3D Frame Fields and Block Decomposition of CAD models

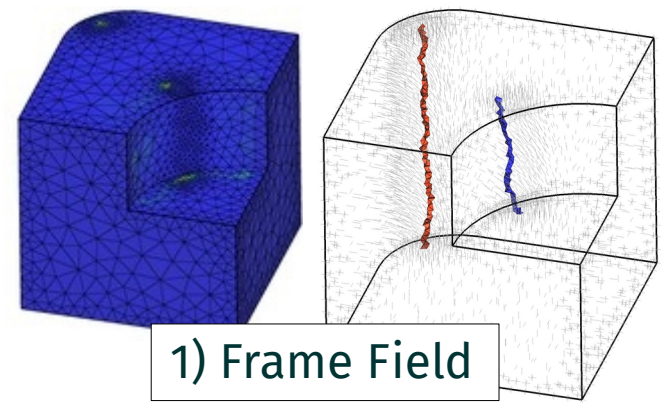
Maxence Reberol
UCLouvain

Bern, 4 March 2020

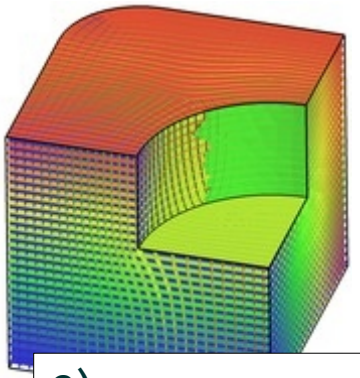
Context

- ~2 years of postdoc at UCLouvain
- Looking for an alternative to CubeCover / mixed-integer
  - Build the big blocks and not the small hexahedra
  - Want something that work on non-trivial CAD models (without model dependent fine-tuning)
  - Preliminary results in this talk, not very successful
- Then, robust full-hex boundary layer, combinatorial approach
  - More complicated than expected, paused
- These days: quad meshing
  - Things work on non-trivial cases, happier (researcher) life :)

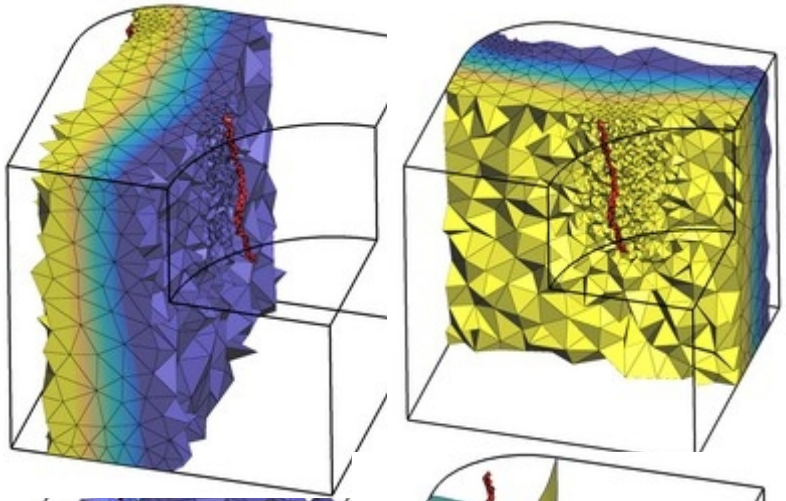
Dual-based block decomposition pipeline



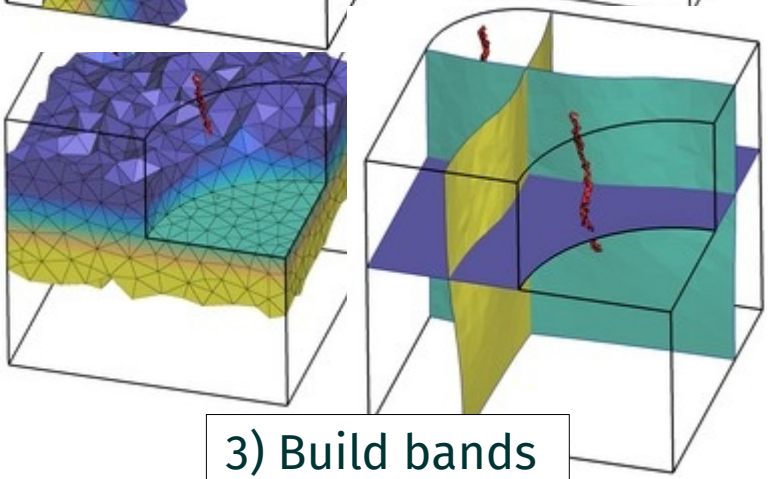
1) Frame Field



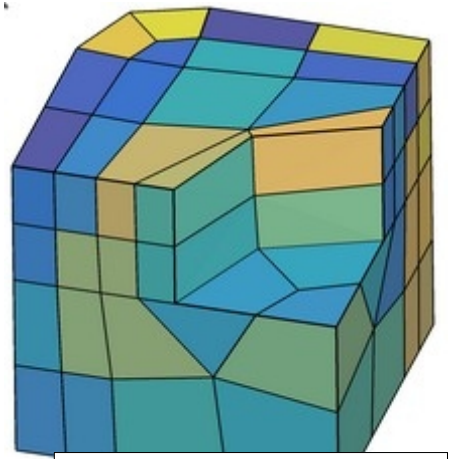
2) uvw-param



3) Build bands



4) Dual decomposition



5) Primalization

Why dual approach ?

- Motivations:
  - Tried primal approach and failed (*Mind the gap 2018*, early postdoc work)
  - Frame Field issues:
    - No corresponding hex mesh (3-5 singularities)
    - CAD features cannot be represented with frame boundary conditions
    - FF not directly integrable (unit vector fields)
  - With dual, circumvent singularities and model boundaries
  - Aim for partial hex-block decomposition, keep remaining non-hex polyhedra
- Related work:
  - *Hexahedral Meshing using midpoint subdivision, Li et al., CMAME 1995*
  - *Dual Surface Based Approach to Block Dcp,* Zheng et al., *IMR 2018*
  - *Loopy Cuts, Livesu et al., Arxiv 2019*

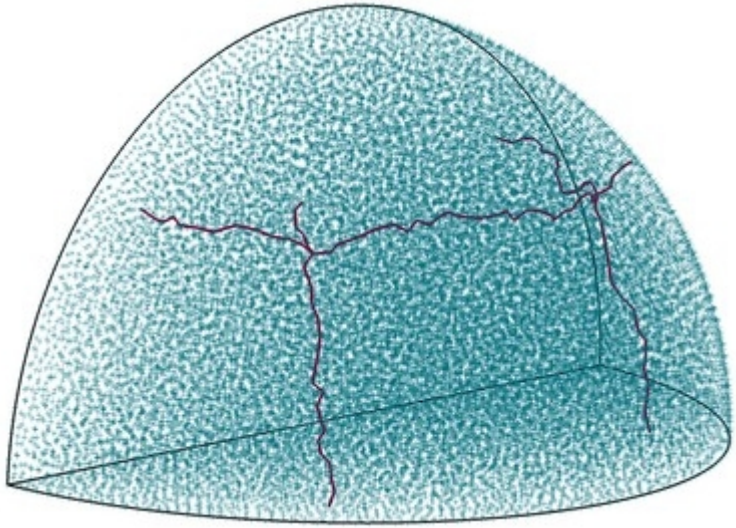
Why dual approach ?

- Motivations:
  - Tried primal approach and failed (*Mind the gap 2018*, early postdoc work)
  - Frame Field issues:
    - No corresponding hex mesh (3-5 singularities)
    - CAD features cannot be represented with frame boundary conditions
    - FF not directly integrable (unit vector fields)
  - With dual, circumvent singularities and model boundaries
  - Aim for partial hex-block decomposition, keep remaining non-hex polyhedra
- Related work:
  - *Hexahedral Meshing using midpoint subdivision, Li et al., CMAME 1995*
  - *Dual Surface Based Approach to Block Dcp,* Zheng et al., *IMR 2018*
  - *Loopy Cuts, Livesu et al., Arxiv 2019*

3D Frame Fields

- Standard formulation

$$\min \int_\Omega ||\nabla \mathbf{f}||^2$$
$$\mathbf{f} \ // \ \mathbf{n} \text{ on } \partial\Omega$$
$$\mathcal{F} = SO(3)/O$$



- Still an active research topic:

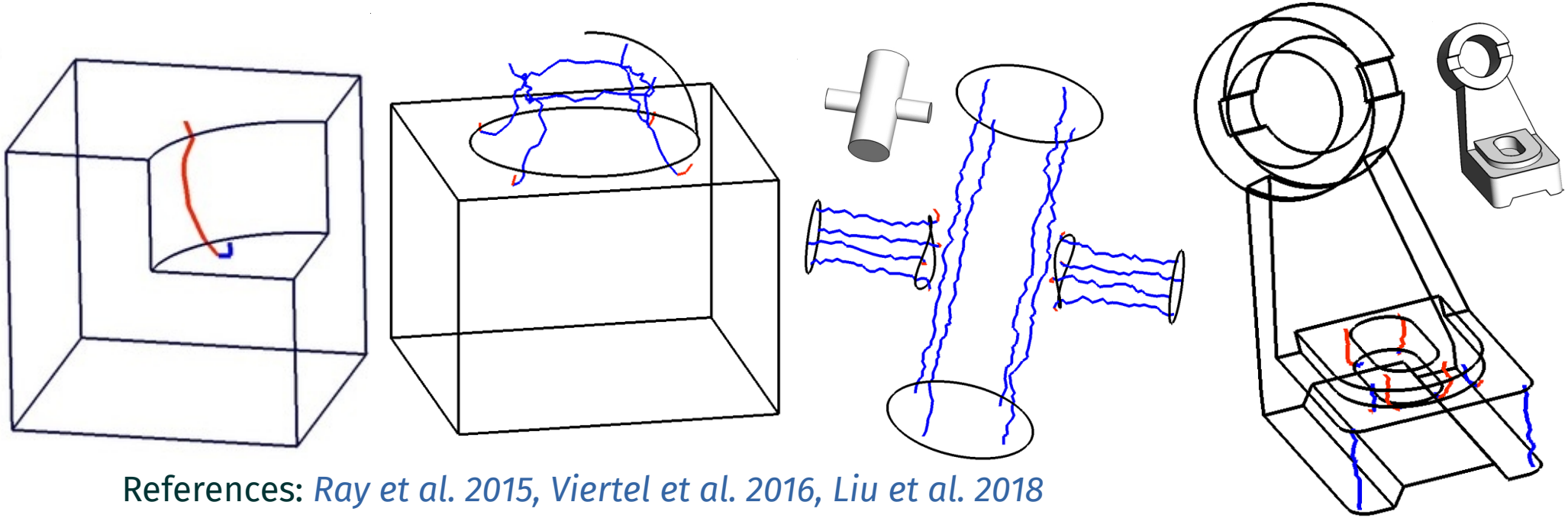
  Huang et al. 2011, Li et al. 2012, Ray et al. 2016, Solomon et al. 2017., Chemin et al. 2018, Palmer et al. 2019, Golovaty et al. 2019, etc

- If singularity graph known: *Liu et al. 2018, Corman et al. 2019*

- Main points:
  - Field of infinitesimal cubes (local geometry info)
  - Singularities are irregular edges of the block decomposition (global topological info)

Frame Field issue: non hex-meshable singularities

- Current frame field formulation produces “non hex-meshable” singularities, e.g.:

