Random Meshes for Carbon Sequestration

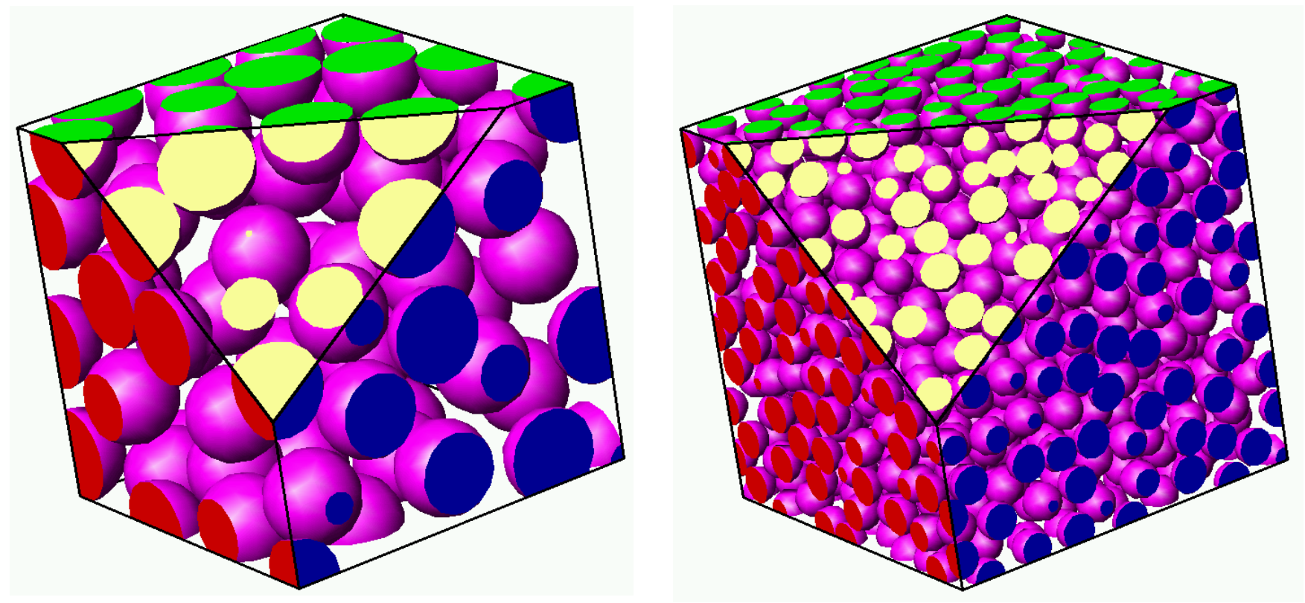
20th IMR, Paris, France, 2011.

Sandia National Laboratories

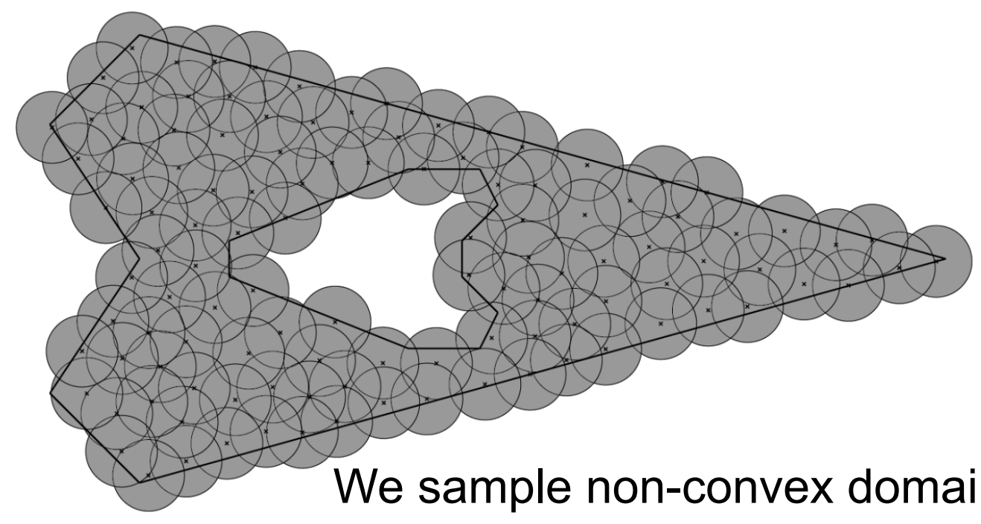
Mohamed S. Ebeida, Scott A. Mitchell, Patrick M. Knupp, Vitus J. Leung, Joseph E. Bishop, Mario J. Martinez

UC DAVIS UNIVERSITY OF CALIFORNIA

Anjul Patney, Andrew A. Davidson, John D. Owens



Unit cubes with non-overlapping $r/2$ spheres.



We sample non-convex domains, but many others can only sample cubes.

1. Maximal Poisson-Disk Sampling (MPS)

MPS selects random points $\{x_i\} = X$, from a domain, D. There is an exclusion/inclusion radius r: no two sample points are closer than r to one another; and every location is within r of a sample. The probability P of selecting the next point x_i from a disk-free region Ω is proportional to Ω's area. These properties lead to Delaunay meshes with uniformly random edge orientations.

