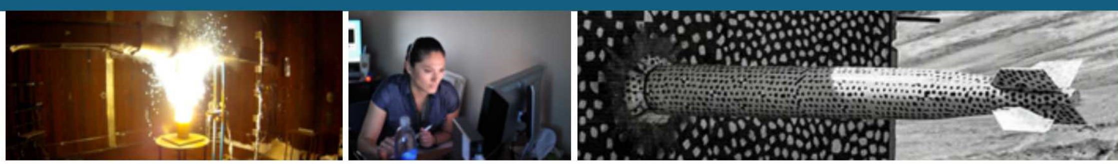
SAND2019-9681C

CENTER FOR COMPUTING RESEARCH

Sandia National Laboratories

SAND2019-XXXX C

The amazing powers of Generalized Moving Least Squares



4th International workshop on Minimum Residual Numerical Methods
Humboldt University, Berlin. September 16-18, 2009

PRESENTED BY

Pavel Bochev

P.Bochev, P. Kuberry, M. Perego, N. Trask

LDRD LABORATORY DIRECTED RESEARCH & DEVELOPMENT

U.S. DEPARTMENT OF ENERGY | Office of Science

ASCR

U.S. DEPARTMENT OF ENERGY NNSA

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.

Outline

SAND2019-XXXX C

- What is GMLS?
- Applications:
  - #1: mesh-“hardened” DG scheme
  - #2: meshfree mimetic divergence
- Conclusions

SAND2019-XXXX C

A GMLS tutorial

Generalized Moving Least Squares (GMLS) is a *non-parametric regression* for **dual spaces**

Statement of the GMLS problem: Given

V, V^* - a function space and its dual

$P = span\{p_i\}_{i=1}^{Q} \subset V$ - a finite dimensional "consistency" space (usually polynomials)

$\Lambda = \{\lambda_1, \ldots, \lambda_N\} \subset V^*$ - a finite set of sampling functionals:

$\omega : V^* \times V^* \to \mathbf{R}$ - a correlation measure between functionals

For every $\tau \in V^*$ **(target) find an approximation** $\overline{\tau} \in V^*$ **given by**

$$\overline{\tau}(u) = \sum_{i=1}^{N} a_i(\tau)\lambda_i(u) \text{ such that } \begin{cases} \overline{\tau}(p) = \tau(p) \quad \forall p \in P & \text{- } P\text{-reproducibility} \\ \omega(\tau, \lambda_i) = 0 \quad \Rightarrow a_i(\tau) = 0 & \text{- local support} \\ \|\mathbf{a}(\tau)\|_{L_1} \le C \quad \forall \tau \in V^* & \text{- uniform boundedness} \end{cases}$$

H. Wendland. Scattered data approximation, Vol. 17. Cambridge university press, 2004.

A GMLS tutorial

SAND2019-XXXX C

Theorem. The GMLS coefficients $a_i(\tau)\in\mathbf{R}$ solve a (local) Quadratic Program (QP):

$$\min \frac{1}{2}\sum_{i=1}^{N}\frac{a_i^2(\tau)}{\omega(\tau,\lambda_i)} \quad \text{such that} \quad \sum_{i=1}^{N} a_i(\tau)\lambda_i(p)=\tau(p) \quad \forall p\in P$$

$\mathbf{a}(\tau)=\left[a_i(\tau)\right]\in\mathbf{R}^N$ $\quad$ $\tau(\mathbf{p})=\left[\tau(p_j)\right]\in\mathbf{R}^Q$ $\quad$ $L=\left[\lambda_j(p_i)\right]\in\mathbf{R}^{Q\times N}$

$W(\tau)=diag\left[\omega(\tau,\lambda_j)\right]\in\mathbf{R}^{N\times N}$ $\quad$ $\mathbf{u}=\left[\lambda_i(u)\right]\in\mathbf{R}^N$ - sample vector

Algebraic form of the QP

$$\min \frac{1}{2}\mathbf{a}(\tau)^T W^{-1}(\tau)\mathbf{a}(\tau) \quad \text{such that} \quad L\mathbf{a}(\tau)=\tau(\mathbf{p})$$

QP solution: GMLS "basis" functions

$$\mathbf{a}(\tau)=W(\tau)L^T\left(LW(\tau)L^T\right)^{-1}\tau(\mathbf{p})$$

GMLS approximation of the action of $\tau\in V^*$ on $u\in V$:

$$\overline{\tau}(u)=\mathbf{u}^T\left[W(\tau)L^T\left(LW(\tau)L^T\right)^{-1}\tau(\mathbf{p})\right]=\mathbf{u}^T\mathbf{a}(\tau)$$

GMLS basis form: sum of field samples $\lambda_i(u)$.

We can also group the terms **as follows**

$$\overline{\tau}(u)=\left[\mathbf{u}^T W(\tau)L^T\left(LW(\tau)L^T\right)^{-1}\right]\tau(\mathbf{p})=\mathbf{b}(\tau)^T\tau(\mathbf{p})$$

The coefficients $\mathbf{b}(\tau)$ solve an algebraic WLS problem:

$$\mathbf{b}(\tau)=\underset{\mathbf{c}\in\mathbf{R}^N}{\operatorname{argmin}}\frac{1}{2}\left(L^T\mathbf{c}-\mathbf{u}\right)^T W(\tau)\left(L^T\mathbf{c}-\mathbf{u}\right)=\underset{\mathbf{c}\in\mathbf{R}^N}{\operatorname{argmin}}\frac{1}{2}\left\|L^T\mathbf{c}-\mathbf{u}\right\|_{W(\tau)}^2$$

GMLS is not a polynomial regression!

