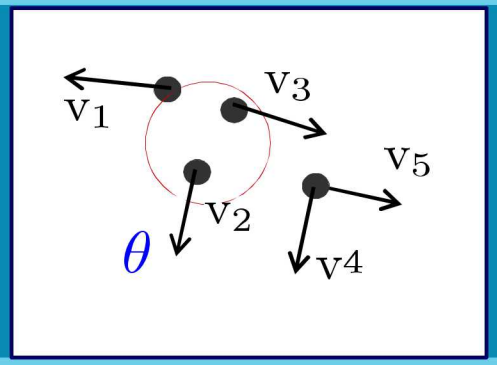
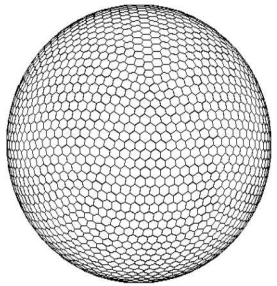
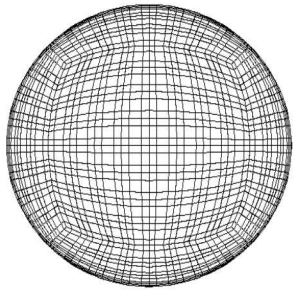
SAND2020-0297C

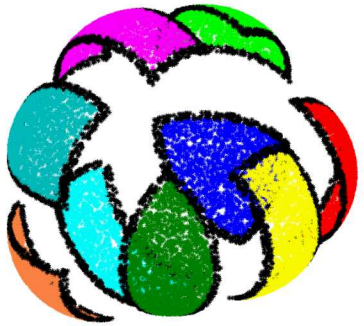
CCR Center for Computing Research

Field reconstruction from Raviart-Thomas, Nedelec, and finite volume degrees-of-freedom using Generalized Moving Least Squares









Paul Kuberry*, Mauro Perego, Nathaniel Trask, Pavel Bochev

Joint Mathematics Meetings 2020

AMS American Mathematical Society

MAA Mathematical Association of America

LDRD

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & 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.

SAND2020-XXXXC

Outline

- Native degrees-of-freedom
  - What they are and why/how they are used
- Generalized Moving Least Squares (GMLS)
  - Description and examples
  - Extension to native fields
  - Numerical examples
- The Compadre Toolkit

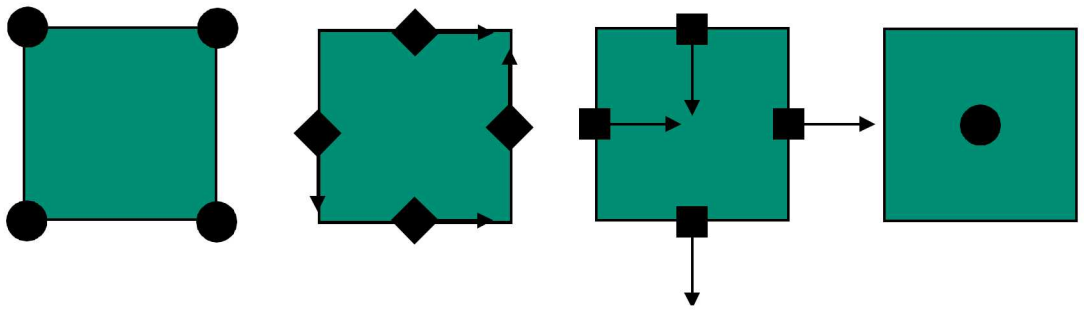
What are Native Degrees-of-Freedom (DoF)?

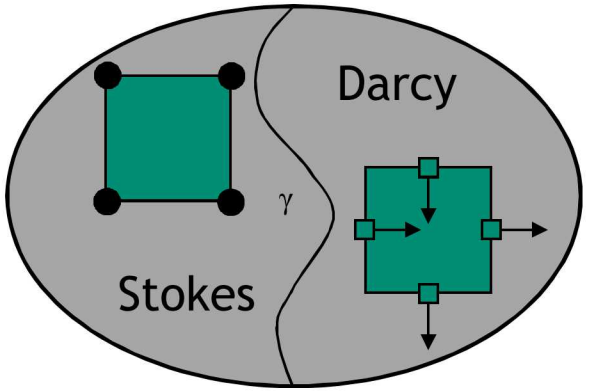
Typical use cases

- Different codes may employ **different discretizations** of the same PDE due to different designs, e.g.,
  - stabilized vs. compatible
- The same field may be **represented differently** in a coupled multi-physics simulation, e.g.,
  - Raviart-Thomas (H(div)) velocity vs. nodal (H1) velocity in Darcy-Stokes coupling
- The field may be represented by the same type of discretization but on a **different cell shape**:
  - Raviart-Thomas on tets, Raviart-Thomas on hexes and mimetic difference on polyhedrons

We consider a DoF type to be "native" to a code that uses it to represent its fields (it is a relative term). This DoF type is non-native to any code that couples to the given code.

Compatible discretization spaces





Research Problem

- Code coupling requires exchange of fields that may have different native DoF sets.
- The coupler should be able to reconstruct the necessary fields from their native DoFs.
- One option is to tell the coupler how each code reconstructs the fields internally, i.e., expose FE bases, quadrature, etc. to the coupler.
- This may be impractical as codes may have multiple options to reconstruct from their native DoFs and/or these options may change over the product's lifecycle.

