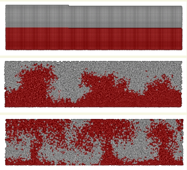
Introduction to Meshfree Methods

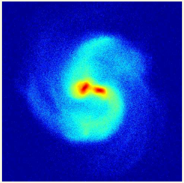
Implementation and Parallelization

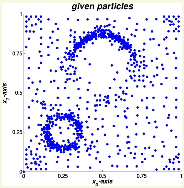
Marc Alexander Schweitzer

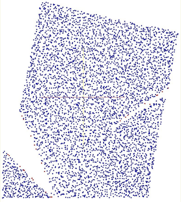
Institut für Numerische Simulation
Rheinische Friedrich–Wilhelms Universität Bonn

ASIM 2007, Aachen







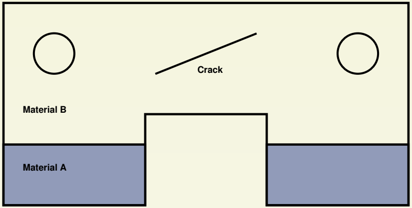


Outline

1. **Motivation**
   - What is a meshfree method?
   - When to use a meshfree method?
2. **Construction of meshfree methods**
   - Kernel Techniques
   - Moving Least Squares, partition of unity
   - Enrichment
3. **Computational challenges**
   - Appropriate data-structures
   - Parallelization
4. **Applications**

Efficient numerical simulation

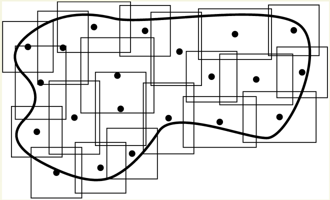
- Optimal complexity automatic algorithms.
  - Discretization with minimal degrees of freedom.
  - Efficient multilevel solver.
  - Load-balanced parallel implementation.
- Utilize a priori knowledge about solution.



  - Material discontinuities.
  - Geometry induced singularities (asymptotic expansions).
  - High resolution local simulations (homogenization, multi-scale information).
  - Spectral analysis.

What is a meshfree method?

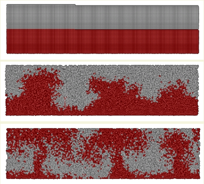
- Particle methods (physics).
  - Consider particles x_i.
  - Dynamics of multi-particle system.
  - Newtonian mechanics, system of ODEs.
- Scattered data approach (reconstruction).
  - Consider points x_i.
  - Choose/construct appropriate function space $\mathcal{V}$ on Ω based on $\mathcal{X}_N = \{x_i \mid i = 1, \ldots, N\} \subset \overline{\Omega}$.
  - Define appropriate energy functional.
  - Minimize energy over function space, system of PDEs.

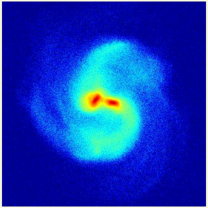


- Note only irregular point cloud assumed.
- Capability of h-adaptivity built-in.

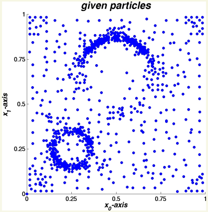
When to use a meshfree method?

- Mesh-generation.
  - Complex geometries / complicated structure in solution.
  - Time-dependent geometries / topological changes / large deformations.





- Higher order problems, i.e. global smoothness.
- Discontinuities and singularities.



Contruction of meshfree methods

- Choice of discretization technique.
  - Collocation.
  - Rayleigh–Ritz–Galerkin.
- Choice of basis functions.
  - Construction from scattered points only.
  - Meshfree zoo of acronyms:
    - SPH, CSPH, MLSPH, ...
    - RBF, WEBS, ...
    - EFGM, MLPG, RKPM, ...
    - GFEM, XFEM, PUFEM, ...
  - Common ingredients in many meshfree methods:
    - Partition of unity.
    - Enrichment basis.
- Three *separate* components.
  - Local approximability.
  - Inter-particle continuity.
  - Geometry resolution.

Special Kernel Techniques

Recovery Problem

Given $\mathcal{X}_N := \{(x_i, f_i) \mid i = 1, \ldots, N, x_i \in \overline{\Omega}\}$. Find $u : \Omega \to R$ such that

$$u(x_i) \approx f_i \quad \text{for all } i = 1, \ldots, N. \tag{1}$$

Smoothed Particle Hydrodynamics

- Convolution with δ-distribution

$$f(y) = \int_\Omega \delta_0(y - x) f(x) dx$$

- Convolution with *approximate* δ-distribution

$$f(y) \approx \int_\Omega \mathcal{W}(y - x) f(x) dx$$

- Discretization of integration

$$f(y) \approx \sum_{i=1}^{N} \alpha_i \mathcal{W}(y - x_i) f(x_i)$$

Abstract Kernel Techniques

Kernel Approximation

- A kernel is a function $K : \Omega \times \Omega \to \mathbb{R}$.
- Trial space

$$\mathcal{K} = \text{span}\langle K(\cdot, y), y \in \Omega\rangle$$

- Generalized interpolation, e.g. $K(x, y) = K(x - y)$

$$f_K(x) = \sum_{j=1}^{N} f_j K(x - x_j)$$

- Integral transformation

$$K_\Omega^* f(x) := \int_\Omega f(y) K(x, y) d\mu(x)$$

- Gaussian $\exp(-\|x - y\|^2)$, RBF $\Phi(\|x - y\|)$, splines, ...

