SAND2012-4984P

Peridigm: A New Paradigm in Computational Peridynamics

10th World Congress on Computational Mechanics

July 8-13 2012

Michael Parks

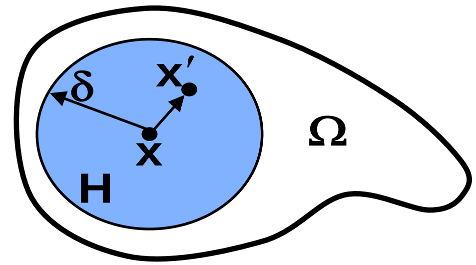
Computing Research Center
Sandia National Laboratories
Albuquerque, New Mexico

SAND2012-XXXXP

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?

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



Peridynamic Domain

- ❑ 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
  - ❑ Utilize same equation everywhere; nothing “special” about cracks
  - ❑ No assumption of differentiable fields (admits fracture)
  - ❑ When bonds stretch too much, they break
  - ❑ No obstacle to integrating nonsmooth functions
  - ❑ f(·, ·) is “force” function; contains constitutive model
  - ❑ f = 0 for particles x,x’ more than δ apart (like cutoff radius in MD!)
  - ❑ PD is “continuum form of molecular dynamics”

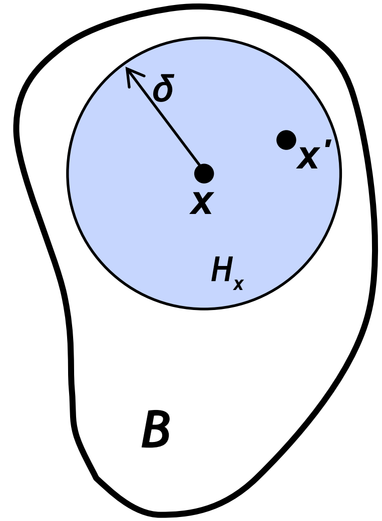
A. Cemal Eringen

- ❑ Impact
  - ❑ Nonlocality
  - ❑ Larger solution space (fracture)
  - ❑ Account for material behavior at small & large length scales (multiscale material model)

- ❑ Ancestors
  - ❑ Kröner, Eringen, Edelen, Kunin, Rogula, etc.

Sandia National Laboratories

Local vs. Nonlocal Models



- What does it mean to have a length scale?
  - What does it mean to be multiscale?
- Example #1: $\ddot{u}(x) = au''(x)$
  - Equation has no length scale; same dynamics at all scales
- Example #2: $\ddot{u}(x) = au''(x) + bu''''(x)$
  - Dimensional analysis gives that sqrt(b/a) has units of length
  - Rescaling x can make first term dominant or second term dominant
  - Scaling of x changes behavior of equation
- Peridynamic horizon δ represents a *length scale*
  - Behavior (dynamics) of EOM vary with length scale
  - Exhibit desired physics on applied length scale
- Peridynamics can provide desired dynamics at multiple length scales!
  - Rescaling space (equivalent to rescaling δ) provides transition from microscale to macroscale (classical) models!
- Connection between nonlocal models and higher-gradient models

Peridynamic Model (nonlocal)

$$\rho\ddot{u}(x,t) = \int_{-\delta}^{\delta} \frac{c}{|\varepsilon|}\left[u(x+\varepsilon,t) - u(x,t)\right]d\varepsilon$$

Taylor series

Higher-Gradient Model (weakly nonlocal)

$$\rho\ddot{u}(x,t) = K_a\left[\frac{d^2u}{dx^2} + \frac{\delta^2}{24}\frac{d^4u}{dx^4} + \frac{\delta^4}{1080}\frac{d^6u}{dx^6} + \ldots\right]$$

