SAND2013-1849P

Peridigm: A New Paradigm in Computational Peridynamics

Workshop on Nonlocal Damage and Failure

Peridynamics and Other Nonlocal Models

March 11, 2013

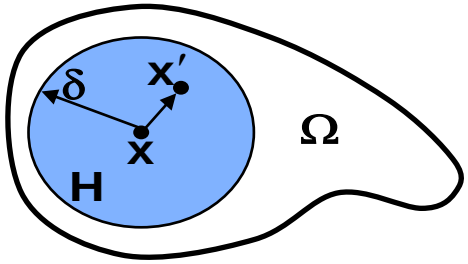
Michael Parks, Dave Littlewood, John Mitchell, Stewart Silling

Computing Research Center
Sandia National Laboratories
Albuquerque, New Mexico

SAND2013-0069P

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.

What is Peridynamics?



Point x interacts directly with all points x' within H

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

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

$$\rho\ddot{\mathbf{u}}(\mathbf{x},t) = \int_{H} \left(\mathbf{T}[\mathbf{x},t]\langle \mathbf{x}'-\mathbf{x} \rangle - \mathbf{T}[\mathbf{x}',t]\langle \mathbf{x}-\mathbf{x}' \rangle \right) dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x},t)$$

  - Replace PDEs with integral equations
  - Utilize same equation everywhere; nothing "special" about cracks
  - No assumption of differentiable fields (admits fracture)
  - Damage incurred when deformation criteria satisfied (critical stretch, etc.)
  - No obstacle to integrating nonsmooth functions
  - Integrand is "force" function; contains constitutive model
  - Integrand = 0 for points x,x' more than δ apart (like cutoff radius in MD!)
  - PD is "continuum form of molecular dynamics"

"It can be said that all physical phenomena are nonlocal. Locality is a fiction invented by idealists."

A. Cemal Eringen

- **Impact**
  - Nonlocality
  - Larger solution space than corresponding classical PDE-based models (fracture)
  - Account for material behavior at small & large length scales **(multiscale material model)**

Sandia National Laboratories

Part I
Brief Overview of Peridynamics

Part II
Demonstration Computations

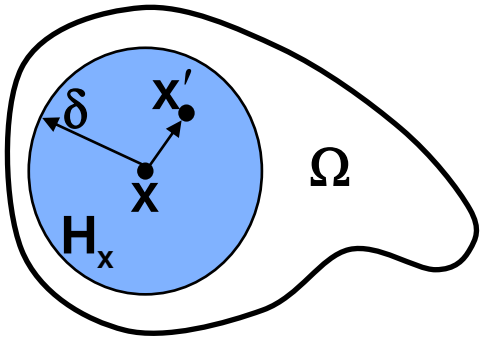
Part III
Peridigm: A Computational Peridynamics Code

Part IV
Peridigm: Tutorial and Example

Sandia National Laboratories

Peridynamics: The Basics

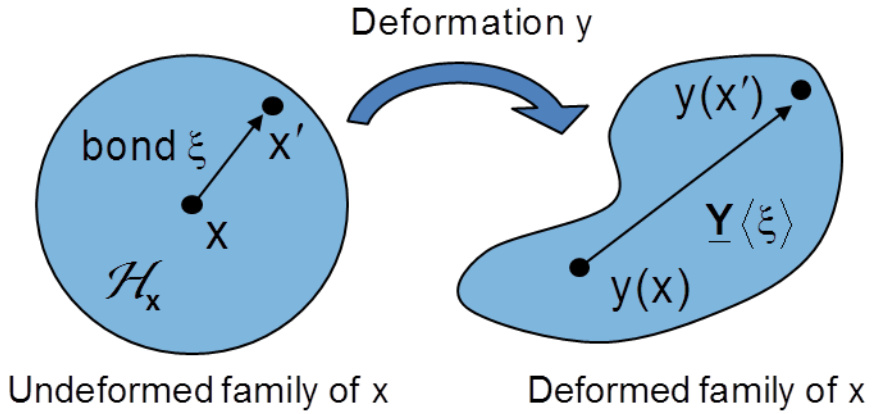
- **Horizon and family**
  - Point **x** interacts directly with all points with distance δ (***horizon***)
  - Material within distance δ of **x** is denoted H_x (***family of x***)



- **Bonds and bond forces**
  - Vector between **x** and any point in its family is called a ***bond:*** **ξ = x' – x**
  - Each bond has ***pairwise force density vector*** applied at both points: **f(x', x, t)**
  - This vector is determined jointly by collective deformation of $\mathbf{H_x}$ and collective deformation of $\mathbf{H_{x'}}$
  - Bond forces are antisymmetric: **f(x', x, t) = - f(x, x', t)**

- **Deformation state**
  - Deformation state operator **Y** maps each bond ξ into its deformed image

$$\underline{\mathbf{Y}}\langle\xi\rangle = \mathbf{y(x')} - \mathbf{y(x)}$$



Sandia National Laboratories

Peridynamics: The Basics

- **Bonds and states**
  - **f(x', x)** has contributions from material models at both **x** and **x'**

$$\mathbf{f}(\mathbf{x}',\mathbf{x}) = \mathbf{T}[\mathbf{x},t]\langle \mathbf{x}'-\mathbf{x}\rangle - \mathbf{T}[\mathbf{x}',t]\langle \mathbf{x}-\mathbf{x}'\rangle$$

  - $\underline{\mathbf{T}}[\mathbf{x}]$ is the ***force state*** – it maps bonds onto bond force densities
  - $\underline{\mathbf{T}}[\mathbf{x}]$ is determined by the constitutive model $\underline{\mathbf{T}} = \hat{\mathbf{T}}(\underline{\mathbf{Y}})$, where $\hat{\mathbf{T}}$ maps deformation state to force state
  - For elastic materials, $\underline{\mathbf{T}}[\mathbf{x}] = \mathbf{W}_{\underline{\mathbf{Y}}}$ (Fréchet derivative)

