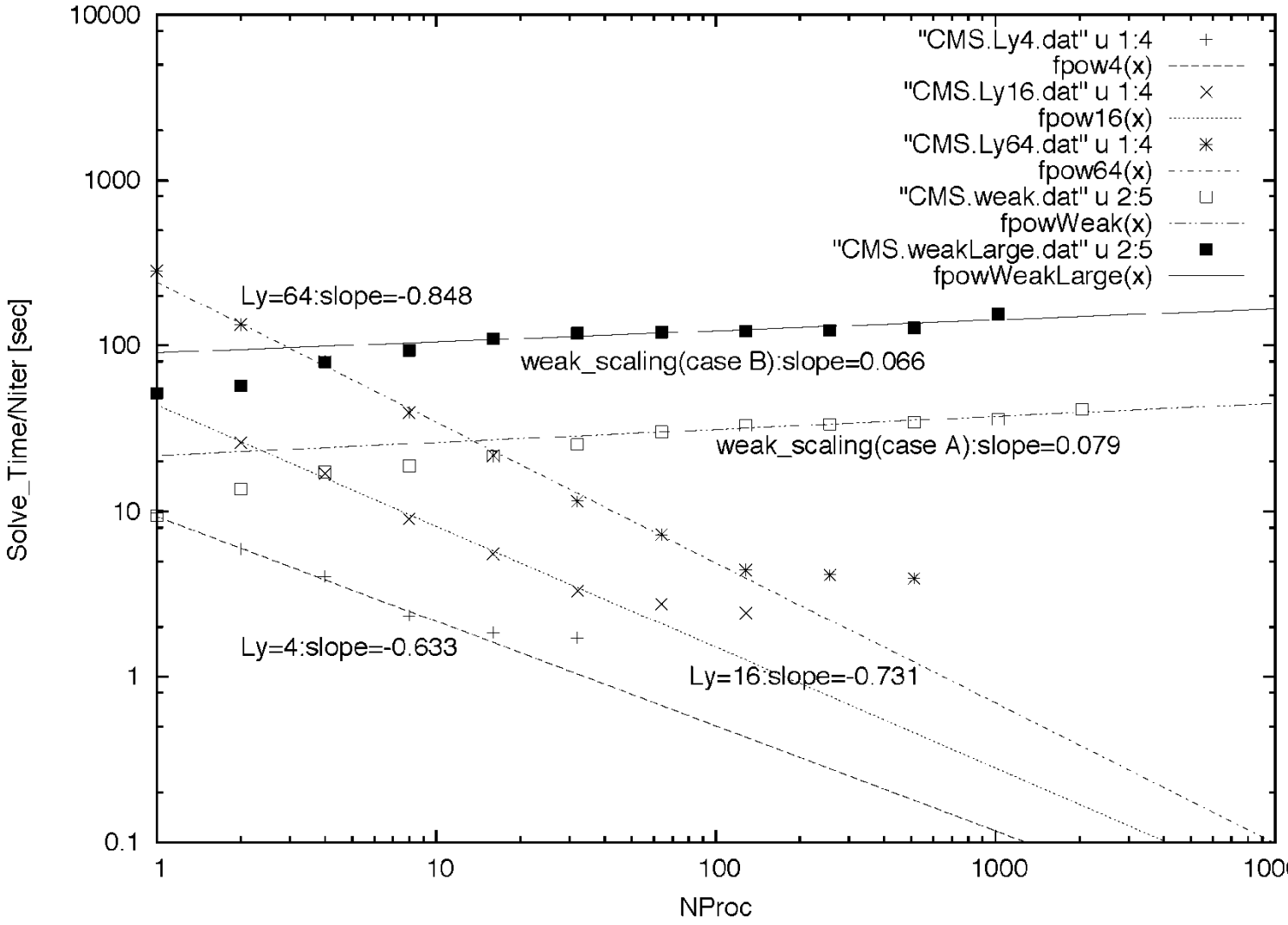
- **Schur system may have up to 80% fewer dofs**
- Big win for hard sphere systems: $\mathbf{A}_{11}$ is diagonal!
- Similar favorable structure to $\mathbf{A}_{11}$ for polymer problems using Chandler-McCoy-Singer (CMS) DFT
- More complex structure for WJDC (Werthim, Jain, Dominik, and Chapman) DFT

* M.A. Heroux, A.G. Salinger, and L.J.D. Frink, *Parallel segregated Schur complement methods for fluid density functional theories*, SIAM J. Sci. Comput., 29, (2007). pp. 1526-1535.