lim δ→0

Local, Scale Invariant

$$\rho\ddot{u}(x,t) = K_a\frac{d^2u}{dx^2}$$

Sandia National Laboratories

Part I
Brief Overview of Peridynamics

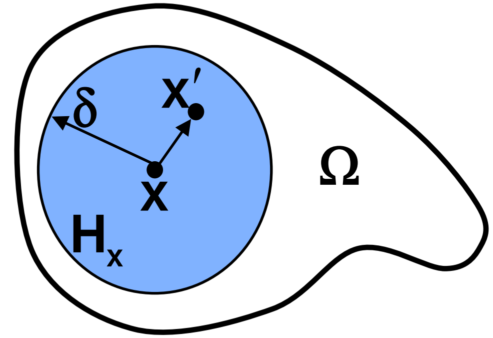
Part II
Demonstration Computations

Part III
Peridigm: A Computational Peridynamics Code

Part IV
Peridigm: Tutorial and Example

Peridynamics: The Basics

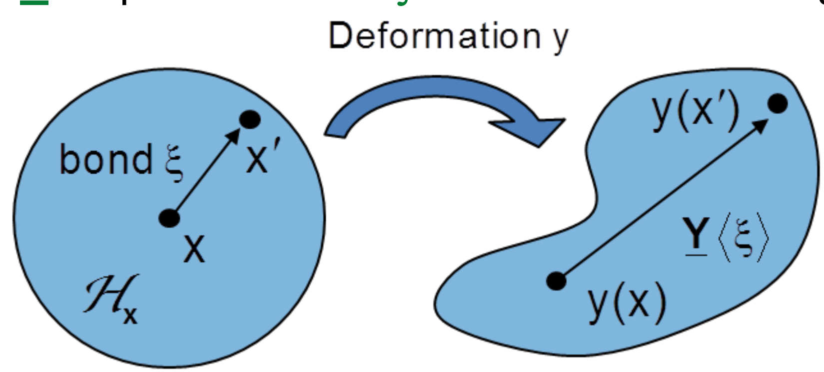
- **Horizon and family**
  - Point **x** interacts directly with all points with distance δ (***horizon***)
  - Material within distance δ of **x** is denoted H_x (***family of x***)



- **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 $\underline{\mathbf{Y}}$ maps each bond ξ into its deformed image

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



Undeformed family of x   Deformed family of x

Peridynamics: The Basics

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

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

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

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

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

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

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

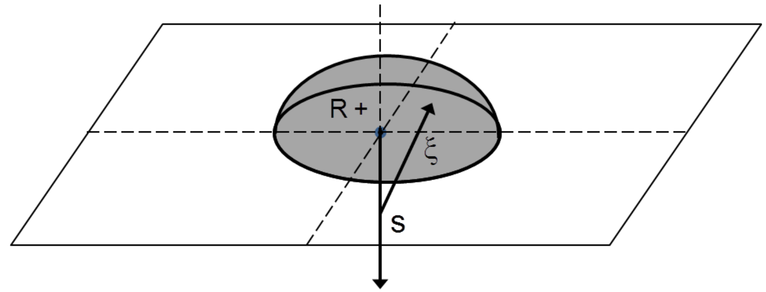
*S.A. Silling, M. Epton, O. Weckner, J. Xu, & E. Askari, Peridynamic States and Constitutive Modeling, J. Elasticity, 88, pp. 151-184, 2007.

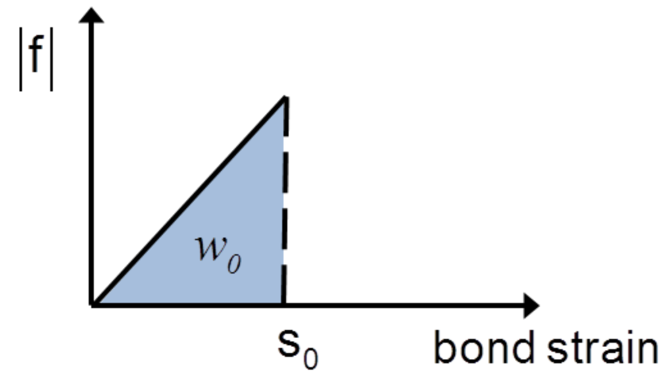
Peridynamic Fracture Modeling

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

$$\mathbf{G} = \int_{0}^{\delta} \int_{R_+} \mathbf{w_0(\xi) dV_\xi ds}$$

  - Can then get the critical strain $\mathbf{s}^*$ for bond breakage in terms of **G** (strain energy release rate), an experimentally measurable quantity





Sandia National Laboratories

Part I
Brief Overview of Peridynamics

Part II
Demonstration Computations

Part III
Peridigm: A Computational Peridynamics Code

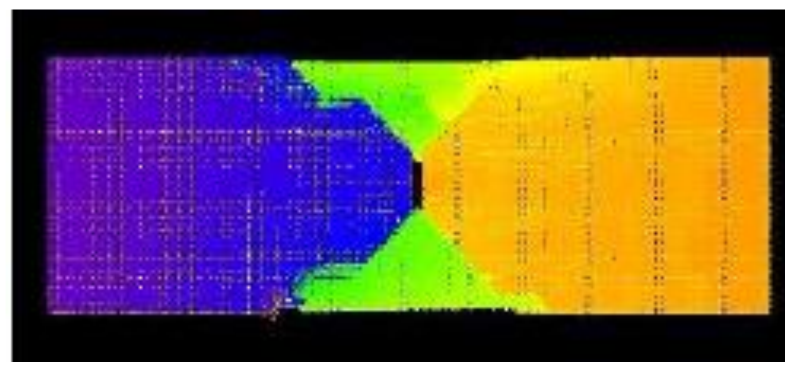
Part IV
Peridigm: Tutorial and Example

Simulation performed with EMU Fortran 90 code (Silling)

