

Analysis-ready domain models of neuronal forests

John Edwards

Department of Computer Science
Computational Visualization Center
Institute of Computational Engineering and Sciences
University of Texas at Austin

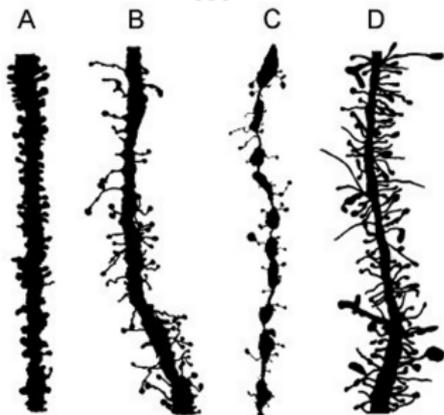
Thesis Proposal, December 13, 2011

- 1 Motivating example and problem statements
- 2 Problem 1 - surface mesh domain models
- 3 Problem 2 - micro-scale multi-compartment model
- 4 Problem 3 - nano-scale FEM model
- 5 Moving forward

- 1 Motivating example and problem statements
- 2 Problem 1 - surface mesh domain models
- 3 Problem 2 - micro-scale multi-compartment model
- 4 Problem 3 - nano-scale FEM model
- 5 Moving forward



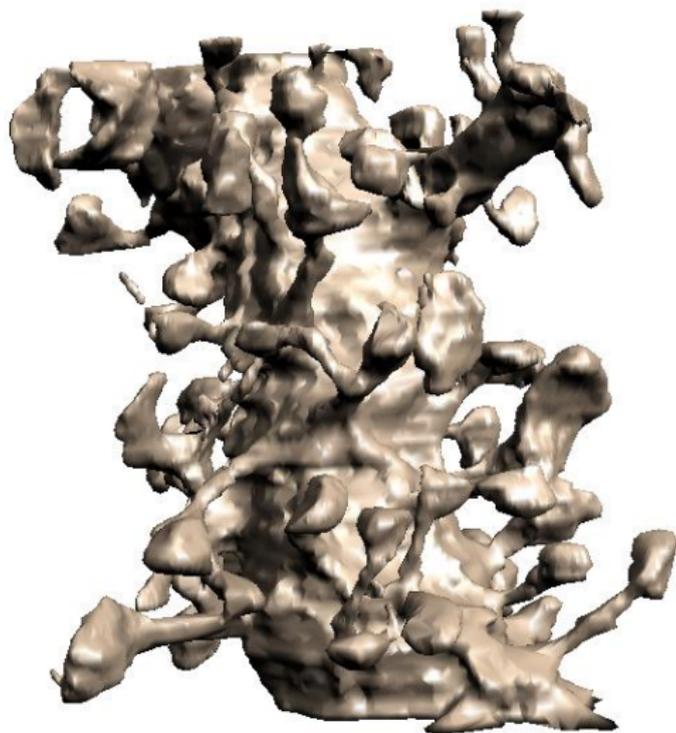
<http://www.homepages.ucl.ac.uk/~sjjgnle/>

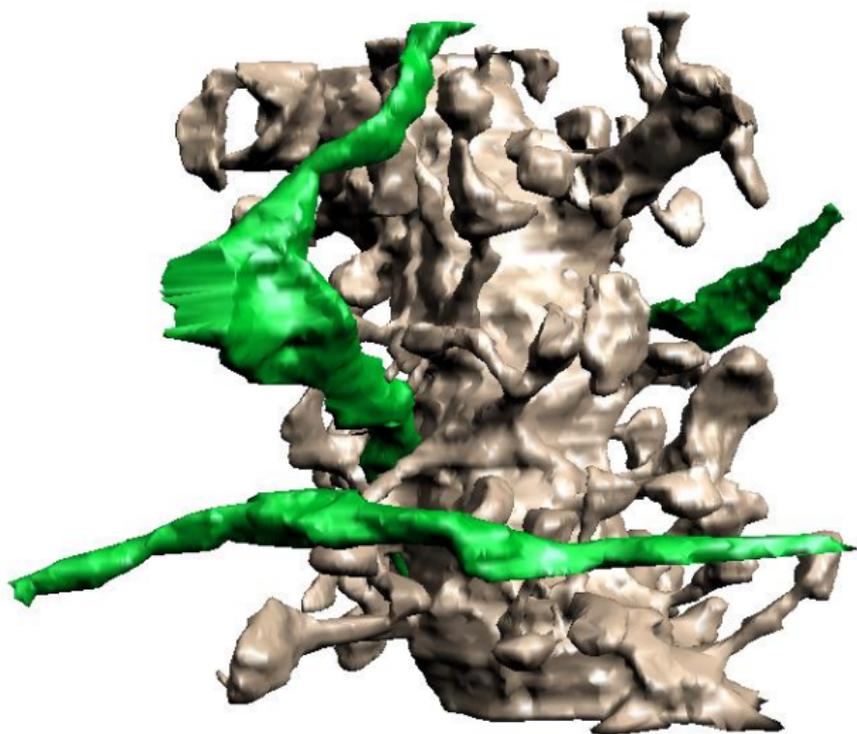


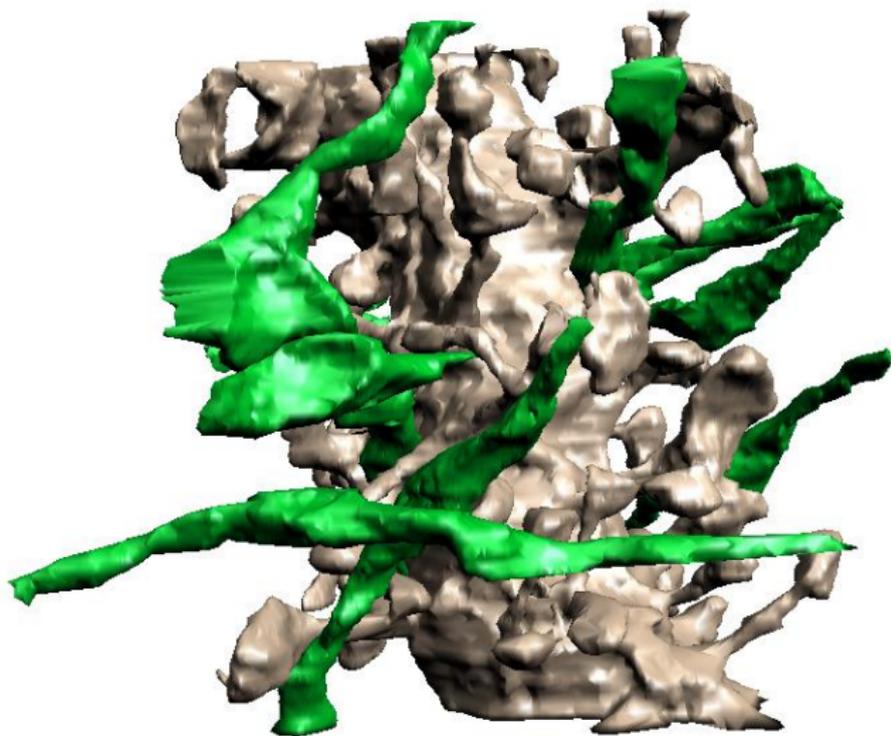
[Fiala et al., 2002]

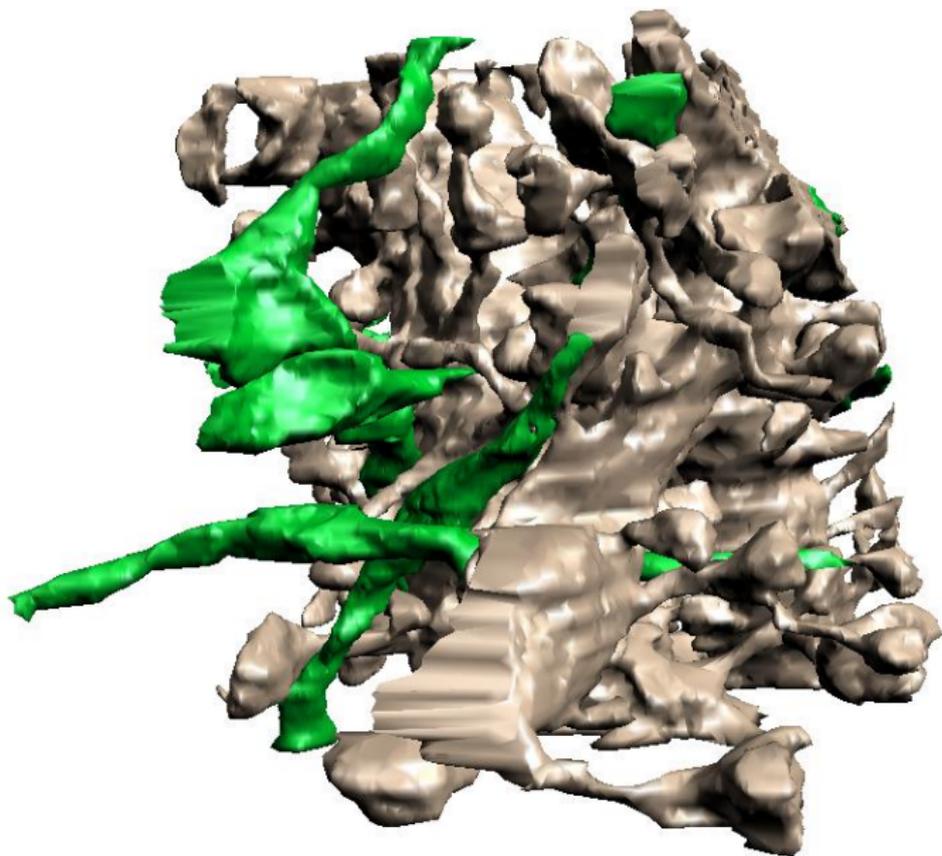
Edwards (Univ. of Texas)

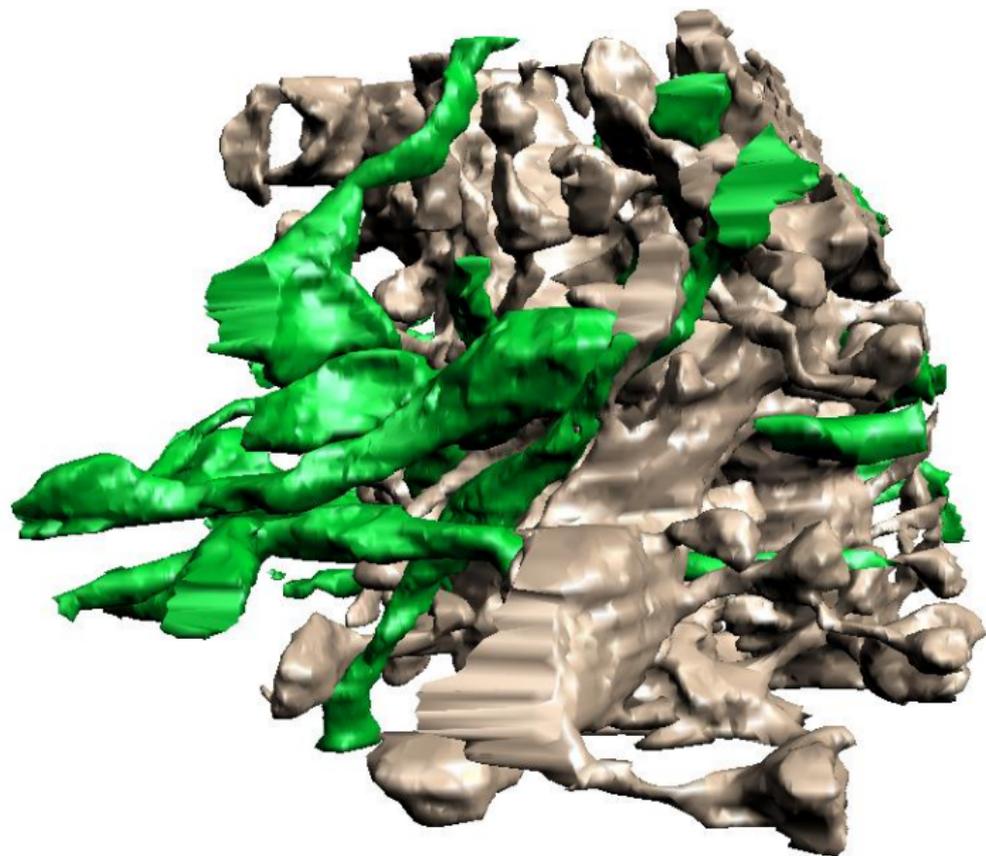
- Geometries play a role:
 - A Neurologically normal
 - B Mentally disabled
 - C Severe neurobehavioral failure
 - D Fragile X syndrome
- Electrophysiological simulations of action potentials elucidate structure-function relationships between geometries and
 - neuronal topology and combinatorics
 - learning, behavior, and memory