Research challenge:

Can we provide a field reconstruction capability that

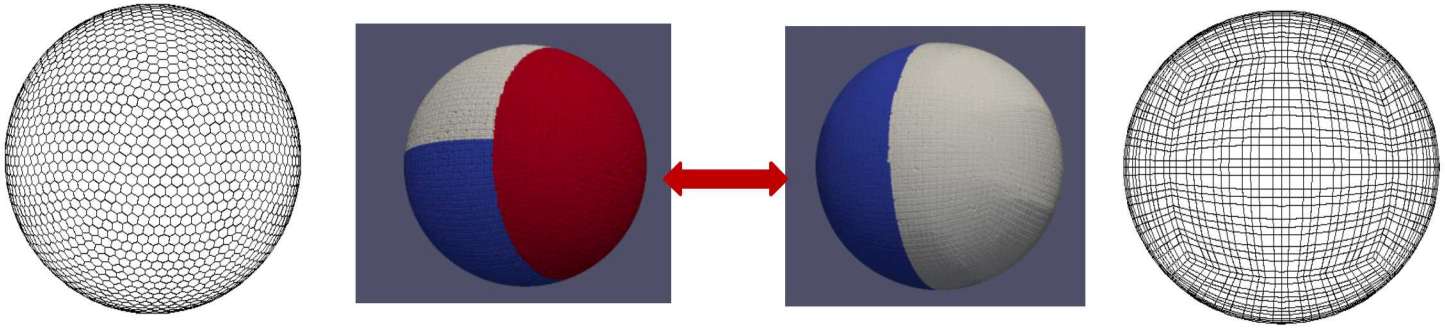
- Accepts any reasonable native DoF type
- Reconstructs accurately a field from that type without having to know the internal reconstruction procedure for each code utilizing this DoF type

Our solution:

Extend Generalized Moving Least Squares beyond point samples to enable reconstruction from all basic types of DoFs in compatible discretizations

Climate Simulation Needs

- E3SM (Energy Exascale Earth System Model, previously ACME)
  - Simulates the fully coupled Earth system at high-resolution (15-25 km, with additional regional refinement where necessary)
  - Subsystem models include land, **ocean**, atmosphere, land ice, and sea ice
  - Discretization "zoo" for the various subsystem models, requiring a high-order accurate and flexible data transfer capability
  - Heterogeneous architectures benefit from improved computation to communication ratio (involves solution of many small but independent QPs)
  - Focus today is on transfer, recovery, or remap of primary field from Face (Raviart-Thomas) basis field data and Edge (Nedelec) basis field data



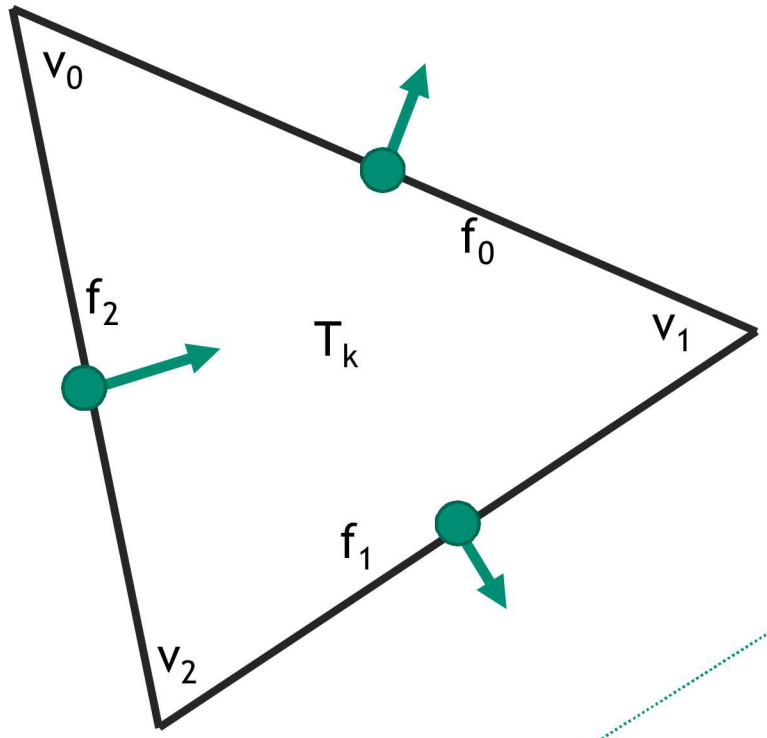
MPAS HOMME

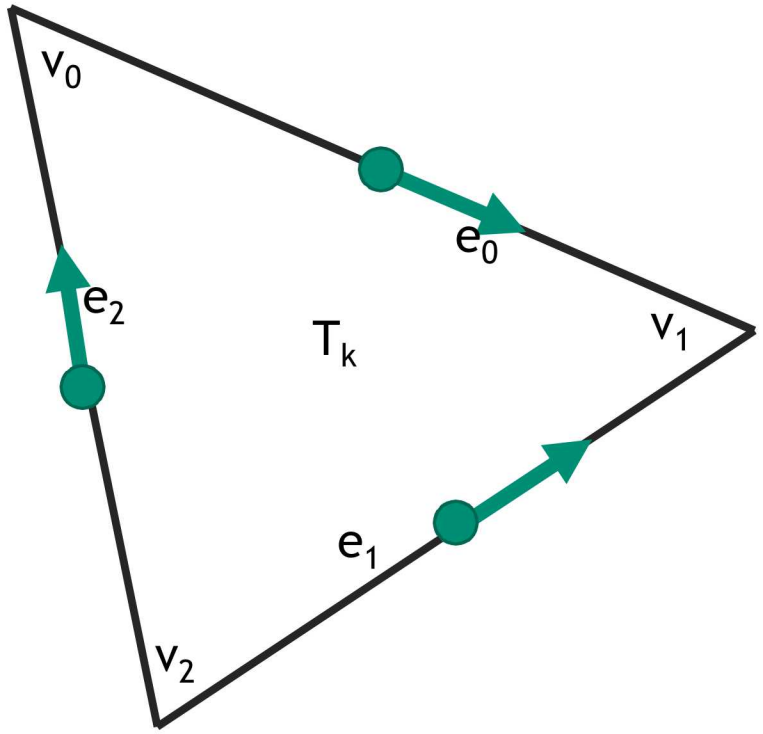
Coupling Approaches for Next Generation Architectures (CANGA)

Examples of Native Field Data

Raviart-Thomas (Face) elements

For low order Raviart-Thomas, data is contained at midpoints of faces $\{f_i\}$ and represents either $u(x_i) \cdot n(x_i)$ or $\int_{f_i} u(x) \cdot n(x)\, df_i$.