Scattered Data Approximation

Recovery Problem

Given $\mathcal{X}_N := \{(x_i, f_i) \mid i = 1, \ldots, N, x_i \in \overline{\Omega}\}$. Find $u : \Omega \to R$ such that

$$u(x_i) \approx f_i \quad \text{for all } i = 1, \ldots, N. \tag{2}$$

Least Squares Fit

Consider the space $\mathcal{P}_k(\Omega)$ of all polynomials p with degree less than k. Minimize the quadratic functional

$$J_{\mathrm{LS}}(\pi) = \sum_{i=1}^{N} (f_i - \pi(x_i))^2 \tag{3}$$

over all polynomials $\pi \in \mathcal{P}_k(\Omega)$.

- Solution u is global polynomial.
- Approximation order determined by k.
- Increasing N does *not* improve quality.

Moving Least Squares Technique

Localized Weighted Least Squares Fit

Consider the space $\mathcal{P}_k(\Omega)$ and a set of weight functions[a]

$$W_i : \mathbb{R}^D \to \mathbb{R} \text{ with } \operatorname{supp}(W_i) = \omega_i.$$

Minimize the *pointwise* quadratic functional

$$J_{\mathrm{MLS}}(\pi)(x) = \sum_{i=1}^{N} W_i(x)(f_i - \pi(x_i))^2 \tag{4}$$

over all polynomials $\pi \in \mathcal{P}_k(\Omega)$.

[a] "Moving" refers to choice $W_i(x) = \mathcal{W}(x - x_i)$.

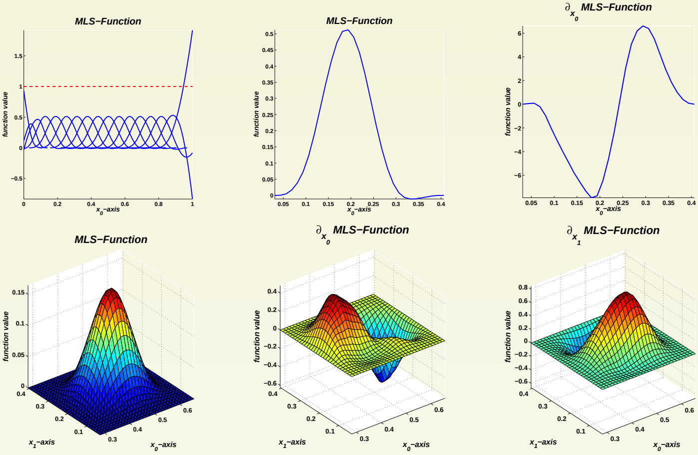
- Localized approximation.
- Approximation order is k.
- Increasing N *improves* quality.
- Smoothness inherited from weights.

- Solution is *not* a global polynomial.
- For each x^* there is $\pi \in \mathcal{P}_k(\Omega)$
  $u_{\mathrm{MLS}}(x^*) = \pi(x^*)$.
- There is a representation
  $u_{\mathrm{MLS}}(x) = \sum f_i \phi_i(x)$.

Representation

With particular basis $P = (p_q)$ and $G_{\mathrm{MLS}}(x^*)_{q,r} := \sum_{i=1}^{N} p_q(x_i) W_i(x^*) p_r(x_i)$

$$\phi_i(x) := W_i(x) P(x_i) \cdot (G_{\mathrm{MLS}}(x))^{-1} P(x).$$



Properties

- Locally supported basis functions ϕ_i

$$\operatorname{supp}(\phi_i) = \operatorname{supp}(W_i) = \omega_i.$$

- Basis known implicitly only.
- In general $\phi_i(x_j) \neq \delta_{i,j}$.
- Existence of $\mathcal{P}_k$-unisolvent subset in $\mathcal{X}_N \cap \omega_i$.
- Smoothness of ϕ_j, u determined by smoothness of all W_i.
- Polynomial basis is *globally* fixed.[1]
- Partition of unity

$$\sum_{i=1}^{N} \phi_i \equiv 1$$

independent of polynomial degree $k \geq 0$.

[1] Can be generalized to other *global* approximation space.

Decomposition of a Function

- Consider a general function $u \in H^s(\Omega)$

$$u = u_{\text{jump}}(u_{\text{smooth}} + u_{\text{singular}}) = H^u_{\text{jump}}(u_{\text{smooth}} + u_{\text{singular}})$$

- Employ approximation scheme with
  - higher order basis in smooth regions,
  - discontinuous basis across local jumps,
  - singular basis in vicinity of singularity.
- Consider an *arbitrary* partition of unity φ_i

$$u = \sum_{i=1}^{N} \varphi_i u = \sum_{i=1}^{N} H^u_{\text{jump}}(\varphi_i u_{\text{smooth}} + \varphi_i u_{\text{singular}})$$

A PU is the perfect glue!