Failure in Fiber-Reinforced Composites

- ❑ **Splitting and fracture mode changes in fiber-reinforced composites***
- ❑ **Fiber orientation between plies strongly influences crack growth**

Typical crack growth in notched laminate (photo courtesy Boeing)



Peridynamic Model

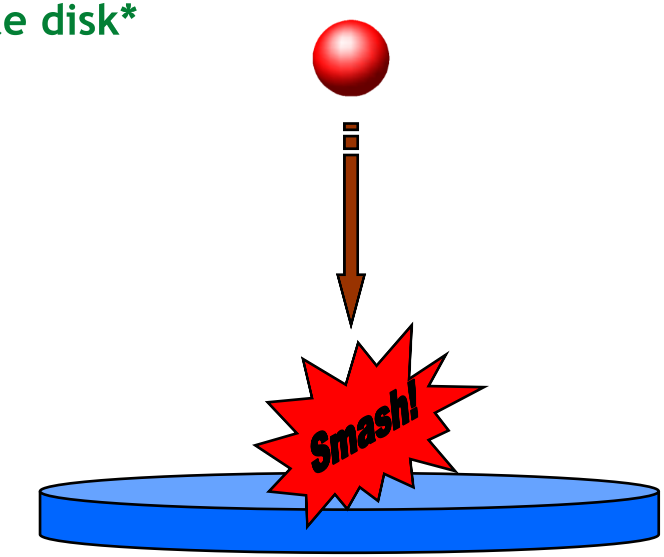
* E. Askari, F. Bobaru, R.B. Lehoucq, M.L. Parks, S.A. Silling, O.Weckner, Peridynamics for multiscale materials modeling, in SciDAC 2008, Seattle, Washington, vol. 125 of Journal of Physics: Conference Series, (012078) 2008.

Sandia National Laboratories

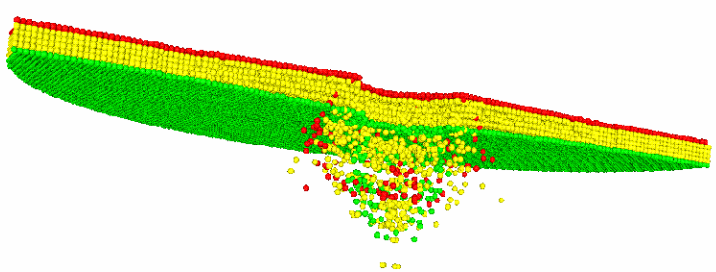
Simulation performed with PDLAMMPS code (Parks)

Hard Sphere Impact on Brittle Disk

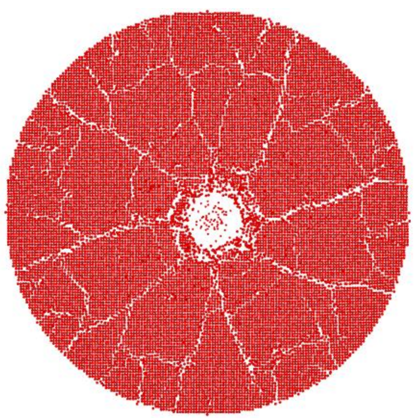
- **Example Simulation:** Hard sphere impact on brittle disk*
- Spherical Projectile
  - Diameter: 0.01 m
  - Velocity: 100 m/s
- Target Disk
  - Diameter: 0.074 m,
  - Thickness: 0.0025 m
  - Elastic modulus: 14.9 Gpa
  - Density: 2200 kg/m^3
- Discretization
  - Mesh spacing: 0.005 m
  - 100,000 particles
  - Simulation time: 0.2 milliseconds



Results



Side View



Top Monolayer

*S.A. Silling and E. Askari, *A meshfree method based on the peridynamic model of solid mechanics,* Comp. and Struct., 83, pp. 1526-1535, 2005.

