Exceptional service in the national interest

Sandia National Laboratories

SAND2019-3371PE

A Conforming Reproducing Kernel Framework

for Agile Simulation of Problems with Complex Geometries

UC San Diego

JACOBS SCHOOL OF ENGINEERING
Structural Engineering

Center for Extreme Events Research

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Jacob Koester[1,2]

[1] Sandia National Laboratories, jkoeste@sandia.gov
[2] Department of Structural Engineering
Center for Extreme Events Research (CEER)
University of California, San Diego

March 21, 2019

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Outline

Introduction

Background, Meshfree Methods

Conforming Window Functions

Construction
Examples, 2D
Implementation, 3D
Examples, 3D

Addressing Volumetric Locking

Review of $\overline{\mathbf{F}}$
New Projection for $\overline{\mathbf{F}}$
Examples using $\overline{\mathbf{F}}$

Working with Low Quality Triangulations

Aggregating and Decimating

Comments and Future Work

Acknowledgments

Sandia National Labs:

Doctoral Study Program, Lab Directed Research & Development

Steve Attaway, Kim Mish, Eliot Fang, Mike Tupek, John Pott, Justine Johannes, Jim Redmond, Scott Mitchell, Joe Bishop, Kendall Pierson, Pavel Bochev, Nat Trask, Scott Roberts, Kevin Long, many more!

UCSD:

Marco Pasetto, Hayoan Wei, Qizhi He, Mike Hillman, Frank Beckwith, Edouard Yreux, Jonghyuk Baek, Xialong He, Georgios Moutsanidis, many more!

Committee:

Professors Randolph Bank, Yuri Bazilevs, Michael Holst, Petr Krysl

Professor J.S. Chen

Analyst’s Goal

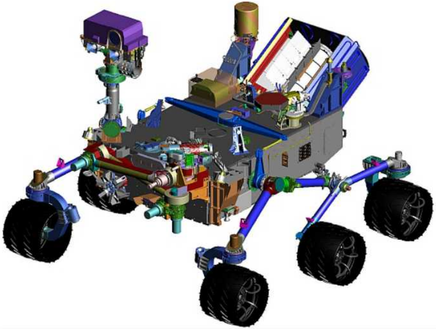
A large portion of people using the finite element method are faced with a general task:

Deliver critical engineering analyses in a timeframe consistent with project requirements

Meshing is Time Consuming

Challenging engineering analyses are common at Sandia. Goal is to have a general solution, must address the more burdensome models: *multi-body / material, complex geometries, contact, nonlinear materials, dynamic loading*

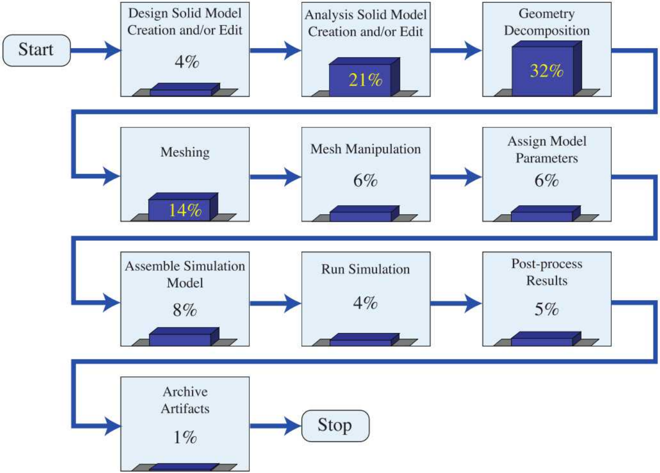
Lithium-Ion
Battery Mesostructure



Mars Rover

source: https://www.nasa.gov

Engineering Analysis, Process Cost Breakdown[1,2]



[1] M. F. Hardwick, R. L. Clay, P. T. Boggs, E. J. Walsh, A. R. Larzelere, and A. Altshuler, "DART system analysis," Sandia National Laboratories, Tech. Rep. SAND2005-4647, 2005.

[2] J. A. Cottrell, T. J. Hughes, and Y. Bazilevs, *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.

Where do we stand today?

Finite Element Methods

- Triangle / tetrahedral elements
    - Meshes are relatively easy to generate
    - Approximation property issues. E.g. volumetric locking
- Quadrilateral / hexahedral elements
    - Better approximation properties
    - Very time consuming to mesh
- Polygonal / polyhedral formulations are a recent research thrust

Immersed / Embedded Domains

- Mesh is structured and doesn't line up with the domain
- Meshing is trivial
- Challenges in obtaining agreeable quality near the implicit surfaces

Where do we stand today?

IsoGeometric Analysis

- Ties simulation to CAD by using the same basis functions for both
- Appealing solution for problems in 2D or shells
- Obtaining a volumetric representation is challenging
- Solution quality tied to the quality of CAD representation

Meshfree Methods

- Easy to generate approximation functions from point clouds
- Challenges in representing boundaries, integrating

Background: Meshfree Methods

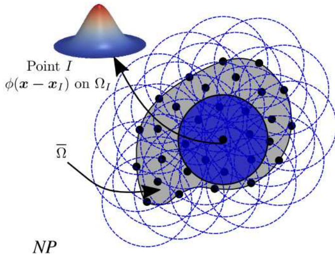
Reproducing Kernel Overview

Approximate solutions are constructed over a point cloud. Shape functions are constructed as the product of a *kernel function* and a *correction function*

$$u^h(x) = \sum_{I=1}^{NP} \Psi_I d_I; \qquad \Psi_I = C(x; x - x_I)\phi_a(x - x_I)$$

$$C(x; x - x_I) = \sum_{i=0}^{n} b_i(x)(x - x_I)^i \equiv \mathbf{H}^T(x - x_I)\mathbf{b}(x)$$

$$\mathbf{H}^T(x - x_I) = [\, 1, x - x_I, (x - x_I)^2, ..., (x - x_I)^n]$$



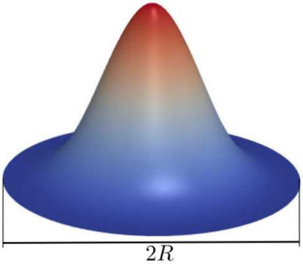
$\mathbf{b}(x)$is obtained by imposing completeness requirement: $\sum_{I=1}^{NP} \Psi_I x_I^i = x^i, \; 0 \leqslant i \leqslant n$

$$\mathbf{b}(x) = \mathbf{H}^T(0)\mathbf{M}^{-1}(x) \qquad \text{where} \qquad \mathbf{M}(x) = \sum_{I=1}^{NP} \mathbf{H}(x - x_I)\mathbf{H}^T(x - x_I)\phi_a(x - x_I)$$

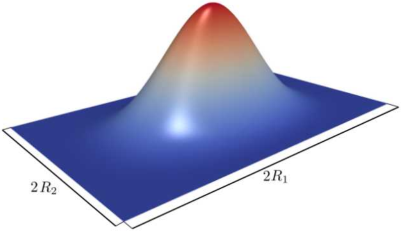
- **Kernel function: compact support, determines smoothness**
- **Correction function: provides completeness**

Window Functions

Example Euclidean Window / Kernel Functions, Cubic B-Spline



Radial

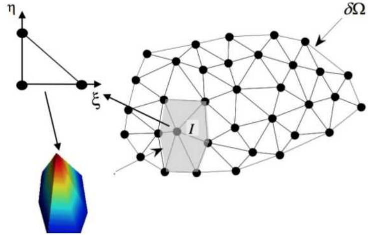


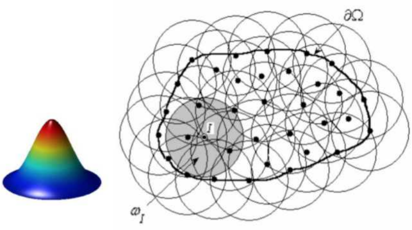
Tensor Product

$$\phi(\bar{r}) = \begin{cases} 1 - 6\bar{r}^2 + 6\bar{r}^3 & \text{for } 0 \leqslant \bar{r} \leqslant \frac{1}{2} \\ 2 - 6\bar{r} + 6\bar{r}^2 - 2\bar{r}^3 & \text{for } \frac{1}{2} \leqslant \bar{r} \leqslant 1 \\ 0 & \text{otherwise,} \end{cases}$$

$\bar{r} = r(\mathbf{x})/R$, the normalized distance.