- Define approximation space

$$V^{\text{PU}} := \sum_{i=1}^{N} \varphi_i V_i(\omega_i).$$

- $V_i = \mathcal{P}_{k(i)}$
- $V_i = H^u_{\text{jump}} \mathcal{P}_{k(i)}$
- $V_i = H^u_{\text{jump}}(\mathcal{P}_{k(i)} + \{r^{\alpha}_{\text{singular}}\})$

Enrichment of Partition of Unity

Error estimate [Babuška, Melenk]

Consider $u \in H^1(\Omega)$, $u_i \in V_i(\omega_i)$, $\{\varphi_i\}$ a PU on $\{\omega_i\}$, and $u^{\mathrm{PU}} := \sum_{i=1}^{N} \varphi_i u_i$

$$\|u - u^{\mathrm{PU}}\|_{L^2(\Omega)} \leq \sqrt{M} C_\infty \Big(\sum_{i=1}^{N} \|u - u_i\|^2_{L^2(\omega_i)} \Big)^{\frac{1}{2}}$$

$$\|\nabla(u - u^{\mathrm{PU}})\|_{L^2(\Omega)} \leq \sqrt{2M} \Big(\sum_{i=1}^{N} \big(\frac{C_\nabla}{\operatorname{diam}(\omega_i)}\big)^2 \|u - u_i\|^2_{L^2(\omega_i)} + C_\infty^2 \|\nabla(u - u_i)\|^2_{L^2(\omega_i)} \Big)^{\frac{1}{2}}$$

Constants M, C_∞, and C_∇ depend on PU only.

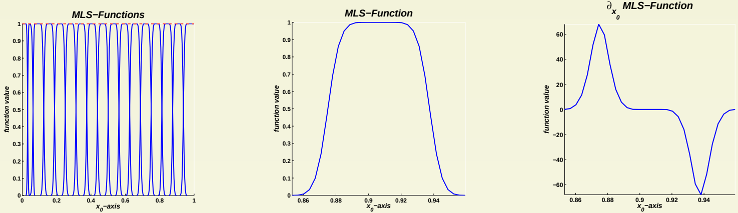
- External p-adaptivity (basis and degree).
- External local enrichment for singularities.
- External local jump enrichment for discontinuities.
- Problem-dependent local approximation spaces.
  - Regularity theory, spectral theory, asymptotic expansion, homogenization.
  - Numerical homogenization, microscale simulation, atomistic simulation.
- Resulting approximation functions

$$\varphi_i(x)(H_i(x) p_q(x)), \quad \text{and} \quad \varphi_i(x)(H_i(x) r^{\alpha_i})$$

- Basis property?

PARTICLE–PARTITION OF UNITY METHOD

- Stability of global basis holds if flat-top PU.



- Zero-order MLS = Shepard functions as PU.
- *Explicit* representation of PU.

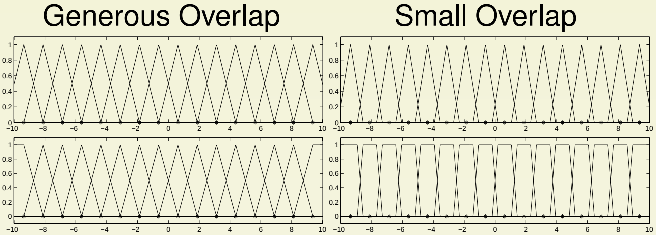
$$\varphi_i(x) = \frac{W_i(x)}{\sum_{k=1}^{N} W_k(x)} = \frac{W_i(x)}{\sum_{\omega_k \cap \omega_i \neq \emptyset} W_k(x)}$$

- $\mathcal{P}_0$-unisolvent iff $\Omega \subset \bigcup_{i=1}^{N} \omega_i$.
- *Small overlap* of patches ω_i for flat-top PU.

Particle–PUM Features

W_i

$\varphi_i := \frac{W_i}{\sum_j W_j}$



- Shepard PU $\{\varphi_i\}$ with spline weight functions W_i.
- Local approximation spaces.
  - Legendre poynomials $\psi_i^n, \ldots$
  - Point / edge singularities $\eta_i^m, \ldots$
- Assembled basis functions

$$V^{\mathrm{PU}} := \mathrm{span}\langle \varphi_i \psi_i^n, \varphi_i \eta_i^m \rangle$$

- Adaptive sub-division sparsegrid integration scheme.
- Automatic a-priori enrichment identification.
- A-posteriori sub-domain error estimator, hp-adaptivity.
- Multilevel solver and nested iteration.

Computational Challenges

- Construction of $\mathcal{C}_\Omega := \{\omega_i\}$ such that $\mathcal{P}_k(\Omega)$-unisolvent subset is contained in each ω_i, and $\overline{\Omega} \subset \bigcup_i \omega_i$.
- Search for all neighbors

$$\mathcal{N}_i := \{x_j \in \mathcal{X}_N \,|\, x_j \in \omega_i\}, \quad \mathcal{C}_i := \{\omega_j \in \mathcal{C}_\Omega \,|\, \omega_j \cap \omega_i \neq \emptyset\}.$$

- Integration of weak form.
  - Analytic integration not feasible.
  - Appropriate numerical quadrature (piecewise rational and singular integrands).
  - Domain approximation.
- Approximation of essential boundary conditions.
- Parallelization and dynamic load-balancing.
  - Irregular point clouds.
  - Varying local polynomial degrees.
  - Varying local enrichment (discontinuous, singular).