Sandia National Laboratories

Segregated Schur Complement Solvers*

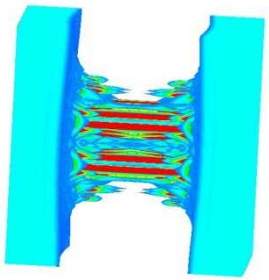
- Parallel scaling study for CMS polymer systems obtained on Jaguar (ORNL) through INCITE award



Sandia National Laboratories

Enabling Mixed-Precision and Precision Neutral Computation

- Rewrite Tramonto solver managers to template scalar, local ordinal, and global ordinal types (templated C++)
  - Arbitrary scalar types: float, complex, dd_real, qd_real **(high precision)**
  - Utilize high-precision arithmetic if double precision inadequate
  - Avoid 4GB limit of int – **allow arbitrarily large problems** (exascale necessity)
  - Enhance performance while maintaining solution accuracy
- **Template scalar type through solver stack**



Fluid-DFT Applications

Tramonto

Fluid DFT Code (C++, Templated)

Scalar Type

Trilinos

Algorithms and Enabling Technologies

Scalar Type

BELOS

Linear Solvers (C++, Templated)

Scalar Type

Tpetra parallel linear algebra library C++, Templated)

Scalar Type

TEUCHOS

Tools Library BLAS/LAPACK wrappers (C++, Templated)

Sandia National Laboratories

Precision Neutral Computation

- Reduce node-level memory bandwidth and size usage
  - Replace double with float

- Example polymer problem from *Tramonto* (8 linear solves inside Newton loop)

NCore	Float	Double	Speedup
1	3.753	10.970	2.923
2	1.766	4.195	2.375
3	1.203	2.086	1.734
4	1.380	2.643	1.915
5	1.211	2.460	2.031
6	1.056	2.313	2.190
7	1.036	2.057	1.986
8	1.524	2.387	1.566

LSQR

- LSQR*
  - Implemented in Trilinos/Belos package (C++, templated) New!
- Krylov method for **Ax=b** based upon Golub-Kahan bidiagonalization process
- Algebraically equivalent to MINRES applied to normal equations $\mathbf{A^H A x = b}$, but with better numerical properties (especially if **A** ill-conditioned)

- Governing equations

$$\mathbf{A^H U_k = V_k B_k^H} \qquad \mathbf{span(U_k) = K(AA^H, b)}$$

$$\mathbf{A V_k = U_{k+1} \bar{B}_k} \qquad \mathbf{span(V_k) = K(A^H A, A^H b)}$$

$$\left\| \mathbf{b - A x_k} \right\| = \min_{\mathbf{y}} \left\| \mathbf{b - A V_k y} \right\| = \min_{\mathbf{y}} \left\| \mathbf{e_1} \beta - \mathbf{\bar{B}_k y} \right\|$$

- Short-term recurrence; Fixed memory-footprint
- Sharp estimates of □A□, □A^{-1}□ -> estimate of cond(A)

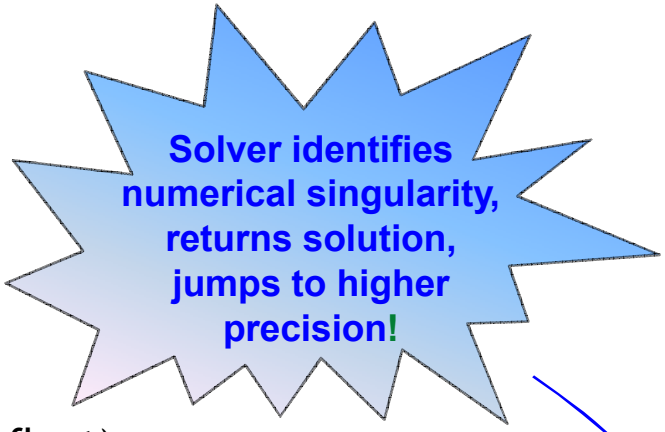
- Robustness under reduced precision
  - Return least-squares solution to Ax=b even is A numerically singular due to use of lower precision

* C. C. Paige and M. A. Saunders, LSQR: An algorithm for sparse linear equations and sparse least squares, TOMS 8(1), 43-71 (1982).