Boundary-Related Challenges





With FEM, elements represent the domain and approximation functions are tied to elements.

With meshfree methods, approximation functions come from point clouds, boundary information is lost. Creating challenges representing:

- Concave geometries
- Material interfaces
- Essential boundaries

Model Problem

Reproducing kernel approximations functions are used in a Galerkin framework, creating the Reproducing Kernel Particle Method (RKPM).

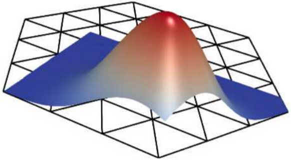
Example, Galerkin form for elastostatics is:

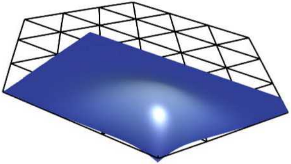
Find $\mathbf{u}^h \in U^h \subset U$ such that $\forall \mathbf{v}^h \in V^h \subset V$

$$\int_\Omega \boldsymbol{\varepsilon}(\mathbf{v}^h):\boldsymbol{\sigma}(\mathbf{u}^h)\, d\Omega = \int_\Omega \mathbf{v}^h \cdot \mathbf{b}\, d\Omega + \int_{\Gamma_h} \mathbf{v}^h \cdot \mathbf{h}\, d\Gamma$$

where $U = H_g^1$ and $V = H_0^1$. For Bubnov-Galerkin, $U^h = V^h$.

Function Properties


Window Function, ϕ


Approximation Function, Ψ

In general, approximation functions lack Kronecker delta

$$\Psi_I(\mathbf{x}_J) \neq \delta_{IJ},$$

are not interpolatory

$$u^h(\mathbf{x}_I) \neq d_I,$$

and lack the weak Kronecker delta property, which is

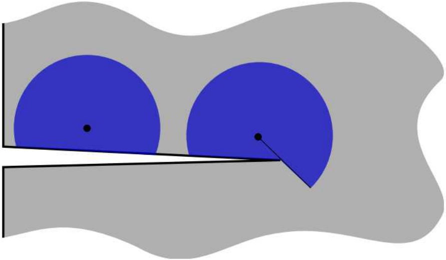
$$\Psi_I(\mathbf{x}) = 0,\ \forall \mathbf{x} \in \Gamma_g,\ I \in \mathcal{V}_o,$$

where $\mathcal{V}_o$ is the set of all points not on the essential boundary, Γ_g.

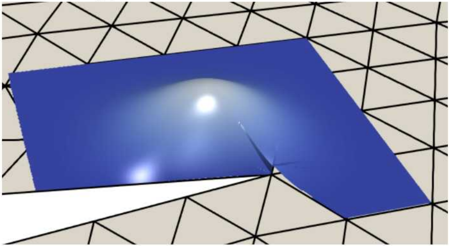
Approximation spaces are not admissible, i.e. $V^h \not\subset H_0^1$ **and** $U^h \not\subset H_g^1$

Addressing Non-Convex Domains

Visibility Method



Approximation Function, Ψ



Require an internal line-of-sight between evaluation locations and window centers. Resulting approximation functions are discontinuous. This can be problematic as the approximation space in **no longer in** H^1, as required in the Galerkin form, but it has been proven to still converge[3].

[3] P. Krysl and T. Belytschko, "Element-free Galerkin method: Convergence of the continuous and discontinuous shape functions," *Computer Methods in Applied Mechanics and Engineering*, vol. 148, no. 3–4, pp. 257–277, 1997.

Conforming Window Functions[4]

[4] J. J. Koester and J.-S. Chen, "Conforming window functions for meshfree methods," *Computer Methods in Applied Mechanics and Engineering*, vol. 347, pp. 588–621, 2019.

Research Goal

In short

- Building simulation models is very time consuming
- Meshfree methods can efficiently provide approximation spaces but complicated boundaries cause problems

Goal: ***Address the boundary challenges of meshfree methods***

- Should be systematic, requiring little or no user input
- Ideally, retain smooth approximation spaces (C^r, $r > 0$)

Conforming Window Functions

The source of the boundary-related complications are that window functions, and thus approximation functions, are not confined to the domain. To address this, a procedure for creating boundary-conforming window functions has been developed.

The conceptual steps in creating conforming windows are:

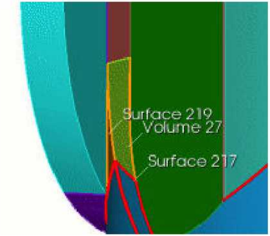
- Choose the **subdivision** strategy and create subdomains for each window
- Define the **function space** (on the subdivisions) for building the window function
- Construct the functions by **specifying the coefficients** of the space

The conforming window functions replace the traditional window functions and the rest of the RK or MLS method remains the same.

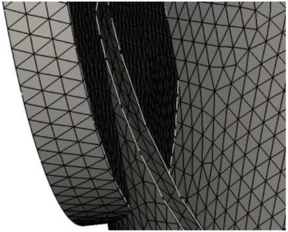
Subdivisions

Window functions will be defined on **local, overlapping, subdomains** that conform to the domain. A global, contiguous mesh is not required. Some options:

- Tensor product spaces: quadrilaterals, hexahedrons
- **Triangulations**: triangles, tetrahedrons



Example Complicated Boundary



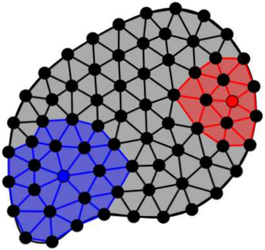
Triangulation

As with FEM meshing, decomposing a body with tensor product subdivisions is challenging, triangulating is much easier.

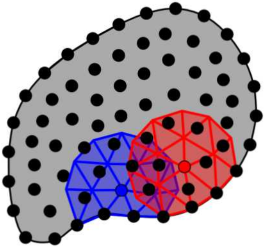
Subdivision

Overlapping window domains are required to build approximation functions

- Extract “stars” from global triangulation
- Construct local, kernel specific triangulations



Example for two vertices, using star^2



Local overlapping triangulations

Function Space for Constructing Window Functions

A means of building window functions on the subdivisions is required. **Bernstein-Bézier polynomials** are constructed on triangles / tetrahedra and theory exists for creating C^r space on triangulations.

A bivariate Bernstein-Bézier polynomial:

$$p = \sum_{i+j+k=d} c_{ijk} B^d_{ijk},$$

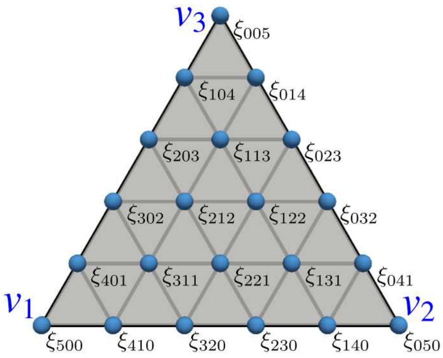
where c_{ijk} are coefficients and B^d_{ijk} are Bernstein basis polynomials of degree d, expressed using the barycentric coordinates, b_1, b_2, b_3

$$B^d_{ijk} := \frac{d!}{i!j!k!} b_1^i b_2^j b_3^k, \;\; i+j+k=d.$$

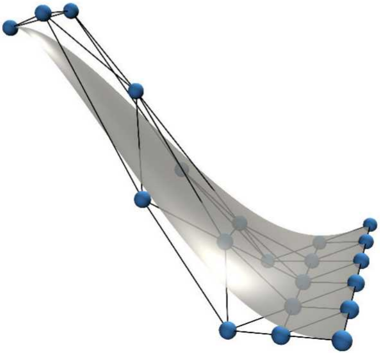
Properties include **partition of unity** and **non-negativity**.