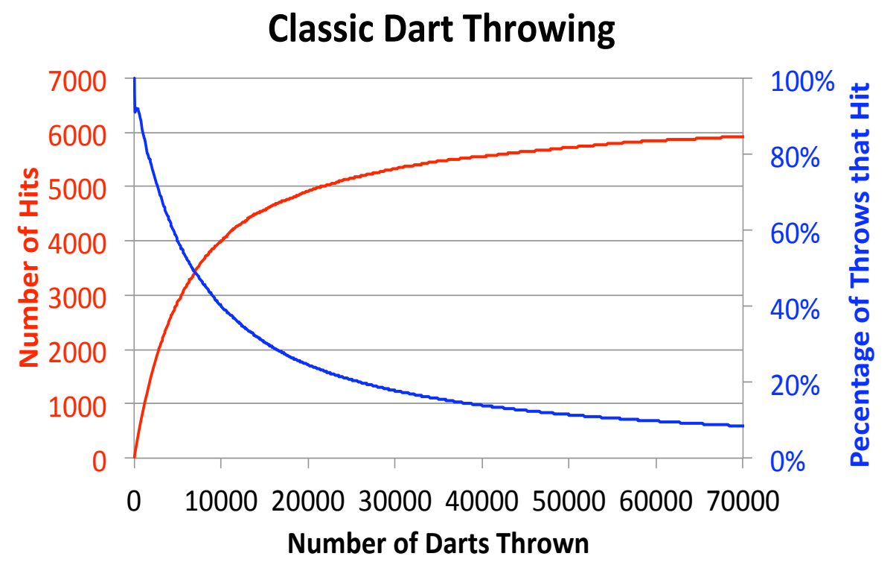
$$\text{Bias-free: } \forall \Omega \subset \mathcal{D}_{i-1} : P(x_i \in \Omega) = \frac{\text{Area}(\Omega)}{\text{Area}(\mathcal{D}_{i-1})} \quad (1a)$$

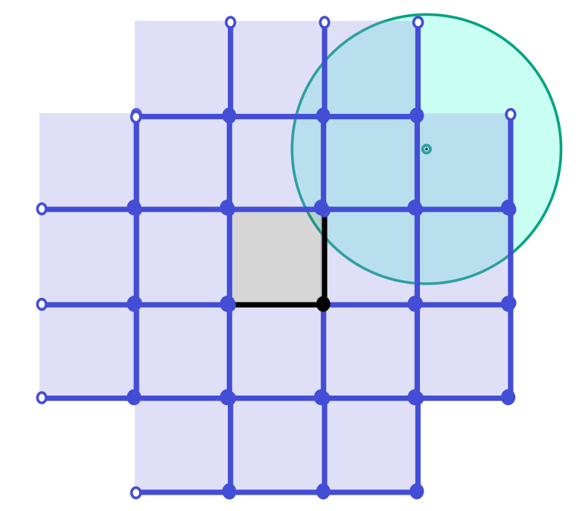
$$\text{Empty disk: } \forall x_i, x_j \in X, i \neq j : ||x_i - x_j|| \geq r \quad (1b)$$

$$\text{Maximal: } \forall p \in \mathcal{D}, \exists x_i \in X : ||p - x_i|| < r \quad (1c)$$

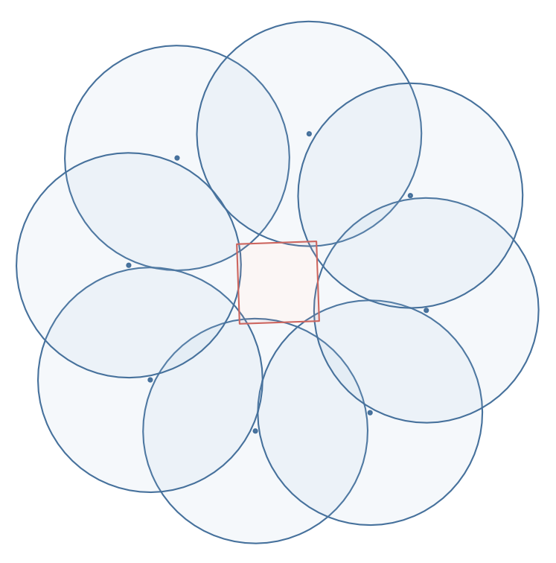
A maximal r-disk sample (1b) (1c) is equivalent to a maximal sample of non-overlapping $r/2$-disks, known as a random close sphere packing. They appear frequently in nature: e.g. sand, atoms in a liquid, trees in a forest. Processes generating them include random sequential adsorption in chemistry, and the hard-core Gibbs process in statistics.

But it is difficult to mimic these natural or statistical processes with an efficient algorithm. The classic approach is *dart-throwing*: successively generate an unbiased point (dart) and reject it if it is too close to a prior point. It is pretty easy to get close to maximality, but placing darts in the last few uncovered voids may take nearly forever.