- **Peridynamics vs. standard equations**
  - Peridynamic operators and relationships between them are nonlocal analogues of standard theory

Relation	Peridynamic Theory	Standard Theory
Kinematics	$\underline{\mathbf{Y}}\langle \mathbf{x}'-\mathbf{x}\rangle = \mathbf{y}(\mathbf{x}')-\mathbf{y}(\mathbf{x})$	$\mathbf{F} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}}(\mathbf{x})$
Linear Momentum Balance	$\rho\ddot{\mathbf{u}}(\mathbf{x}) = \int_{\mathcal{H}_\mathbf{x}} \{\underline{\mathbf{T}}[\mathbf{x},t]\langle \mathbf{x}'-\mathbf{x}\rangle - \underline{\mathbf{T}}[\mathbf{x}',t]\langle \mathbf{x}-\mathbf{x}'\rangle\} dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x})$	$\rho\ddot{\mathbf{u}}(\mathbf{x}) = \nabla\cdot\boldsymbol{\sigma}(\mathbf{x}) + \mathbf{b}(\mathbf{x})$
Constitutive Model	$\underline{\mathbf{T}} = \widehat{\mathbf{T}}(\underline{\mathbf{Y}})$	$\boldsymbol{\sigma} = \widehat{\boldsymbol{\sigma}}(\mathbf{F})$
Angular Momentum Balance	$\int_{\mathcal{H}_\mathbf{x}} \{\underline{\mathbf{Y}}\langle \mathbf{x}'-\mathbf{x}\rangle \times \underline{\mathbf{T}}\langle \mathbf{x}'-\mathbf{x}\rangle\} dV_{\mathbf{x}'} = 0$	$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T$

Peridynamic Material Modeling

- **Linear Peridynamic Solid (LPS)***
- Nonlocal analog to linear isotropic elastic solid
- k is bulk modulus, μ is shear modulus

$$\rho\ddot{\mathbf{u}}(\mathbf{x},t) = \int_{\mathbf{H}} \left(\mathbf{T}[\mathbf{x},t]\langle \mathbf{x}' - \mathbf{x} \rangle - \mathbf{T}[\mathbf{x}',t]\langle \mathbf{x} - \mathbf{x}' \rangle \right) dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x},t)$$

$$\mathbf{T}[\mathbf{x},t]\langle \mathbf{x}' - \mathbf{x} \rangle = \left(\frac{3k\theta}{m} \underline{\omega}\underline{\mathbf{x}} + \frac{15\mu}{m} \underline{\omega}\mathbf{e}^{\mathbf{d}} \right) \frac{\mathbf{x}' - \mathbf{x}}{\|\mathbf{x}' - \mathbf{x}\|}$$

- Many other peridynamic material models available: elastic-plastic, viscoelastic, etc.

- Can wrap classical material models (existing material libraries) in peridynamic "skin"

*S.A. Silling, M. Epton, O. Weckner, J. Xu, & E. Askari, Peridynamic States and Constitutive Modeling, J. Elasticity, 88, pp. 151-184, 2007.

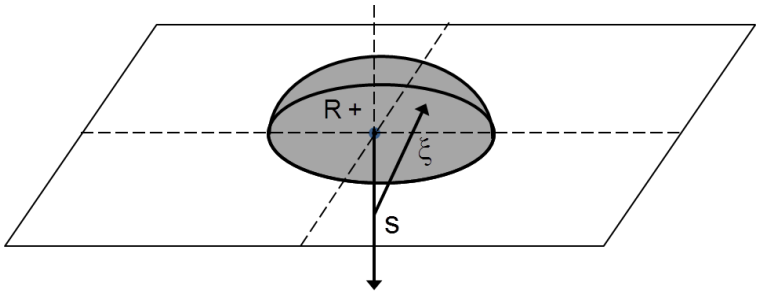
Sandia National Laboratories

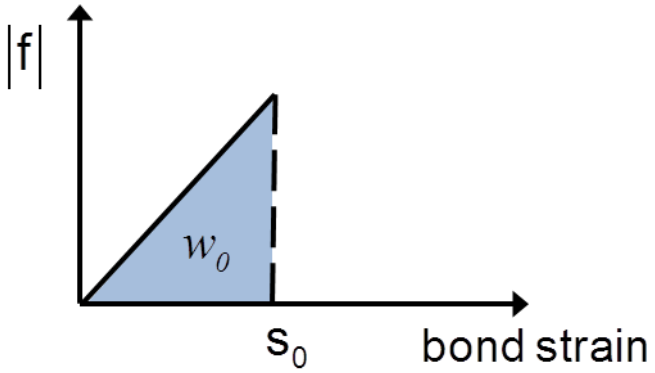
Peridynamic Fracture Modeling

- **Fracture ("Critical Stretch")**
  - Break bond if bond stretch **s** exceeds critical stretch s_0
  - If work to break bond ξ is $\mathbf{w_0(\xi)}$, then energy release rate found by summing this work per unit crack area

$$G = \int_0^{\delta} \int_{R_+} w_0(\xi) dV_\xi ds$$

  - Can then get the critical strain s_0 for bond breakage in terms of **G** (strain energy release rate), an experimentally measurable quantity





- **Fracture ("Critical Energy")***
  - Break bond if work done on bond exceeds critical value*

* J.T. Foster, S.A. Silling, W. Chen, An energy based failure criterion for use with peridynamic states, J. Multi. Comp. Engng. 9 (6): 675–687 (2011).