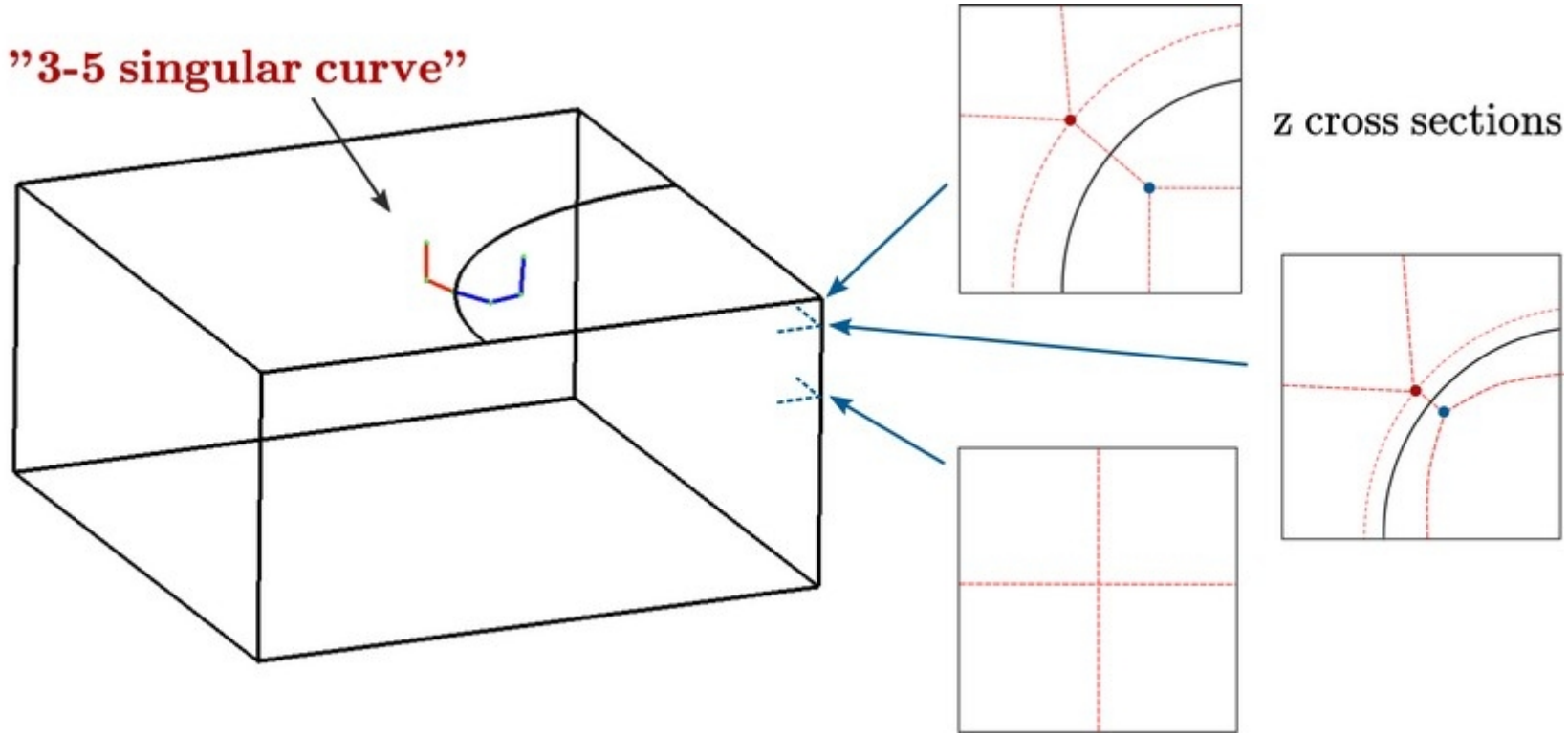


References: *Ray et al. 2015, Viertel et al. 2016, Liu et al. 2018*

- Specific to 3D, no equivalence in 2D

Frame Field issue: source of non hex-meshable singularities

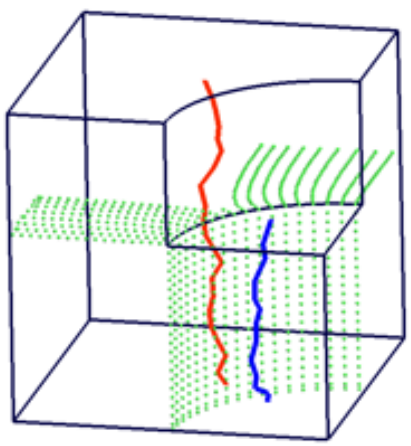
- Energy shortcut, singular curve not tangent with frame field



Frame Field issue: correction of non hex-meshable singularities

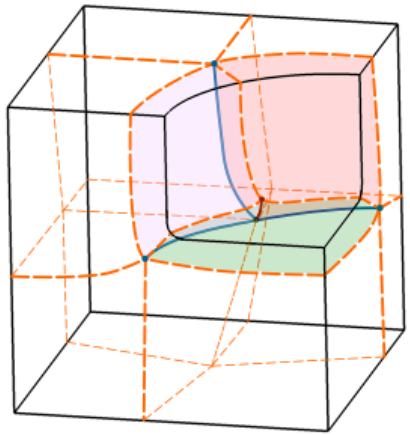
- Tried heuristic-based post-processing correction in previous IMR paper

Multiple approaches to frame field correction for CAD models, Reberol M., Chemin A., Remacle J.F., 2019



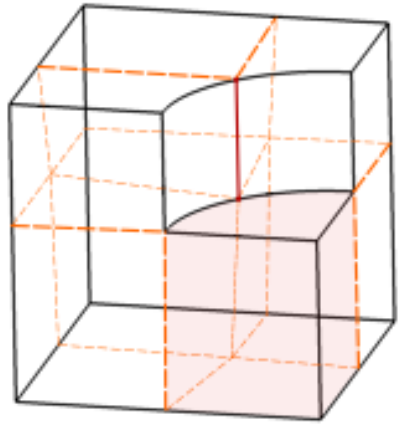
Feature extrusion

Not reliable



Feature smoothing

Increase complexity



Singularity snapping

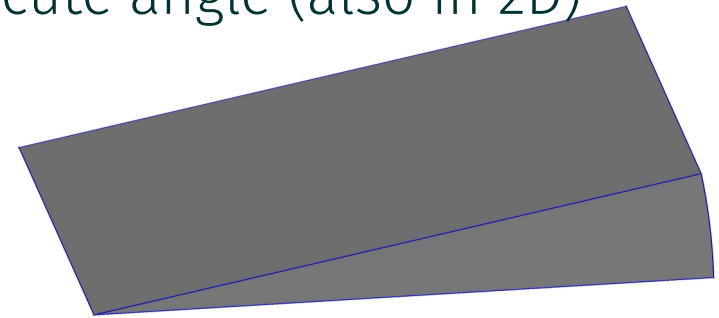
Degenerate uvw-param

- Poor solutions. Todo: frame field formulation without non hex-meshable singularities

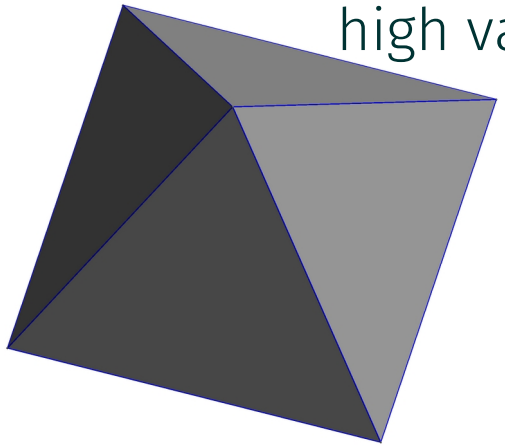
Frame Field issues: boundary conditions for CAD

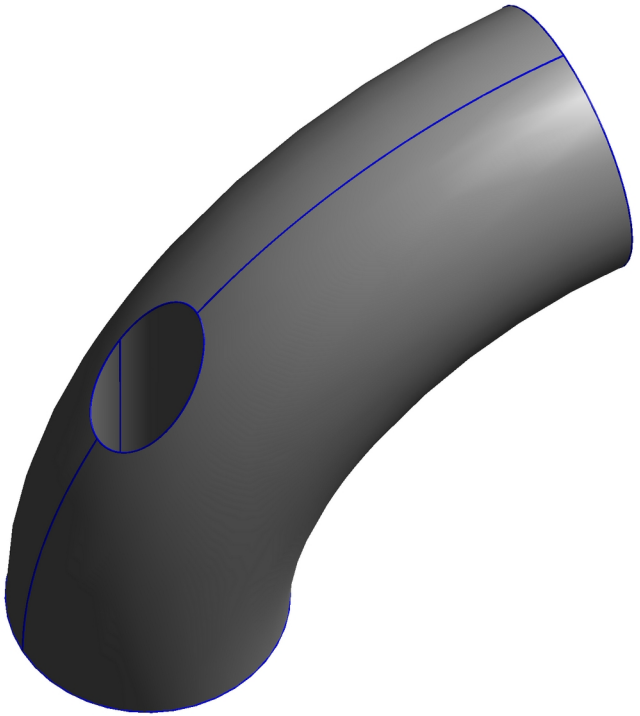
- Many CAD features cannot be represented by frames (3 orthogonal axis)

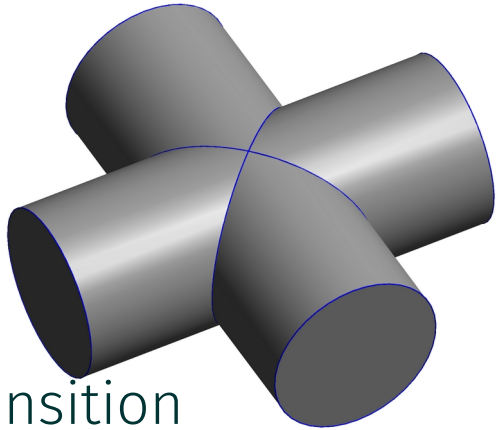
acute angle (also in 2D)