Sandia National Laboratories

Simulation performed with PDLAMMPS code (Parks)

Fracture in Glass Plate

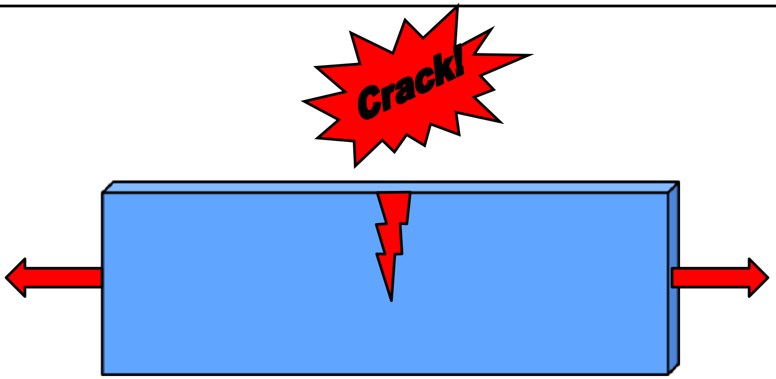
- **Example simulation:** Dynamic brittle fracture in glass
  - Joint with Florin Bobaru, Youn-Doh Ha (Nebraska), & Stewart Silling (SNL)

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

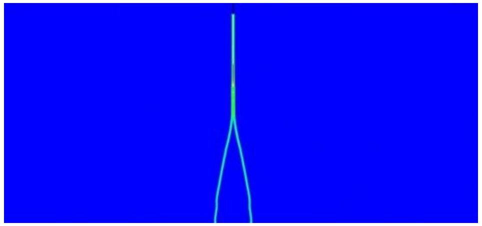
- Discretization (finest)
  - Mesh spacing: 35 microns
  - Approx. 82 million particles
  - Time: 50 microseconds (20k timesteps)

Setup

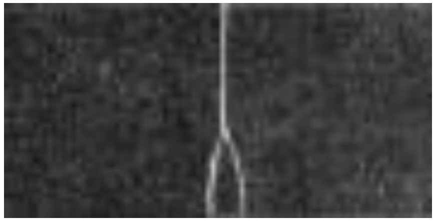
- Glass microscope slide
- Dimensions: 3” x 1” x 0.05”
- Notch at top, pull on ends



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.

Simulation performed with PDLAMMPS code (Parks)

Fracture in Glass Plate

- 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

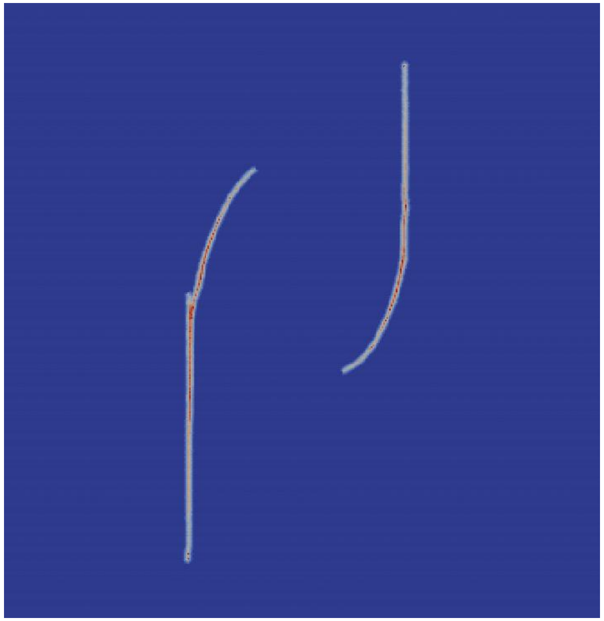
Sandia National Laboratories

Simulation performed with PDLAMMPS code (Parks)

Two Interacting Cracks

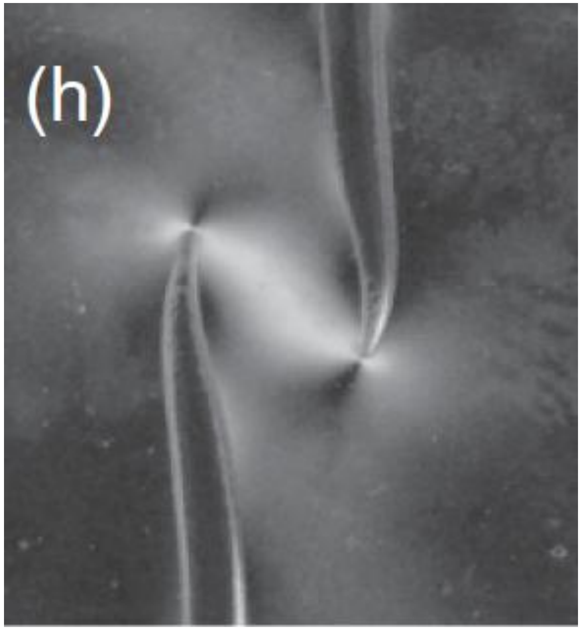
- Offset notches thin rectuangular elastic plate
- Uniaxial strain applied from sides
- Approaching cracks produce “en passant” crack pattern