Sandia National Laboratories

Part I
Brief Overview of Peridynamics

Part II
Demonstration Computations

Part III
Peridigm: A Computational Peridynamics Code

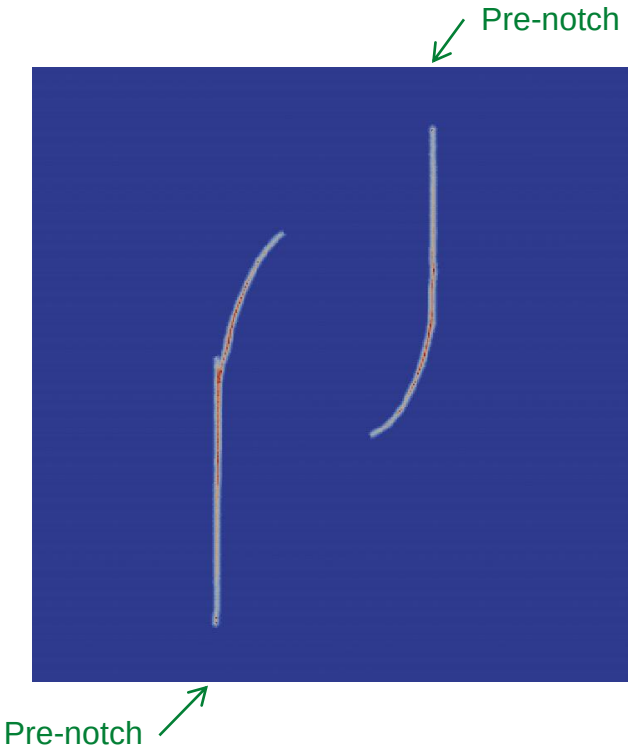
Part IV
Peridigm: Tutorial and Example

Sandia National Laboratories

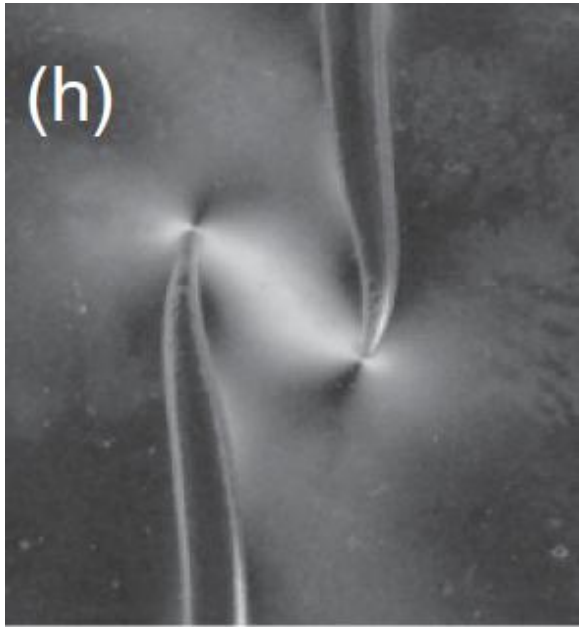
Simulation performed with PDLAMMPS code (Parks)

Two Interacting Cracks

- Offset notches thin rectangular elastic plate
- Uniaxial strain applied from sides
- Approaching cracks produce “en passant” crack pattern



Peridynamics



Physical Experiment*

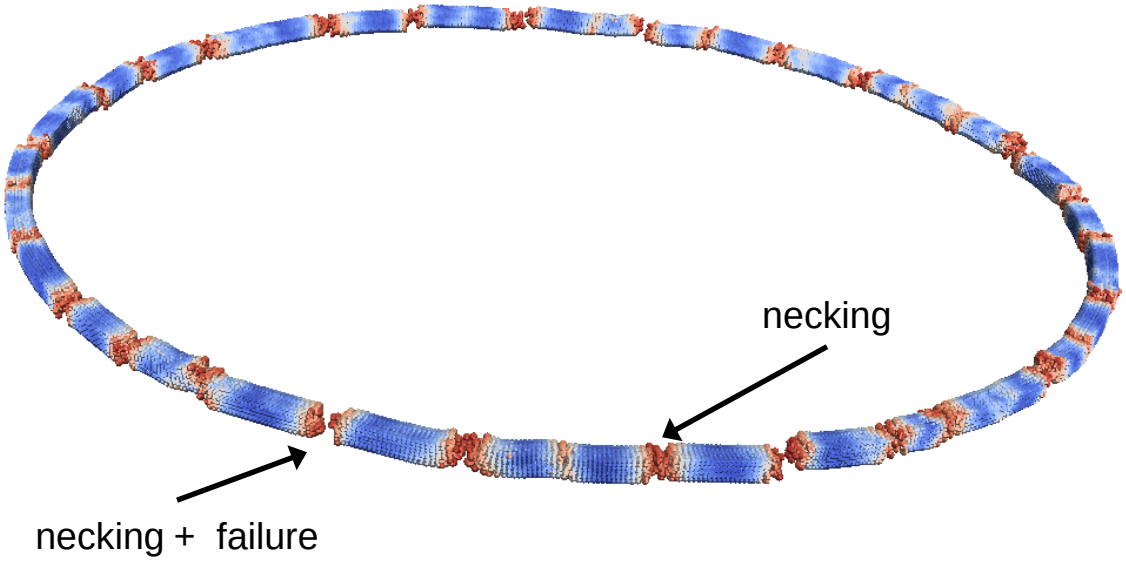
* M. Fender, F. Lechenault, and K. Daniels, Universal Shapes Formed by Two Interacting Cracks, Phys. Rev. Lett. 105, 125505 (2010) .