Nedelec (Edge) elements

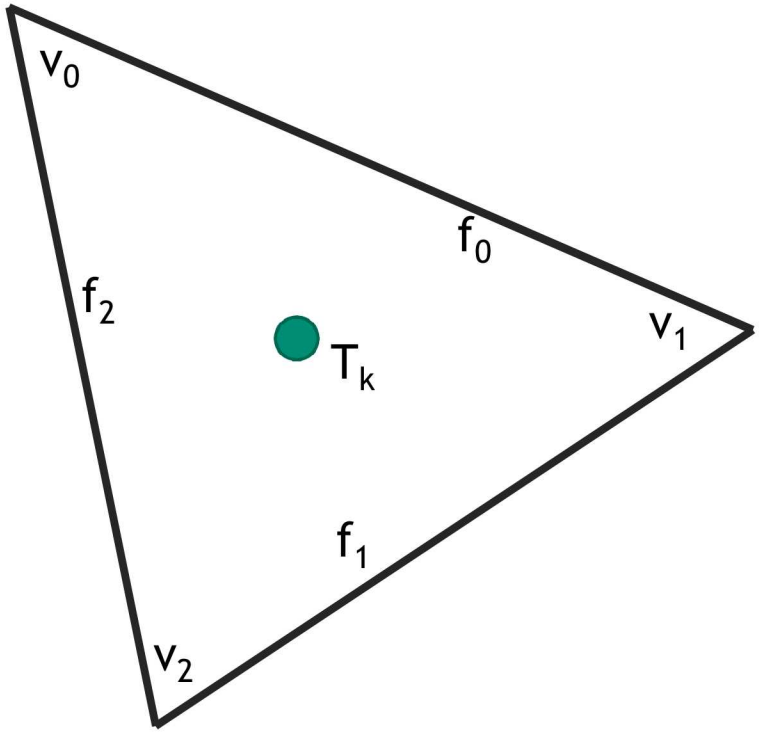
For low order Nedelec, data is contained at midpoints of edges $\{e_i\}$ and represents either $u(x_i) \cdot t(x_i)$ or $\int_{e_i} u(x) \cdot t(x)\, de_i$.

Examples of Native Field Data

Cell-averaged (FV) degrees of freedom

For finite volume methods, data is stored as one value per cell which represents the average of the solution over a cell.

$$\frac{\int_{T_k} u(x)\, dx}{measure(T_k)}$$



Generalized Moving Least Squares

Ingredients:

V, V^* - a function space (e.g. continuous functions) and its dual

$P = span\{p_i\}_{i=1}^{Q} \subset V$ - an approximation finite dimensional space, e.g. polynomials

$\Lambda = \{\lambda_1, \ldots, \lambda_N\} \subset V^*$ - a finite set of sampling functionals (e.g. point evaluations, integrals, normal components, tangent components, etc…)

$\tau \in \mathcal{T} \subset V^*$ - a target functional (or a family of target functionals)

$W(\tau, \lambda_i) : (\mathcal{T} \cup \Lambda) \times (\mathcal{T} \cup \Lambda) \to \mathbb{R}$ - a window function correlating functionals (e.g. a radial kernel) determines the smoothness of reconstruction

Example, MLS case:

Point cloud $X_h = \{\mathbf{x}_i\}_{i=1}^{N_h} \subset \Omega$, with filling distance $h = \sup_{\mathbf{x}\in\Omega} \min_{i=1,\ldots,N_h} |\mathbf{x} - \mathbf{x}_i|$.

$u \in V = C^{k+1}(\Omega), \quad P = \Pi^k(\Omega), \quad \lambda_i(u) = u(\mathbf{x}_i), \quad \tau_{\mathbf{x}} = u(\mathbf{x}), \quad W = W(|\mathbf{x} - \mathbf{x}_i|)$

Generalized Moving Least Squares

1. (GMLS Approximate) Constrained Optimization formulation:

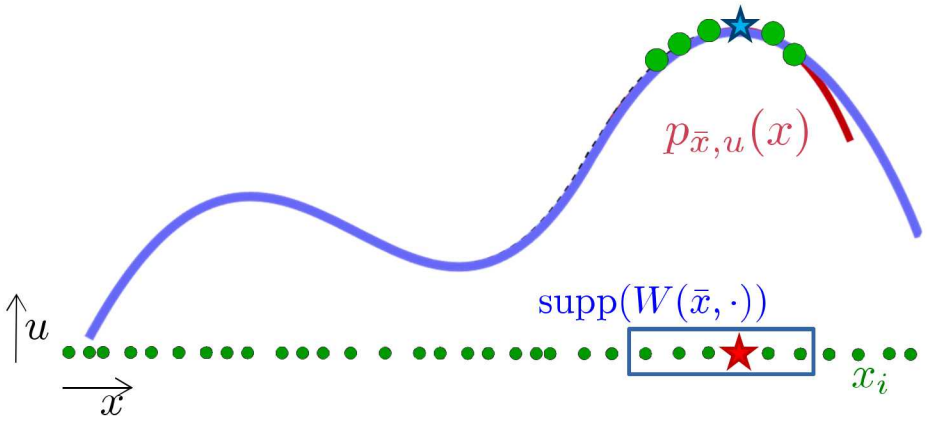
$$\tau^h_{\bar{\mathbf{x}}}(u) := \sum_{i \in I_{\bar{\mathbf{x}}}} a^i_{\tau_{\bar{\mathbf{x}}}} \lambda_i(u), \quad \{a^i_{\tau_{\mathbf{x}}}\} = \arg\min_{a^i} \sum_{i \in I_{\bar{\mathbf{x}}}} \frac{|a^i|^2}{W(\bar{\mathbf{x}}, \mathbf{x}_i)},$$

$$\text{s. t. } \tau_{\bar{\mathbf{x}}}(p) = \sum_{i \in I_{\bar{\mathbf{x}}}} a^i \lambda_i(p), \ \forall p \in P \qquad \left(\tau^h_{\bar{\mathbf{x}}}(p) = \tau_{\bar{\mathbf{x}}}(p)\right)$$

2. (Practical Recipe) Least Squares formulation:

$$\tau^h_{\bar{\mathbf{x}}}(u) := \tau_{\bar{\mathbf{x}}}(p_{\bar{\mathbf{x}},u}), \quad p_{\bar{\mathbf{x}},u} = \arg\min_{p \in P} \sum_{i \in I_{\bar{\mathbf{x}}}} \left(\lambda_i(u) - \lambda_i(p)\right)^2 W(\bar{\mathbf{x}}, \mathbf{x}_i)$$

$\mathbf{I}_{\bar{\mathbf{x}}} := \{i : W(\bar{\mathbf{x}}, \mathbf{x}_i) > 0\}$

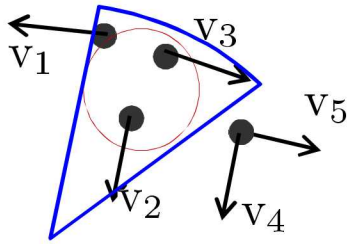


Reconstructing Vector Functions (from projections along given directions)

Associate to each particle i a position $\mathbf{x}_i$ and a unit vector $\mathbf{v}_i \in \mathbb{R}^d$.