Sandia National Laboratories

LSQR

- Balance speed and solution accuracy by dynamically adapting solver precision
  - (1) Solve Ax=b in float
  - (2) If condest(A) < machEpsSingle return
  - (3) Else solve Ax=b in double
  - (4) If condest(A) < machEpsDouble return
  - (5) Else solve Ax=b in double-double
  - (6) If condest(A) < machEpsDouble-Double return
  - (7) …
- Shield end user from details of adaptive precision!
- Adaptive precision example with LSQR
  - Case #1: Well-conditioned matrix (nonsingular in float)
  - Requested relative residual tolerance = 5e-4



Scalar Type	Solve Time (s)	# Iters	CondTest	Residual Norm	Outcome
float	1.049	826	Nonsingular	4.98e-4	Success

  - Case #2: Ill-conditioned matrix (singular in float, nonsingular in double)
  - Requested relative residual tolerance = 1e-6

Scalar Type	Solve Time (s)	# Iters	CondTest	Residual Norm	Outcome
float	8.155	528	Singular	9.80e-6	Failure
double	107.568	4658	Nonsingular	9.99e-7	Success

Block Recycling Linear Solvers

- Leverage two important algorithmic techniques: **Krylov recycling + block methods**

- **Krylov subspace recycling** ♻
  - In Krylov subspace methods, building search space is dominant cost
  - For sequences of systems, get fast convergence rate and good initial guess immediately by recycling selected search spaces from previous systems
  - Family of recycling methods: Recycling GMRES (GCRODR), recycling CG (RCG), recycling MINRES (RMINRES), recycling BiCG (RBiCG).

- **Block methods**
  - Performance advantages over single-vector methods (BLAS 1 $\rightarrow$ BLAS3, SpMV $\rightarrow$ SpMM)
  - Reduce per-core bandwidth usage
  - Introduce fictitious right-hand-sides to enhance search space

Block Recycling GMRES (BGCRODR)

- Block Recycling GMRES New!
  - Implemented in Trilinos/Belos package (C++, templated)

(1) Solve $\mathbf{A_1 X_1 = B_1}$

(2) Compute **k** recycle vectors $\mathbf{U_k}$ (for example, harmonic Ritz vectors)

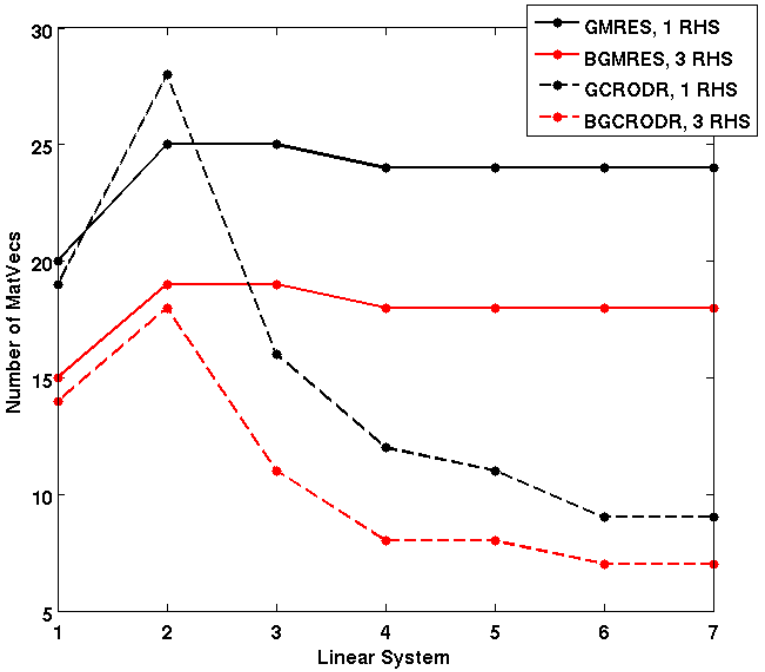
(3) Solve next linear system $\mathbf{A_2 X_2 = B_2}$ by iterating orthogonally to image of $\mathbf{U_k}$:

$$\mathbf{A_2}\begin{bmatrix}\mathbf{U_k} & \mathbf{W_m}\end{bmatrix} = \begin{bmatrix}\mathbf{C_k} & \mathbf{W_{m+1}}\end{bmatrix}\begin{bmatrix}\mathbf{I_k} & \mathbf{B_k} \\ \mathbf{0} & \underline{\mathbf{H}}_{\mathbf{m}}\end{bmatrix}$$

$$\mathbf{B_k = C_k^H A W_m} \quad \mathbf{C_k = A_2 U_k} \quad \mathbf{C_k^H C_k = I_k}$$