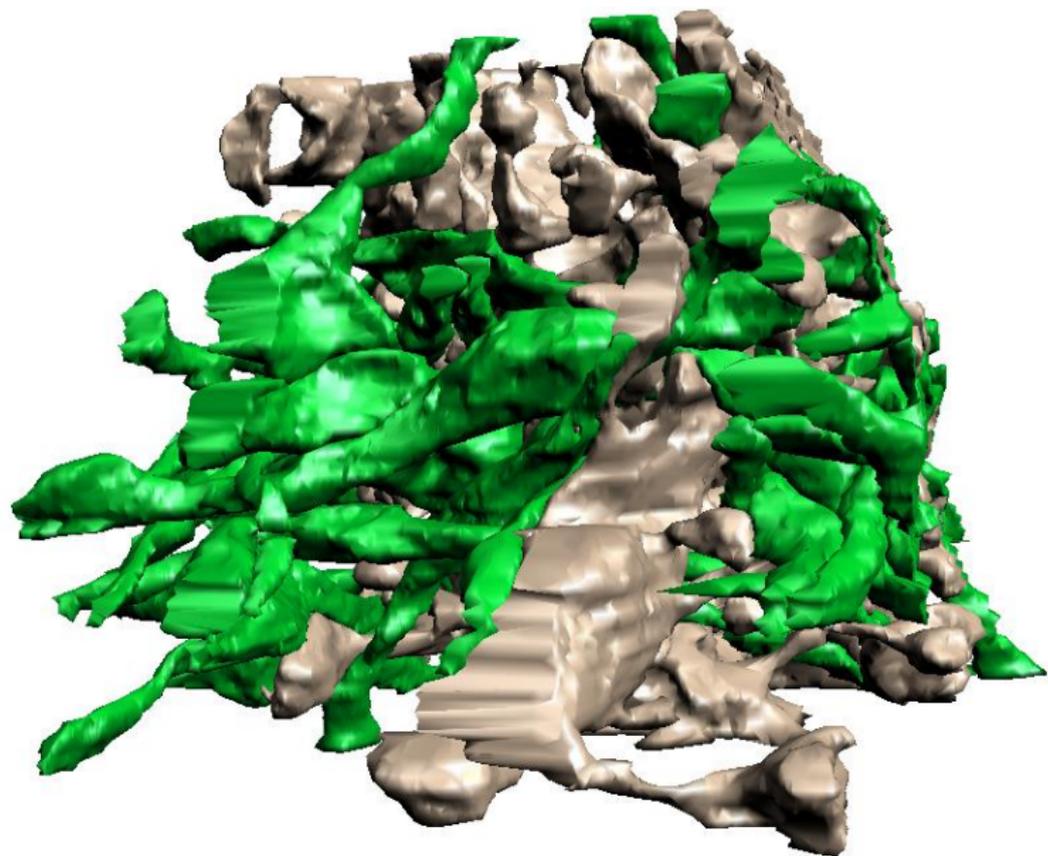


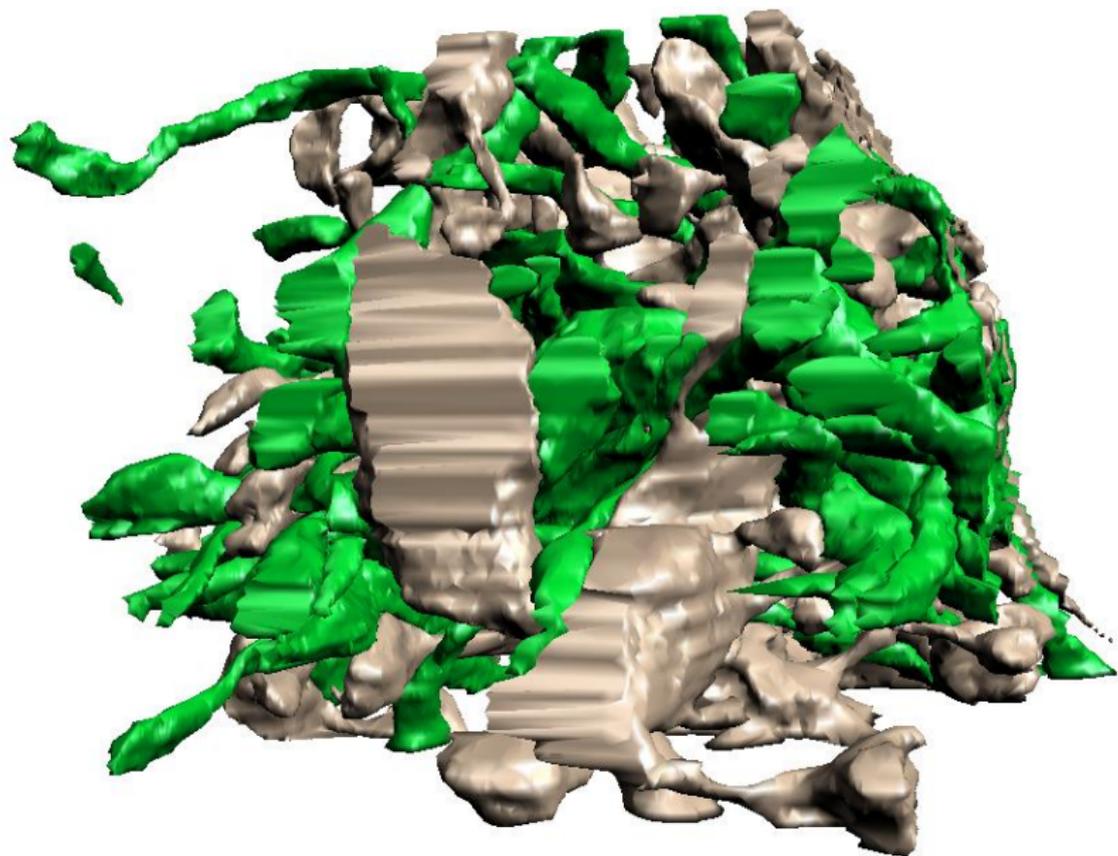


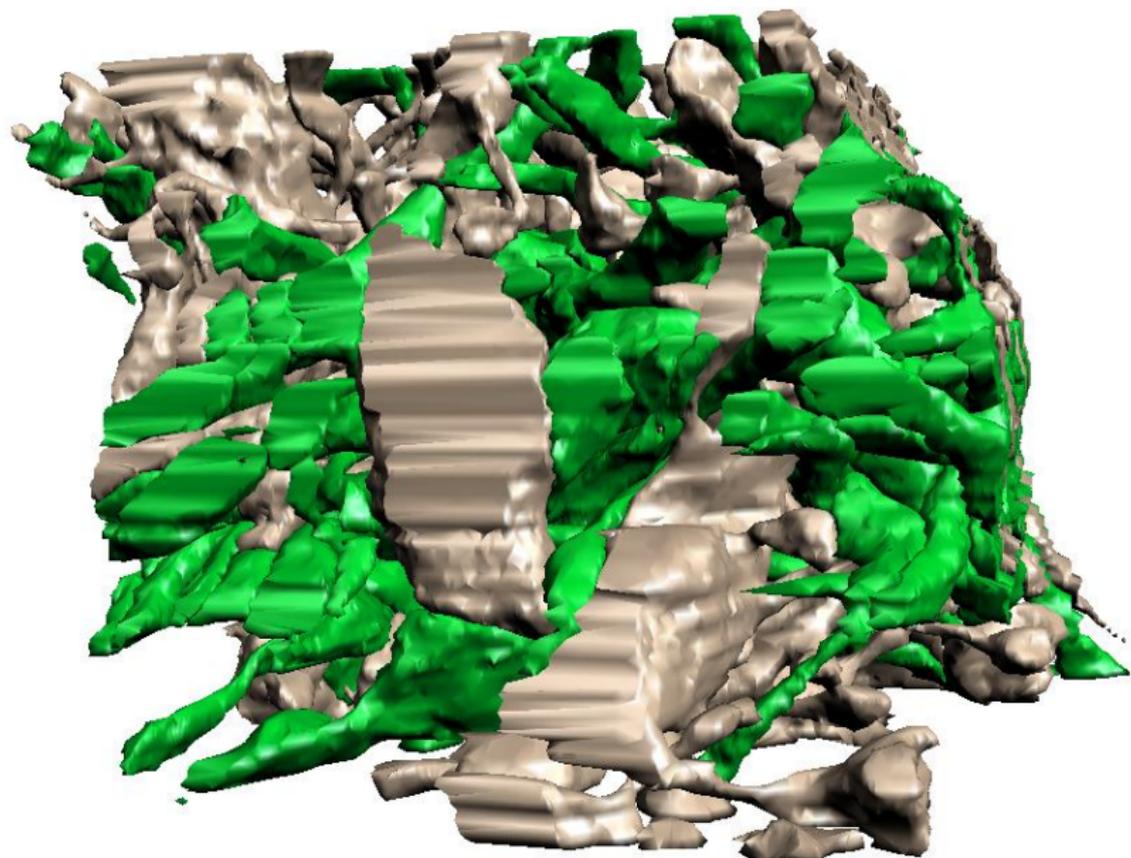


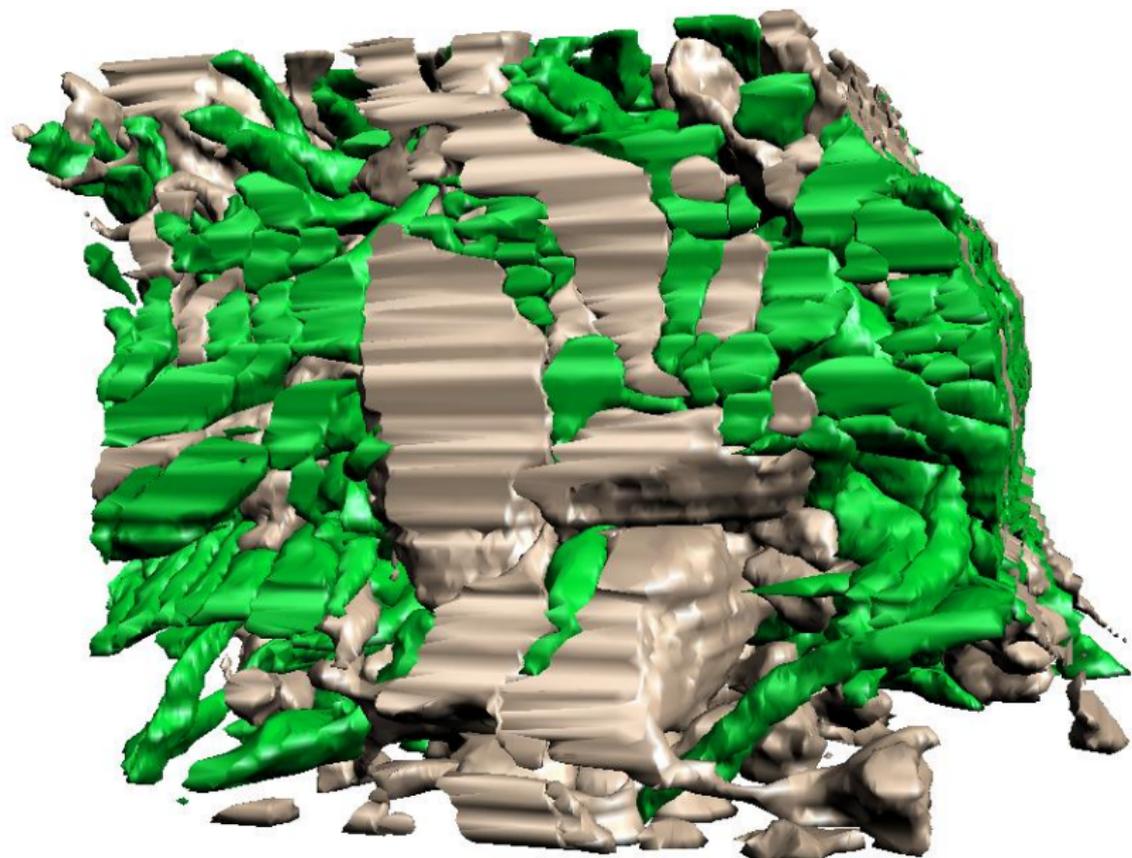


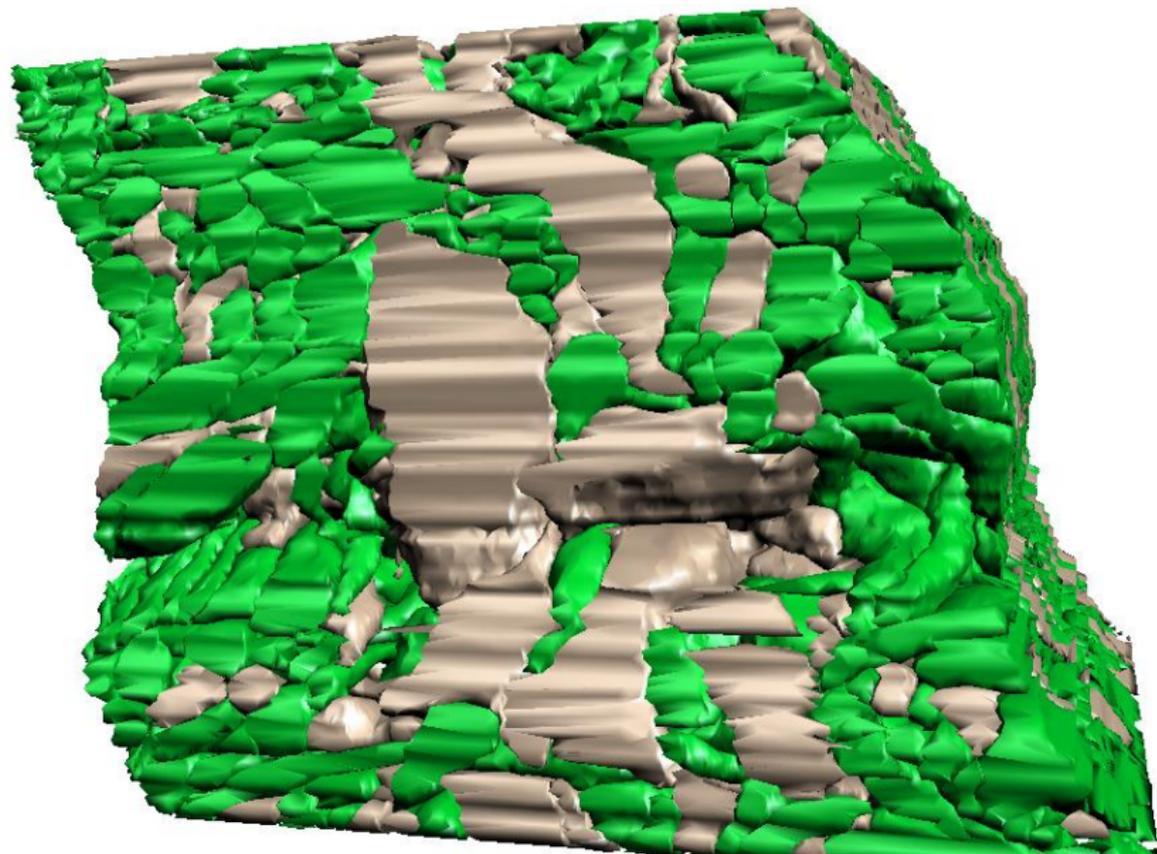


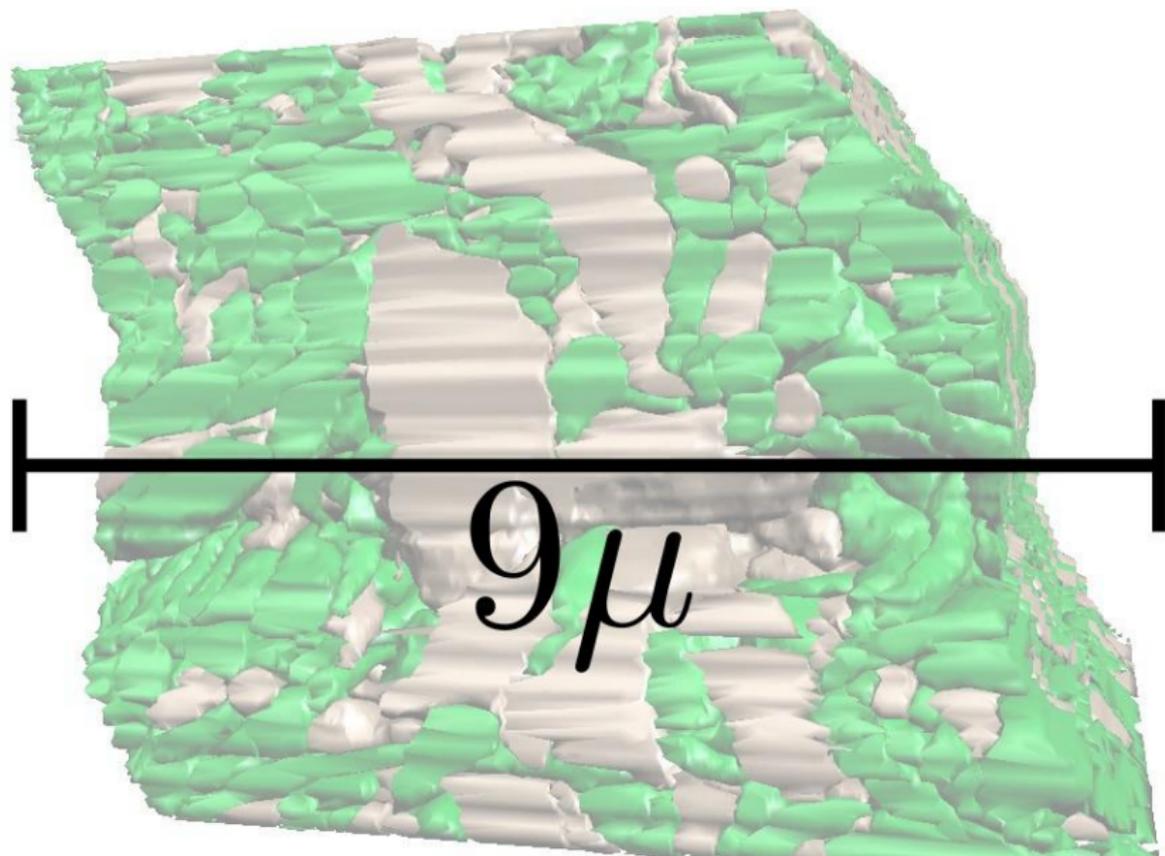


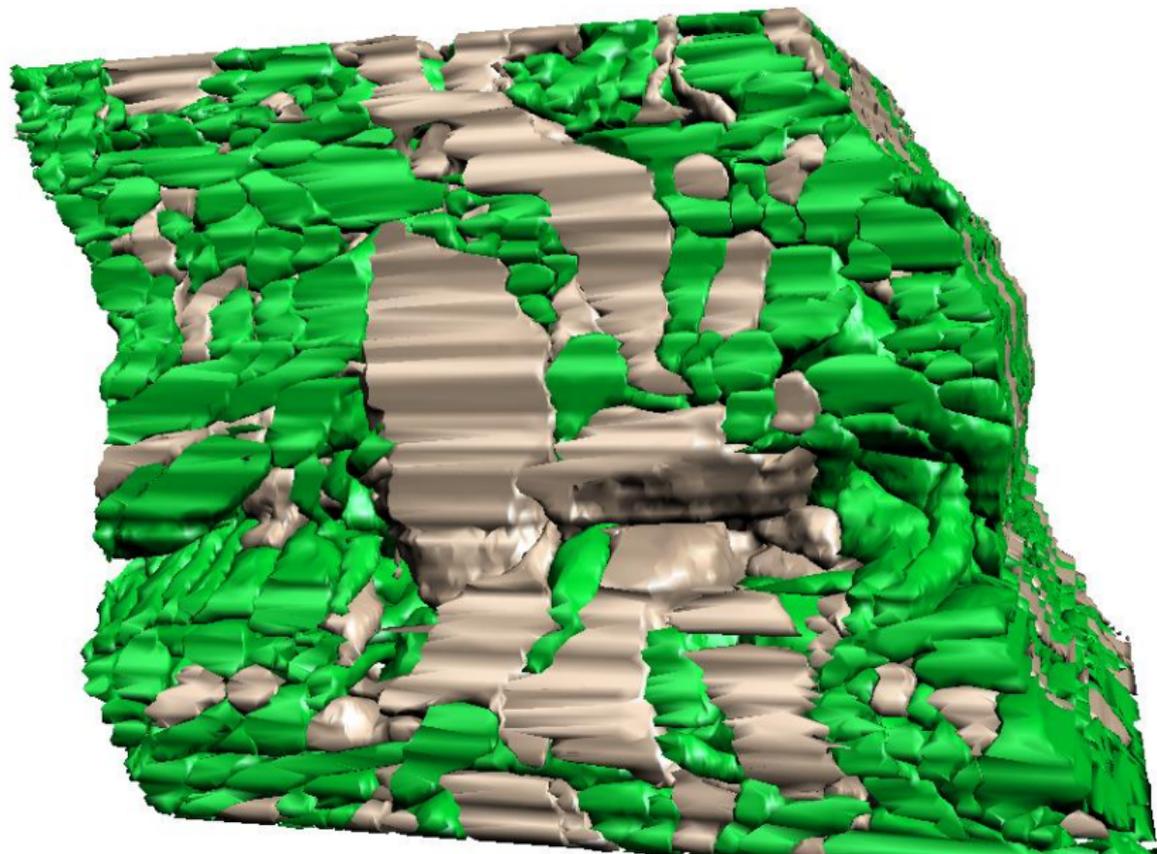


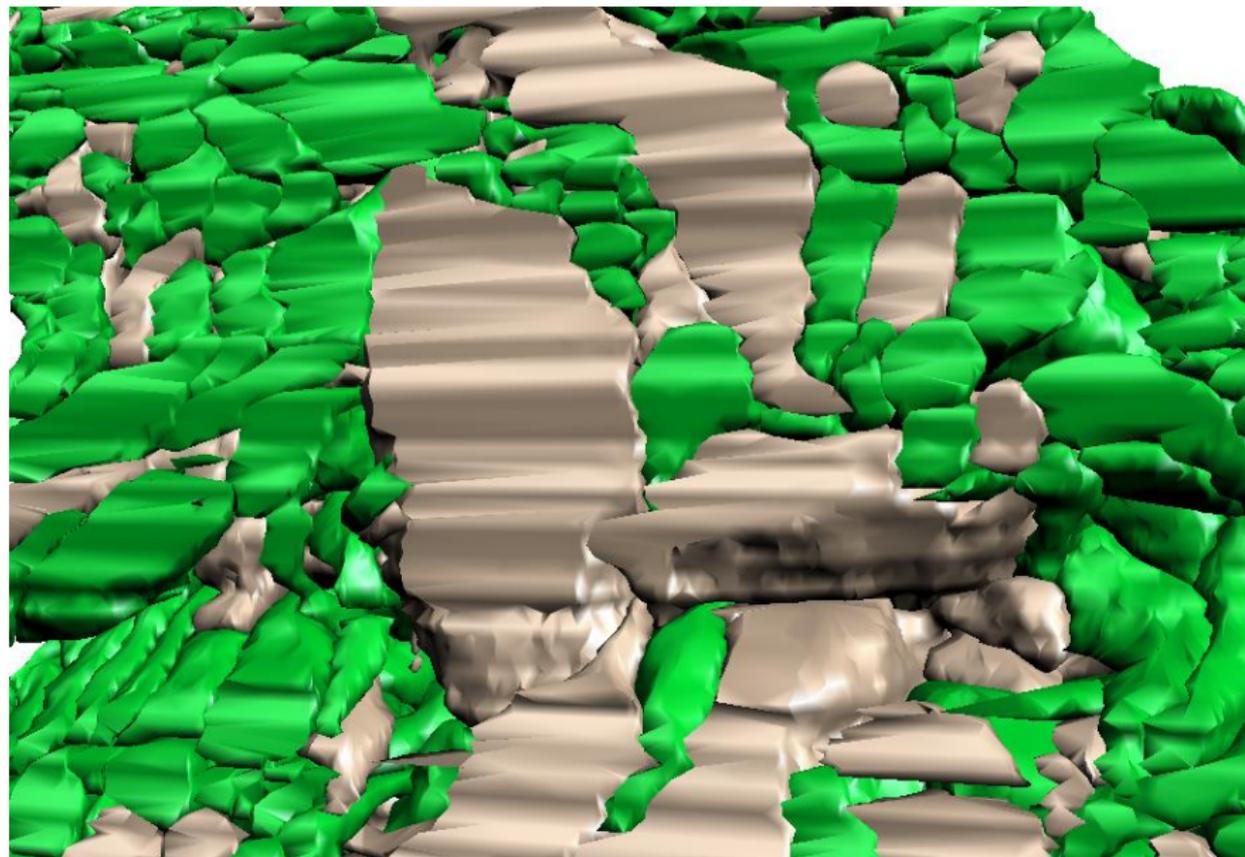


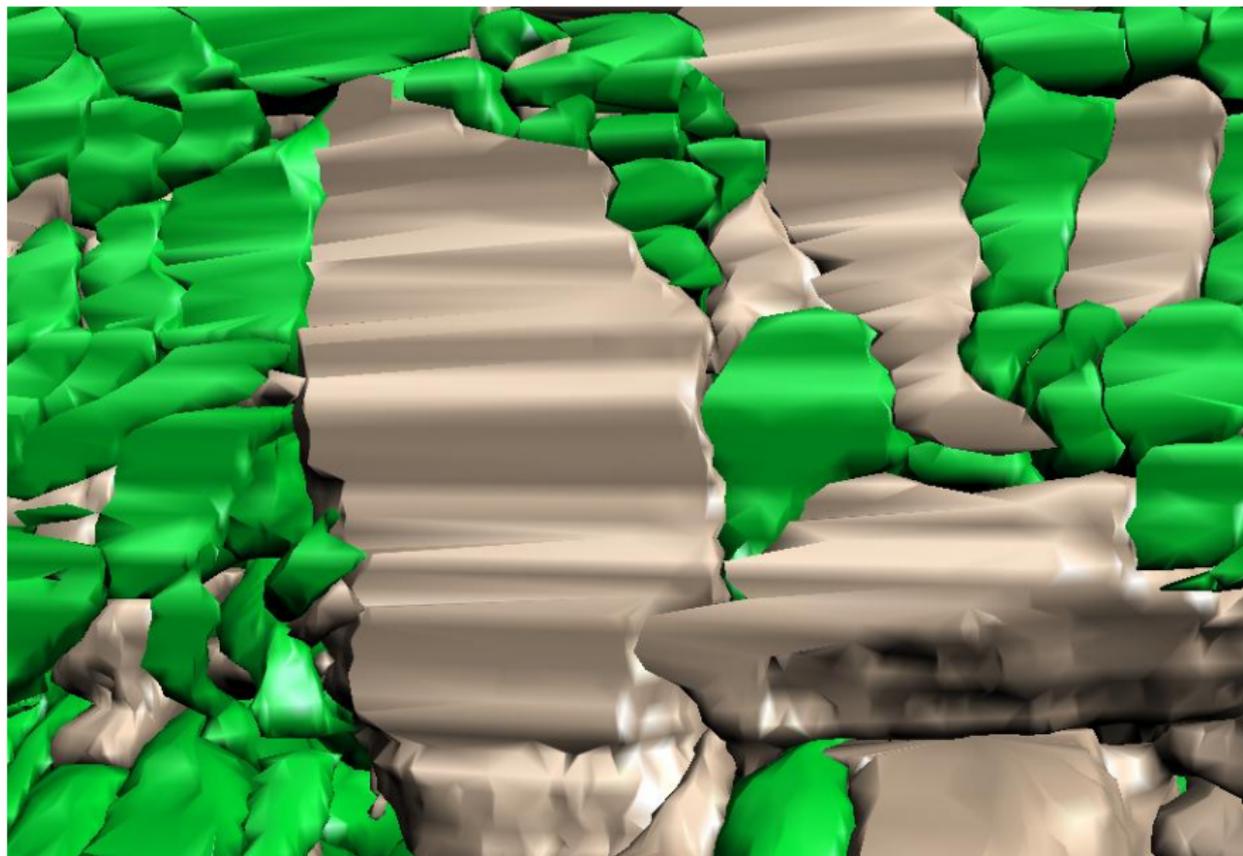


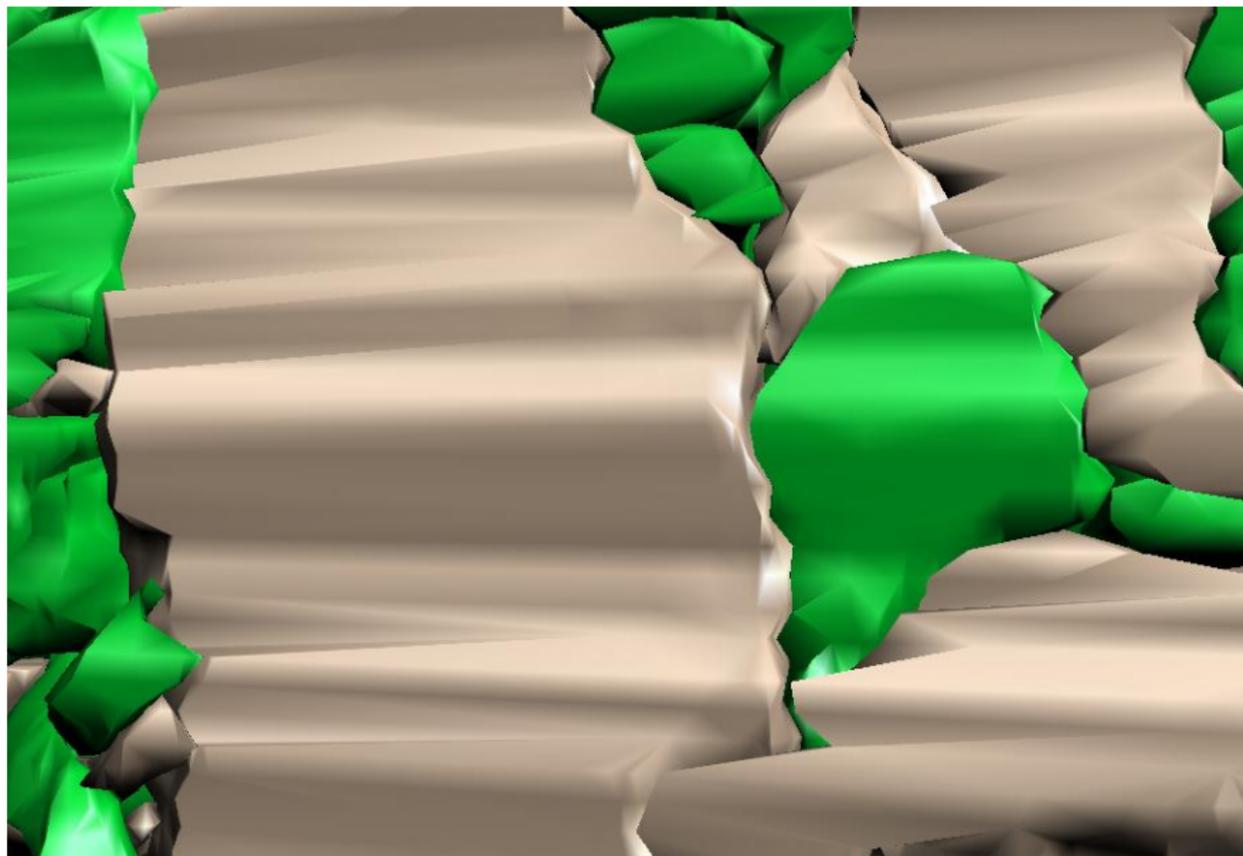


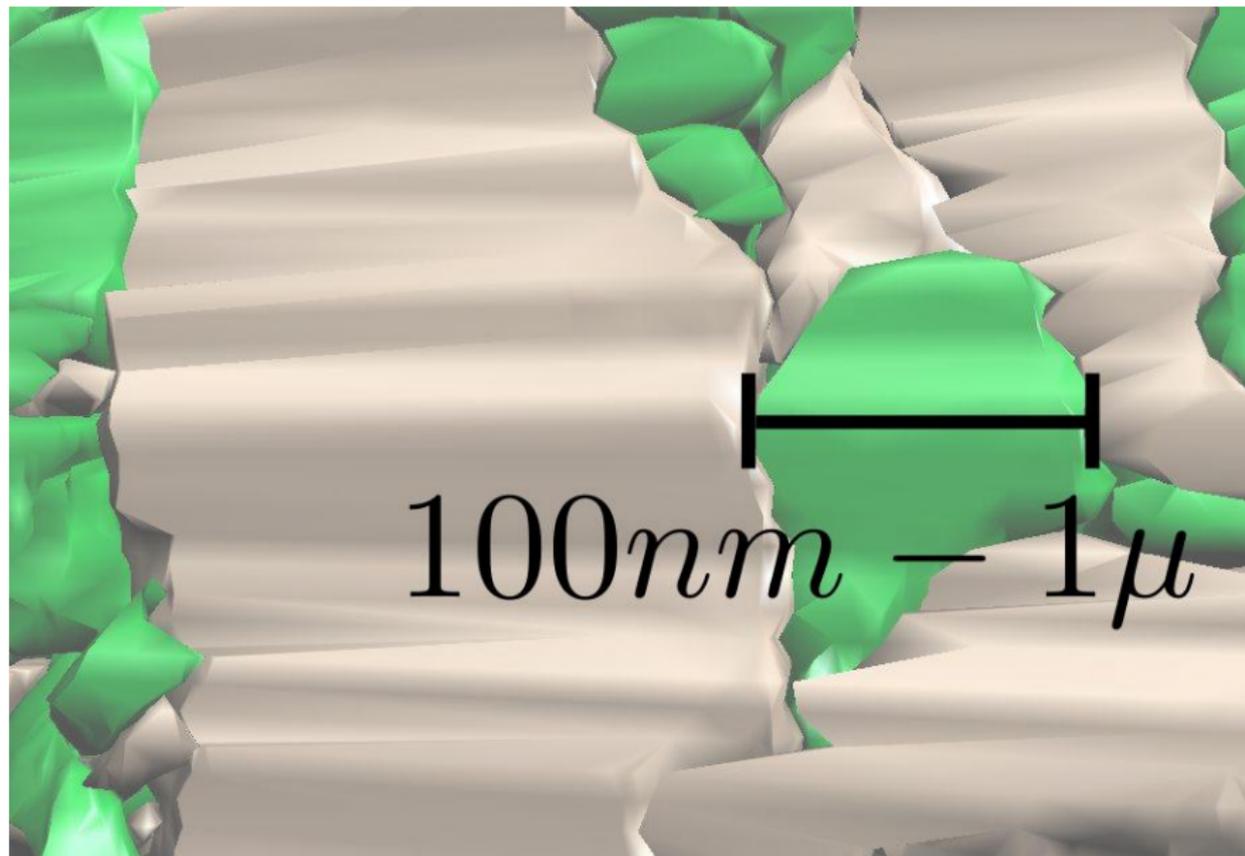


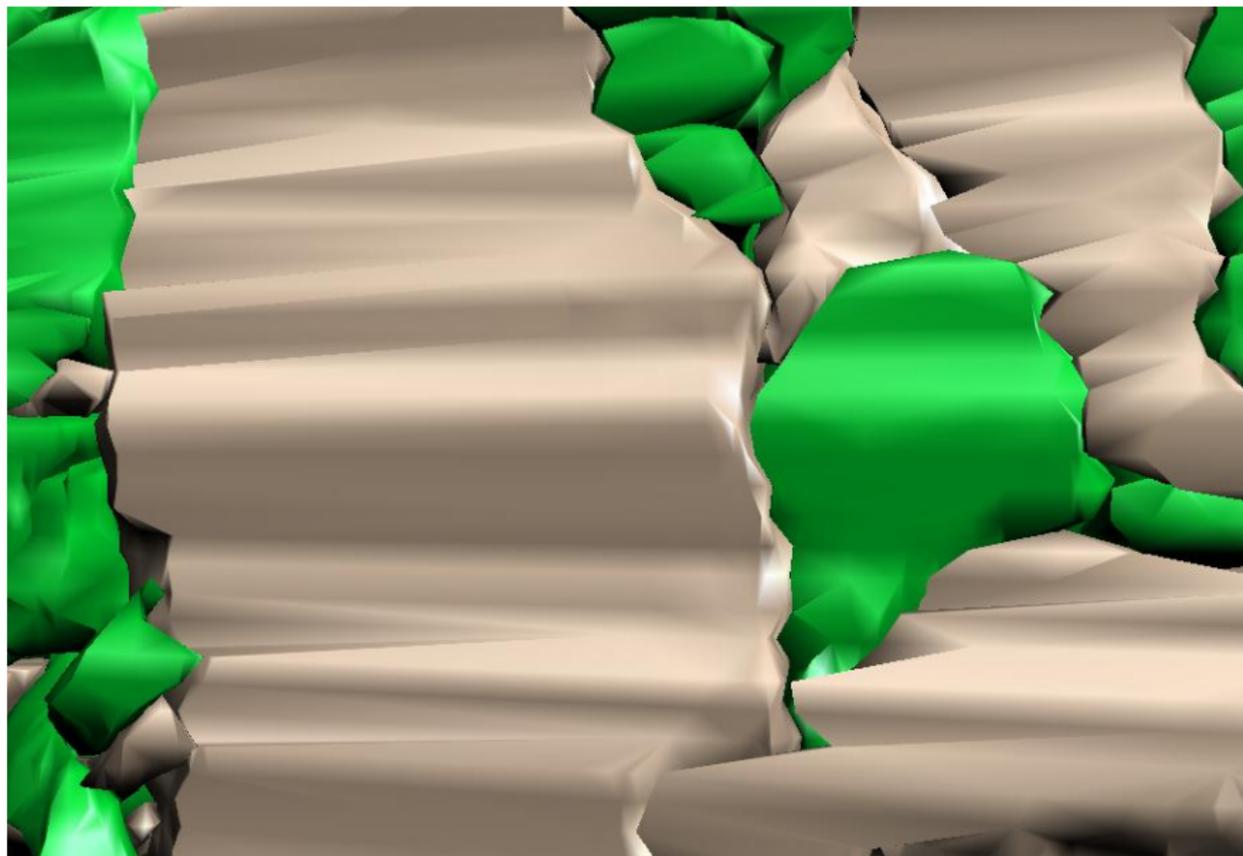




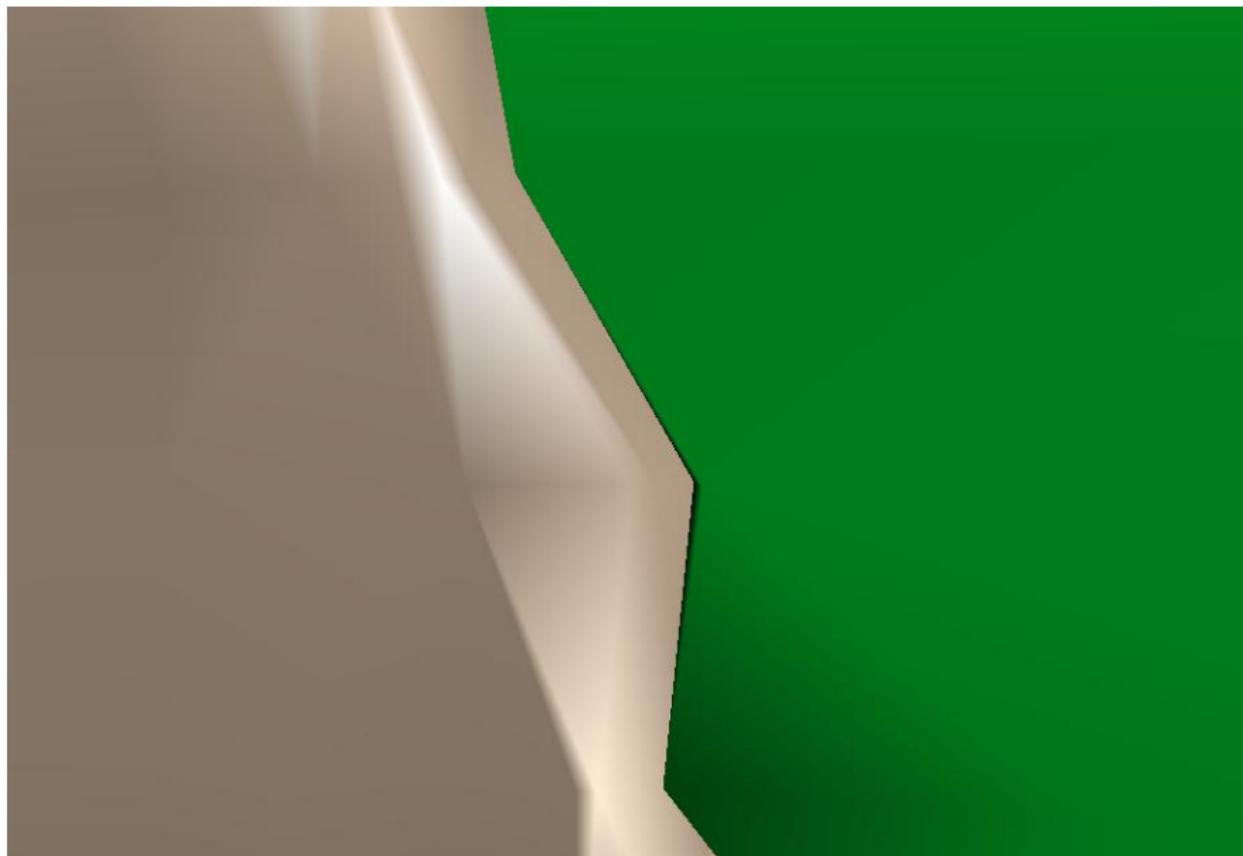


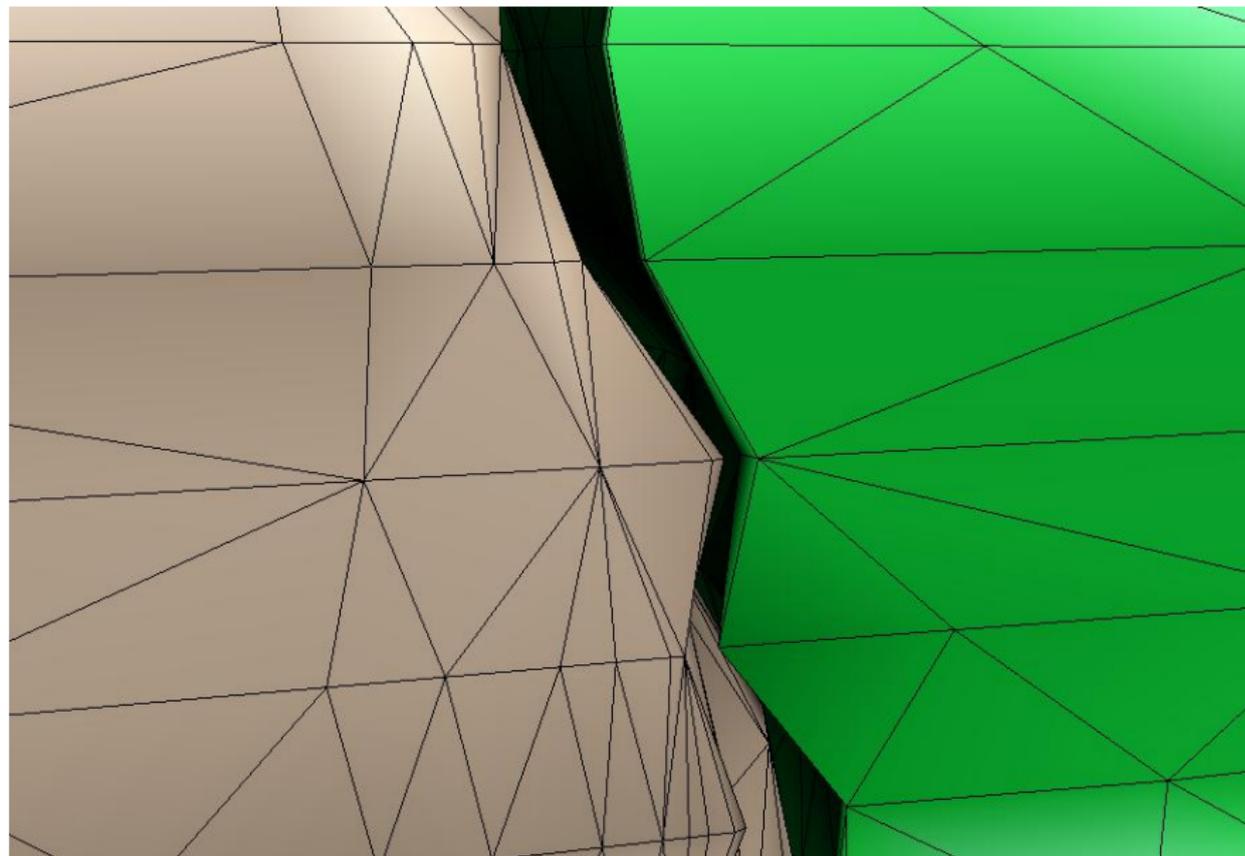


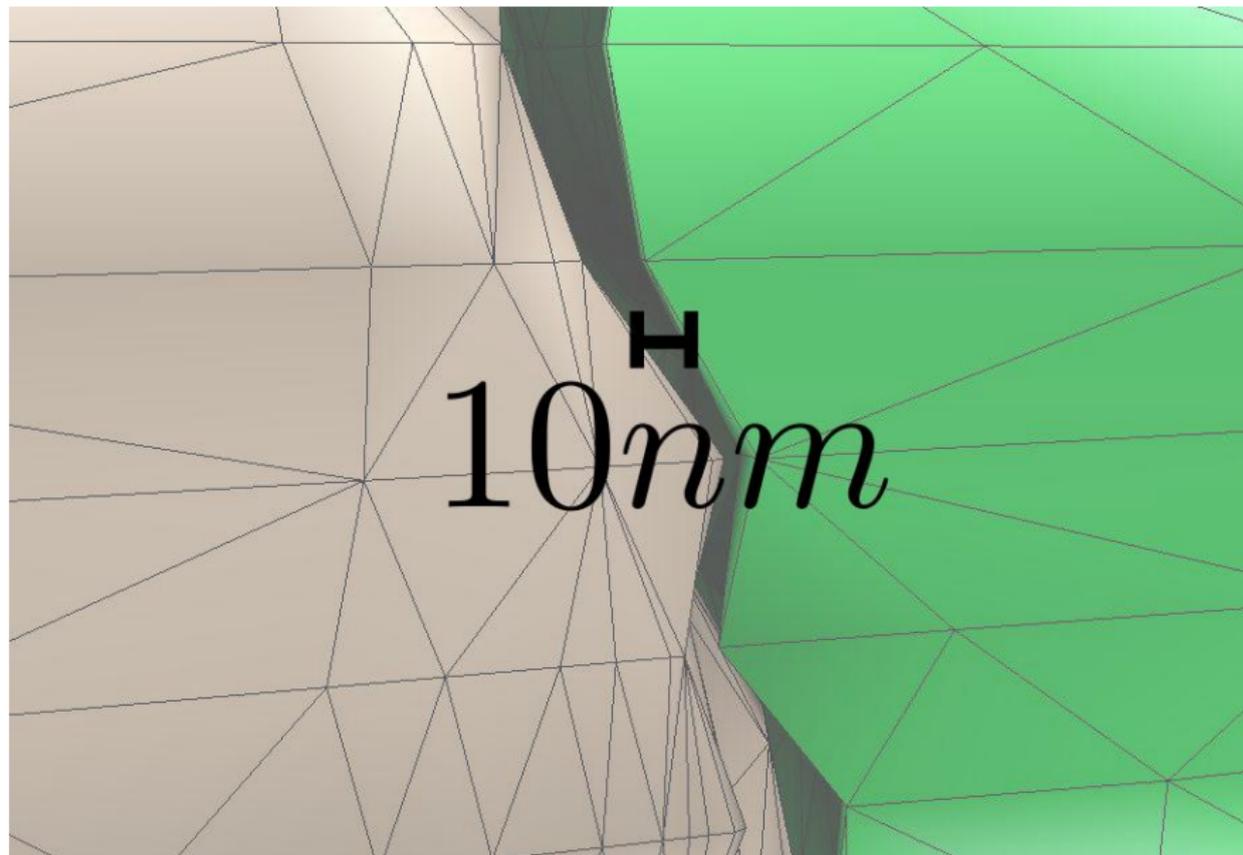