Essential Boundary Conditions

Model problem:

$$-\nabla \cdot \mu \nabla u + \nu u = f \text{ in } \Omega, \quad u = g_D \text{ on } \Gamma_D, \quad \nabla u \cdot n = g_N \text{ on } \Gamma_N$$

Approaches in meshfree methods:

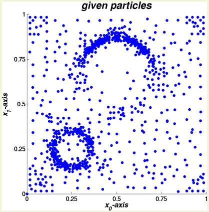
- Collocation.
- Penalty formulation.
- Lagrange multipliers.
- Nitsche's method.

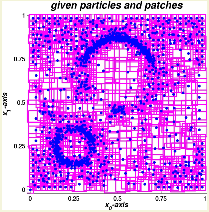
$$a(u,v) = \int_\Omega \mu \nabla u \cdot \nabla v + \nu uv - \int_{\Gamma_D} \mu (\nabla u \cdot n v + u \nabla v \cdot n) + \beta \int_{\Gamma_D} uv$$

$$l(v) = \int_\Omega fv + \int_{\Gamma_N} g_N v - \int_{\Gamma_D} \mu g_D \nabla v \cdot n + \beta \int_{\Gamma_D} g_D v$$

- Symmetric positive definite system.
- Optimal error bounds, provided inverse estimate holds.
- Regularization parameter β can be estimated efficiently.

Tasks & Data Structures



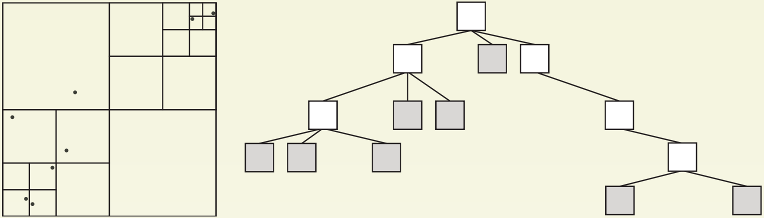


- Construction of a cover from points.
  - Delaunay triangulation, Voronoi cells.
  - Sub-division approach, trees.
    - Dimension-recursive construction.
    - Partition Ω in sub-domains $\mathcal{C}_i$ with simple shape.
    - Use a minimal number of sub-domains $\mathcal{C}_i$.
- Geometric search problem.
  - k-nearest neighbors.
  - Minimal trees (e.g. kd-trees).
  - Geometric trees (e.g. PR-trees).
    - Fast insert operations, adaptivity and dynamics.
    - Aspect ratios of cells.

Space Trees

Divison of Space by Quadtree, Octree, ...

- Top node / root is associated with complete domain.
- Non-leaf node splits region into 2^D equal sized sub-cells.
- Leaf node is associated with at most q number of points.



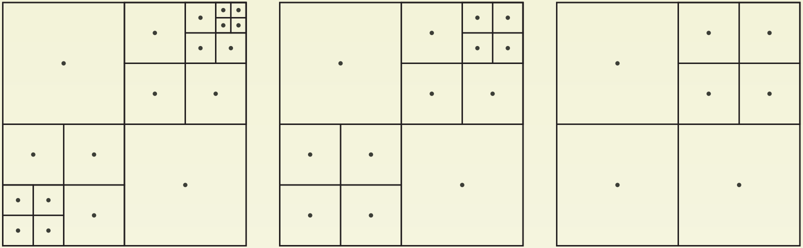
Mesh applications

- Mesh-generation.
- Parallel h-adaptive FEM.

Particle applications

- Multipol methods.
- Barnes–Hut method.
- Tree-SPH.

Geometric Hierarchy



$V_J^{\mathrm{PU}} = \sum \varphi_{i,J} V_{i,J}^{p_{i,J}}$ $\quad V_{J-1}^{\mathrm{PU}} = \sum \varphi_{i,J-1} V_{i,J-1}^{p_{i,J-1}}$ $\quad V_{J-2}^{\mathrm{PU}} = \sum \varphi_{i,J-2} V_{i,J-2}^{p_{i,J-2}}$

- Define patches $\omega_i = \alpha \mathcal{C}_i$ on leaf cells $\mathcal{C}_i$, $\alpha > 1$.
- Set weights W_i on patches ω_i, choose local space V_i on ω_i.

$$V_k^{\mathrm{PU}} := \sum_i \varphi_i V_i = \sum_i \frac{W_i}{\sum_m W_m} V_i$$

- Coarsen tree by removing appropriate subset of leafs.
- Refine tree using local estimate on patch ω_i using $\mathcal{V}_i \supset V_i$.
- No treatment of hanging nodes, i.e., arbitrary irregularity of tree.
- Spaces V_k^{PU} non-nested, i.e., $V_k^{\mathrm{PU}} \not\subset V_{k+1}^{\mathrm{PU}}$.
- Multilevel solver with appropriate interlevel transfers.

Tree Implementations

Pointer-based data structure

```
treenode {
datatype data;
treenode *successors[2^d];
...
}
```

- Store links to successors in tree-node.
- Topology explicitly given via graph of links.
- Parallel computation: Pointers to remote nodes?

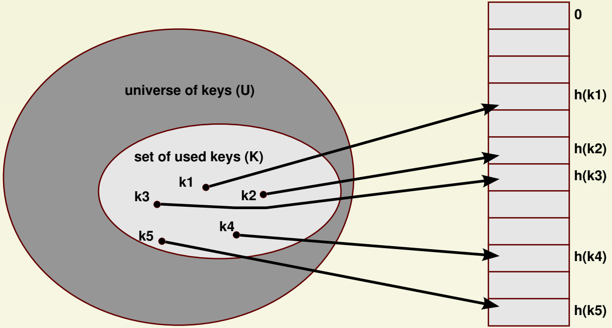
Key-based data structure

```
map<keytype, datatype> tree;
hash_map<keytype, datatype> tree;
```

- Without any explicit links between tree nodes.
- Topology implicitly encoded by key-labels.
- Store complete tree in (hashed) associative container.
- Parallelization by sub-division of key-range.