Bernstein-Bézier Polynomial

Domain points and Bézier patch for a triangle with polynomial order, $d = 5$.



Domain points



Bézier patch with triangulation of B-coefficients

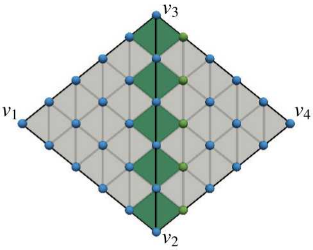
Smooth Joins

A C^r smooth join along edge $e := \langle v_2, v_3 \rangle$ of polynomials p and $\widetilde{p}$ on triangles $T := \langle v_1, v_2, v_3 \rangle$ and $\widetilde{T} := \langle v_4, v_3, v_2 \rangle$:

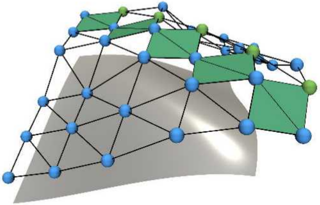
$$D^n_{\mathbf{u}} p(\mathbf{v}) = D^n_{\mathbf{u}} \widetilde{p}(\mathbf{v}), \ \ \forall\, \mathbf{v} \in e \text{ and } n = 0, \cdots, r,$$

where $\mathbf{u}$ is any direction not parallel with e. This leads to the continuity constraints,

$$\widetilde{c}_{njk} = \sum_{\nu+\mu+\kappa=n} c_{\nu\, k+\mu\, j+\kappa} B^n_{\nu\mu\kappa}(v_4), \ \ j+k = d-n, \ \ n = 0, \cdots, r.$$



C^1 Join, domain points

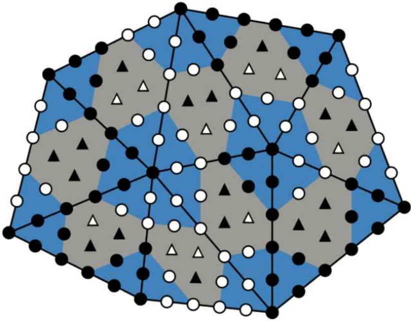


Bézier patch with triangulation of B-coefficients

Smooth Spaces

Example smooth space $\mathcal{S}_5^{1,2}$ (Argyris Space, quintic, C^1 on edges, C^2 at vertices)

Minimal determining set example. Circles are coefficients associated with a vertex (blue region), triangles for edges, filled are free and open are constrained.



Nodal minimal determining set: Values and derivatives set at specific locations (Hermite interpolation) to define functions:

$$
\begin{aligned}
\mathcal{N} &:= \bigcup_{v \in \mathcal{V}} \mathcal{N}_v \cup \bigcup_{e \in \mathcal{E}} \mathcal{N}_e, \\
\mathcal{N}_v &:= \left\{ \epsilon_v D_{\mathbf{x}}^{\alpha} D_{\mathbf{y}}^{\beta} \right\}, 0 \leqslant \alpha + \beta \leqslant 2, \\
\mathcal{N}_e &:= \left\{ \epsilon_{\boldsymbol{\eta}_e} D_{\mathbf{u}_e} \right\}
\end{aligned}
$$

where $\boldsymbol{\eta}_e$ and $\mathbf{u}_e$ are the midpoint and unit vector normal of edge e. ϵ_t is the point evaluation functional defined as $\epsilon_t f = f(t)$.

Building Conforming Window Functions

Subdivisions with spaces are defined, need to build the window functions by specifying the coefficients of the spaces.

To set the coefficients, Hermite interpolation is used. Function values and derivatives needed at nodal parameter locations. Two methods of determining the values have been developed.

- Star windows: replace Euclidean with graph distances
- Snap windows: projection of traditional window functions to the conforming window space

Star Window Functions

Function values and derivative for Hermite interpolation.

Modify a traditional, radial cubic B-spline window function

Traditional window function:

$$\phi(\bar{r}) = \begin{cases} 1 - 6\bar{r}^2 + 6\bar{r}^3 & \text{for } 0 \leqslant \bar{r} \leqslant \frac{1}{2} \\ 2 - 6\bar{r} + 6\bar{r}^2 - 2\bar{r}^3 & \text{for } \frac{1}{2} \leqslant \bar{r} \leqslant 1 \\ 0 & \text{otherwise,} \end{cases}$$

$\bar{r} = r(\mathbf{x})/R$, the normalized distance.

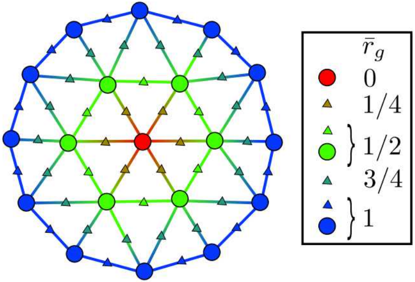
Replace $\bar{r}$ with $\bar{r}_g$, the normalized graph distance:

$$\bar{r}_g = \begin{cases} 1 & \forall\, v_I \in \mathcal{N}_b,\ \ v_0 \notin \mathcal{N}_b \\ d_g(v_0, v_I)/R_g & \text{otherwise} \end{cases}$$

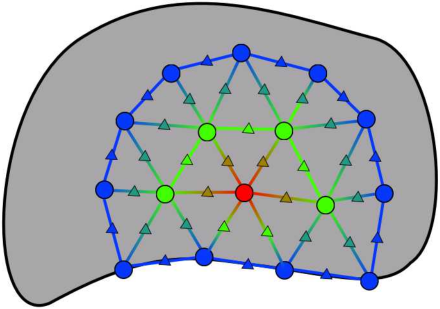
$d_g(v_0, v_I)$ is the graph distance (integer) between vertex v_I and the center v_0, R_g is the chosen graph extent (an integer, e.g. star^{R_g}), $\mathcal{N}_b$ is the set of nodal parameter locations on conforming boundaries.

Normalized Graph Distances

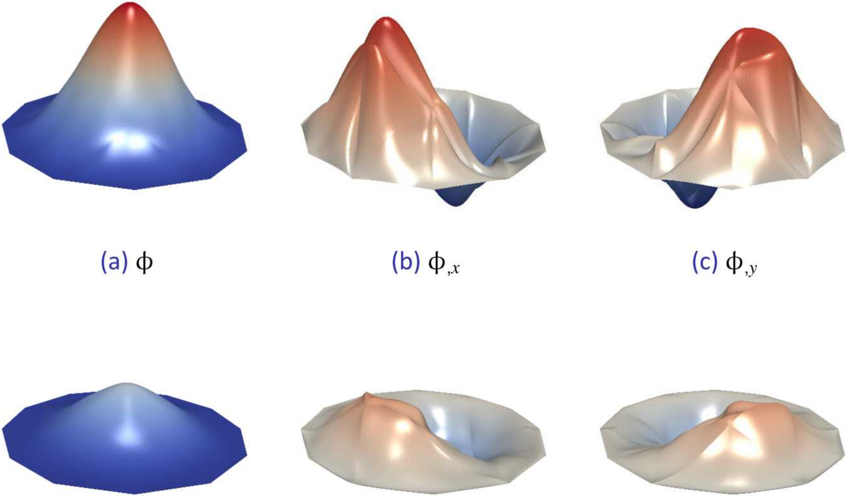
Normalized graph distances, d_g, at the $\mathcal{S}_5^{1,2}$ nodal parameter locations for a second order stars:



Away from a boundary



Near a boundary



(a) ϕ

(b) $\phi_{,x}$

(c) $\phi_{,y}$

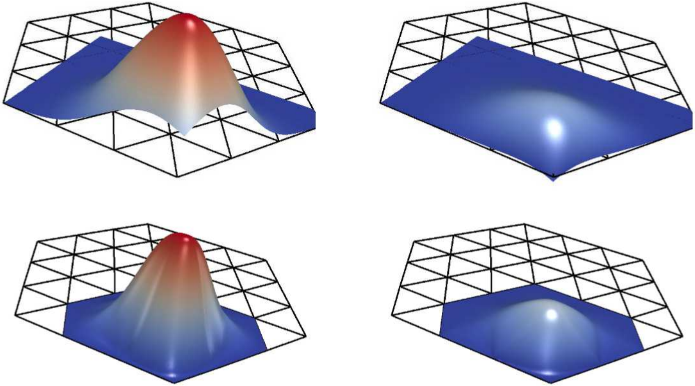
(d) Ψ

(e) $\Psi_{,x}$

(f) $\Psi_{,y}$

Figure: Interior conforming windows and approximation functions

Conforming to Essential Boundary



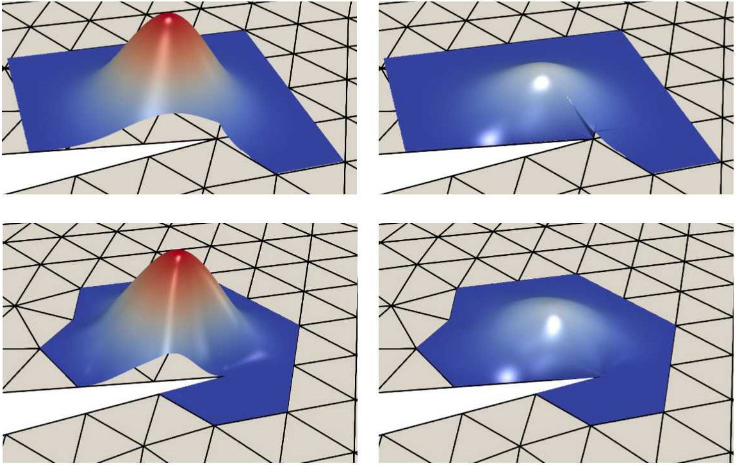
Non-conforming at essential boundary

Conforming at essential boundary

ϕ_I
(Window Function)

Ψ_I
(Approximation Function)

Near Non-Convex Region



Non-conforming window with visibility check

Conforming window with convexity improvement

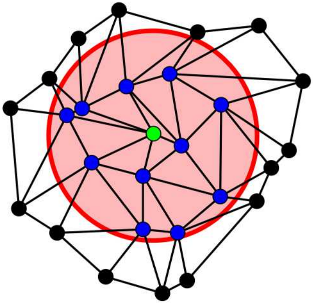
ϕ_I
(Window Function)

Ψ_I
(Approximation Function)

Snap Star

Project traditional window function to conforming window space.

- Use all elements that are contained or intersect a Euclidean ball
- Use normalized **Euclidean distance** for nodal locations inside the ball
- Set the normalized distance for nodal location outside the ball



Properties

- Less sensitive to quality of the triangulation
- Retains control for conforming to boundaries, improving near non-convex regions

Snap Star

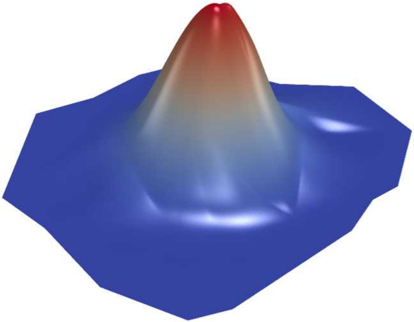


Figure: Snap Star

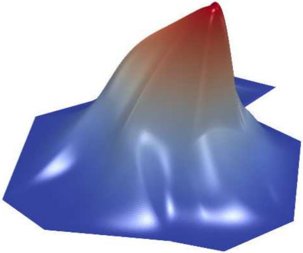


Figure: Star^2

Local Star

Construct window function using local triangulations, not from a global mesh

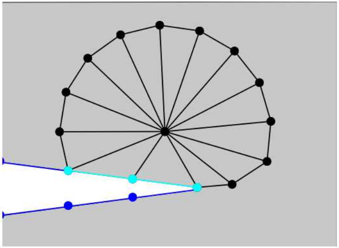


Figure: Local Triangulation

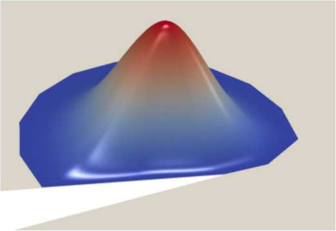


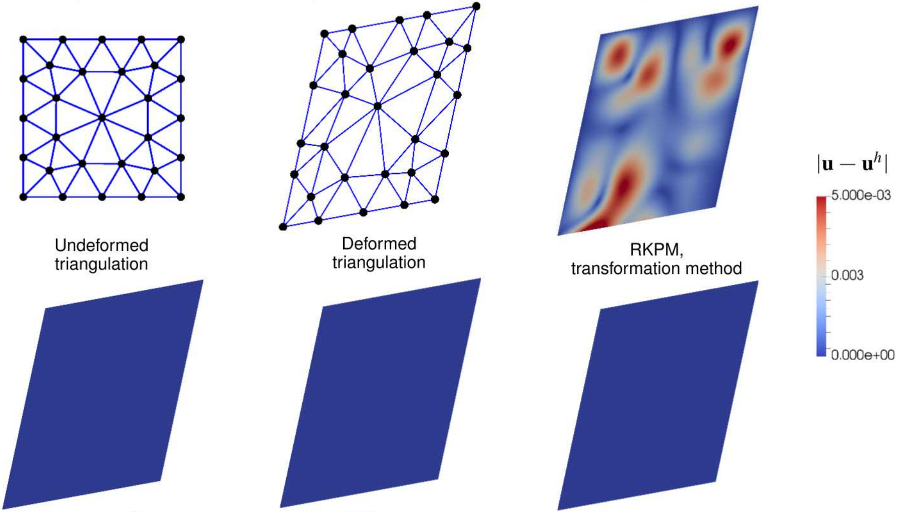
Figure: Local Star

Examples, 2D

Conforming Reproducing Kernel (CRK)

Essential Boundary Conditions

Elastic, patch test. Boundary conditions give a linear displacement field.



Undeformed triangulation

Deformed triangulation

RKPM, transformation method

CRK $star^2$

CRK snap

CRK local

Essential Boundary Conditions

Patch test error:

Method	L^2	H_1
RKPM with transformation method	2.05e-03	2.44e-02
CRK, $star^2$, static condensation	7.65e-17	1.04e-15
CRK, snap, static condensation	8.57e-17	1.78e-15
CRK, local, static condensation	6.55e-17	8.63e-15

Transformation method:

- $\Psi_I(\mathbf{x}_J) = \delta_{IJ}$
- $u^h(\mathbf{x}_I) = d_I$
- No Weak Kronecker-delta, inadmissible approximation spaces

Conforming window:

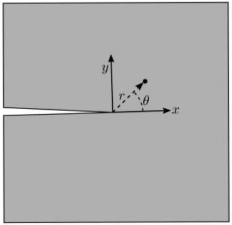
- $\Psi_I(\mathbf{x}_J) \neq \delta_{IJ}$
- $u^h(\mathbf{x}_I) = d_I$
- Weak Kronecker-delta → **admissible approximation spaces**
- → **Directly impose values along entire boundary** (like FEM)

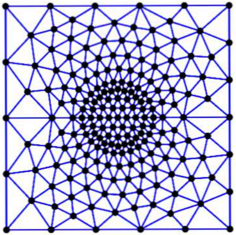
Non-Convex Domain

An elastic plate with an edge crack

- $E = 3.E7$, $\nu = 0.3$
- Mode I loading
- Exact displacement along edges (except re-entrant edges)
- Plane strain
- $\lambda = 0.5$
- $Q = 1/3$

$$\sigma_{xx} = \lambda r^{\lambda-1}\left[(2 - Q(\lambda+1))\cos((\lambda-1)\theta) - (\lambda-1)\cos((\lambda-3)\theta)\right]$$
$$\sigma_{yy} = \lambda r^{\lambda-1}\left[(2 + Q(\lambda+1))\cos((\lambda-1)\theta) + (\lambda-1)\cos((\lambda-3)\theta)\right]$$
$$\sigma_{xy} = \lambda r^{\lambda-1}\left[(\lambda-1)\sin((\lambda-3)\theta) + Q(\lambda+1)\sin((\lambda-1)\theta)\right]$$





