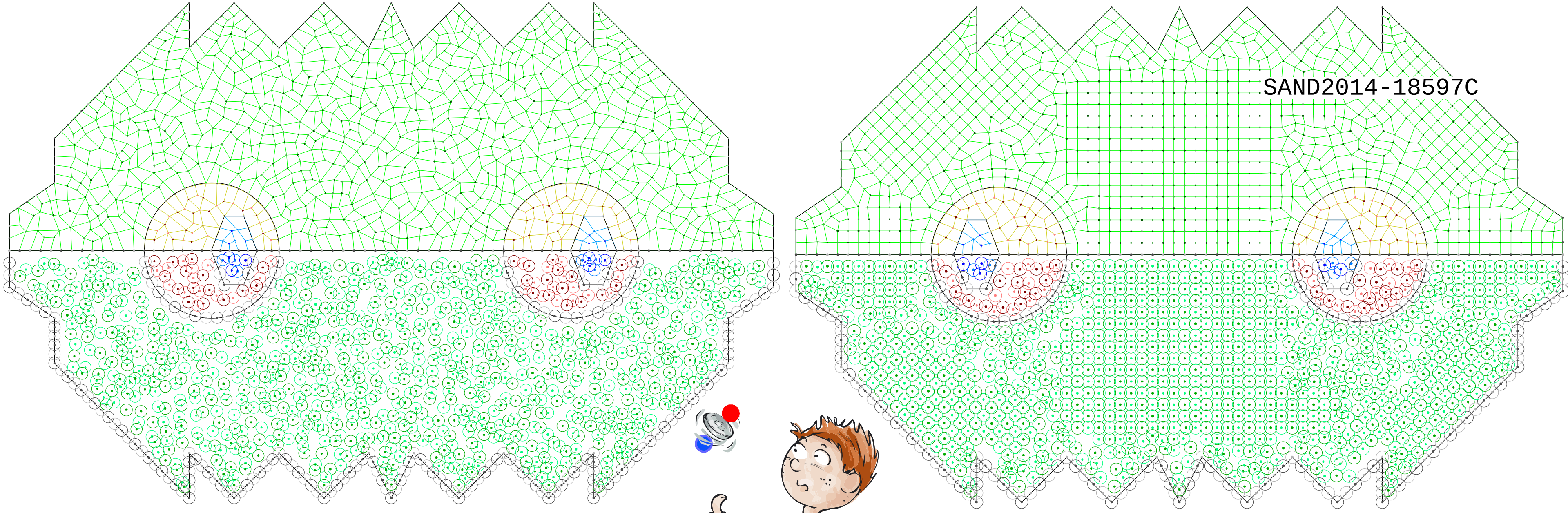


Delaunay quadrangulation by two-coloring vertices

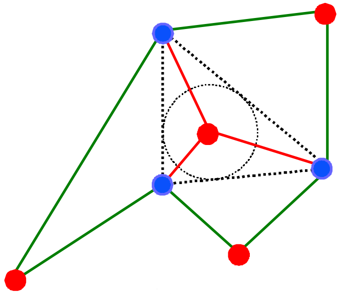
or

How to turn a triangle mesh into a quad one with no global problems

Scott A. Mitchell, Mohammed Amr Mohammed,
Ahmed Hassan Mahmoud, Mohamed Salah Ebeida

Computing Research, Sandia National Labs

Alexandria University



23rd International Meshing Roundtable (IMR23)
London, 15 Oct 2014
11:30-11:55am Wed, 20 minutes

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy's National Nuclear Security Administration
under contract DE-AC04-94AL85000.

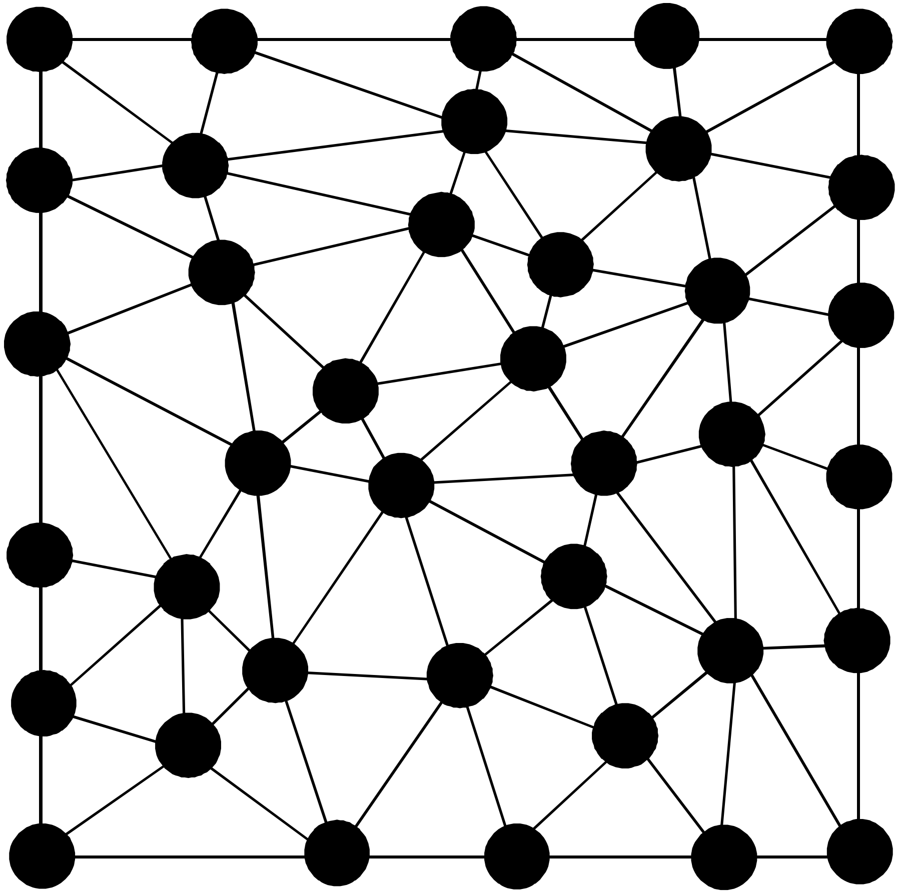
Outline

- Algorithm on one page
- Two-coloring idea
  - Contrast to Triangle Pairing
  - Even-sided polygons
  - Templates for provable quality (appendix on website)
    - Fixing by constrained incircle refinement
      - QTran
    - Constrained median refinement
- (Random Algorithm)
  - Well-spacedness properties, achieved by
  - Delaunay refinement triangulation as input
  - MPS (sphere packing) for provable quality triangulation
    - center director asks Mohamed, “can you do this for quad meshes?” no, but two years later...
  - Generating multi-class blue noise
  - Ideal spacing
  - Heuristics for better quality
- Example meshes
  - Dare to show raw output, before cleanup
- Advancing Front Algorithm
  - Row, column, repeat. Reseed.
- Conclusions
  - Three centers: Circumcenter, Incenter, Centroid
    - orthocenter feeling left out
  - Some quad meshes are not two-colorable
  - Not for hex meshes

Convert tri mesh to quads
One-slide Algorithm

- **Generate (or given) well-spaced points**
- **Delaunay triangulate**
- Color points red or blue
  - intersperse colors
- Discard red-red and blue-blue edges
- Quads mostly
  - good quality, some large angles
- 6, 8, 10 sided polygons sometimes
  - constrained incircle refinement
  - median template for reflex quads
- All quads with provable quality

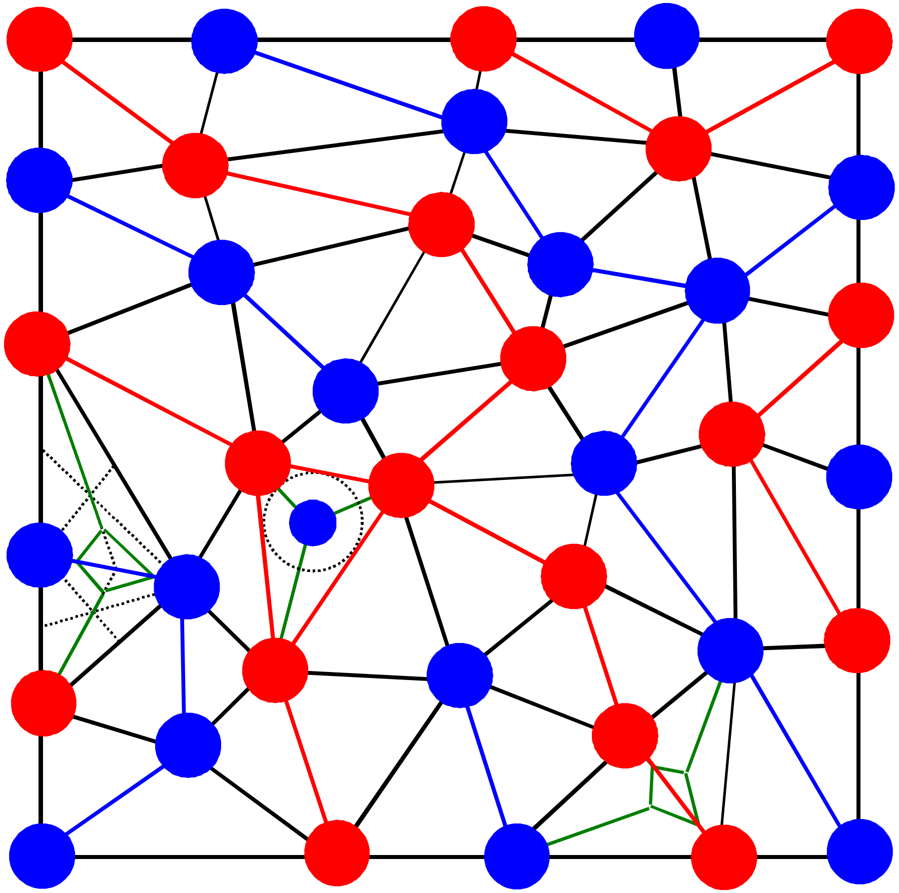
- Coloring and position heuristics improve quality in practice



Convert tri mesh to quads
One-slide Algorithm

- Generate (or given) well-spaced points
- Delaunay triangulate
- Color points red or blue
  - intersperse colors
- Discard red-red and blue-blue edges
- Quads mostly
  - good quality, some large angles
- 6, 8, 10 sided polygons sometimes
  - constrained incircle refinement
  - median template for reflex quads
- All quads with provable quality

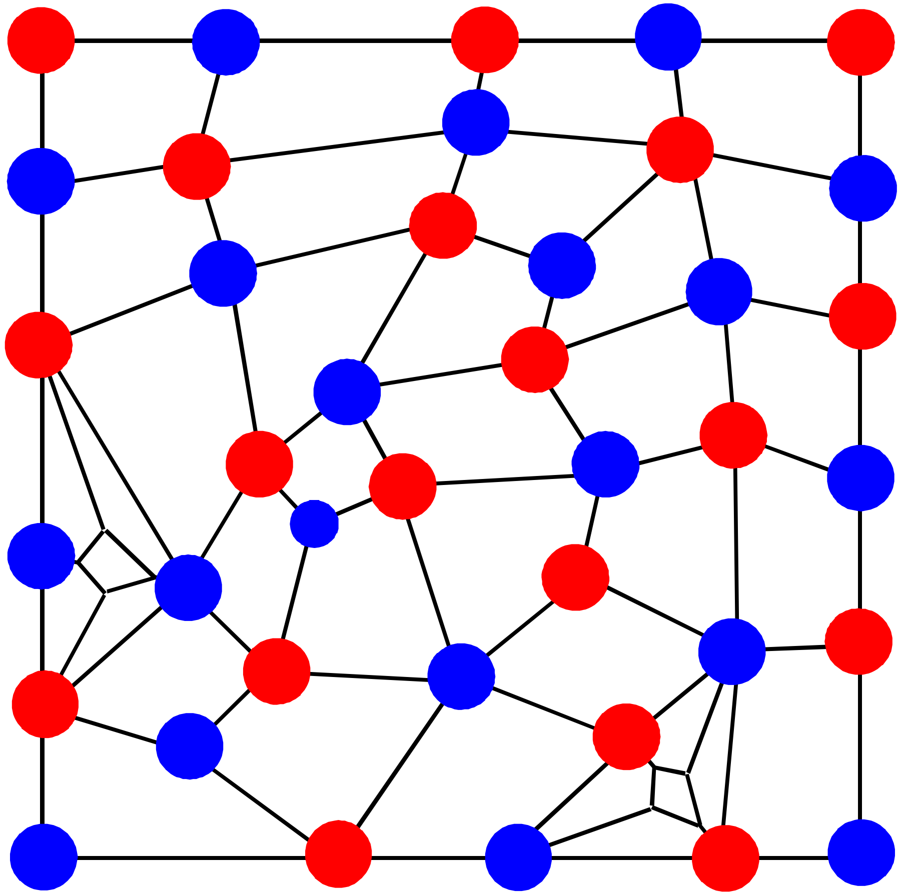
- Coloring and position heuristics improve quality in practice