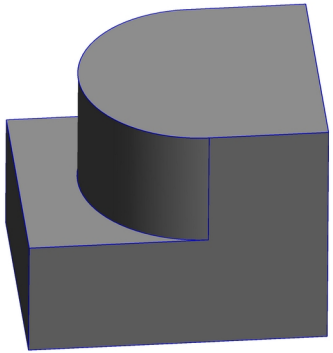
high valence corners





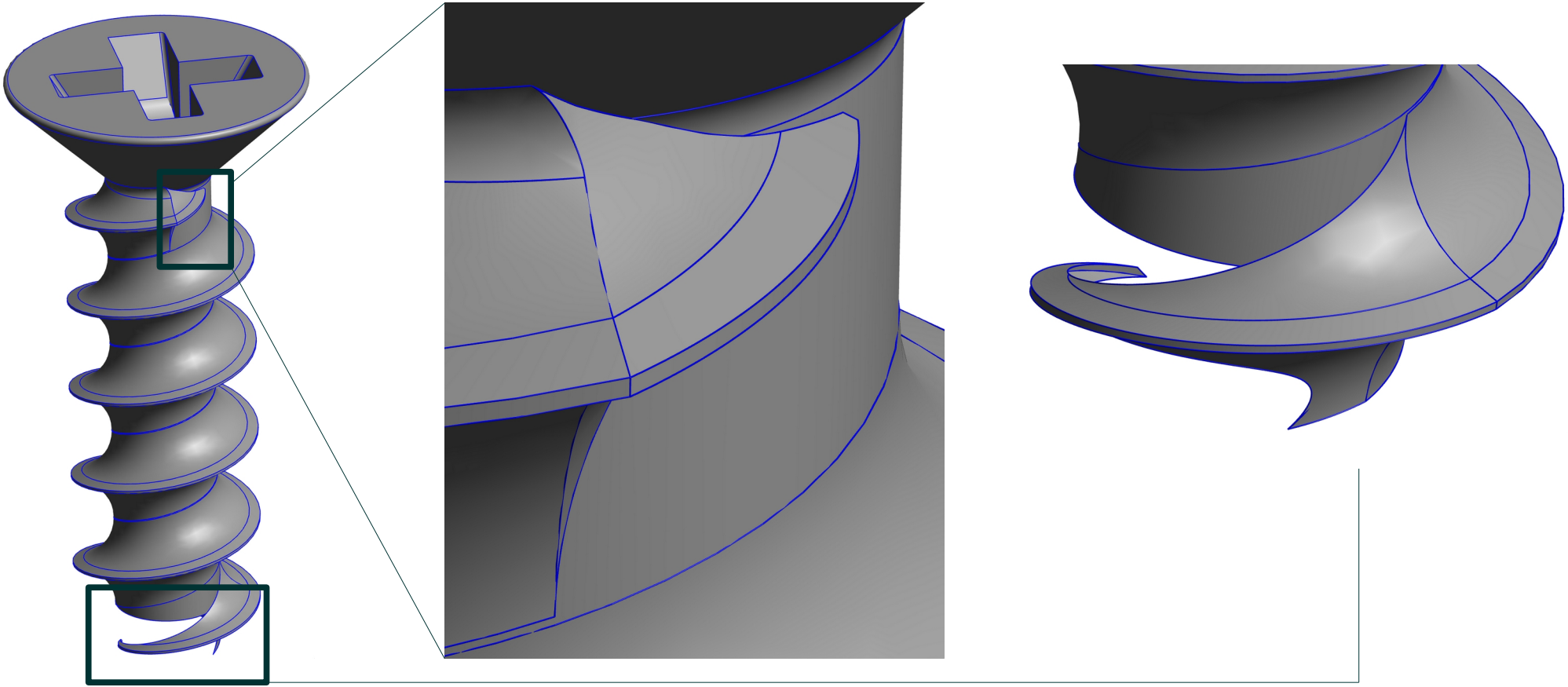


angle transition



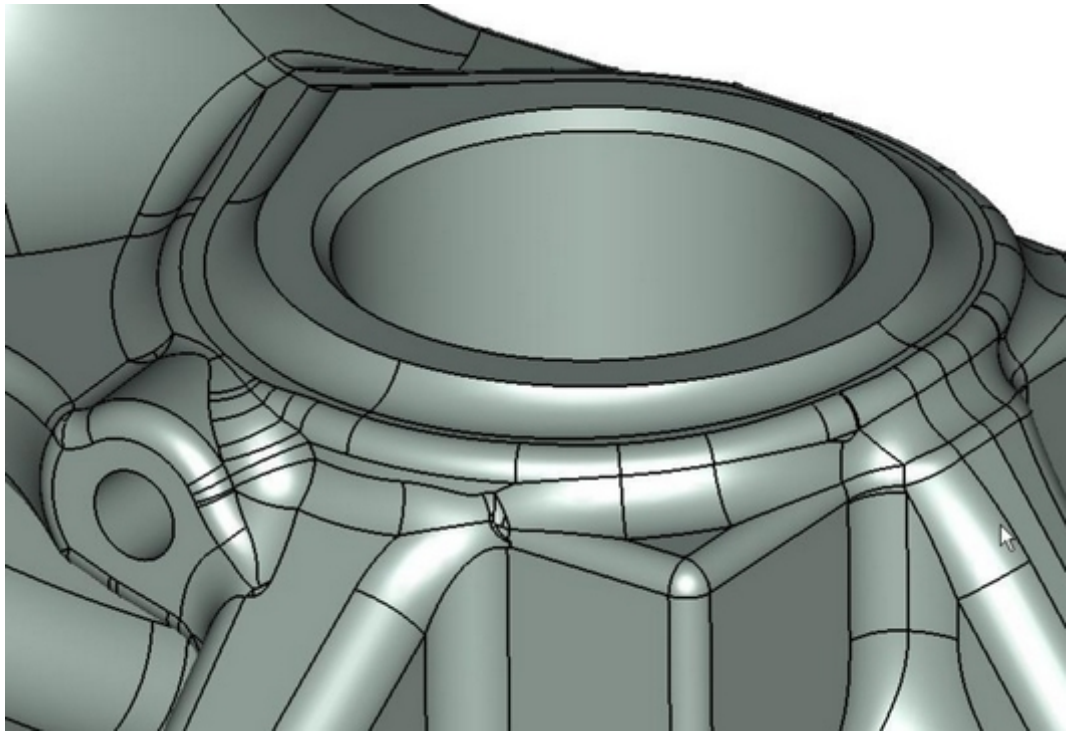
Frame Field issues: boundary conditions for CAD

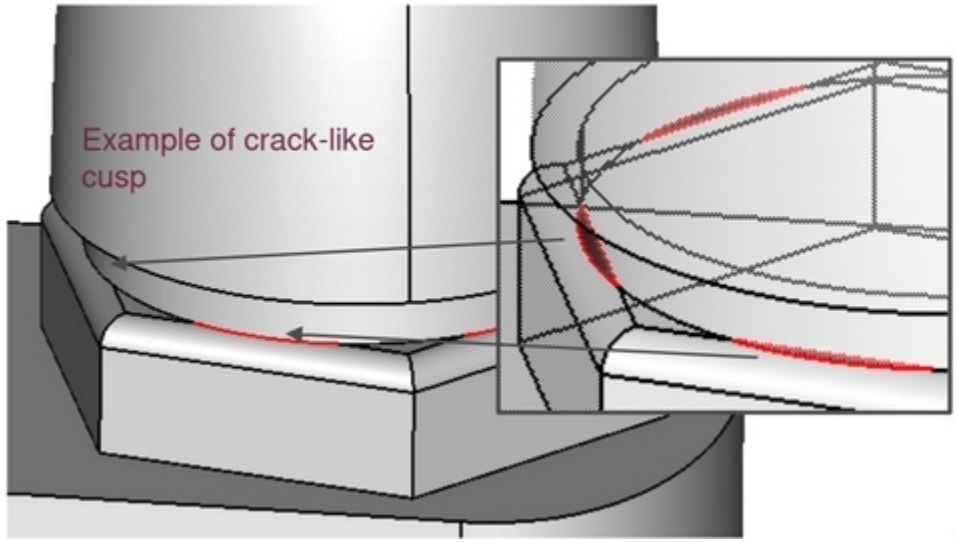
- Many CAD features cannot be represented by frames (3 orthogonal axis)



Real world CAD is complicated

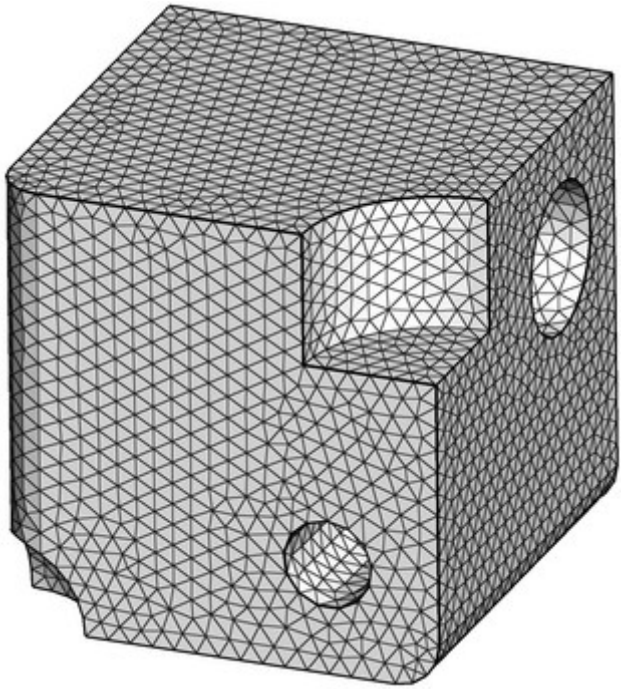
- Images from Mark Gammon (CADfix) talk at Tetrahedron workshop VI

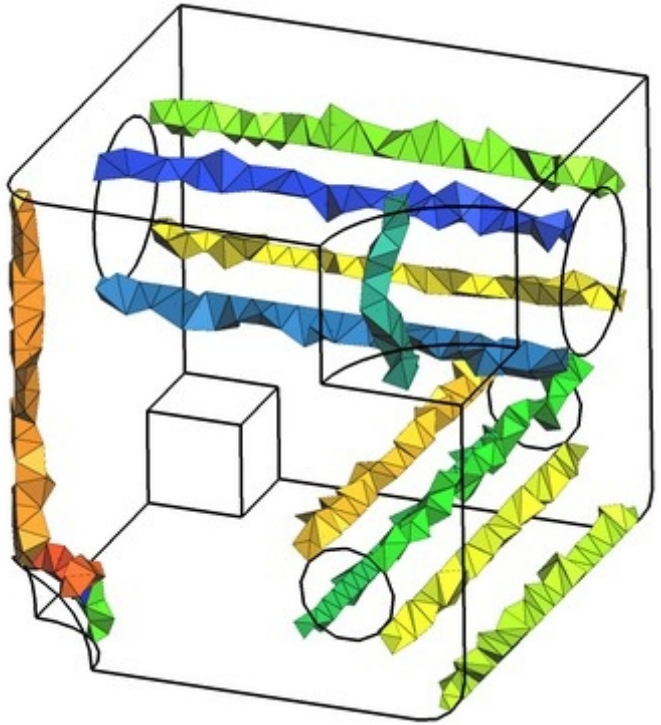




Frame Field: require sufficient resolution for exploitation

- In CAD, features (e.g. fillet) appear at various scales. FF must capture them.

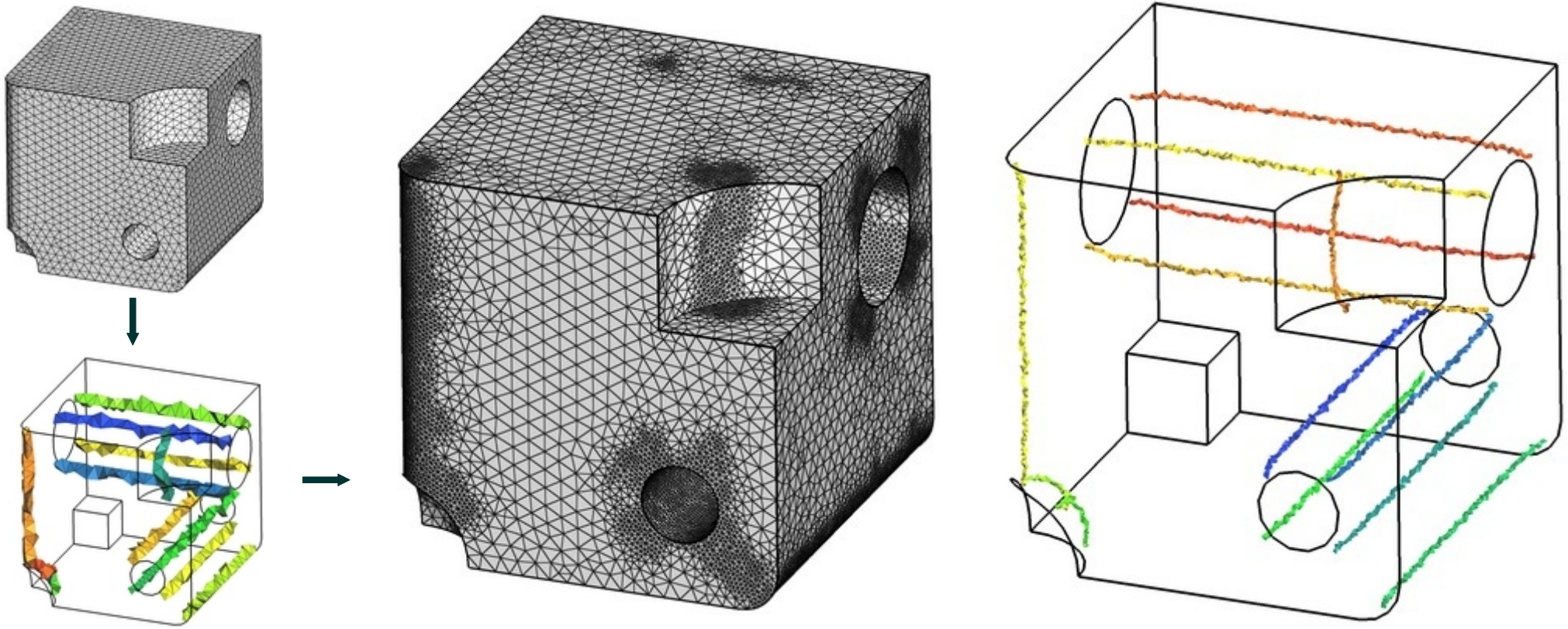




- Uniform refinement too expensive. Require adaptive mesh refinement.