(4) Repeat

- Example hard sphere problem from Tramonto (electrostatics + attractions)
- 7 linear solves in from Newton loop
- Savings: 60 matvecs / 36% (1 RHS),
  50 matvecs / 40%, (3 RHS)



BGCRODR on Tramonto Polymer Example

Sandia National Laboratories

Part I
Enabling Science

Part II
Summary of Previous Work

Part III
Proposed Work

Sandia National Laboratories

Overview of Proposed Work

- **Develop new modeling capabilities for peridynamics and fluid-DFTs**
  - Capability development driven by leading edge large scale application science
- **Use Trilinos-based codes *Peridigm* and *Tramonto* as development vehicles**
- **Deploy new contributions in Trilinos**

- **Communication-avoiding and communication-eliminating capabilities**
  - Preconditioning communication-avoiding methods
  - Communication-eliminating methods (via inexact Krylov)
  - Avoiding communication with dense operators (for Schur complements)

- **Nonlocal domain decomposition**
  - Develop nonlocal extension of workhorse solvers
  - Enable scalable solution of implicit, quasistatic models

- **Scalable preconditioners for fluid-DFT WJDC functionals**
  - Enable scalable performance for important new class of fluid-DFT functional

- **Reformulate linear systems for modern hardware**
  - Reformulate linear system structure to reduce memory movement and size usage

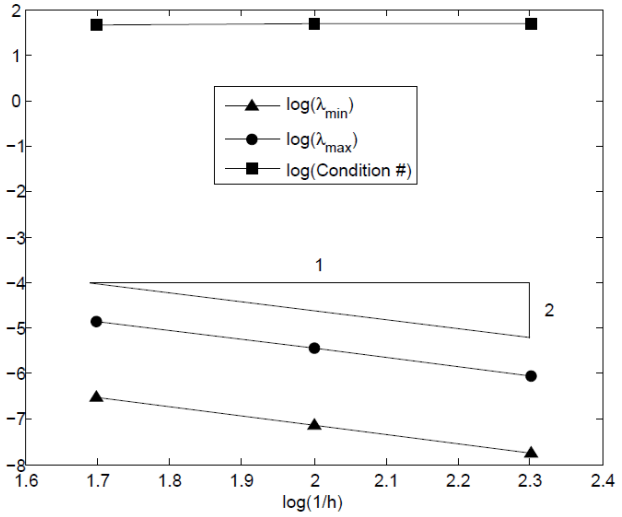
Nonlocal Domain Decomposition

- DD methods are workhorse solvers in classical computational solid mechanics
- **Goal: Extend these methods to nonlocal setting; Enable extreme scale PD**

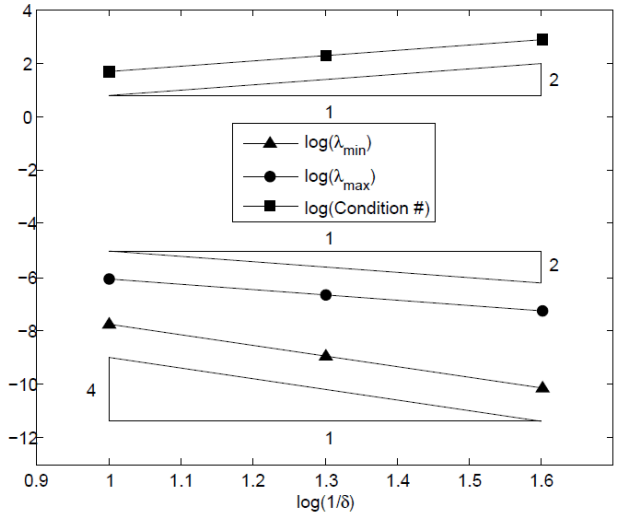
- **Nonlocal models have different numerical properties than their local counterparts!**
  - Let δ be the maximum nonlocal interaction distance. Then*,

$$\textbf{cond(K)} \leq \delta^2$$

- At most weak h-dependence; No preconditioner!



