Barycentric Subdivision Meshes in Computational Solid Mechanics

Julian Rimoli
School of Aerospace Engineering
Georgia Institute of Technology

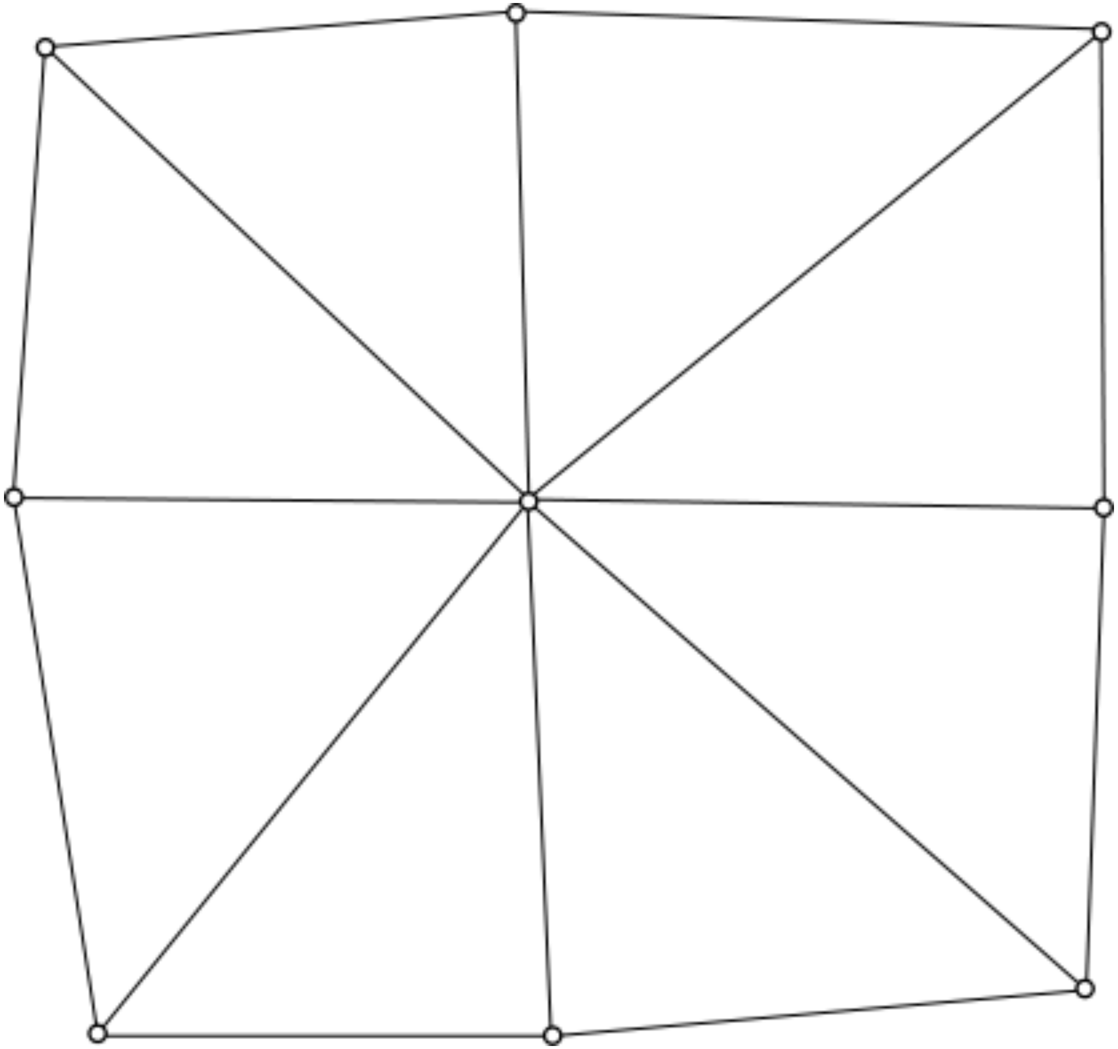
NSF Workshop on Barycentric Coordinates in
Geometry Processing and Finite/Boundary Element Methods
Columbia University, New York, July 25-27 2012

Georgia Institute of Technology
Daniel Guggenheim
School of Aerospace Engineering

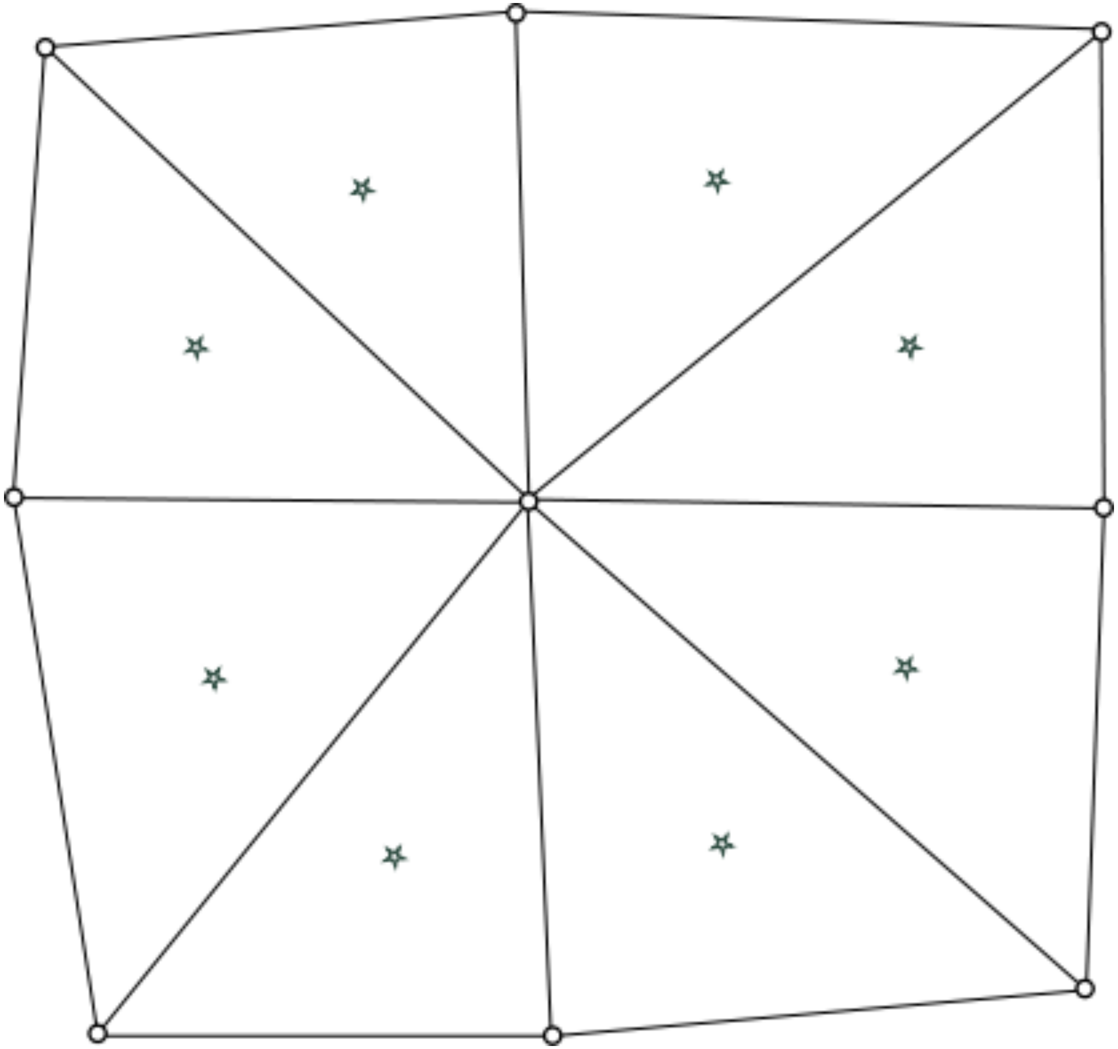
Outline

- Barycentric subdivision and barycentric dual of simplicial complexes
- Applications in computational solid mechanics:
  - Geometric representation of polycrystals
  - Mesh generation for cohesive zone modeling of crack propagation problems

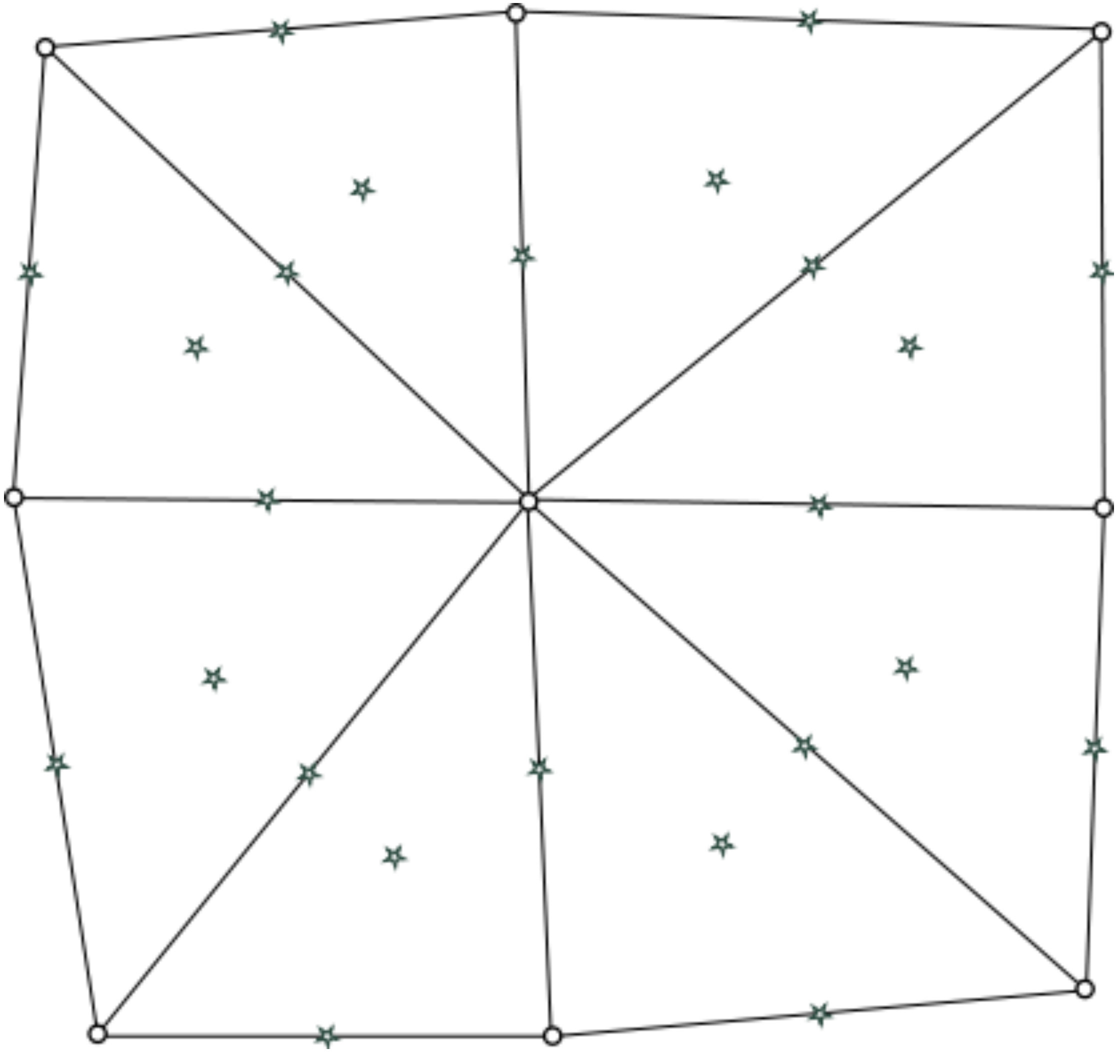
Barycentric subdivision of simplicial complexes



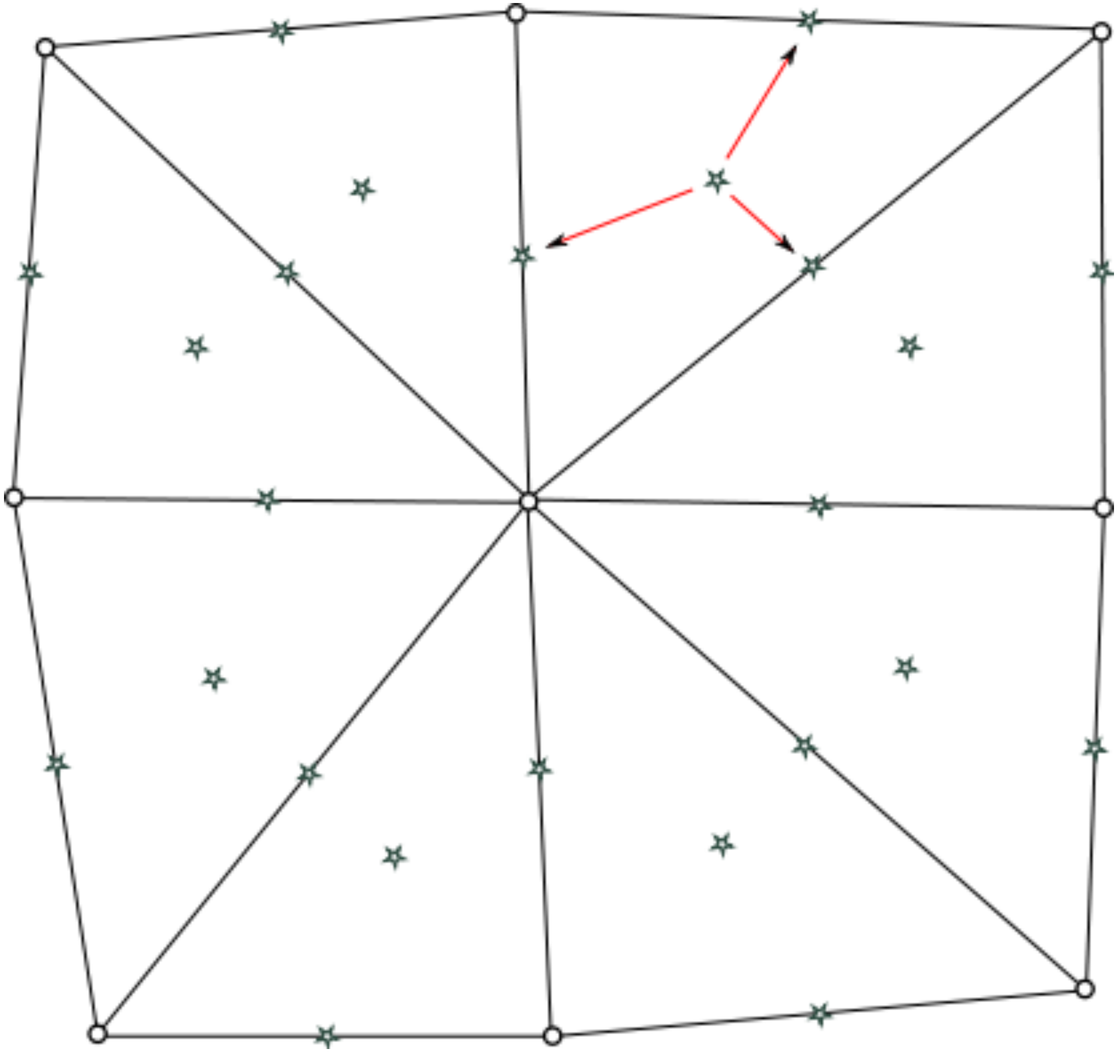
Barycentric subdivision of simplicial complexes



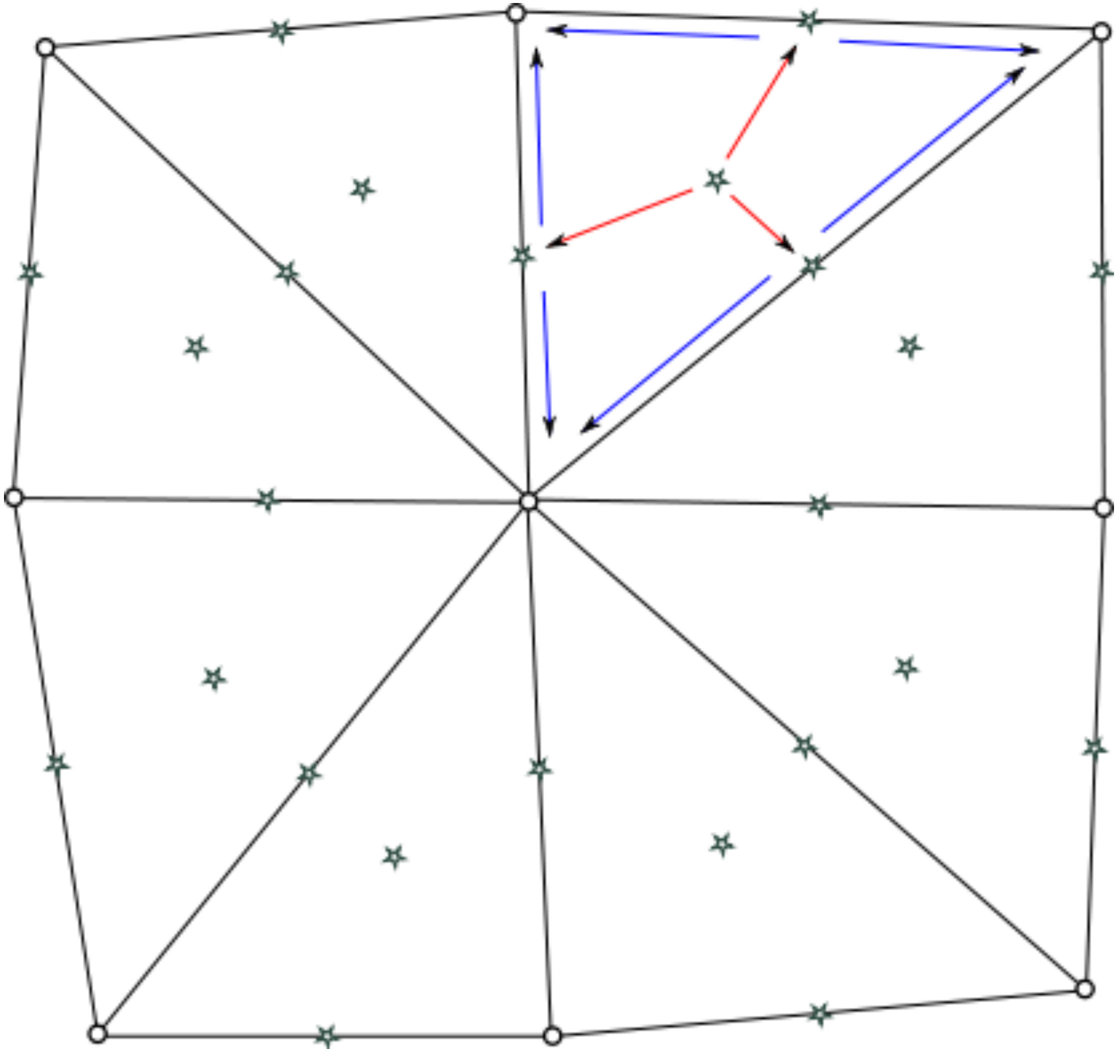
Barycentric subdivision of simplicial complexes



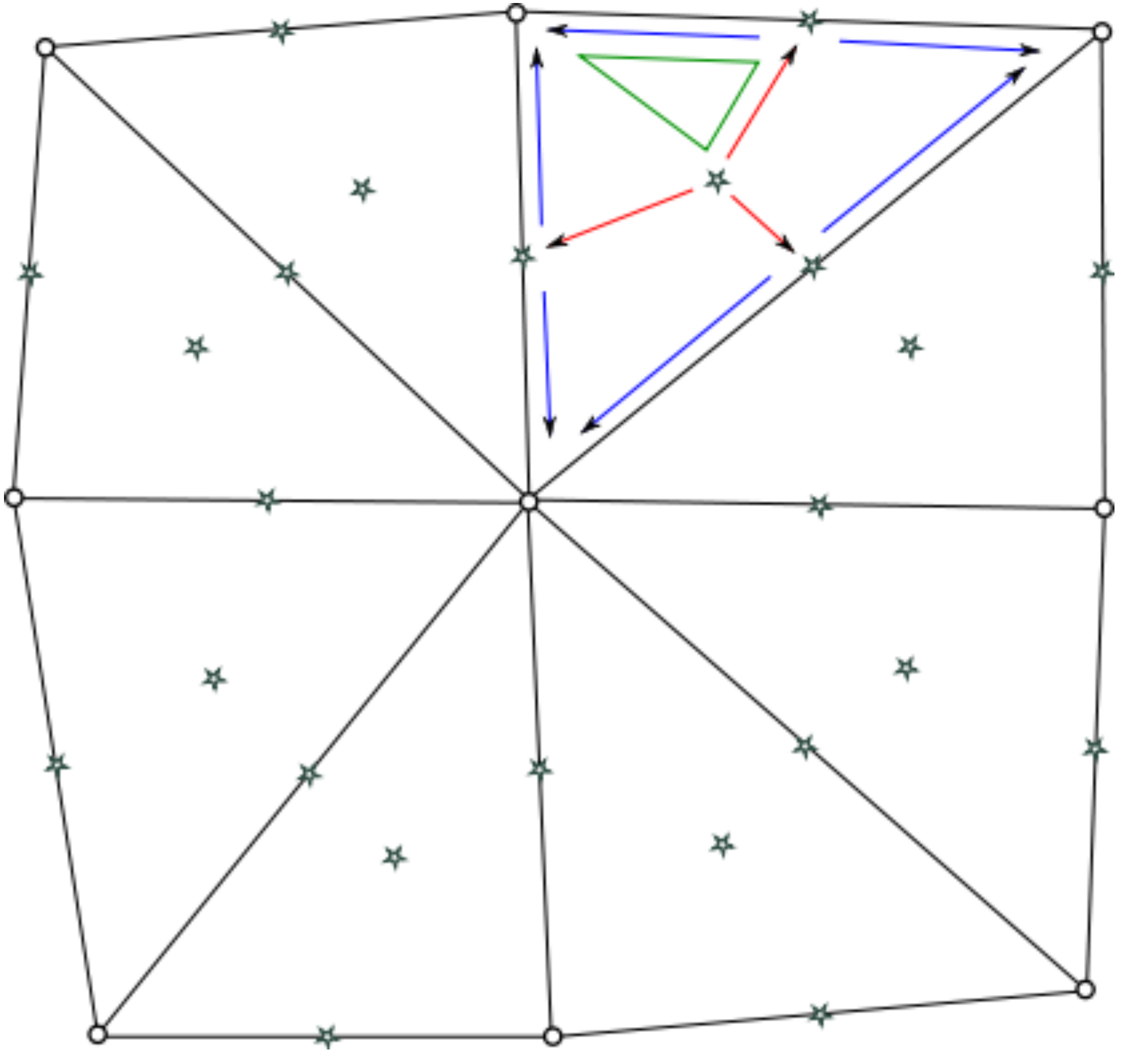
Barycentric subdivision of simplicial complexes