Example: field approximation from point values

Not a polynomial! ➡

$$p_\tau(x) = \sum_{i=1}^{Q} b_i(x) p_i(x)$$

$p_i(x)$ A *polynomial* basis function

$b_i(x)$ A coefficient depending on the *spatial location*

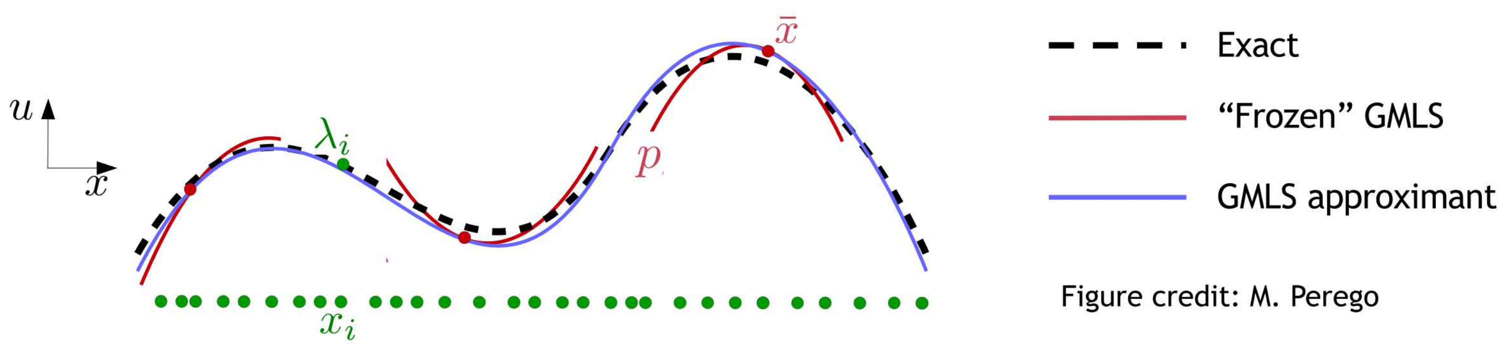


Figure credit: M. Perego

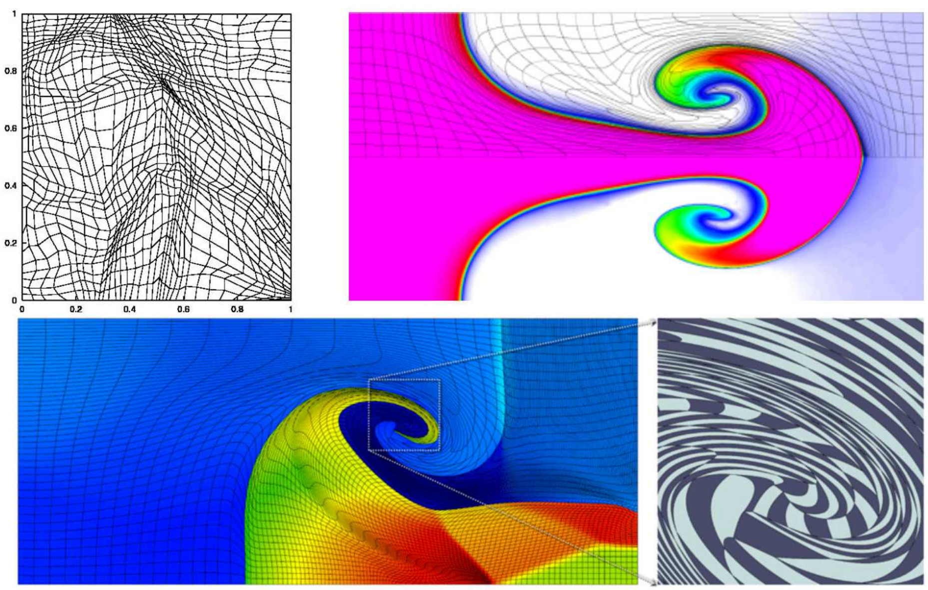
This is non-parametric regression of the data: approximant not known in closed form but can be effectively computed at any given point.

Application #1: "mesh-hardened" DG

SAND2019-XXXX C

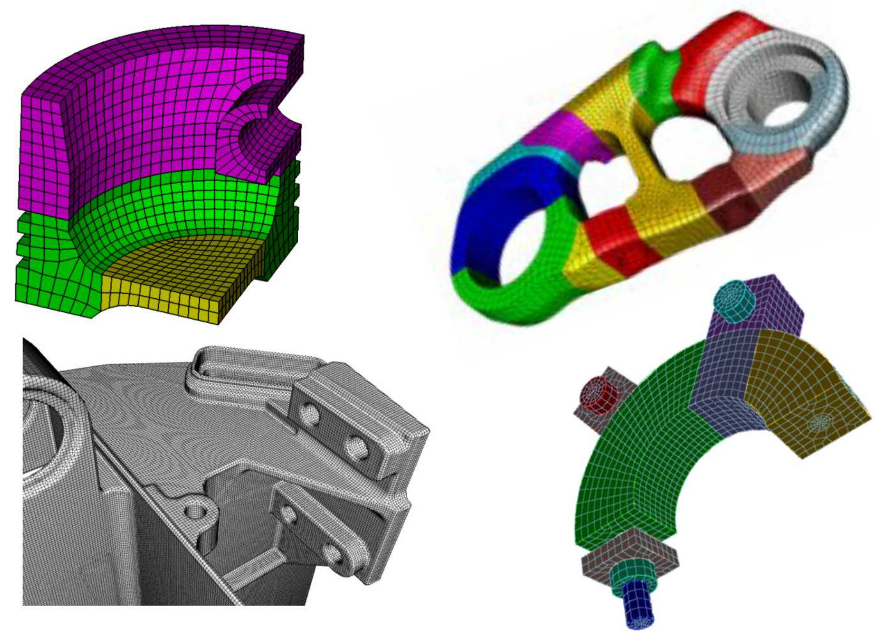
What is the problem? Finite Element Methods (FEM) work best on shape-regular meshes. But…

Sometimes we have **no choice** but compute on highly distorted meshes as in Lagrangian and ALE hydro codes:



Tzanio Kolev (BLAST hydro code, LLNL), M. Shashkov (LANL)

Generation of high-quality grids can take **up to 75%** of the total time-to-solution (Dart System Analysis: SAND2005-4647)



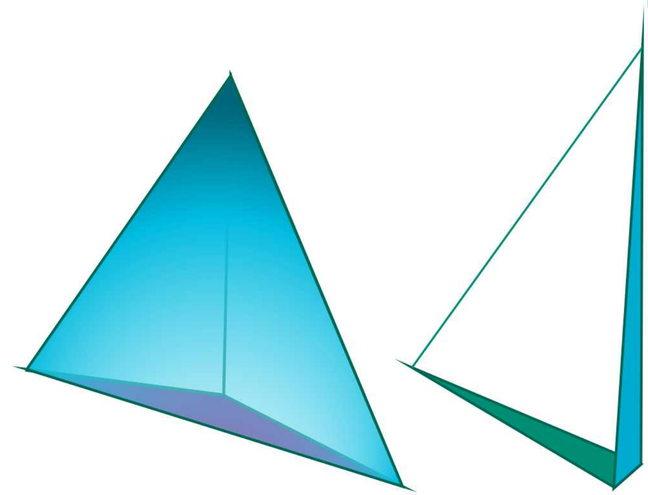
http://cubit.sandia.gov

What can we do about this problem?

Problem:

FEM solution quality **depends** on shape function quality:

Quality of standard FEM shape functions is **tied** to mesh quality



I. Babuska and A. Aziz. On the angle condition in the finite element method. SIAM Journal on Numerical Analysis, 13(2):214–226, 1976

Solution:

Divorce shape function quality from the mesh quality!

How can we do this?

Go Meshless!

- **We will use the mesh only for integration which can be done accurately even on a bad mesh**
- **We will extend Generalized Moving Least Squares (GMLS) to approximate bilinear forms**

A little bit of history

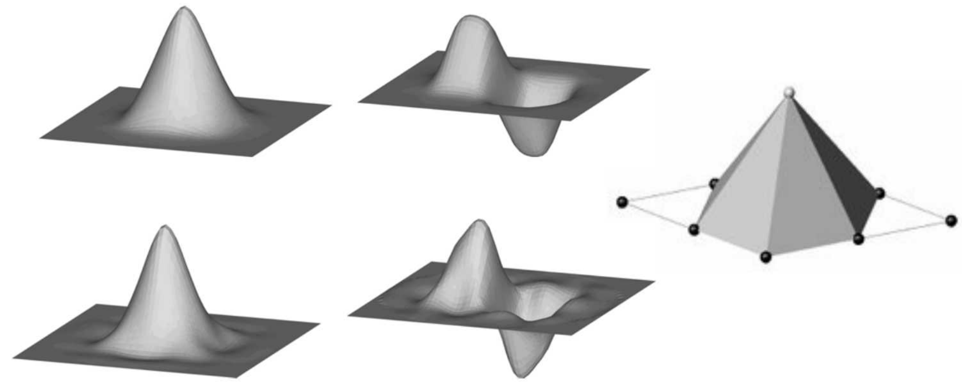
SAND2019-XXXX C

The GMLS “basis” functions $a_i(\tau)$ have been used as FE shape functions by Nayroles (1992), Belytschko (1994), Atluri (1998), Mirzaei (2013) and others.

However, $a_i(\tau)$ is non-polynomial and requires expensive quadrature.

J.S. Chen uses under-integration to avoid this problem and obtain computationally efficient methods for large deformation solid mechanics.

However, under-integration introduces instabilities and requires development of problem-specific stabilization techniques.



A (G)MLS shape function and its derivative for Gaussian and regularized $\omega(.,.)$ and a standard *P1* shape function on triangles.

T. Most, C. Bucher, *Structural engineering and mechanics*, 21/3, pp.315-332

Chen, J.S., Hillman, M., Rüter, M.: An arbitrary order variationally consistent integration for Galerkin meshfree methods. Int. J. Num. Meth. Engrg. 95(5), 387– 418 (2013).

Chen, J.S., Wu, C.T., Yoon, S., You, Y.: A stabilized conforming nodal integration for Galerkin mesh-free methods. Int. J. Num. Meth. Engrg. 50(2), 435–466 (2001).