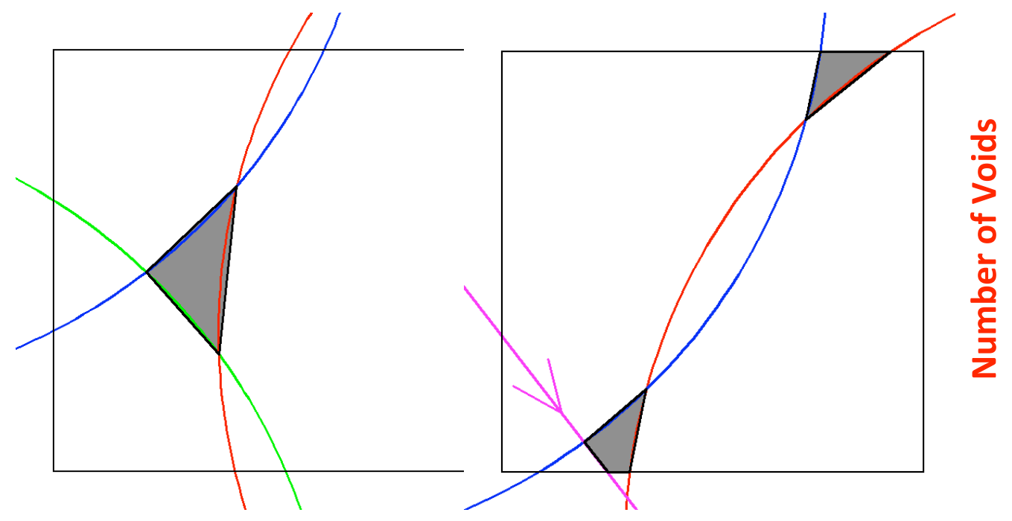


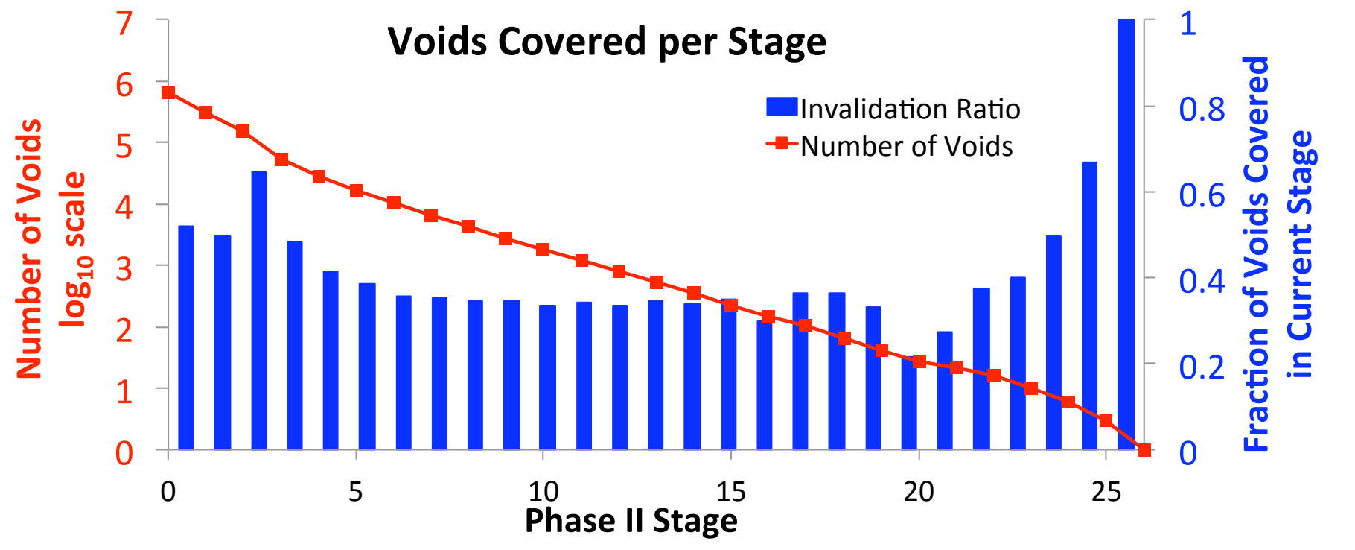


In 2011, we published two methods [1,2] to solve this problem. They both use a background grid of squares to guide the process. The disks for a void (uncovered area) are local, and there are less than 10 of them.

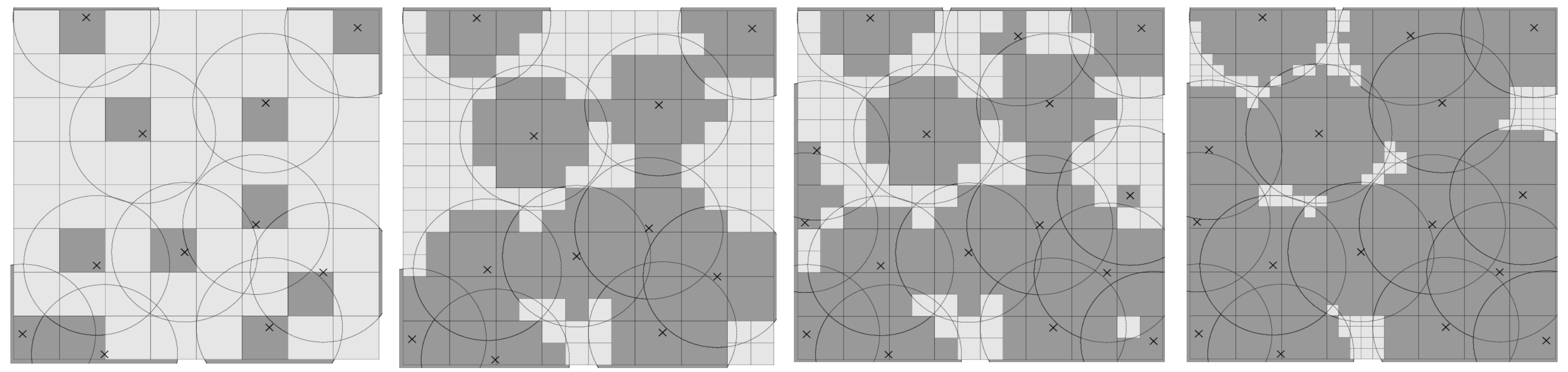


[1]. We gave the first optimal MPS algorithm satisfying all the criteria, in 2d. It constructs polygonal outer approximations to the remaining uncovered voids. To be bias free, our sampling process chooses from these polygons based on their area. Polygons are sufficiently close to the arc-gon voids, so that a constant fraction of the generated points are disk-free. Only *O(n)* dart throws are needed. It takes *O(n* log *n)* expected time and *O(n)* deterministic memory.





[2]. Our second method is simpler and more memory efficient, especially for higher dimensions *d*. Instead of creating polygons, we use a flat quadtree to successively refine our approximation to the uncovered domain. This is very efficient, since cells are described by *d* indices and one global level: no tree, no pointers, no geometry. Prior approaches refine the quadtree locally, but this is not needed. Our method works in any dimension and has empirical time and memory complexity $O(n)$.