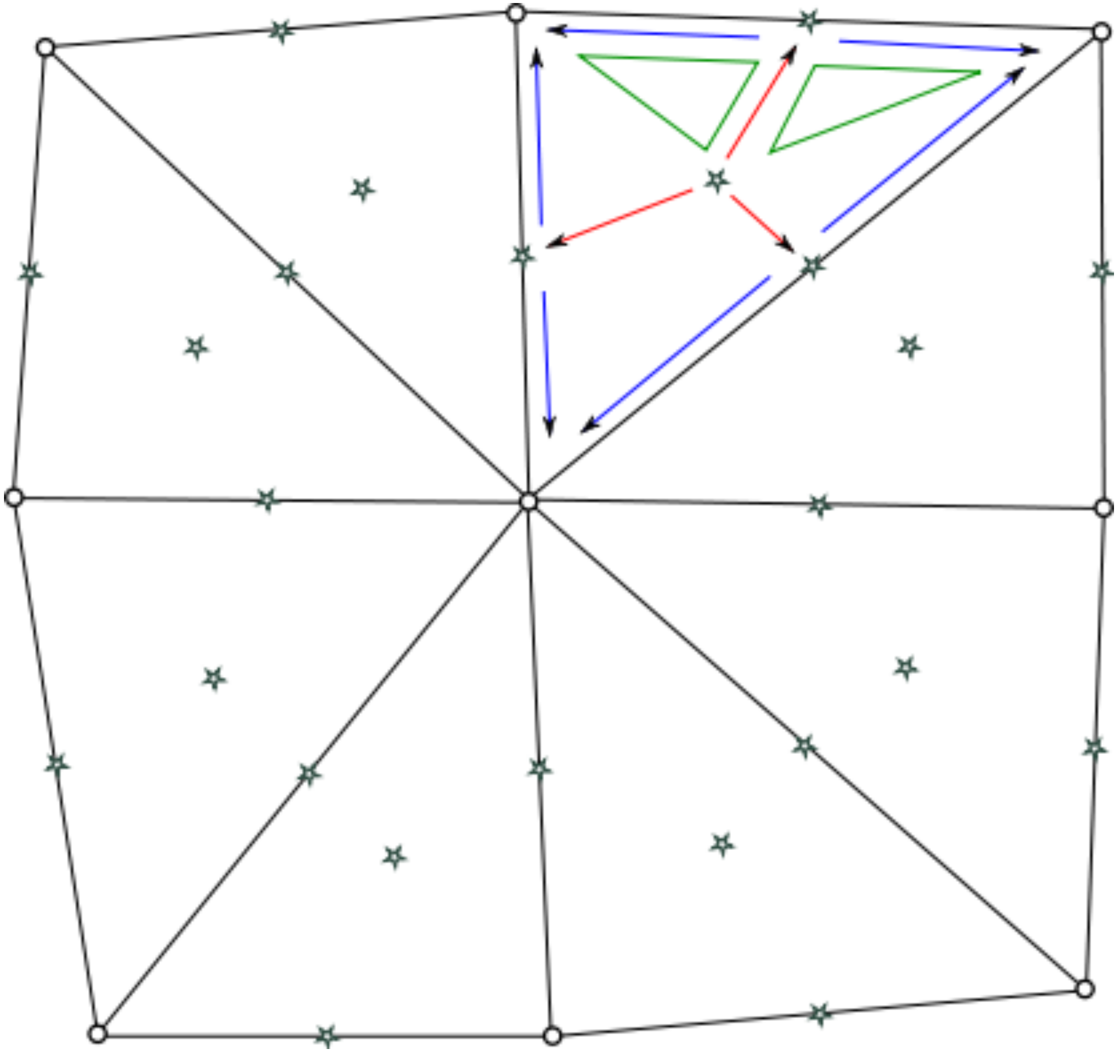
Barycentric subdivision of simplicial complexes



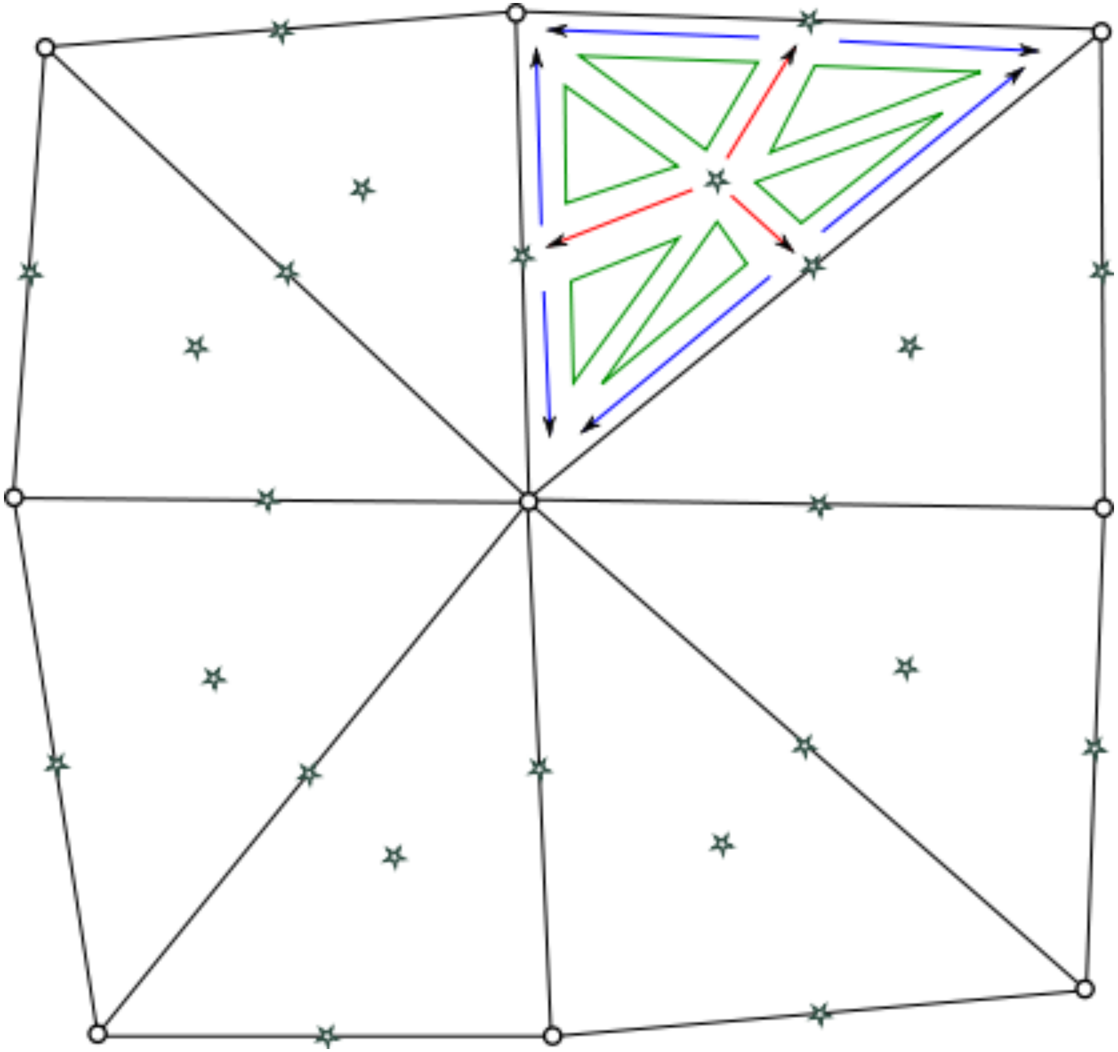
Barycentric subdivision of simplicial complexes



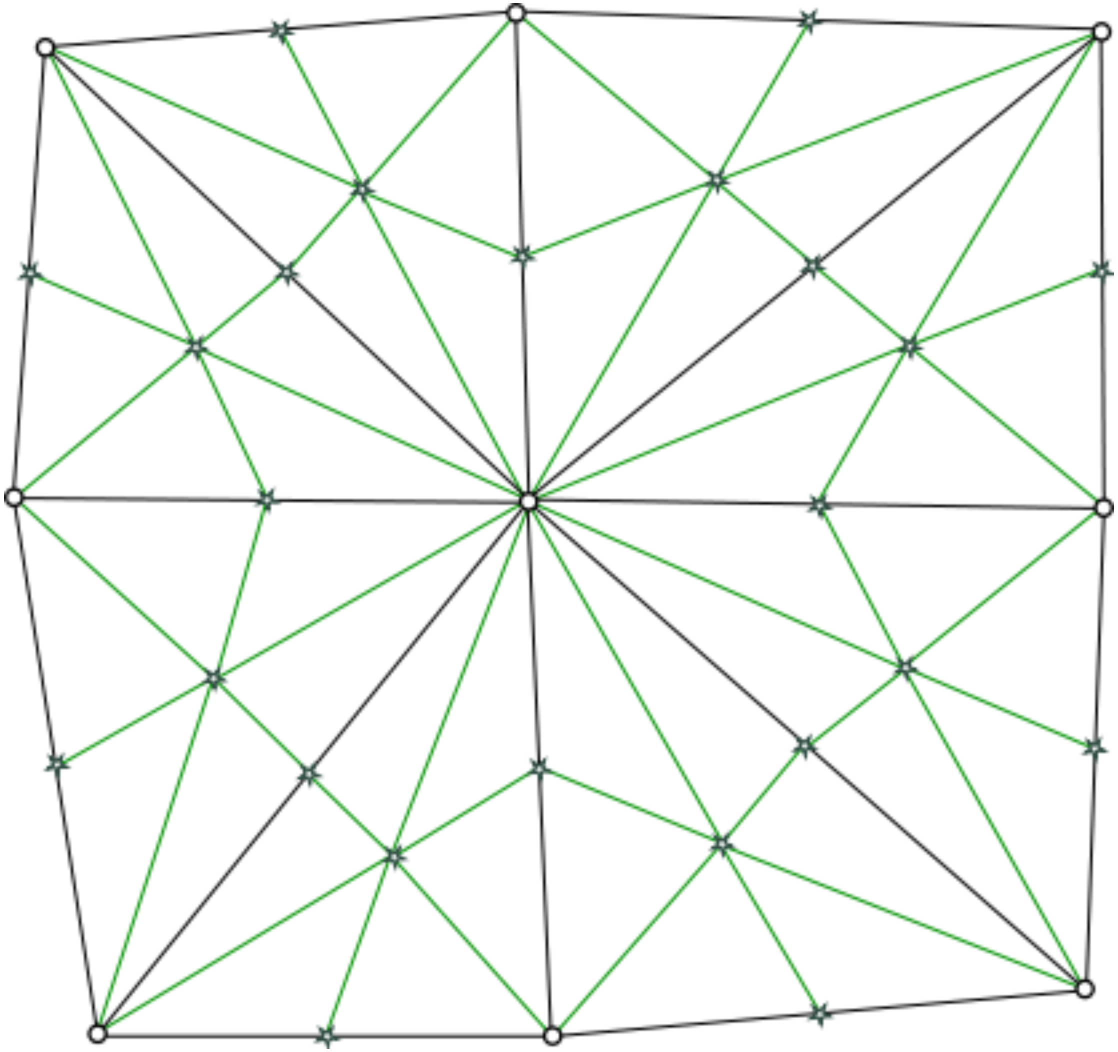
Barycentric subdivision of simplicial complexes



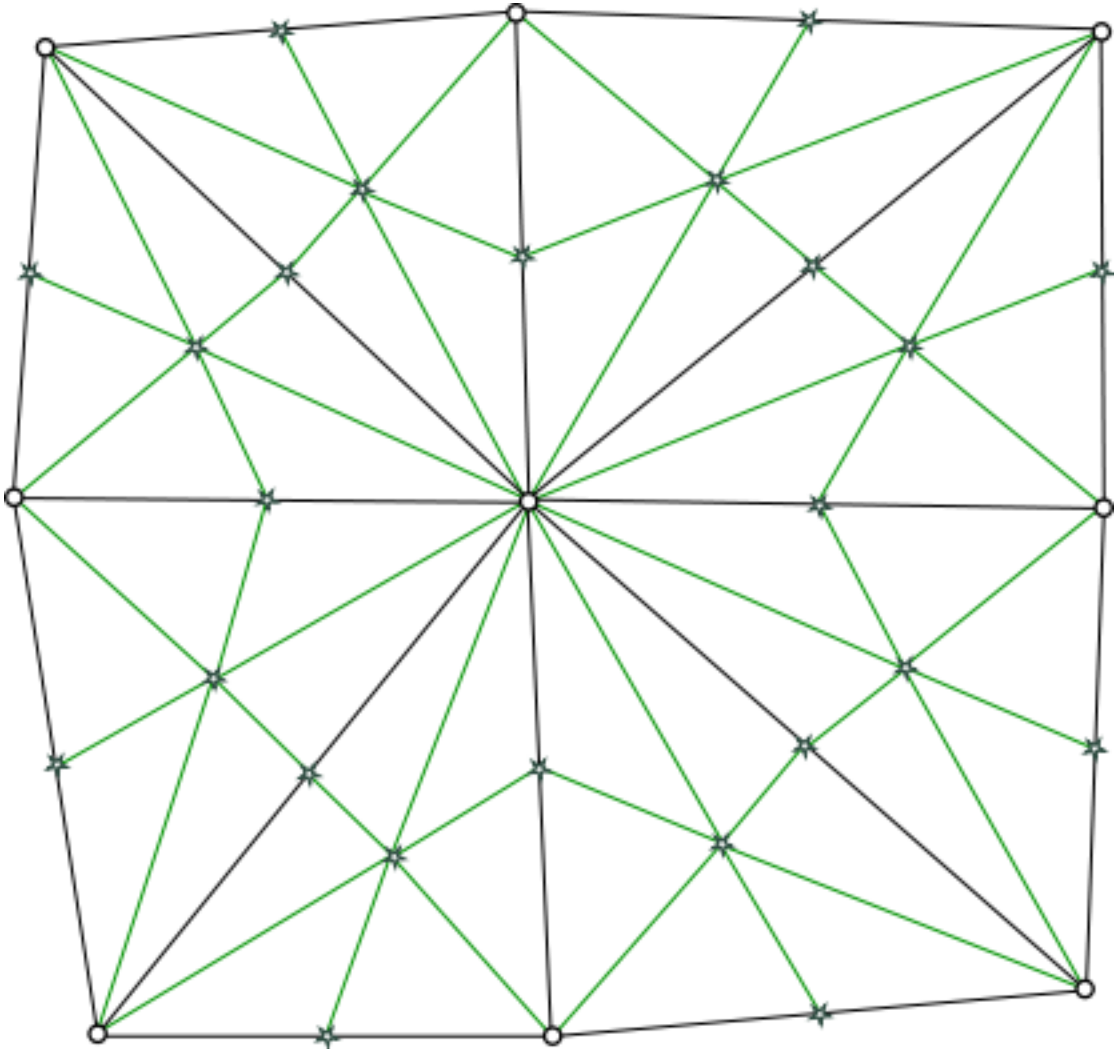
Barycentric subdivision of simplicial complexes



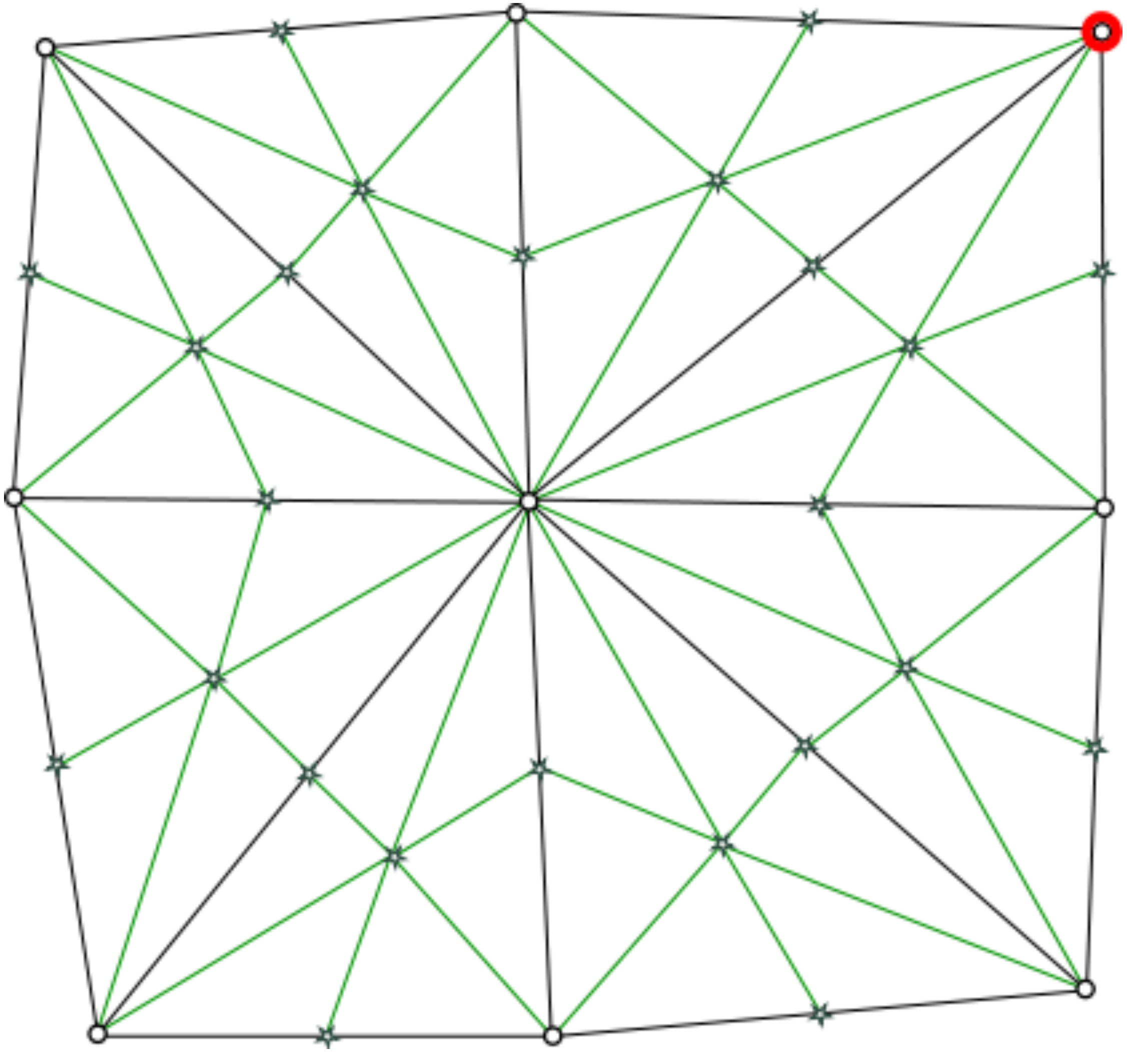
Barycentric subdivision of simplicial complexes