Frame Field: require sufficient resolution for exploitation

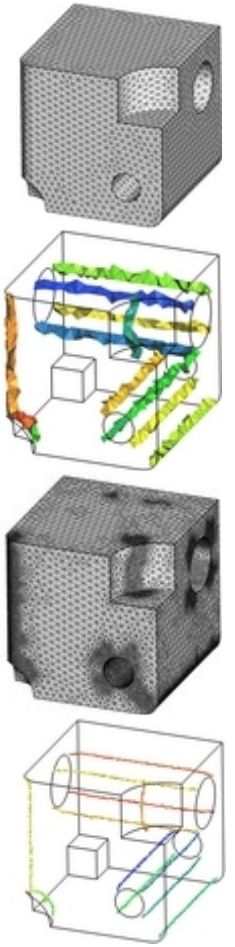
- In CAD, features (e.g. fillet) appear at various scales. FF must capture them.



Frame Field: require sufficient resolution for exploitation

- Issue: with most frame-field solvers, refinement repels singularities

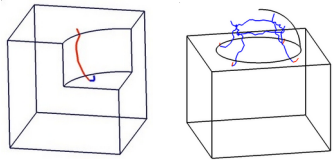
  (because minimizes Dirichlet energy, which tends to infininity at singularities)
- Current hackish algorithm:
  - Compute initial FF on coarse uniform mesh
  - Build a sizemap based on distance to singularities
  - Generate an adaptative mesh (with gmsh or mmg3d)
  - Project FF from coarse to fine mesh
  - Compute a new FF with initial guess (same local minimim)



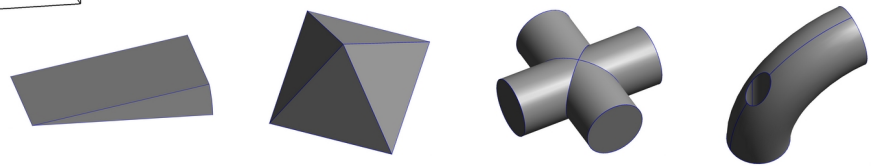
- Works if FF topology initially captured and not too much “repulsion”
- Better solution: mesh insensitive FF solver, Palmer et al. 2019?

Reasons for a dual approach

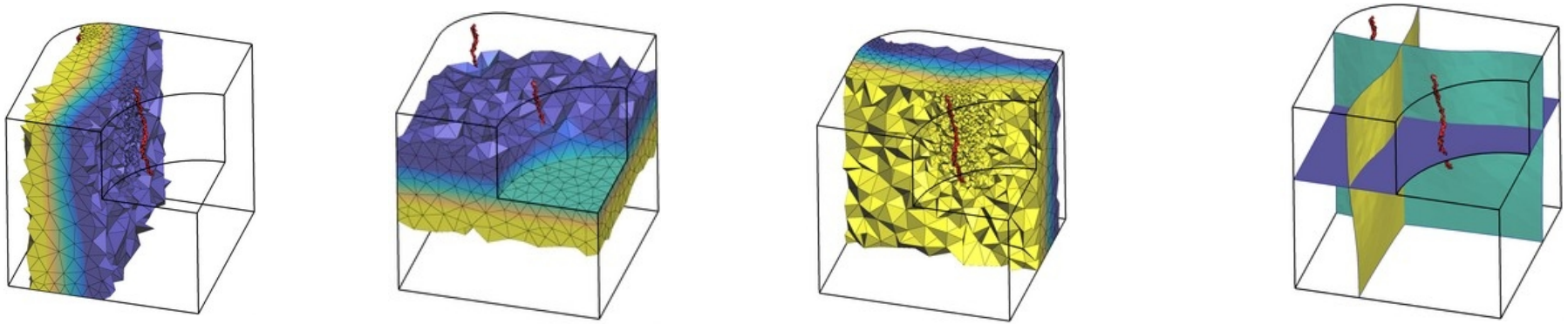
- Avoid 3-5 singular curves



- Avoid unrepresentable CAD features

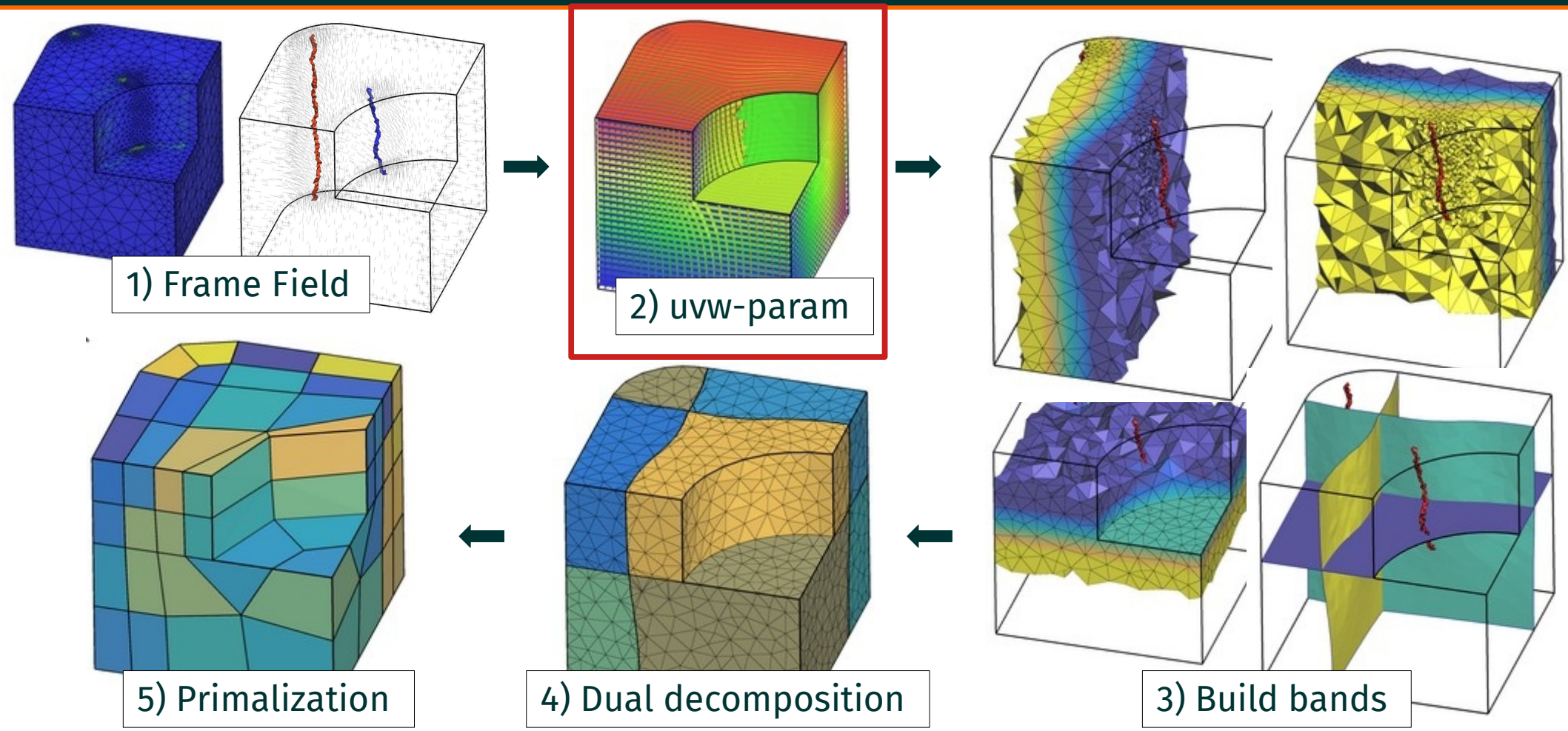


- Stay in smooth regions and avoid high gradient regions (singularities)



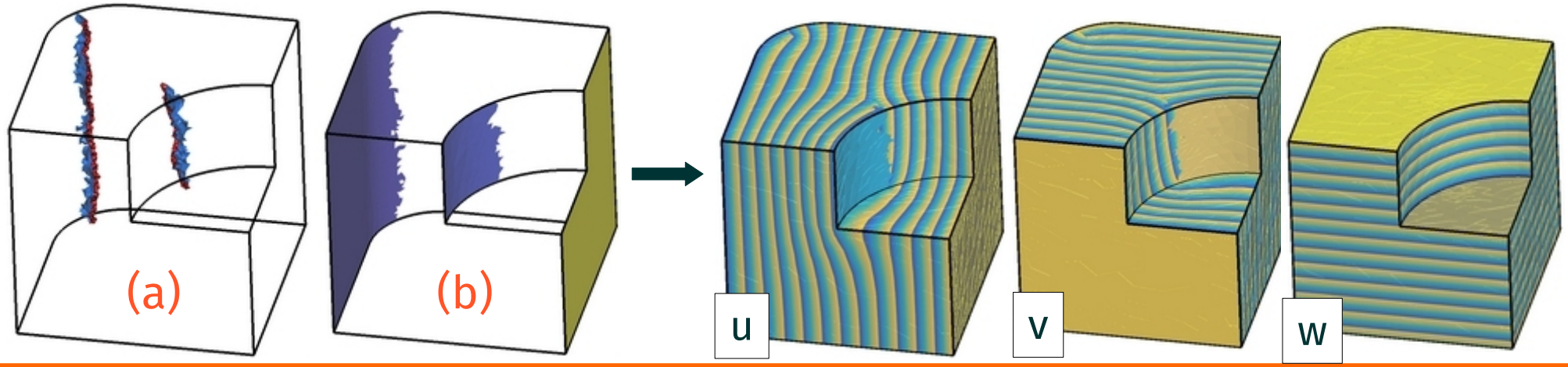
- Avoid mixed-integer (demanding, expensive) and HexEx (heavy post-processing)