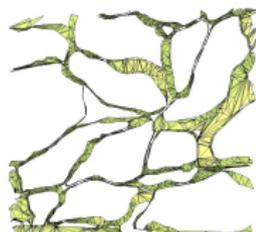




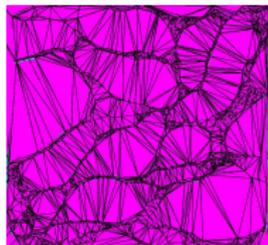








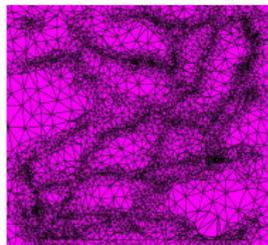
$1\mu \times 1\mu \times 100nm$



Not quality:
32817 tetrahedra



$1\mu \times 1\mu \times 100nm$



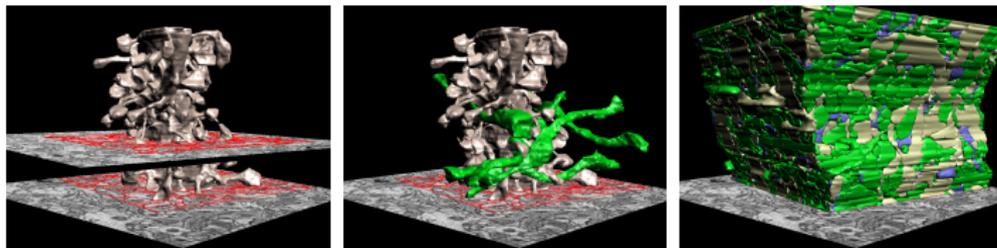
Quality:
124458 tetrahedra

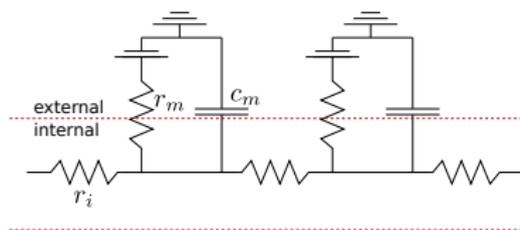
Tetrahedralize entire block and use FEM:

- 3×10^8 tetrahedra
- 3×10^8 variables to solve for

Problem 1

Develop and describe methods to generate a surface mesh domain model D from a labeled stack of 2D contours.





The cable equation is

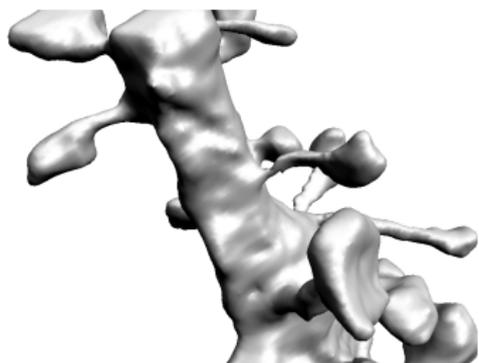
$$\frac{1}{r_i} \frac{\partial^2 V}{\partial x^2} = c_m \frac{\partial V}{\partial t} + \frac{V}{r_m} \quad (1)$$

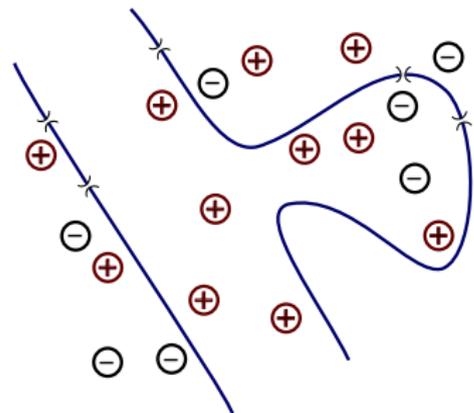
- r_i^* internal resistance
- r_m^* membrane resistance
- V potential on the cell membrane boundary
- x position on the compartment
- c_m capacitance of the membrane

Starred (*) variables are dependent on surface area, cross-sectional area and volume of the compartment. The compartments are combined using Kirchhoff's first law (conservation of current in a circuit) giving the final governing equations K .

Problem 2

Given a surface mesh of domain D , compute a volumetric decomposition M_K for analysis based on equations K .