KEY-BASED MEMORY ACCESS

- Direct-address tables / ordinary arrays.
- Allocate memory for every possible key $O(\#U)$.
- Assumption universe of keys is *small*.



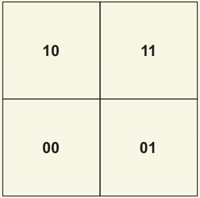
- Indirect-address tables / associative containers.
- Universe of keys is *large*.
- Allocate memory for $N = O(\#K)$ only.
- Hash-function $h : U \to [0, N-1]$.

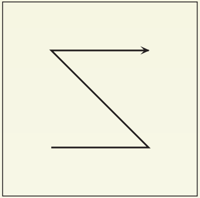
Key Generation

- Tree topology:
    - Parent/child information easily accessible.
- Tree node L corresponds to geometric cell $\mathcal{C}_L$.
- Cheap unique keys (small number of bits).

Construction of *path key*[2] k_L for cell $\mathcal{C}_L$:

- Assign initial key value $\mathrm{k}_L = 1$ at root cell.
- Concatenate key k_L with d binary descent decisions.
- Descend tree in direction of cell $\mathcal{C}_L$.





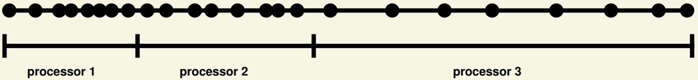
[2] Also known as bit interleaving.

Parallelization

- Data equi-distributed among processors (memory).
- Work load equi-distributed among processors (CPU).
- Number of neighboring processors minimal (latency).
- Boundary of data among processors minimal (bandwidth).

Consider one-dimensional problem.

- Data locations are contained in interval [0, 1].
- Data are fully *ordered*.
- Linear walk over data gives optimal partition.

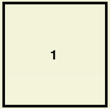


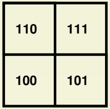
Load-Balancing

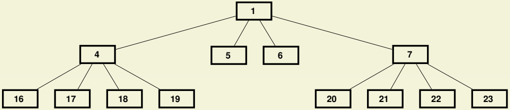
Memory load and work load estimate per tree-node.
Summation according to ordering of data.

Ordering data in higher dimensions? Use order in key range!

Parallelization of Tree







- Key range is one-dimensional.
- Simple sub-domain description via

$$0 = r_0 \leq r_1 \leq \cdots \leq r_{\wp} = \mathsf{k}_{\max}, \quad \Omega_q := \{\mathcal{C}_L \,|\, \mathsf{k}_L \in [r_q, r_{q+1})\}.$$

- Path keys induce *horizontal* order of tree.
  - Maps levels to processors (parallel traversal?).
  - All-to-all communication.
  - # boundary data $\approx$ # volume data.
- Transformation of keys to obtain *vertical* ordering.
  - Maps sub-trees to processors (local traversal!).
  - Small number of neighbors.
  - # boundary data $\ll$ # volume data.

Space-filling Curves

Space-filling Curve

A *space-filling curve* is the graph of a continuous *surjective* mapping

$$c : [0,1] \to \Omega \Subset \mathbb{R}^D$$

for Ω with $\mu_{\mathbb{R}^D}(\Omega) > 0$.

- There is no such injective mapping for smooth $\partial\Omega$.
- Iterative construction procedure for some SFC.

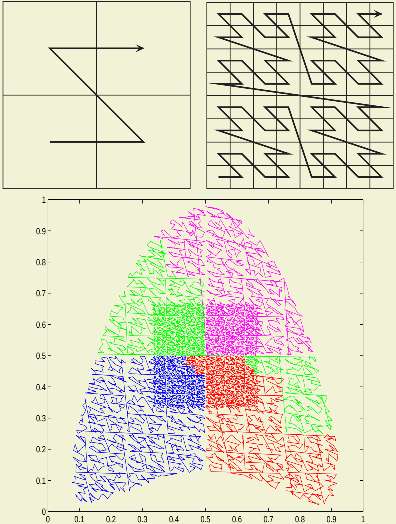
Discretizations of Space-filling Curve

The curves c_n associated with the nth iteration of a Lebesgue or Hilbert SFC construction are injective, i.e., self-avoiding.

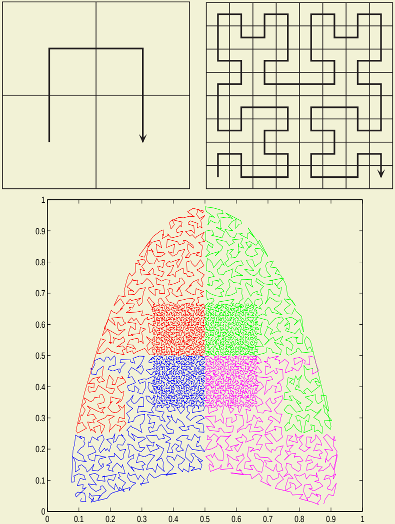
- Travelling sales man.
- Computer graphics.
- Cache optimization.
- Parallelization.