SAND2019-XXXX C

Extension of GMLS to approximation of bilinear forms

An abstract setting: Consider a variational problem set in a Hilbert space *V*:

$$seek \;\; u \in V \;\; such\ that \;\; a(u,v) = f(v) \;\; for\ all \;\; v \in V$$

$$a(\cdot,\cdot): V \times V \to \mathbb{R}; \quad f(\cdot): V \to \mathbb{R}$$

$f(\cdot)$ is a linear functional ⇒ can use GMLS to approximate it:

$$f(v) \approx \tilde{f}(v) \coloneqq \boldsymbol{b}(\boldsymbol{v}) \cdot f(\boldsymbol{p})$$

$a(\cdot,\cdot)$ is not a linear functional so GMLS does not readily apply.

$$a(u,v) \approx \tilde{a}(u,v) \coloneqq b(\boldsymbol{v}) \cdot a(\boldsymbol{p},\boldsymbol{p}) \cdot \boldsymbol{b}(\boldsymbol{u}).$$

How can we extend GMLS? Here's the key idea:

- Holding the test function $v \in V$ fixed gives a linear functional $a_v(\cdot) \coloneqq a(\cdot, v)$
- Holding the trial function fixed gives another linear functional $a_u(\cdot) \coloneqq a(u,\cdot)$

$$seek \;\; \boldsymbol{u} \in \boldsymbol{R}^N \;\; such\ that \;\; \boldsymbol{b}(\boldsymbol{v}) \cdot a(\boldsymbol{p},\boldsymbol{p}) \cdot \boldsymbol{b}(\boldsymbol{u}) = \boldsymbol{b}(\boldsymbol{v}) \cdot f(\boldsymbol{p}) \;\; for\ all \;\; \boldsymbol{u} \in \boldsymbol{R}^N$$

The (global) discrete problem

Application to a model PDEs: 1. **Setup**

SAND2019-XXXX C

1. Variational problem

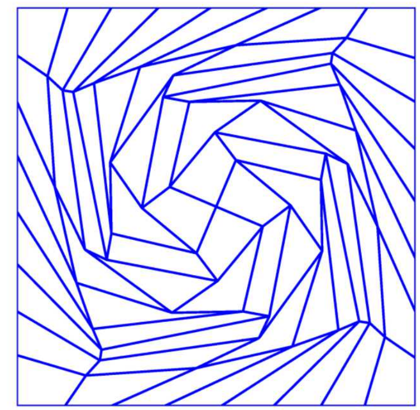
$$seek \;\; u \in V \;\; such\ that \;\; a(u,v) = f(v) \;\; for\ all \;\; v \in V$$

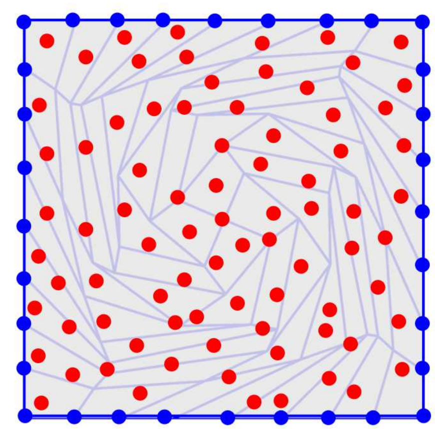
representing a weak form of $-\varepsilon \Delta u + \mathbf{b} \cdot \nabla u = f \;\text{ in } \Omega \;\text{ and }\; u = 0 \;\text{ on } \Gamma$

2. FE mesh Ω^h with elements $\{\mathcal{K}_k\}_{k=1}^{N_e}$ that may be low-quality.

3. A quadrature rule on each element that is exact for linear fields.

4. A point cloud $X^\eta \subset \Omega$ comprising points $\{\boldsymbol{x}_i\}_{i=1}^{N_p}$.





- No relationship assumed between the mesh and point cloud.
- Of course, mesh nodes can be points in the cloud.
- We seek an approximate solution on the point cloud, not the mesh nodes.

SAND2019-XXXX C

Application to a model PDE: 2. **Local discrete problems**

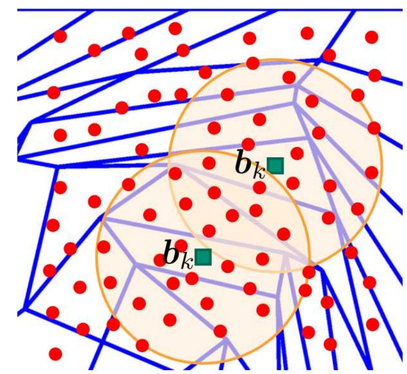
1. We start by writing the bilinear form and the RHS functional as sums over the elements:

$$a(u,v) \;=\; \sum_{k=1}^{N_e} a_k(u,v) \qquad\qquad f(v) \;=\; \sum_{k=1}^{N_e} f_k(v)$$

2. We consider a GMLS with $P = P_m$ and kernel $w(\mathcal{K}_k, \boldsymbol{x}_j) := \rho(|\boldsymbol{b}_k - \boldsymbol{x}_j|)$ where

- $\boldsymbol{b}_k$ is the centroid of element $\mathcal{K}_k$.
- $\rho(\cdot)$ is a radially symmetric, positive kernel function with $\operatorname{supp}\rho = O(h)$.

Local sampling set



$$S_k = \{\delta_{\boldsymbol{x}_j} \,|\, w(\mathcal{K}_k, \boldsymbol{x}_j) > 0\}$$

3. We construct approximations of a_k and f_k locally from point values:

$S^k = \{u_1^k, \ldots, u_{n_k}^k\}$ local approximation space = local DoF set

$$\tilde{a}_k(u^k, v^k) \coloneqq \boldsymbol{b}(v^k)\cdot a_k(\boldsymbol{p},\boldsymbol{p})\cdot \boldsymbol{b}(u^k); \qquad \tilde{f}_k(v^k) \coloneqq \boldsymbol{b}(v^k)\cdot f_k(\boldsymbol{p}); \qquad u^k,\, v^k \in S_k$$

SAND2019-XXXX C

Application to a model PDE: 3. **Local and global assembly**

Local stiffness matrix and load vector: $\boldsymbol{p} = \{p_1, \cdots, p_q\}$. **Local GMLS basis** – spans P_m

$$(\boldsymbol{A}_k)_{ij} = a_k(p_j, p_i) \quad a_k(p_j, p_i) = \int_{\mathcal{K}_k} \varepsilon \nabla p_j \cdot \nabla p_i + (\mathbf{b} \cdot \nabla p_j) p_i \, dx$$

$$(\boldsymbol{f}_k)_i = f_k(p_i) \qquad f_k(p_i) = \int_{\mathcal{K}_k} f \cdot p_i \, dx$$

- Requires only integration of **polynomials**!
- Can be performed by any **standard rule.**

Global problem discrete problem:

$[S] = \cup_{\mathcal{K}_k \in \Omega^h} S^k$ global approximation space = union of all local DoF sets

$$\widetilde{a}([u],[v]) := \sum_{\mathcal{K}_k \in \Omega^h} \widetilde{a}_k(u^k, v^k)$$

$$\widetilde{f}([v]) := \sum_{\mathcal{K}_k \in \Omega^h} \widetilde{f}_k(v^k)$$

Seek $[u] \in [S]$ such that $\tilde{a}([u],[v]) = \tilde{f}([v]) \quad \forall\, [v] \in [S]$

The global problem is a non-conforming discretization!