(a) Constant δ, vary h.



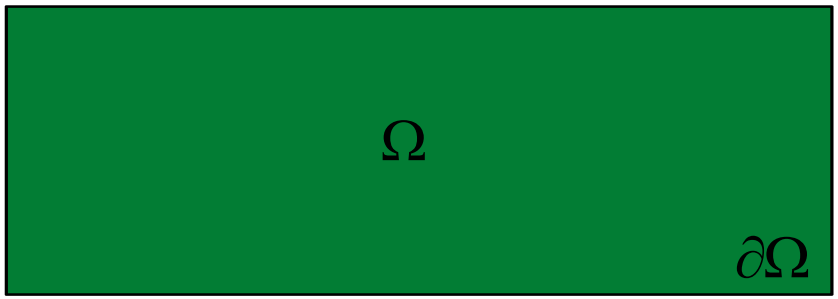
(b) Constant h, vary δ.

*B. Aksoylu and M.L. Parks, Variational Theory and Domain Decomposition for Nonlocal Problems. Applied Mathematics and Computation, 217, pp. 6498-6515, 2011.

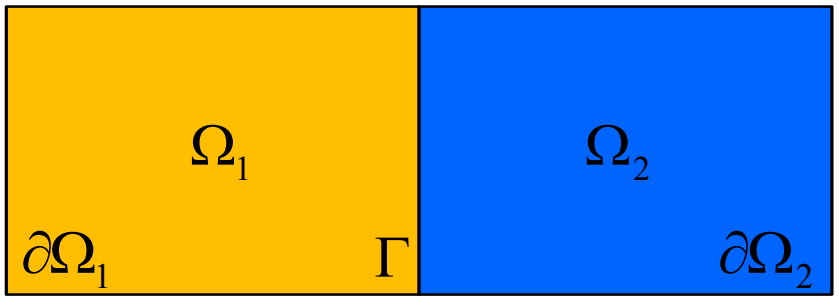
Sandia National Laboratories

Review: Classical Substructuring

- One, two domain strong formulations



$$-\nabla^2 u(x) = f \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \partial\Omega$$



$$-\nabla^2 u_1(x) = f \quad \text{in } \Omega_1 \qquad -\nabla^2 u_2(x) = f \quad \text{in } \Omega_2$$
$$u_1 = 0 \quad \text{on } \partial\Omega_1 \qquad u_2 = 0 \quad \text{on } \partial\Omega_2$$

One domain and two domain formulations equivalent

(assuming f sufficiently regular)

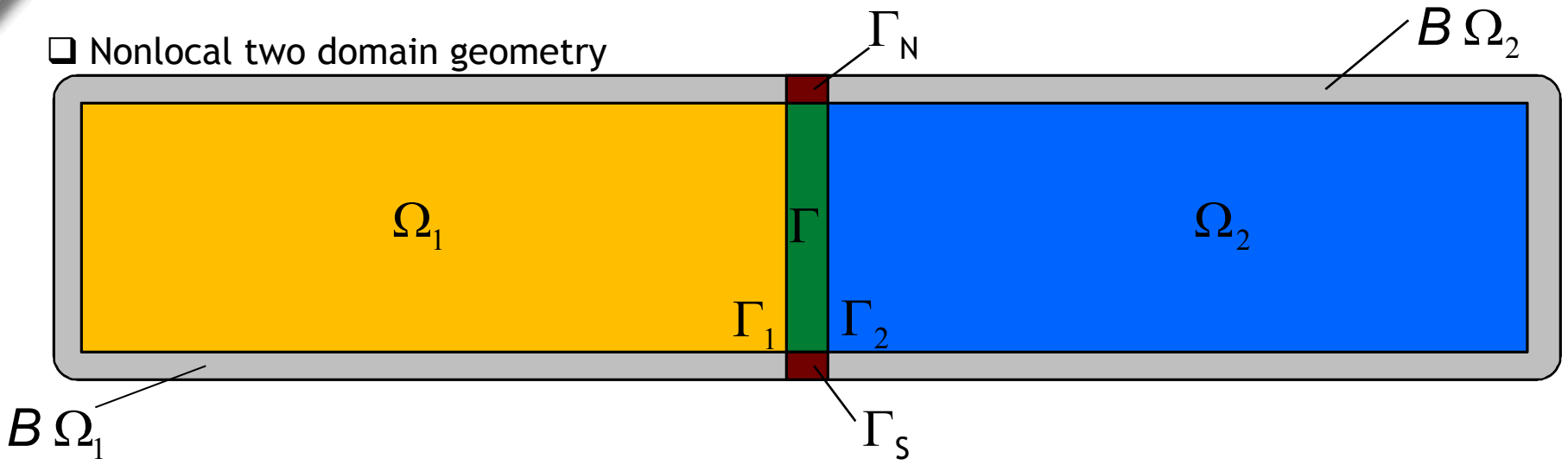
$$u_1 = u_2 \quad \text{on } \Gamma$$
$$\frac{\partial u_1}{\partial n} = -\frac{\partial u_2}{\partial n} \quad \text{on } \Gamma$$