Sampling functional: $\lambda_i(\mathbf{u}) := \mathbf{u}(\mathbf{x}_i) \cdot \mathbf{v}_i$

Filling distance h_ω is the radius of the smallest ball that centered at any point of Ω contains d particles whose versors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ contain a basis for $\mathbb{R}^d$ with associated determinant bigger than ω ($|\det[\mathbf{v}_1, \ldots, \mathbf{v}_d]| \geq w$).



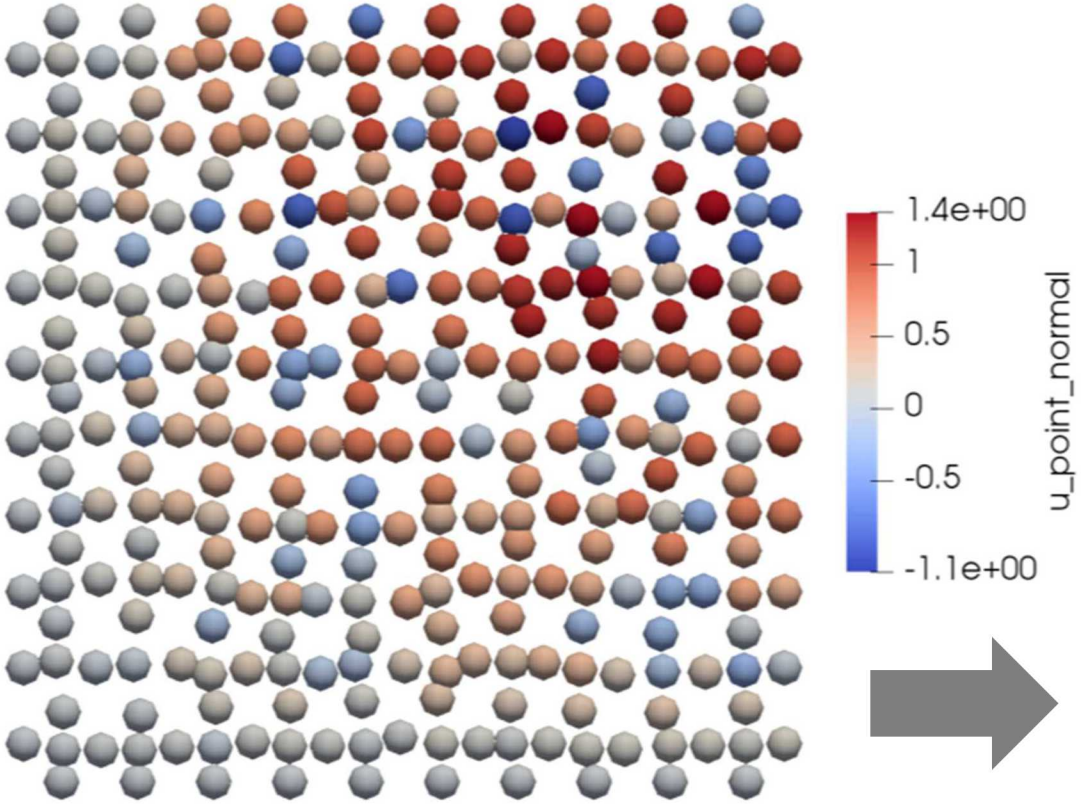
For $h_\omega \leq C(\theta, R, m, w)$, $\forall p \in [\Pi^m]^d$,
$\exists\, \lambda_{i_p} \in \Lambda_h$, such that $|\lambda_{i_p}(p)| \geq \rho_\omega \|p\|_{\Omega,\infty}$

Other sampling functionals we considered:

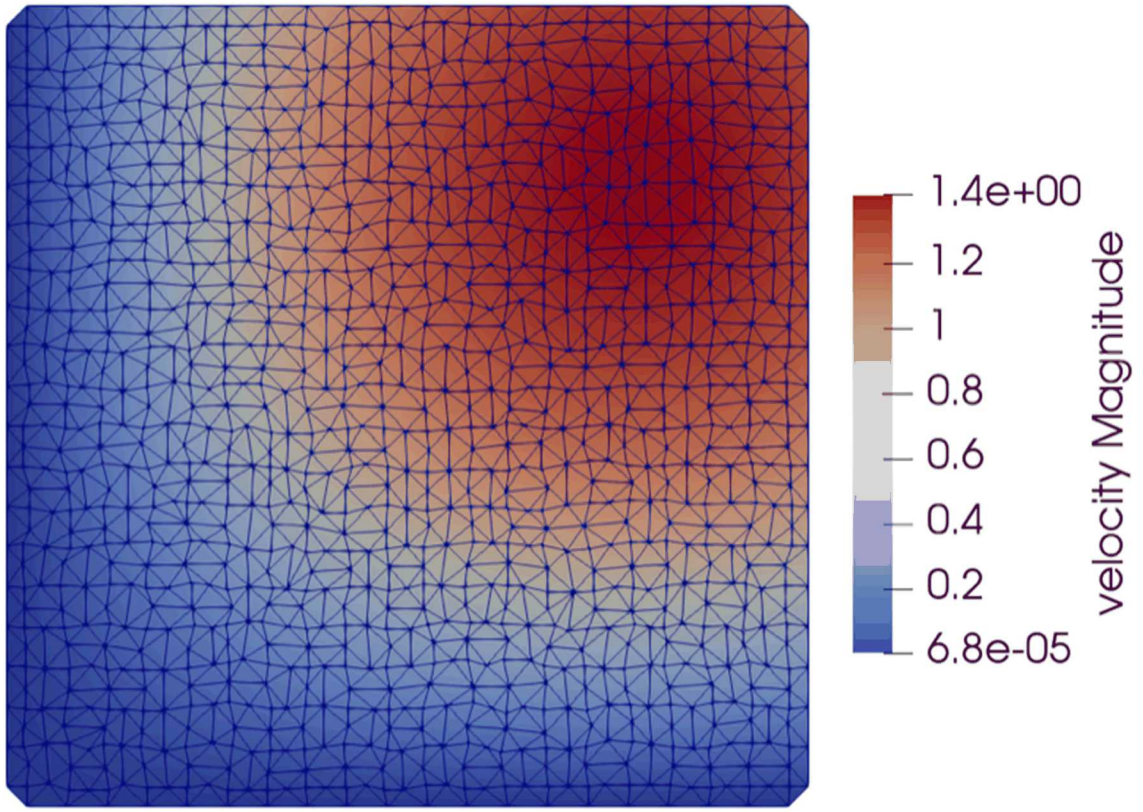
$$\lambda_i^e(\mathbf{u}) := \frac{1}{|e_i|} \int_{e_i} \mathbf{u} \cdot \mathbf{t}_i \qquad \lambda_i^f(\mathbf{u}) = \frac{1}{|f_i|} \int_{f_i} \mathbf{u} \cdot \mathbf{n}_i \qquad \lambda_i^v(u) := \frac{1}{|V_i|} \int_{V_i} u(\boldsymbol{y})\, d\boldsymbol{y}$$