Results Comparison

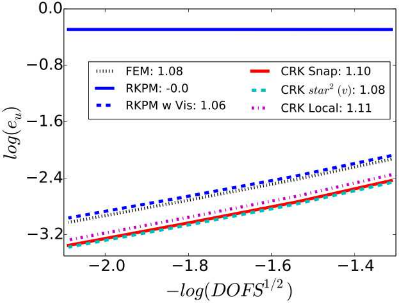


Figure: Convergence in $\mathbf{u}$

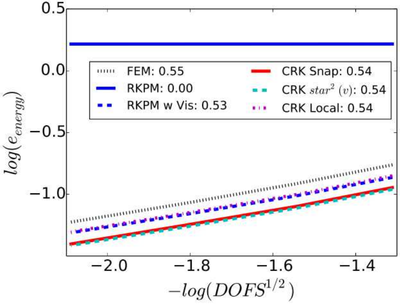
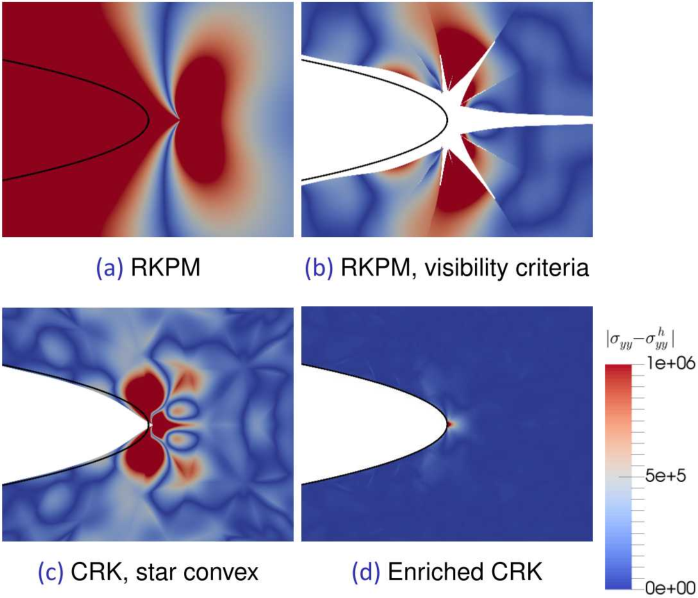


Figure: Convergence in energy


(a) RKPM

(b) RKPM, visibility criteria

(c) CRK, star convex

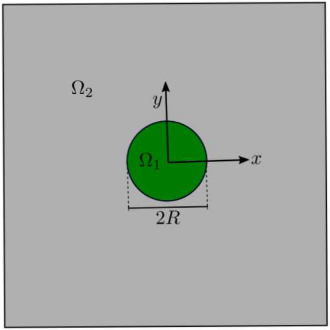
(d) Enriched CRK

Figure: Error in σ_{yy} near the crack tip. Nodal spacing $h_1 = 0.02$.

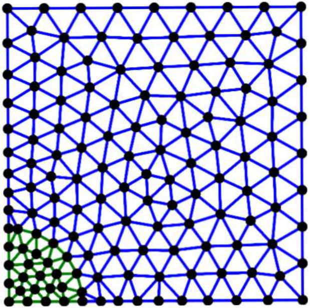
Panel with an Inclusion

An elastic panel with an inclusion

- (4x4) panel, $R = 1$ for inclusion
- Inclusion: $E = 10.E4$, $\nu = 0.3$
- Panel: $E = 10.E3$, $\nu = 0.3$



- Tension in x direction
- Exact displacement on symmetry planes
- Exact traction on other edges



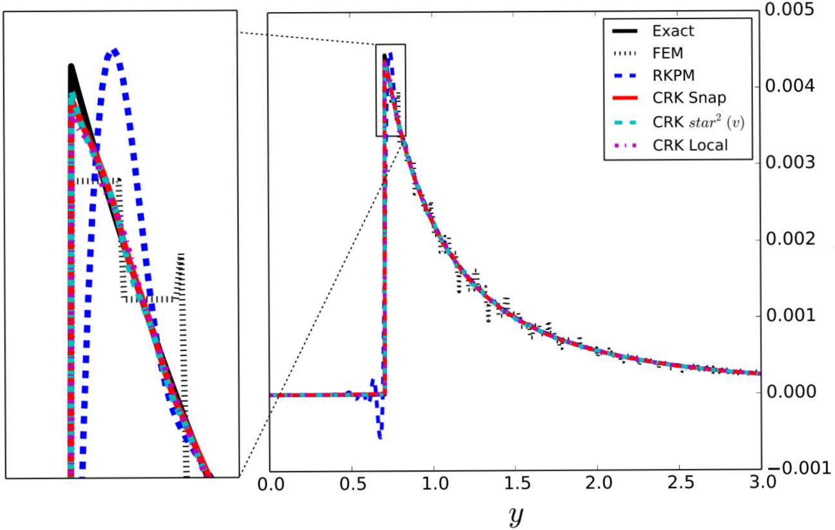


Figure: ε_{xy} near the material interface.

Results Comparison

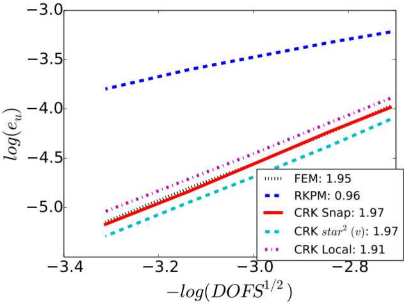


Figure: Convergence in $\mathbf{u}$

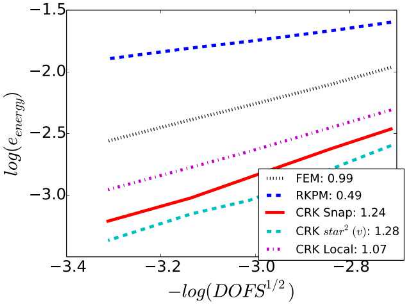


Figure: Convergence in energy

3D Implementation

3D Conforming Windows

The steps for creating conforming window in three dimensions directly translate from two dimensions

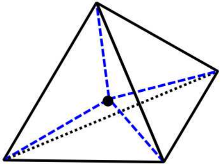
- **Subdivision:** triangulations → tetrahedral partitions
- **Function spaces:** Bernstein-Bézier spaces on triangulations → tetrahedral partitions
- **Specifying the coefficients**: More parameters for the extra dimension

However, complexity increases for smooth Bernstein-Bézier spaces in three dimensions. Evaluation of window values is more expensive.

Spaces on Tetrahedral Partitions

Extending $\mathcal{S}_5^{1,2}$ (the Argyris Space) to three dimensions leads to $\mathcal{S}_9^{1,4,2}(\triangle)$, where $1, 4, 2$ is the smoothness at the faces, vertices and edges, respectively.

The solution to decrease complexity has been to define spaces on *splits* of tetrahedral partition. For example, a C^1 space on the *Alfeld split*, $\mathcal{S}_5^1$



Alfeld Split

Required coefficients at different locations:

Space	Vertex	Face	Edge	Tetrahedron
$\mathcal{S}_9^{1,4,2}$	35	8	7	4
$\mathcal{S}_5^1$	10	2	3	1

(locations in un-split tetrahedral partition)

Simplification of Conforming Kernel Implementation

For conforming window functions, adding a constraint to the process greatly reduces the complexity.

Require integration points to be nodal locations (i.e. Hermite interpolation locations) of the window function space. In other words, triangulate the integration points.