Iteration 0 end

Iteration 1 start

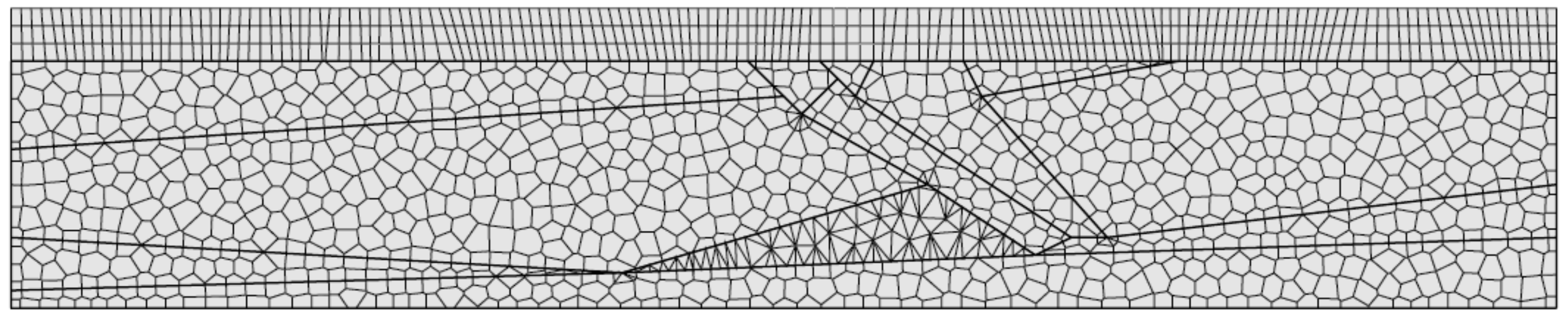
Iteration 1 end

Iteration 2 start

Summary: Our new meshing paradigm is based on random sampling. Underground storage sites for carbon sequestration might fracture when receiving high pressure gases. When simulating this, we prefer random meshes because they produce more realistic cracks. We create hybrid meshes that conform to naturally occurring complex structures, such as long, thin layers with twisting, faulting, and pinch-offs.

Mesh points are from a maximal Poisson-disk sample. The boundary of the domain is sampled differently for primal (simplicial) and dual (Voronoi polyhedral) meshes. A polyhedral mesh is body-fitted, but its dual is not. Mesh elements have good quality.

Except for some minor log n operations when generating unbiased points, every step is local and can be done in constant time, leading to O(n) time and memory. The locality and fixed size of the steps facilitate scalability and also GPU implementations. Our point codes take less memory and time than the alternatives, and triangulating the points is competitive with Triangle.

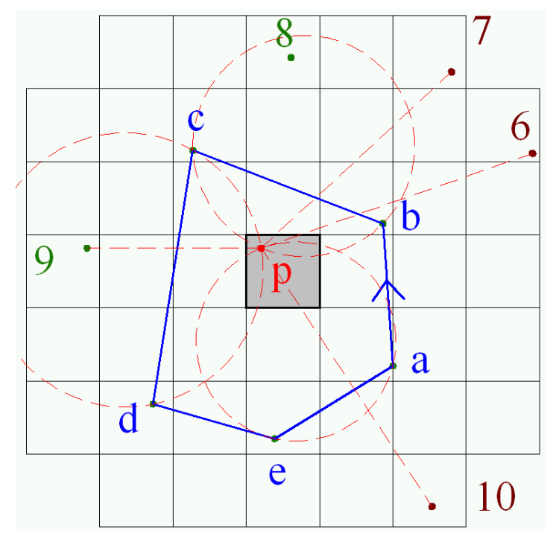


2. Hybrid Meshing

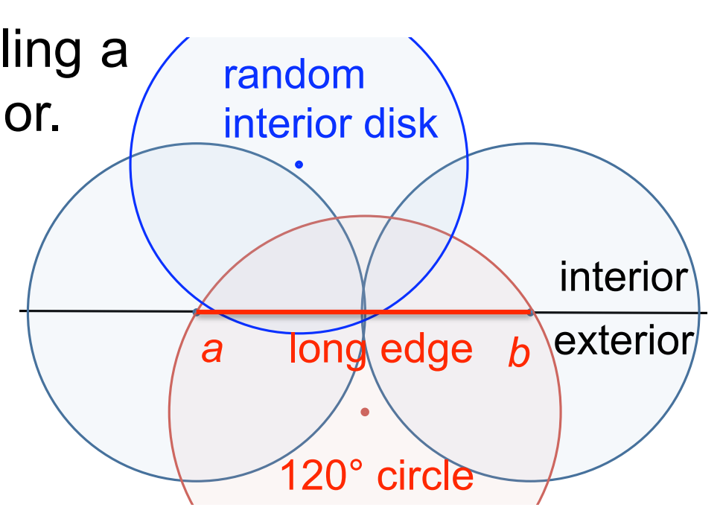
The hybrid mesher "glues together" different meshing algorithms. We have triangle, quadrilateral, and polyhedron meshes. An algebraic method generates structured quad meshes; otherwise points are generated using Poisson-disk samples. Meshes are conformal, with hanging nodes between primal and Voronoi regions. The MeshingGenie software will be released openly with Trilinos [6].

3. Constrained Delaunay Triangulation (CDT)

We [3] insert points to produce an unbiased maximal sampling, and provably get good quality triangles as a byproduct. In contrast Delaunay refinement algorithms insert deterministic points to remove poor-quality triangles, and get a biased maximal sampling as a byproduct.

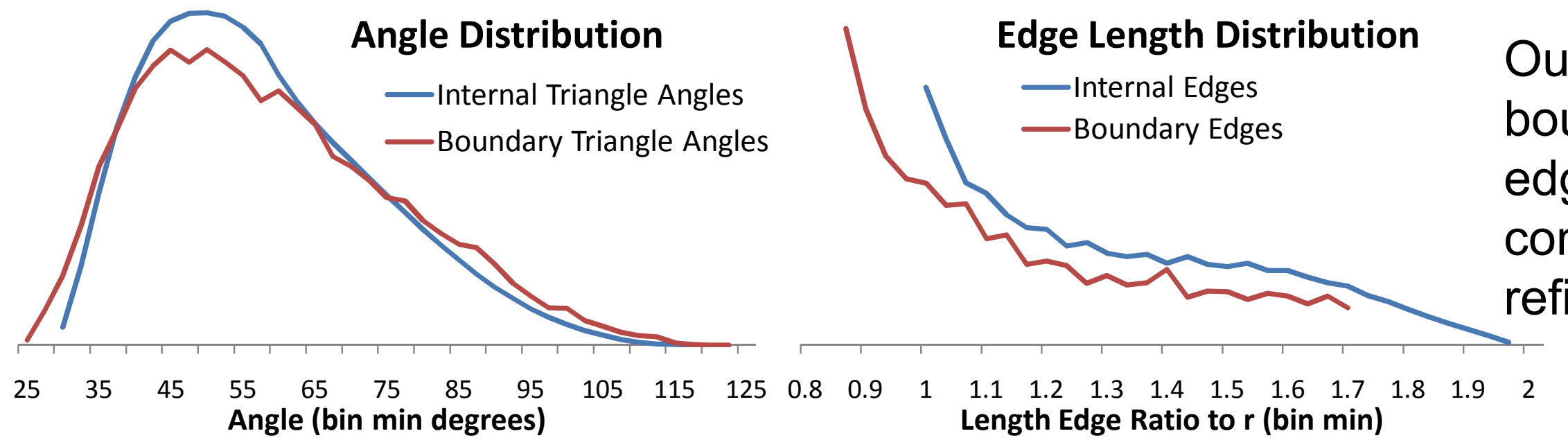


We protect the boundary by sampling a neighborhood of it before the interior.