Data Transfer from Face Elements (RT) to P1

Face normal (Raviart-Thomas) field data



P1 Lagrange field data



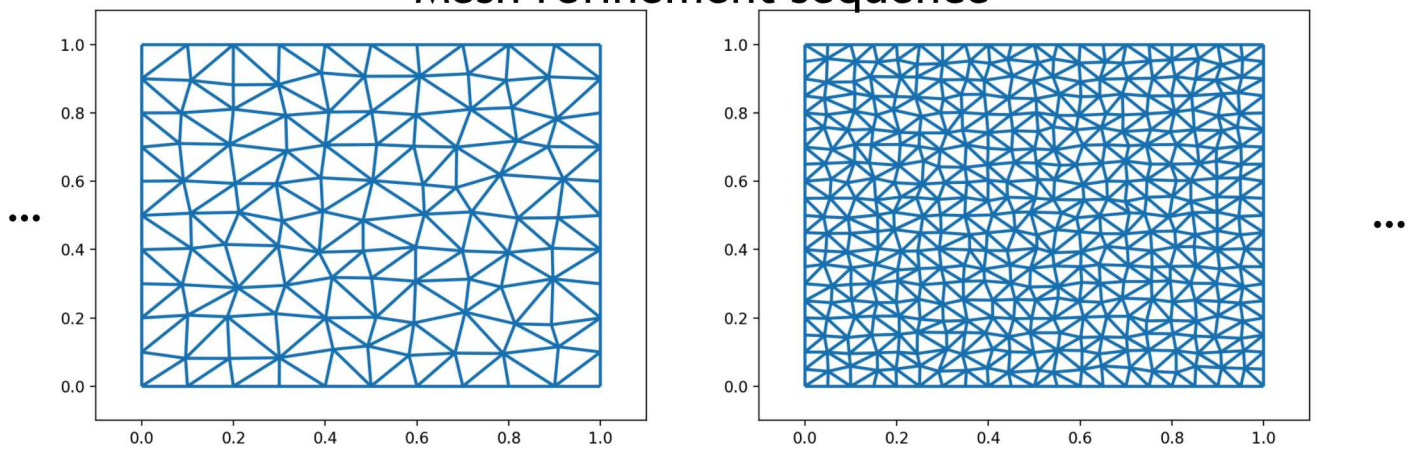
Convergence Study I (Quasi-uniform mesh)

Exact solution:

$$\vec{v}(x,y) = \begin{pmatrix} \sin(x)\sin(y) \\ -\sin(x)\sin(y) \end{pmatrix}$$

Degree of basis for reconstruction: 4

Mesh refinement sequence



	Regular Mesh							
	Face Elements (Raviart-Thomas)				Edge Elements (Nedelec)			
	Integral		Point		Integral		Point	
h	l2 Error	Rate	l2 Error	Rate	l2 Error	Rate	l2 Error	Rate
0.02	6.64559E-05	-	7.28634E-05	-	4.94356E-05	-	6.77058E-05	-
0.01	4.78883E-06	3.79	4.58658E-06	3.99	2.34348E-06	4.40	2.5238E-06	4.75
0.005	8.4105E-08	5.83	8.22289E-08	5.80	7.40259E-08	4.98	6.62357E-08	5.25
0.0025	2.10075E-09	5.32	1.95143E-09	5.40	2.03705E-09	5.18	1.71913E-09	5.27
0.00125	5.11811E-11	5.36	4.7762E-11	5.35	4.57059E-11	5.48	4.04744E-11	5.41