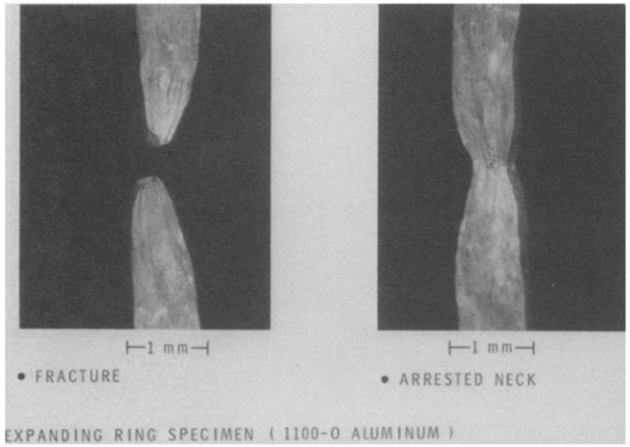
Sandia National Laboratories

Simulation performed with Peridigm

Electromagnetically loaded ring

- 1100-0 aluminum ring (ductile)
- Motivated by ring fragmentation experiments of Grady & Benson*
- Used peridynamic elastic/plastic model**





Fracture and arrested neck region from dynamic expansion of ring*

* D. Grady, D. Benson, Fragmentation of metal rings by electromagnetic loading, Experimental Mechanics, 23(4), pp. 393-400, 1983
** J. Mitchell, A Nonlocal, Ordinary, State-Based Plasticity Model for Peridynamics, SAND2011-3166, 2011.

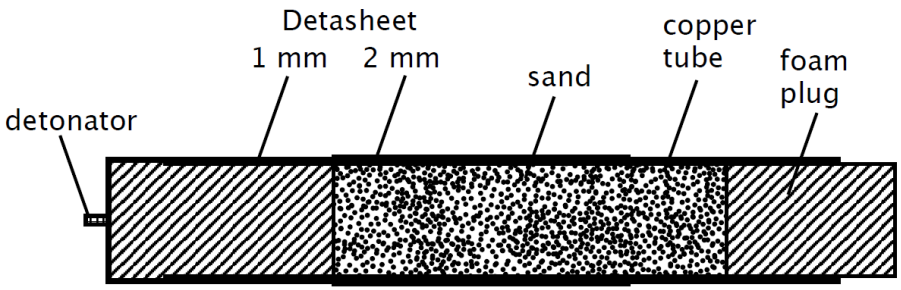
Sandia National Laboratories

Simulation performed with Peridigm

Explosively Compressed Cylinder*

- Motived by experiments of Vogler & Lappo*
- Commonly used for consolidation of powders
- Copper cylinders filled with granular material and wrapped with Detasheet explosive
- Polyurethane foam plugs used to keep granular sample in tube.

- Geometry and Material Properties
- Copper tubes 305 mm long, ID 50.8 mm, wall thickness of 1.52 mm
- PETN based Detasheet with thicknesses of 1, 2, 4, or 6 mm were used, and a
- Detonation traveled down length of tube, compressing both tube and sand fill



Cylinder schematic

Cylinder after compression

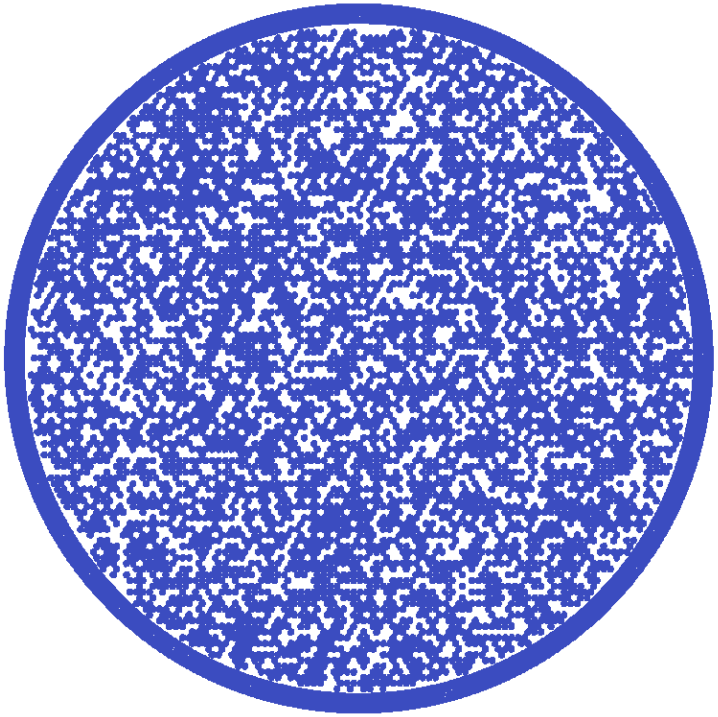
* T.J. Vogler and K.M. Lappo, Cylindrical Compaction of Granular Ceramics: Experiments and Simulations, The 12th Hypervelocity Impact Symposium. 2012.

Sandia National Laboratories

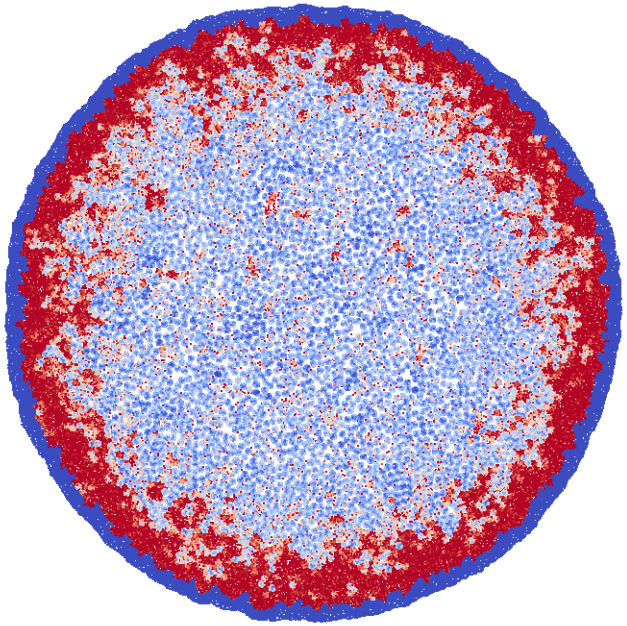
Simulation performed with Peridigm

Explosively Compressed Cylinder

- Peridigm computational results (with C. Hoffarth (ASU), D. Littlewood (SNL))
  - Color indicates damage (blue = undamaged, red = damaged)



Before



After

Sandia National Laboratories

Simulation performed with Peridigm

Tensile Test

- Stainless steel dogbone specimen.
- Apply displacement boundary conditions corresponding to 1% strain.
- Quasistatic solve

before

after

Color proportional to yield.