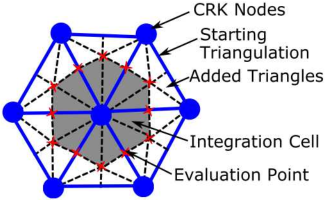
- Values are set at nodal locations, determination of coefficients and evaluation of Bernstein-Bézier function is not required.
- As such, *no construction of Bernstein-Bézier spaces required.* Values are explicitly set at the integration points, implied elsewhere.
- Pairs well with integration using smoothed gradients[5]

[5] J.-S. Chen, C.-T. Wu, S. Yoon, and Y. You, “A stabilized conforming nodal integration for Galerkin mesh-free methods,” *International Journal for Numerical Methods in Engineering*, vol. 50, no. 2, pp. 435–466, 2001.

Simplified Kernel Construction

Example: Stabilized Conforming Nodal Integration (SCNI) with boundary edge integration using trapezoid rule. Triangulations extracted from a global mesh.

- Input triangulation in blue
- A *Powell-Sabin split* (black-dashes) is used to create nodal integration domains (shaded region)
- The split also provides the window function triangulations
- A boundary integral is completed using values at nodal parameter locations (red marks) and provided the smoothed strain-displacement matrix $\widetilde{\mathbf{B}}$ or deformation gradient $\widetilde{\mathbf{F}}$, used in the SCNI method.



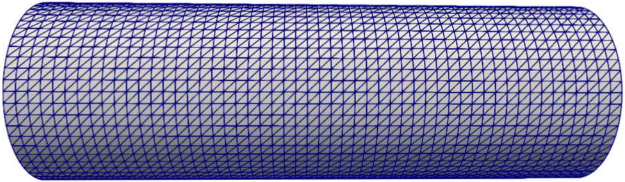
3D Examples

Taylor Bar Impact

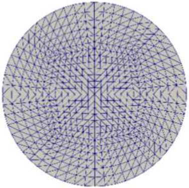
Dynamic problem, solved using explicit time integration (central difference).

- Height $= 2.346cm$
- Radius $= 0.391cm$

- Initial velocity $= 373m/s$
- J2 plasticity with power law hardening
- Axial displacement fixed on impact face

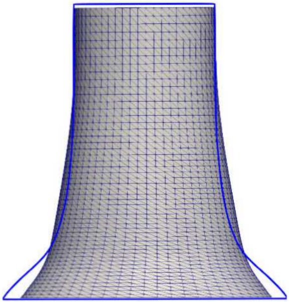


Side View

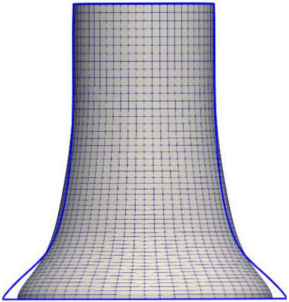


End View

3D Example: Taylor Bar Impact



CRK,
Tet Integration

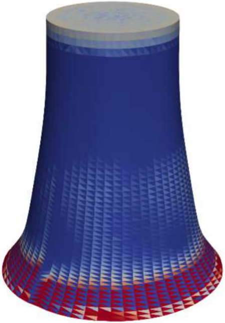


FEM Hex
Reduced Integration

- CRK Tet Integration: tet elements as gradient smoothing domains
- FEM Hex: reduced integration w/ hourglass control
- Blue Line: reference displacement outline (highly refined SD Hex)

Taylor Bar Results

The predictions have issues: volumetric locking, oscillations in hydrostatic stress



Hydrostatic Stress

Addressing Volumetric Locking[6]

[6] G. Moutsanidis, J. Koester, M. Tupek, Y. Bazilevs, and J.-S. Chen, "$\overline{\mathbf{B}}$ and $\overline{\mathbf{F}}$ approaches for the treatment of near-incompressibility in meshfree and immersed-particle methods," *Computational Particle Mechanics*, accepted.

Volumetric Locking in FEM and Meshfree Methods

As material approaches incompressiblity, approximations spaces have challenges meeting the constraint and providing a reasonable displacement field. This has been addressed in FEM in a few ways:

- Mixed methods with both pressure and displacement degrees-of-freedom (DOFS)
  - Should satisfy LBB condition
  - More expensive, complicated
- Reduced integration
  - Projects volumetric and deviatoric response to a lower order space
  - Efficient, requires stabilization, setting of parameters
- $\overline{\mathbf{B}}$ and $\overline{\mathbf{F}}$
  - Projects volumetric response to a lower order space, keeps deviatoric as-is
  - Improved coarse mesh accuracy, more expensive per DOF

Volumetric Locking in Meshfree Methods

The techniques from FEM have have been extended into meshfree methods. A popular approach is Stabilized Conforming Nodal Integration (SCNI):

- Comparable to reduced order elements
- Provides consistent integration for meshfree methods
- Efficient, has low energy modes requiring additional stabilization and setting of parameters

Goal: Develop a method that addressing volumetric locking and:

- Does not require setting stabilization parameters
- Is variationally consistent, efficient

Leverage ideas from SCNI and $\overline{\mathbf{B}}$ and $\overline{\mathbf{F}}$.

Review of and $\overline{\mathbf{F}}$

The $\overline{\mathbf{B}}$ method for small strains is extended to finite deformation with the $\overline{\mathbf{F}}$ method. The current position $\mathbf{x}$ of particle that was originally at location $\mathbf{X}$ is given by

$$\mathbf{x} = \phi(\mathbf{X}, t) \quad \text{with displacement} \quad \mathbf{u} = \phi(\mathbf{X}, t) - \mathbf{X} = \mathbf{x} - \mathbf{X}$$

The deformation gradient is defined as

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \frac{\partial \mathbf{X} + \mathbf{u}}{\partial \mathbf{X}} = \frac{\partial \mathbf{X}}{\partial \mathbf{X}} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \mathbf{I} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}}$$

A multiplicative split is used to decompose the deformation gradient

$$\mathbf{F} = \mathbf{F}^{dil}\,\mathbf{F}^{dev}$$

$$\text{with } \det \mathbf{F} = J = \det \mathbf{F}^{dil}, \quad \det \mathbf{F}^{dev} = 1, \quad \mathbf{F}^{dev} = J^{-1/3}\mathbf{F}, \quad \mathbf{F}^{dil} = J^{1/3}\mathbf{I}$$

The volumetric part of $\mathbf{F}$ is replaces with a projected value

$$\overline{\mathbf{F}} = \overline{\mathbf{F}}^{\,dil}\,\mathbf{F}^{dev},$$

where

$$\overline{\mathbf{F}}^{\,dil} = \pi(\mathbf{F}^{dil}) = \overline{J^{1/3}}\,\mathbf{I},$$

New Projection for $\overline{\mathbf{F}}$

New Projection for $\overline{\mathbf{F}}$

A $\overline{\mathbf{F}}$-based projection is designed to be compatible with the window simplification procedure. The projection leverages the *smoothed gradient* operator.

$$\widetilde{\nabla} f(\Omega_L) = \frac{1}{V_L} \int_{\Omega_L} \nabla f \, d\Omega,$$

$$V_L = \int_{\Omega_L} d\Omega,$$

where Ω_L is the smoothing volume surrounding corresponding to a material point L. Applying the divergence theorem leads to

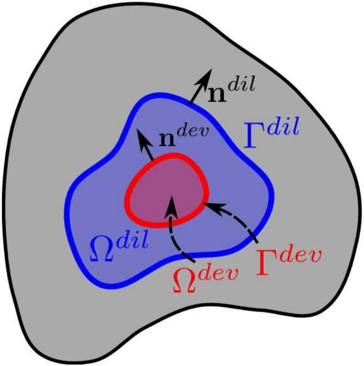
$$\widetilde{\nabla} f(\Omega_L) = \frac{1}{V_L} \int_{\Gamma_L} f \mathbf{n} \, d\Gamma,$$

$\mathbf{n}$ is the outward facing surface normal for boundary Γ_L of the smoothing volume. A smoothed deformation gradient $\mathbf{F}$ around material point L to give

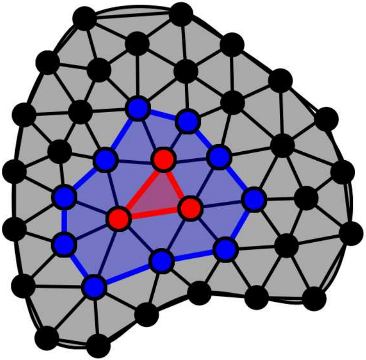
$$\widetilde{F_{ij}}(\Omega_L) = \frac{1}{V_L} \int_{\Omega_L} \left(\frac{\partial u_i^h}{\partial X_j} + \delta_{ij} \right) d\Omega = \frac{1}{V_L} \int_{\Omega_L} \left(\frac{\partial u_i^h}{\partial X_j} \right) d\Omega + \delta_{ij} = \frac{1}{V_L} \int_{\Gamma_L} u_i^h n_j \, d\Gamma + \delta_{ij}$$

New Projection for $\overline{\mathbf{F}}$

For the conforming window $\overline{\mathbf{F}}$ method two different domains are used for each material point, a smaller domain Ω_L^{dev} provides the deviatoric portion and a larger domain Ω_L^{dil} provides the dilatational portion.