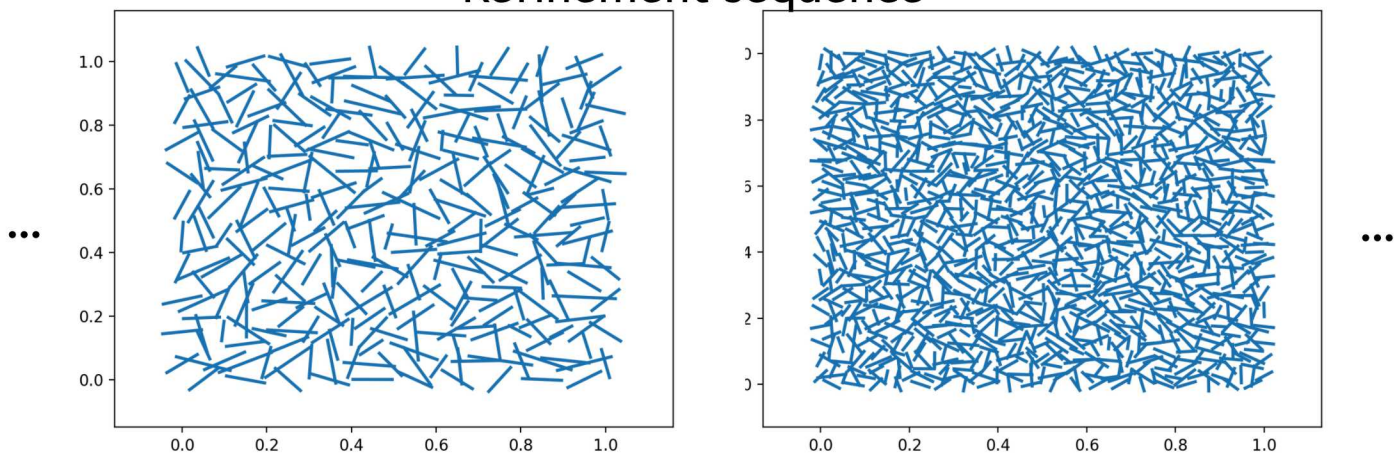
Convergence Study II (Randomly rotated edges)

Exact solution:

$$\vec{v}(x,y) = \begin{pmatrix} \sin(x)\sin(y) \\ -\sin(x)\sin(y) \end{pmatrix}$$

Degree of basis for reconstruction: 4

Refinement sequence



	Randomly Rotated Edges							
	Face Elements (Raviart-Thomas)				Edge Elements (Nedelec)			
	Integral		Point		Integral		Point	
h	l2 Error	Rate	l2 Error	Rate	l2 Error	Rate	l2 Error	Rate
0.02	4.63995E-05	-	4.86964E-05	-	5.27721E-05	-	5.56072E-05	-
0.01	1.15421E-06	5.33	1.31781E-06	5.21	1.02272E-06	5.69	9.83186E-07	5.82
0.005	3.14008E-08	5.20	2.78324E-08	5.57	3.11048E-08	5.04	3.2778E-08	4.91
0.0025	7.28335E-10	5.43	6.65194E-10	5.39	6.69842E-10	5.54	6.17752E-10	5.73
0.00125	2.0614E-11	5.14	1.81714E-11	5.19	2.02698E-11	5.05	1.82303E-11	5.08

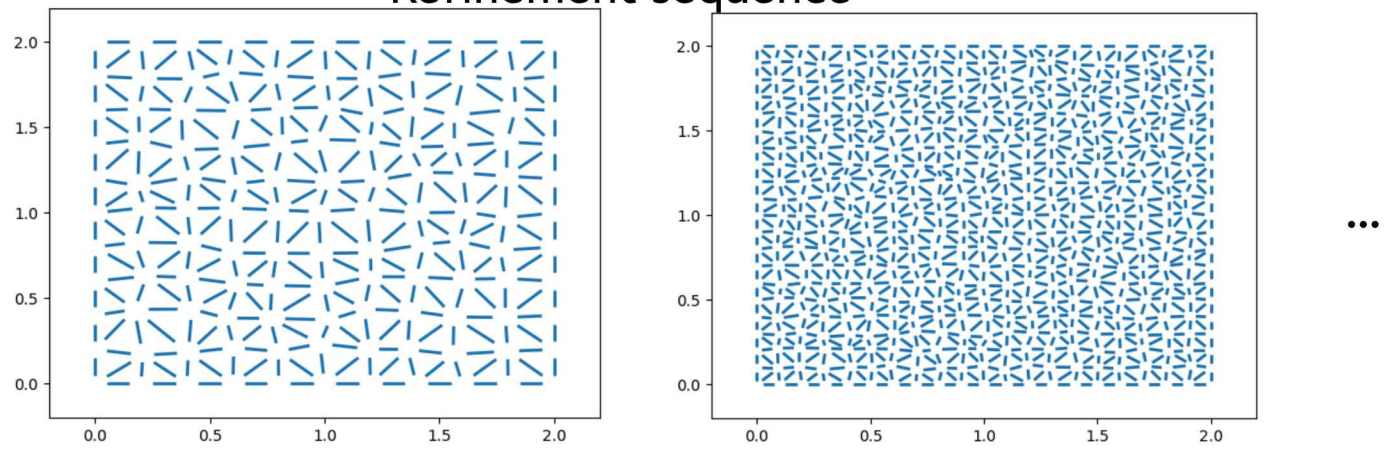
Convergence Study III (Blown-up edges from regular mesh)

Exact solution:

$$\vec{v}(x,y) = \begin{pmatrix} \sin(x)\sin(y) \\ -\sin(x)\sin(y) \end{pmatrix}$$

Degree of basis for reconstruction: 4

Refinement sequence



	Dilated from Mesh							
	Face Elements (Raviart-Thomas)				Edge Elements (Nedelec)			
	Integral		Point		Integral		Point	
h	l2 Error	Rate	l2 Error	Rate	l2 Error	Rate	l2 Error	Rate
0.02	0.0018622	-	0.0019948	-	0.0025625	-	0.0026576	-
0.01	0.0001834	3.34	0.0001686	3.56	0.0001177	4.44	0.0001146	4.53
0.005	2.714E-06	6.08	2.587E-06	6.03	3.209E-06	5.20	2.988E-06	5.26
0.0025	6.61E-08	5.36	6.238E-08	5.37	8.03E-08	5.32	7.442E-08	5.33
0.00125	1.577E-09	5.39	1.463E-09	5.41	1.933E-09	5.38	1.775E-09	5.39