Convert tri mesh to quads
One-slide Algorithm

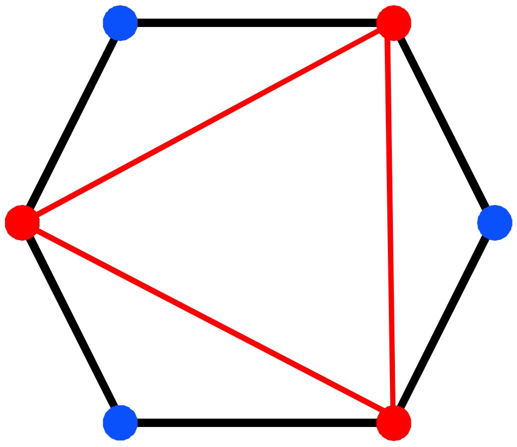
- Generate (or given) well-spaced points
- Delaunay triangulate
- Color points red or blue
  - intersperse colors
- Discard red-red and blue-blue edges
- Quads mostly
  - good quality, some large angles
- 6, 8, 10 sided polygons sometimes
  - constrained incircle refinement
  - median template for reflex quads
- **All quads with provable quality**

- **Coloring and position heuristics improve quality in practice**



Why does this work?

- Any planar quad mesh can be two-colored
  - converse, convert a two-colored graph into quads
- Two-coloring makes even-sided cells
  - 4, 6, 8, 10...
- Any triangulation of 6+ polygon with monochromatic edges has a monochromatic triangle
  - avoid or remove mono-triangles to get all quads

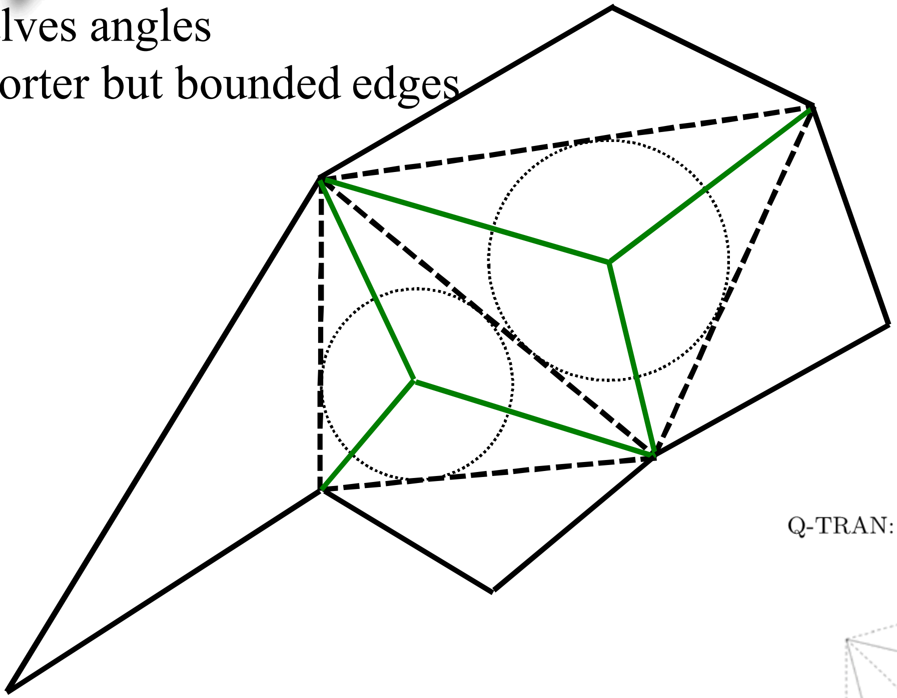


This is the only hexagon triangulation using only red edges, since every blue vertex must be in an ear

- Input tri good quality
  - template quads good quality (proofs in 8 page appendix, available online)
    - http://www.cs.sandia.gov/~samitch/papers/delaunayquadproof.pdf
    - google: mitchell sandia. click on papers

Incircle refinement details

halves angles
shorter but bounded edges

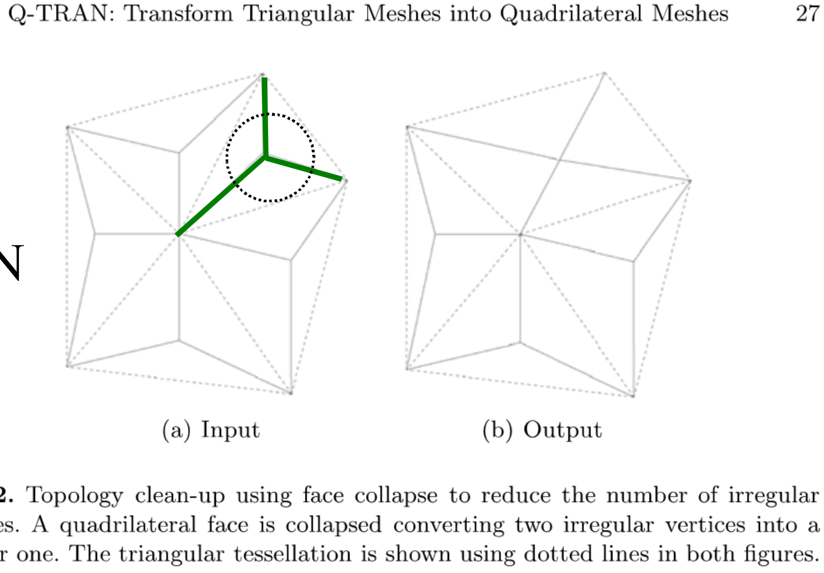


2n-gon has n-2 mono-triangles

6 -> 1
8 -> 2
10 -> 3...

adjacent mono-triangles OK
makes a red quad

Compare to Q-TRAN



Q-TRAN: Transform Triangular Meshes into Quadrilateral Meshes 27

(a) Input

(b) Output

Fig. 2. Topology clean-up using face collapse to reduce the number of irregular vertices. A quadrilateral face is collapsed converting two irregular vertices into a regular one. The triangular tessellation is shown using dotted lines in both figures.

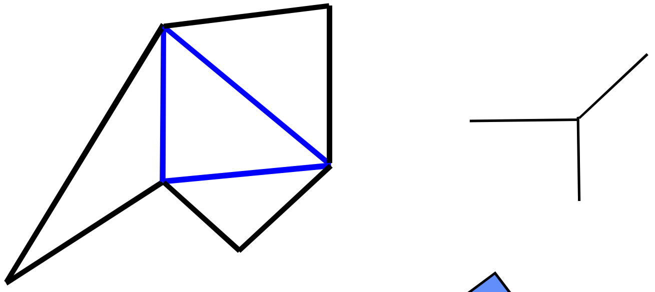
[5] M. S. Ebeida, K. Karamete, E. Mestreau, S. Dey, Q-TRAN: a new approach to transform triangular meshes into quadrilateral meshes locally, in: International Meshing Roundtable, volume 19, Sandia National Laboratories, 2010, pp. 23–34.

OK, it works. Why would you want to use it?

Locality!

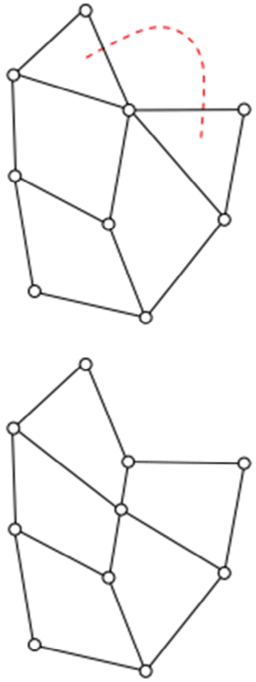
Why two-coloring vertices? Why not matching triangles?

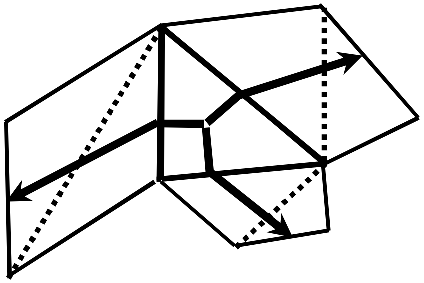
must make a change



Impossible topology
for pairing or coloring

Pairing

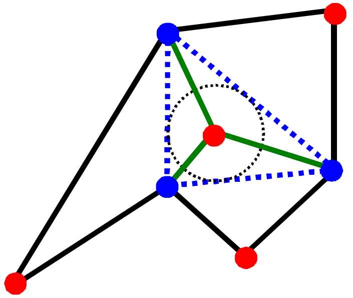




global refine chords
steering heuristics

Blossom-quad
local refinements

Coloring



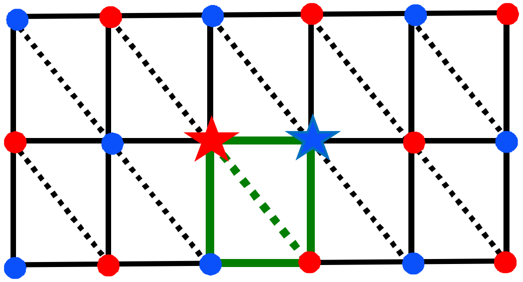
local template
deterministic, provable quality

good (best?) triangle pairing algorithm

[21] J.-F. Remacle, J. Lambrechts, B. Seny, E. Marchandise, A. Johnen, C. Geuzainet, Blossom-quad: A non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm, International Journal for Numerical Methods in Engineering 89 (2012) 1102–1119.

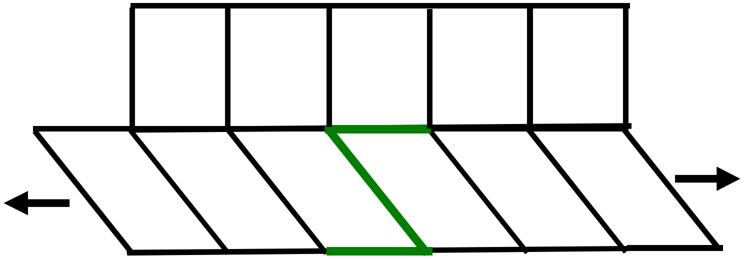
Why two-coloring vertices?
Why not matching triangles?

want a change



swap diagonal

Pairing

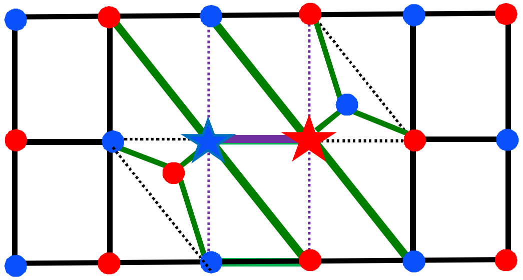


cascade of swaps

- **local changes have global scope**

  alternative is local refinements

Coloring



star neighborhood of starred vertices

+ **local changes have local scope**

Why two-coloring vertices?
Why not matching triangles?