Dual-based block decomposition pipeline

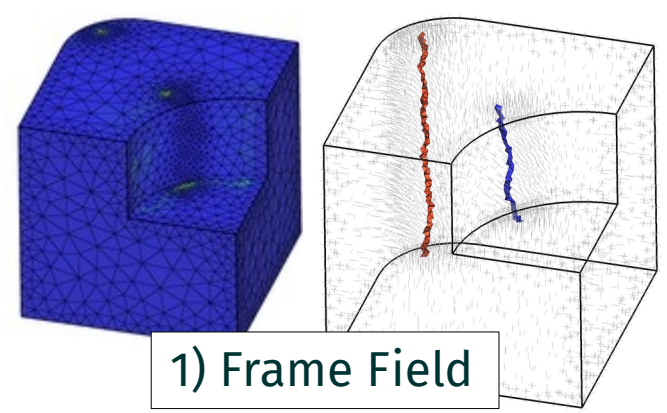


Step 2: uvw-parametrization

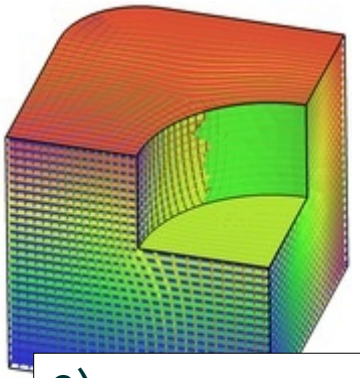
- Build cut-graph to get topological ball (cotree > primal > pruning) (a)
  → 3 continuous unit vector field on cut mesh: $\mathbf{F}_u, \mathbf{F}_v, \mathbf{F}_w$
- Vector field integration via least-square system:
  - Minimize alignment energy: $\min(\|\nabla u - \mathbf{F}_u\|^2)$ (not exact because unit vector field)
  - Boundary conditions: u or v or w constant on each boundary patch (b)
  - No constraints on singular tetrahedra
  - Constant jump constraint on cut-graph not very stable (disabled by default)



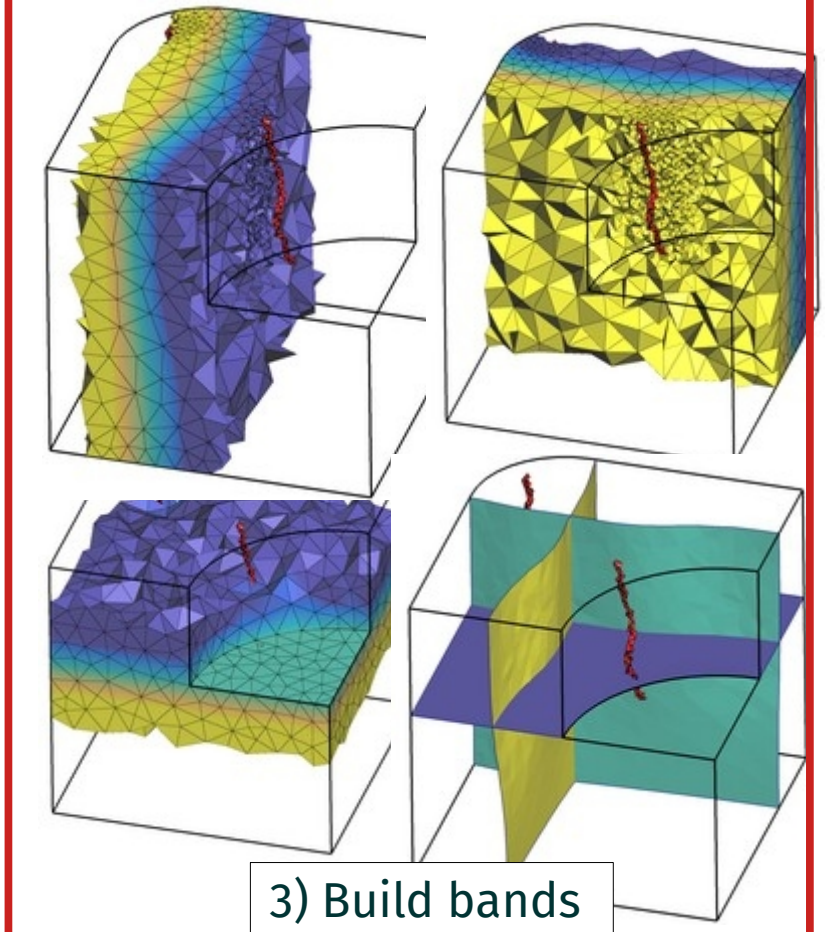
Dual-based block decomposition pipeline



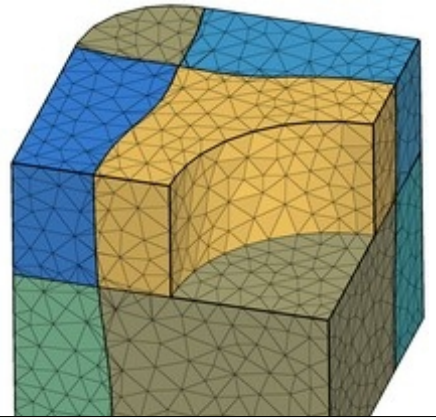
1) Frame Field



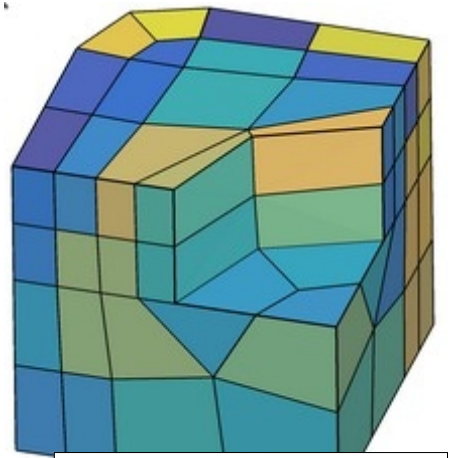
2) uvw-param



3) Build bands



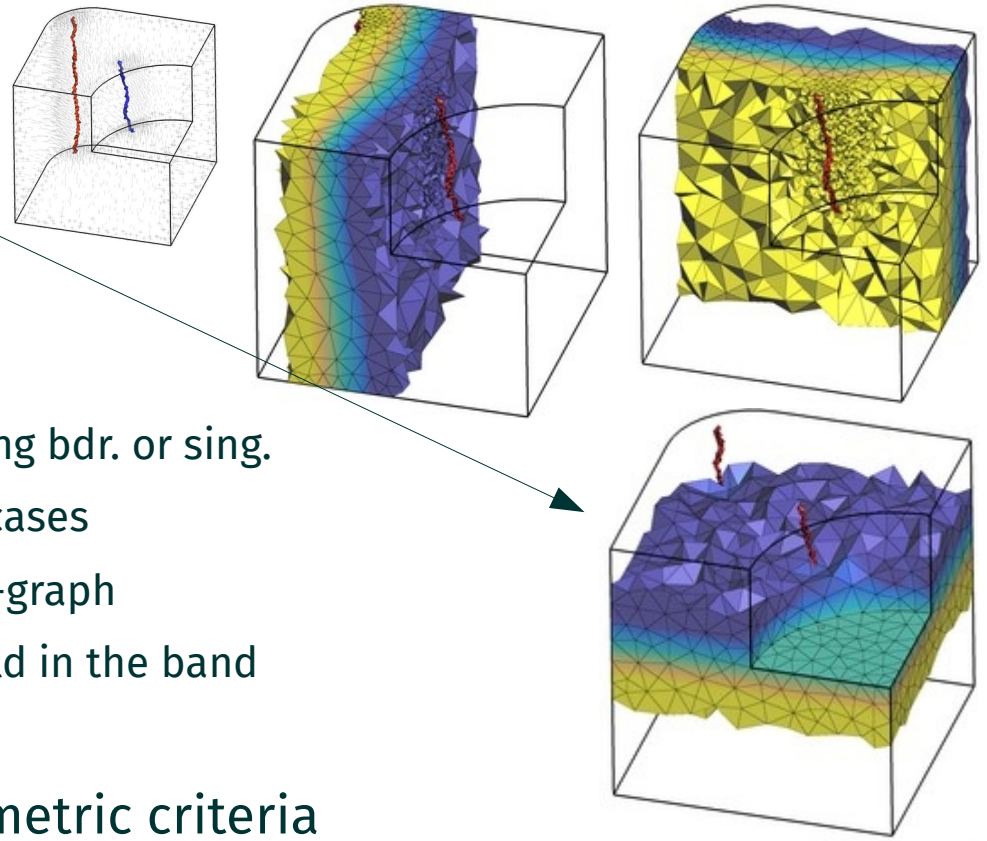
4) Dual decomposition



5) Primalization

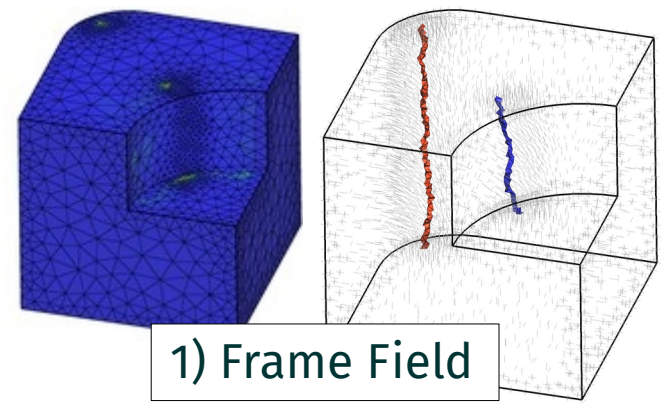
Step 3: build dual bands

- Band = largest possible interval [u_min,u_max] made with “extracted” scalar field
- Separate singularities
- Extend concave feature curves
- No band between bdr. and sing.
- BFS-based construction
  - Seeds from all paths between sing.
  - Propagate scalar field until tangentially reaching bdr. or sing.
  - Lot of heuristics (1k+ loc) to deal with specific cases
  - Rotation + jump + averaging when crossing cut-graph
  - Re-parametrization from continuous vector field in the band