Pre-notch



Pre-notch

Peridynamics



Physical Experiment*

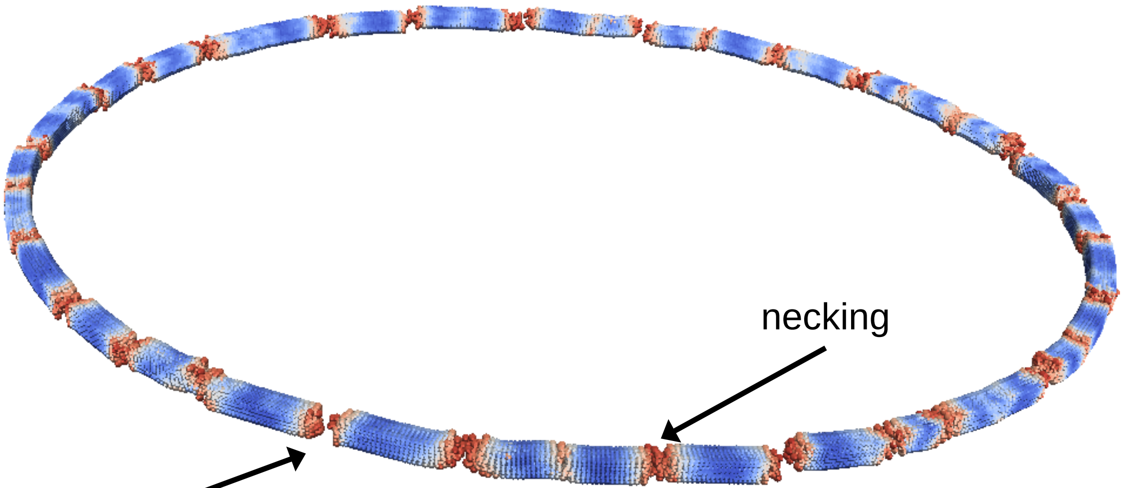
* M. Fender, F. Lechenault, and K. Daniels, Universal Shapes Formed by Two Interacting Crack, Phys. Rev. Lett. 105, 125505 (2010) .

Sandia National Laboratories

Simulation performed with Peridigm

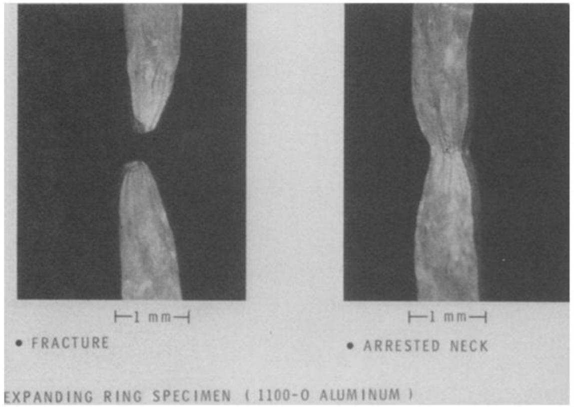
Fragmenting Metal Ring

- ❑ Motivated by ring fragmentation experiments of Grady & Benson*
- ❑ Note regions of necking and failure
- ❑ Utilized peridynamic plasticity model**



necking

necking + failure



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

Part I

Brief Overview of Peridynamics

Part II

Demonstration Computations

Part III

Peridigm: A Computational Peridynamics Code

Part IV

Peridigm: Tutorial and Example

Peridigm: A Computational Peridynamics Code

- 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
  - State-based linear elastic, elastic-plasticity, viscoelastic models
  - DAKOTA interface for UQ/optimization/calibration, etc. (dakota.sandia.gov)

- Solvers
  - Explicit time integration (Velocity-Verlet)
  - Implicit time integration (Newmark-beta method)
  - Quasistatics
  - Nonlinear (Newton/Krylov, nonlinear CG)
  - Linear (preconditioned Krylov subspace methods)