Matching triangles	Two-coloring vertices
+ match for quad quality	– colors don’t measure quality
– slow	+ fast
- global matching alg	+ local coloring alg
- quadratic runtime	+ near linear runtime
– rare isolated tri (unmatched)	– rare isolated tri (monchromatic)
tri 1->3 quad refine	tri 1->3 tri refine
+ fixed vertices	- adds vertices
- global propagation	+ no propagation
+ alternative local refine	
(complicated, several rules)	
– local pair swap global cascade	+ local color flip local change
– global difficulties	+ local difficulties

“-” means a negative feature
“+” means a positive feature

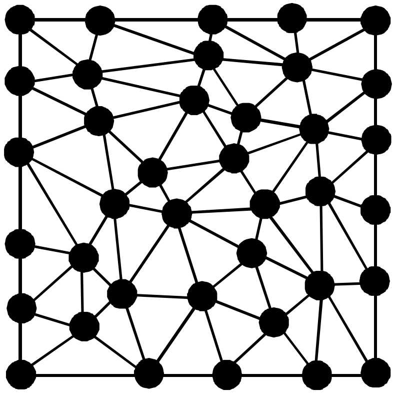
Back to the algorithm

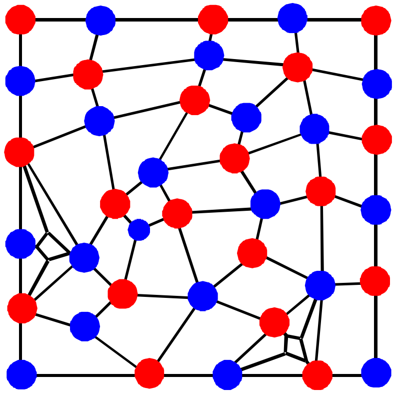
- **Generate (or given) well-spaced points**
- Delaunay triangulate
- Color points red or blue
  - intersperse colors
- Discard red-red and blue-blue edges
- Quads mostly
  - good quality, some large angles
- 6, 8, 10 sided polygons sometimes
  - constrained incircle refinement
  - median template for reflex quads
- All quads with provable quality

- **Coloring and positions improve quality in practice**

We can take any Delaunay Refinement triangulation and two-color vertices arbitrarily.

We can do better!





Sandia National Laboratories

Sphere packing, better control than Delaunay Refinement

- **Delaunay Refinement**
  - **build quality, packing results**
  - **If triangle has bad quality**
    - **Then add a point**
  - **On termination, we have a sphere packing**
- **MPS**
  - **build packing, quality results**
  - **If packing is not maximal**
    - **Then add a point**
  - **On termination, the Delaunay triangulation will have good quality.**

- **Equivalent in *theory***

bad quality = empty sphere is large (non-maximal) compared to edge length (empty-disk)

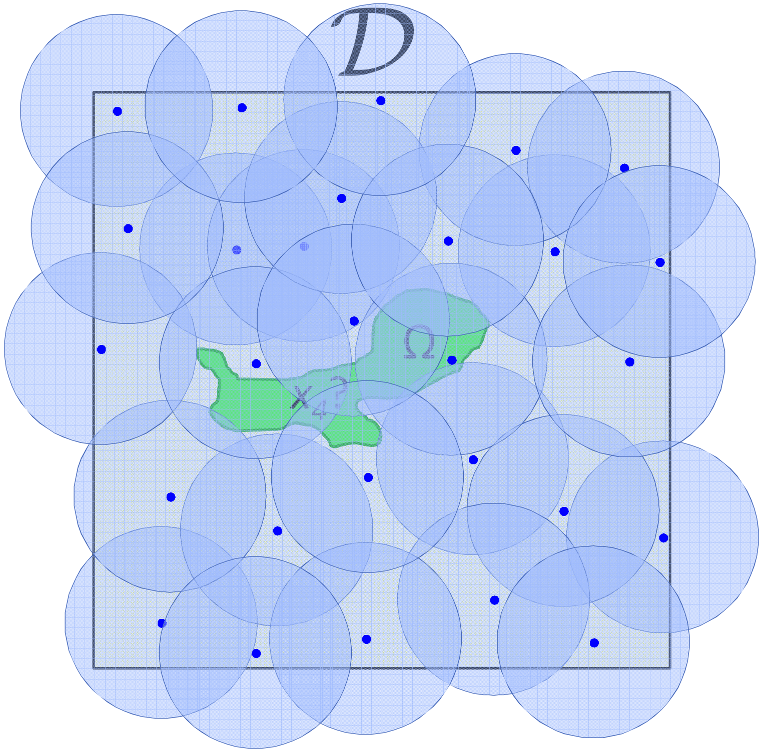
Empty disk: $\forall x_i, x_j \in X, x_i \neq x_j : ||x_i - x_j|| \geq r$

Maximal: $\forall x \in \mathcal{D}, \exists x_i \in X : ||x - x_i|| < r$

Provable angle bounds by Central Angle Theorem

- **We claim in *practice,* sphere packing has better (direct) spacing control**

- **What is MPS? Sphere packing, output of**
  - **Insert random points**
    - **With "Poisson" process, and rejection**

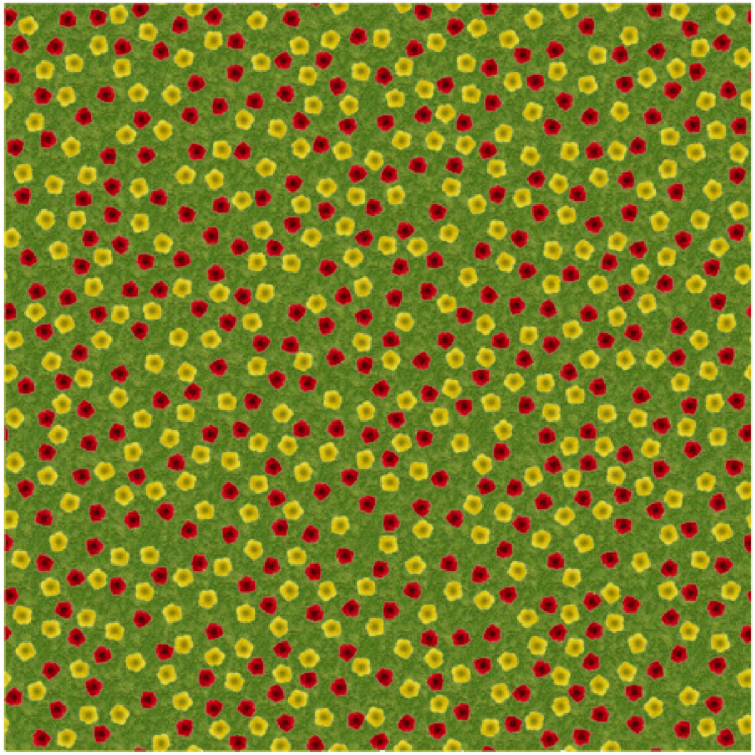


- **sphere packing algorithms are practical, our 2011-2013 work**
- **see Labelle thesis for quality theory**

Reducing the Frequency of Mono-tris
Sphere Packing with Two Colors

[17] L.-Y. Wei, Multi-class blue noise sampling, ACM Trans. Graph. 29 (2010) 79:1–79:8.

2 classes of objects

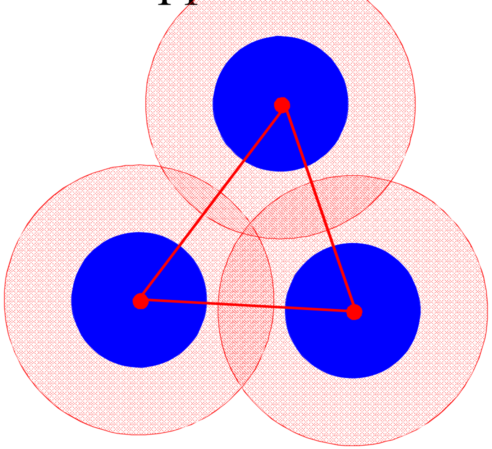


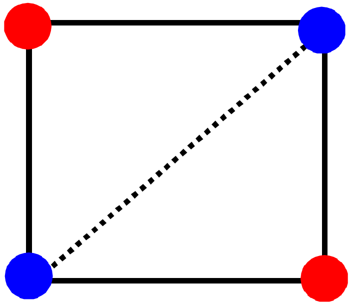
Blue-blue farther than blue-red

Motivation:

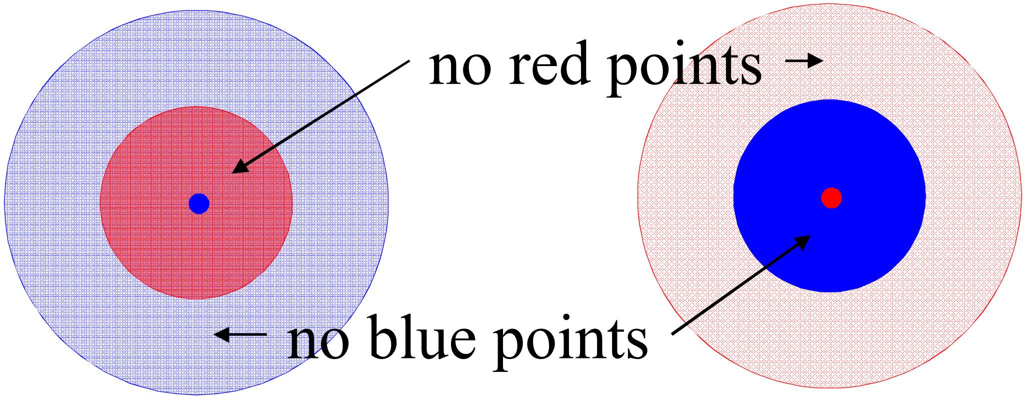
mono-triangles less likely

can fit opposite-color inside





discard long diagonals
for square-like quads

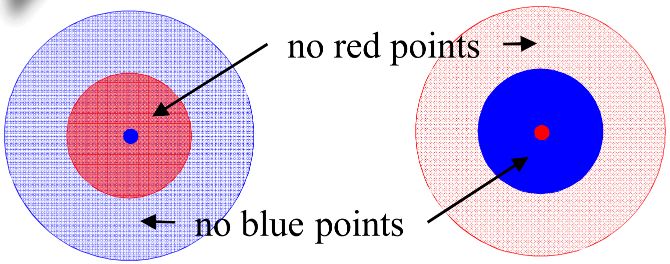


no red points →

← no blue points

Sandia National Laboratories

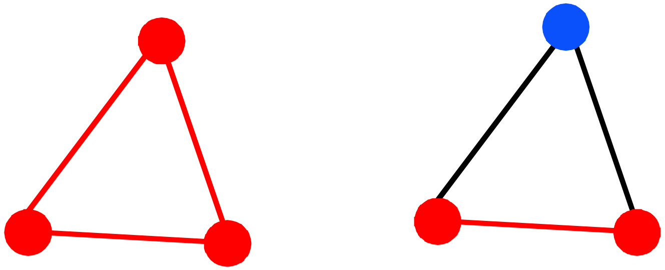
Reducing the Frequency of Mono-tris
Two-radii Plus Color Switching