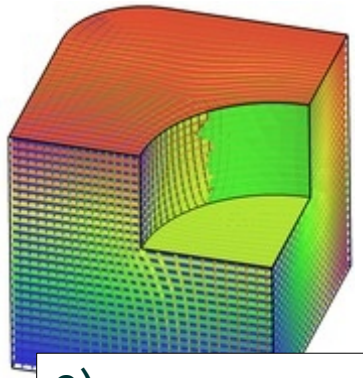


- Keep only bands which satisfy some geometric criteria

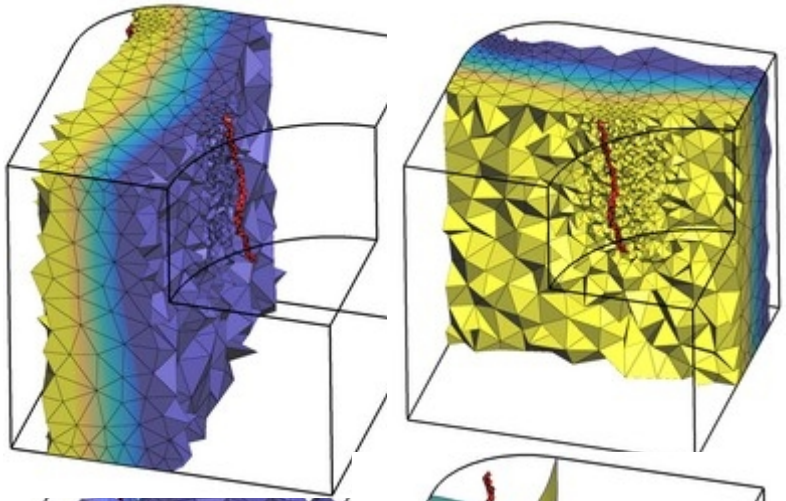
Dual-based block decomposition pipeline



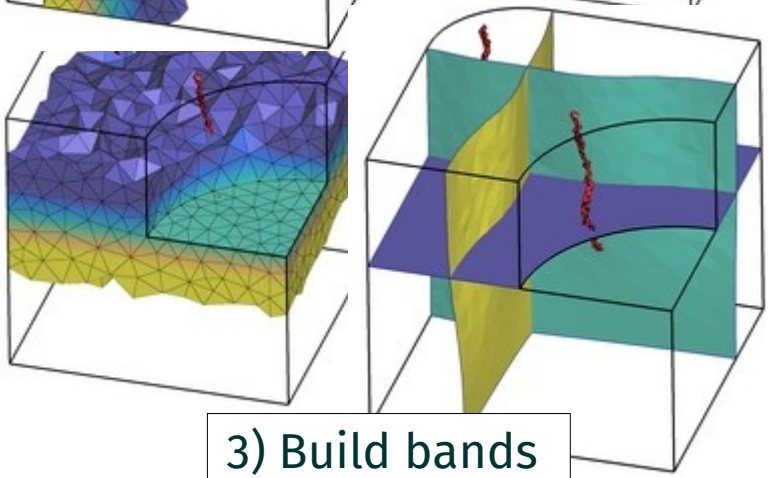
1) Frame Field

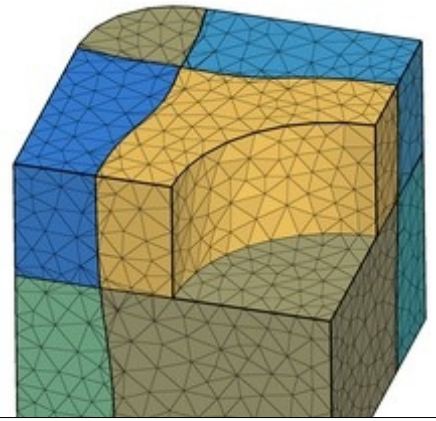


2) uvw-param

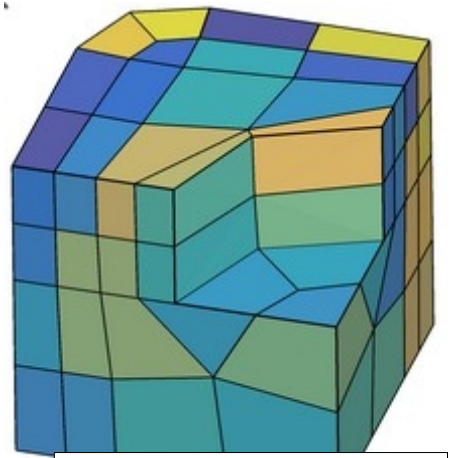


3) Build bands





4) Dual decomposition

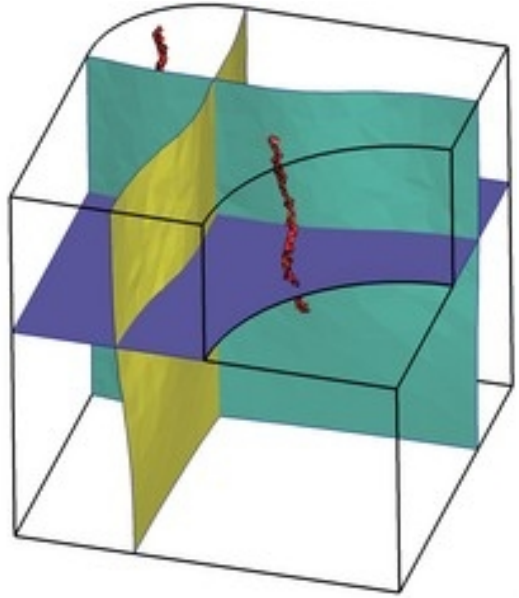


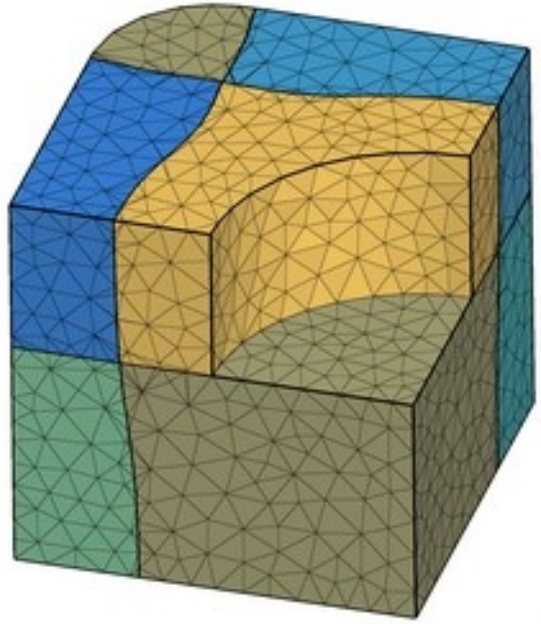
5) Primalization

Step 4: build dual decomposition

- Cut a tetrahedral mesh with each band isovalue 0

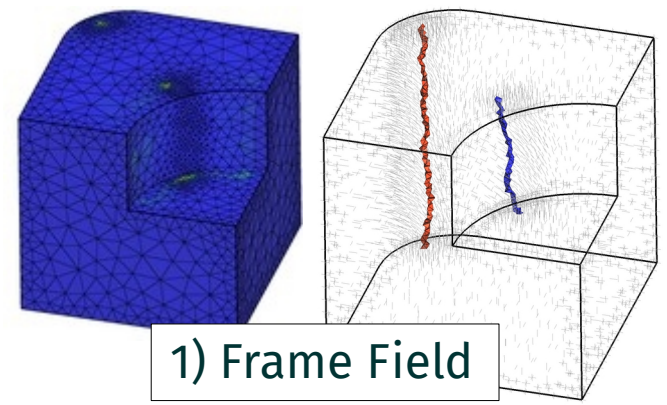
  (be cautions with numerical errors after successive cuts, using snapping and clamping tricks)



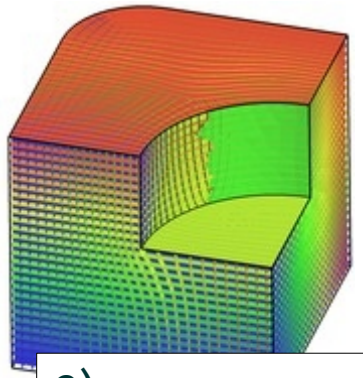


- Build the BRep representation of the polyhedral mesh
- Fail when one isosurface is not manifold

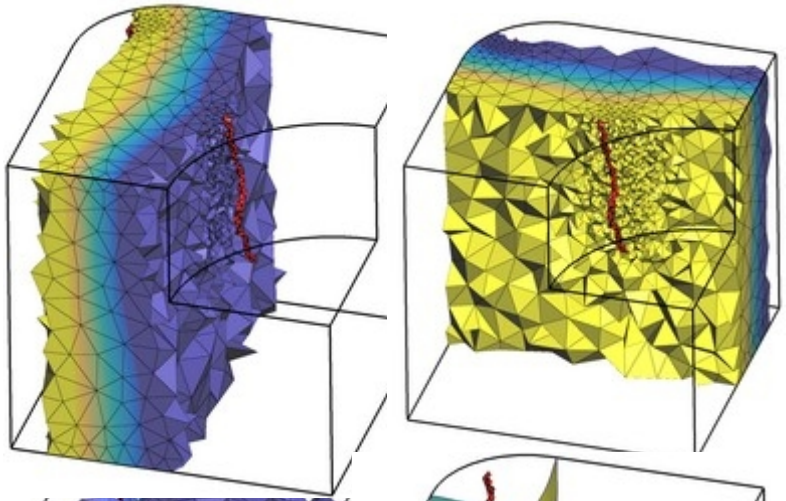
Dual-based block decomposition pipeline



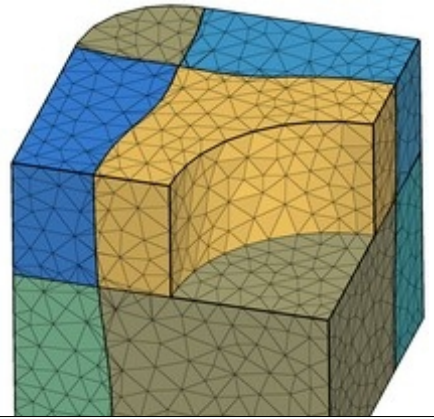
1) Frame Field



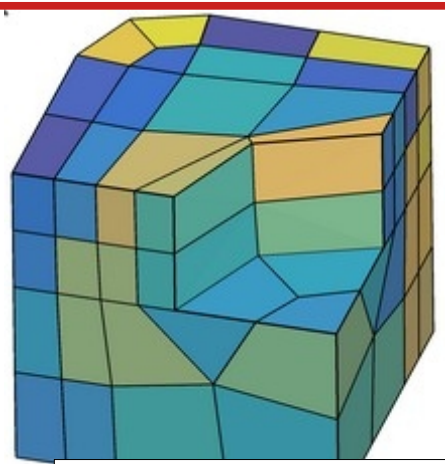
2) uvw-param



3) Build bands



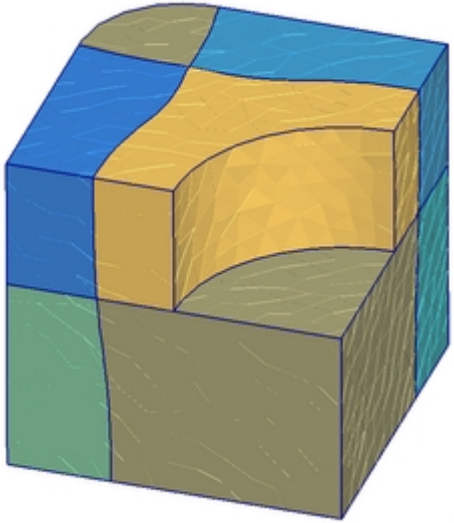
4) Dual decomposition

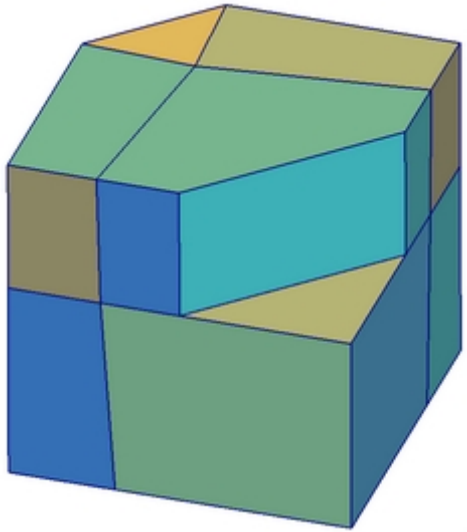


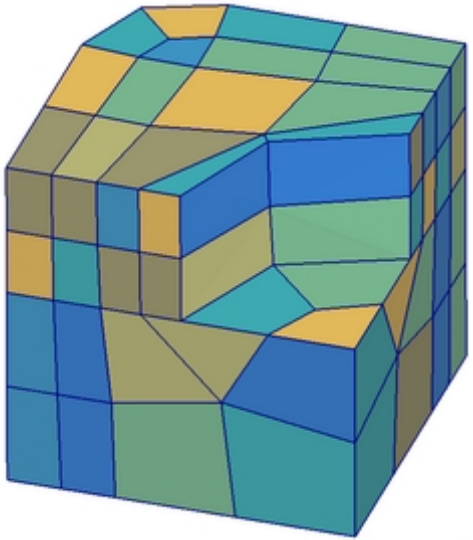
5) Primalization

Step 5: primalization to get final block decomposition

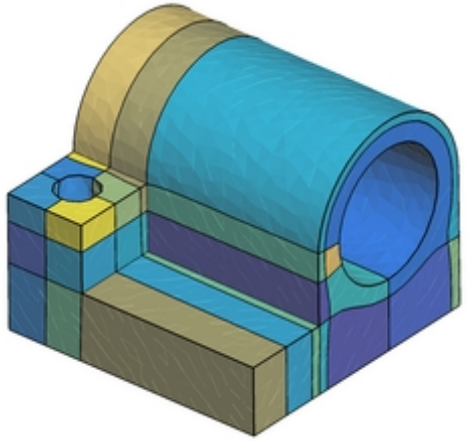
- Apply midpoint subdivision
  - *Hexahedral Meshing using midpoint subdivision, Li et al., CMAME 1995*
  - Hex blocks if all cell corners are valence three in the dual (polyhedral) mesh
- For non “midpoint subvisible” cells, plan to use:
  - *Finding hexahedrizations for small quadrangulations of the sphere, Verhetsel et al., TOG 2019*