The Poisson-Nernst-Planck equations are

$$\frac{\partial c_k}{\partial t} = \vec{\nabla} \cdot [-\vec{J}_k] \quad (2)$$

where $\vec{J}_k = D_k(\vec{\nabla}c_k + \left(\frac{c_k}{\alpha_k}\right)\vec{\nabla}V(\vec{r}, t))$ and

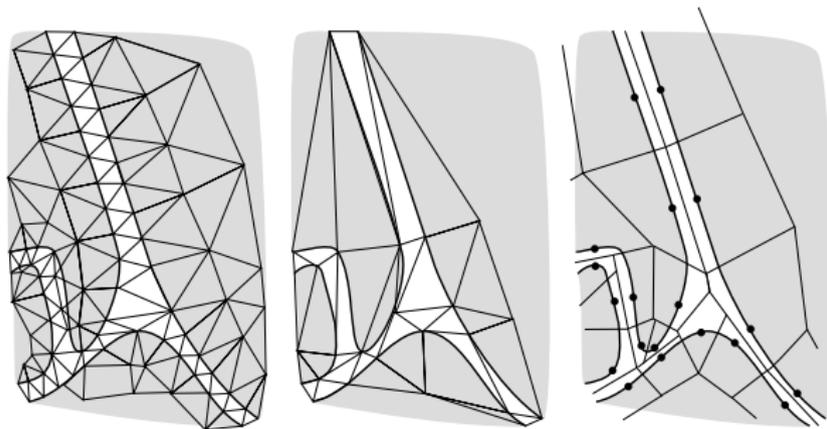
$$\vec{\nabla} \cdot [\epsilon(\vec{r})\vec{\nabla}V(\vec{r}, t)] = \rho(\vec{r}) \quad (3)$$

c_k	ion concentration of species k
\vec{J}_k	ion flux
$\epsilon(\vec{r})$	dielectric constant
$\{D_k, \alpha_k, z_k, F\}$	constants
$\rho(\vec{r}) = \sum_k c_k z_k F$	charge density

These equations are solved over a volumetric mesh and require accurate boundary representation (the Neumann boundary condition in our problem is $\hat{n} \cdot \nabla c_k = 0$ to ensure that ions don't diffuse across boundaries) and correct surface normals (after applying the divergence theorem). We call these equations PNP.

Problem 3

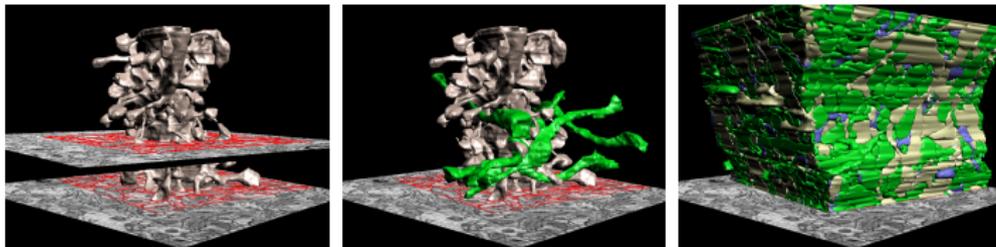
Given a surface mesh of domain D , compute a quality 3D mesh M_{PNP} that is suitable for solving equations PNP.



- 1 Motivating example and problem statements
- 2 Problem 1 - surface mesh domain models**
- 3 Problem 2 - micro-scale multi-compartment model
- 4 Problem 3 - nano-scale FEM model
- 5 Moving forward

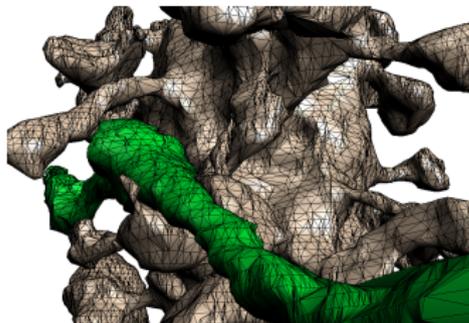
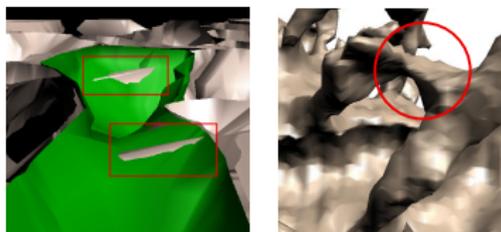
Problem 1

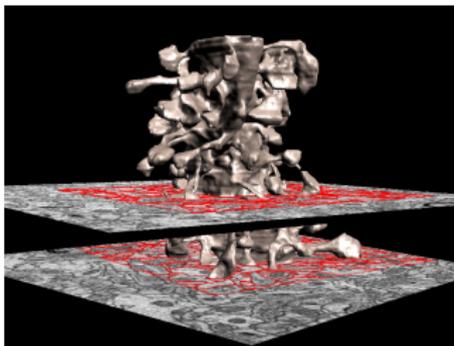
Develop and describe methods to generate surface mesh domain model D from a labeled stack of 2D contours.



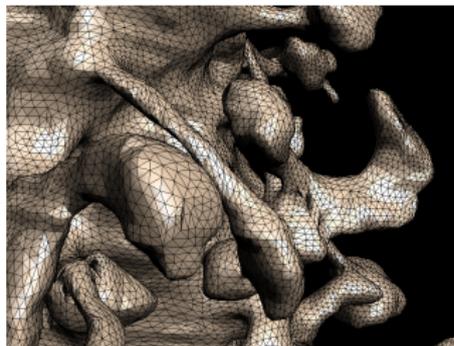


- A surface mesh is suitable if it
 - is water-tight
 - has oriented surface normals
 - is non-intersecting
 - has no mesh irregularities
 - has manifold edges and vertices
 - is composed of low aspect ratio triangles
 - is topologically correct
 - is close to the true surface

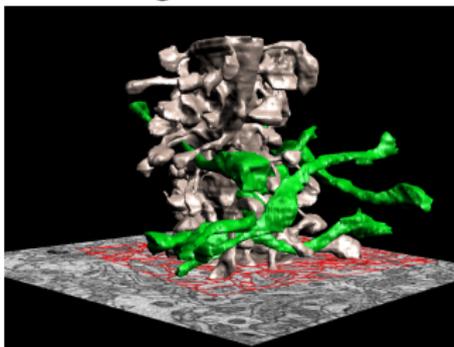




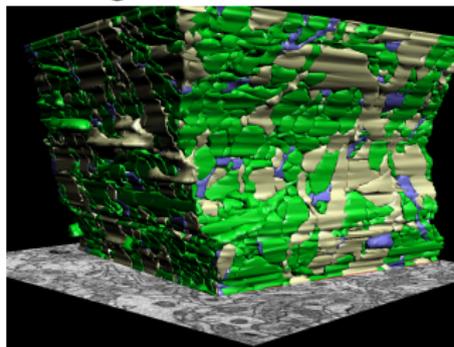
Images, contours



single reconstruction

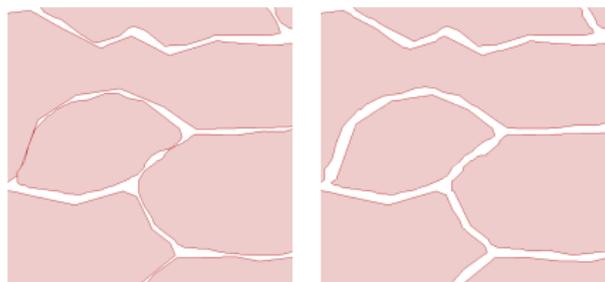
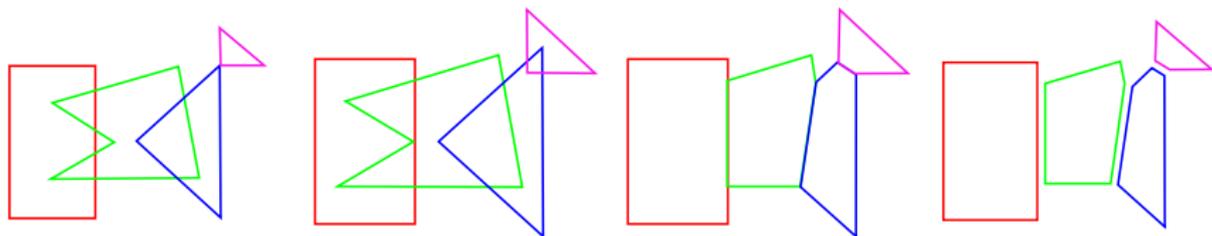


multi reconstruction



full domain

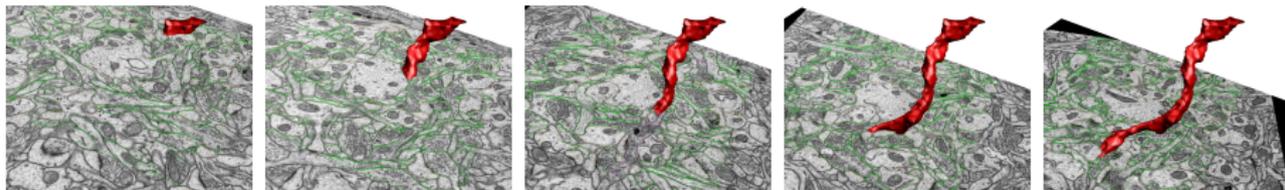
Reconstruction - 2D contour curation (CAD 2011) contribution

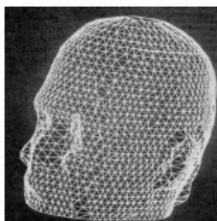


Algorithm:

- 1 Dilate contours by $\delta/2$
- 2 Sweep line to find intersections
- 3 Cut on line connecting adjacent intersections
- 4 Erode contours by $\delta/2$

Single component reconstruction problem: Given a stack of planar contours, reconstruct a surface mesh.

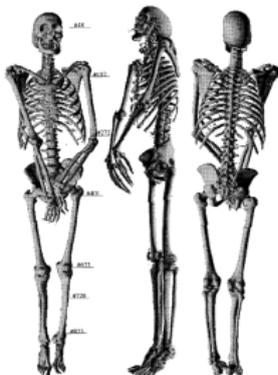




[Fuchs et al., 1977]



[Barequet and Sharir, 1994]

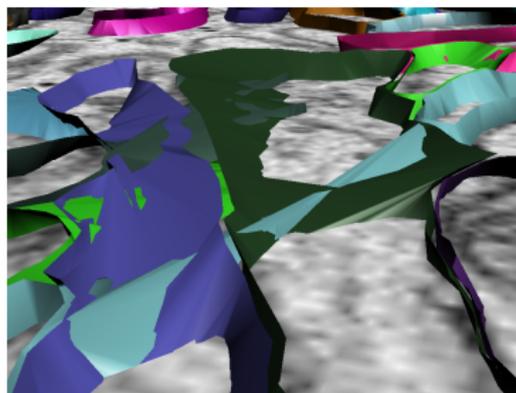


[Bajaj et al., 1996]

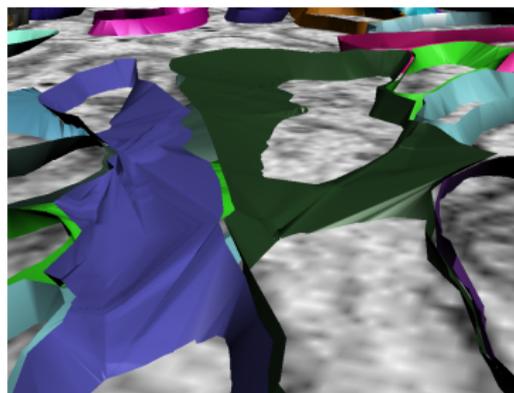
- One of the seminal works was [Fuchs et al., 1977] who posed the problem and presented a triangulation solution.
- [Barequet and Sharir, 1994] introduced a method using linear interpolations between slices of medical images.
- [Bajaj et al., 1996] expanded on their work to support arbitrary topologies. We use Bajaj's algorithm.

The problem: when we add singly-reconstructed components back together we get many interesections. This is because the data is

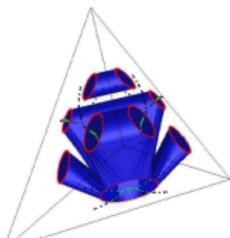
- anisotropic (spacing between slices is very large)
- tortuous
- tightly packed



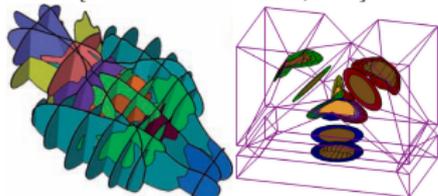
Many intersections between components



Without inter-component intersections

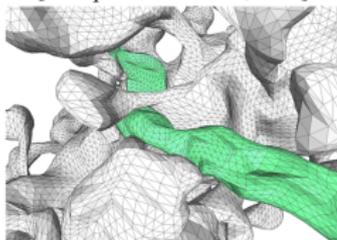


[Boissonnat and Memari, 2007]



[Liu et al., 2008]

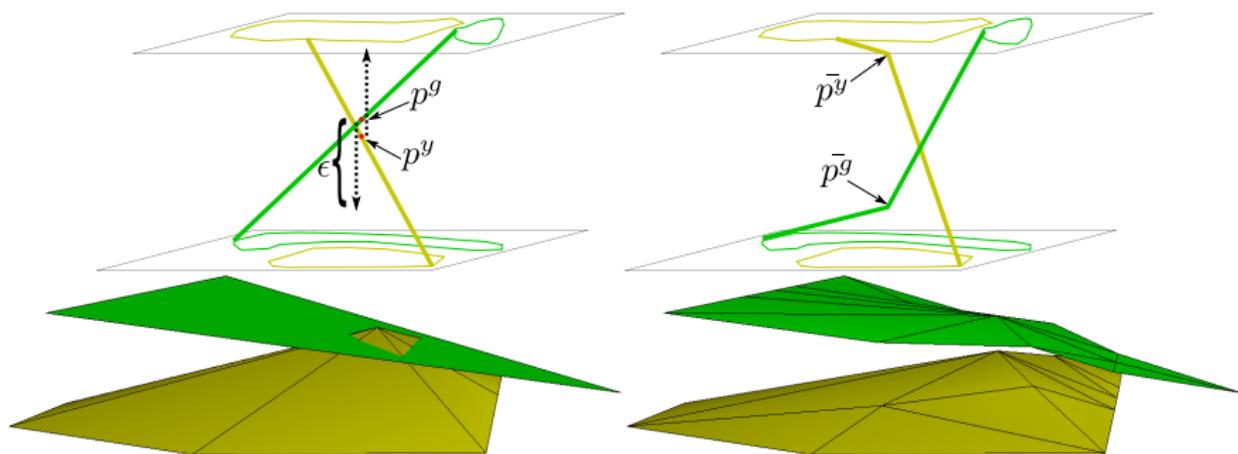
[Barequet and Vaxman, 2009]



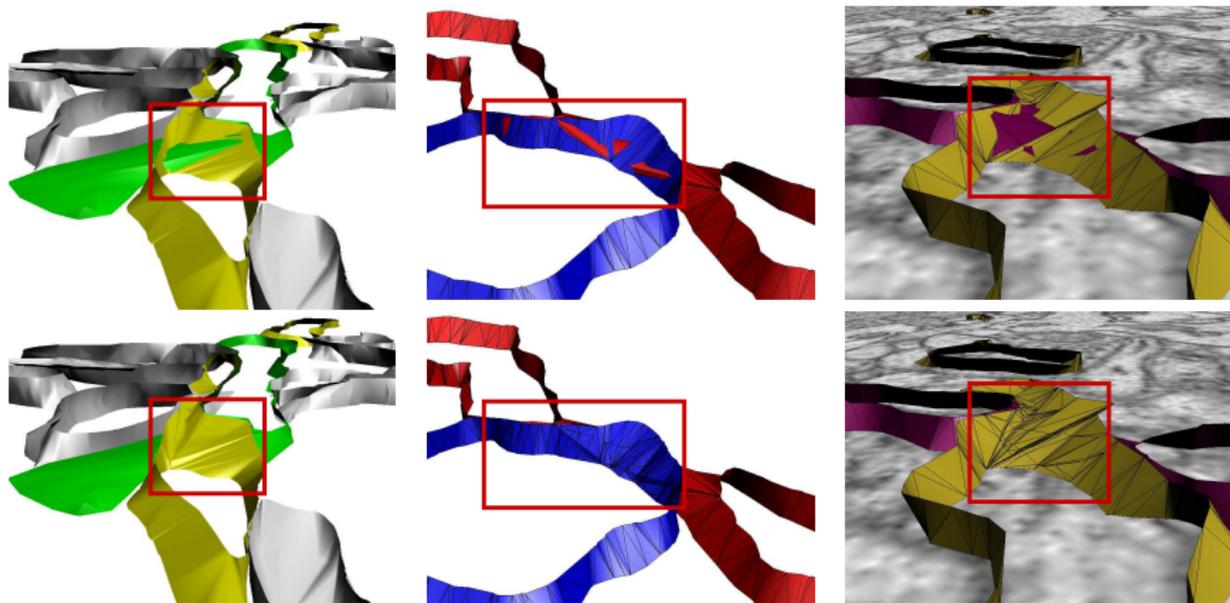
[Bajaj and Gillette, 2008]

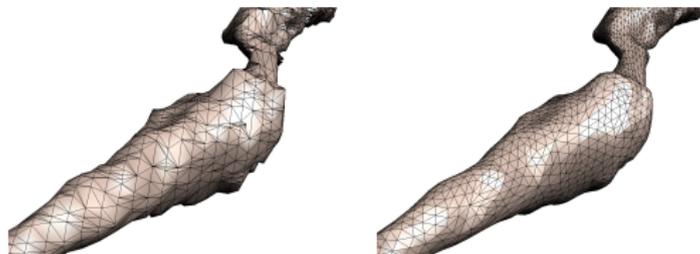
- Recent work by [Boissonnat and Memari, 2007] reconstructs single structures from non-parallel slices.
- Two approaches by [Liu et al., 2008] and [Barequet and Vaxman, 2009] reconstruct from non-parallel slices and additionally reconstruct multiple components at the same time, avoiding inter-component intersections.
- [Bajaj and Gillette, 2008] perform single component reconstruction using [Bajaj et al., 1996] and then remove intersections by removing contour overlaps in intermediate planes.