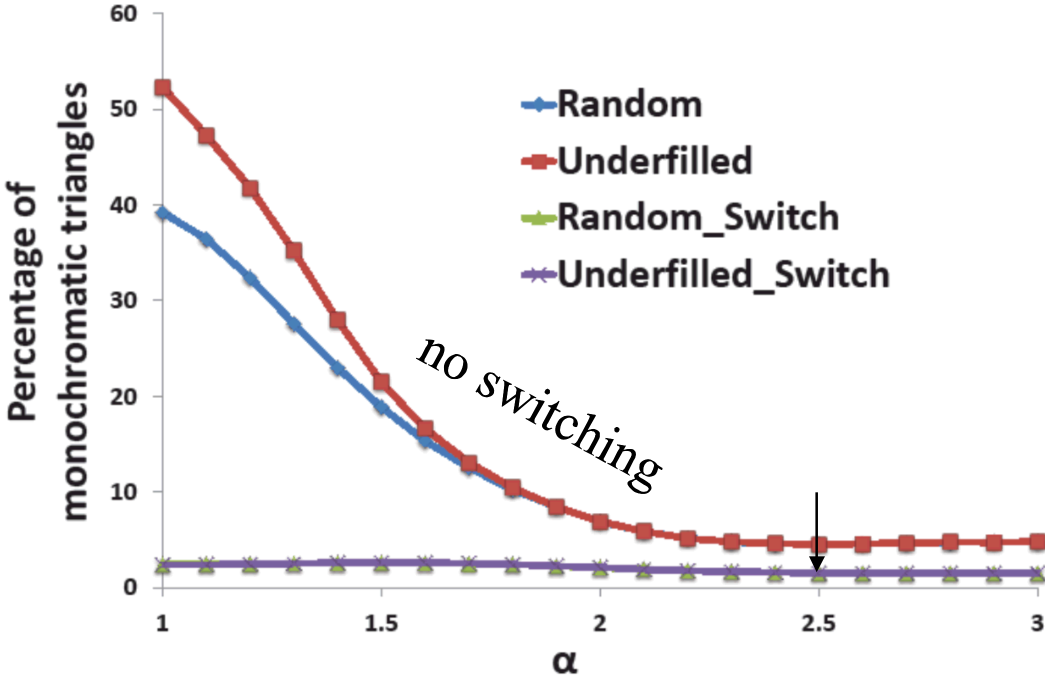
no red points

no blue points

Color switching works even better!
 if mono-tri, change color of one vertex



Ratio 2.5 works well,
> sqrt(2) square diagonal

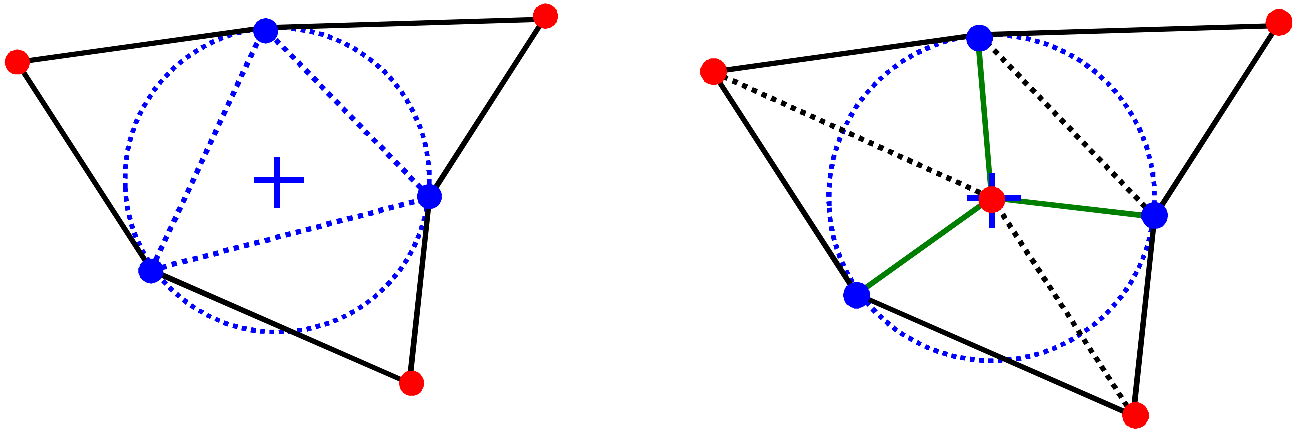


no switching

switching, stubborn 2%

Resolving Stubborn Mono-tris

Circumcircle Delaunay Refinement

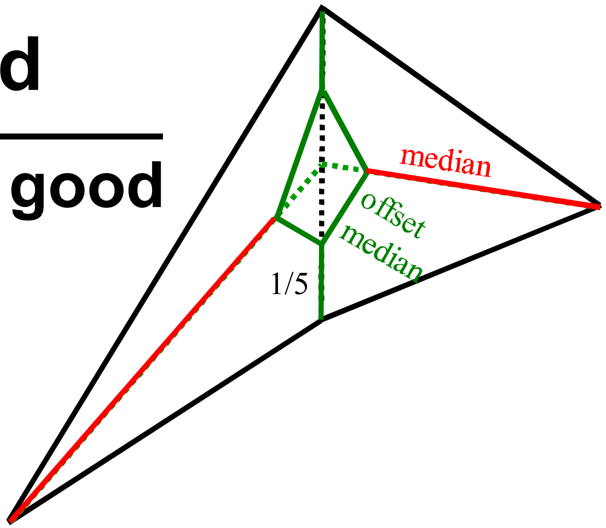


Removes the mono-tri

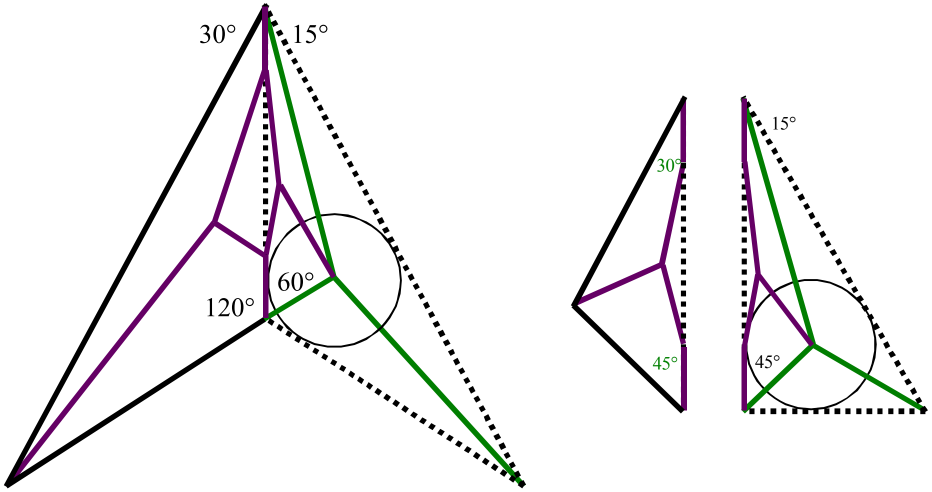
- Rare cases produce another mono-tri, requiring more refinement...

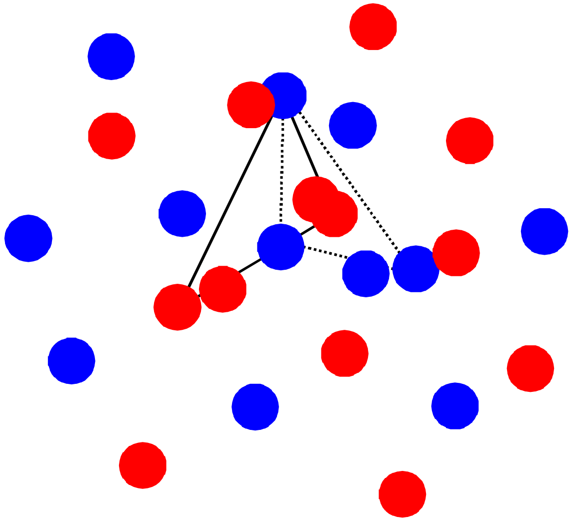
Avoiding Reflex Quad

- Median-refinement template is provably good
  - but not that good, 10-174 degrees
- Practical alternative
  - remove vertices and resample locally
  - works every time in practice



worst case:
incircle followed by reflex refine



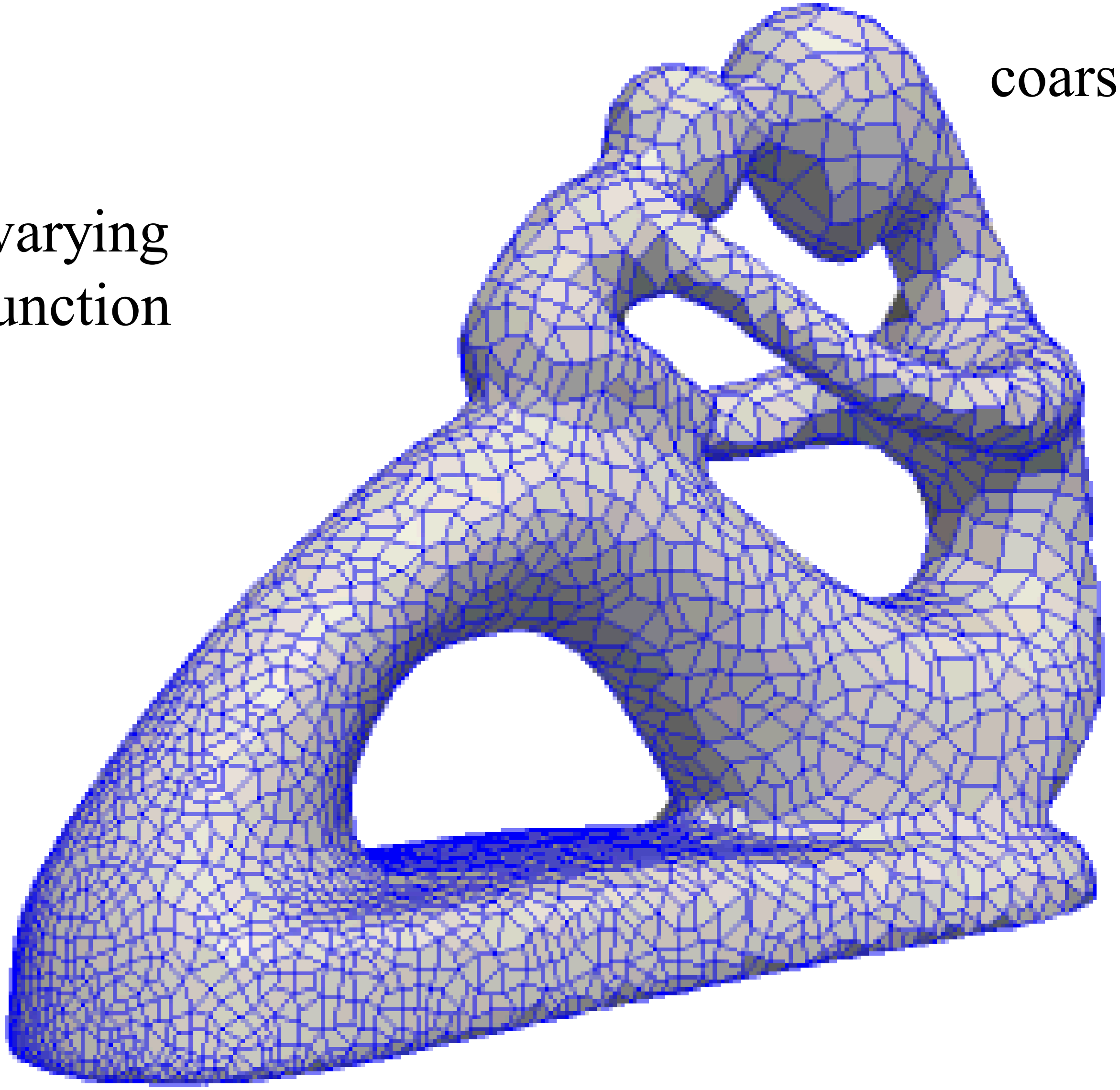


Heuristic Summary

- Reduce frequency of mono-tris
  - Two-color multiclass sampling with radii ratio 2.5
  - Color switching
- Resolve mono-tris
  - Delaunay incircle refinement
- Avoid reflex quads
  - Local resampling

- Traditional cleanup may also be applied post-process
  - we provide ok quality, convex-element starting point