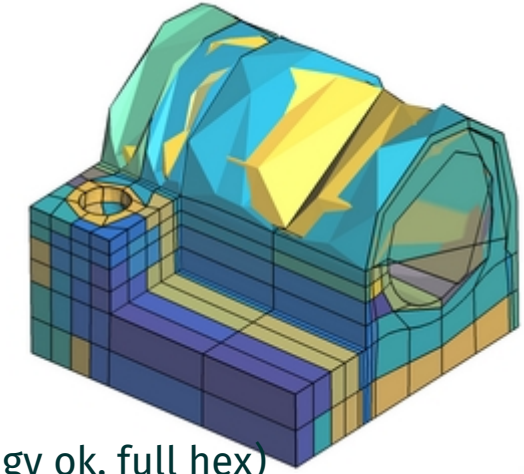




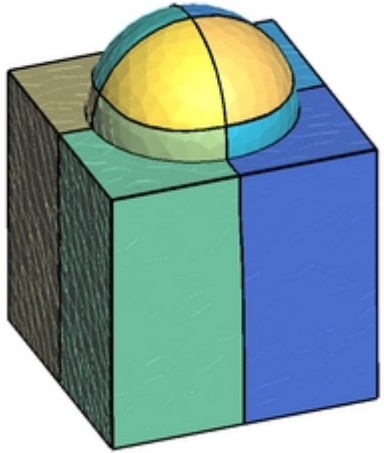


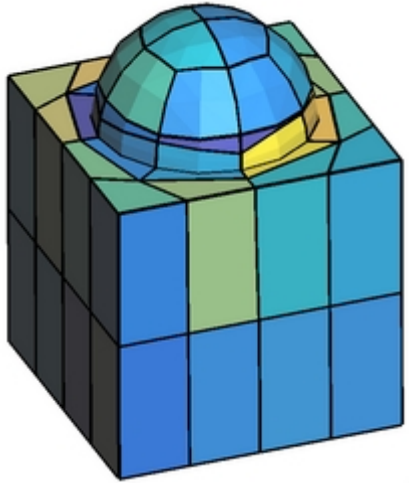
Block decomposition: successful applications

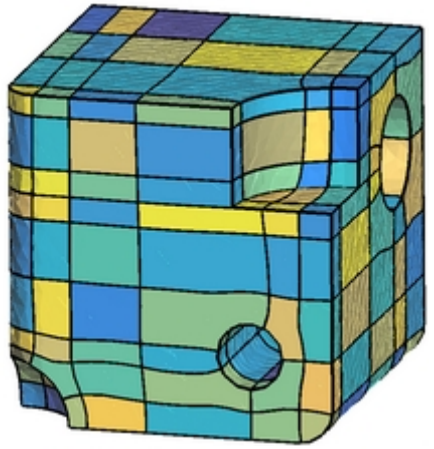


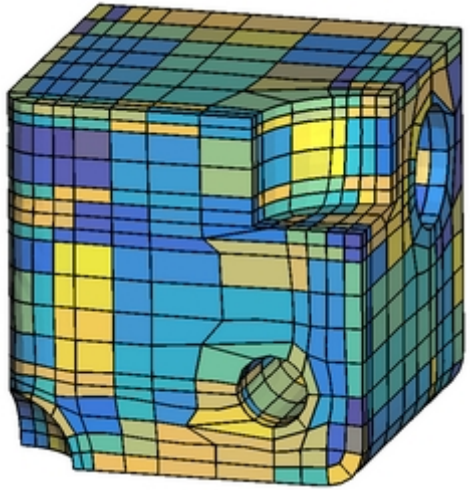


(topology ok, full hex)

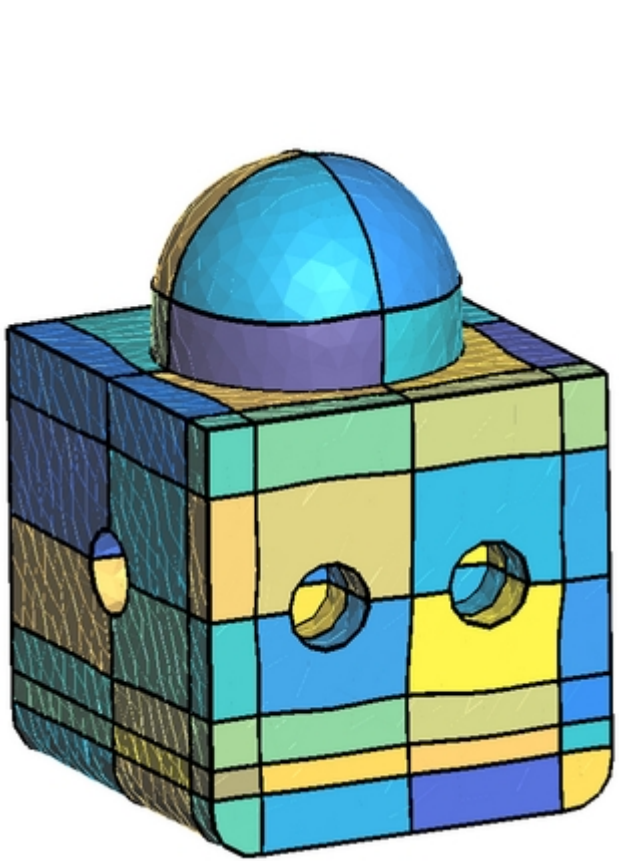




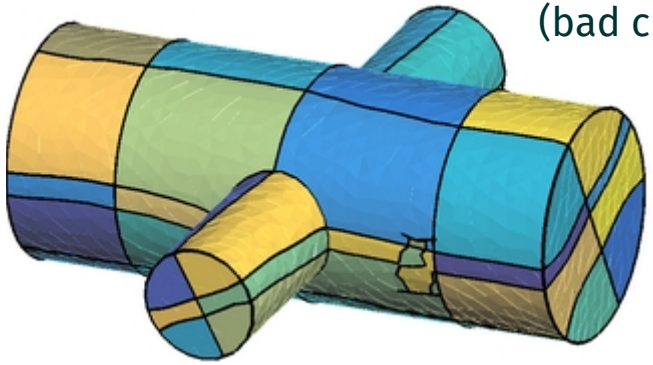


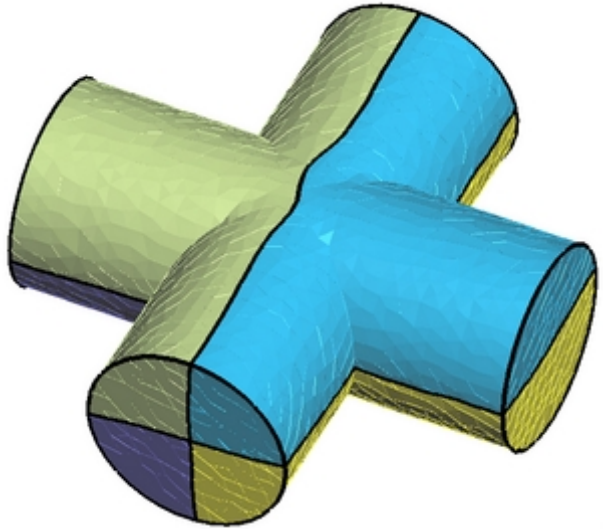


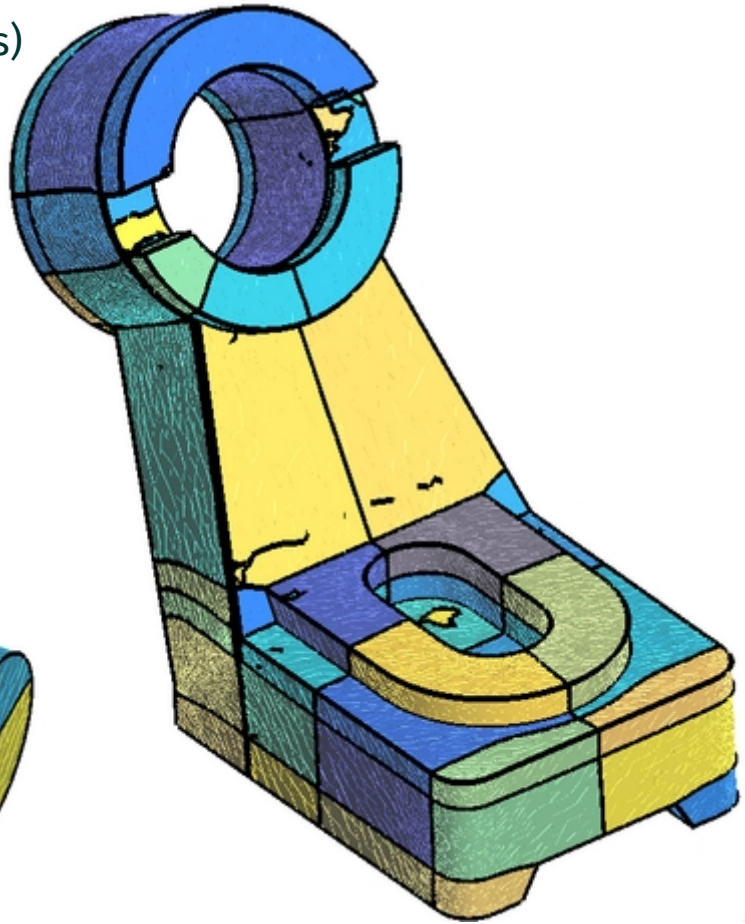
Block decomposition: failures and bugs (most of the cases)



(bad cuts)







Major issue: automation and robustness

- Lot of steps, many are not robust
- Almost never work “automatically” on a new model
- Typical issues and manual interventions:
  - Manually choose tet mesh sizing (to get sufficient FF resolution and not too much DOFs)
  - Verify CAD feature flagging is “ok” (else, adjust threshold angles on classification, etc)
  - Verify FF extracted singularities are “ok” (else, try again with another resolution or change thresholds)
  - Verify uvw-param is not garbage (else, try again with another resolution)
  - Verify dual bands are “ok” (else, change threshold in geometric criteria)
  - Primalization will fail if dual BRep is not perfect

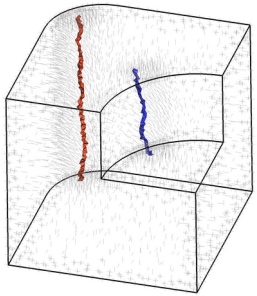
    (e.g. one dual curve (among hundreds/thousands) has 3 extremities)

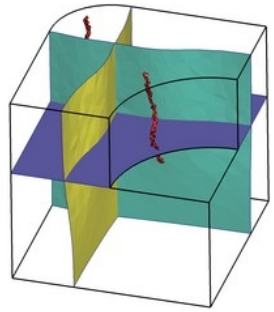
Conclusion and perspectives

- On 3D Frame Fields
  - Presence of non meshable singularities is still a major issue (fixable)
  - Would be nice to have convergent mesh adaptivity scheme (already possible ?)
  - No idea / hope for non-representable CAD features

- On dual-based block decomposition approach
  - Some preliminary results, would require more work (automation / robustness)
  - Difficult to get right (lot of engineering, not much theory)
  - Will explore cut surfaces unrelated to frame fields

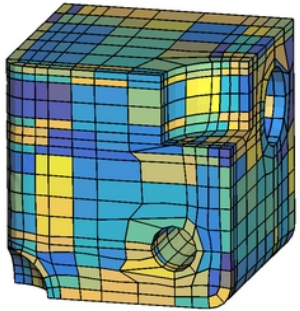
- Main mistake: accumulating geometric criteria and fixes is a never ending process ...