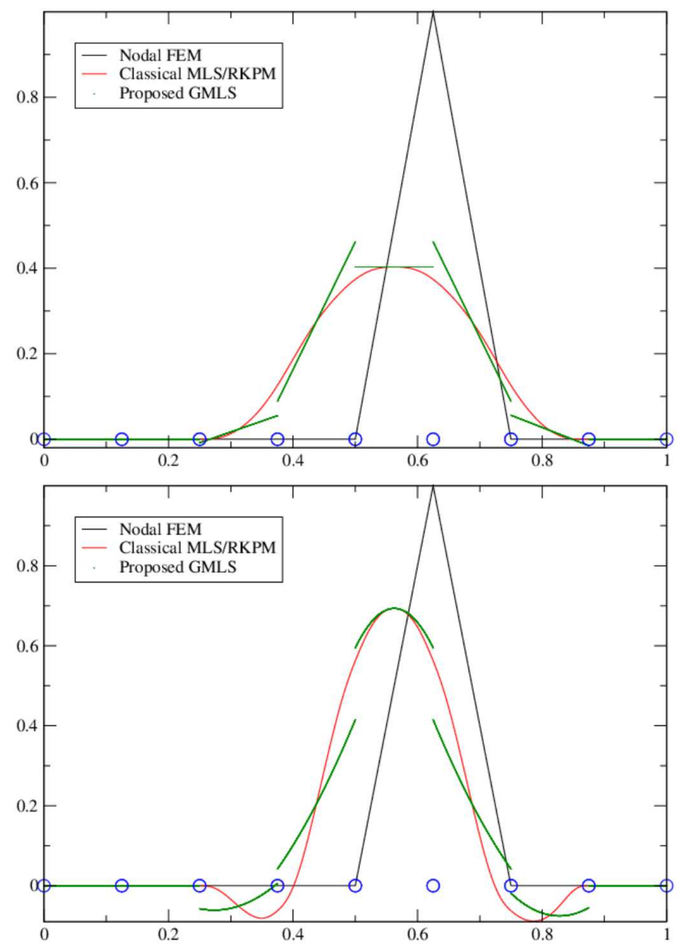
One can show that the global discrete problem is equivalent to a non-conforming discretization in terms of discontinuous piecewise polynomial shape functions.

The nature of non-conformity is similar to that in, e.g., Interior Penalty and Discontinuous Galerkin (DG) methods.

Thus, we can use standard techniques from IP and DG to stabilize our formulation by adding suitable penalized jump terms.

Here we shall use the same treatment of advective and diffusive terms as in the "original" DG method; see

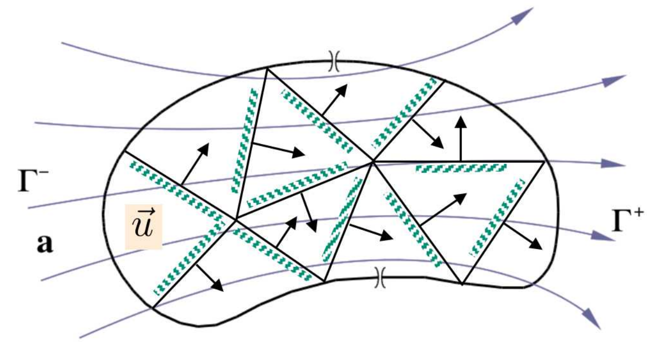
Cockburn, B., Dong, B., Guzman, J.: Optimal convergence of the original DG method for the transport-reaction equation on special meshes. SIAM J. Numer. Anal. 46(3), 1250–1265 (2008)



A stabilized global formulation using a classical DG approach

Stabilization of the advective term

$$\vec{a}_k(u,v) = \sum_{\mathcal{K}_k \in \Omega^h} \int_{\mathcal{K}_k} \nabla u \cdot \nabla v dx - \int_{\mathcal{K}_k} u\mathbf{b} \cdot \nabla v dx + \int_{\partial\mathcal{K}_k} \vec{u} v \mathbf{b} \cdot \boldsymbol{n}_k dS$$



Stabilization of the diffusive term

$$a_k^{DG}(u,v) = \vec{a}_k(u,v) - \sum_{\mathcal{F}} \int_{\mathcal{F}} \{\{\nabla u\}\} \cdot [\![v]\!] dS + \int_{\mathcal{F}} v \cdot [\![u]\!] dS - \frac{\delta}{h} \int_{\mathcal{F}} [\![u]\!] \cdot [\![v]\!]\rangle_{\mathcal{F}} dS$$

$$\{\{\boldsymbol{q}\}\} = \frac{1}{2}(\boldsymbol{q}_1 + \boldsymbol{q}_2)$$

$$\{\{\varphi\}\} = \frac{1}{2}(\varphi_1 + \varphi_2)$$

$$[[\boldsymbol{q}]] = \boldsymbol{q}_1 \cdot \boldsymbol{n}_1 + \boldsymbol{q}_2 \cdot \boldsymbol{n}_2$$

$$[[\varphi]] = \varphi_1 \boldsymbol{n}_1 + \varphi_2 \boldsymbol{n}_2$$

Average & Jump

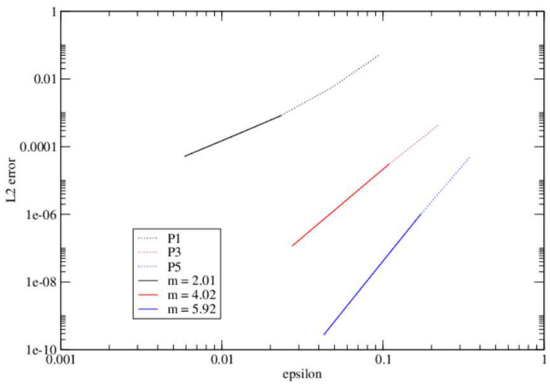
Global stabilized discrete problem

Seek $[u] \in [S]$ such that $\tilde{a}^{DG}([u],[v]) = \tilde{f}([v]) \;\; \forall\, [v] \in [S]$

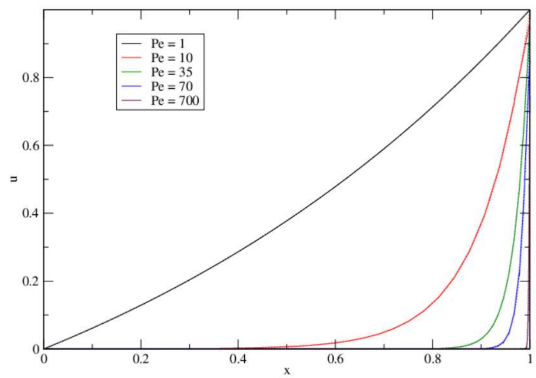
Bochev, Kuberry, Trask, Perego. Mesh-hardened finite element analysis through a GMLS approximation of variational problems. Springer LNCS, 2020.