Displacement exaggerated 10×

Sandia National Laboratories

Part I
Brief Overview of Peridynamics

Part II
Demonstration Computations

Part III
Peridigm: A Computational Peridynamics Code

Part IV
Peridigm: Tutorial and Example

Sandia National Laboratories

Peridigm

- **Peridigm** (Open Source, C++)
- Developers: Parks, Littlewood, Mitchell, Silling
- **https://software.sandia.gov/trac/peridigm**
- Intended as Sandia's primary open-source PD code
- Built upon Sandia's Trilinos Project (**trilinos.sandia.gov**)

- **Features/Capabilities**
- Massively parallel, Exodus mesh input, Multiple material blocks

- Only state-based material models
  - elastic, elastic-plastic, elastic-plastic with hardening, viscoelastic

- Explicit time integration (Velocity-Verlet)
- Implicit time integration (Newmark-beta method)
- Quasistatics
- Nonlinear (Newton/Krylov, nonlinear CG)
- Linear (preconditioned Krylov subspace methods)
- Automatic differentiation Jacobians
- DAKOTA interface for UQ/optimization/calibration, etc. (dakota.sandia.gov)

Sandia National Laboratories

Peridigm: Peridynamics via Agile Components

Peridigm

Software Quality Tools

- Mailing Lists
- Version Control
- Build System
- Testing (CTest)
- Project Management
- Issue Tracking
- Wiki

DAKOTA

- UQ
- Optimization
- Error Estimation
- Calibration

ParaView

- Visualization

Visualization Toolkit

- Service Tools

Trilinos

Parallelization Tools

- Data Structures (Epetra)
- Load Balancing (Zoltan)

Analysis Tools

- UQ (Stokhos)
- Optimization (MOOCHO)

Services

- Interfaces (Thyra)
- Tools (Teuchos, TriUtils)
- Field Manager (Phalanx)
- DAKOTA Interface (TriKota)

Solver Tools

- Iterative Solvers (Belos)
- Direct Solvers (Amesos)
- Nonlinear Solvers (NOX)
- Eigensolvers (Anasazi)
- Preconditioners (IFPack)
- Multilevel (ML)

Sandia National Laboratories

Peridynamic Codes

- **EMU** (Export controlled, F90)
  - Developer: Silling (www.sandia.gov/emu/emu.htm)
  - Research code
- **PDLAMMPS (Peridynamics-in-LAMMPS)** (Open source, C++)
  - Developers: Parks, Seleson, Plimpton, Silling, Lehoucq
  - Particular discretization of PD has computational structure of molecular dynamics (MD)
  - LAMMPS: Sandia's open-source massively parallel MD code (lammps.sandia.gov)
  - More info & user guide: www.sandia.gov/~mlparks
- **Peridigm** (Open Source, C++)
  - Developers: Parks, Littlewood, Mitchell, Silling
  - Intended as Sandia's primary open-source PD code
  - Built upon Sandia's Trilinos Project (trilinos.sandia.gov)
  - Massively parallel, Exodus mesh input, Multiple material blocks
  - Explicit, implicit time integration
  - State-based linear elastic, elastic-plasticity, viscoelastic models
  - DAKOTA interface for UQ/optimization/calibration, etc. (dakota.sandia.gov)
- **Peridynamics in Sierra/SolidMechanics** (Export controlled, C++)
  - Developer: Littlewood
  - Sandia engineering analysis code
- **Peridynamics is a capability that can be added to (almost) any analysis code!**

Peridigm Material Models

- **Linear Peridynamic Solid (LPS)***
  - Nonlocal analog to linear isotropic elastic solid
  - k is bulk modulus, μ is shear modulus

$$\rho\ddot{\mathbf{u}}(\mathbf{x},t) = \int_{\mathbf{H}} \left(\mathbf{T}[\mathbf{x},t]\langle \mathbf{x}'-\mathbf{x}\rangle - \mathbf{T}[\mathbf{x}',t]\langle \mathbf{x}-\mathbf{x}'\rangle \right) dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x},t)$$

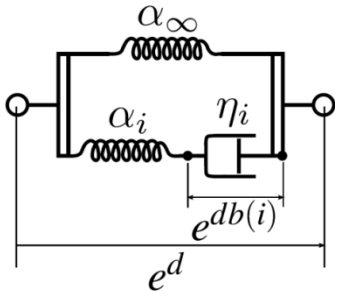
$$\mathbf{T}[\mathbf{x},t]\langle \mathbf{x}'-\mathbf{x}\rangle = \left(\frac{3k\theta}{m}\omega\underline{\mathbf{x}} + \frac{15\mu}{m}\omega\mathbf{e}^{\mathbf{d}} \right) \frac{\mathbf{x}'-\mathbf{x}}{\|\mathbf{x}'-\mathbf{x}\|}$$

- Many other peridynamic material models available:

- **Elastic-Plastic Model****
  - Nonlocal analogue to perfect plasticity model

- **Elastic-Plastic Model with Hardening (J. Foster, UTSA)**

- **Viscoelastic Model*****
  - Nonlocal analog to standard linear solid



*S.A. Silling, M. Epton, O. Weckner, J. Xu, & E. Askari, Peridynamic States and Constitutive Modeling, J. Elasticity, 88, pp. 151-184, 2007.
**J. Mitchell, A Nonlocal, Ordinary, State-Based Plasticity Model for Peridynamics, SAND2011-3166, 2011.
***J. Mitchell, A Non-local, Ordinary-State-Based Viscoelasticity Model for Peridynamics, SAND2011-8064, 2011.