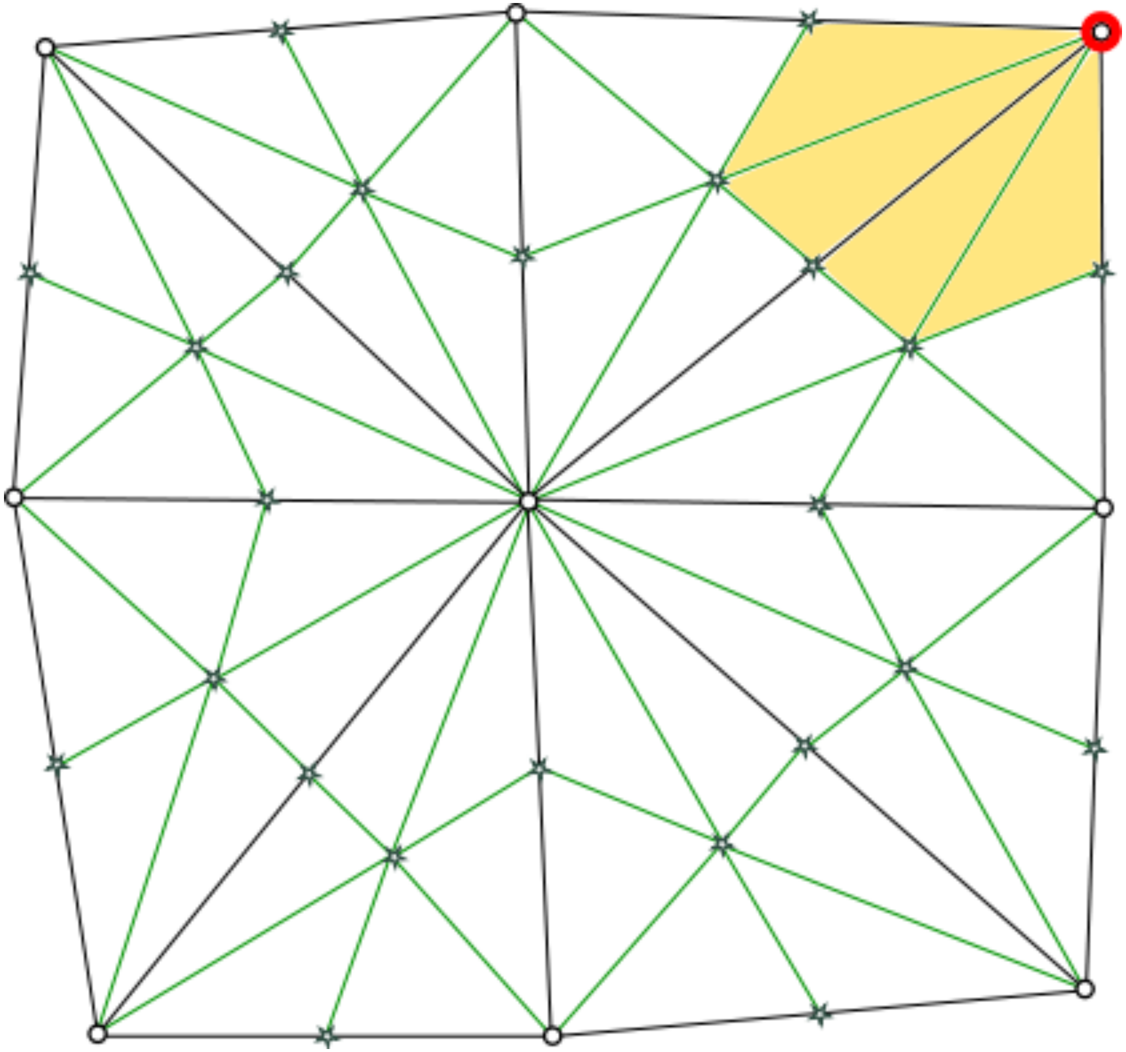
Barycentric Dual



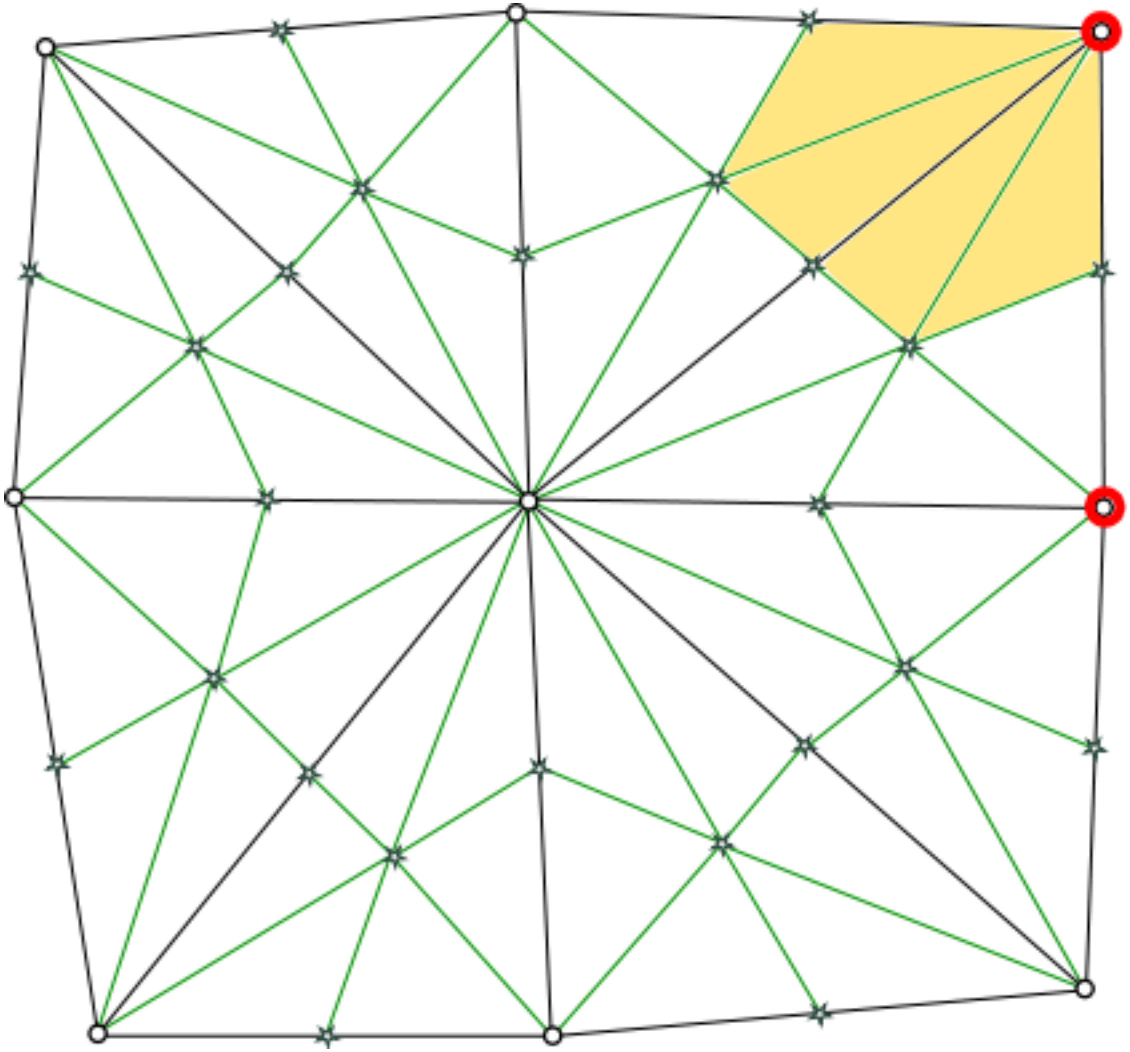
Barycentric Dual



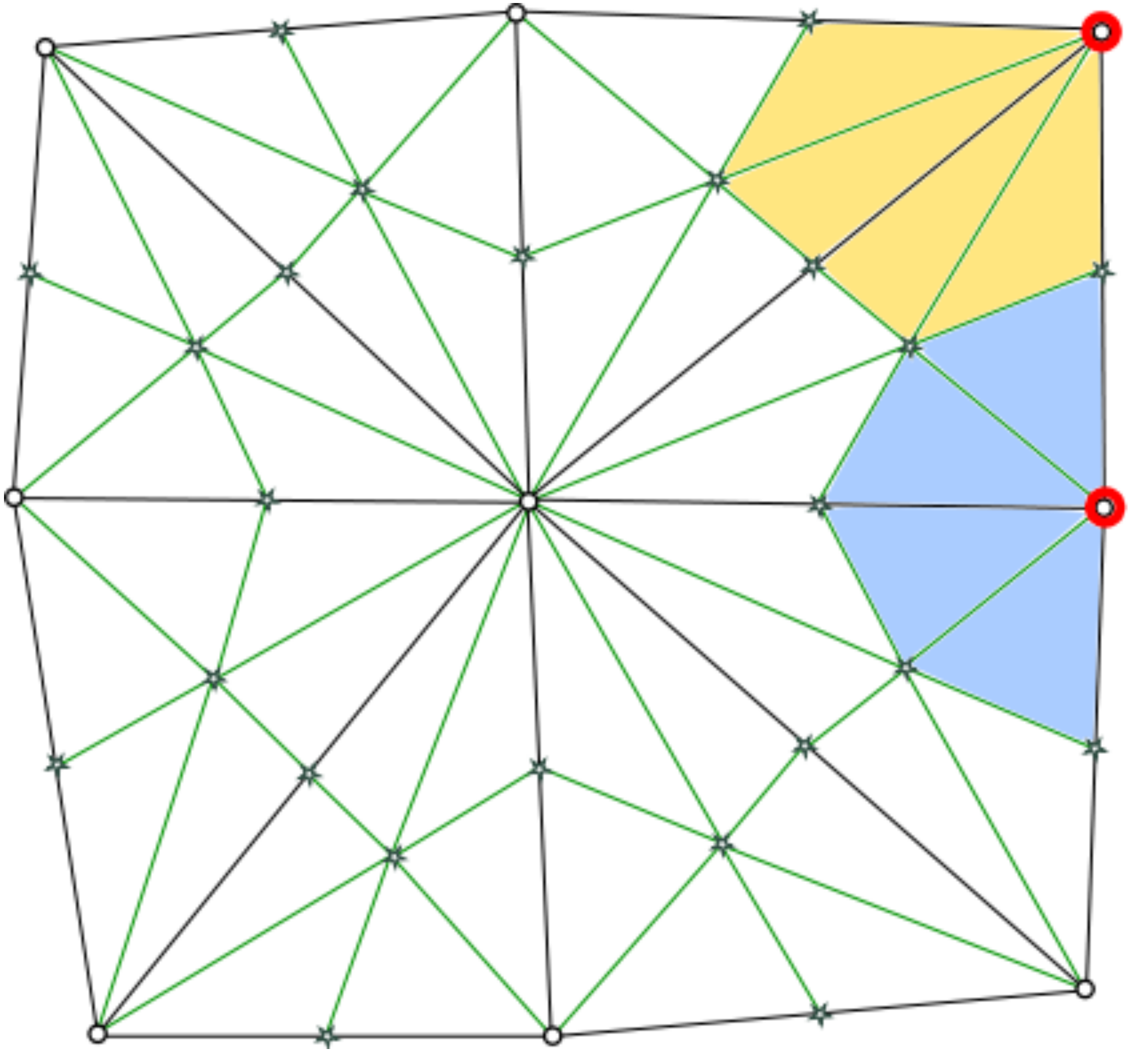
Barycentric Dual



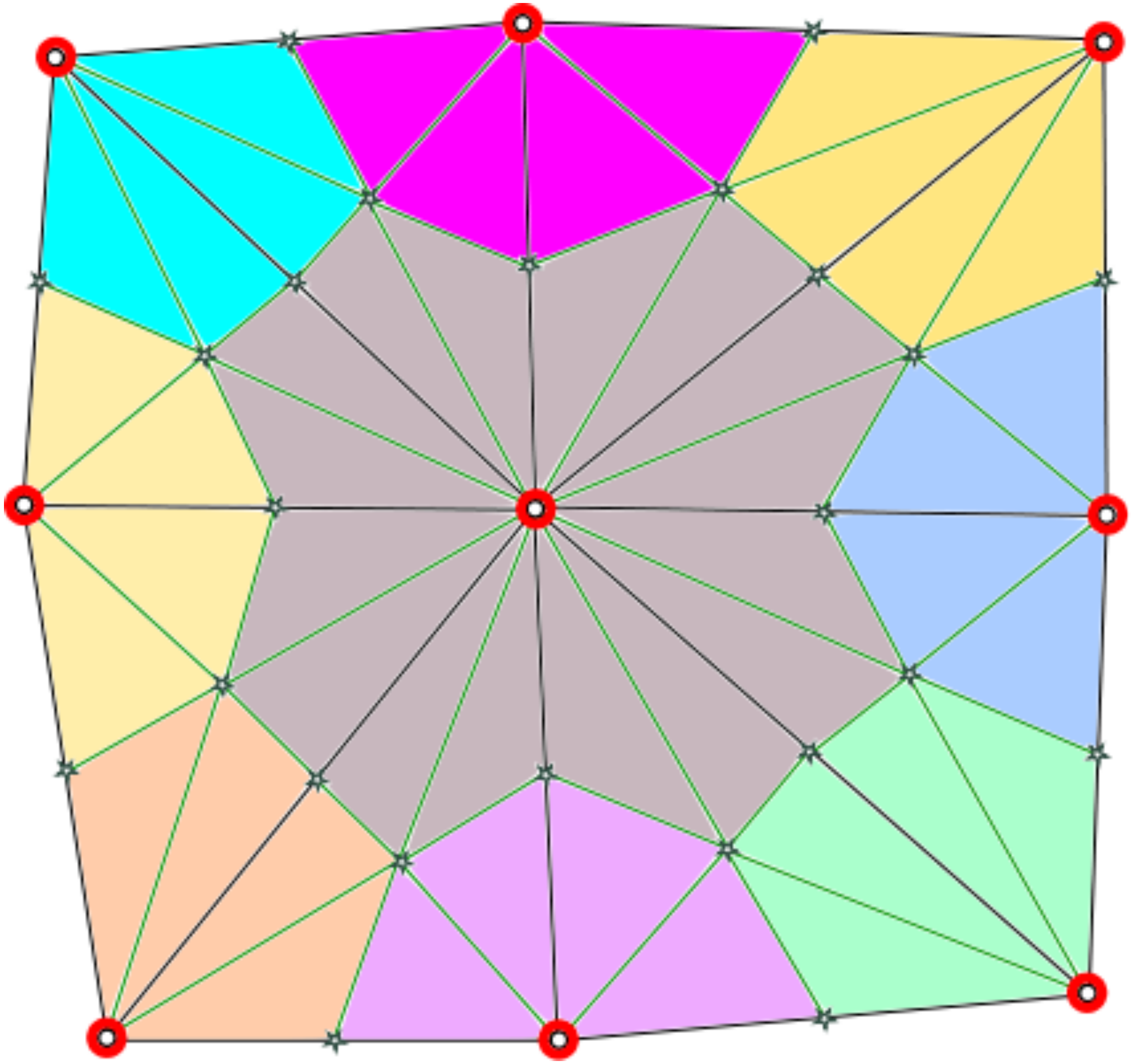
Barycentric Dual



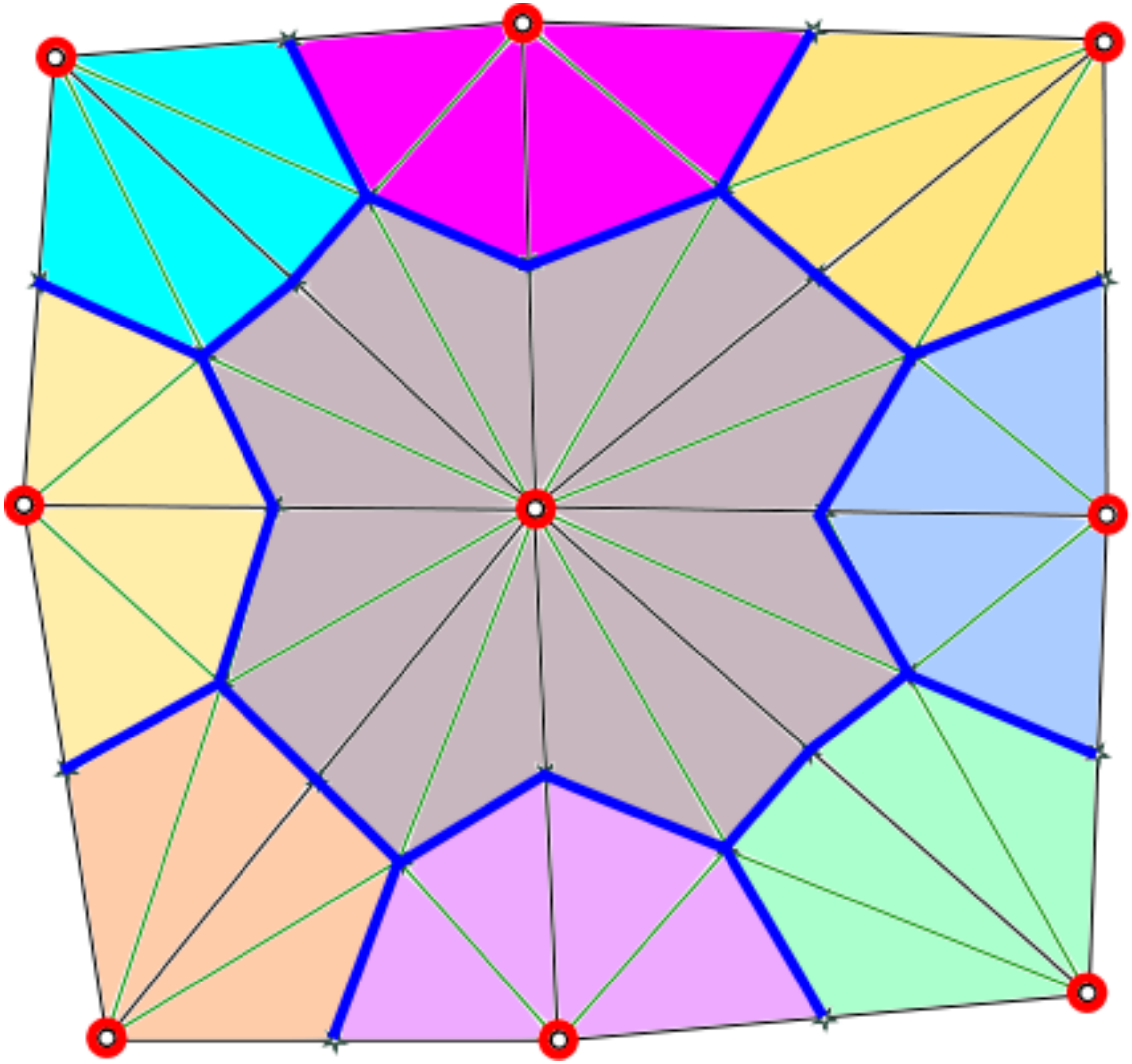
Barycentric Dual



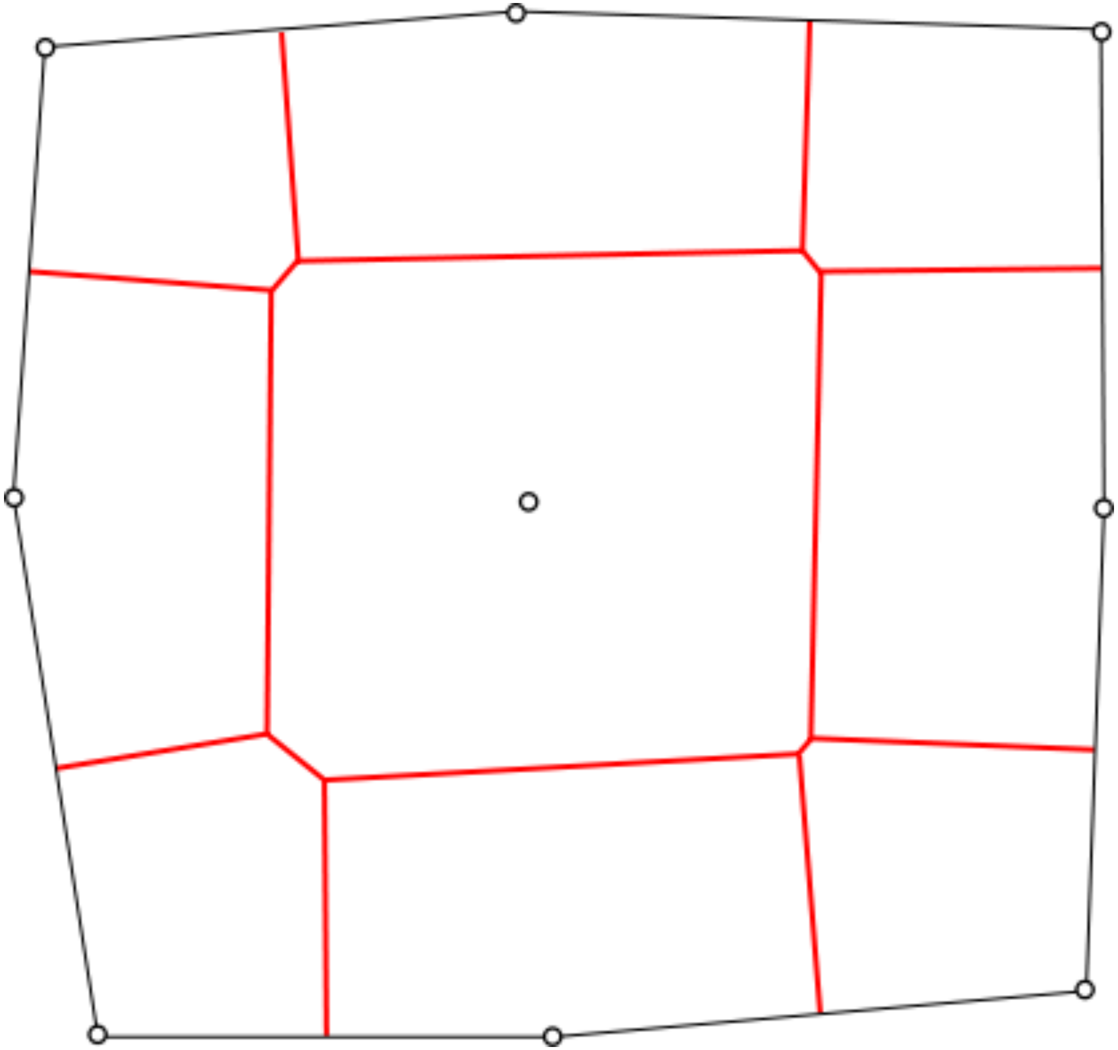
Barycentric Dual



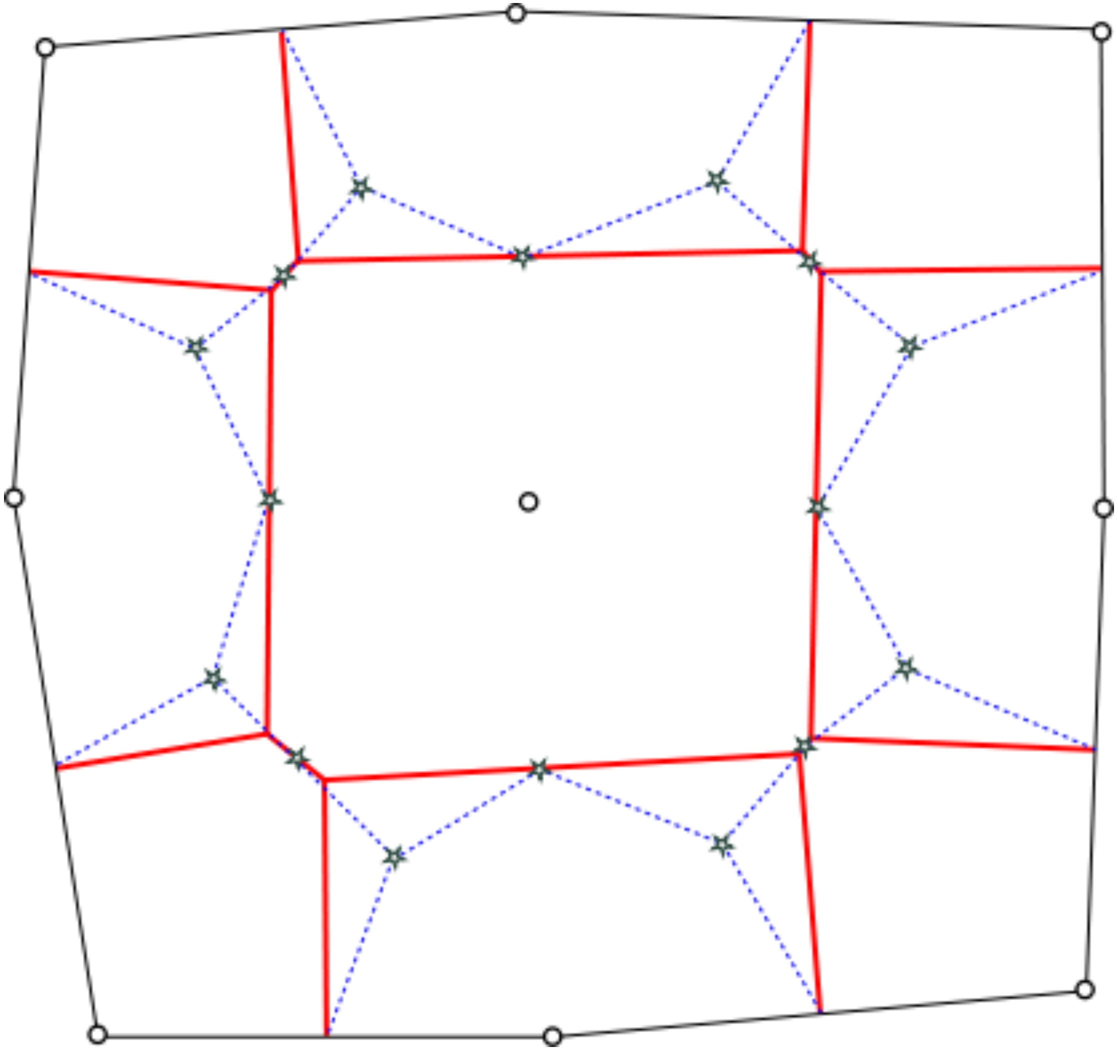
Barycentric Dual



Barycentric dual vs Voronoi dual



Barycentric dual vs Voronoi dual

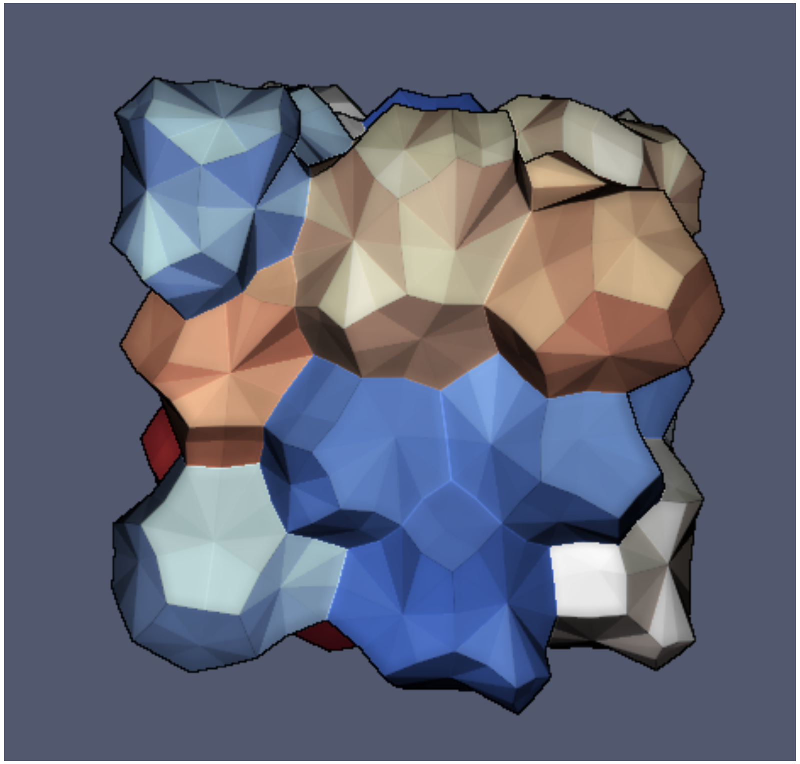


Remarks on barycentric subdivisions and duals

- Barycentric subdivision & dual are purely topological operations
- Barycentric subdivision performed in O(Nel) and barycentric dual in O(Nno)
- Shortest edge of barycentric subdivision/dual is bounded by worst element in seed mesh
- Mesh quality decreases due to barycentric subdivision

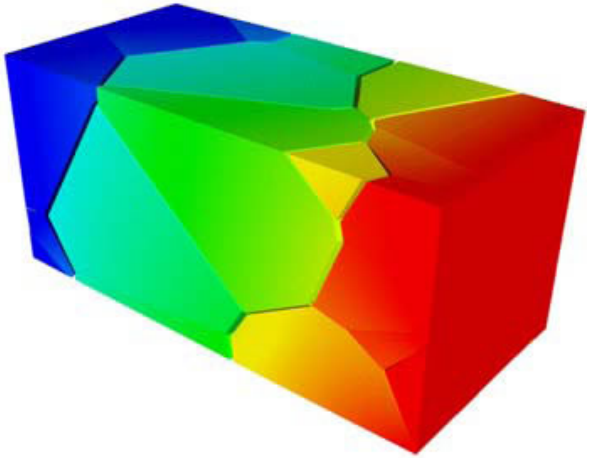
Application 1: geometric representation of polycrystals

- Direct numerical simulation (DNS) of polycrystalline materials (common practice in CSM) requires a geometric representation of the individual grains.
- Due to limited computational resources, early polycrystal DNS approaches resorted to geometrical simplifications (2D, cubic grains, tetrakaidecahedral grains, etc).
- Such simplifications may be adequate for predicting texture evolution in some materials but tend to break down when local strain heterogeneities are of interest.

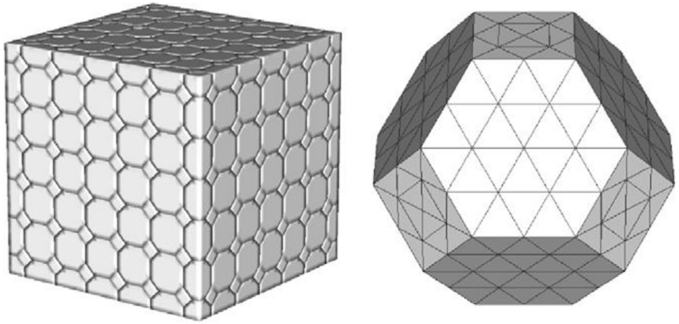


Common approaches

- Voronoi diagram of specified grain seeds:
  - Not straightforward for arbitrary geometries.
  - Hard to get conforming tetrahedral meshes from the tesselation.
- Space-filling tetrakaidecahedrons:
  - Well suited for modeling polycrystalline cuboids.
  - Extremely difficult to generate arbitrary geometries.