Nested smoothing domains



Meshed domains

New Projection for $\overline{\mathbf{F}}$

For $\overline{\mathbf{F}}$ at material point L, we have

$$\overline{\mathbf{F}}_L = \overline{\mathbf{F}}_L^{dil}\,\mathbf{F}_L^{dev},$$

with the deviatoric part coming from the smoothed deformation gradient of the smaller cell,

$$\mathbf{F}_L^{dev} = \widetilde{\mathbf{F}}(\Omega_L^{dev}),$$

and the dilatational part coming from the smoothed deformation gradient of the larger cell,

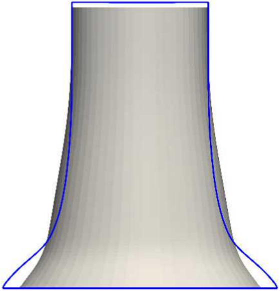
$$\overline{\mathbf{F}}_L^{dil} = \overline{J^{1/3}}(\Omega_L^{dil})\,\mathbf{I},$$

$$\overline{J}(\Omega_L^{dil}) = \frac{1}{V^{dil}} \sum_{c\in C} V_c J_c,\ V^{dil} = \sum_{c\in C} V_c,$$

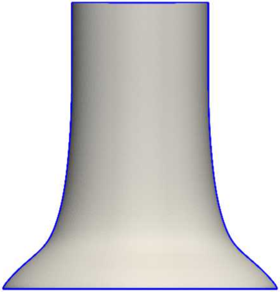
where C is the collection of cells/elements in the volumetric domain. This allows J to be compute one time for every cell then aggregated as needed for $\overline{J}$

Examples using $\overline{\mathbf{F}}$

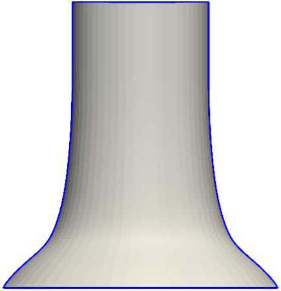
3D Example: Taylor Bar Impact



CRK,
Tet Integration



CRK,
SCNI

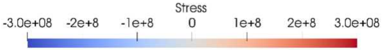


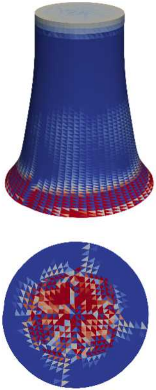
CRK,
$\overline{\mathbf{F}}$

- CRK Tet Integration: tet elements as gradient smoothing domains
- CRK $\overline{\mathbf{F}}$: tet elements for $\mathbf{F}^{dev}$, star of elements $\overline{\mathbf{F}}^{\,dil}$.
- Blue Line: reference displacement outline (highly refined SD Hex)

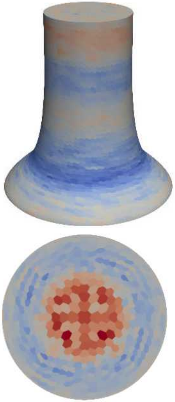
Taylor Bar, Axial Stress

Hex mesh converted to tets

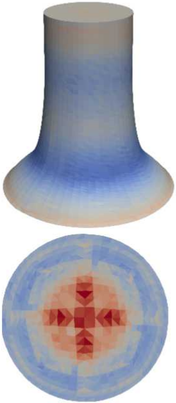




CRK, Tet Integration



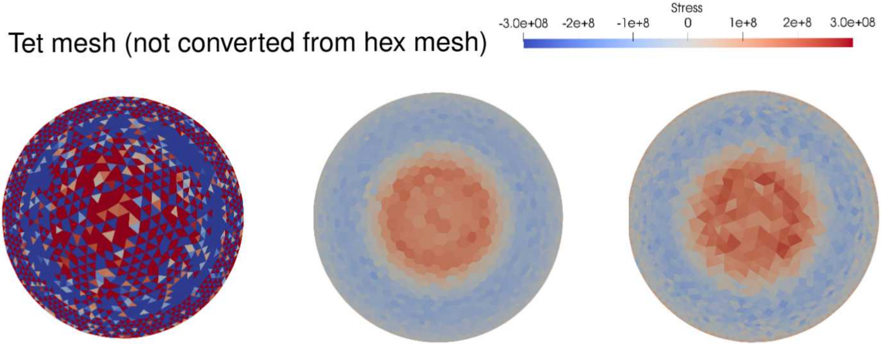
CRK, SCNI



CRK, $\overline{\mathbf{F}}$

Taylor Bar, Axial Stress

Tet mesh (not converted from hex mesh)



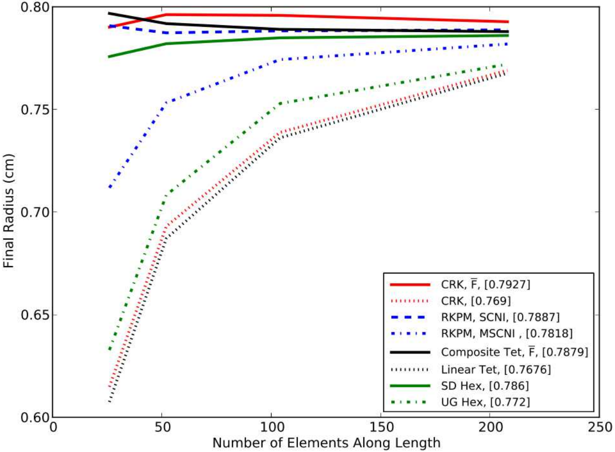
CRK, Tet Integration

CRK, SCNI

CRK, $\overline{\mathbf{F}}$

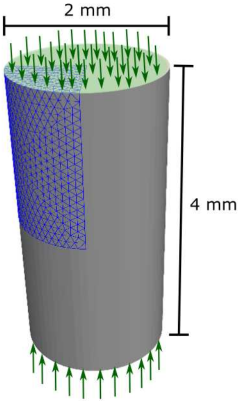
Taylor Bar Impact

Convergence of radial displacement with mesh refinement

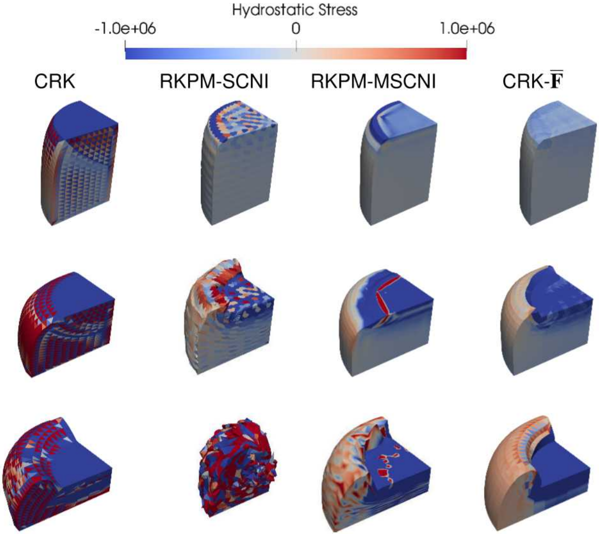


Compression of Elastomer Billet

- Compressed from both ends
- No in-plane deformation on the ends, free elsewhere
- Symmetric, 1/8 of domain simulated
- Gent Model: $K = 100$ MPa, $E = 1$ MPa $\rightarrow \nu = 0.4983$
- CRK: $star^2$ kernel
- $\overline{\mathbf{F}}$ uses each tetrahedron for $\mathbf{F}^{dev}$ and the star of elements around each tetrahedron for $\overline{\mathbf{F}}^{dil}$.
- Compared with Modified SCNI (MSCNI) with stabilization coefficient $\alpha = 0.05$.