Sandia National Laboratories

Peridigm: Structure & User Interface

- Peridigm designed to be user extensible
  - User defined material models, compute classes, etc.

- Compute class: Compute any user-defined quantity
  - Class must declare what data it needs allocated
    - Examples: per-element or per-node scalar, vector, tensor, etc.
  - Class must provide routine that computes user-defined quantity
  - User writes only serial code -- parallel communication handled "auto-magically"
- Example: Compute Acceleration (for output)

```
//! Fill the acceleration vector
int PeridigmNS::Compute_Acceleration::compute( Teuchos::RCP< std::vector<Block> > blocks ) const {
  int retval(0);

  Teuchos::RCP<Epetra_Vector> force, acceleration;
  std::vector<Block>::iterator blockIt;
  for(blockIt = blocks->begin() ; blockIt != blocks->end() ; blockIt++){
    force        = blockIt->getData(m_forceDensityFieldId, PeridigmField::STEP_NP1);
    acceleration = blockIt->getData(m_accelerationFieldId, PeridigmField::STEP_NP1);
    *acceleration = *force;
    double density = blockIt->getMaterialModel()->Density();
    // Report if any calls to Scale() failed.
    retval = retval || acceleration->Scale(1.0/density);
  }

  return retval;
}
```

- Declare "acceleration" as output field in input deck
  - Only specified fields are computed

Sandia National Laboratories

Peridigm: Structure & User Interface

- Peridigm material models: Create your own!

```
 //! Returns a vector of field specs that specify the variables associated with the material
 virtual Teuchos::RCP< std::vector<Field_NS::FieldSpec> > VariableSpecs() const = 0;

//! Initialize the material model.
virtual void initialize(const double dt,
                        const int numOwnedPoints,
                        const int* ownedIDs,
                        const int* neighborhoodList,
                        PeridigmNS::DataManager& dataManager) const {}


//! Evaluate the forces on the cells
virtual void computeForce(const double dt,
                          const int numOwnedPoints,
                          const int* ownedIDs,
                          const int* neighborhoodList,
                          PeridigmNS::DataManager& dataManager) const = 0;


//! Evaluate the Jacobian
virtual void computeJacobian(const double dt,
                             const int numOwnedPoints,
                             const int* ownedIDs,
                             const int* neighborhoodList,
                             PeridigmNS::DataManager& dataManager,
                             PeridigmNS::SerialMatrix& jacobian) const;


// other routines
```

Sandia National Laboratories

Peridigm: Recent New Features

- Surface Correction Factor (SCF): John Mitchell
  - In PD, points close to boundary have fewer bonds than points in bulk
  - This makes the material response different near the boundary
  - Must alter material response near boundary to correct for “missing” bonds
  - **This is extremely important for engineering simulations!**
  - **See John Mitchell’s talk: *On the 'DSF' and the 'Dreaded Surface Effect‘***

- Thermal stress: Dave Littlewood

- Bond-cutting planes: John Mitchell, Dave Littlewood

Part I
Brief Overview of Peridynamics

Part II
Demonstration Computations

Part III
Peridigm: A Computational Peridynamics Code

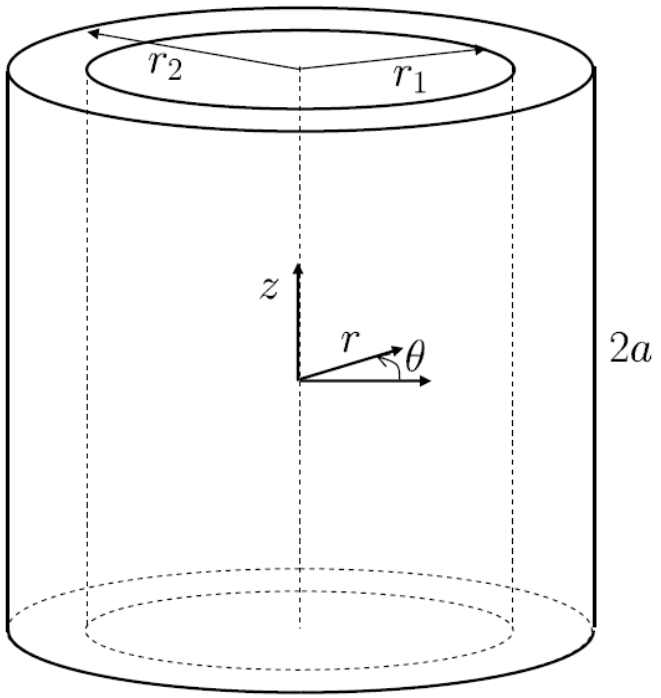
Part IV
Peridigm: Tutorial and Example

Sandia National Laboratories

Fragmenting Cylinder

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

- **Material properties**
  - Inner radius r1 = 0.020 m
  - Outer radius r2 = 0.025 m
  - Length 2a = 0.1 m
  - Density ρ = 7800 kg/m^3
  - Bulk modulus k = 130 GPa
  - Shear modulus μ = 78 GPa
  - Yield stress Y = 500 GPa
  - Ultimate stress σ = 700 GPa
  - Elongation at failure ε = 0.02



- **Initial Velocity**
  - **$v(r) = V_{r0} - V_{r1}(a/z)^2$**
  - **$v(z) = V_{z0}(a/z)$**
  - **$v(\theta) = 0$**

  where **V_{r0} = 200 m/s, V_{r1} = 50 m/s, V_{z0} = 100 m/s**

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

Sandia National Laboratories

Running Peridigm

- **Usage:**

```
Peridigm Input.xml
```

where Input.xml is a properly formatted input deck using Trilinos/Teuchos XML ParameterList :

```
<ParameterList>
        ... input section here...
</ParameterList>
```