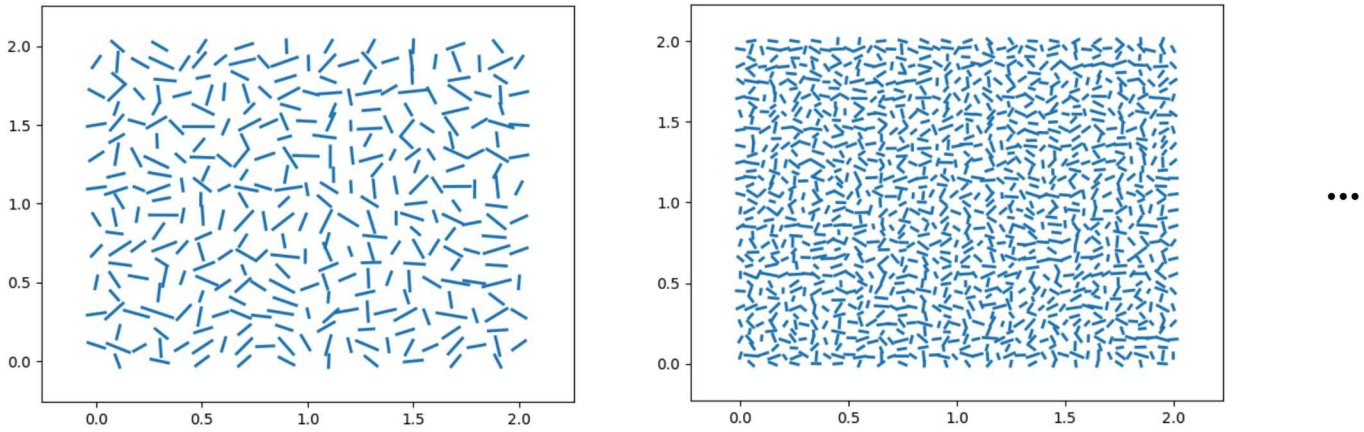
Convergence Study IV (Blown-up randomly rotated edges)

Exact solution:

$$\vec{v}(x,y) = \begin{pmatrix} \sin(x)\sin(y) \\ -\sin(x)\sin(y) \end{pmatrix}$$

Degree of basis for reconstruction: 4

Refinement sequence



	Randomly Rotated Edges from Dilated Mesh							
	Face Elements (Raviart-Thomas)				Edge Elements (Nedelec)			
	Integral		Point		Integral		Point	
h	l2 Error	Rate	l2 Error	Rate	l2 Error	Rate	l2 Error	Rate
0.02	0.0016478	-	0.0017719	-	0.0016324	-	0.0016149	-
0.01	5.777E-05	4.83	5.941E-05	4.90	3.549E-05	5.52	3.466E-05	5.54
0.005	1.075E-06	5.75	1.039E-06	5.84	1.03E-06	5.11	1.045E-06	5.05
0.0025	2.647E-08	5.34	2.583E-08	5.33	2.458E-08	5.39	2.388E-08	5.45
0.00125	7.168E-10	5.21	6.746E-10	5.26	7.279E-10	5.08	6.875E-10	5.12

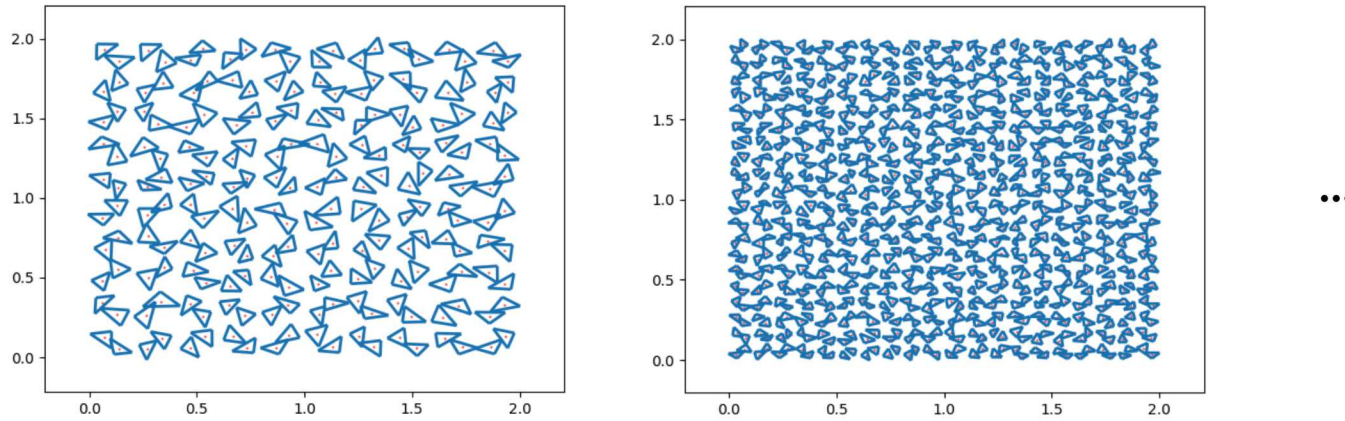
Convergence Study V (Various configurations with Cell-Averaged)

Exact solution:

$$v(x,y) = cell\ average(\sin(x)\sin(y))$$

Degree of basis for reconstruction: 4

Refinement sequence

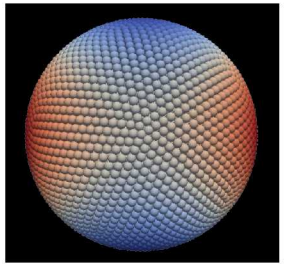


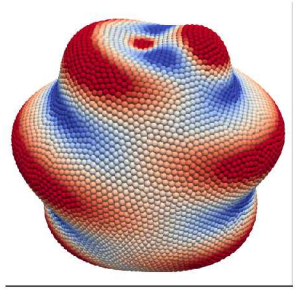
	Cell-Averaged (FV)							
	Quasi-Uniform Mesh		Randomly Rotated Cells		Blown-Up Cells		Rotated+Blown-Up Cells	
h	l2 Error	Rate	l2 Error	Rate	l2 Error	Rate	l2 Error	Rate
0.02	7.668E-08	-	6.870E-08	-	1.080E-06	-	1.032E-06	-
0.01	1.294E-09	5.89	1.259E-09	5.77	2.317E-08	5.54	2.391E-08	5.43
0.005	2.435E-11	5.73	2.613E-11	5.59	5.436E-10	5.41	5.493E-10	5.44
0.0025	6.001E-13	5.34	5.947E-13	5.46	1.360E-11	5.32	1.336E-11	5.36
0.00125	1.630E-14	5.20	1.550E-14	5.26	3.604E-13	5.24	3.491E-13	5.26