Numerical examples

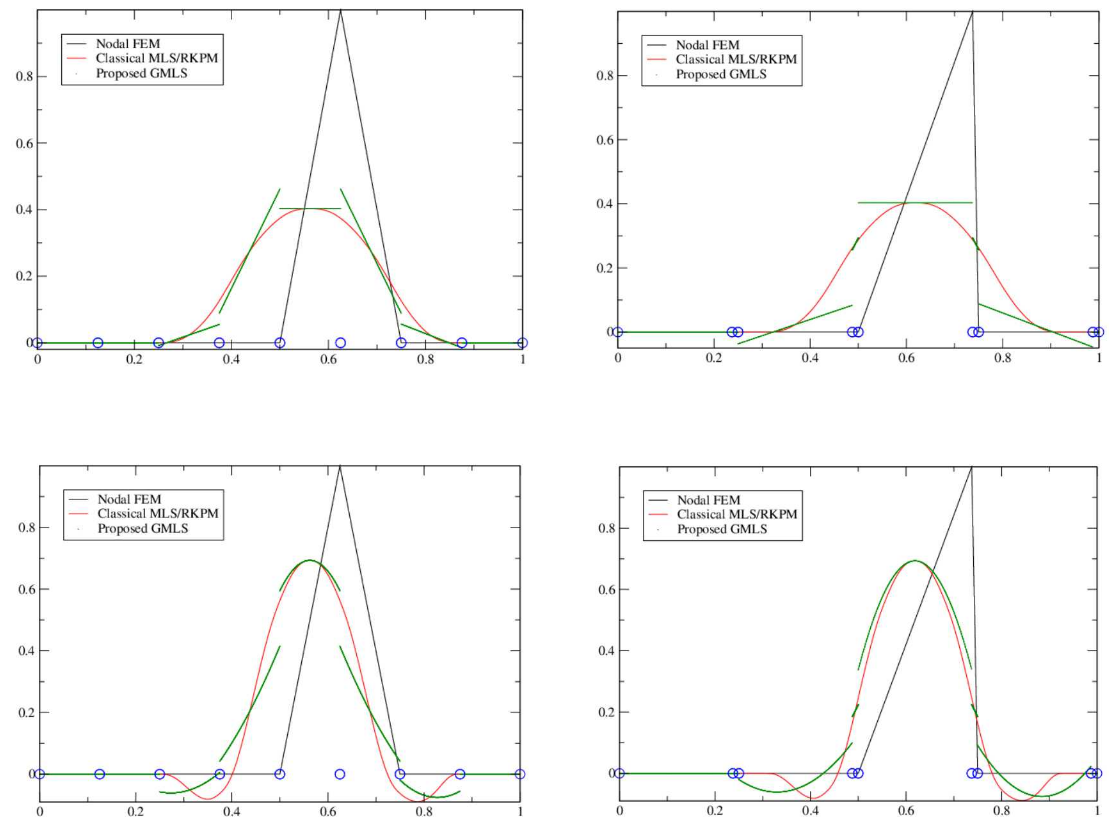
Convergence



Solution for increasing Pe



Shape function comparison



Uniform mesh

Non-uniform mesh

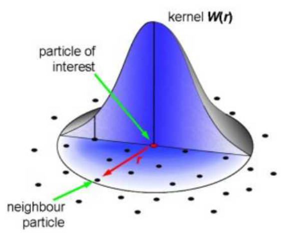
Application #2: meshfree mimetic divergence

What is the problem: Something is missing in the meshfree universe!

Today, most meshfree methods for $D^\alpha u(x) = f(x)$ look like this:

$$u_\varepsilon^h(x) = \sum_p u(x_p) W_\varepsilon(x - x_p) \quad \rightarrow \quad D^\alpha u_\varepsilon^h(x) = \sum_p u(x_p) D^\alpha W_\varepsilon(x - x_p) \quad \rightarrow \quad \sum_p u(x_p) D^\alpha W_\varepsilon(x - x_p) = f(x)$$

Local kernel estimate of the field — Derivative approximation — PDE discretization



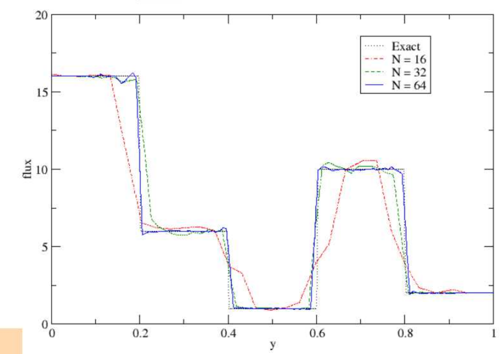
Creates conflicts between consistency and conservation

- SPH is conservative but not P0 concistent. RKPM is P1 consistent but not conservative

Mathematically equivalent to node-based (or collocated) methods

- Unsuitable for mixed discretizations needed in Drift-Diffusion, subsurface flow,…

Many such methods perform poorly on the **5-strip problem**, which tests their ability to reproduce fields that are in *H(div)* but not in H^1 – a critical requirement for **mixed discretizations**



→ **“Compatible” or “mimetic” meshless methods lag behind their mesh-based cousins!**

What can we do about this problem?

SAND2019-XXXX C

We will build a Meshfree Mimetic Divergence operator (MMD)! How?

By mimicking the construction of a mimetic mesh-based divergence operator $DIV : F \to C$ **:**

Divergence theorem

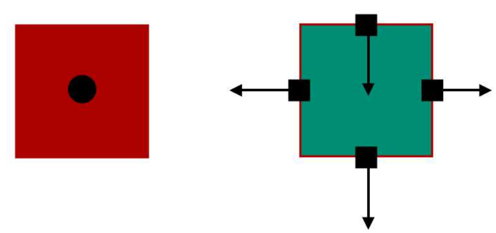
$$\int_C \nabla \cdot \boldsymbol{u}\, dV = \int_{\partial C} \boldsymbol{u} dA$$

Discrete Stokes theorem on cochains

$$\int_{\boldsymbol{c}} \nabla \cdot \boldsymbol{u}\, dV = \sum_{f \in \partial \boldsymbol{c}} \int_f \boldsymbol{u} dA$$

Mimetic divergence operator

$$DIV(\mathcal{F})|_C = \frac{1}{\mu_C} \sum_{f \in \partial C} \mathcal{F}_f \mu_f$$

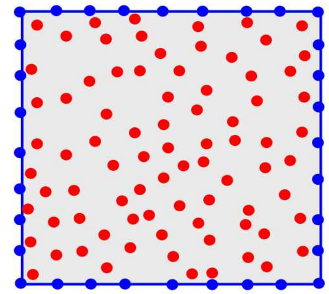


The mimetic divergence *DIV* is constructed from the following data:

- **Field data:** given by the *face fluxes* $\mathcal{F}_f$
- **Topological data:** given by the action of the *boundary operator* ∂ on cells.
- **Metric data:** given by the measures $\mu_C = |C|$ and $\mu_f = |f|$, of a cell and its faces, respectively

SAND2019-XXXX C

An abstract Meshfree Mimetic Divergence (MMD) operator



A standard meshfree environment:

- A point cloud $X \subset \Omega$ comprising points $\{\boldsymbol{x}_i\}_{i=1}^N$.
- Field representations by point samples at $\boldsymbol{x}_i$: $\boldsymbol{u} \rightarrow \{\boldsymbol{u}^h \in \boldsymbol{R}^N \,|\, \boldsymbol{u}_i^h = \boldsymbol{u}(\boldsymbol{x}_i)\}$

An MMD habitat:

- Notions of *virtual cells* $C = \{c_i\}$, indexed by $\boldsymbol{x}_i \in X$ and *virtual faces* $F = \{f_i\}$,
- A notion of *virtual boundary operator* $\partial' : C \rightarrow F$
- A set of *real numbers* $\{\boldsymbol{\mu_c}\}$ and a set of *real vectors* $\{\boldsymbol{\mu}_f\}$, indexed by *virtual cells* and *faces*, and
- An operator $T : X \rightarrow F$ mapping point samples $\{\boldsymbol{u}^h\}$ to *real vectors* $\{\boldsymbol{u}_f\}$ indexed by *virtual faces*.

$\boldsymbol{u}^h \in \boldsymbol{R}^N$
$\boldsymbol{u}_f \in \boldsymbol{R}^M$
$\boldsymbol{\mu}_f \in \boldsymbol{R}^M$
$\boldsymbol{\mu}_c \in \boldsymbol{R}$

This MMD habitat provides abstractions of the data needed to construct the mimetic *DIV*

- **Field data:** given by the *real vectors* $\{\boldsymbol{u}_f\}$: **field moments**
- **Topological data:** given by the action of the virtual operator ∂'on *virtual* cells.
- **Metric data:** given by the *real numbers* $\{\boldsymbol{\mu_c}\}$ and the *real vectors* $\{\boldsymbol{\mu}_f\}$: **metric moments**

Assumptions on the MMD habitat

SAND2019-XXXX C

"Topological" assumptions

Virtual boundary operator

T.0 $\partial' : C \to F$ satisfies $\partial'(\cup_{c \in C} c) = \partial\Omega$, i.e., ∂' recovers the physical domain boundary

Metric and field moments

T.1 $\boldsymbol{\mu_c} > 0$, $\boldsymbol{\mu_c} = O(h^d)$ and $\sum_c \boldsymbol{\mu_c} = |\Omega|$

T.2 $\boldsymbol{\mu}_{\vec{f}} = -\boldsymbol{\mu}_{\overleftarrow{f}}$ anti-symmetry

T.3 $\boldsymbol{u}_{\vec{f}} = +\boldsymbol{u}_{\overleftarrow{f}}$ symmetry of T