- **An input deck contains these sections:**

```
<ParameterList>
  [Discretization Section]
  [Materials Section]
  [Damage Models Section]
  [Blocks Section]
  [Contact Section]
  [Boundary Conditions Section]
  [Solver Section]
  [Output Sections]
</ParameterList>
```

Running Peridigm

Refresh Example

❑ Input Deck

```
<ParameterList name="Discretization">
   <Parameter name="Type" type="string" value="PdQuickGrid" />
   <Parameter name="Horizon" type="double" value="0.00417462" />
   <Parameter name="NeighborhoodType" type="string" value="Spherical" />
   <ParameterList name="TensorProductCylinderMeshGenerator">
      <Parameter name="Type" type="string" value="PdQuickGrid" />
      <Parameter name="Inner Radius" type="double" value="0.020" />
      <Parameter name="Outer Radius" type="double" value="0.025" />
      <Parameter name="Cylinder Length" type="double" value="0.100" />
      <Parameter name="Ring Center x" type="double" value="0.0" />
      <Parameter name="Ring Center y" type="double" value="0.0" />
      <Parameter name="Z Origin" type="double" value="0.0" />
      <Parameter name="Number Points Radius" type="int" value="5" />
   </ParameterList>
</ParameterList>
```

Alternative: Read from Exodus/Genesis mesh generated by meshing tool (example: CUBUIT)

```
<ParameterList name="Materials">
       <ParameterList name="My Linear Elastic Material">
         <Parameter name="Material Model" type="string" value="Elastic"/>
         <Parameter name="Density" type="double" value="7800.0"/>
         <Parameter name="Bulk Modulus" type="double" value="130.0e9"/>
         <Parameter name="Shear Modulus" type="double" value="78.0e9"/>
       </ParameterList>
 </ParameterList>
```

Material properties

```
<ParameterList name="Damage Models">
       <ParameterList name="My Critical Stretch Damage Model">
         <Parameter name="Damage Model" type="string" value="Critical Stretch"/>
         <Parameter name="Critical Stretch" type="double" value="0.02"/>
       </ParameterList>
 </ParameterList>
```

Damage model properties

Sandia National Laboratories

Running Peridigm

```
<ParameterList name="Blocks">
       <ParameterList name="My Group of Blocks">
         <Parameter name="Block Names" type="string" value="block_1"/>
         <Parameter name="Material" type="string" value="My Linear Elastic Material"/>
         <Parameter name="Damage Model" type="string" value="My Critical Stretch Damage Model"/>
       </ParameterList>
 </ParameterList>

<ParameterList name="Boundary Conditions">
       <ParameterList name="Initial Velocity X">
         <Parameter name="Type" type="string" value="Initial Velocity"/>
         <Parameter name="Node Set" type="string" value="All"/>
         <Parameter name="Coordinate" type="string" value="x"/>
         <Parameter name="Value" type="string"
                                value="(200 - 50*((z/0.05)-1)^2)*cos(atan2(y,x)) + rnd(5)"/>
       </ParameterList>
       <ParameterList name="Initial Velocity Y">
         <Parameter name="Type" type="string" value="Initial Velocity"/>
         <Parameter name="Node Set" type="string" value="All"/>
         <Parameter name="Coordinate" type="string" value="y"/>
         <Parameter name="Value" type="string"
                                value="(200 - 50*((z/0.05)-1)^2)*sin(atan2(y,x)) + rnd(5)"/>
       </ParameterList>
       <ParameterList name="Initial Velocity Z">
         <Parameter name="Type" type="string" value="Initial Velocity"/>
         <Parameter name="Node Set" type="string" value="All"/>
         <Parameter name="Coordinate" type="string" value="z"/>
         <Parameter name="Value" type="string"
                                value="(100*((z/0.05)-1)) + rnd(5)"/>
       </ParameterList>
 </ParameterList>
```

Used to associate blocks of elements with specific material models

Function parser: Enter any mathematical function to specify initial velocities or boundary conditions as a function of space & time

Alternative: Specify nodesets in exodus/genesis database

Sandia National Laboratories

Running Peridigm

```
<ParameterList name="Solver">
       <Parameter name="Verbose" type="bool" value="false"/>
       <Parameter name="Initial Time" type="double" value="0.0"/>
       <Parameter name="Final Time" type="double" value="2.5e-4"/>
       <ParameterList name="Verlet">
         <Parameter name="Fixed dt" type="double" value="1.0e-8"/>
       </ParameterList>
 </ParameterList>

 <ParameterList name="Output">
       <Parameter name="Output File Type" type="string" value="ExodusII"/>
       <Parameter name="Output Format" type="string" value="BINARY"/>
       <Parameter name="Output Filename" type="string" value="fragmenting_cylinder"/>
       <Parameter name="Output Frequency" type="int" value="250"/>
       <Parameter name="Parallel Write" type="bool" value="true"/>
       <ParameterList name="Output Variables">
         <Parameter name="Proc_Num" type="bool" value="true"/>
         <Parameter name="Displacement" type="bool" value="true"/>
         <Parameter name="Velocity" type="bool" value="true"/>
         <Parameter name="Acceleration" type="bool" value="true"/>
         <Parameter name="Force_Density" type="bool" value="true"/>
         <Parameter name="ID" type="bool" value="true"/>
         <Parameter name="Dilatation" type="bool" value="true"/>
         <Parameter name="Damage" type="bool" value="true"/>
         <Parameter name="Weighted_Volume" type="bool" value="true"/>
       </ParameterList>
 </ParameterList>
```