Curved Surfaces
Mesh Size may Vary

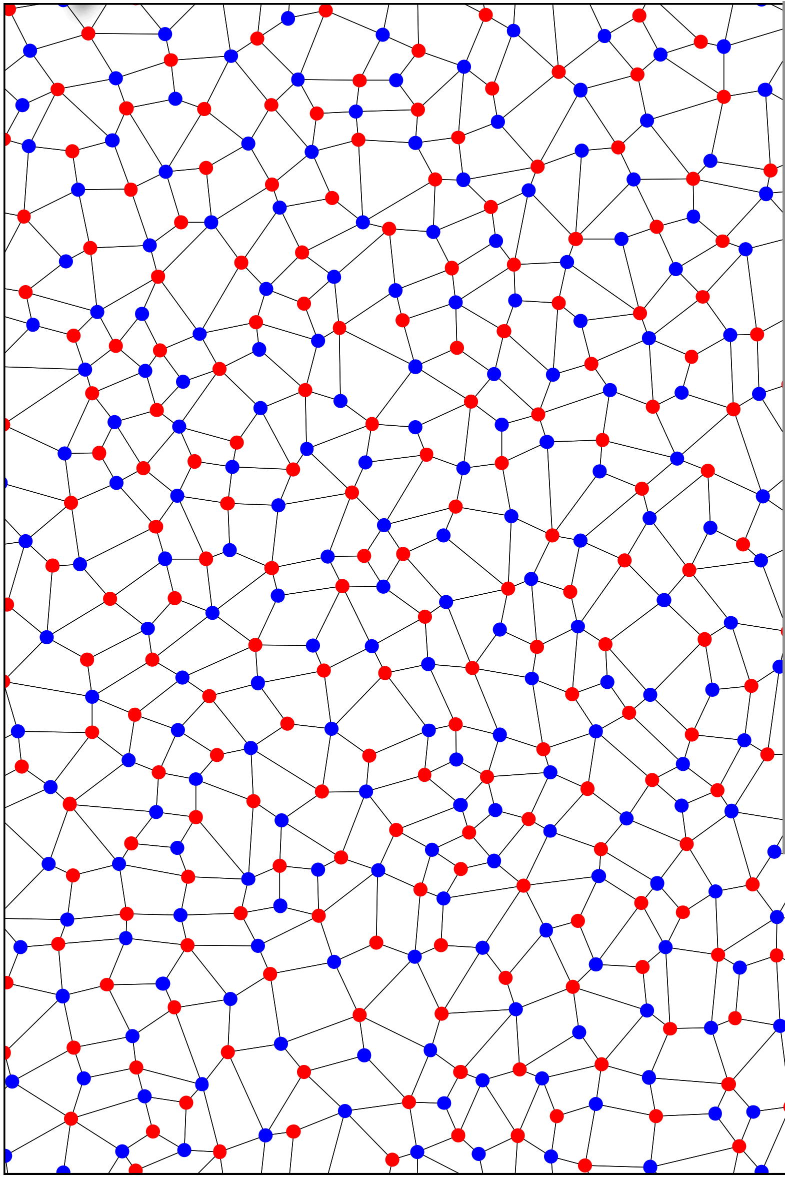


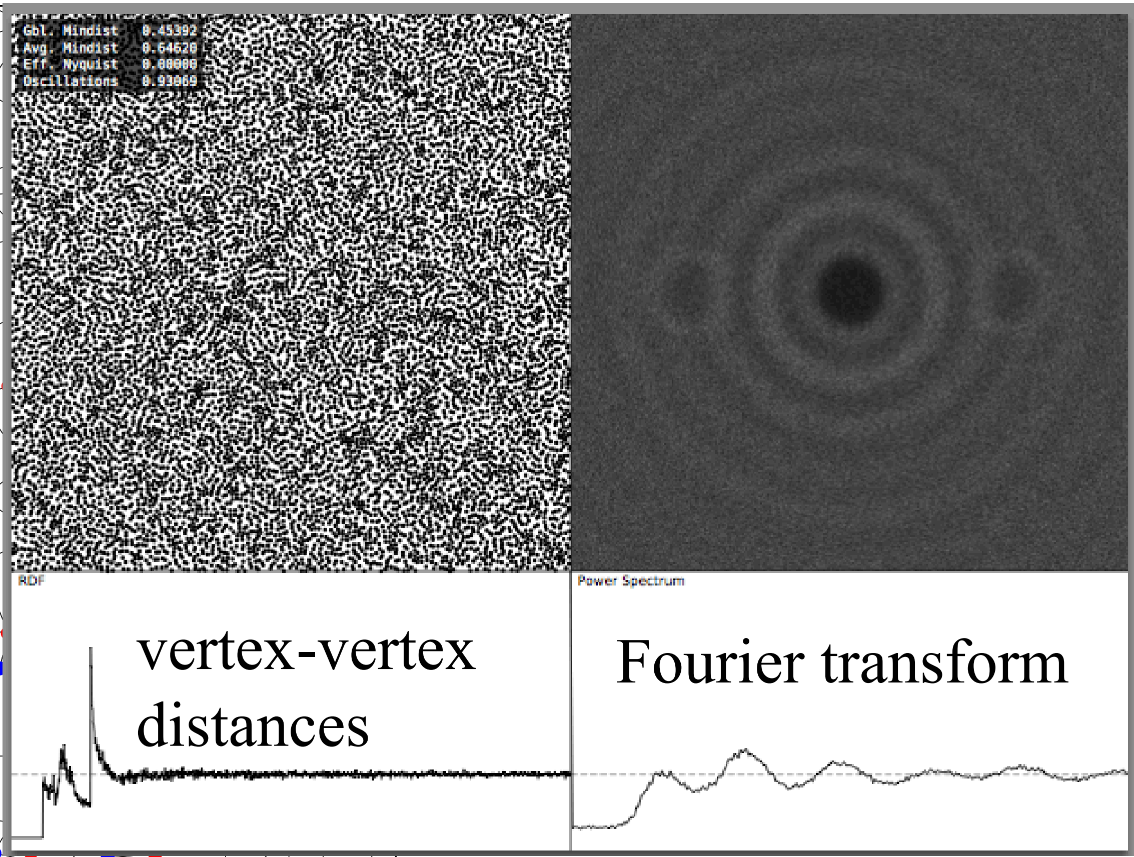
coarse

slowly varying
sizing function

fine

Fourier Spectrum Analysis





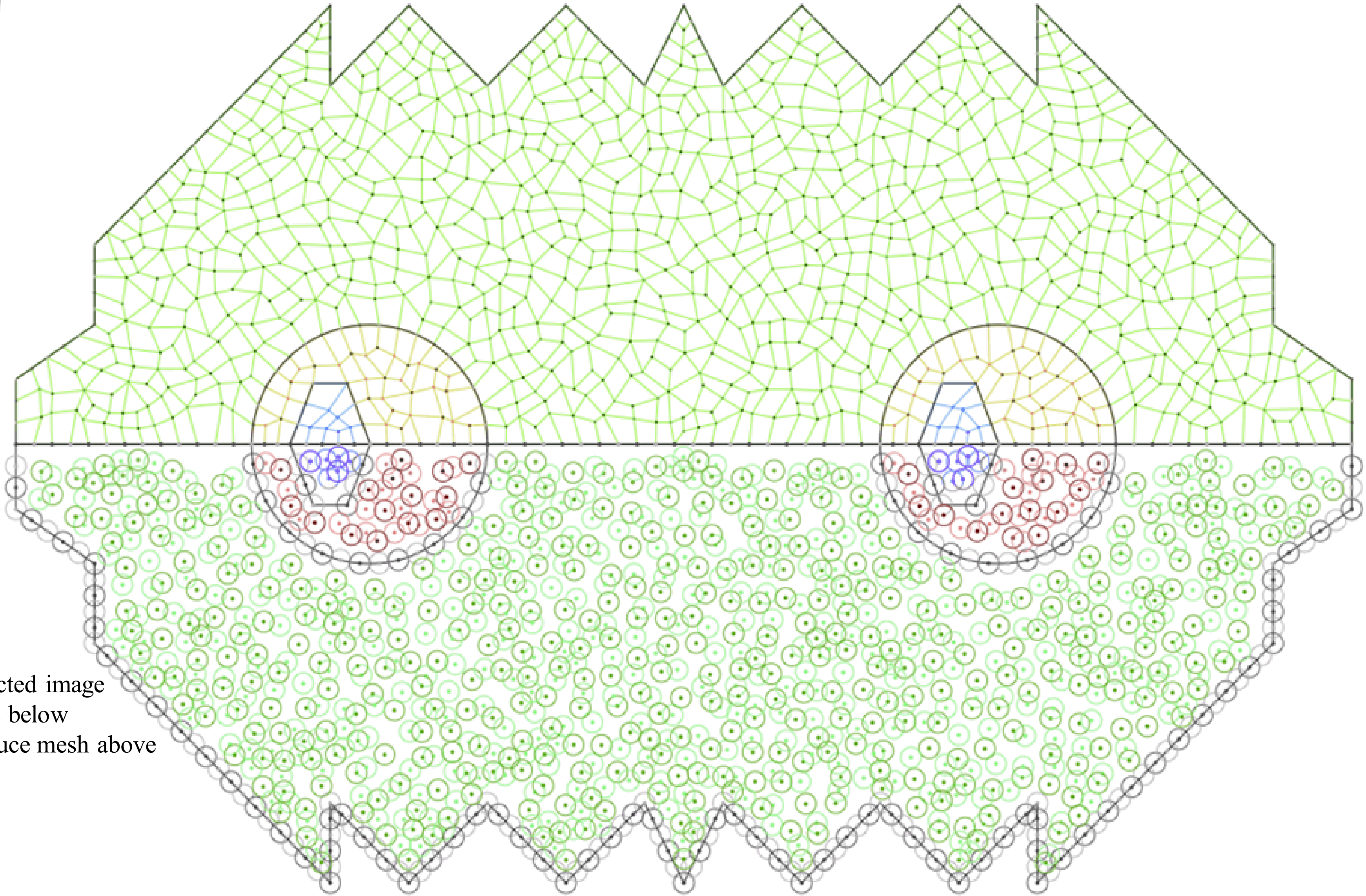
```
Gbl. Mindist    0.45392
Avg. Mindist    0.64620
Eff. Nyquist    0.88088
Oscillations    0.93969
```

vertex-vertex distances

Fourier transform

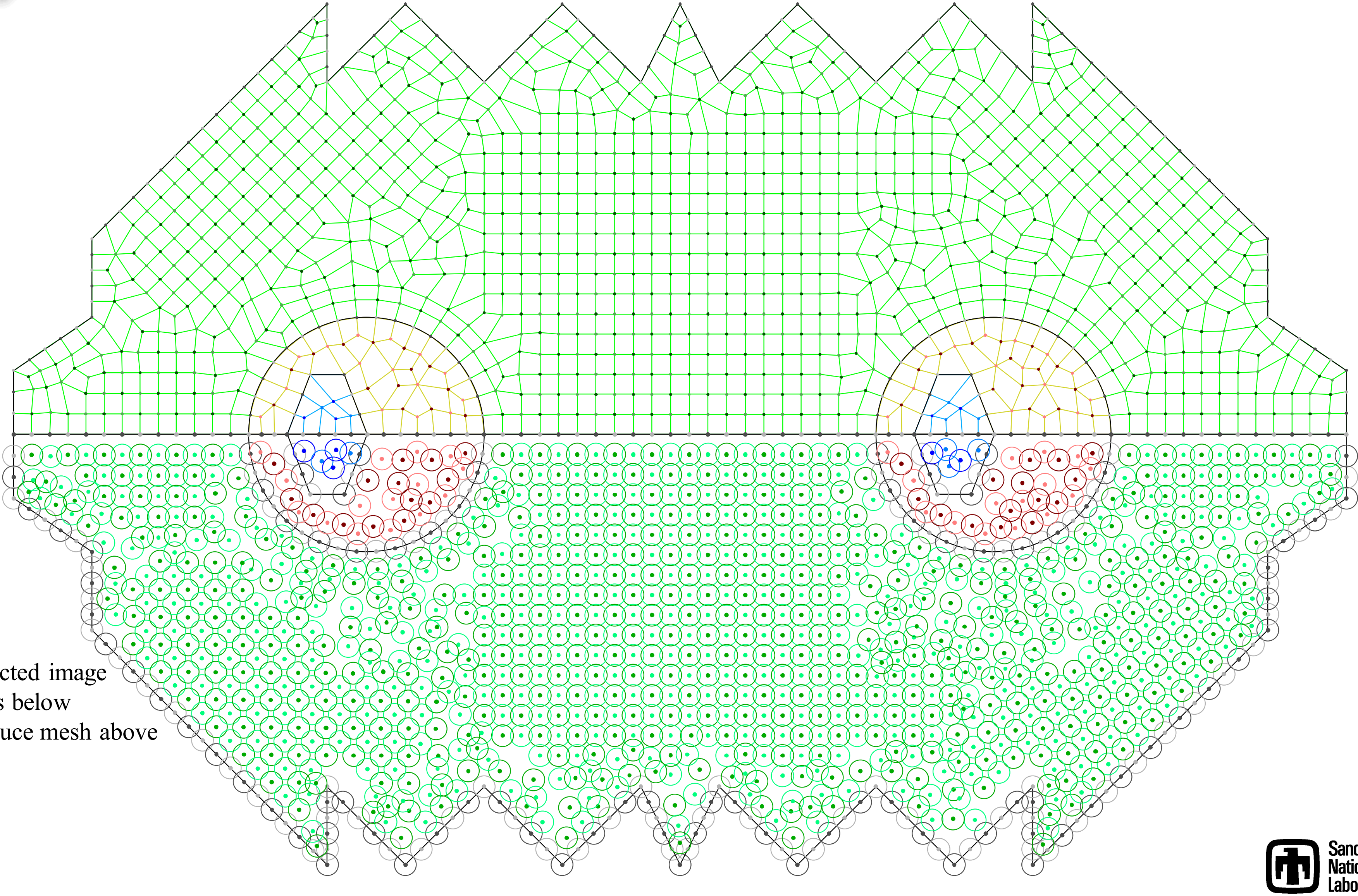
Some graphics applications rely on random positions to avoid artifacts