$\{\boldsymbol{\mu_c}\}, \{\boldsymbol{\mu}_f\}$
Metric moments

$\boldsymbol{u}_f = T(\boldsymbol{u}^h)$
Field moments

"Metric" assumptions

Local Lipschitz continuity: For any C^2 vector fields $\boldsymbol{u}$ and $\boldsymbol{v}$ with point samples $\{\boldsymbol{u}^h\}$ and $\{\boldsymbol{v}^h\}$

$$\left|\boldsymbol{u}_f \cdot \boldsymbol{\mu}_f - \boldsymbol{v}_f \cdot \boldsymbol{\mu}_f\right| \le C h^{d-1} \left\|\boldsymbol{u}^h - \boldsymbol{v}^h\right\|_\infty$$

The numbers $\{\boldsymbol{\mu_c}\}, \{\boldsymbol{\mu}_f\}$ and the operator T are a P_1 -reproducing pair:

$$\nabla \cdot \boldsymbol{p}(\boldsymbol{x_c}) = \frac{1}{\boldsymbol{\mu_c}} \sum_{f \in \partial' c} \boldsymbol{p}_f \cdot \boldsymbol{\mu}_f \qquad \forall \boldsymbol{p} \in P_1; \quad \forall \boldsymbol{x_c} \in X$$

∂' is the virtual boundary!

$\boldsymbol{p}_f = T(\boldsymbol{p}^h)$

SAND2019-XXXX C

MMD definition and analysis

Define the abstract meshfree mimetic divergence (MMD) operator $DIV: X \to C$ as

$$DIV\,\boldsymbol{u}_h := \frac{1}{\boldsymbol{\mu}_c} \sum_{f \in \partial' C} \boldsymbol{u}_f \cdot \boldsymbol{\mu}_f$$

- $\boldsymbol{u}^h$ is a point sample of a vector field
- $\boldsymbol{u}_f = T(\boldsymbol{u}^h)$ are the field moments,
- $\{\boldsymbol{\mu}_c\}, \{\boldsymbol{\mu}_f\}$ are the metric moments,
- ∂' is the virtual boundary!

$\boldsymbol{u}^h \in \boldsymbol{R}^N$
$\boldsymbol{u}_f \in \boldsymbol{R}^M$
$\boldsymbol{\mu}_f \in \boldsymbol{R}^M$
$\boldsymbol{\mu}_c \in \boldsymbol{R}$

Theorem.

Assume that the metric and field moments satisfy T.1-T.3, the local Lipschitz condition and the P1 reproduction property. Then, the abstract MMD operator is

- **Locally conservative:** $\sum_{c \in \omega} \boldsymbol{\mu}_c DIV \boldsymbol{u}_h = \sum_{f \in \partial' \omega} \boldsymbol{u}_f \cdot \boldsymbol{\mu}_f$ for any $\omega = \cup_{i \in I} c_i; \quad I \subseteq X$
- **First-order accurate:** $\|\nabla \cdot \boldsymbol{u} - DIV\, \boldsymbol{u}_h\| \leq Ch$ for any $\boldsymbol{u} \in C^2(\Omega)^d$

We will consider two instances of the abstract MMD operator:

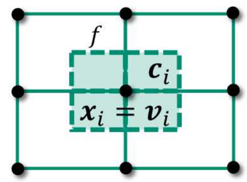
- #1 – **with** background mesh
- #2 – **without** background mesh

SAND2019-XXXX C

MMD Instance #1: with a background mesh

An MMD habitat with a background primal-dual mesh having dual cells C and dual faces F:

- *Virtual cells* $C = \{c_i\}$ $\longrightarrow$ dual mesh cells;
- *Virtual faces* $F = \{f_i\}$ $\longrightarrow$ dual mesh faces,
- *Virtual boundary* ∂' $\longrightarrow$ geometric boundary $\partial: C \to F$



$$\int_{c} \nabla \cdot \boldsymbol{u}\, dV = \sum_{f \in \partial c} \int_{f} \boldsymbol{u} dA$$

GMLS

- *Metric moments* $\{\boldsymbol{\mu}_c\}$ $\longrightarrow$ $|\boldsymbol{c}|$ dual cell volumes
- *Metric moments* $\{\boldsymbol{\mu}_f\}$ $\longrightarrow$ $\tau_f(\boldsymbol{p})$ GMLS basis moments
- *Field Moments* $\{\boldsymbol{u}_f\}$ $\longrightarrow$ $\boldsymbol{b}_f(\boldsymbol{u}^h)$ GMLS coefficients

$$\boldsymbol{b}_f(\boldsymbol{u}^h) \cdot \tau_f(\boldsymbol{p}) \approx \tau_f(\boldsymbol{u})$$

approximation target

Abstract MMD

$$DIV \boldsymbol{u}_h := \frac{1}{\boldsymbol{\mu}_c} \sum_{f \in \partial' C} \boldsymbol{u}_f \cdot \boldsymbol{\mu}_f$$

$$DIV_1\, \boldsymbol{u}^h := \frac{1}{|\boldsymbol{c}|} \sum_{f \in \partial \boldsymbol{c}} \boldsymbol{b}_f(\boldsymbol{u}^h) \cdot \tau_f(\boldsymbol{p})$$

MMD Instance #1

- MMD #1 satisfies a discrete divergence theorem
- Useful for poor quality meshes with near singular basis functions.

SAND2019-XXXX C

MMD Instance #2: without a background mesh

Without a background mesh some of the data necessary to instantiate the abstract MMD is **missing**:

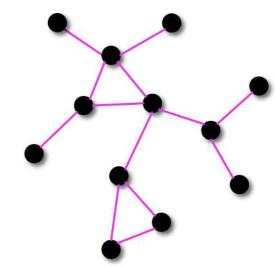
- ✓ **Field data:** $u_f = b_f(u^h)$ **GMLS coefficients**
- ✗ **Topological data:** $\partial c_i = \{f_{ij}\}$ **geometric boundary**
- ✗ **Metric data:** $\mu_c = |c|$ **and** $\mu_f = \tau_f(p) = \int_f p \, dA$

$$DIV\, \boldsymbol{u}_h := \frac{1}{\boldsymbol{\mu_c}} \sum_{f \in \partial' c} \boldsymbol{u}_f \cdot \boldsymbol{\mu}_f$$

- The **missing pieces of data** are exactly the ones that could be trivially obtained on the mesh!
- We will construct analogues of the missing data that are actually cheaper than building a mesh!

Our plan for the second MMD instance:

- ✓ **Field data:** $u_f = b_f(u^h)$ **keep the GMLS coefficients**
- ✓ **Topological data:** **use the ε-ball graph of the point cloud as a mesh surrogate**
- ✓ **Metric data:** **define by solving a suitable algebraic problem**



$$Ax = b$$

SAND2019-XXXX C

MMD Instance #2: Topological Data

We endow *X* with a virtual primal-dual mesh complex (**a mesh surrogate**) as follows:

Virtual primal mesh: $G_{\varepsilon_g}(V,E)$ **the ε-ball graph of the point cloud**

Vertices: *V*=*X* (the points in the cloud)

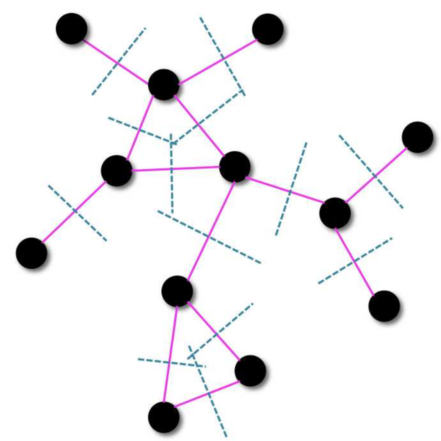
Edges: $E := \{\boldsymbol{e}_{ij} = (\boldsymbol{x}_i, \boldsymbol{x}_j) \in V \times V \,\big|\, |\boldsymbol{x}_i - \boldsymbol{x}_j| < \varepsilon_g\}$

Virtual dual mesh: $G'_{\varepsilon_g}(C,F)$ **the "formal" dual of** $G_{\varepsilon_g}(V,E)$

Cells: assign a virtual cell $\boldsymbol{c}_i$ to every vertex $\boldsymbol{x}_i$

Faces: assign a virtual face $\boldsymbol{f}_{ij}$ to every edge $\boldsymbol{e}_{ij}$

Virtual boundary: $\partial' \boldsymbol{c}_i = \{\boldsymbol{f}_{ij}\}$



$G_{\varepsilon_g}(V,E)$ can be constructed with *O(N)* complexity using binning algorithms.