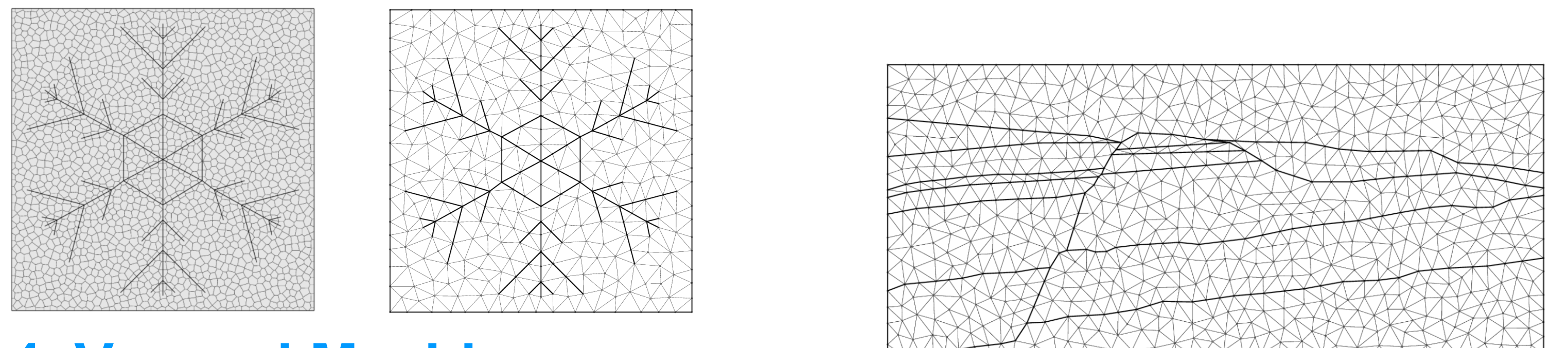


The CDT of a point is constructed in constant time: use the background grid to find potential neighbors, then use angular sorting and empty-circle checks.

We iterate over each sample point. Communication between the triangulations for different points is not required where the CDT is unique. The algorithm is *O(n)* in time and space.



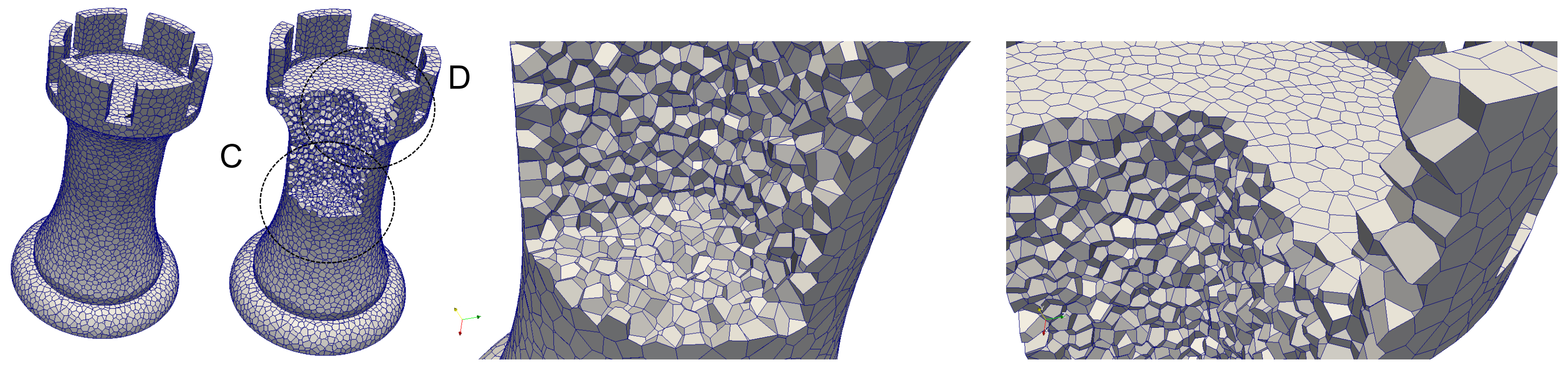
Our provable and empirical bounds on angles and edge lengths are comparable to Delaunay refinement methods.



4. Voronoi Meshing

Our polyhedral mesh [5] is **not** the dual of a body-fitted simplicial mesh! This is because we do not sample the boundary, except near reflex features. This is better because it gives random dihedrals at the boundary, leading to more realistic fracture simulations. The final mesh has bounded facet dihedrals and polyhedron aspect ratios.

To construct a Voronoi cell, the background grid is very helpful, as it was for the CDT: gather all the nearby points in constant time; successively cut a bounding box by the Voronoi planes of the nearby points. The algorithm is *O(n)* in time and space, and works in 3d.