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

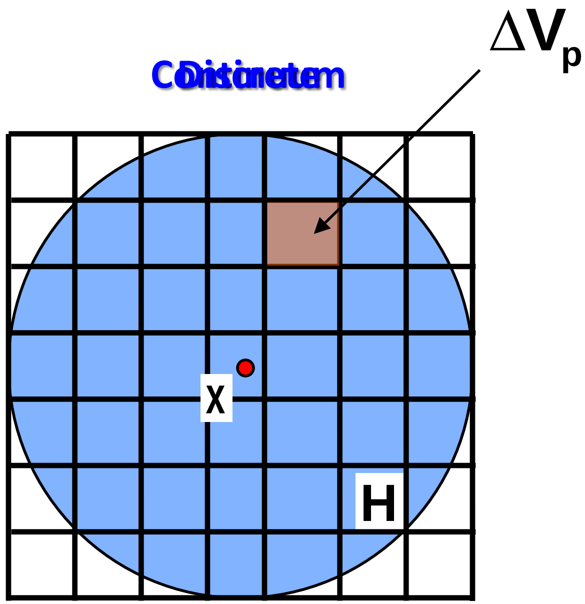
- Interfaces (Thyra)
- Tools (Teuchos, TriUtils)
- Field Manager (Phalanx)
- DAKOTA Interface (TriKota)

Solver Tools

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

Discretizing Peridynamics

- Spatial Discretization
  - Approximate integral with sum*
  - Midpoint quadrature
  - Piecewise constant approximation



$$\int_H f(u(x',t) - u(x,t), x' - x)\, dV'$$

$$\sum_p f(u(x_p,t) - u(x_i,t), x_p - x_i)\, \Delta V_p$$

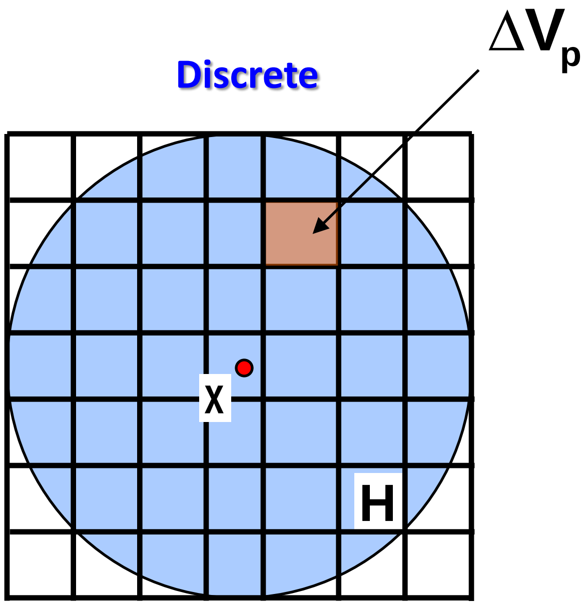
*S.A. Silling and E. Askari, *A meshfree method based on the peridynamic model of solid mechanics*, Computers and Structures, 83, pp. 1526-1535, 2005.

Discretizing Peridynamics

Spatial Discretization

- Approximate integral with sum*
- Midpoint quadrature
- Piecewise constant approximation

Discrete



$$\sum_p f(u(x_p,t) - u(x_i,t), x_p - x_i)\Delta V_p$$

Temporal Discretization

- Explicit central difference in time

$$\ddot{u}(x,t) \approx \ddot{u}_i^n = \frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2}$$

- Velocity-Verlet

$$v_i^{n+1/2} = v_i^n + \left(\frac{\Delta t}{2m}\right) f_i^n$$

$$u_i^{n+1} = u_i^n + (\Delta t) v_i^{n+1/2}$$

$$v_i^{n+1} = v_i^{n+1/2} + \left(\frac{\Delta t}{2m}\right) f_i^{n+1}$$

- Must satisfy nonlocal CFL condition
- Allows larger timesteps than local models

*S.A. Silling and E. Askari, *A meshfree method based on the peridynamic model of solid mechanics*, Computers and Structures, 83, pp. 1526-1535, 2005.