- [Edwards and Bajaj, 2011] post-processes surfaces and removes intersections.
- We can resolve “conflict points” by moving them in the directions of their penumbral contours without worrying about causing additional intersections (proof on slide A1). Once all conflict points are resolved, all intersections are removed.



Results

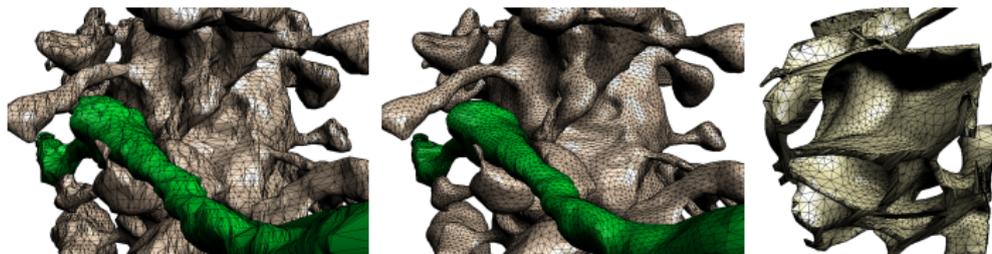




- We currently use QSlim [Garland, a] for surface mesh decimation and geometric flow [Zhang et al., 2005] for smoothing which produces a mesh with quality triangles. Many other approaches exist (e.g. [Garland and Heckbert, 1997, Cohen-Steiner et al., 2004, Lindstrom and Turk, 1998, Hoppe, 1996, Klein et al., 1996, Garland et al., 2001]).
 - No guarantees about maintaining intersection-free geometries
 - No error guarantees (and in most cases error isn't even reported)
 - Extracellular width isn't maintained

Sub-problem

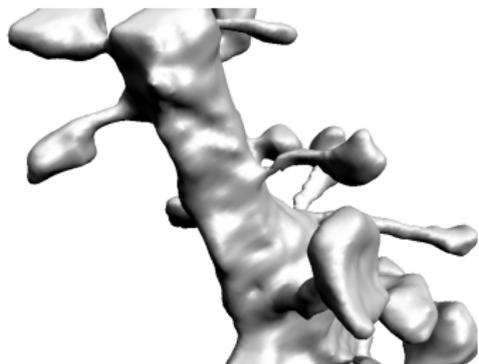
Decimate and smooth surface meshes while maintaining extracellular width

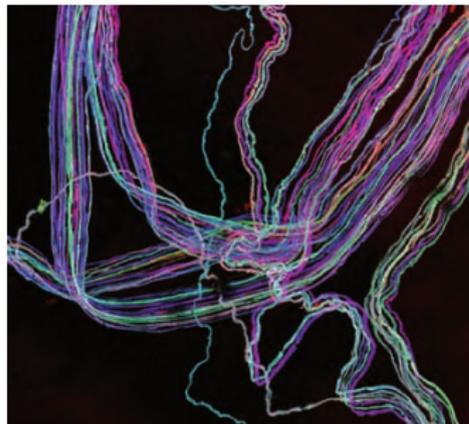


- 1 Motivating example and problem statements
- 2 Problem 1 - surface mesh domain models
- 3 Problem 2 - micro-scale multi-compartment model**
- 4 Problem 3 - nano-scale FEM model
- 5 Moving forward

Problem 2

Given a surface mesh of domain D , compute a volumetric decomposition M_K for analysis based on equations K .



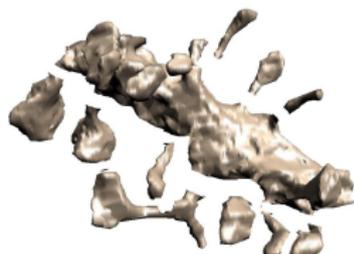


[Lichtman et al., 2008]

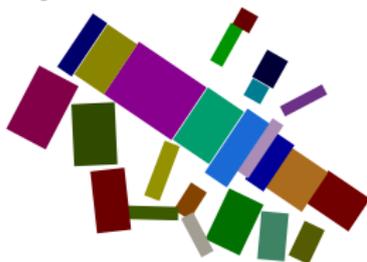
- Multi-compartment models are typically complete or nearly-complete neurons
- Imagery used is micron scale (light microscopy)
- There is significant literature in neuron tracing and diameter estimation [Peng et al., 2010, Xie et al., 2010, Cohen et al., 1994]
- But we are interested in deriving models from much higher-resolution microscopy
 - We deal with much smaller fields of view as a consequence



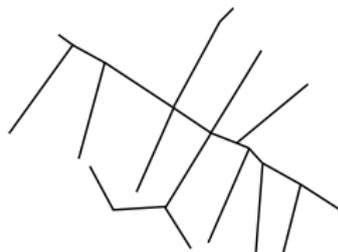
Step 1. Start with a dendrite (or axon).



Step 2. Decompose volume.



Step 3. Assign proxies with identical surface area, cross-sectional area, and volume.



Step 4. Add compartments with resistances and capacitances derived from geometric properties of proxies.

- Existing approaches include [Fiala, 2005, Helmstaedter et al., 2011, Jeong et al., 2010]. They are largely manual and may not preserve sum volumes.
- We seek an accurate and *automatic* algorithm.

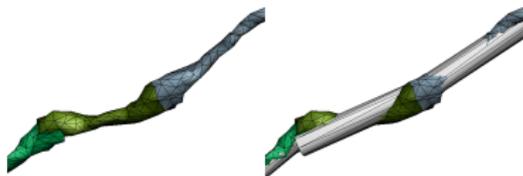
1 Fit cylinder to each segment in skeleton

- Care must be taken not to overcount volume
- Existing methods use this approach



2 Partition surface and fit cylinders

- Use variational approach [Cohen-Steiner et al., 2004]
- Difficult to prove performance

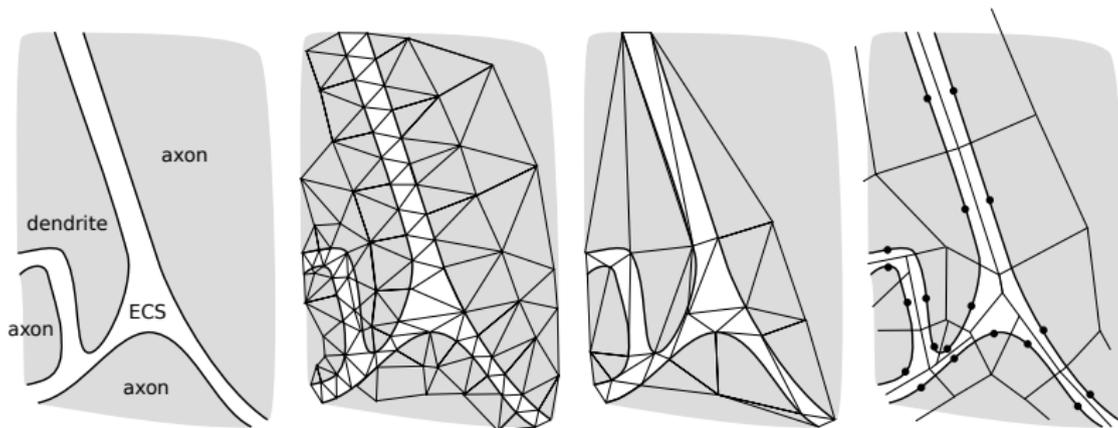


Outline

- 1 Motivating example and problem statements
- 2 Problem 1 - surface mesh domain models
- 3 Problem 2 - micro-scale multi-compartment model
- 4 Problem 3 - nano-scale FEM model**
- 5 Moving forward

Problem 3

Given a surface mesh of domain D , compute a quality 3D mesh M_{PNP} that is suitable for solving equations PNP.



Consider the equation $-u'' = f(x)$. The finite element method of solving it comprises the following steps:

- Derive a weak formulation of the equation.
 - Introduce a test function v to yield $-u''v = fv$.
 - Take the integral of both sides and integrate by parts: $\int_{\Omega} u'v' dx = \int_{\Omega} fv dx$.
 - Solving this equation for all $v \in H^1$ yields the solution function u .
- Discretize the domain Ω and define basis functions $\{\phi_i\}$.
 - Basis functions must be integrable, typically polynomials.
 - A “finite element” e_i is a discrete cell $c_i \in \Omega$ and basis functions $\{\phi_j\}$.
- Cast u and v as linear combinations of $\{\phi_i\}$.
- Solve

$$\int_{\Omega} \sum_i \phi_i \beta_i \sum_j \phi_j \alpha_j dx = \int_{\Omega} f \sum_i \phi_i \beta_i$$

We can rearrange this into a linear system

$$K\alpha = F$$

where $K_{ij} = \int_{\Omega} \phi_i' \phi_j'$ and $F_i = \int_{\Omega} f \phi_i$.

The error of the solution from the finite element method is bounded as follows:

$$\|e\| \leq Ch^P$$

where

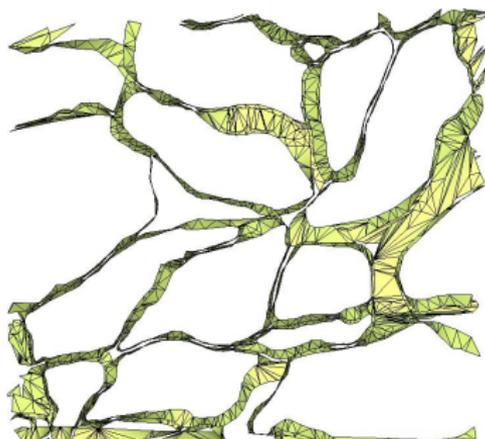
C depends on the curvature of the function

h size of the mesh elements

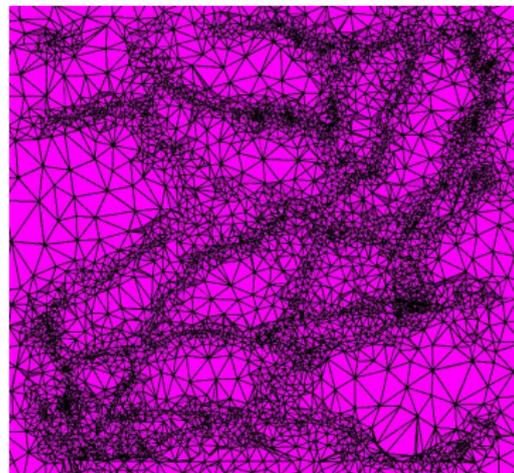
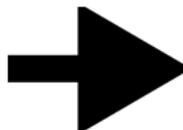
P depends on the norm used and the basis functions

For each equation and domain we must determine

- Discretization of domain
- Basis functions

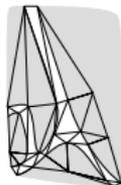


High anisotropy...



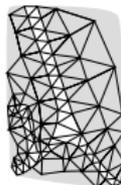
...explosion of elements

- This example is $1\mu \times 1\mu \times 100nm$ and yields 124458 tetrahedra
- Entire block yields 3×10^8 variables to solve using FEM
- **Goal:** produce as few elements as possible without compromising numerical solution



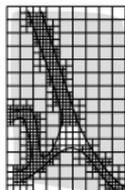
Constrained anisotropic triangulation

- Small and large angles cause badly conditioned linear systems
- Large angles hurt error convergence



Constrained quality triangulation

- Well-conditioned systems
- Large number of elements



Grid (meshless)

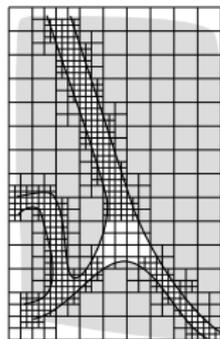
- Use scaffolds rather than boundary-conforming discretization
- Need weight functions for boundary conditions



Aligned (novel)

- Few elements
- Starting point for further decomposition

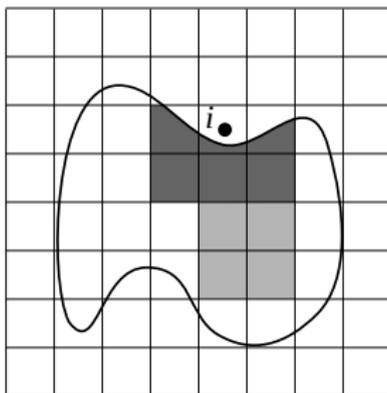
- WEB-spline is a meshless method
- B-splines are used as the basis
- B-splines are multiplied by weight functions $\{\omega_i\}$ that vanish on $\partial\Omega$ to ensure boundary conditions
- Often extra cells need to be “adjoined” to ensure stability



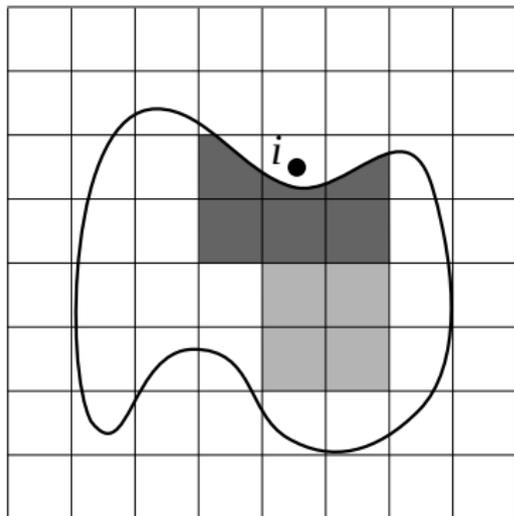
Grid with tensor-product
B-splines



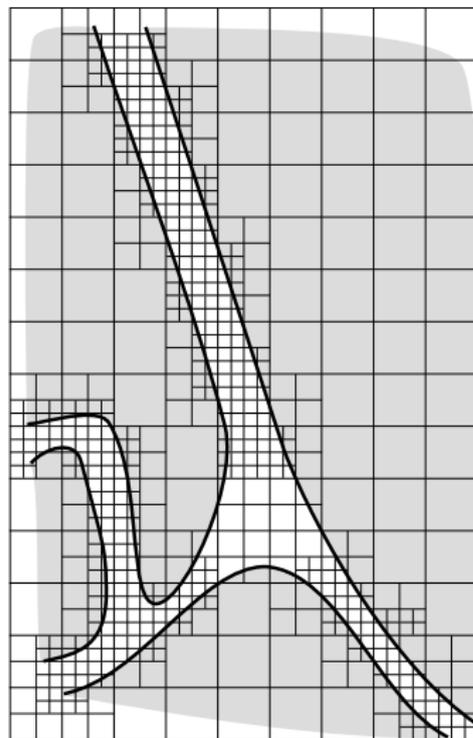
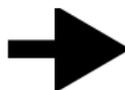
Mesh with weight functions



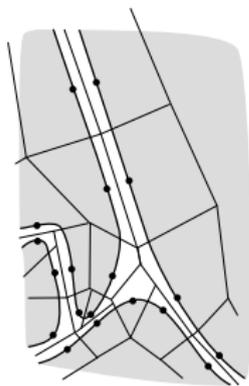
Dark gray - support cells. Light gray - adjoined support cells.



Often the requirements for stability results in...



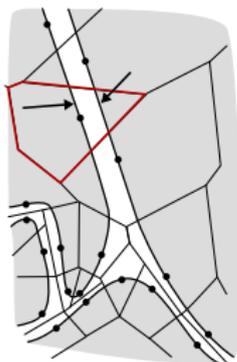
...large numbers of cells!



Aligned elements

Path to a solution: produce large cells with certain properties that we

- May be able to use as-is as a decomposition
- Can at least be used to derive non-axis-aligned decompositions for meshless methods
- Can use to derive anisotropic meshes for traditional FEM



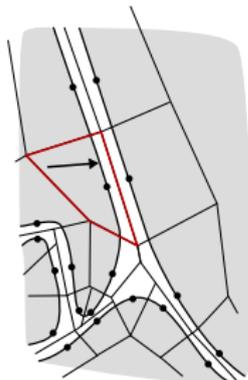
Fails criterion 2

Criteria:

- ❶ Polygons are convex
- ❷ Immersed boundaries have only one connected component inside a given cell. That is, let b be all immersed boundaries and element $e \in M_{\text{PNP}}$. Then $b \cap e$ must have a single connected component.
- ❸ The number of elements $|M_{\text{PNP}}|$ is small compared to a constrained, quality Delaunay tetrahedralization.