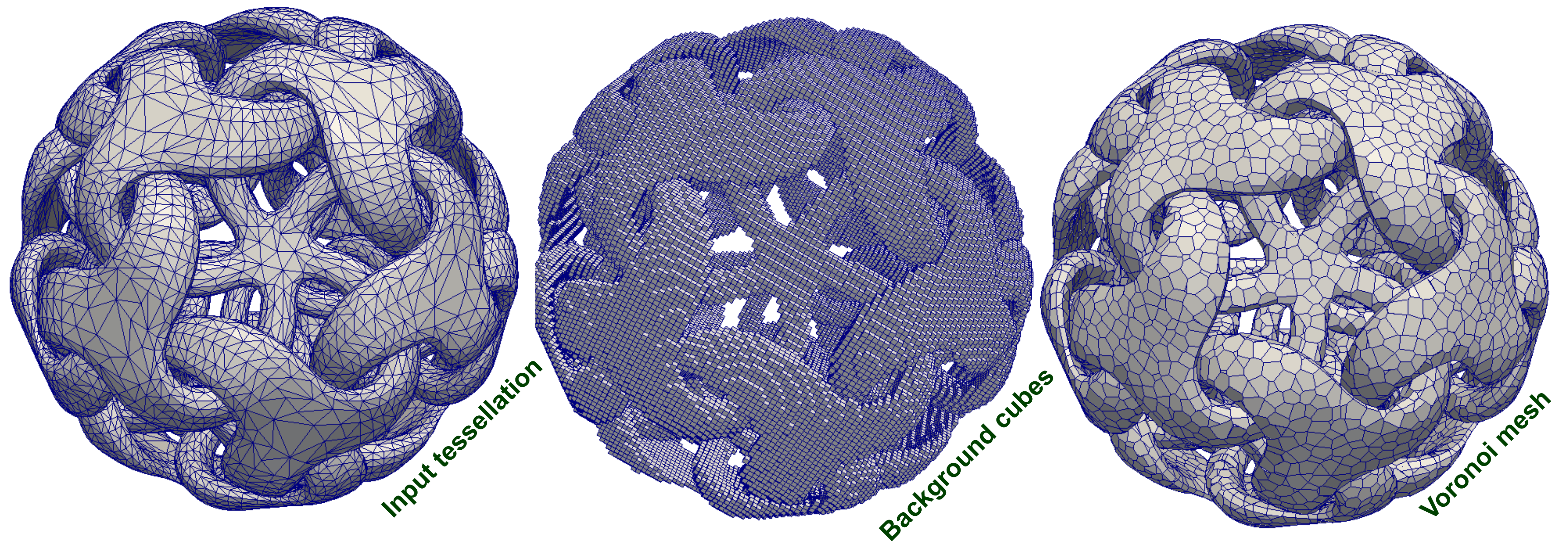


A. Mesh

B. Clipped mesh

C. Zoom-in

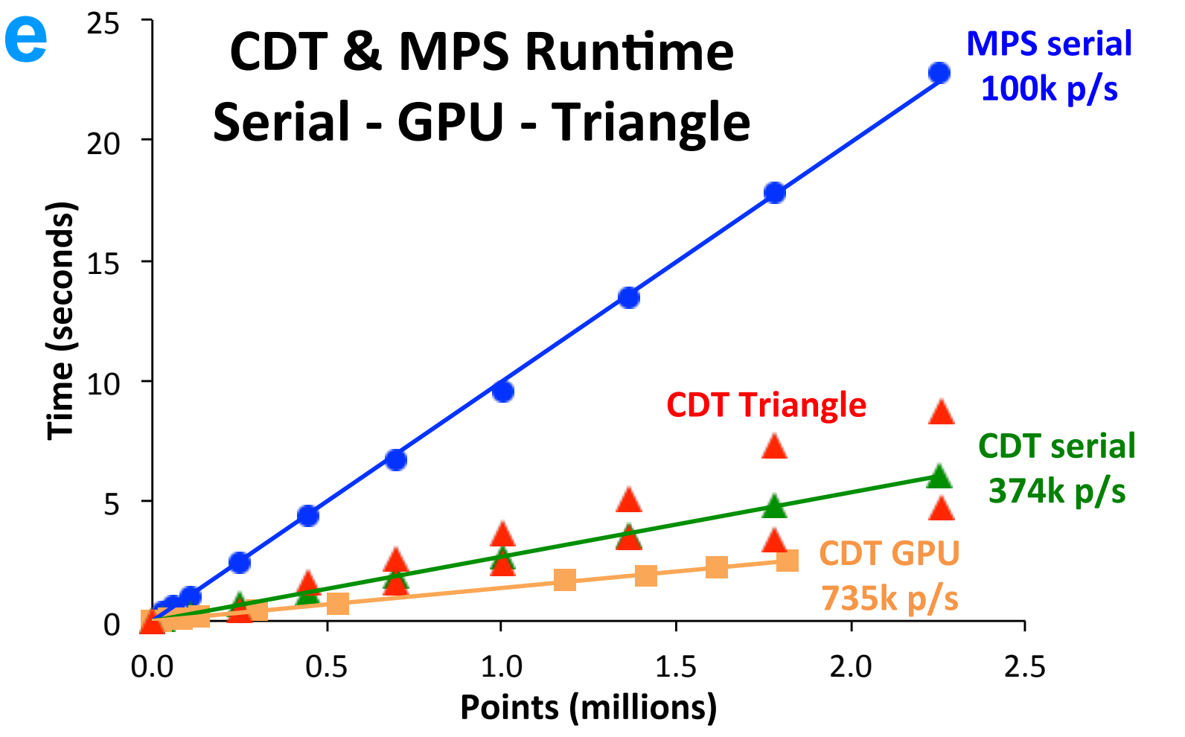
D. Zoom-in



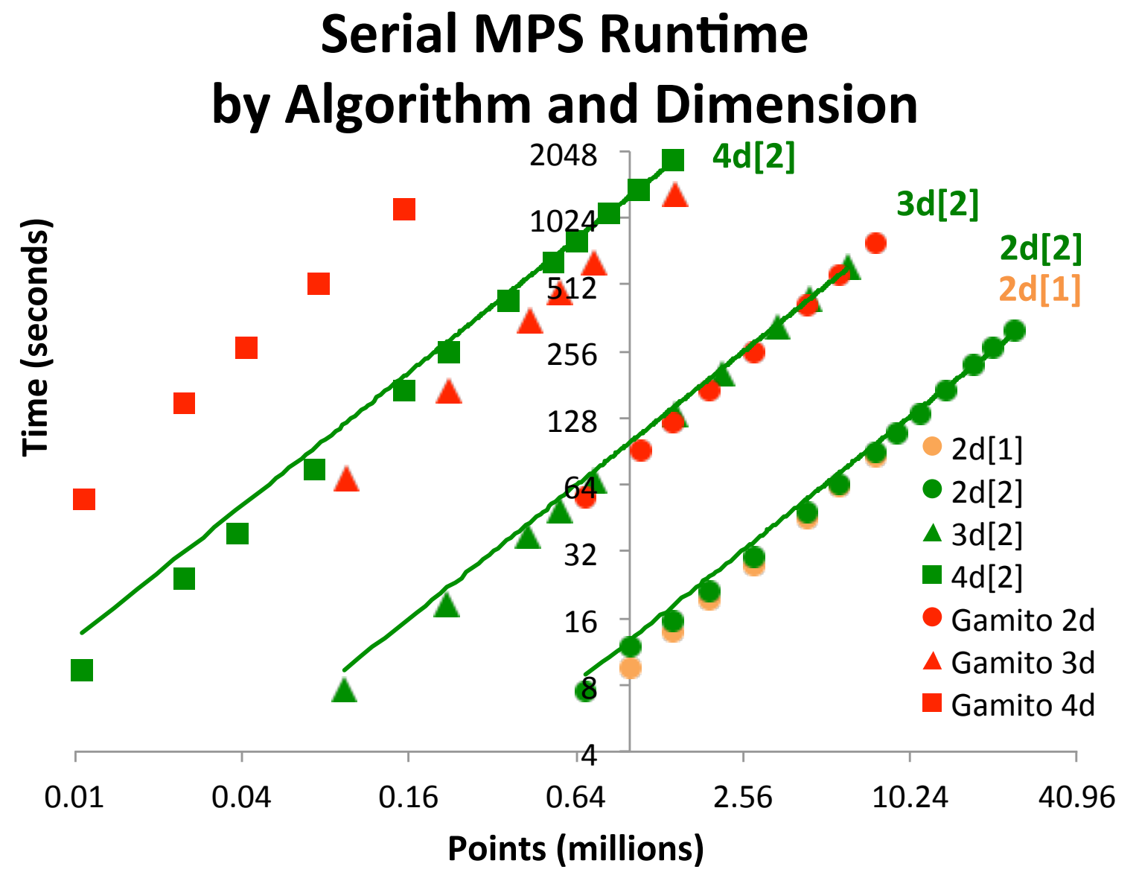
5. GPU Codes and Performance

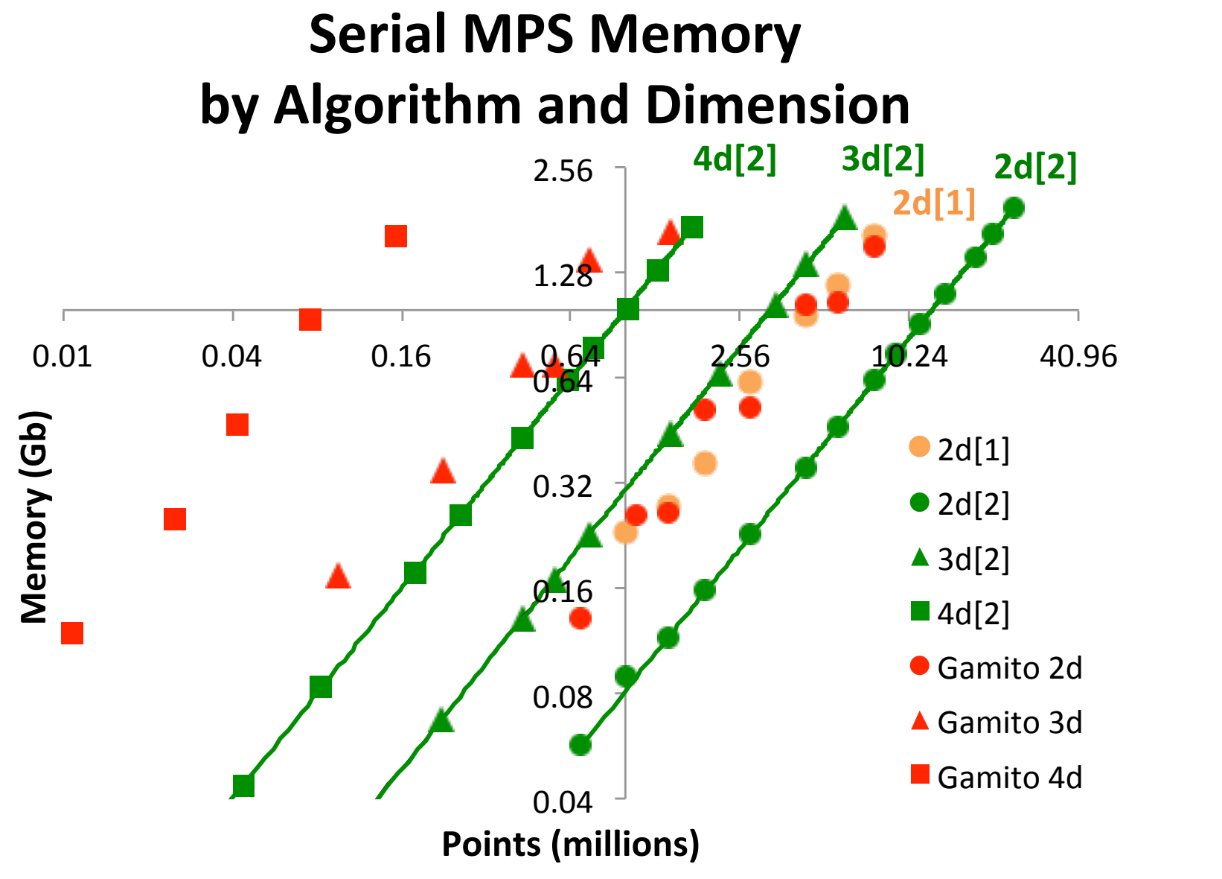
Our serial CDT code [3] is close to the speed of the popular Triangle software. (But not for generating points.) Our GPU code is 2x faster.

Our GPU code uses a large pool of threads to generate points in parallel. To remain unbiased, ties between points with overlapping disks are broken using thread ids. Memory bandwidth and capacity are the limiting factors.