Space-filling Curve Partitioning

Lebesgue Curve



Hilbert Curve



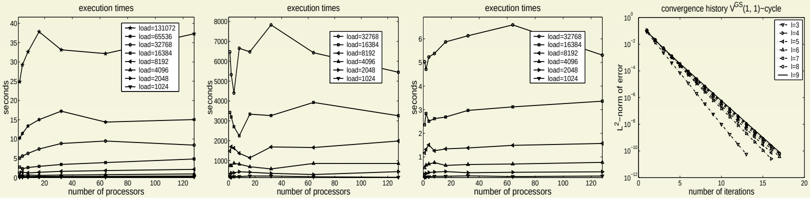
Performance of Particle–PUM

- Simple construction of $V^{\mathrm{PU}} = \mathrm{span}\langle \varphi_i \psi_i^n \rangle$: $O(N_0 \log N_0)$
- Efficient assembly $A\tilde{u} = \hat{f}$: $O(N(p^d + e)^2)$
- Multilevel solution: $O(N(p^d + e)^3)$
- Adaptive refinement: $O(N((p+1)^d + e)^3)$
- *Optimal* with respect to number of particles, almost optimal with respect to local approximation spaces.
- Highly flexible general purpose solver.
  - General particle input.
  - Choice of local spaces / enrichments.
  - Automatic refinement in h and p.
- Load-balanced parallel implementation / *optimal* scaling.

Parallel Performance

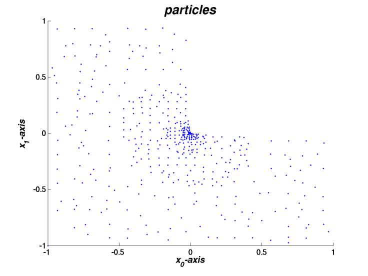
Model problem:

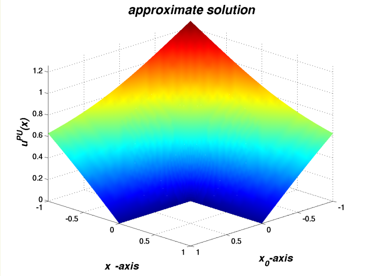
$$\int_{\Omega} \boldsymbol{\sigma}(u) : \boldsymbol{\epsilon}(u) + \int_{\Gamma_D} \beta u \cdot v - u \cdot (\boldsymbol{\sigma}(v) \cdot n) - (\boldsymbol{\sigma}(v) \cdot n) \cdot v$$



- Scaling of cover construction, neighbor search, and load-balancing step.
- Scaling of assembly of discrete linear system.
- Scaling of a $V - (1, 1)$-cycle.
- Convergence history of $V(1, 1)$-iteration.

SINGULARITIES IN TWO DIMENSIONS (H-ADAPTIVE)

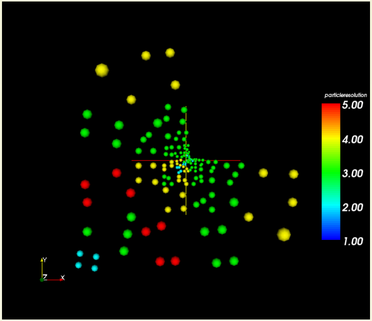




J	dof	e_{L^∞}	ρ_{L^∞}	e_{L^2}	ρ_{L^2}	e_{H^1}	ρ_{H^1}	$e^*_{H^1}$	$\rho^*_{H^1}$	$\epsilon^*_{H^1}$
14	17937	2.851_{-4}	1.02	5.109_{-5}	1.17	7.102_{-3}	0.54	5.047_{-3}	0.53	0.71
15	26235	1.796_{-4}	1.22	3.806_{-5}	0.77	5.863_{-3}	0.50	4.173_{-3}	0.50	0.71
16	41598	1.131_{-4}	1.00	2.404_{-5}	1.00	4.710_{-3}	0.47	3.347_{-3}	0.48	0.71
17	67266	7.126_{-5}	0.96	1.344_{-5}	1.21	3.654_{-3}	0.53	2.602_{-3}	0.52	0.71
18	99162	4.489_{-5}	1.19	9.895_{-6}	0.79	2.999_{-3}	0.51	2.138_{-3}	0.51	0.71
19	157779	2.828_{-5}	0.99	6.324_{-6}	0.96	2.410_{-3}	0.47	1.714_{-3}	0.48	0.71
20	259047	1.781_{-5}	0.93	3.465_{-6}	1.21	1.861_{-3}	0.52	1.325_{-3}	0.52	0.71
21	383805	1.122_{-5}	1.18	2.532_{-6}	0.80	1.521_{-3}	0.51	1.085_{-3}	0.51	0.71
22	612792	7.070_{-6}	0.99	1.621_{-6}	0.95	1.220_{-3}	0.47	8.686_{-4}	0.47	0.71
23	1014804	4.454_{-6}	0.92	8.828_{-7}	1.20	9.396_{-4}	0.52	6.695_{-4}	0.52	0.71
24	1509102	2.806_{-6}	1.16	6.403_{-7}	0.81	7.659_{-4}	0.51	5.465_{-4}	0.51	0.71
25	2412603	1.767_{-6}	0.98	4.109_{-7}	0.95	6.143_{-4}	0.47	4.375_{-4}	0.47	0.71
26	4014459	1.113_{-6}	0.91	2.230_{-7}	1.20	4.723_{-4}	0.52	3.366_{-4}	0.51	0.71
27	5983155	7.014_{-7}	1.16	1.610_{-7}	0.82	3.844_{-4}	0.52	2.743_{-4}	0.51	0.71
28	9575469	4.419_{-7}	0.98	1.034_{-7}	0.94	3.082_{-4}	0.47	2.195_{-4}	0.47	0.71
29	15969915	2.784_{-7}	0.90	5.600_{-8}	1.20	2.368_{-4}	0.52			