Polycrystal generated using Voronoi tesselation.
(http://www.cs.sandia.gov/materials_methods/news1.html)



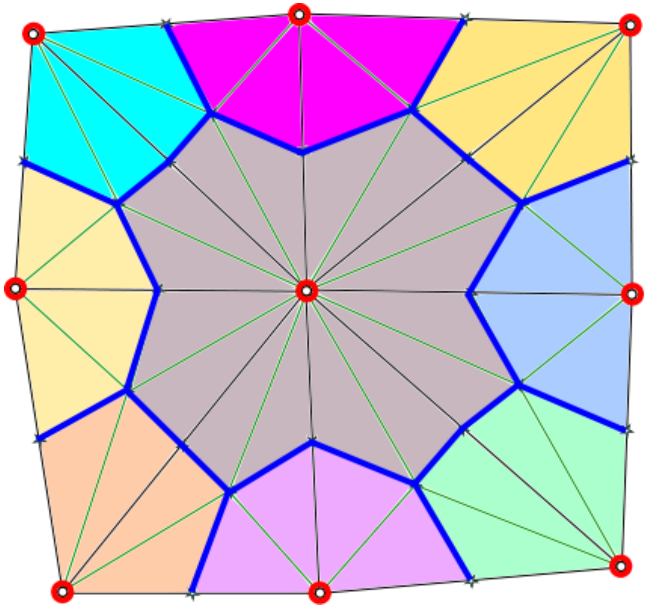
Polycrystal generated using tetrakaidecahedrons.
(Zhao et al 2007)

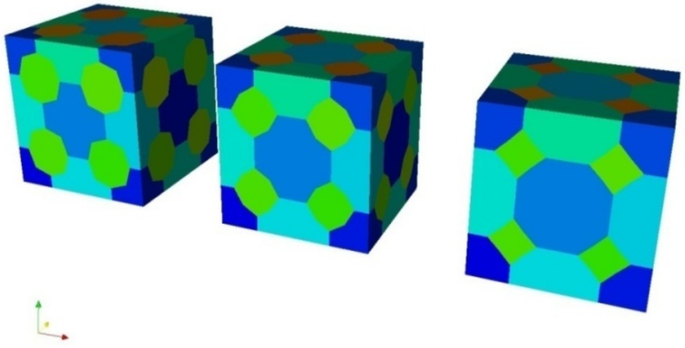
Objectives

- Develop a meshing tool for generating geometric representations of polycrystals such that:
  - Uses a standard tetrahedral mesh as the initial representation of the overall polycrystal shape (provided by any standard mesher)
  - Generates a conforming tetrahedral mesh as output (best suited for FE analysis)
  - Produces realistic grain shapes
  - Provides some control over grain size and grain size distribution

Our Approach

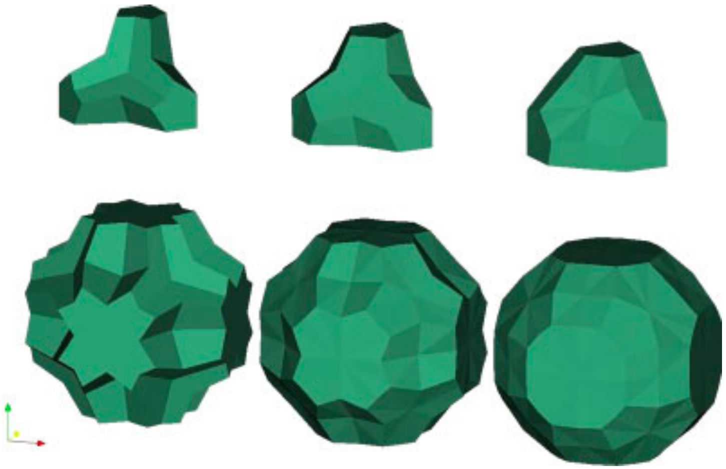
- Two step algorithm:
  - **TOPOLOGICAL STEP**: an unrelaxed polycrystal geometry is generated as the barycentric dual of a pre-existing triangulation of the solid.
  - **RELAXATION STEP**: in order to render the shape of the resulting grains more physical, grain boundaries are relaxed by minimizing grain boundary energy.





Relaxation Step

- A convexification algorithm is implemented:
  - The total grain boundary energy of the model is minimized.
  - Minimization is achieved by means of a gradient flow algorithm followed by CG.

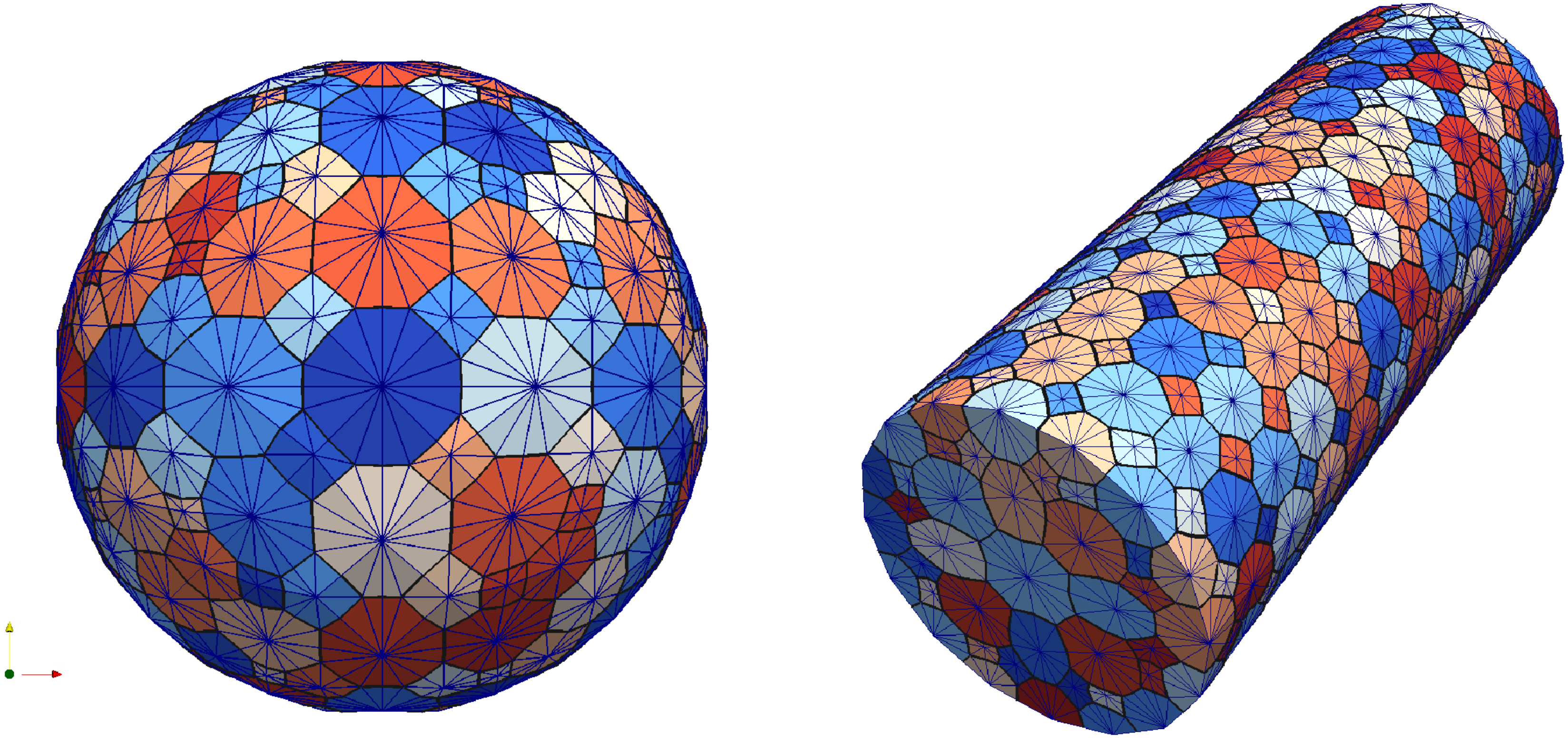


Evolution of a polycrystalline cube during grain boundary relaxation.

$$E(x) = \int_{\Gamma} \gamma \, \mathrm{d}A,$$

$$DE_a^k(x) = \sum_{p=1}^{Q} w_p \frac{\gamma}{2} \sqrt{\det(J^{\mathrm{T}}(\xi_p) J(\xi_p))} [J(\xi_p)(J^{\mathrm{T}}(\xi_p) J(\xi_p))^{-1}]_{\alpha}^{k} N_{a,\alpha}(\xi_p).$$

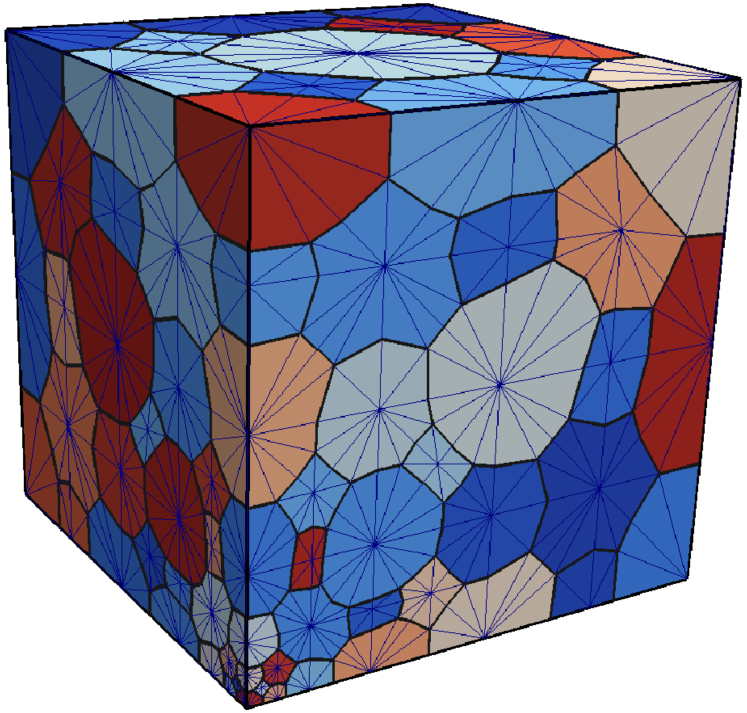
Examples of Application: RDC polycrystals of arbitrary shape



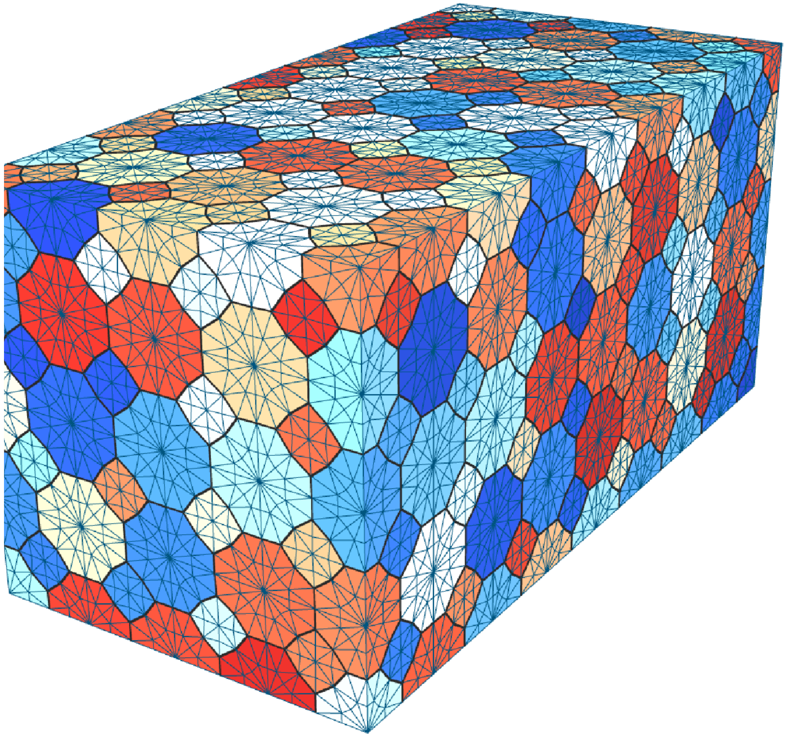
The examples demonstrate the ability of the RDC method to deal with arbitrary solid geometries.