Transmission Conditions

Sandia National Laboratories

Nonlocal Domain Decomposition

- Nonlocal two domain geometry



- **Differences from classical (local) DD**
  - Interface region is volumetric (of width δ) to decompose domains
  - Flux balance transmission condition also contains governing equation for interface region

- **Extend preliminary work to 3D solids; Implement & deploy in *Peridigm/Trilinos***
- **Hardware-aware hybrid solvers (MPI+threads)**
  - For scalable preconditioner, $\text{cond}(K) \approx O((1+\log(H/h))^2)$
  - As # cores increases, subdomain size H decreases
  - Make subdomain size H scale with # nodes, not # cores!
  - At node level, use (for example) multithreaded direct solver

Avoiding & Eliminating Communication

- Theory for communication-avoiding (CA) Krylov methods understood
- Production implementations in progress

- Open questions:

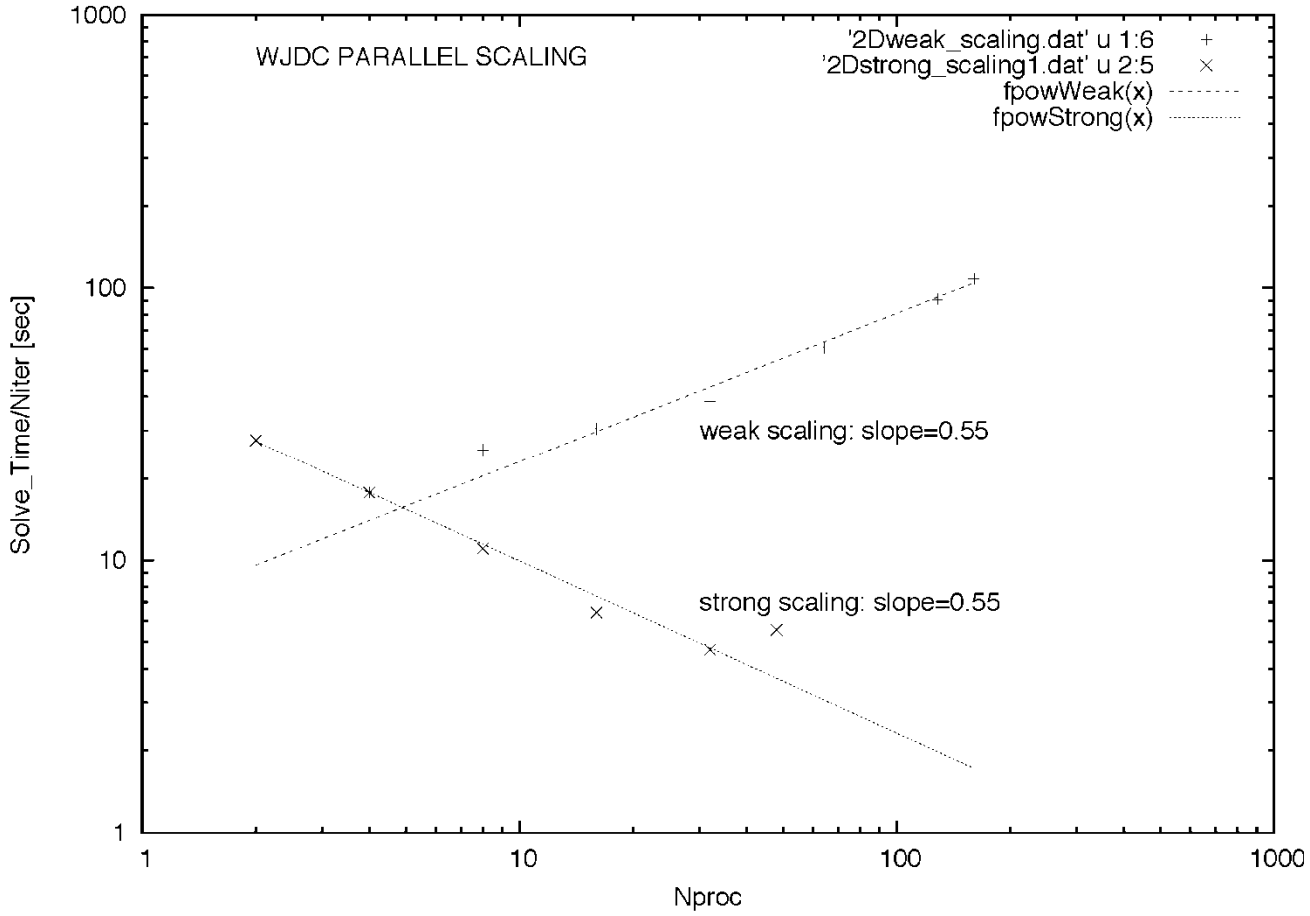
- How to precondition CA methods?
  - Apply techniques of CA matvecs to preconditioner (must know structure of PC)
  - Must interleave applications of matrix and PC