Ideas:

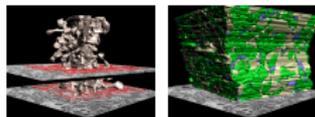
- ❶ Resample points P on $\partial\Omega$ and use Voronoi diagram
- ❷ Resample points P on medial sheet and use Voronoi diagram



Meets criterion 2

Outline

- 1 Motivating example and problem statements
- 2 Problem 1 - surface mesh domain models
- 3 Problem 2 - micro-scale multi-compartment model
- 4 Problem 3 - nano-scale FEM model
- 5 Moving forward**



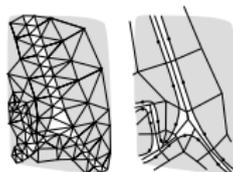
Problem 1 - 3D surface mesh

- Largely done
- Decimation that respects ECS is a possible task



Problem 2 - multi-compartment models

- Investigate skeleton vs. variational approaches
- Determine which best decomposes volume



Problem 3 - FEM models

- Further characterize mesh requirements
- Determine algorithms with provable quality

Thank you!

References

- [Amenta et al., 1998] Amenta, N., Bern, M. and Eppstein, D. (1998).
The crust and the beta-skeleton: combinatorial curve reconstruction.
Graphical models and image processing 60, 125–135.
- [Au et al., 2008] Au, O. K.-C., Tai, C.-L., Chu, H.-K., Cohen-Or, D. and Lee, T.-Y. (2008).
Skeleton extraction by mesh contraction.
ACM Transactions on Graphics 27, 44:1–10.
- [Bajaj et al., 1996] Bajaj, C., Coyle, E. and Lin, K. (1996).
Arbitrary topology shape reconstruction from planar cross sections.
Graphical Models and Image Processing 58, 524–543.
- [Bajaj and Gillette, 2008] Bajaj, C. and Gillette, A. (2008).
Quality Meshing of a Forest of Branching Structures.
In Proceedings of the 17th International Meshing Roundtable pp. 433–449, Springer-Verlag.
- [Barequet and Sharir, 1994] Barequet, G. and Sharir, M. (1994).
Piecewise-Linear Interpolation between Polygonal Slices.
In Computer Vision and Image Understanding pp. 93–102,.
- [Barequet and Vaxman, 2009] Barequet, G. and Vaxman, A. (2009).
Reconstruction of multi-label domains from partial planar cross-sections.
In SGP '09: Proceedings of the Symposium on Geometry Processing pp. 1327–1337, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland.
- [Boissonnat and Memari, 2007] Boissonnat, J. and Memari, P. (2007).
Shape reconstruction from unorganized cross-sections.
In Proceedings of the fifth Eurographics symposium on Geometry processing pp. 89–98, Eurographics Association.
- [Cohen et al., 1994] Cohen, A., Roysam, B. and Turner, J. (1994).
Automated tracing and volume measurements of neurons from 3-D confocal fluorescence microscopy data.
Journal of microscopy 173, 103–114.

References]References

- [Cohen-Steiner et al., 2004] Cohen-Steiner, D., Alliez, P. and Desbrun, M. (2004).
Variational shape approximation.
ACM Transactions on Graphics 23, 905–914.
- [Du and Wang, 2005] Du, Q. and Wang, D. (2005).
Anisotropic centroidal Voronoi tessellations and their applications.
SIAM Journal on Scientific Computing 26, 737–761.
- [Edwards and Bajaj, 2011] Edwards, J. and Bajaj, C. (2011).
Topologically correct reconstruction of tortuous contour forests.
Computer-Aided Design 43, 1296 – 1306.
- [Fiala, 2005] Fiala, J. (2005).
Reconstruct: a free editor for serial section microscopy.
Journal of Microscopy 218, 52–61.
- [Fiala et al., 2002] Fiala, J., Spacek, J. and Harris, K. (2002).
Dendritic spine pathology: cause or consequence of neurological disorders?
Brain Research Reviews 39, 29–54.
- [Fuchs et al., 1977] Fuchs, H., Kedem, Z. and Uselton, S. (1977).
Optimal surface reconstruction from planar contours.
Communications of the ACM 20, 693–702.
- [Garland, a] Garland, M.
QSlim.
<http://mgarland.org/software/qslim.html>.

[[

- [Garland and Heckbert, 1997] Garland, M. and Heckbert, P. (1997).
Surface simplification using quadric error metrics.
In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques pp. 209–216, ACM Press/Addison-Wesley Publishing Co.
- [Garland et al., 2001] Garland, M., Willmott, A. and Heckbert, P. (2001).
Hierarchical face clustering on polygonal surfaces.
In Proceedings of the 2001 symposium on Interactive 3D graphics pp. 49–58, ACM.
- [Helmstaedter et al., 2011] Helmstaedter, M., Briggman, K. and Denk, W. (2011).
High-accuracy neurite reconstruction for high-throughput neuroanatomy.
Nature Neuroscience 14, 1081–1088.
- [Hoppe, 1996] Hoppe, H. (1996).
Progressive meshes.
In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques pp. 99–108, ACM.
- [Jeong et al., 2010] Jeong, W., Beyer, J., Hadwiger, M., Blue, R., Law, C., Vázquez-Reina, A., Reid, R., Lichtman, J. and Pfister, H. (2010).
Secrett and neurotrace: interactive visualization and analysis tools for large-scale neuroscience data sets.
IEEE Computer Graphics and Applications 30, 58.
- [Klein et al., 1996] Klein, R., Liebich, G. and Straßer, W. (1996).
Mesh reduction with error control.
In Visualization'96. Proceedings. pp. 311–318, IEEE.
- [Lichtman et al., 2008] Lichtman, J., Livet, J. and Sanes, J. (2008).
A technicolour approach to the connectome.
Nature Reviews Neuroscience 9, 417–422.

[[

- [Lindstrom and Turk, 1998] Lindstrom, P. and Turk, G. (1998).
Fast and memory efficient polygonal simplification.
In Visualization Conference, IEEE pp. 279–286, IEEE Computer Society.
- [Liu et al., 2008] Liu, L., Bajaj, C., Deasy, L., Low, D. and Ju, T. (2008).
Surface Reconstruction From Non-parallel Curve Networks.
Computer Graphics Forum 27, 155–163.
- [Peng et al., 2010] Peng, H., Ruan, Z., Long, F., Simpson, J. and Myers, E. (2010).
V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets.
Nature biotechnology 28, 348–353.
- [Wu and Kobbelt, 2005] Wu, J. and Kobbelt, L. (2005).
Structure recovery via hybrid variational surface approximation.
In Computer Graphics Forum vol. 24, pp. 277–284, Wiley Online Library.
- [Xie et al., 2010] Xie, J., Zhao, T., Lee, T., Myers, E. and Peng, H. (2010).
Automatic Neuron Tracing in Volumetric Microscopy Images with Anisotropic Path Searching.
Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010 6362, 472–479.
- [Yan et al., 2006] Yan, D., Liu, Y. and Wang, W. (2006).
Quadric surface extraction by variational shape approximation.
pp. 73–86, Springer.
- [Yan et al., 2009] Yan, D., Wintz, J., Mourrain, B., Wang, W., Boudon, F. and Godin, C. (2009).
Efficient and robust reconstruction of botanical branching structure from laser scanned points.
In Computer-Aided Design and Computer Graphics, 2009. CAD/Graphics' 09. 11th IEEE International Conference on pp. 572–575, IEEE.

[[

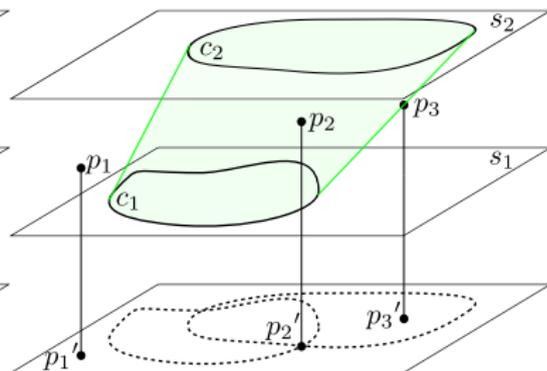
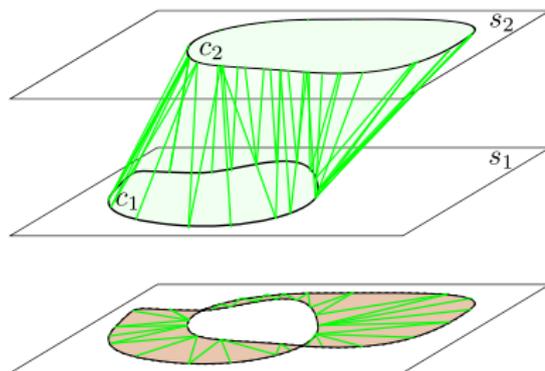
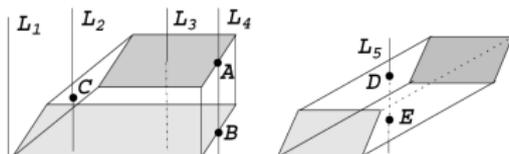
[Zhang et al., 2005] Zhang, Y., Bajaj, C. and Sohn, B. (2005).

3D finite element meshing from imaging data.

Computer methods in applied mechanics and engineering *194*, 5083–5106.

Claim

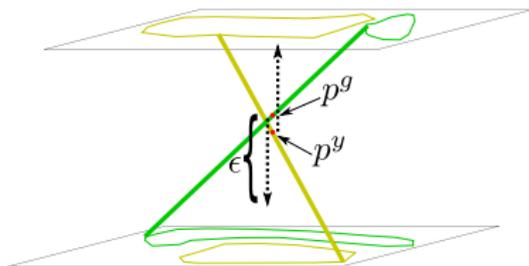
All intersections occur in *penumbral regions*. A point's *penumbral contour* is the contour whose projection contains the projected point.



A *conflict point* is a point of intersection. Somewhat more formally:

Definition

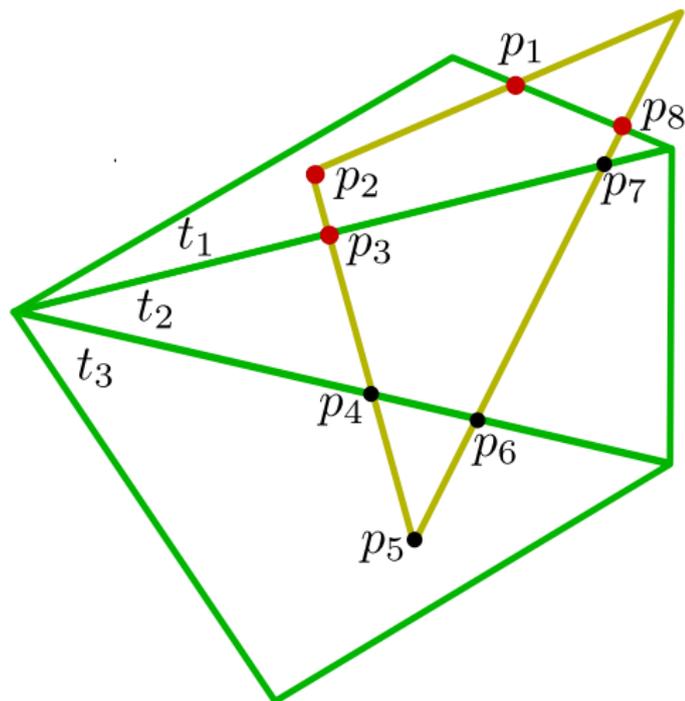
Point p^g is called a *conflict point* if there is some point p^y such that the projections are equal ($p^{y'} = p^{g'}$) and p^y is closer to p^g 's penumbral contour than p^g is.



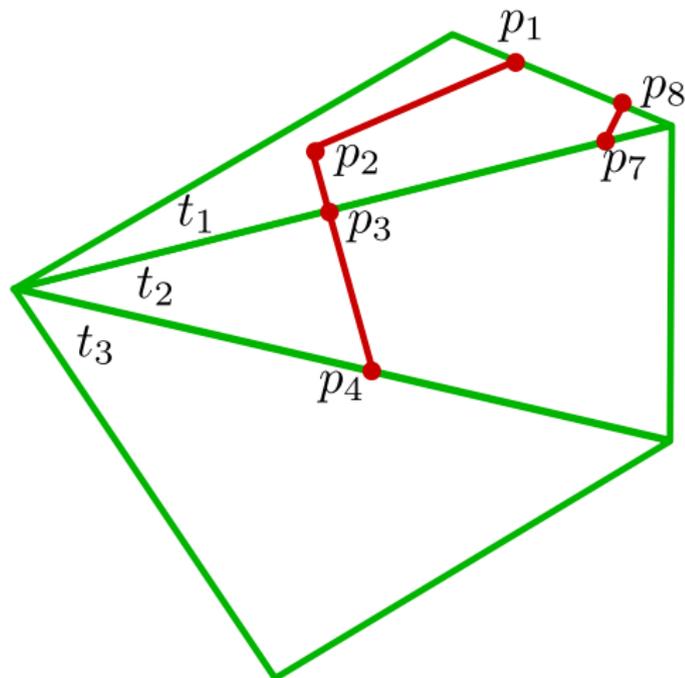
Claim

Two components C^g and C^y intersect if and only if there is at least one conflict point on the surface of either component.

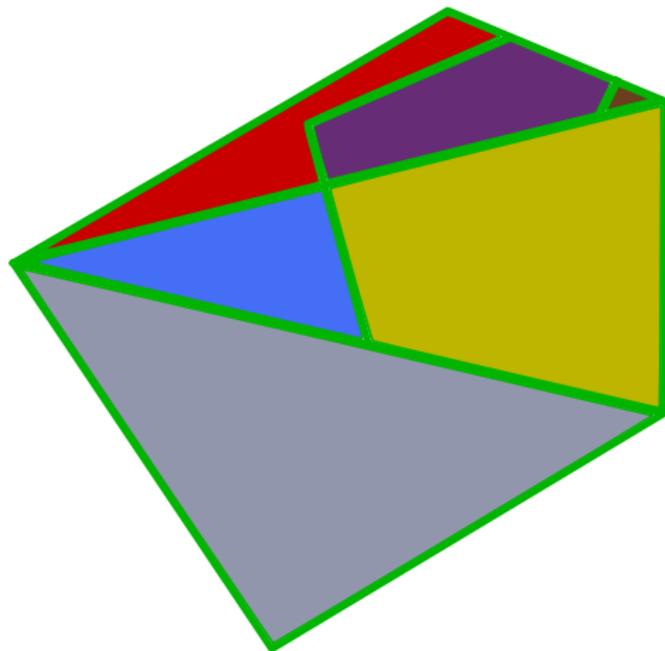
- 1 Detect conflict points.
- 2 Trace paths between conflict points along edges of yellow tile. We call these *cut paths*.
- 3 Use original tiles and cut paths to induce new polygons.
- 4 Triangulate polygons and move conflict points along z-axis.



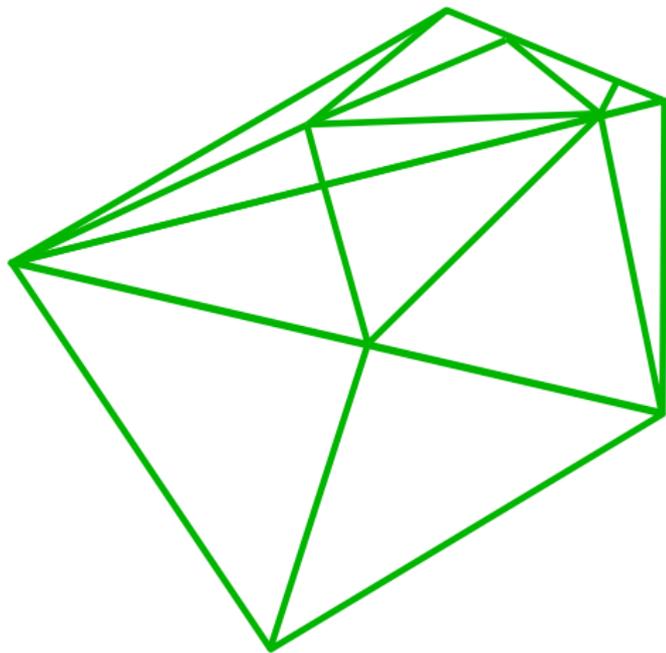
- 1 Detect conflict points.
- 2 Trace paths between conflict points along edges of yellow tile. We call these *cut paths*.
- 3 Use original tiles and cut paths to induce new polygons.
- 4 Triangulate polygons and move conflict points along z-axis.