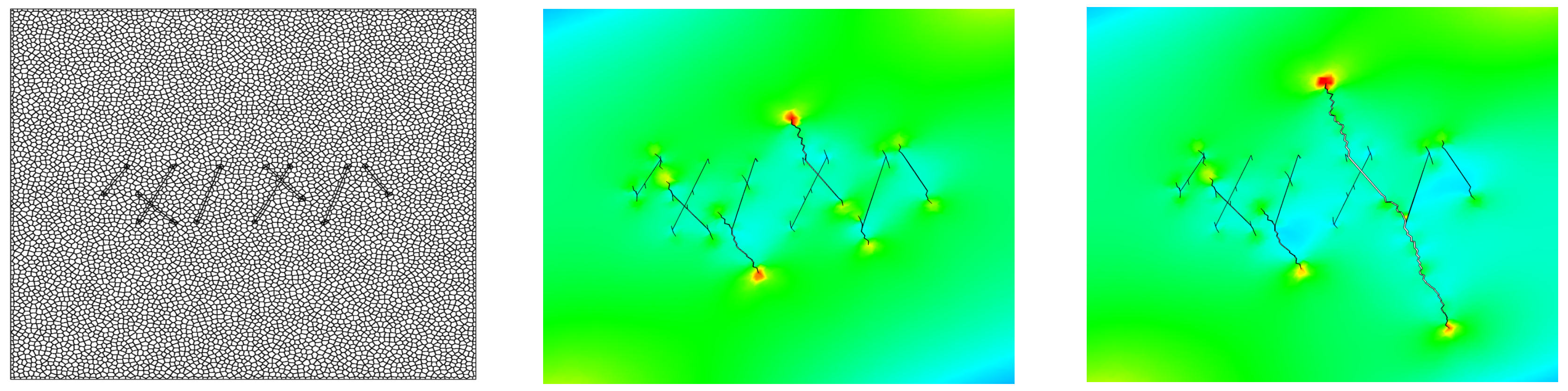
Discrete algorithms are notoriously difficult to parallelize, due to random memory access patterns. Nonetheless, our GPU code beats the serial one by a factor of ten for generating points, and a factor of two for triangulating them. Our simpler serial algorithm [2] is faster and takes less memory than the best alternative, by Gamito.





6. Caprock Fracture

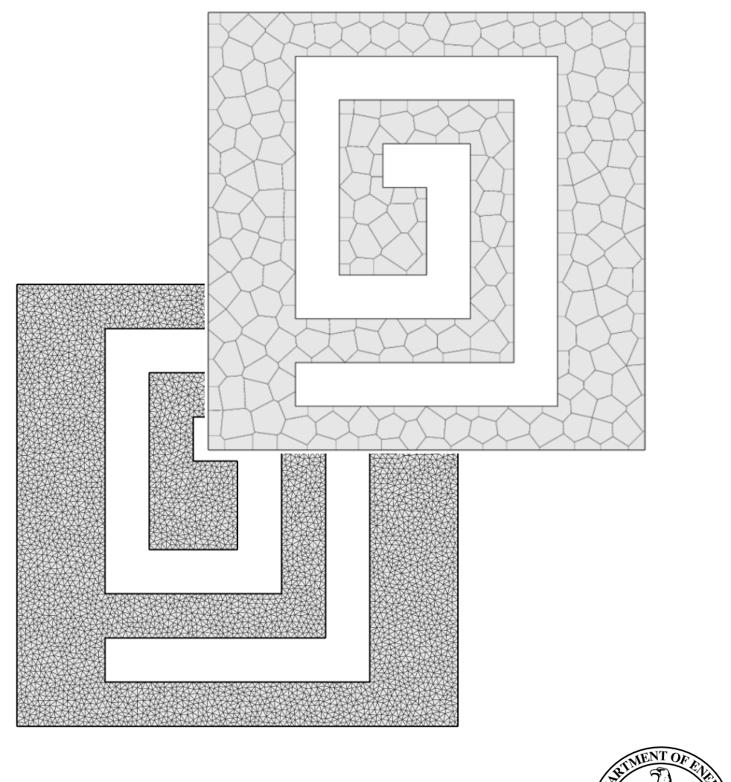
We [4] simulated the injection of CO_2 below a sub-scale model of a caprock layer. We used a Voronoi mesh. The initial fractures represent joints that were sealed, but reactivated by the injection into the reservoir below the caprock. Crack nucleation and growth derives from a limit surface of the allowable stress states. A cohesive law decays as the crack opens. The mesh randomness models epistemic uncertainty in the material strength. An ensemble of meshes predicts a range of plausible outcomes.



Snapshots over time. Color represents maximum principal stress, with red being the most extreme.

Publications

1. Mohamed S. Ebeida, Anjul Patney, Scott A. Mitchell, Andrew A. Davidson, Patrick M. Knupp and John D. Owens, "Efficient maximal Poisson-disk sampling," SIGGRAPH2011, August 7-11, Vancouver, Canada, SAND2011 - 0260C.
2. Mohamed S. Ebeida, Scott A. Mitchell, Anjul Patney, Andrew A. Davidson, and John D. Owens, "A simpler algorithm for bigger maximal Poisson-disk samples in higher dimensions," SAND2011 - 3606C. (under review)
3. Mohamed S. Ebeida, Scott A. Mitchell, Andrew A. Davidson, Anjul Patney, Patrick M. Knupp and John D. Owens, "Efficient and good Delaunay meshes from random points," SIAM conference on Geometric and Physical Modeling (GD/SPM11), October 24-27, 2011, Orlando, Florida, SAND2011 - 0519C.
4. Mohamed S. Ebeida, Patrick M. Knupp, Vitus J. Leung, Joseph E. Bishop, and Mario J. Martinez, "Mesh generation for modeling and simulation of carbon sequestration process," SciDAC2011, July 10-14, 2011, Denver, CO, SAND2011 - 3771A.
5. Mohamed S. Ebeida, and Scott A. Mitchell, "Uniform random Voronoi meshes," 20th International meshing roundtable, October 23-26, Paris, France 2011.
6. Mohamed S. Ebeida and Patrick M. Knupp, "LBMD: A Layer-based Mesh Data structure Tailored for Generic API Infrastructures", 20th AIAA Computational Fluid Dynamics Conference, June 27-30, Honolulu, Hawaii, SAND2010 - 8006C.



Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND2011-7609P.