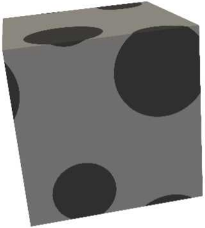
Compression of Elastomer Billet



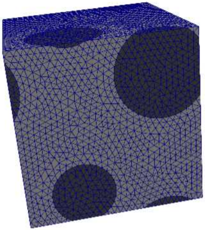
Carbon Black Rubber

- (29 x 29 x 29 nm) cube
- Shear displacement
- No in-plane deformation on the ends, free elsewhere
- Elastomer, Gent Model: $G = 950$ kPa, $K = 920$ MPa $\rightarrow \nu = 0.4995$
- Inclusions, Elastic: $E = 25$ GPa, $\nu = 0.25$

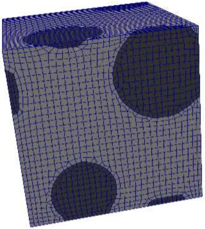
- CRK: $star^2$ kernel
- $\overline{\mathbf{F}}$ uses each tetrahedron for $\mathbf{F}^{dev}$ and the star of elements around each tetrahedron for $\overline{\mathbf{F}}^{dil}$.



Domain

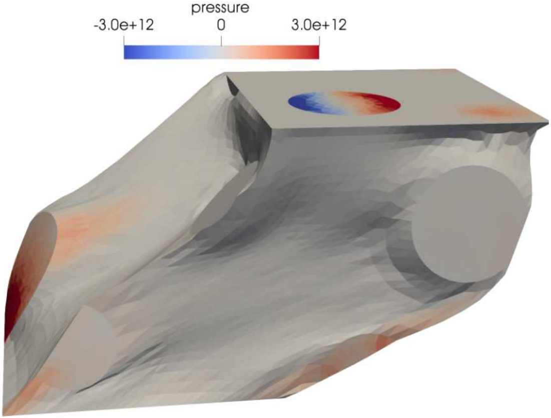


Example Tet Mesh

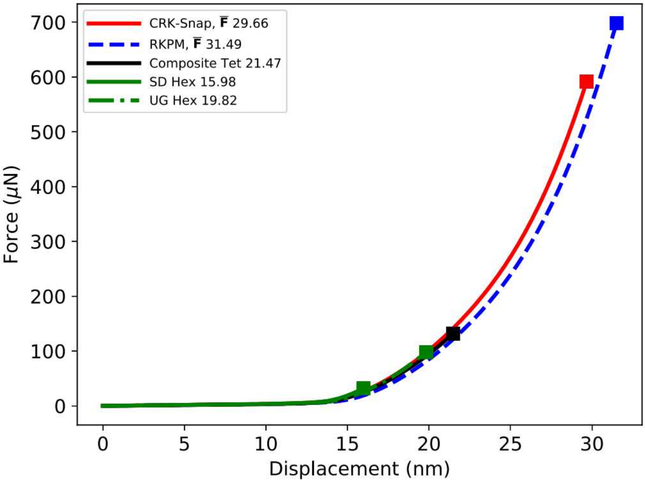


Example Hex Mesh

Carbon Black Rubber



Carbon Black Rubber



Working with Low Quality Triangulations

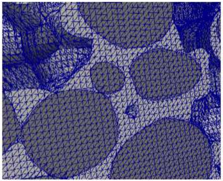
Low Quality Triangulations

Generating subdivisions of complex geometries is often faced with many challenges

- Small features need to be removed
- Scanned systems may be represented with STLs, requiring extra processing

Challenging to get a mesh of quality tetrahedra. However, many methods exist that readily produce meshes with low quality elements. E.g. Conformal Decomposition Finite Element Method (CDFEM)[7]

Lithium Ion Battery



CDFEM Mesh, Close-Up

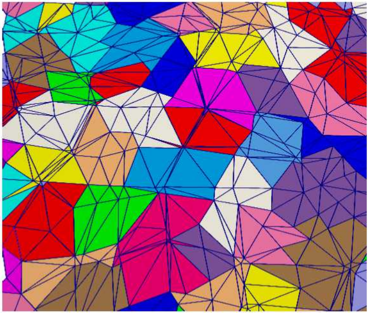
Goal: Develop and approach to use low quality triangulations but still have agreeable accuracy, robustness and efficiency.

[7] S. A. Roberts, H. Mendoza, V. E. Brunini, and D. R. Noble, "A verified conformal decomposition finite element method for implicit, many-material geometries," *Journal of Computational Physics*, vol. 375, pp. 352–367, 2018.

Addressing Low Quality Meshes

A method has been developed to work with low quality meshes. In short, use mesh only as a guide.

- **Decimate** by selecting a subset of vertices to be nodes carrying DOFs
- **Aggregate** elements into better shaped integration cells.



Example of clusters

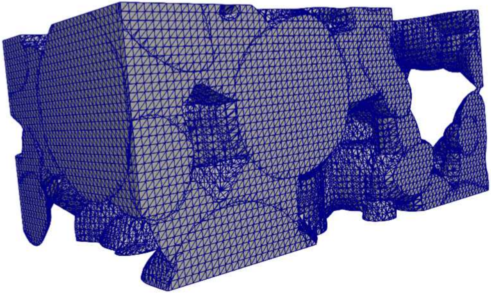
Addressing Low Quality Meshes

A k-means clustering process is used to accomplish the aggregation and decimation:

- Group elements into larger integration domains (polyhedra)
  - Added cluster volume limiters to balance cluster size
- Optionally split domains into subdomain and use with $\overline{\mathbf{F}}$ process
- Select one node per integration domain to carry degrees-of-freedom, the rest just provide structure for the integration domains

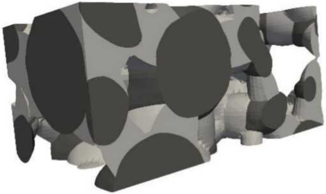
Lithium Ion Battery

CDFEM mesh with 821,437 elements and 173,917 nodes, aggregated to give 50,000 cells, each containing one node and two subcells. Loaded in compression.

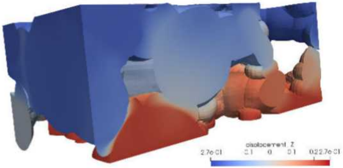


Initial Mesh

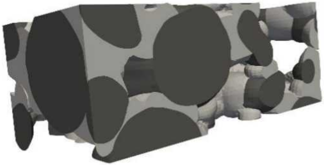
Lithium Ion Battery



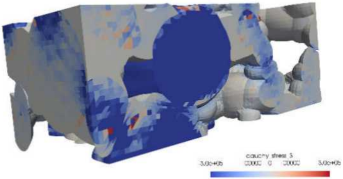
Initial State



Final Displacement



Final State



Final Stress

≈200x time step advantage over a linear tet. Ran to completion

Comments and Future Work

Agile Design-to-Simulation Framework

Goal: *Improving the Analyst's Response Time*

Approach: *Utilize the flexibility that meshfree methods provide, supply more control where needed*

- Conforming window functions to handle boundary / geometry challenges of meshfree methods.
- An $\overline{\mathbf{F}}$ method was developed to address volumetric locking.
- A decimation and aggregation procedure was created to address low quality triangulations.

Future Work:

- Better classification for element aggregation / pair with other meshing techniques
- Extend to handle fracture, very large deformation

Questions?