- 1 Detect conflict points.
- 2 Trace paths between conflict points along edges of yellow tile. We call these *cut paths*.
- 3 Use original tiles and cut paths to induce new polygons.
- 4 Triangulate polygons and move conflict points along z-axis.



- 1 Detect conflict points.
- 2 Trace paths between conflict points along edges of yellow tile. We call these *cut paths*.
- 3 Use original tiles and cut paths to induce new polygons.
- 4 Triangulate polygons and move conflict points along z-axis.



- Algorithm is $O(n^2)$ where n is the number of tiles.
 - Average case is closer to $n \log n$ complexity of sweep line algorithm as large majority of 2D intersections are not conflict points.
- Original contours remain unchanged – only makes changes in interpolated data between slices
- Topologically correct and water tight
- Generates large number of extra triangles in intersecting regions

Separating by a given delta

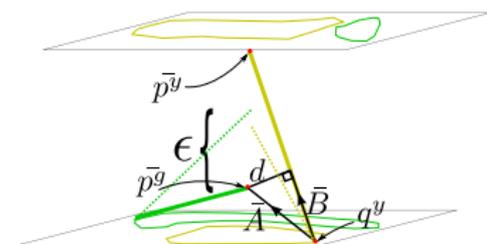
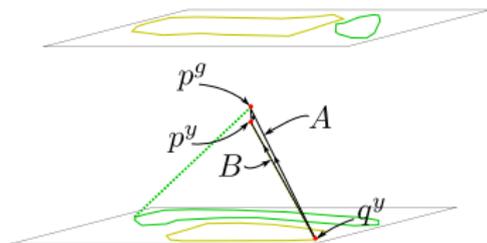
$$d = \frac{|\bar{\mathbf{A}} \times \bar{\mathbf{B}}|}{|\bar{\mathbf{B}}|}$$

Substituting for $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$:

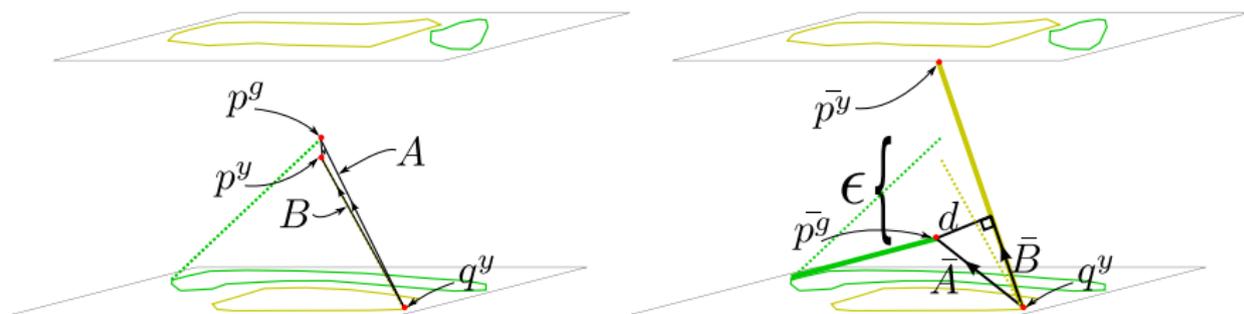
$$\begin{aligned} d^2 = & ((A_y(B_z - \epsilon) - (A_z + \epsilon)B_y)^2 \\ & + ((A_z + \epsilon)B_x - A_x(B_z - \epsilon))^2 \\ & + (A_xB_y - A_yB_x)^2) / (B_x^2 + B_y^2 + (B_z + \epsilon)^2) \end{aligned}$$

After collecting ϵ :

$$\begin{aligned} 0 = & \epsilon^2((A_y + B_y)^2 + (A_x + B_x)^2 - d^2) \\ & + \epsilon(2)((A_x + B_x)(A_zB_x - A_xB_z) \\ & - (A_y + B_y)(A_yB_z - A_zB_y) - d^2A_z) \\ & + (A_yB_z - A_zB_y)^2 + (A_zB_x - A_xB_z)^2 \\ & + (A_xB_y - A_yB_x)^2 - d^2(B_x^2 + B_y^2 + B_z^2) \end{aligned}$$



Separating by a given delta

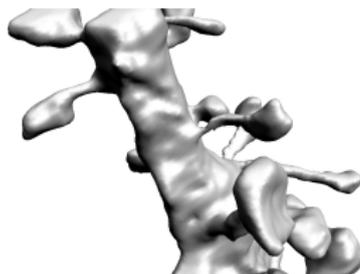


Theorem

$$\epsilon < |p^g - \mathcal{L}(p^g)| \text{ and } \epsilon < |p^y - \mathcal{L}(p^y)|$$

Idea of proof: as points approach original contours, which are separated by d , the chords will be separated by at least d in the limit.

- Extract skeleton from C using mesh contraction in [Au et al., 2008].
- Fit cylinder to each segment in skeleton



- Problem: $\sum_{c \in M'_K} \text{Volume}(c) > \text{Volume}(C)$

Our approach is to partition the *surface*, fit cylinders, then match volumes.

- Given a single component $C \in D$, we seek a partition P of C and a topology-preserving bijective function $f : M_K \mapsto P$ where M_K is the set of cylinders.
- Quality criteria for P :
 - Let P_e be the region that element $e \in R$ is assigned to in the partition P . If $e_i \in R$ and $e_j \in R$ are in different branches, then $P_{e_i} \neq P_{e_j}$.
 - Elements in a section between branches all belong to the same $r \in P$.
- Quality criteria for M_K : $\forall c \in M_K$
 - $\text{Area}(c) = \text{Area}(f(c))$
 - $\sum_{c \in M_K} \text{Volume}(c) = \text{Volume}(R)$

Cohen-Steiner et al [Cohen-Steiner et al., 2004] proposed a variational approach to shape approximation

- Represent each region P_i of a partition P with a tuple $\mathcal{P}_i = (X_i, \vec{N}_i)$ where X_i (resp. \vec{N}_i) is the “average” point (normal). Proxies are planes.
- Define $\Pi_i(\cdot)$ to be the orthogonal projection onto proxy plane \mathcal{P}_i .
- Define an error metric:

$$L^2(R_i, \mathcal{P}_i) = \iint_{x \in R_i} \|x - \Pi_i(x)\|^2 dx$$

- Optimal shape proxies: a set \mathcal{P} of proxies \mathcal{P}_i that minimizes

$$L^2(R, \mathcal{P}) = \sum_{i=1..k} L^2(R_i, \mathcal{P}_i)$$

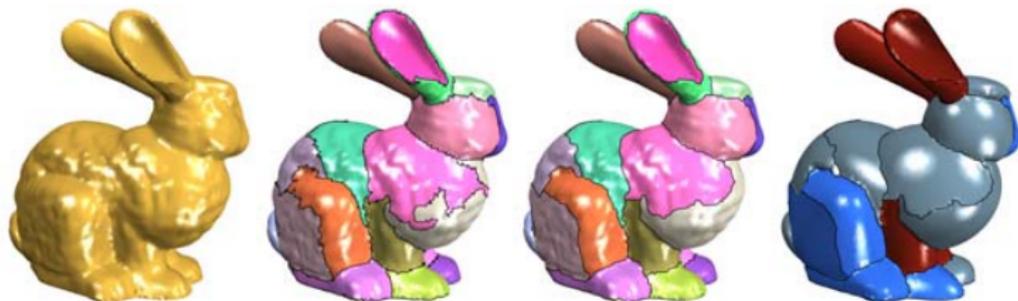


A partition R and proxy set \mathcal{P} must be found. Use Lloyd's algorithm (similar to cluster k-means, CVT, etc).

- Partition randomly. Choose k seeds and flood-fill.
- Find \mathcal{P}_i for each partition. X_i is the average of the barycenters of the triangles. \vec{N}_i is found using the covariance matrix of the normals of the triangles.
- Repartition: Each triangle T_j gets assigned to proxy \mathcal{P}_i that minimizes $L^2(T_j, \mathcal{P}_i)$.
- Iterate to convergence



- Other works have extended the proxies used to cylinders [Yan et al., 2009], spheres, cylinders and rolling-ball blend patches [Wu and Kobbelt, 2005] and general quadrics [Yan et al., 2006].
- Our approach will be to use cylinders



Algorithm:

- 1 Produce an initial partition of the surface mesh triangles of C .
- 2 Fit a cylinder to each partition.
- 3 Repartition the triangles.
- 4 Iterate over (2) and (3) until convergence.

Initial partition: Choose a random triangle $t \in T$. Set $T := T - t$. Iteratively choose adjacent triangle t_j in the direction of maximum curvature. Set $T := T - t_j$. If set of chosen triangles $|c_i| > |C|/n$ then increment i and iterate.

Algorithm:

- 1 Produce an initial partition of the surface mesh triangles of C .
- 2 Fit a cylinder to each partition.
- 3 Repartition the triangles.
- 4 Iterate over (2) and (3) until convergence.

Cylinder fitting: Get the minimum curvature directions $\{\gamma_{\min}\}$ for each vertex. Use the average direction as the cylinder axis d . Now project the barycenters of the triangles to the plane with normal d . Then fit a circle to the projected points, finalizing the cylinder parameters. The circle is fit by minimizing the error $E(T_j, c_k) = \sum_{i=1}^3 g(T_j^{i,\perp}, c_k) |T_j|$ where $T_j^{i,\perp}$ is the projection of the i th vertex of triangle T_j . $|T_j|$ is the area of triangle T_j and $g(x, y, c_k) = A(x^2 + y^2) + Bx + Cy + D = 0$.

Algorithm:

- 1 Produce an initial partition of the surface mesh triangles of C .
- 2 Fit a cylinder to each partition.
- 3 Repartition the triangles.
- 4 Iterate over (2) and (3) until convergence.

Repartitioning: Distance from each triangle T_i to each cylinder c_k is computed and the T_i is assigned the closest cylinder. Distance is computed as

$$d(T_i, c_k) = (1/6)(d_1^2 + d_2^2 + d_3^2 + d_1d_2 + d_1d_3 + d_2d_3)|T_i|$$

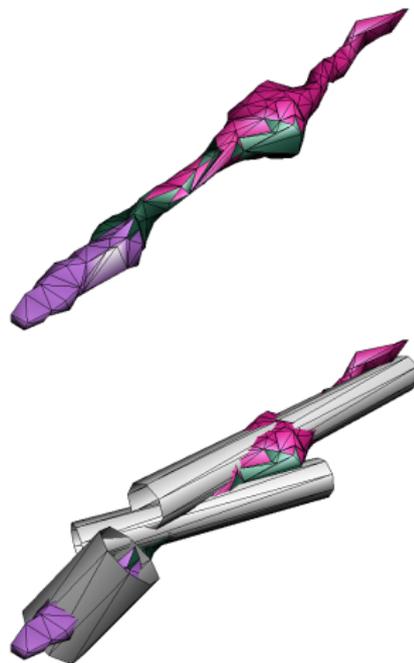
where d_j is the distance of the j^{th} vertex of T_i to c_k and $|T_i|$ is the area of T_i .

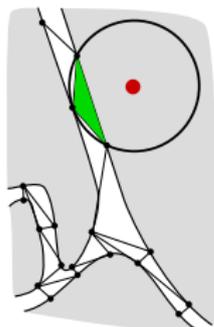
Algorithm:

- ① Produce an initial partition of the surface mesh triangles of C .
- ② Fit a cylinder to each partition.
- ③ Repartition the triangles.
- ④ Iterate over (2) and (3) until convergence.

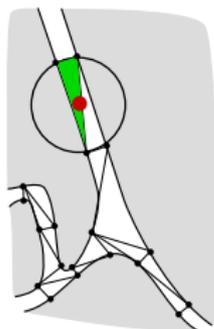
Convergence: Convergence is reached when no triangles change partitions in the repartitioning step.

- Clearly our implementation needs work.
- Need to fix bugs and add topology constraint.
- Would this approach possibly work for skeletonization?





Not well-aligned



Well-aligned

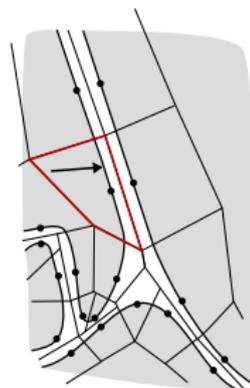
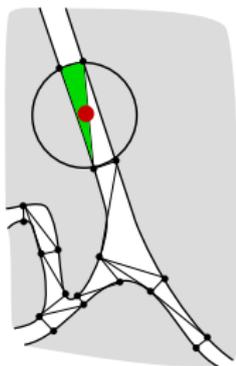
Definition: well-aligned

Let P be a sampling of points on $\partial\Omega$ and let $T(P)$ be a constrained Delaunay triangulation. Let S be the set of all Delaunay spheres in $T(P)$ and let $\mathcal{C}(s)$ be the centroid of sphere s . P is well-aligned if $T(P)$ is a conforming Delaunay triangulation and $\forall s \in S, \mathcal{C}(s) \in \Omega$.

Conjecture

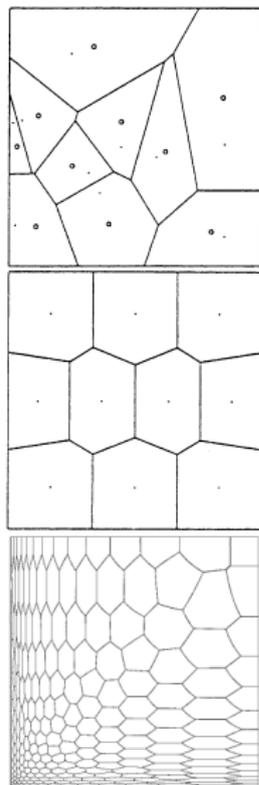
Given well-aligned points P on $\partial\Omega$, the Voronoi diagram of P is a decomposition that satisfies criterion C2.

We could just sample points to be no more than $lfs(p)$ apart (where $lfs(p)$ is the local feature size at p as defined in [Amenta et al., 1998]), but this would violate criterion C3.



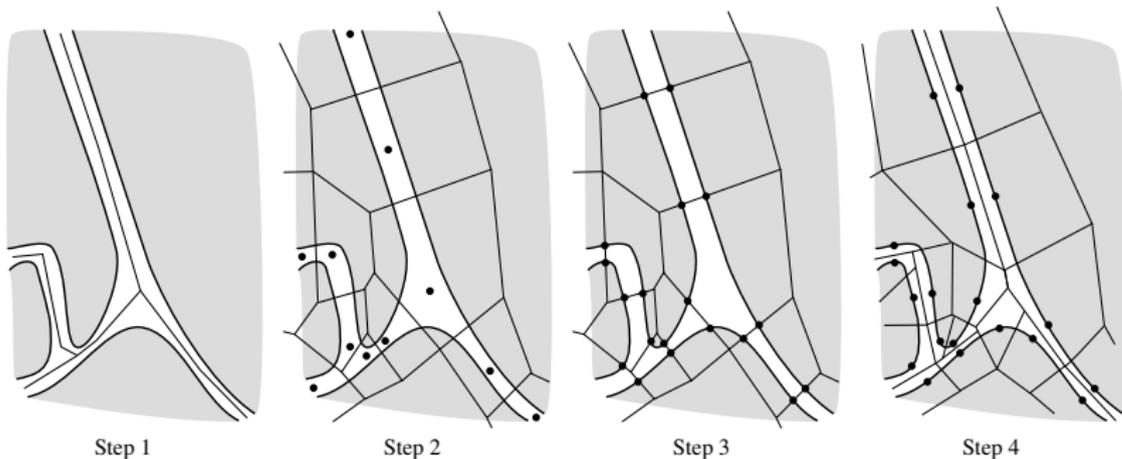
Well-aligned points yield...

...cells that meet criterion 2



- The centroidal voronoi tessellation (CVT) is a voronoi diagram where each generating point lies at the center of mass of its voronoi cell.
- Points can be resampled iteratively using Lloyd's algorithm to generate a CVT.
- We will require an anisotropic CVT (ACVT). One approach is to use a directional distance function [Du and Wang, 2005].

- 1 Compute the medial sheet
- 2 Resample points on the sheet using an anisotropic version of the CVT
- 3 Intersect all edges of the CVT with $\partial\Omega$ and induce new surface vertices at the intersections
- 4 Compute the Voronoi diagram using these new surface points



- An earlier version of this algorithm has been implemented by Pan Hao in Professor Wenping Wang's lab.
- The CVT run on the medial sheet is isotropic