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)**

```
int PeridigmNS::Compute_Acceleration::compute(const int numOwnedPoints,
                                              const int* ownedIDs,
                                              const int* neighborhoodList,
                                              PeridigmNS::DataManager& dataManager) const {
  int retval;
  Teuchos::RCP<Epetra_Vector> force, acceleration;
  force        = dataManager.getData(Field_NS::FORCE_DENSITY3D, Field_ENUM::STEP_NP1);
  acceleration = dataManager.getData(Field_NS::ACCEL3D, Field_ENUM::STEP_NP1);
  *acceleration = *force;

  double density = peridigm->getMaterialModels()->operator[](0)->Density();
  retval = acceleration->Scale(1.0 / density);

  return retval;
}
```

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

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 {}

//! Update the constitutive data based on the current configuration.
virtual void updateConstitutiveData(const double dt,
                                    const int numOwnedPoints,
                                    const int* ownedIDs,
                                    const int* neighborhoodList,
                                    PeridigmNS::DataManager& dataManager) const = 0;

//! 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;
```

Sandia National Laboratories

Part I
Brief Overview of Peridynamics

Part II
Demonstration Computations

Part III
Peridigm: A Computational Peridynamics Code

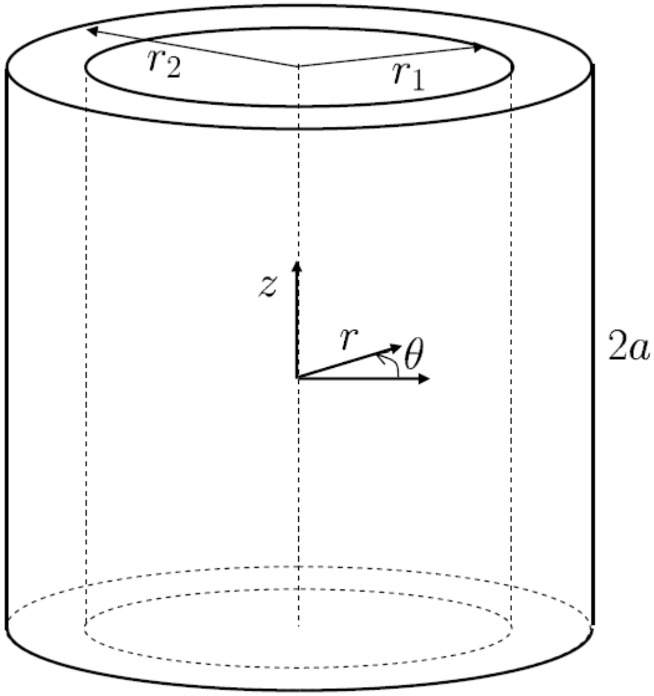
Part IV
Peridigm: Tutorial and Example

Sandia National Laboratories

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

where $\mathbf{V_{r0} = 200}$ **m/s**, $\mathbf{V_{r1} = 50}$ **m/s**, $\mathbf{V_{z0} = 100}$ **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 three sections:**

```
<ParameterList>
      [Problem Section]
      [Solver Section]
      [Output Section]
</ParameterList>
```

Running Peridigm

❑ Problem section

```
<ParameterList name="Problem">
        <Parameter name="Verbose" type="bool" value="false"/>
        <ParameterList name="Discretization">
                <Parameter name="Horizon" type="double" value="0.00417462" />
                <Parameter name="Type" type="string" value="PdQuickGrid" />
                <Parameter name="NeighborhoodType" type="string" value="Spherical" />
                <ParameterList name="TensorProductCylinderMeshGenerator">
                        <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>
        <ParameterList name="Material">
                <ParameterList name="Linear 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 name="Damage Model">
                                <Parameter name="Type" type="string" value="Critical Stretch"/>
                                <Parameter name="Critical Stretch" type="double" value="0.02"/>
                        </ParameterList>
                </ParameterList>
        </ParameterList>

        ....
        ....
```

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

Running Peridigm

- Problem section

```
    ....
    ....

     <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>
</ParameterList>
```

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

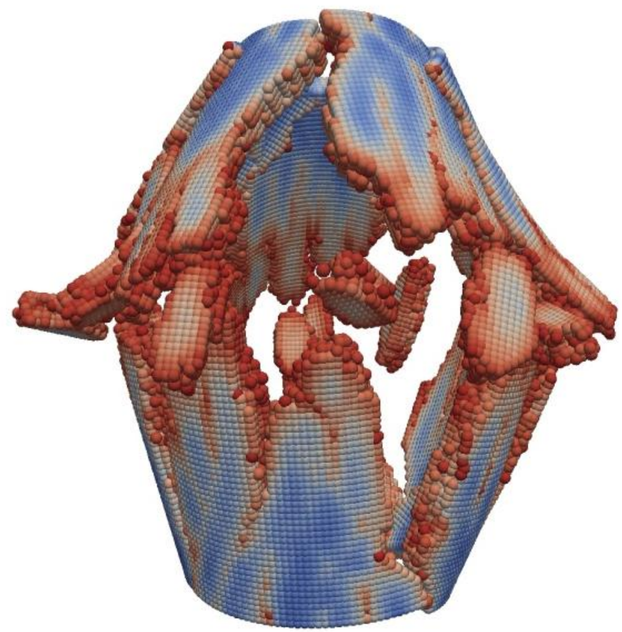
User-defined function. Define your own and reference them in the input deck

Simulation performed with Peridigm

The Results…

- ❑ **Fragmenting Brittle Cylinder**

Before



After (brittle)

- ❑ What about ductile failure?

Running Peridigm

- Modify the problem section