(Rimoli and Ortiz, IJNME, 2011)

Examples of Application: coupling of adaptive mesh refinement to RDC



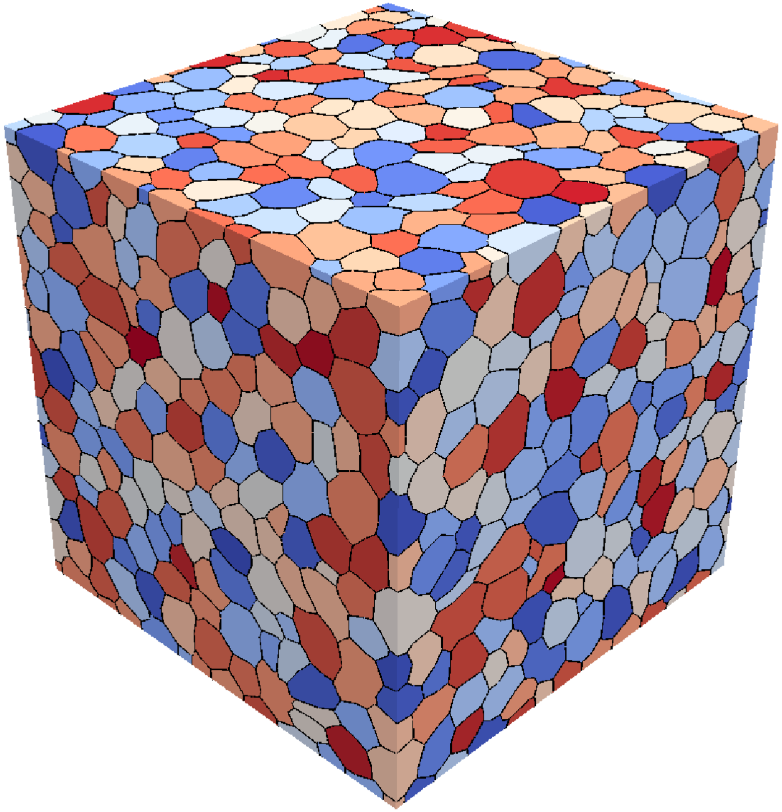
Recursive edge bisection of the input triangulation to obtain a functionally graded cubic polycrystal with a grain-size distribution bearing a power-law dependence to the distance to one of the vertices and

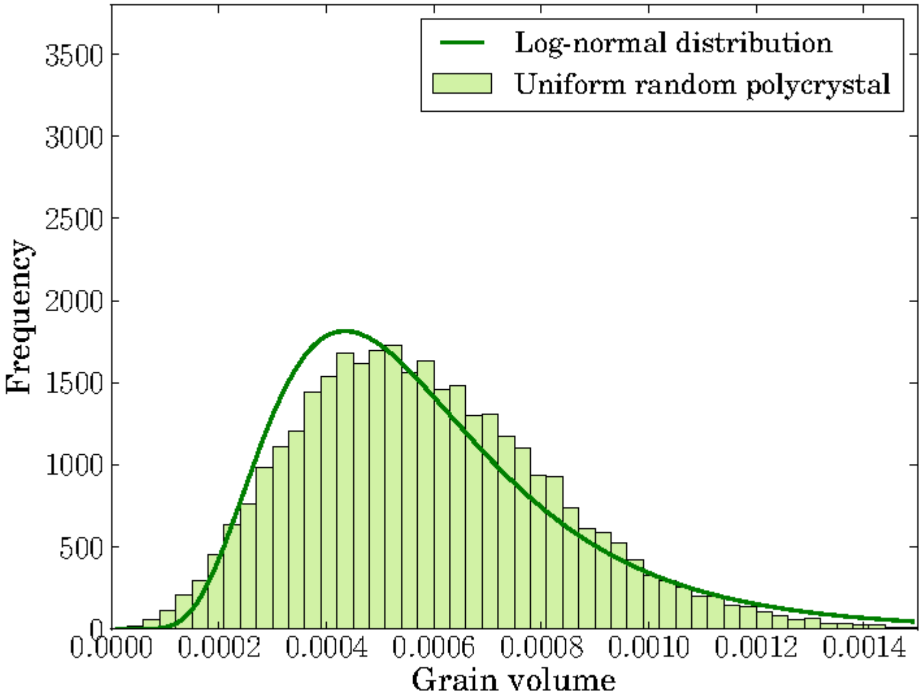


Uniform edge bisection of the output triangulation to enhance the resolution of the triangulation within the grains.

(Rimoli and Ortiz, IJNME, 2011)

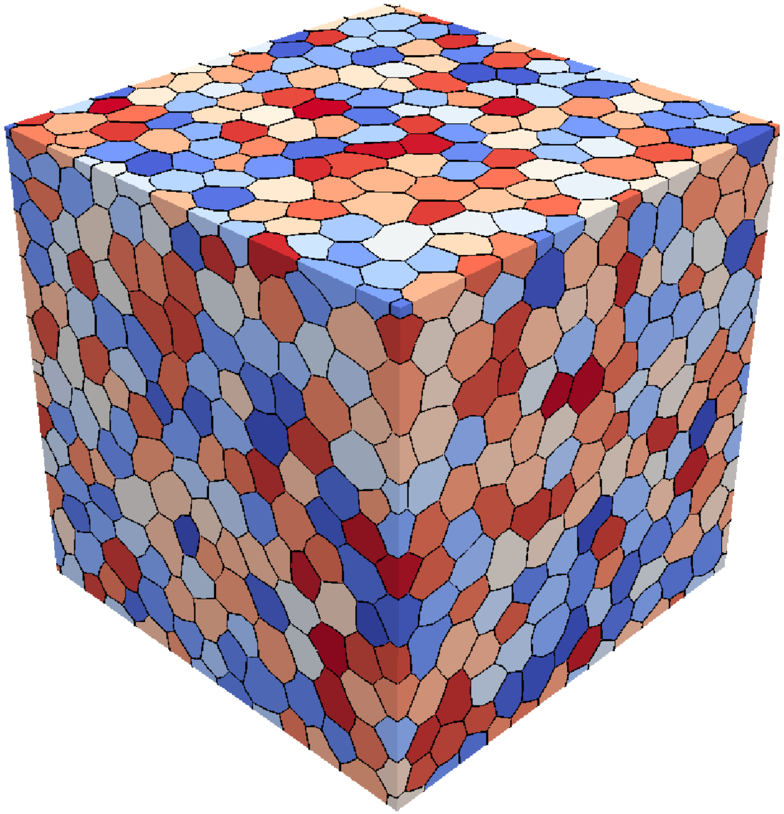
Examples of application: Grain-size distribution for a uniform probability random polycrystal

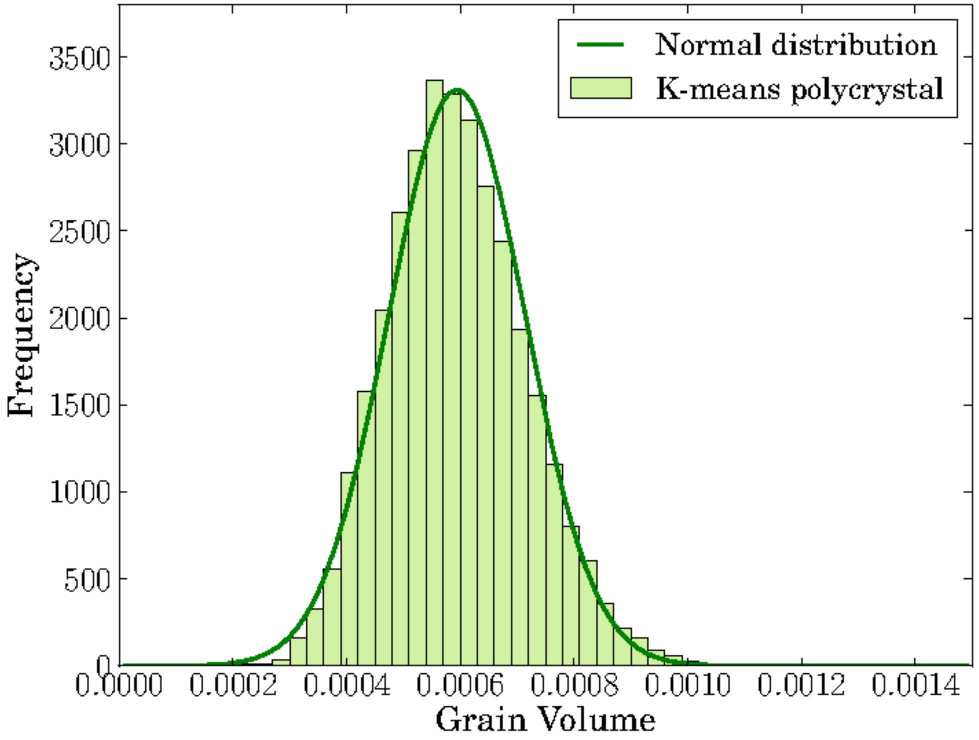




(Rimoli and Ortiz, IJNME, 2011)

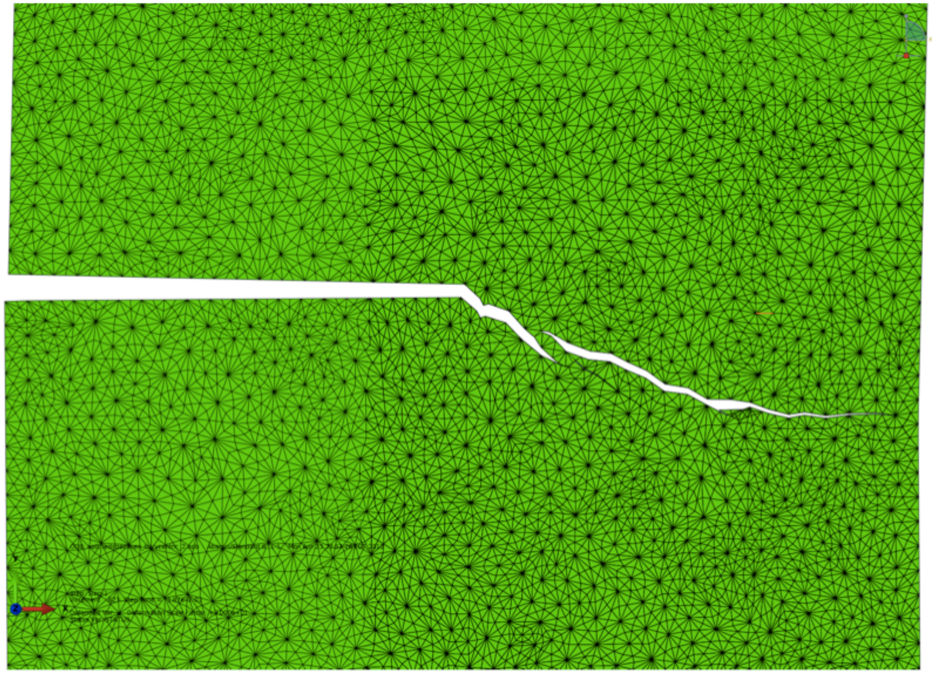
Examples of application: Grain size distribution for a **k**-means polycrystal

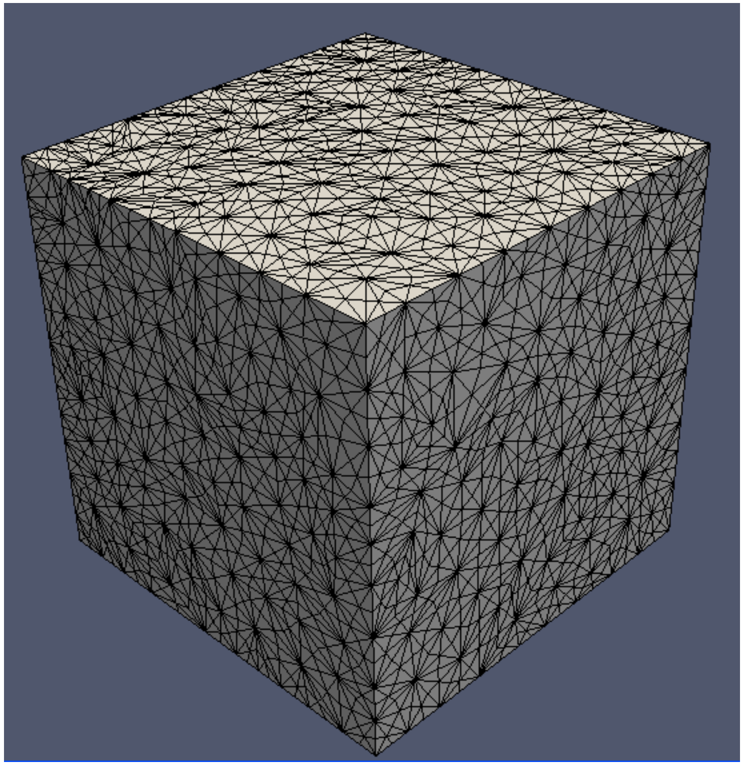




(Rimoli and Ortiz, IJNME, 2011)

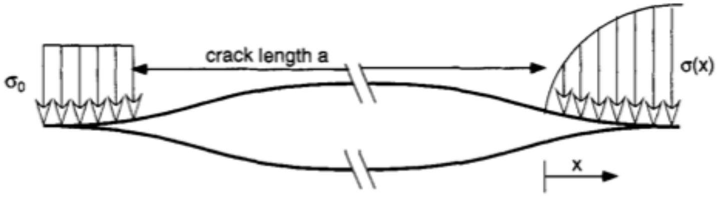
Application 2: conjugate-directions meshes for crack propagation analysis





Cohesive zone models

- The original idea for the CZM was proposed by Dugdale and Barenblatt considering that infinite stresses at the crack tip are not realistic
- Later models assumed that the cohesive stresses depend on crack opening (and not distance to crack tip)
- First implementations in FE code date back to 1976 (Hillerborg)



Dugdale (left) and Barenblatt (right) crack models.

From "Comprehensive Structural Integrity", Vol 3

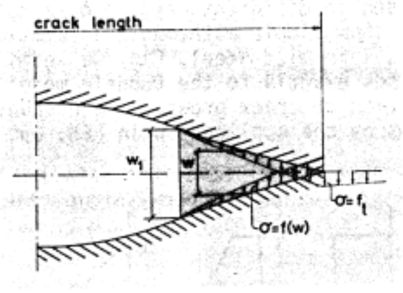
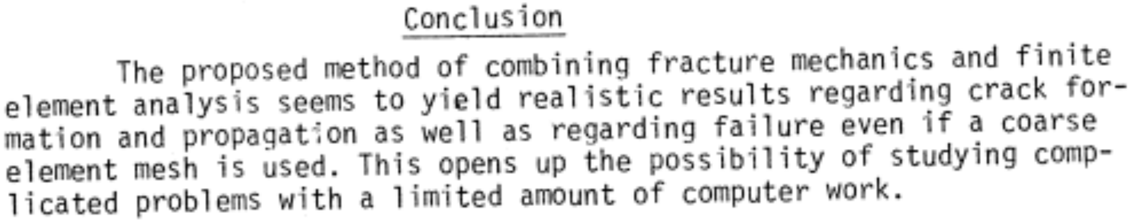
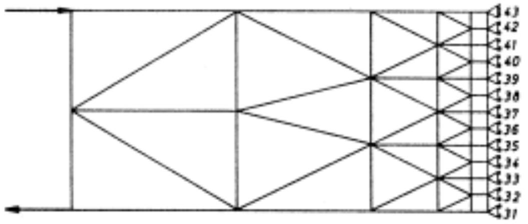


Fig. 2 Proposed model



Conclusion

The proposed method of combining fracture mechanics and finite element analysis seems to yield realistic results regarding crack formation and propagation as well as regarding failure even if a coarse element mesh is used. This opens up the possibility of studying complicated problems with a limited amount of computer work.

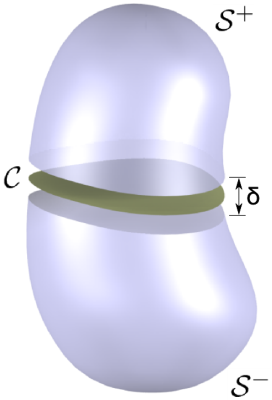


Hillerborg et al., Cement Concrete Res, 1976

Numerical implementation of CZM through cohesive elements

- Model must account for jumps in the displacement fields.
- The presence of a cohesive surface results in the addition of a new term to the deformation power identity (the work done by the cohesive tractions on the displacement jump):

$$P_{def} = \sum_{\pm} \int_{\varphi(\mathcal{S}^{\pm})} \boldsymbol{\sigma} \cdot \mathrm{sym}(\nabla \mathbf{v})\; dv + \int_{\varphi(\mathcal{C})} \mathbf{t} \cdot [[\mathbf{v}]]\; da$$



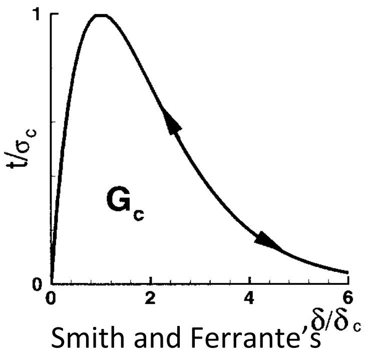
Body with cohesive surface.

- Under the right constitutive assumptions a free energy can be defined as function of crack opening, e.g.

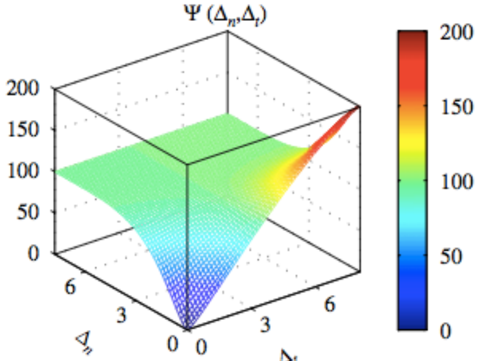
$$A(\delta) = e\sigma_c\delta_c \left[1 - \left(1 + \frac{\delta}{\delta_c}\right) e^{-\frac{\delta}{\delta_c}}\right]$$

- The corresponding binding law is:

$$t = e\sigma_c \frac{\delta}{\delta_c} e^{-\frac{\delta}{\delta_c}}$$



Smith and Ferrante's universal binding law.



Unified mixed-mode potential (Parks, Paulino et al., JMPS, 2009)

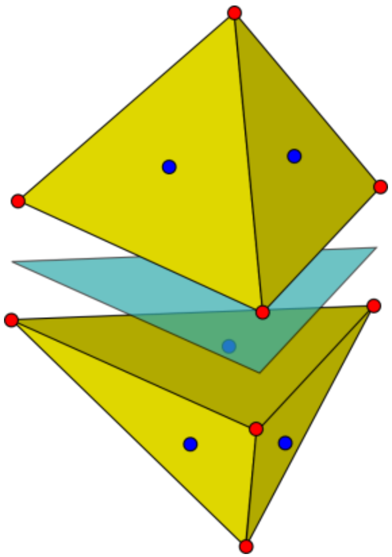
Numerical implementation of CZM through cohesive elements

- CZM have been implemented numerically by using cohesive elements by Needleman, Ortiz and others.
- Cohesive finite elements are inserted along the faces of the 3D mesh (between solid elements)
- Following Ortiz's formulation the nodal forces follow from the tractions as:

$$f_{ia}^{\pm} = \mp \int_{\mathcal{C}} t_i p_a \, dA$$

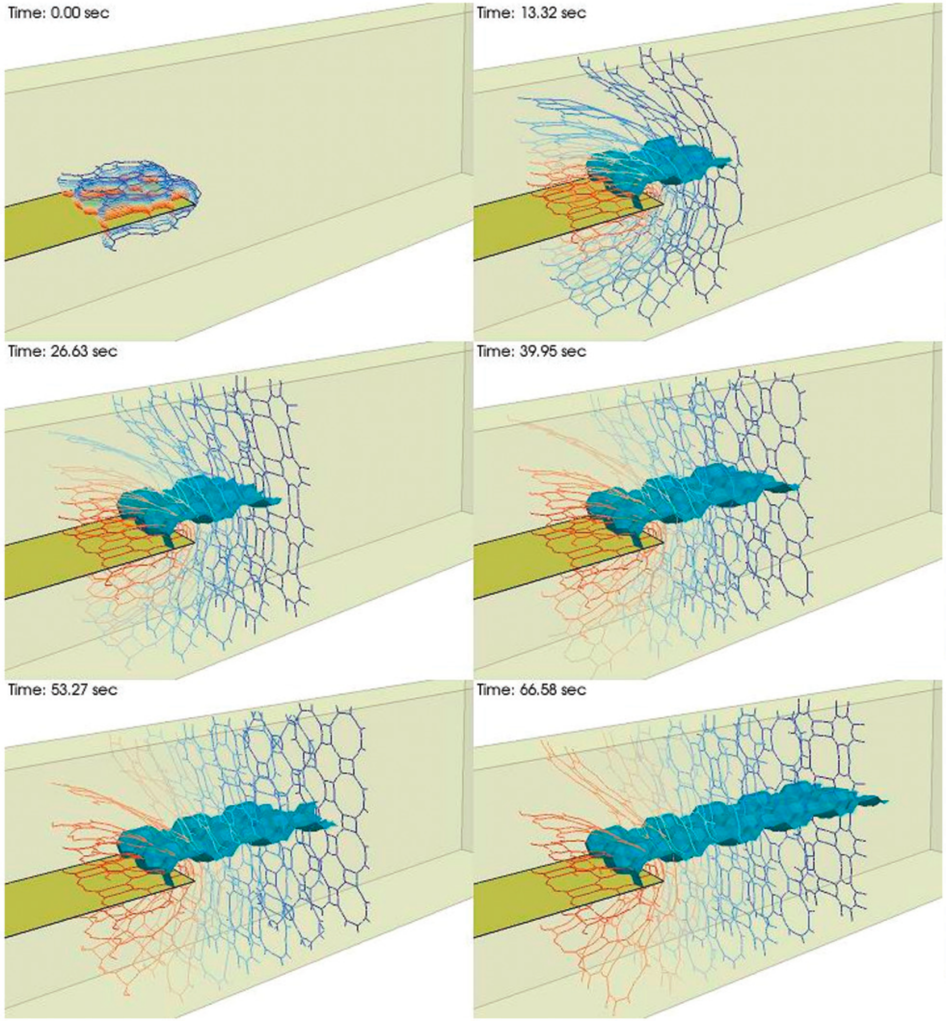
- The corresponding stiffness matrix (neglecting geometric terms) is