Random



reflected image
disks below
produce mesh above

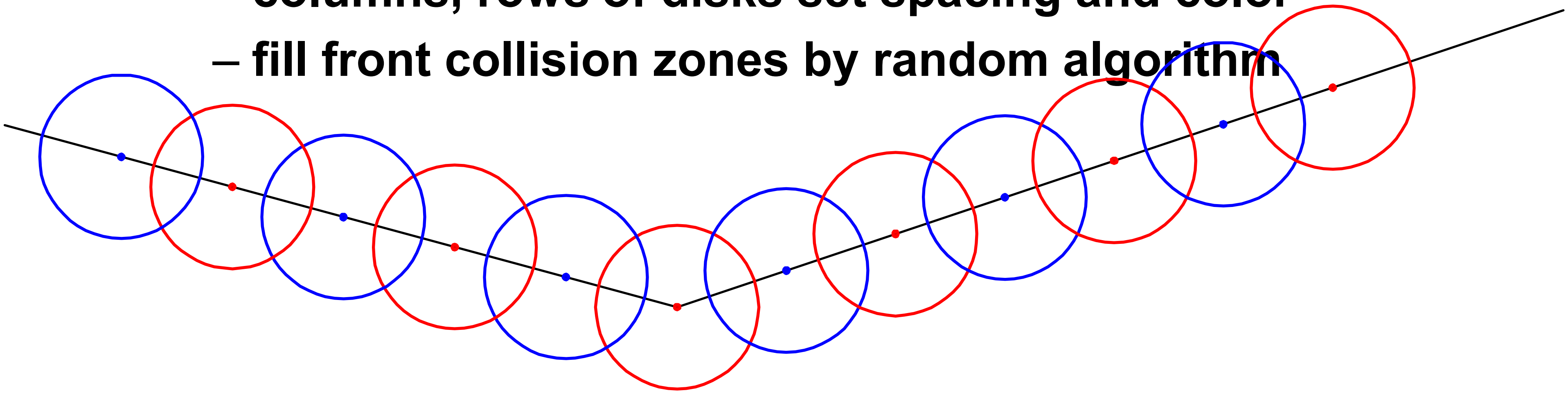
Structured



reflected image
disks below
produce mesh above

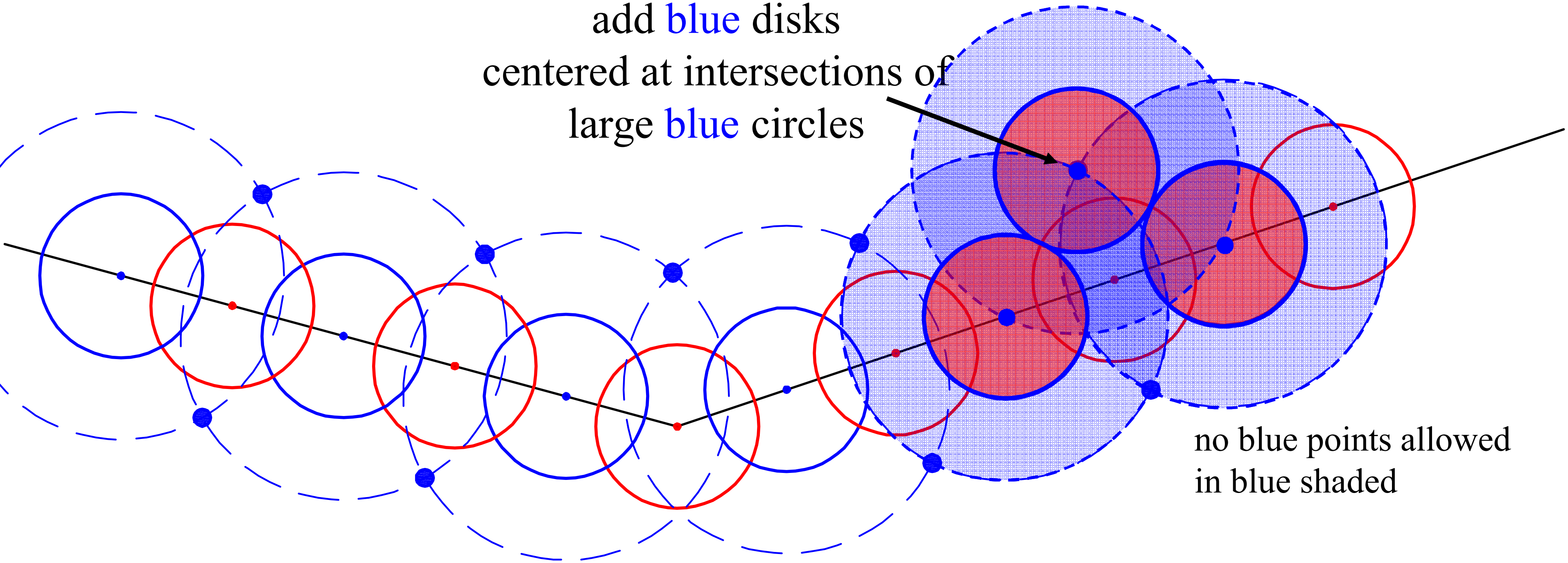
Advancing Front for Structured

- **Advance front**
  - columns, rows of disks set spacing and color
  - fill front collision zones by random algorithm



domain boundary

Advancing Front for Structured



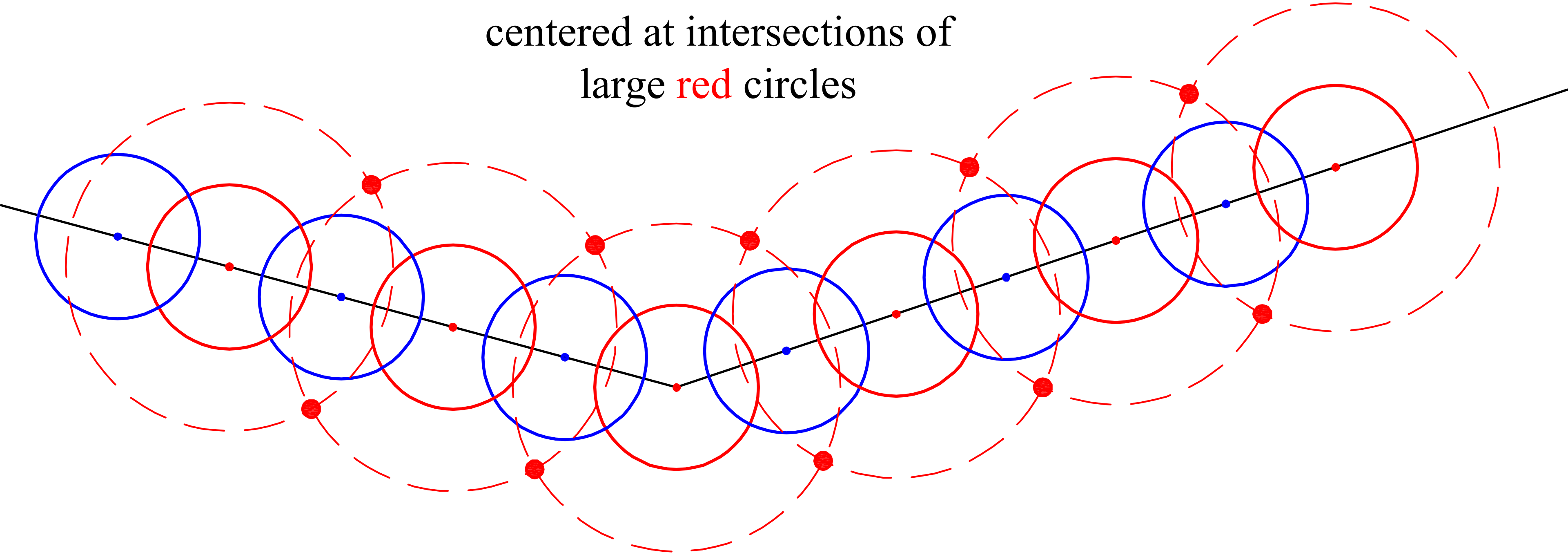
add blue disks
centered at intersections of
large blue circles

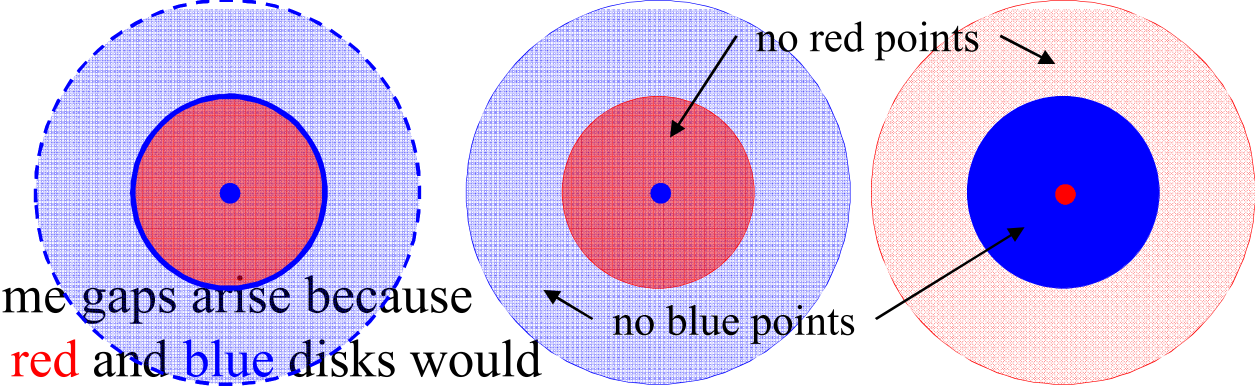
no blue points allowed
in blue shaded

This is the closest we can place blue disks
and not violate the blue-blue separation distance