```
<ParameterList name="Problem">
        <Parameter name="Verbose" type="bool" value="false"/>
        <ParameterList name="Discretization">
                <Parameter name="Horizon" type="double" value="0.00417462" />
                <Parameter name="Type" type="string" value="PdQuickGrid" />
                <Parameter name="NeighborhoodType" type="string" value="Spherical" />
                <ParameterList name="TensorProductCylinderMeshGenerator">
                        <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>
        <ParameterList name="Material">
                <ParameterList name="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="Material Horizon" type="double" value="0.00417462"/>
                        <Parameter name="Yield Stress" type="double" value="176.0e6"/>
                        <ParameterList name="Damage Model">
                                <Parameter name="Type" type="string" value="Critical Stretch"/>
                                <Parameter name="Critical Stretch" type="double" value="0.12"/>
                        </ParameterList>
                </ParameterList>
        </ParameterList>
        ....
        ....
```

Change material type and parameters

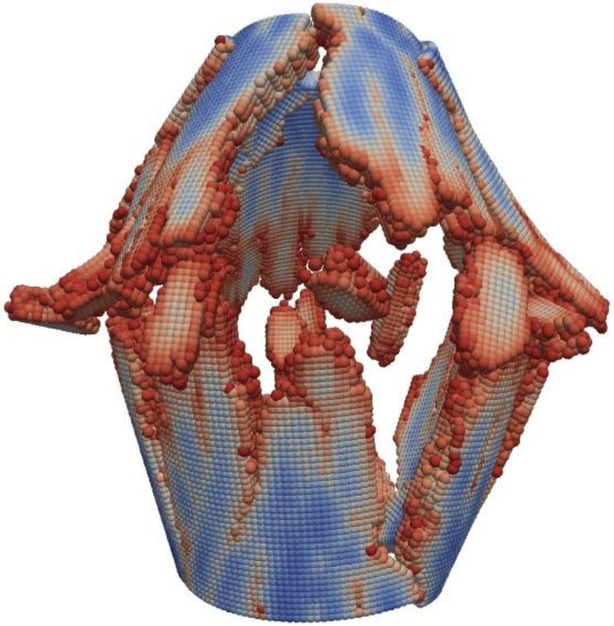
Sandia National Laboratories

Simulation performed with Peridigm

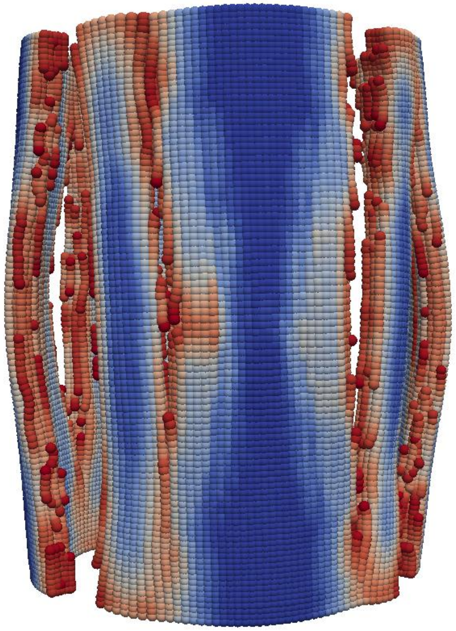
The Results…

- ❑ **Fragmenting Cylinders**
  - ❑ **Brittle fracture characteristically different from ductile fracture**

Before



After (brittle)



After (ductile)

Summary

- Peridynamics overview
  - Theory
  - Material models
  - Fracture
- Demonstration computations
- Peridigm overview
  - Open source
  - Massively parallel
  - Based upon Trilinos software components
- Peridigm tutorial
  - Cylinder setup
  - Input deck
  - Results: brittle, ductile failure

- **Codes, Papers: www.sandia.gov/~mlparks, mlparks@sandia.gov**

- **Thank you!**