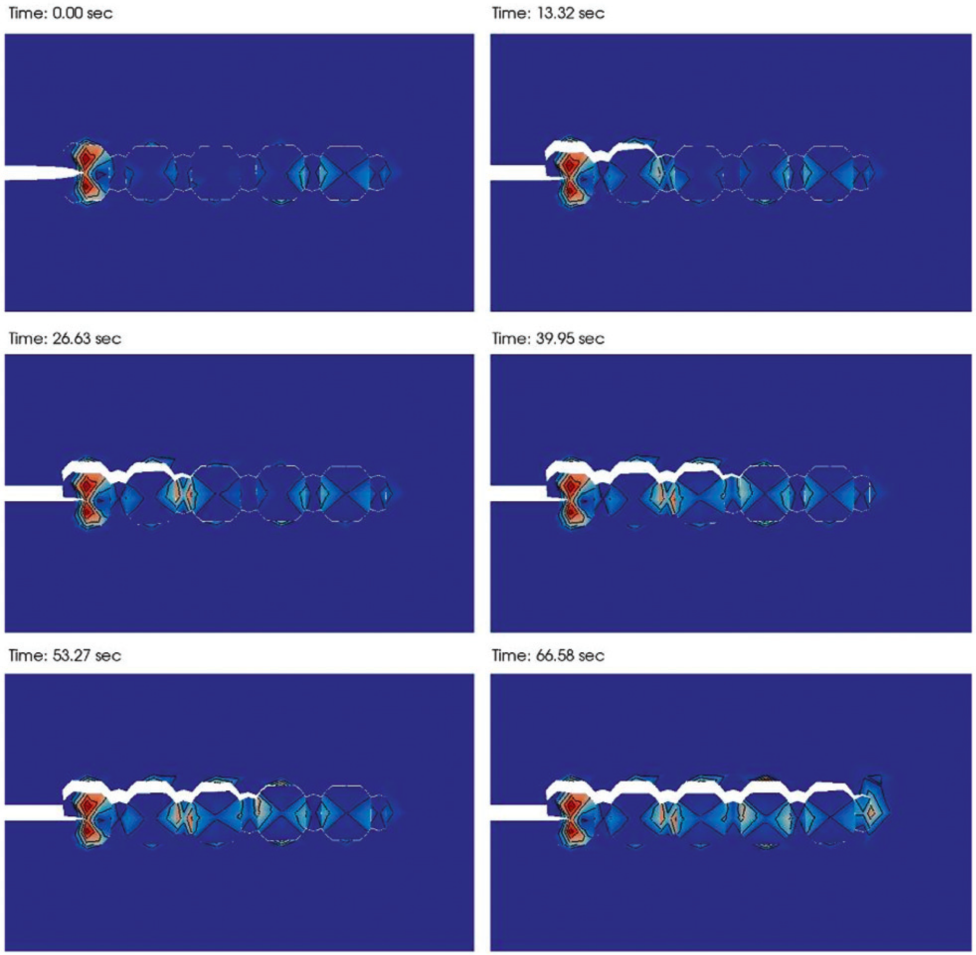
$$K_{iakb}^{\pm\pm} = \mp\mp \int_{\mathcal{C}} \frac{\partial t_i}{\partial \delta_k} p_a p_b \, dA$$



Cohesive element between two solid elements.

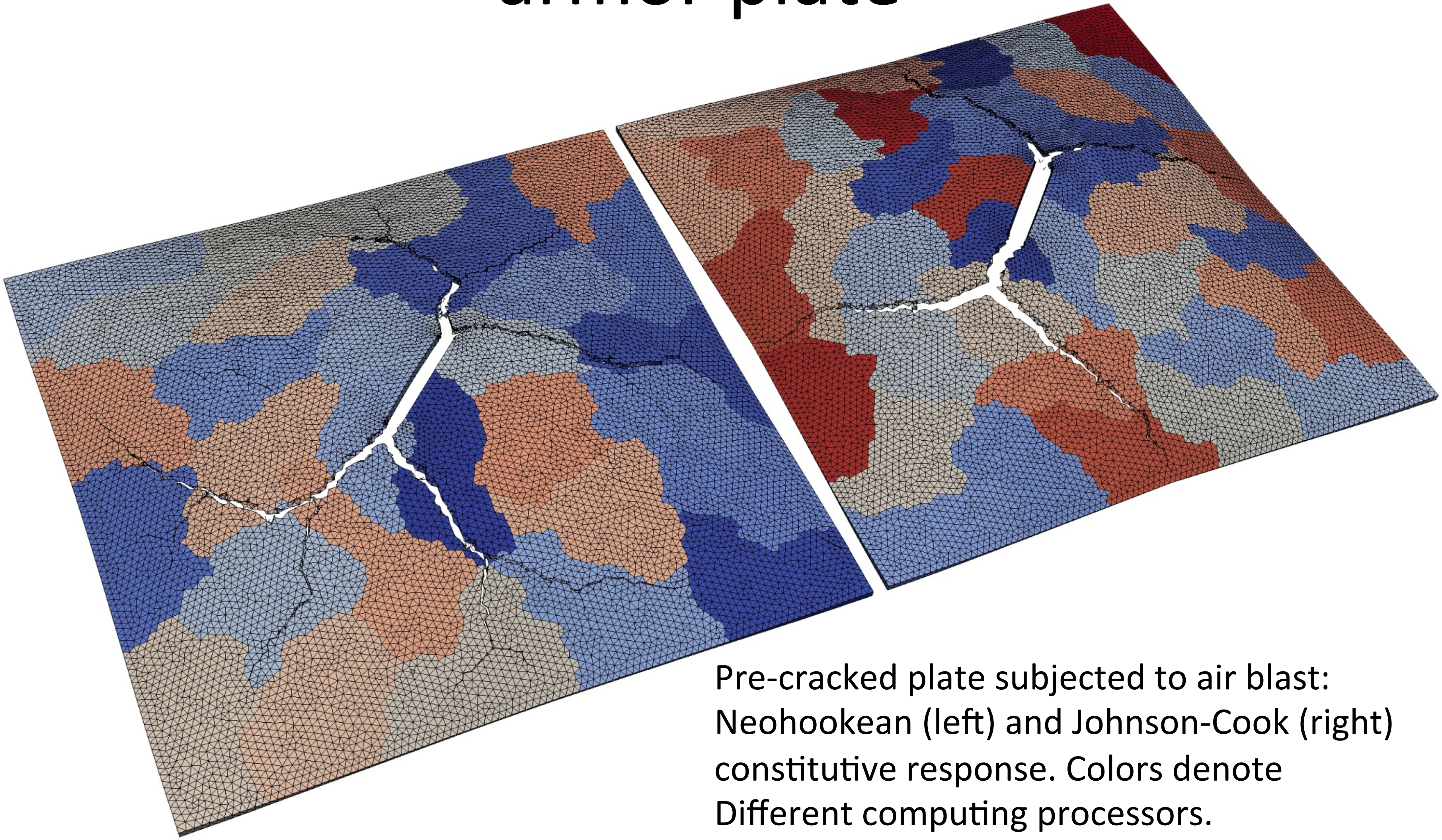
Some applications: intergranular hydrogen embrittlement of polycrystals





Rimoli and Ortiz, Phil Mag, 2010

Some applications: sand blast on armor plate



Pre-cracked plate subjected to air blast: Neohookean (left) and Johnson-Cook (right) constitutive response. Colors denote Different computing processors.

Cohesive elements are…

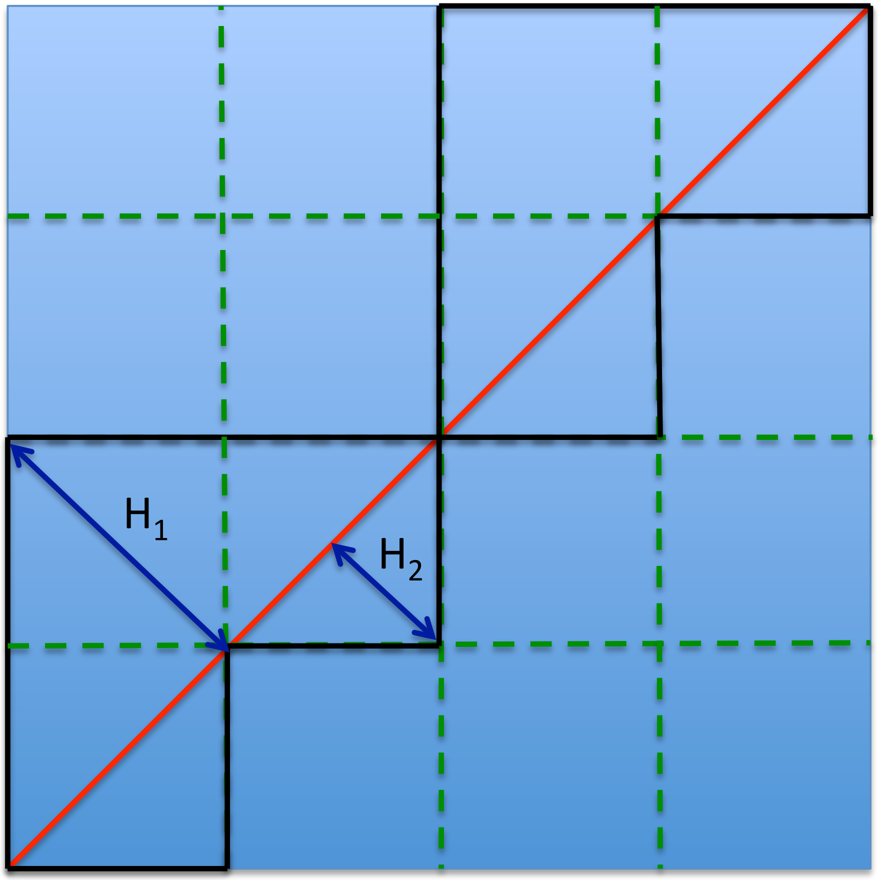
- Great!
  - They are “multi-scale” in nature: all phenomena going on in the process zone are condensed in a cohesive surface with a given cohesive law.
  - They are extremely easy to integrate into standard finite element software

- But they are not a panacea!
  - The cohesive zone length must be resolved by the mesh… or renormalization of the cohesive law must be performed.
  - They introduce an artificial compliance on the object being modeled… but DG or adaptive insertion can come to the rescue.
  - If the crack path is not know a priory mesh dependent effects appear:
    - Lack of convergence
    - Mesh induced toughness
    - Mesh induced anisotropy

Our focus

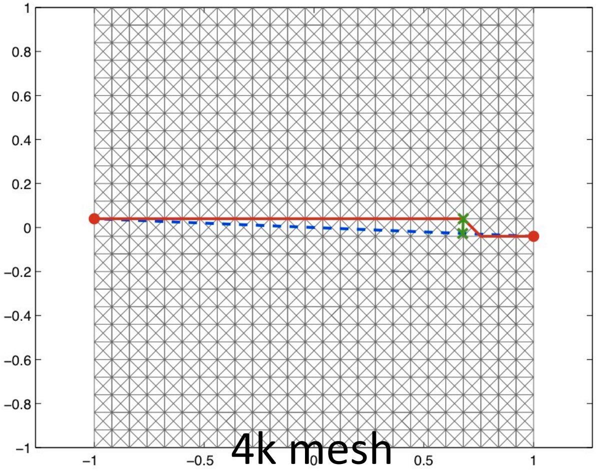
Lack of convergence

- Let us define some quantities:
  - The Hausdorff distance **H** is the maximum distance between continuum and discrete cracks.
  - Euclidean distance **Le** is the distance between the end points of the continuum crack.
  - The graph distance **Lg** is the Distance between the same points over the shortest discrete path.
  - The path deviation ratio is defined as η=**Lg/Le**
- Obviously this problem converges in the sense of the Hausdorff distance
- But **this problem does not converge in the sense of the path deviation ratio**!

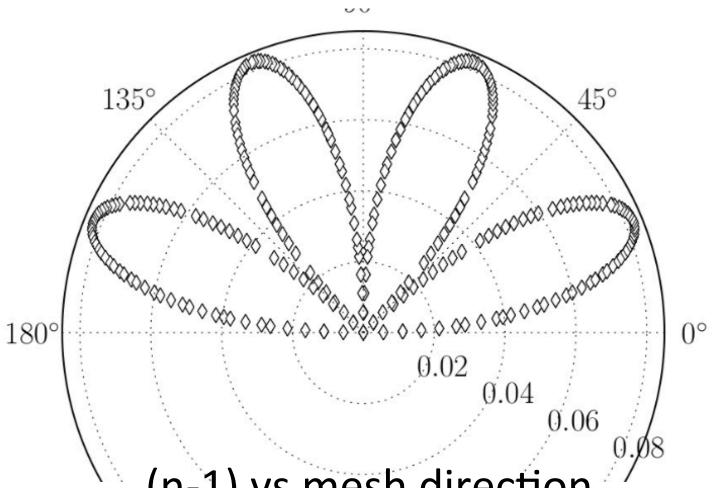


Mesh-induced anisotropy and mesh-induced toughness

- For illustration purposes, let us consider a 4k mesh.
- η = 1 for the horizontal, vertical, and ±45° directions.
- However, η≈1.08 for intermediate directions.
- In energetics terms, **directions for which η is lower provide favorable paths for crack propagation**
- In addition, **the introduction of a discrete mesh necessarily leads to a larger effective toughness for the model**



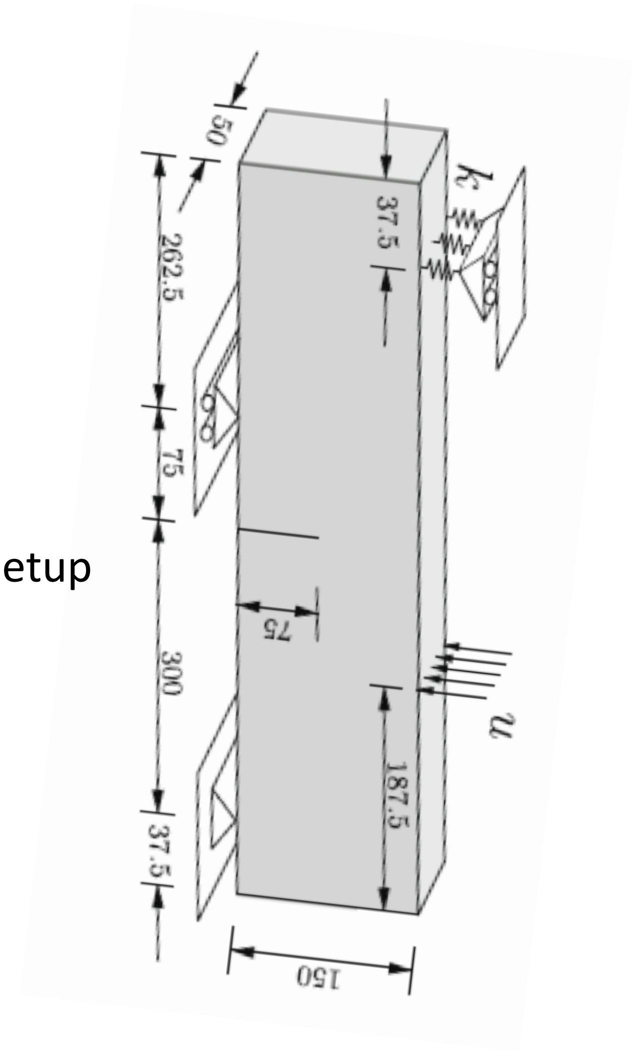
4k mesh

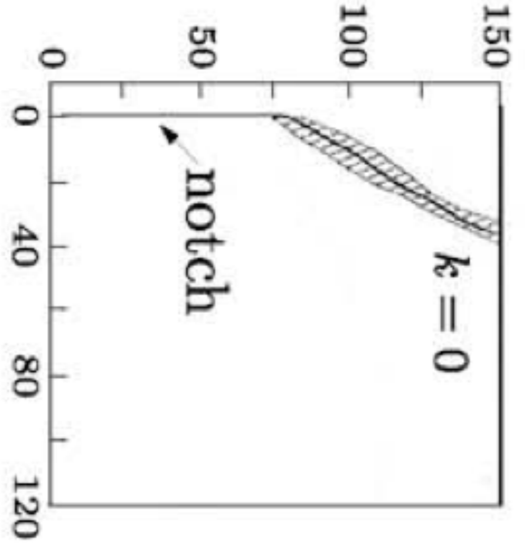


(η-1) vs mesh direction

Quick example of mesh induced effects

Specimen setup



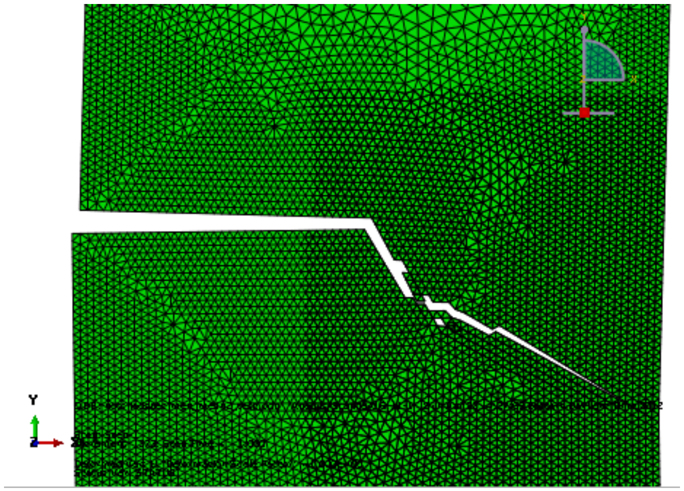


Crack pattern
(30 degrees angle expected)

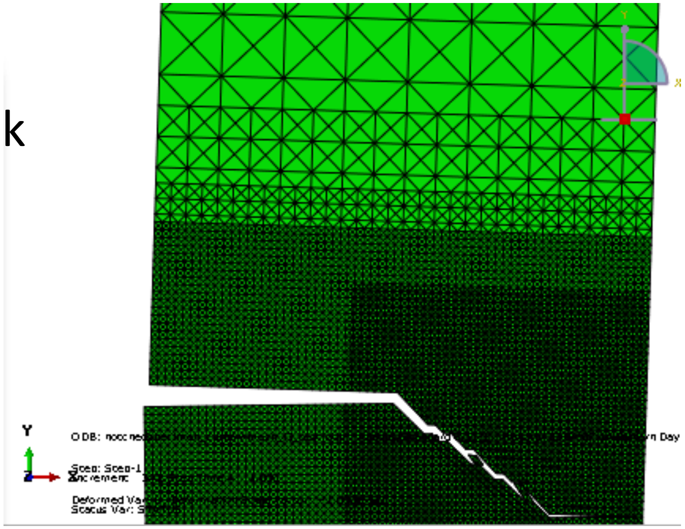
Gurses and Miehe, 2009

Mesh-induced effects

- Problem solved using 2 types of mesh on Abaqus:
  - 1) Default Abaqus mesh obtained by just specifying mesh size
  - 2) 4K mesh
- Abaqus standard mesh generator introduces some structure, showing extreme mesh dependence (crack strongly aligned with mesh directions)
- As expected, crack propagates in the 45 degrees direction for the 4k mesh (strongly preferred direction!)
- **This highlights the necessity of better meshes when using cohesive zone modeling.**



Default Abaqus mesh



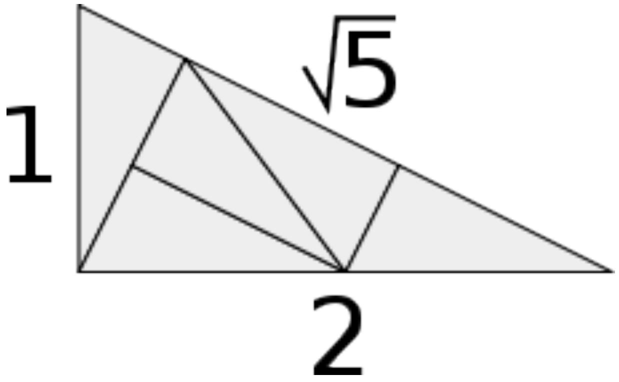
4K mesh

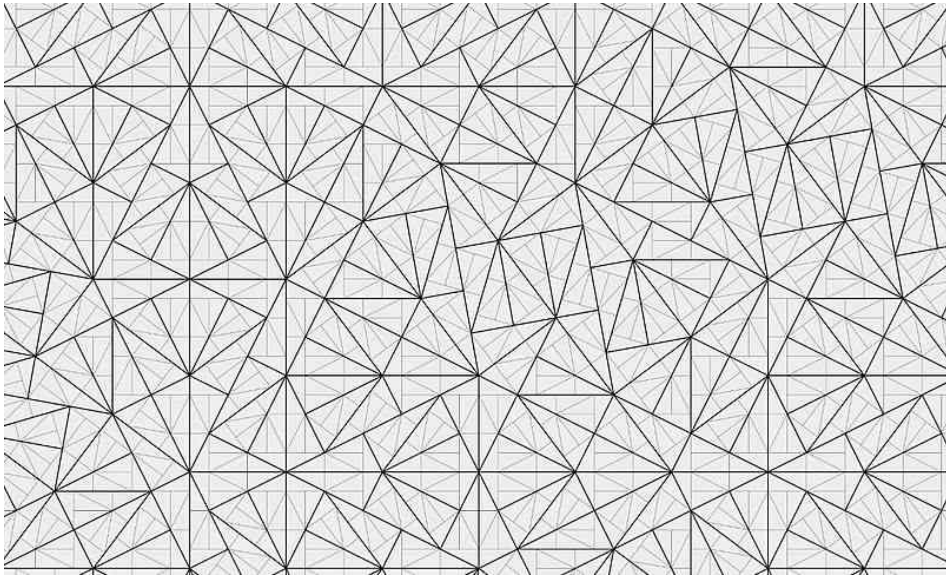
Towards isoperimetric meshes

- A **necessary condition** for cohesive zone models **to converge** (for problems where crack path is not known a priori) is to have isoperimetric meshes, i.e., meshes such that the **path deviation ratio tend to one as the mesh size decreases**, for any direction in the mesh.