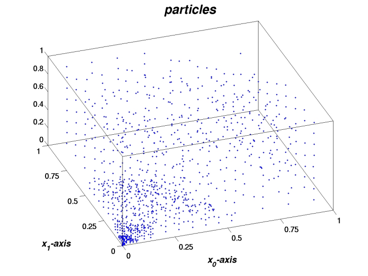
optimal rates: $\rho_{L^2} = 1 = \frac{2}{d}$ and $\rho_{H^1} = \frac{1}{2} = \frac{1}{d}$

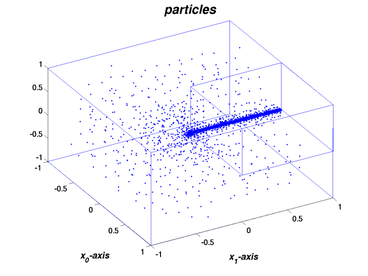
SINGULARITIES IN TWO DIMENSIONS (EHP-ADAPTIVE)



J	dof	e_{L^∞}	ρ_{L^∞}	e_{L^2}	ρ_{L^2}	e_{H^1}	ρ_{H^1}	$e^*_{H^1}$	$\rho^*_{H^1}$	$\epsilon^*_{H^1}$
2	30	7.824_{-2}	0.13	3.138_{-2}	0.53	1.407_{-1}	0.29	9.962_{-2}	0.45	0.71
3	57	4.255_{-2}	0.95	1.288_{-2}	1.39	1.059_{-1}	0.44	7.899_{-2}	0.36	0.75
4	105	2.678_{-2}	0.76	5.366_{-3}	1.43	7.193_{-2}	0.63	5.158_{-2}	0.70	0.72
5	156	1.686_{-2}	1.17	3.265_{-3}	1.25	4.846_{-2}	1.00	3.490_{-2}	0.99	0.72
6	218	9.969_{-3}	1.57	2.730_{-3}	0.54	3.230_{-2}	1.21	2.329_{-2}	1.21	0.72
7	280	6.248_{-3}	1.87	1.536_{-3}	2.30	2.192_{-2}	1.55	1.620_{-2}	1.45	0.74
8	358	3.883_{-3}	1.94	7.558_{-4}	2.88	1.477_{-2}	1.61	1.093_{-2}	1.60	0.74
9	454	2.446_{-3}	1.95	5.363_{-4}	1.44	9.885_{-3}	1.69	7.367_{-3}	1.66	0.75
10	612	1.503_{-3}	1.63	2.393_{-4}	2.70	6.540_{-3}	1.38	4.828_{-3}	1.41	0.74
11	852	9.326_{-4}	1.44	1.464_{-4}	1.49	4.373_{-3}	1.22	3.235_{-3}	1.21	0.74
12	1220	5.834_{-4}	1.31	5.753_{-5}	2.60	2.739_{-3}	1.30	2.010_{-3}	1.33	0.73
13	1664	3.675_{-4}	1.49	4.750_{-5}	0.62	1.812_{-3}	1.33	1.337_{-3}	1.31	0.74
14	2018	2.315_{-4}	2.40	2.671_{-5}	2.99	1.177_{-3}	2.24	8.683_{-4}	2.24	0.74
15	2369	1.458_{-4}	2.88	1.640_{-5}	3.04	7.914_{-4}	2.47	5.885_{-4}	2.43	0.74

Singularities in three dimensions (h-adaptive)