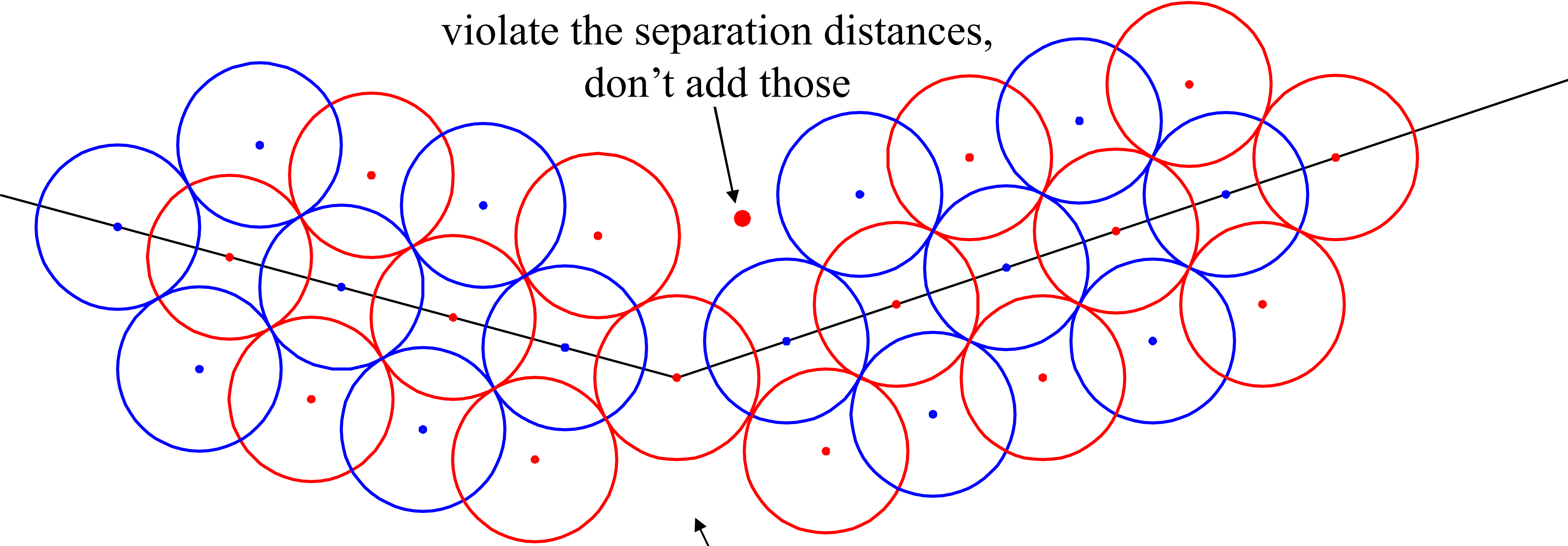
Advancing Front for Structured

add red disks
centered at intersections of
large red circles



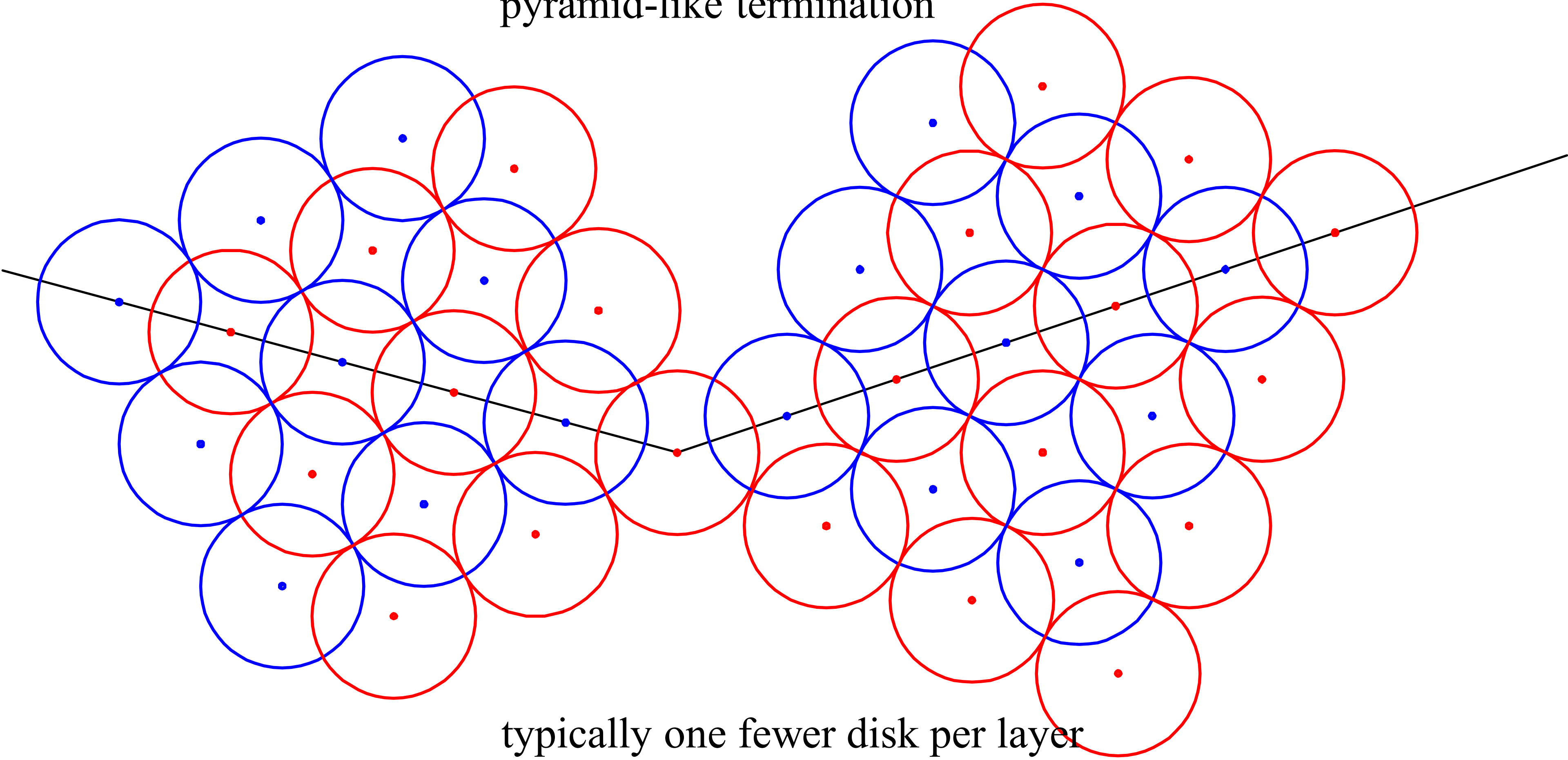


some gaps arise because
some red and blue disks would
violate the separation distances,
don’t add those

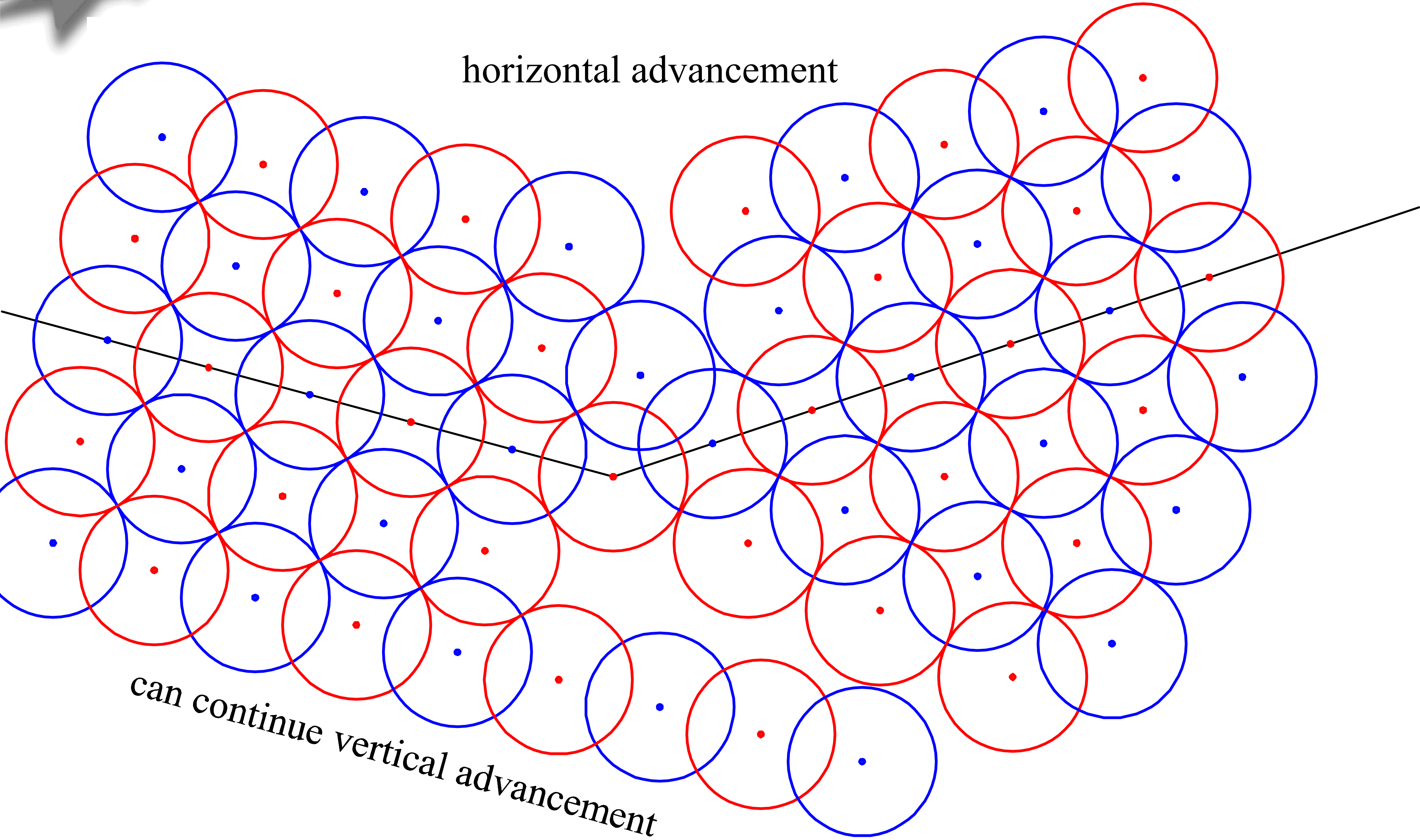


some gaps arise
from diverging fronts
e.g. large blue circles didn’t intersect

pyramid-like termination



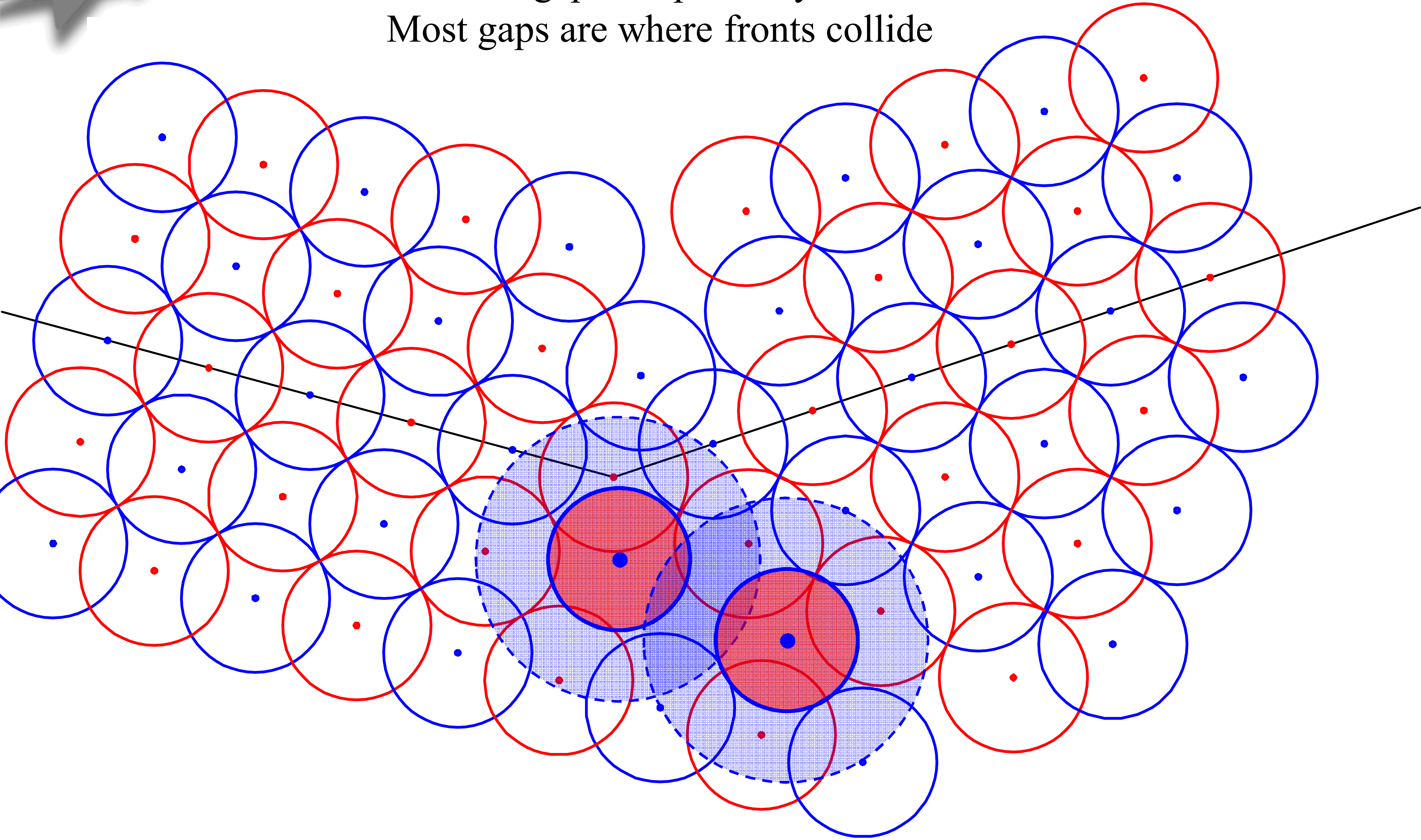
typically one fewer disk per layer
on each front