Pinwheel meshes

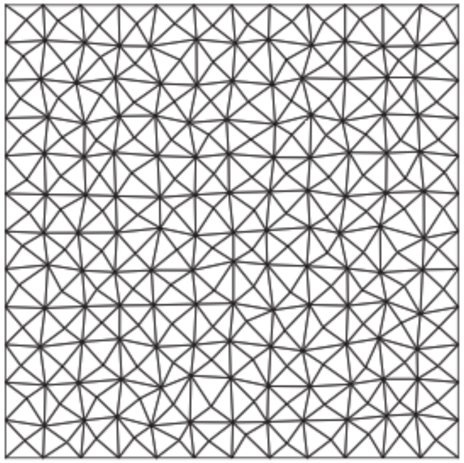
- Pinwheel tiling (Radin, Sadun, 1994): on the limit for the mesh size tending to zero, pinwheel meshes have the isoperimetric property (η->1 for all directions)
- Very difficult to mesh arbitrary domains.
- No known extension for the 3D case.



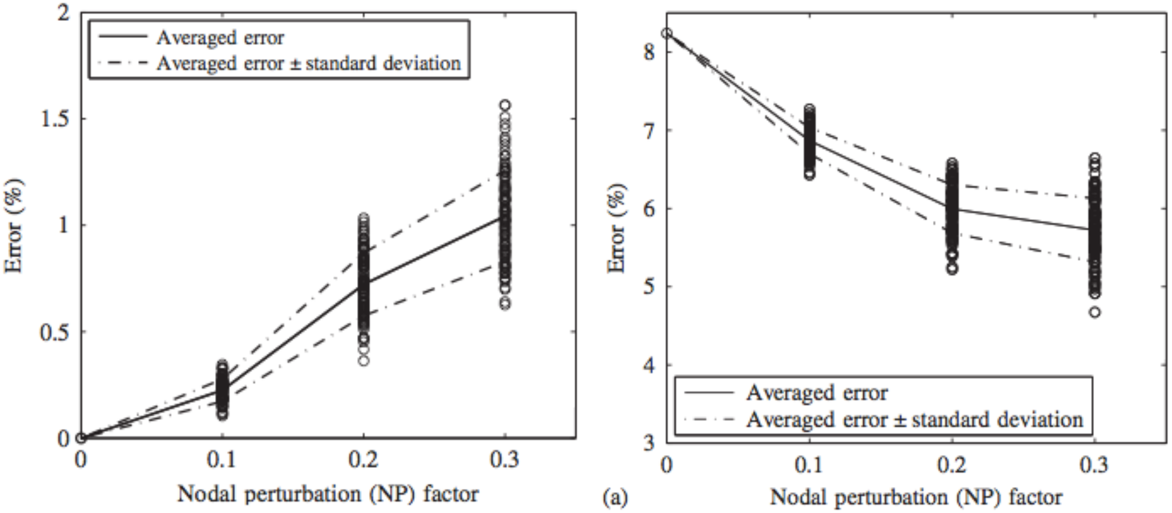


Perturbed 4k meshes

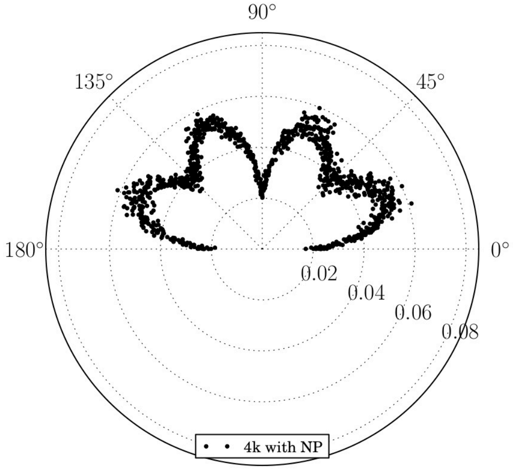
- Paulino et al. (2010) studied the convergence of perturbed 4k meshes by introducing nodal perturbation (NP) and edge swap (ES) operators.
- The effect of these operators is to increase η for the favorable directions and decrease it for the intermediate ones
- They state “the additional error along the 45◦ direction, introduced by the NP, is less than the error reduction along the 67.38◦ direction due to the NP”



Paulino et al. (2010)

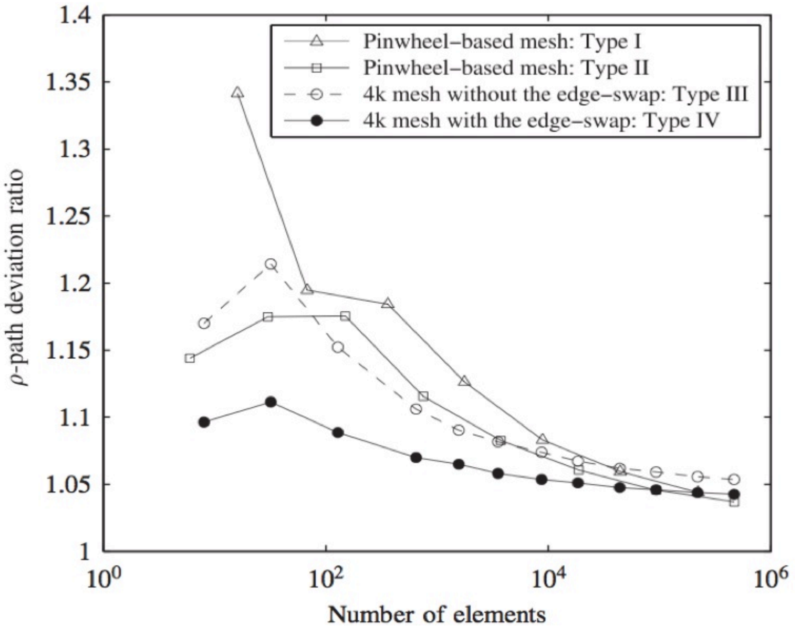


Paulino et al. (2010)



Convergence of 4k meshes

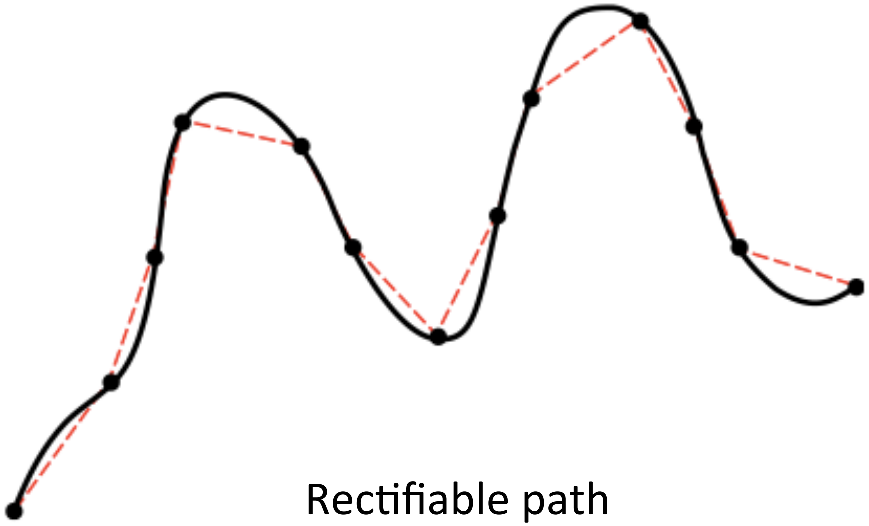
- Look at the path deviation ratio as the number of elements increases
- The best values observed are 1.044 and 1.037 for NP and NP+ES meshes respectively, as good as pinwheel meshes for reasonable mesh sizes



Paulino et al. (2010)

Rectifiable paths

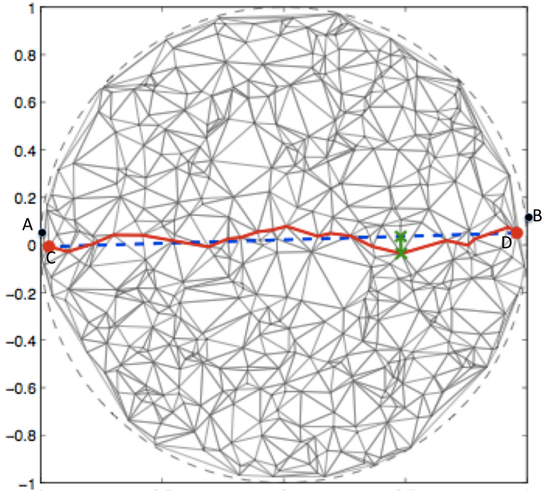
- In general, the crack path length can be computed by rectification of the irregular curve
- The approximated crack length is computed as the sum of the length of the corresponding segments.
- The actual length could be then obtained by computing the limit of the approximated length for the segment’s length tending to zero.
- Thus, **a necessary condition for a mesh to represent an arbitrary crack, is that it should be able to represent a straight segment**.
- From now on we focus our discussion on straight segments.

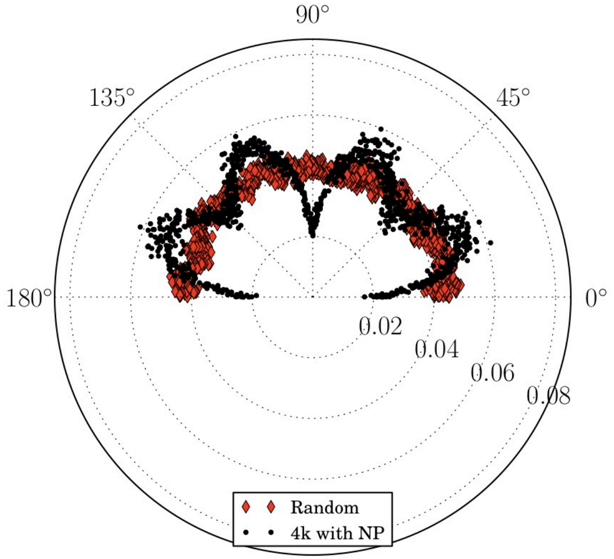


Rectifiable path

An illustrative example: how about randomly generated meshes?

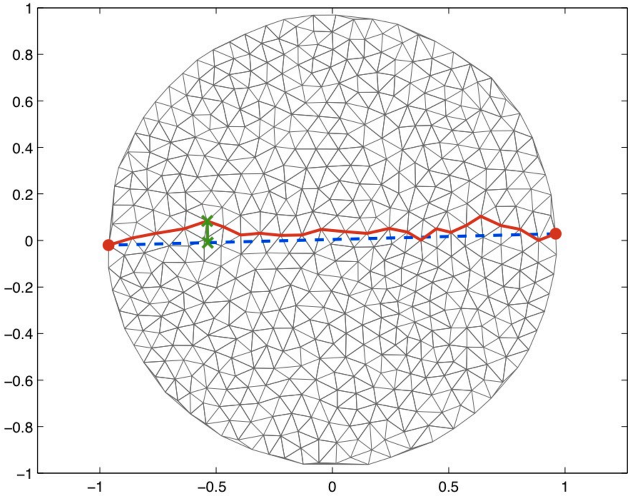
- Random mesh: generated by throwing random points and performing Delaunay’s triangulation.
- To compare meshes, we introduce a new parameter **λ=Le/h$_{av}$** where h$_{av}$ is the mean value of the edge length
- λ is a measure of how many segments are needed to represent the crack.

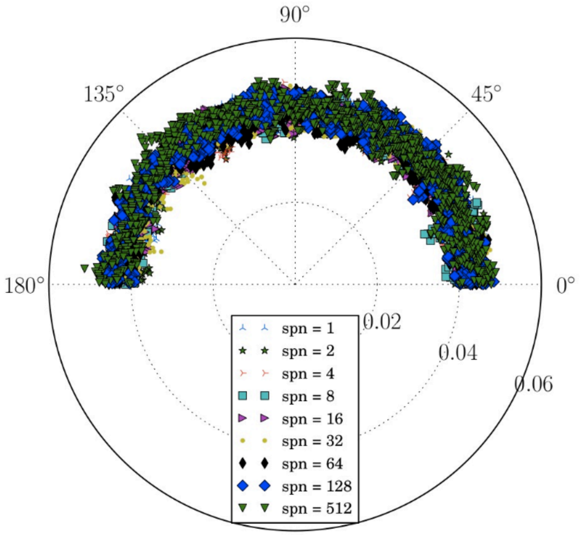




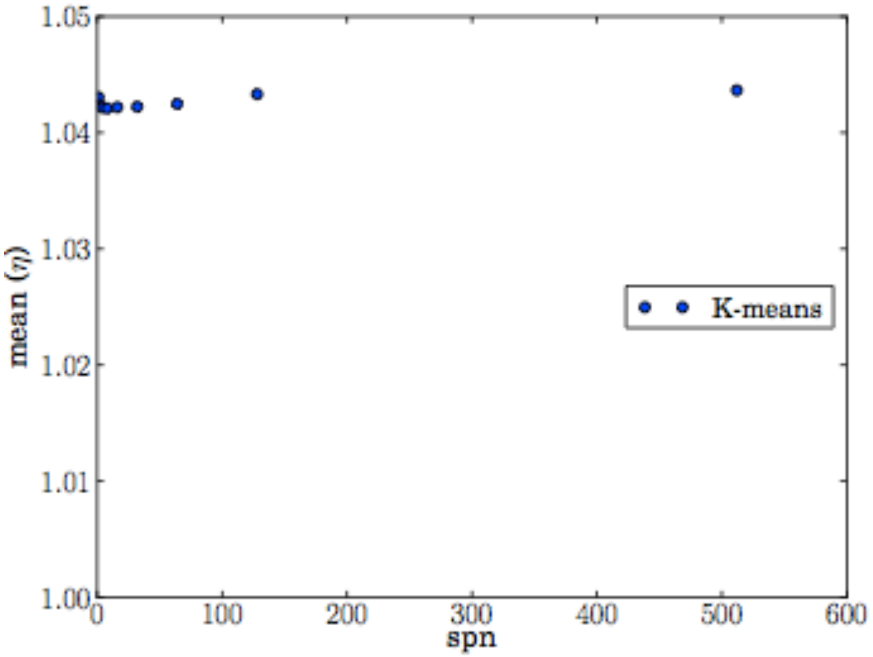
Better random meshes: K-means meshes

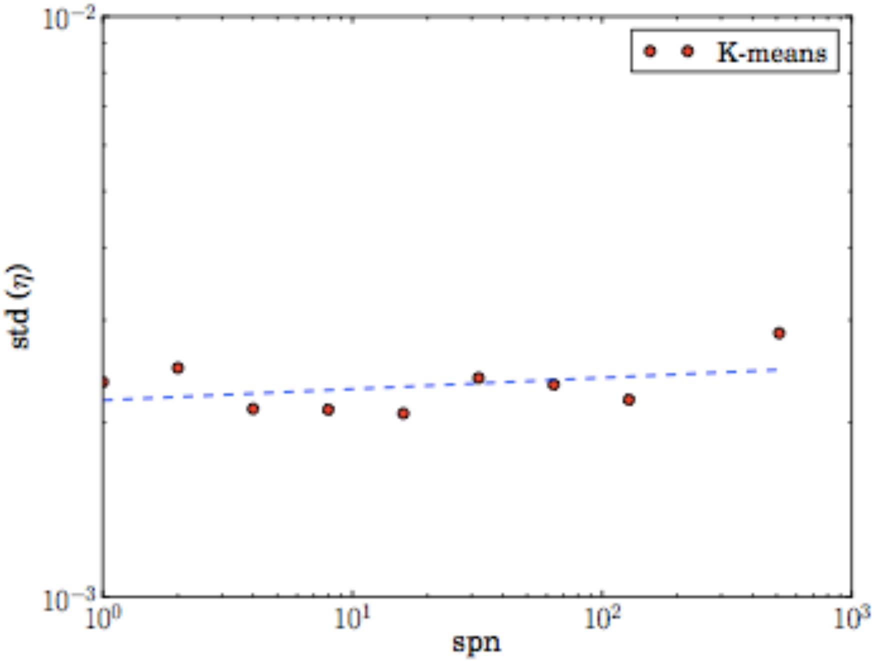
- We use the k-means clustering algorithm to improve the mesh quality of a random mesh
- We generate a mesh with N nodes by clustering (spn*N) random points
- The parameter spn is a measure of the clustering level
- The larger spn, the smoother the mesh, and **spn=1 corresponds to the random mesh**





K-means meshes
Clustering effect





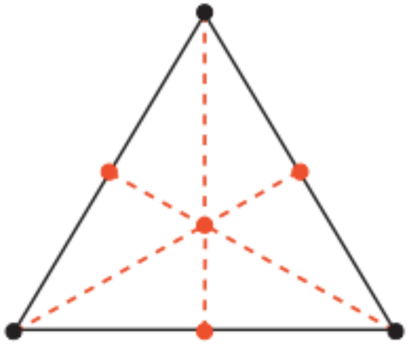
- The level of smoothing does not affect the path deviation ratio significantly

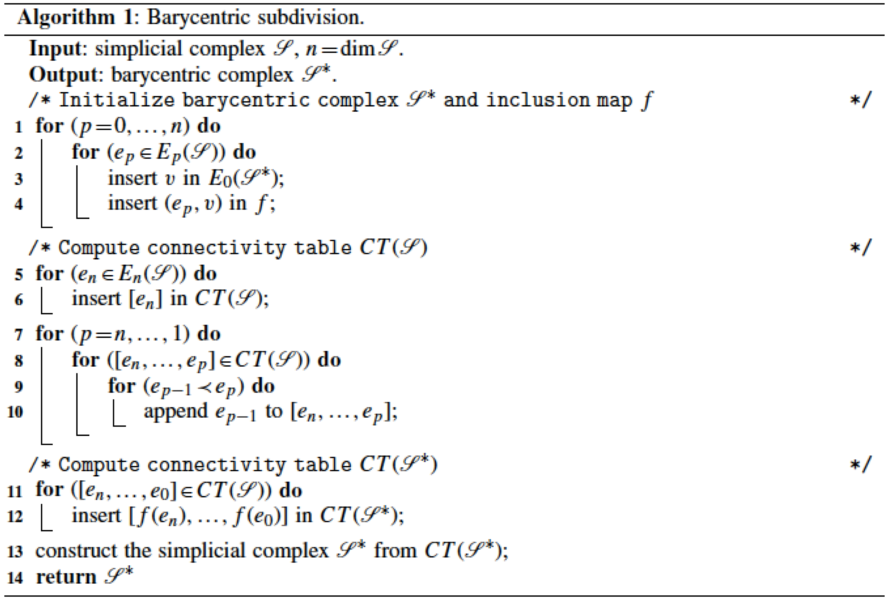
It's all about directions

- The key to achieve the isoperimetric goal is to have meshes able to provide infinitely many directions as the pinwheel tiling does
- How can we make our mesh have more directions?
  - One (brute force) way is just increasing the number of nodes of the mesh…
  - But we can do better than that **by performing well defined topological operations on the mesh**!

Barycentric subdivision: adding conjugate directions!