- How to improve upon CA methods? Eliminate Communication!
  - CA Krylov methods minimize communication. How to improve?
  - Eliminate some communication entirely; matvecs become inexact
  - Theory for inexact Krylov methods tells us we can drop exactness of matvecs as iteration proceeds without harming convergence of accuracy!

- Do CA methods make sense with dense operators?
  - Schur-complement methods central to large-scale PD and Fluid-DFT solvers
  - CA methods less effective on dense operators (low surface-to-volume ratio)
  - Again exploit inexact-Krylov methods to drop communication

Scalable Preconditioners

- Polymer "WJDC" DFT functional of main interest
  - WJDC scaling study on RedSky (Sandia)
  - Current performance unacceptable
  - Improve by addressing data structure needs, physics specific-preconditioner needs



Sandia National Laboratories

Improved Linear System Formulations

- Reformulate linear systems to reduce memory size usage and bandwidth
  - Introduce additional first-order parameters
  - Increase #DOFs, decrease #NZ, bandwidth

- Example: Rewrite linear system with peridynamic dilatation θ as first-order variable

	$[K][x]=[r]$	$\begin{bmatrix} K_x & K_{x\theta} \\ K_{\theta x} & K_\theta \end{bmatrix}\begin{bmatrix} x \\ \theta \end{bmatrix}=\begin{bmatrix} r_x \\ r_\theta \end{bmatrix}$
# DOF	N	4/3 N
Bandwidth (~ δ)	k	5/6 k
# Non-zeros	r	7/9 r

(all values normalized)

Results of Efforts

- **New foundational capabilities for fracture/failure modeling and fluid-DFTs**
  - Capability development driven by leading edge large scale application science
  - Use Trilinos-based codes *Peridigm* and *Tramonto* as development vehicles
  - Deploy new contributions in Trilinos

- **Communication-avoiding and communication-eliminating capabilities**
  - Preconditioning communication-avoiding methods
  - Communication-eliminating methods (via inexact Krylov)
  - Avoiding communication with dense operators (i.e., Schur complements)

- **Nonlocal domain decomposition**
  - Develop nonlocal extension of workhorse solvers
  - Enable scalable solution of implicit, quasistatic models

- **Scalable preconditioners for fluid-DFT WJDC functionals**
  - Enable scalable performance for important new class of fluid-DFT functional

- **Reformulate linear systems for modern hardware**
  - Reformulate linear system structure to reduce memory movement and size usage