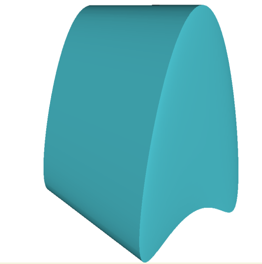


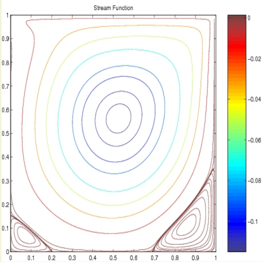


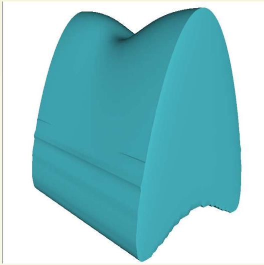
J	dof	$e_{L\infty}$	$\rho_{L\infty}$	e_{L2}	ρ_{L2}	e_{H1}	ρ_{H1}	e^*_{H1}	ρ^*_{H1}	ϵ^*_{H1}
1	32	1.424_{-1}	0.25	1.323_{-2}	0.49	1.702_{-1}	0.07	1.206_{-1}	0.14	0.71
2	256	1.124_{-1}	0.11	4.048_{-3}	0.57	1.007_{-1}	0.25	6.927_{-2}	0.27	0.69
3	480	8.913_{-2}	0.37	1.987_{-3}	1.13	7.291_{-2}	0.51	4.992_{-2}	0.52	0.68
4	704	7.074_{-2}	0.60	1.674_{-3}	0.45	6.183_{-2}	0.43	4.220_{-2}	0.44	0.68
5	928	5.614_{-2}	0.84	1.644_{-3}	0.07	5.792_{-2}	0.24	3.947_{-2}	0.24	0.68
6	2440	4.521_{-2}	0.22	8.566_{-4}	0.67	4.102_{-2}	0.36	2.750_{-2}	0.37	0.67
7	4540	3.537_{-2}	0.40	4.784_{-4}	0.94	3.229_{-2}	0.39	2.143_{-2}	0.40	0.66
8	7424	2.805_{-2}	0.47	3.924_{-4}	0.40	2.795_{-2}	0.29	1.847_{-2}	0.30	0.66
9	15964	2.226_{-2}	0.30	2.610_{-4}	0.53	2.178_{-2}	0.33	1.423_{-2}	0.34	0.65
10	30076	1.767_{-2}	0.36	1.426_{-4}	0.95	1.705_{-2}	0.39	1.110_{-2}	0.39	0.65
11	53148	1.402_{-2}	0.41	1.028_{-4}	0.57	1.425_{-2}	0.32	9.227_{-3}	0.33	0.65
12	110100	1.113_{-2}	0.32	7.111_{-5}	0.51	1.137_{-2}	0.31	7.314_{-3}	0.32	0.64
13	213840	8.830_{-3}	0.35	4.137_{-5}	0.82	8.885_{-3}	0.37	5.723_{-3}	0.37	0.64
14	372124	7.007_{-3}	0.42	2.792_{-5}	0.71	7.401_{-3}	0.33	4.752_{-3}	0.34	0.64

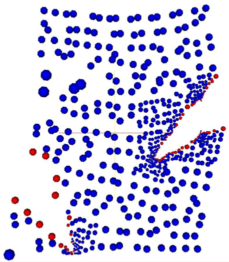
optimal rates: $\rho_{L2} = \frac{2}{3} = \frac{2}{d}$ and $\rho_{H1} = \frac{1}{3} = \frac{1}{d}$

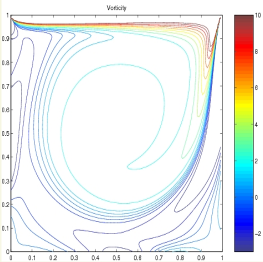
Continuum Mechanics





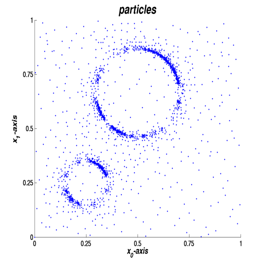


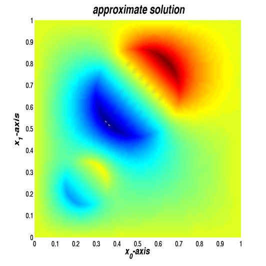




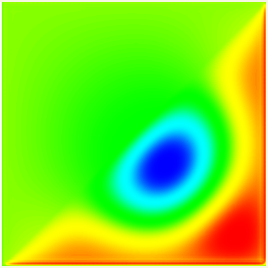
MAGNETOSTATICS AND PHOTONIC CRYSTALS

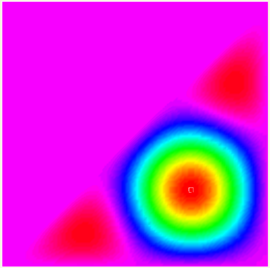
$$-\nabla \cdot \mu \nabla u = \nabla \cdot (\mu M)$$





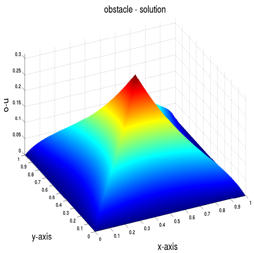
$$-\nabla \cdot \mu \nabla u - k^2 u = 0$$

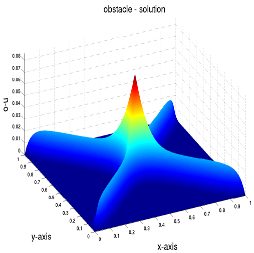




Obstacle problem

$$
\begin{aligned}
-\Delta u &\leq f && \text{in } \Omega = [0,1]^2, & u \leq o &= \mathrm{dist}_\Omega && \text{in } \Omega,\\
u &= 0 && \text{on } \partial\Omega, & (-\Delta u - f)(u - o) &= 0 && \text{in } \Omega,
\end{aligned}
$$





- Variational inequalities, minimization in *cone* of valid functions.
- Coarse local approximation spaces such that fine constraints satisfied.

Summary

- Meshfree methods.
  - Many different names, many common ingredients.
  - Partition of unity and enrichment.
  - Fluid dynamics, structural mechanics, multi-scale phenomena.
- Data-structures.
  - Key-based tree implementation.
- Dynamic load-balancing.
  - Space-filling curves.

4th International Workshop on Meshfree Methods for PDE, 17.–20. September 2007, Bonn, Germany.
`http://wissrech.ins.uni-bonn.de/meshfree`

- Coupling atomistic to continuum models.
- Multiscale simulation techniques.