Alternative: Implicit time integrator (Newmark-Beta) or quasistatic solver (damped Newton)

Output from compute class defined earlier

Sandia National Laboratories

Simulation performed with Peridigm

The Results…

- Fragmenting Brittle Cylinder

Before

After (brittle)

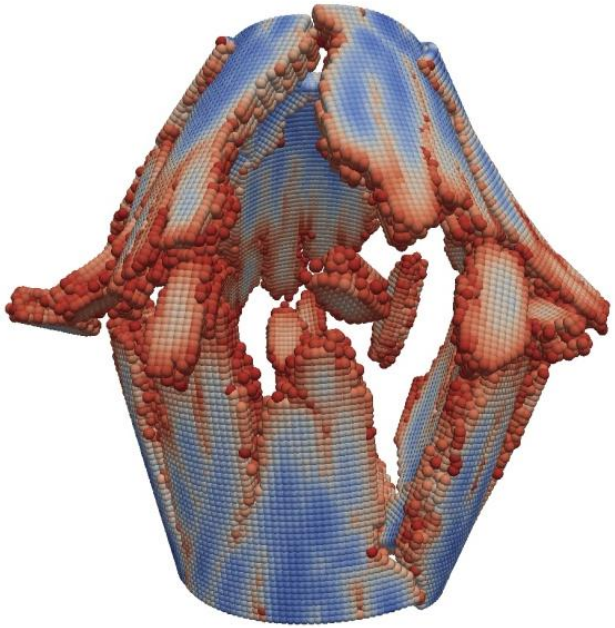
Sandia National Laboratories

Simulation performed with Peridigm

The Results…

- Fragmenting Brittle Cylinder

Before



After (brittle)

- What about ductile failure?

Sandia National Laboratories

Running Peridigm

- Modify the material section and damage model

```
<ParameterList name="Materials">
       <ParameterList name="My Elastic Plastic Material">
         <Parameter name="Material Model" type="string" value="Elastic Plastic"/>
         <Parameter name="Density" type="double" value="7800.0"/>
         <Parameter name="Bulk Modulus" type="double" value="130.0e9"/>
         <Parameter name="Shear Modulus" type="double" value="78.0e9"/>
         <Parameter name="Yield Stress" type="double" value="176.0e6"/>
       </ParameterList>
 </ParameterList>
```

Change material type and parameters

```
<ParameterList name="Damage Models">
       <ParameterList name="My Critical Stretch Damage Model">
         <Parameter name="Damage Model" type="string" value="Critical Stretch"/>
         <Parameter name="Critical Stretch" type="double" value="0.12"/>
       </ParameterList>
 </ParameterList>
```

Increase critical stretch to be consistent with elongation at failure for material.

Allows material to yield plastically before failure.

Sandia National Laboratories

Simulation performed with Peridigm

The Results…

- Fragmenting Ductile Cylinder

Before

After (ductile)

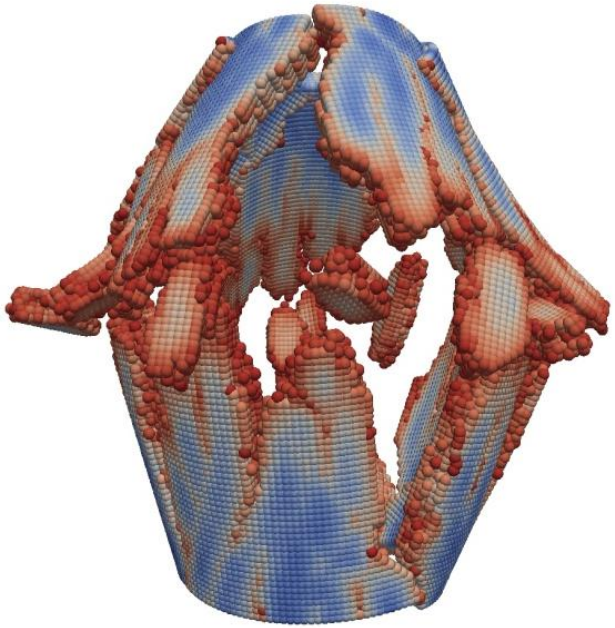
Sandia National Laboratories

Simulation performed with Peridigm

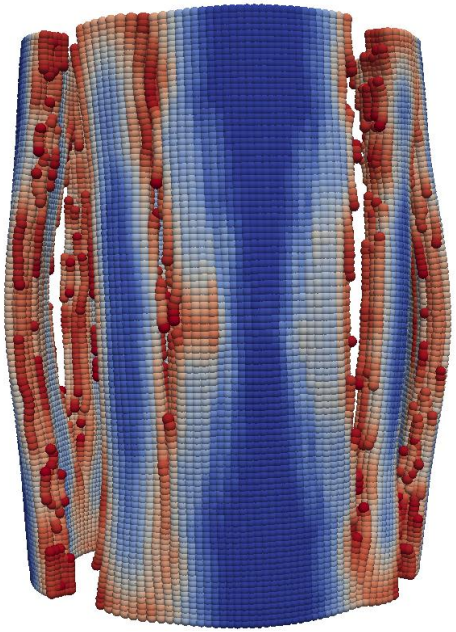
The Results…

- Fragmenting Cylinders
  - Brittle fracture characteristically different from ductile fracture

Before



After (brittle)



After (ductile)

Sandia National Laboratories

Summary

- Peridynamics overview
- Example computations
- Peridigm overview
- Peridigm fragmenting cylinder demonstration
  - Input deck distributed with Peridigm…
  - Other interesting examples (disk impact, tensile test)

- Peridigm
  - https://software.sandia.gov/trac/peridigm

- Contact me for more info: mlparks@sandia.gov

- Papers, etc.: www.sandia.gov/~mlparks

- **Thank you!**

Sandia National Laboratories