MMD Instance #2: Metric Data

Constructing the metric data on a point cloud X with N points:

Virtual cell volumes: assume quasi-uniform point cloud

$$\boxed{\boldsymbol{\mu}_c := \frac{|\boldsymbol{\Omega}|}{N}}$$

$$q_{X_\Omega} \leq h_{X_\Omega,\Omega} \leq c_{qu} q_{X_\Omega}$$

$$h_{X_\Omega,\Omega} = \sup_{\boldsymbol{x}\in\Omega} \min_{\boldsymbol{x}_i \in X_\Omega} |\boldsymbol{x} - \boldsymbol{x}_i| \quad \longleftarrow \text{fill}$$

$$q_{X_\Omega} = \frac{1}{2} \min_{i \neq j} |\boldsymbol{x}_i - \boldsymbol{x}_j| \quad \longleftarrow \text{separation}$$

Virtual face areas: seek in terms of a scalar potential

$$\boxed{\boldsymbol{\mu}_f^k = \left(GRAD\ \phi^k\right) \boldsymbol{p}^k}$$

p^k kth basis function of P

ϕ^k scalar function on X

$GRAD: V \longrightarrow E$ topological gradient

MMD Instance #2: Metric Data

SAND2019-XXXX C

Equations for the virtual face areas

Recall the P-reproducibility condition on the virtual metric data:

$$\nabla \cdot \boldsymbol{p}(\boldsymbol{x}_c) = \frac{1}{\boldsymbol{\mu}_c} \sum_{f \in \partial' C} \boldsymbol{b}^T(\boldsymbol{p})\boldsymbol{\mu}_f \quad \forall \boldsymbol{p} \in P_1; \quad \forall \boldsymbol{x}_c \in X$$

Inserting the virtual face area ansatz $\boldsymbol{\mu}_f^k = \left(GRAD\ \phi^k\right) p^k$ yields

$$\boxed{DIV\left(GRAD\ \phi^k\right) \boldsymbol{p}^k(\boldsymbol{x}_c) = \nabla \cdot \boldsymbol{p}^k(\boldsymbol{x}_c)}$$

- A weighted graph Laplacian problem for each basis function. Solution cost *O(N)* using AMG
- Trades a challenging computational geometry problem (meshing) for a benign algebraic one.

The two instances of the abstract MMD operator at a glance

#1: MMD with a background mesh:		#2: MMD without a background mesh:
$DIV\,\boldsymbol{u}_h := \frac{1}{\lvert C\rvert}\sum_{f\in\partial C} \boldsymbol{b}(\boldsymbol{u})\cdot\tau_f(\boldsymbol{p})$		$DIV\,\boldsymbol{u}_h := \frac{1}{\boldsymbol{\mu_c}}\sum_{f\in\partial C} \boldsymbol{b}(\boldsymbol{u})\cdot\boldsymbol{\mu}_f(\boldsymbol{p})$
• Defined by the GMLS coefficients: $T\colon \boldsymbol{u}^h \to \boldsymbol{b}(\boldsymbol{u})$	**Field moments (the map T)**	• Defined by the GMLS coefficients: $T\colon \boldsymbol{u}^h \to \boldsymbol{b}(\boldsymbol{u})$
• Defined by the GMLS target: $\boldsymbol{\mu}_f := \tau_f(\boldsymbol{p});\quad \tau_f(\boldsymbol{p}) = \int_f \boldsymbol{p}\,dA$	**Face moments**	• Defined by a graph Lapacian: $\boldsymbol{\mu}_f^k = \left(GRAD\,\phi^k\right)p^k;\quad \Delta\,\phi^k = \nabla\cdot\boldsymbol{p}^k$
• Defined by actual cell volumes: $\boldsymbol{\mu_c} := \lvert\boldsymbol{C}\rvert$	**Cell moments**	• Defined algebraically: $\boldsymbol{\mu_c} := \frac{\lvert\boldsymbol{\Omega}\rvert}{N}$

SAND2019-XXXX C

Application of the abstract MMD theory

Assumptions checklist

Property	MMD with mesh	MMD without mesh	
T.1 $\boldsymbol{\mu_c} > 0,\ \boldsymbol{\mu_c} = O(h^d);\ \sum_c \boldsymbol{\mu_c} = \lvert\Omega\rvert$	✓	✓	Topological
T.2 $\boldsymbol{\mu}_{\vec{f}} = -\boldsymbol{\mu}_{\overleftarrow{f}}$	✓	✓	
T.3 $\boldsymbol{u}_{\vec{f}} = +\boldsymbol{u}_{\overleftarrow{f}}$	✓	✓	
Local Lipschitz	✓	TBD	Metric
P1 reproduction	✓	TBD	

MMD with a background mesh:

- **Locally conservative**
- **Provably** first-order accurate

MMD without a background mesh:

- **Locally conservative**
- **Numerically** first-order accurate

SAND2019-XXXX C

A historical perspective

The idea to construct virtual metric data had been used before in:

The Uncertain Grid Method (UGM)

O. Diyankov. Uncertain grid method for numerical solution of PDEs. Technical report, NeurOK Software, 2008.

- First example of a meshfree "finite volume" scheme
- Uncertain refers to faces between two adjacent points (our virtual face)
- First-order accurate

The Conservative Meshfree Scheme (CMS)

E. Kwan-yu Chiu, Q. Wang, R. Hu, and A. Jameson. A conservative mesh-free scheme and generalized framework for conservation laws. SISC, 34(6) 2012.

- Similar in principle to UGM
- First-order accurate

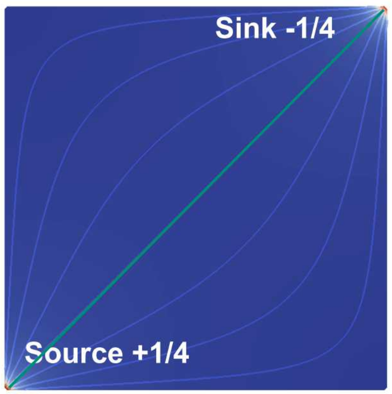
The key differences with our approach:

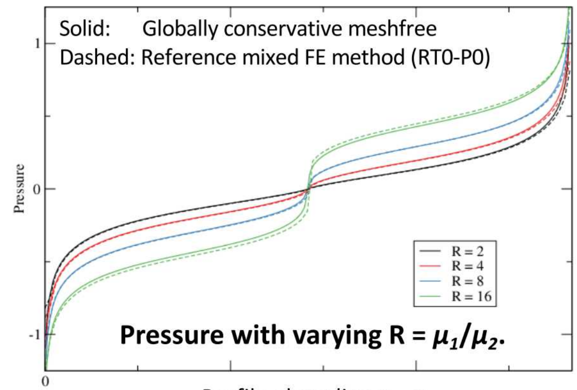
- GMLS enables extension of our scheme to high-order accuracy (in progress)
- Both UGM and CMS involve **expensive** **global constrained optimization** problems:
  - UGM ☛ LP solved by primal-dual log-barrier method (involves Newton)
  - CMS ☛ QP which requires a specialized QP solver