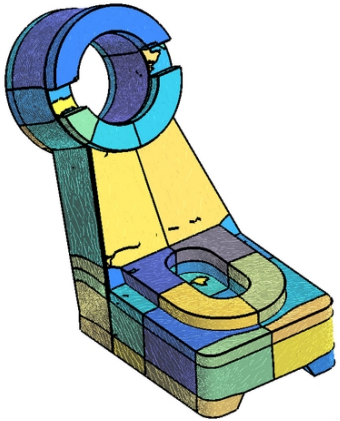




Thank you for you attention

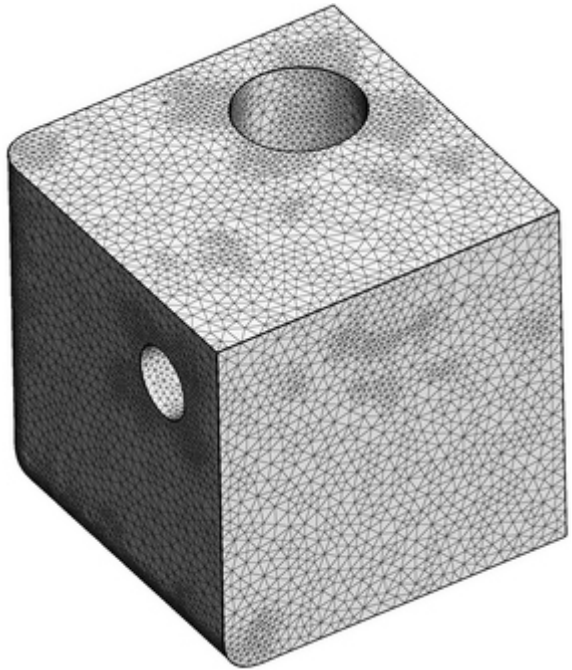
Questions ?

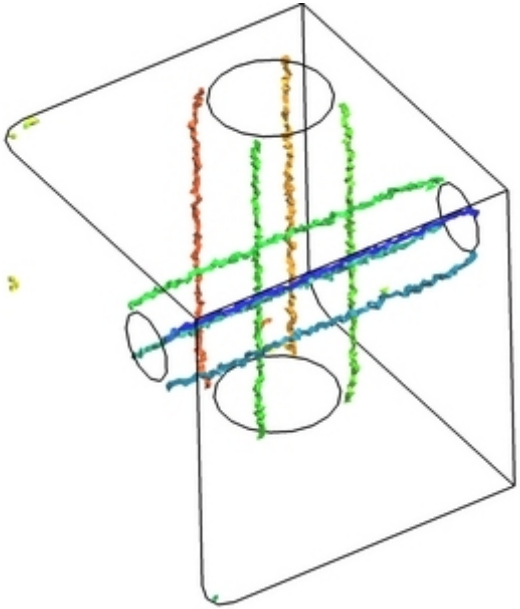


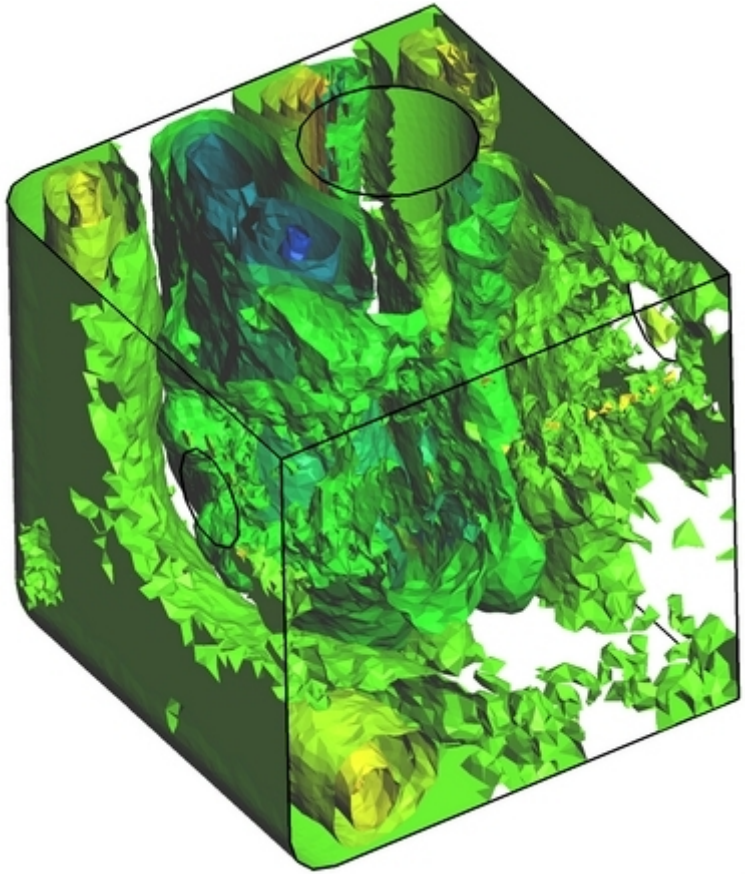


Block decomposition: pipeline is not robust

- Issue: wrong singularity flagging -> wrong cut-graph -> wrong uvw

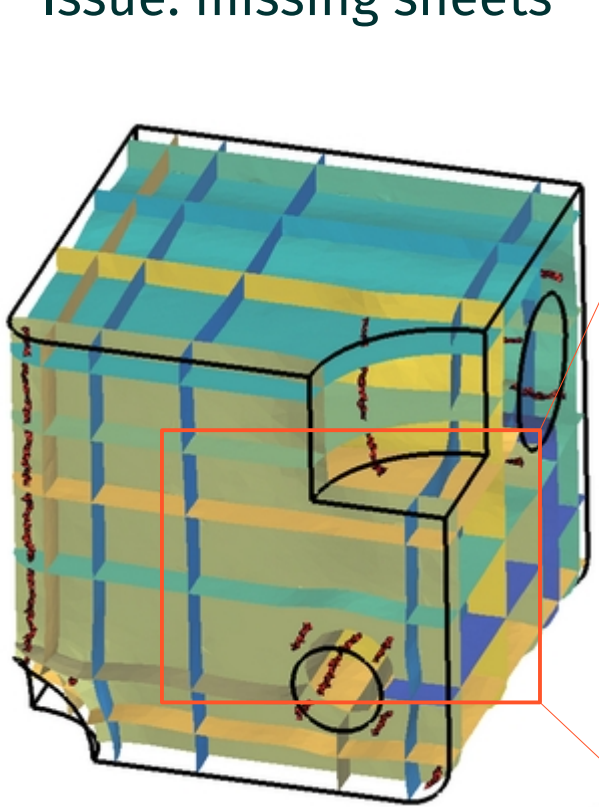


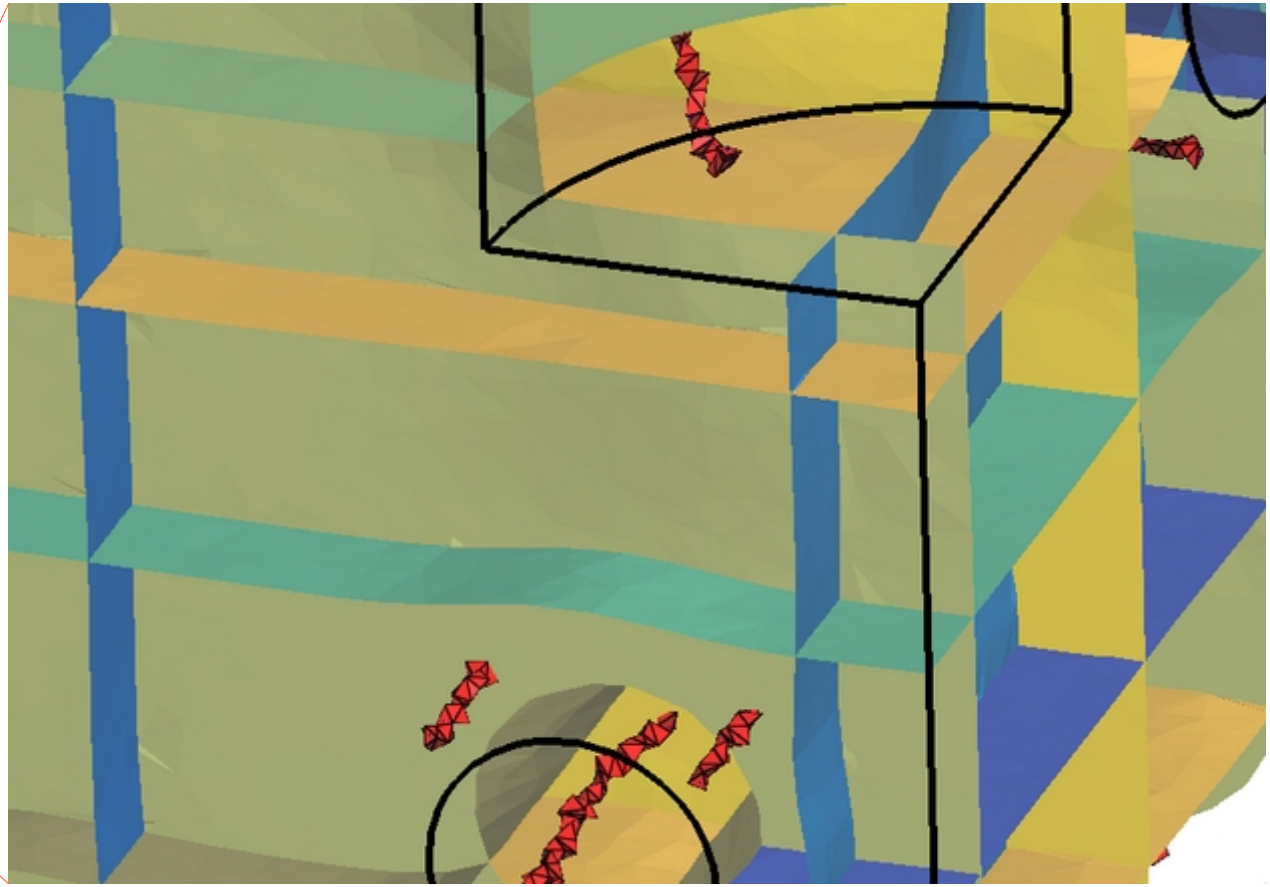




Block decomposition: pipeline is not robust

- Issue: missing sheets





Block decomposition: pipeline is not robust

- Issue: missing sheets