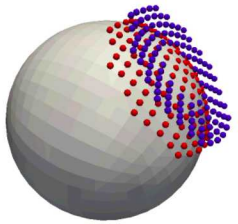
The Compadre Toolkit

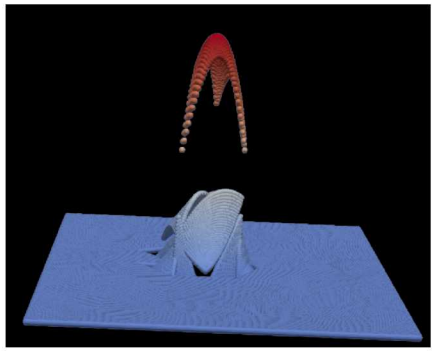
Remap

- Remote remap of fields between codes with independent data distributions (domain decomposition, halo search, neighbor search)
- Remap for targets not limited to point reconstruction (gradient, curl, divergence, integrals, etc.., along with surface equivalents on manifolds)
- **Remap from samples that are point evaluations, integrals over cells/edges/balls, and native fields such as face normal velocities**
- Sets up, solves, and applies solutions to GMLS problems
- Utilities for generating tangent space approximations for problems posed on manifolds









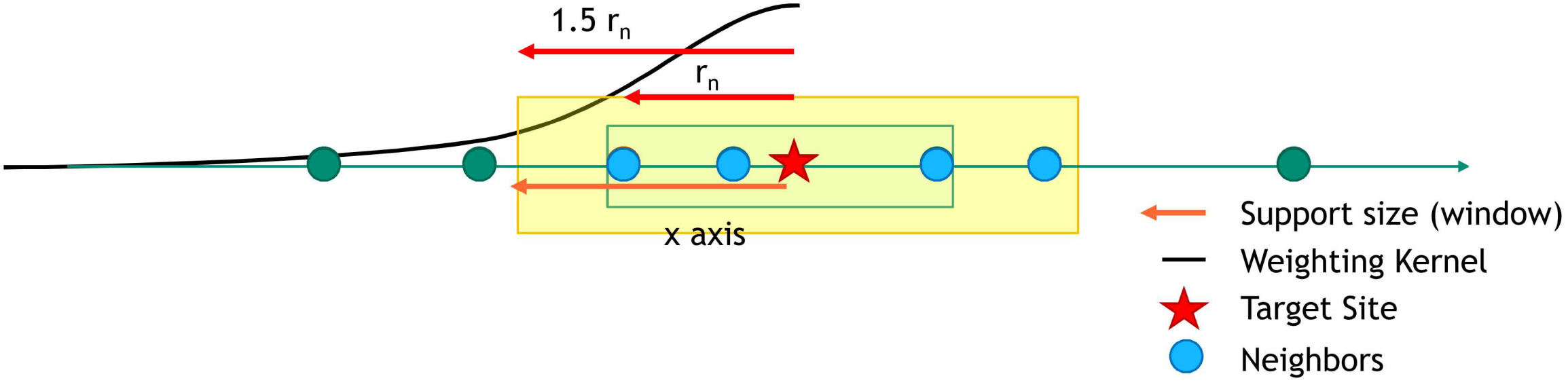
PDE Solution

- Same GMLS problem setup / solution application can be used as a **stencil for degrees of freedom (DoFs)** to form a global linear system.
- **Strong-form PDE solution** support for assembly, boundary conditions, and source terms (Laplacian, Laplace-Beltrami, Stokes, elastodynamics, shallow water, etc…)
- **Weak-form PDE solution** (Non-conforming polynomial shape functions)
- **Block physics** with block preconditioning (GMRES, CG, AMG, ILU, Jacobi, G-S, etc… using Trilinos solver packages)

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} * \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

Under the Hood [Adaptive neighbor search in 1D]

1. Determine number of neighbors needed for unisolvency given degree of basis used for reconstruction
In 1D, with p = 2 is just n = p+1 = **3**
2. Do KNN (k-nearest neighbors) search with a k-d tree for 3 neighbors
3. Identify n^{th} neighbor, and its distance r_n
4. Multiply r_n by some multiplier (~1.5) to ensure non-zero weighting kernel value for n^{th} neighbor
5. Do radius search and keep all neighbors within distance $1.5r_n$



Compadre Toolkit Info

C++ Project

SCAN ME

https://github.com/SNLComputation/compadre

Python Module
(builds C++ project and provides Python interface)

SCAN ME

https://pypi.org/project/compadre/

pip install compadre

https://doi.org/10.5281/zenodo.2557929

Conclusions

- Native degrees-of-freedom can include complex representations of fields that are **useful** due to benefits in regularity and/or compatibility, **but** coupling from one representation to another can be **challenging**.
- Traditional interpolation schemes can not interpolate between many field representations.
- GMLS is a powerful tool for native fields due to the **flexibility in choice of sampling functional**, allowing remap/reconstruction of commonly encountered H(grad), H(div), and H(curl) bases, as well as cell averaged representations of data.
- **Compadre Toolkit** provides a massively parallel, **performance portable** solution for setting up and solving GMLS problems. It runs on GPUs and also has a Python interface.

References

- H. Wendland, *Scattered Data Approximation*, Cambridge University Press, Cambridge, UK, 2004.
- Douglas N Arnold et al. Compatible spatial discretizations. Vol. 142. Springer Science & Business Media, 2007.
- Pavel B. Bochev and James M. Hyman. “Principles of Mimetic Discretizations of Differential Operators”. In: Compatible Spatial Discretizations. Ed. by Douglas N. Arnold et al. New York, NY: Springer New York, 2006, pp. 89–119. isbn: 978-0-387-38034-6.
- D. Mirzaei, R. Schaback, and M. Dehghan. “On generalized moving least squares and diffuse derivatives”. In: IMA Journal of Numerical Analysis 32.3 (2012), pp. 983–1000.
- P. Kuberry, P. Bosler, and N. Trask. Compadre Toolkit. Feb. 2019. doi: 10.5281/zenodo.2560287. url: https://doi.org/10.5281/zenodo.2560287. Results from SHA:59333da7b4f in *harness* branch.