SAND2019-XXXX C

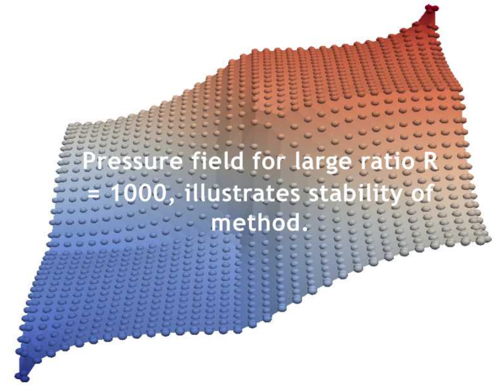
Numerical examples

The five spot problem : Tests conservation

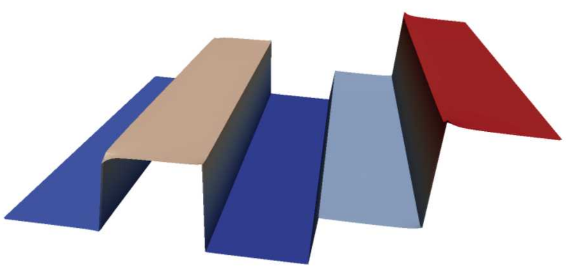


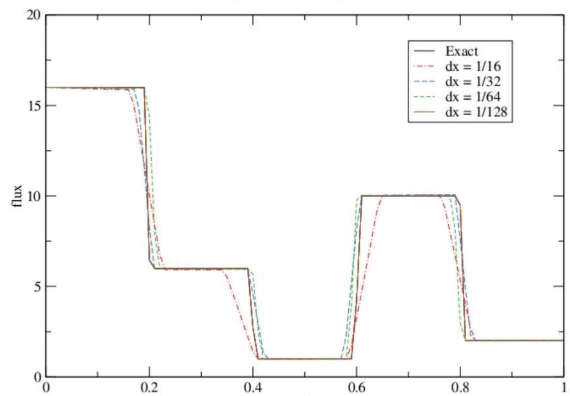


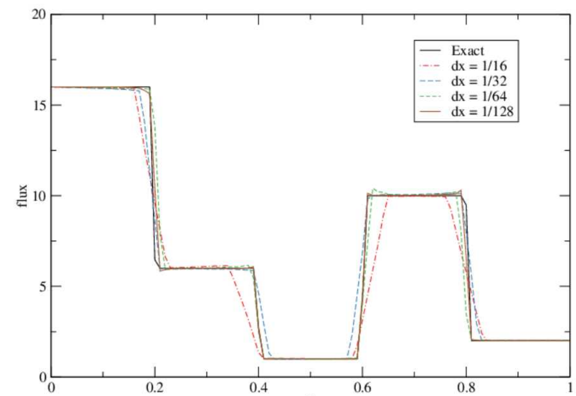
Profile along line y = x



The five strip problem (Hughes et al): Tests *H(div)* compliance



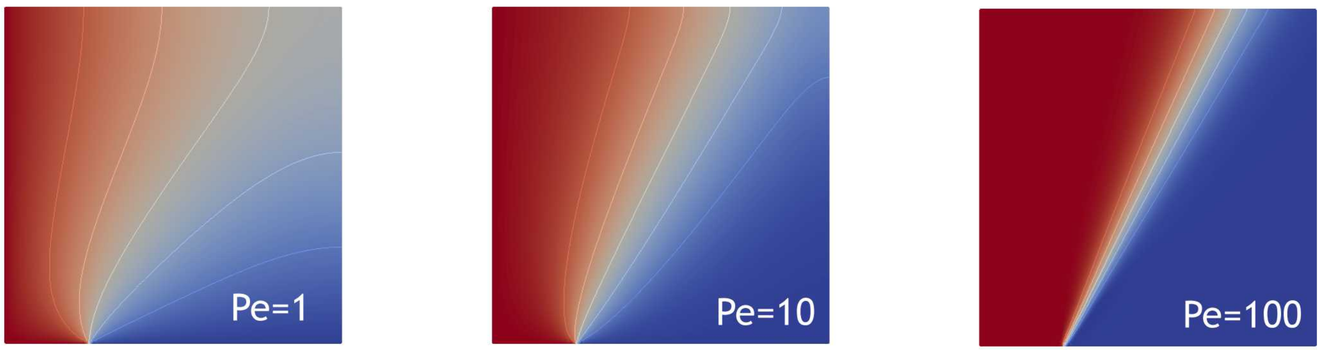




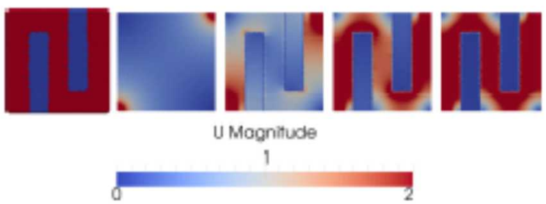
T.J.R Hughes et al, CMAME

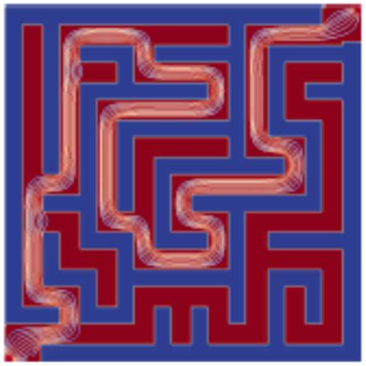
Fun numerical examples

Advection-diffusion



Finding your way out of a maze





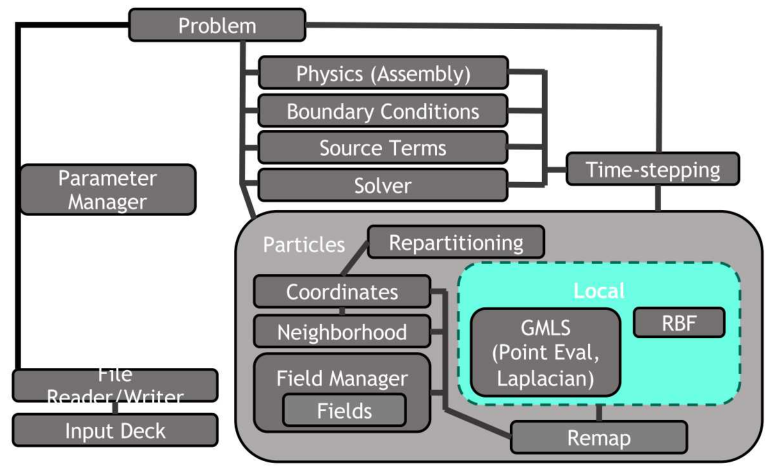
The Compadre Toolkit

Coupling Approaches for Next Generation Architectures (CANGA)

SAND2019-XXXX C

A modern, performant software library for meshfree and particle methods on different architectures.

- Everything you need to do particle/meshfree methods
- Developers: P. Kubbery, N. Trask, P. Bosler
- Initial development funded by Sandia's LDRD program
- Continuing support by the CANGA SciDAC project



Compadre Toolkit v. 1.0 DOI: [10.11578/dc.20190411.1](10.11578/dc.20190411.1)

Manages

- Input deck parsing
- Parallel file reading/writing (ASCII CSV, VTK, and Netcdf)
- Neighborhood searches (k-d tree)
- Euclidean & Spherical coordinate systems
- Registering fields of various dimensions
- Sets of particles
  - Data transfer between sets (remap)
  - Partitioning/repartitioning particle sets over multiple processors (Zoltan2)

Leverages Trilinos Tools

- Trilinos/Zoltan2 particles over processor partitioning
- Trilinos solvers (Amesos2, Ifpack2, MueLu, Belos, Teko, Thyra, Stratimikos) for stationary problems (time-dependent pending)

Distribution

https://www.osti.gov/doecode

DOE CODE U.S. Department of Energy Office of Scientific and Technical Information

https://github.com/SNLComputation/compadre

Conclusions

Good news!

- We extended GMLS to approximation of bilinear forms
- This approximation is equivalent to a non-conforming FE
    - Quality of these shape functions does not depend on the mesh quality
    - Their integration can be performed by standard FE quadrature
- Standard DG and IP techniques can be used to stabilize the formulation
- Preliminary results demonstrate optimal accuracy

More good news!

- A computationally efficient mimetic meshfree divergence exists!