- In the limit of equilateral triangles, the algorithm provides, for each edge, a perpendicular direction
- In this way, **we are adding more directions to the mesh… but conjugate directions!**
- Of course we are adding more triangles as well, but the parameter lambda will ensure that we compare apples to apples





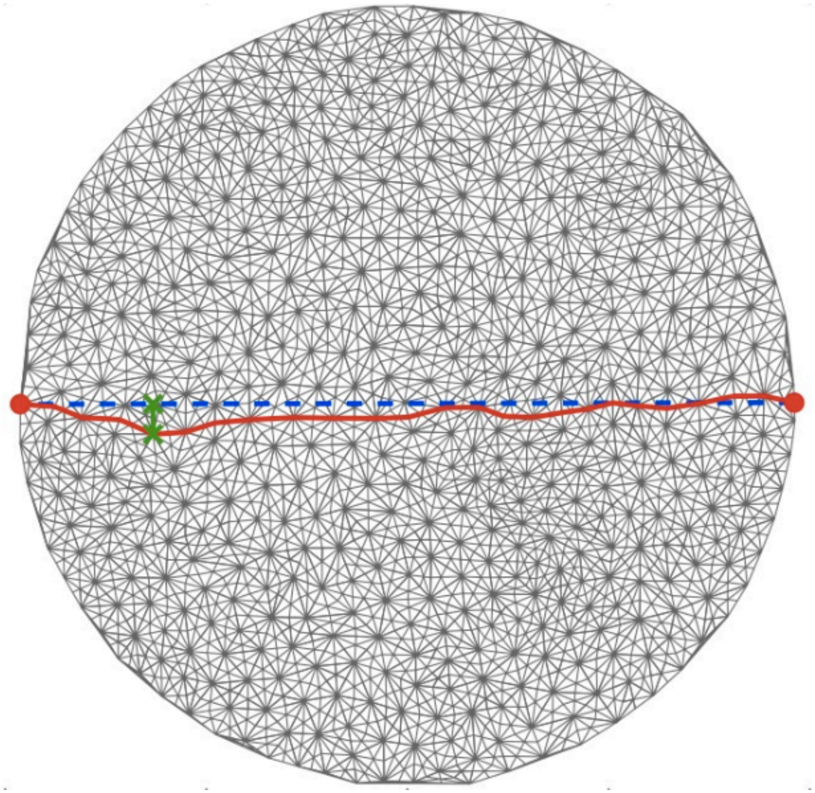
Algorithm 1: Barycentric subdivision.

```
Input: simplicial complex $\mathscr{S}$, $n = \dim \mathscr{S}$.
Output: barycentric complex $\mathscr{S}^*$.
/* Initialize barycentric complex $\mathscr{S}^*$ and inclusion map $f$ */
for ($p = 0, \dots, n$) do
    for ($e_p \in E_p(\mathscr{S})$) do
        insert $v$ in $E_0(\mathscr{S}^*)$;
        insert $(e_p, v)$ in $f$;
/* Compute connectivity table $CT(\mathscr{S})$ */
for ($e_n \in E_n(\mathscr{S})$) do
    insert $[e_n]$ in $CT(\mathscr{S})$;
for ($p = n, \dots, 1$) do
    for ($[e_n, \dots, e_p] \in CT(\mathscr{S})$) do
        for ($e_{p-1} \prec e_p$) do
            append $e_{p-1}$ to $[e_n, \dots, e_p]$;
/* Compute connectivity table $CT(\mathscr{S}^*)$ */
for ($[e_n, \dots, e_0] \in CT(\mathscr{S})$) do
    insert $[f(e_n), \dots, f(e_0)]$ in $CT(\mathscr{S}^*)$;
construct the simplicial complex $\mathscr{S}^*$ from $CT(\mathscr{S}^*)$;
return $\mathscr{S}^*$
```

Rimoli and Ortiz, IJNME, 2011

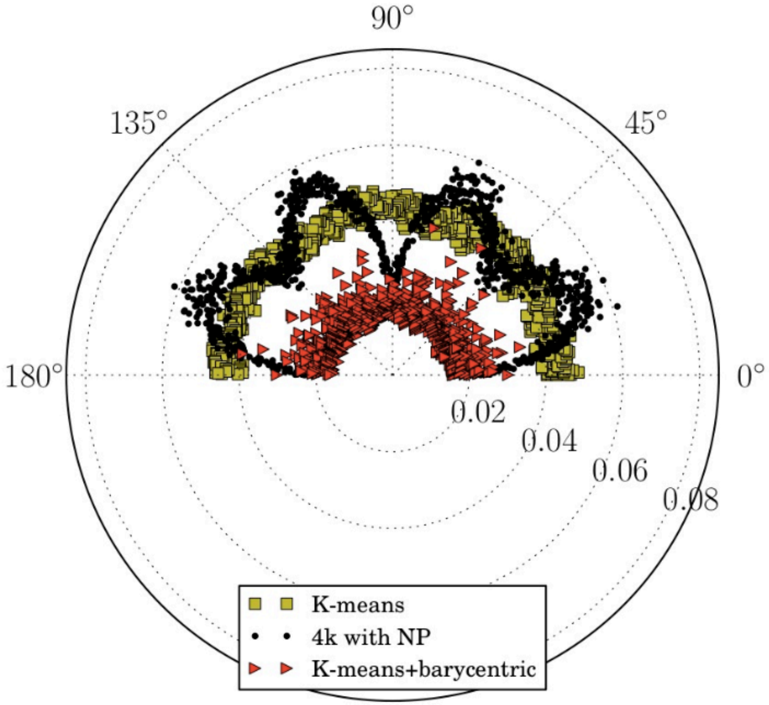
Conjugate-directions meshes

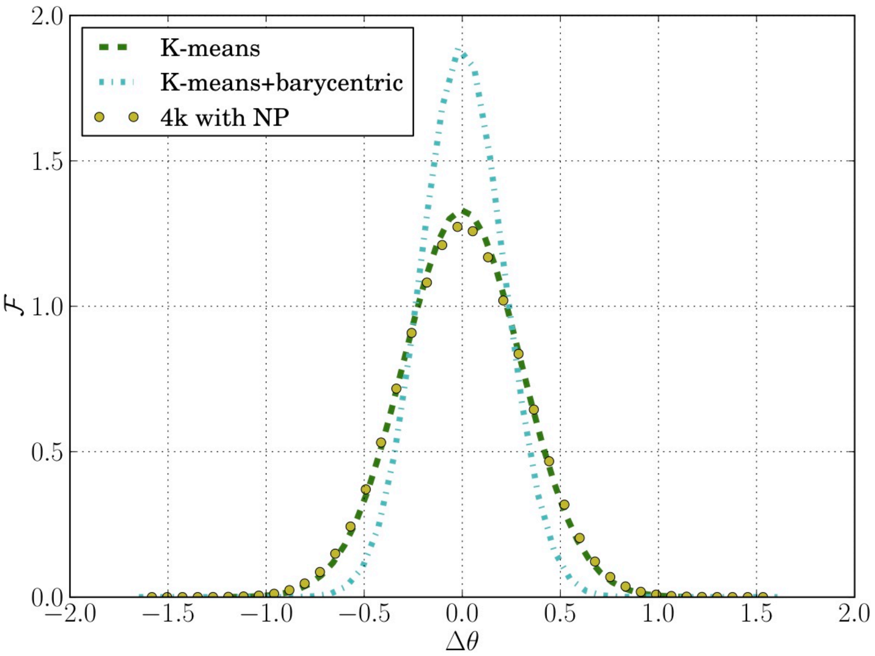
- **We call conjugate-directions meshes to those generated by the application of the barycentric subdivision to a random k-means mesh**
- A conjugate direction mesh has worse quality that the seed mesh used for its generation
- Mesh quality is generally not a problem when the seed mesh is a smooth k-means mesh



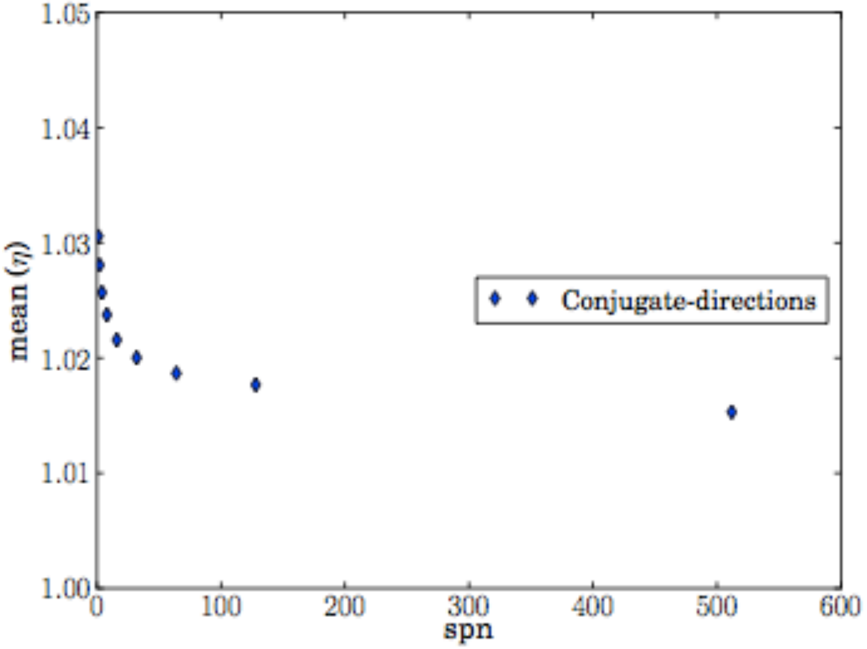
Conjugate-directions meshes

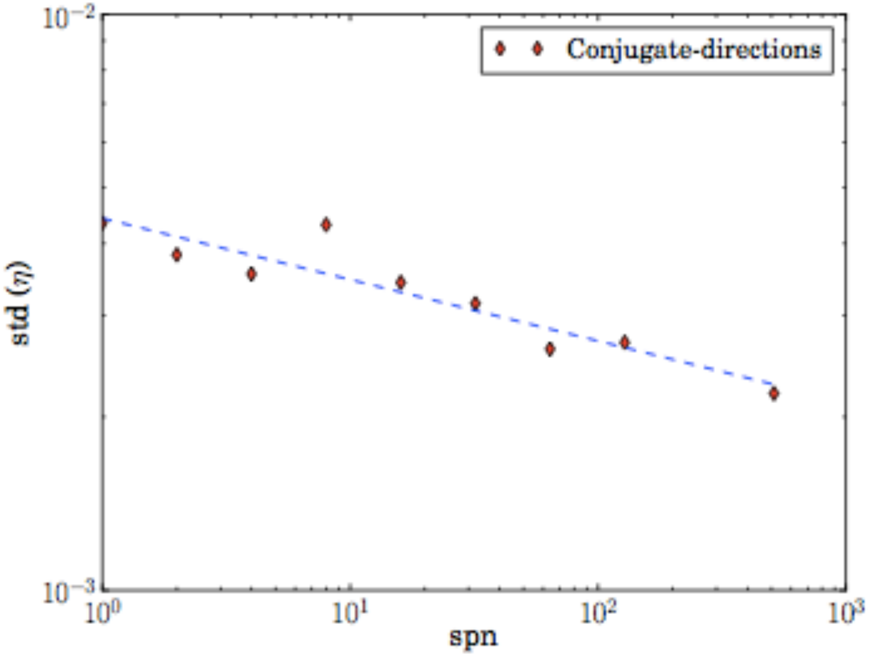
- For same lambda, the conjugate direction mesh significantly outperforms both the 4k mesh with NP and the k-means mesh





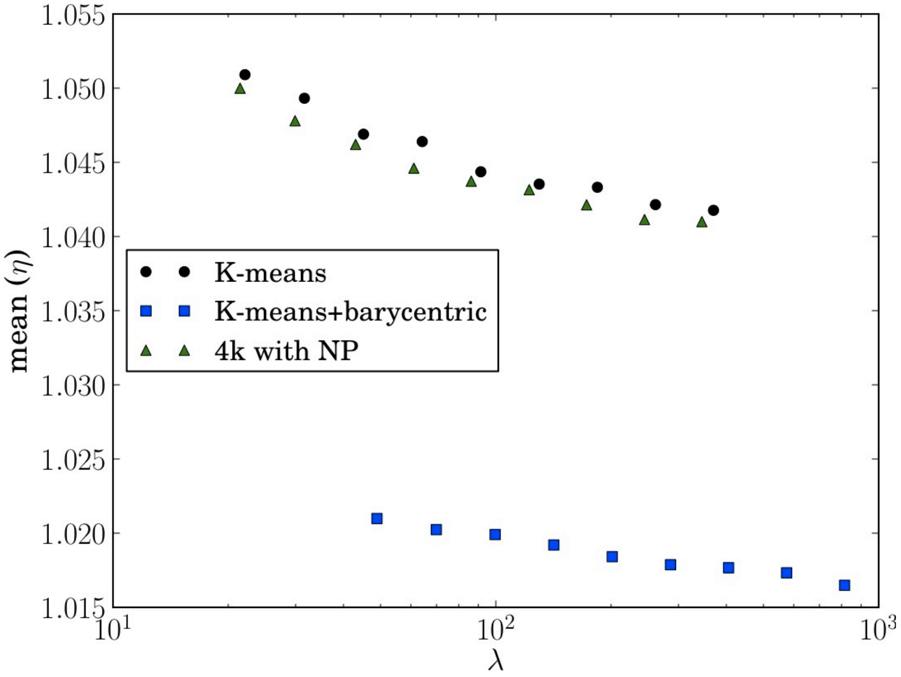
Conjugate-directions meshes
Clustering effect

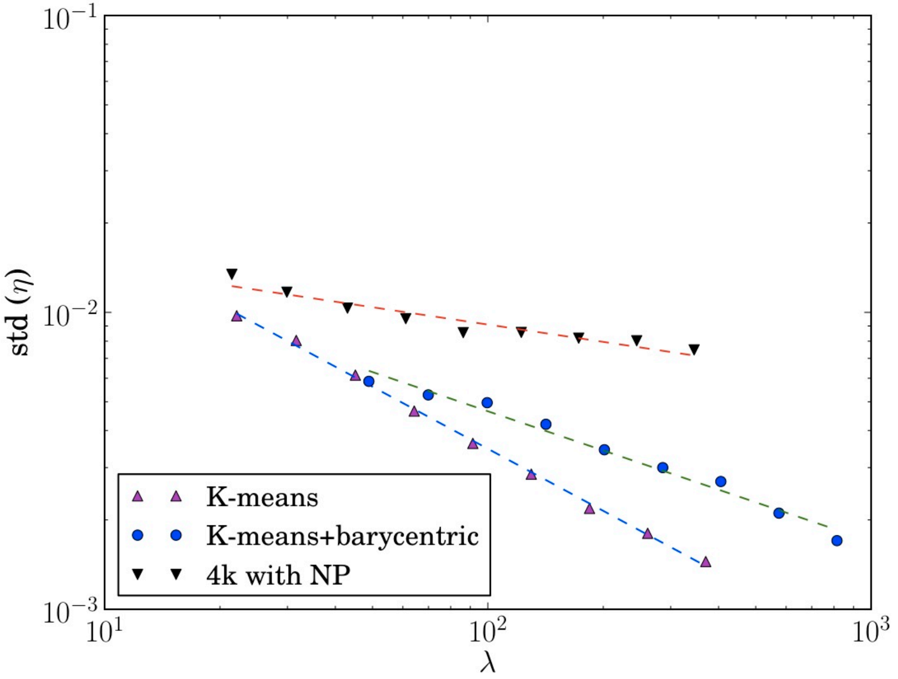




- The level of smoothing prior to the barycentric subdivision greatly affect the path deviation ratio.
- **For the same mesh, the error can be reduced by 50% just by smoothing before performing the barycentric subdivision!**

Numerical study of convergence

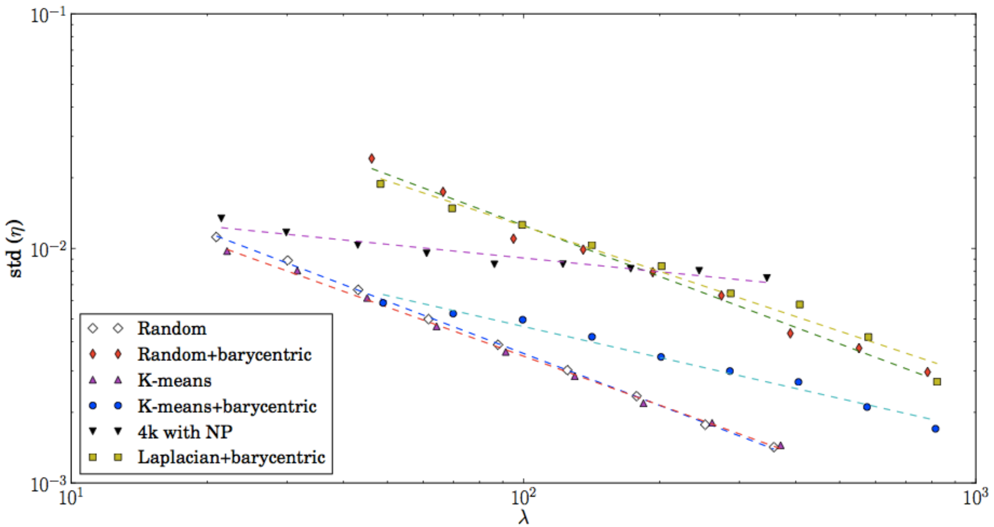


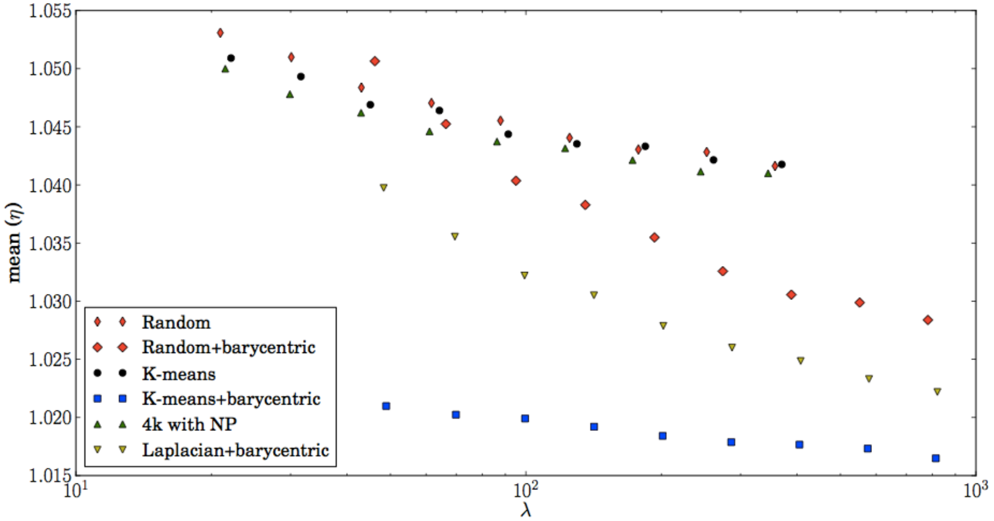


- The average value of the path deviation ratio is significantly better for the conjugate directions mesh (**error reduction in the range of 2.5x for reasonable mesh sizes!**)
- Trends in the standard deviation suggest that probabilistic bounds for the path deviation ratio may exist

Convergence (cont.)

- We studied other types of meshes and the barycentric subdivision seems to always decrease significantly the path deviation ratio, but best results are obtained when the seed mesh is "smooth"



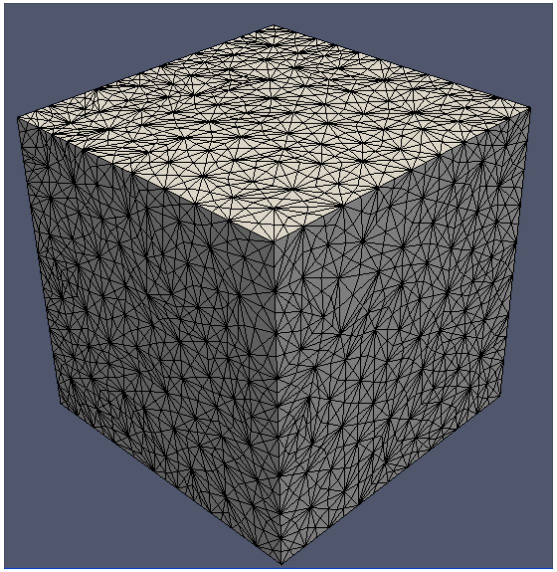


Summary

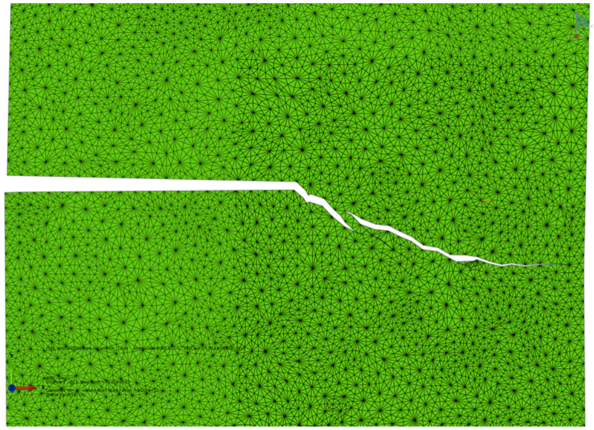
- When dealing with CZM for problems in which the crack path is not known a priory, mesh dependent effects appear:
  - Lack of convergence
  - Mesh induced toughness
  - Mesh induced anisotropy
- Attempts to address these issues have been made:
  - Pinwheel meshes: hard to generate and not extendable to 3D cases
  - 4k meshes with NP and ES: performance similar to pinwheel for reasonable mesh sizes but there might still be some anisotropy
- **K-means meshes perform, in average, as well as 4k meshes with NP but with no observed anisotropy**
- **Conjugate direction meshes perform significantly better than all other meshes investigated, particularly when the seed k-means mesh has high spn value (smooth seed mesh)**
- **For the same mesh, the error can be reduced by 50% just by smoothing before performing the barycentric subdivision!**
- **Error reduction in the range of 2.5x for reasonable mesh sizes!**

Current work and future directions

- We are focusing our work on:
  - Studying different smoothing algorithms for seed meshes and their effect on conjugate direction meshes.
  - Performing numerical simulations of brittle crack propagation to asses the convergence of conjugate direction meshes
  - Extend our analysis (K-means and conjugate-directions) to 3D



3D C-D mesh



Preliminary crack propagation result shows correct angle!

Acknowledgements

- Special thanks to Juan J. Rojas, Gautam Puri (grad students) and Frah Khemani (undergraduate summer researcher) for their work.
- Funding from Sandia (Livermore) and GT startup fund

Thank you!