horizontal advancement

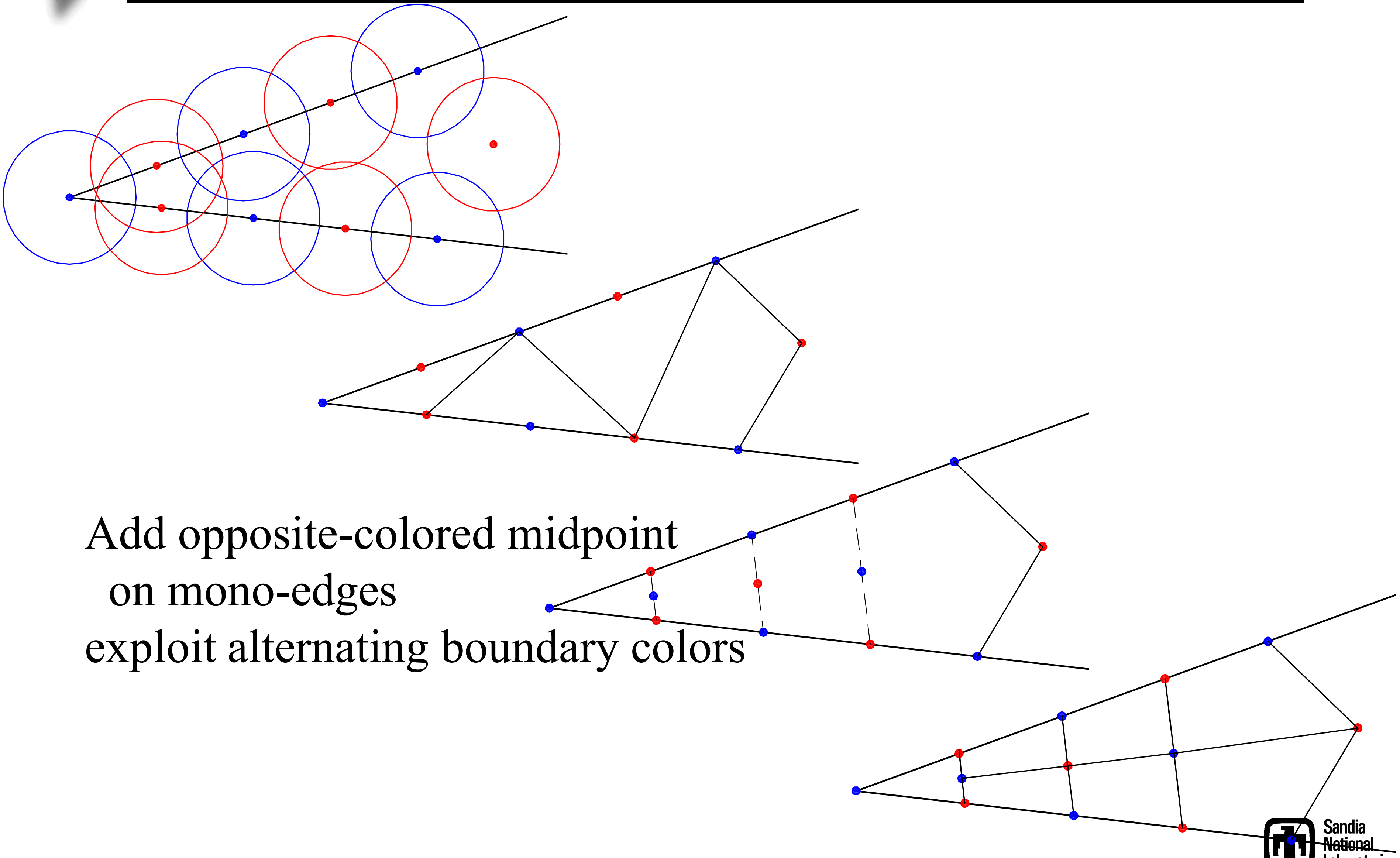
can continue vertical advancement

fill large gaps with random algorithm
small gaps are provably OK
Most gaps are where fronts collide



Sandia National Laboratories

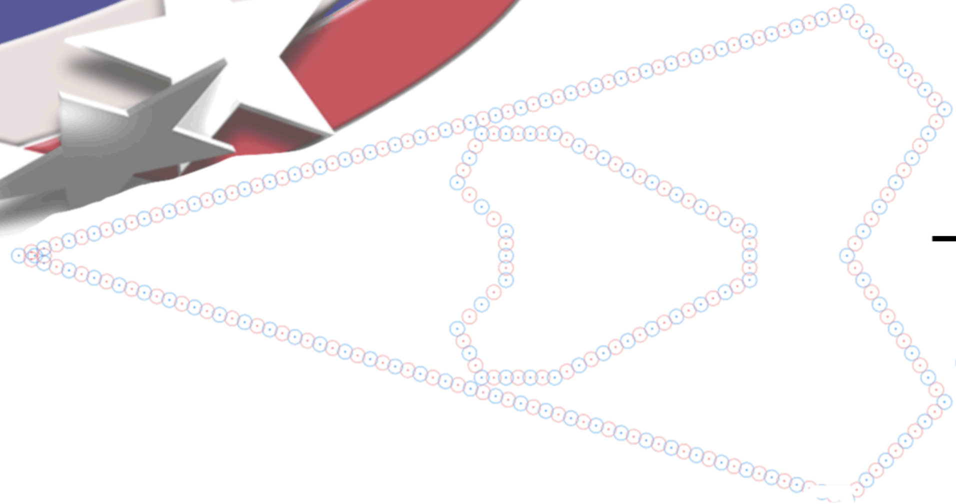
Sharp Corners

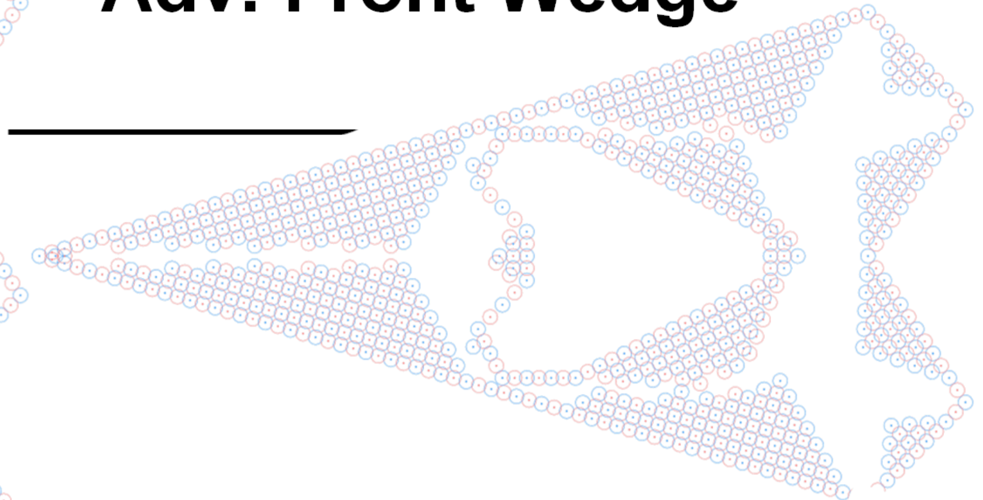


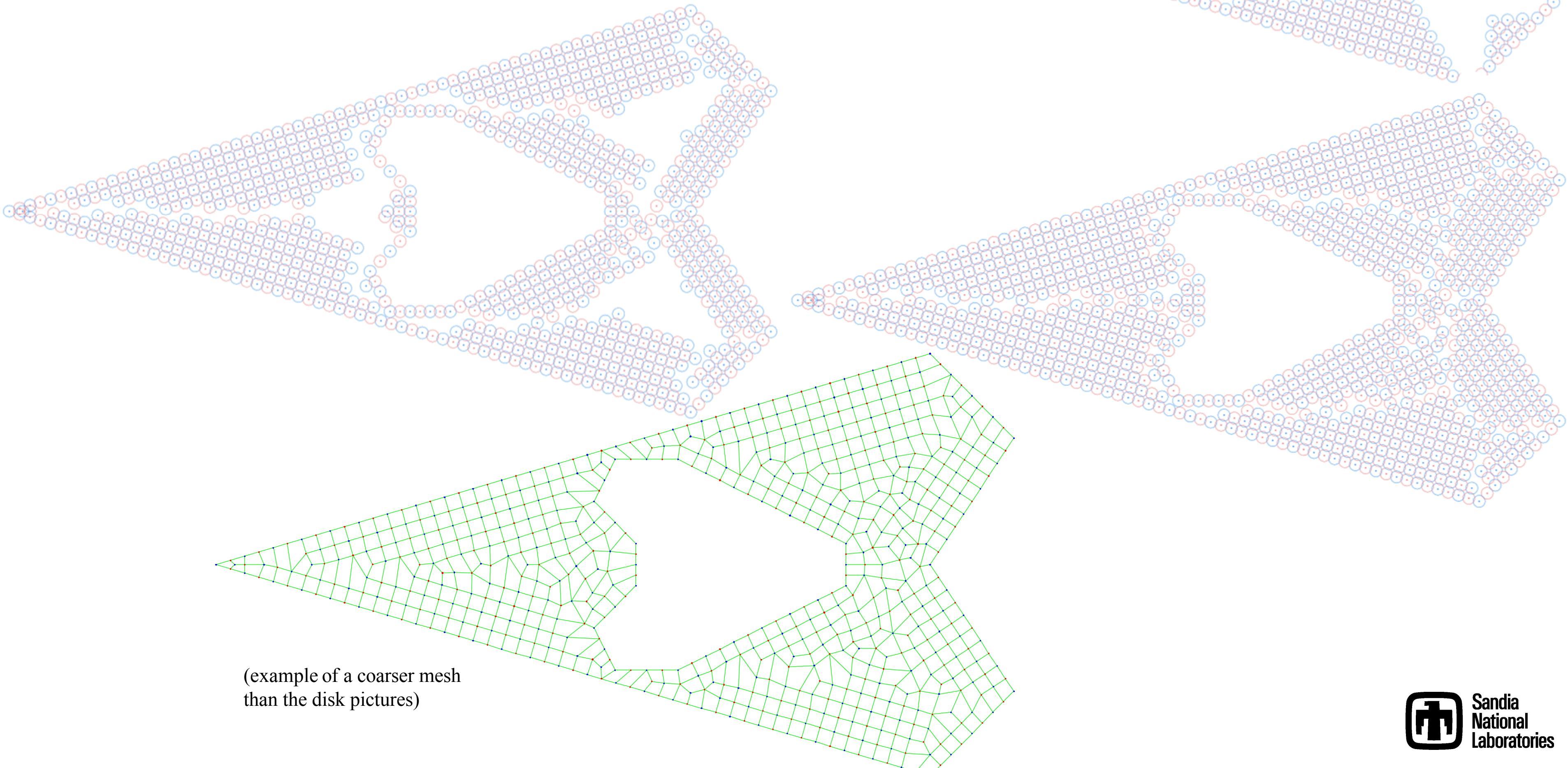
Add opposite-colored midpoint
on mono-edges

exploit alternating boundary colors

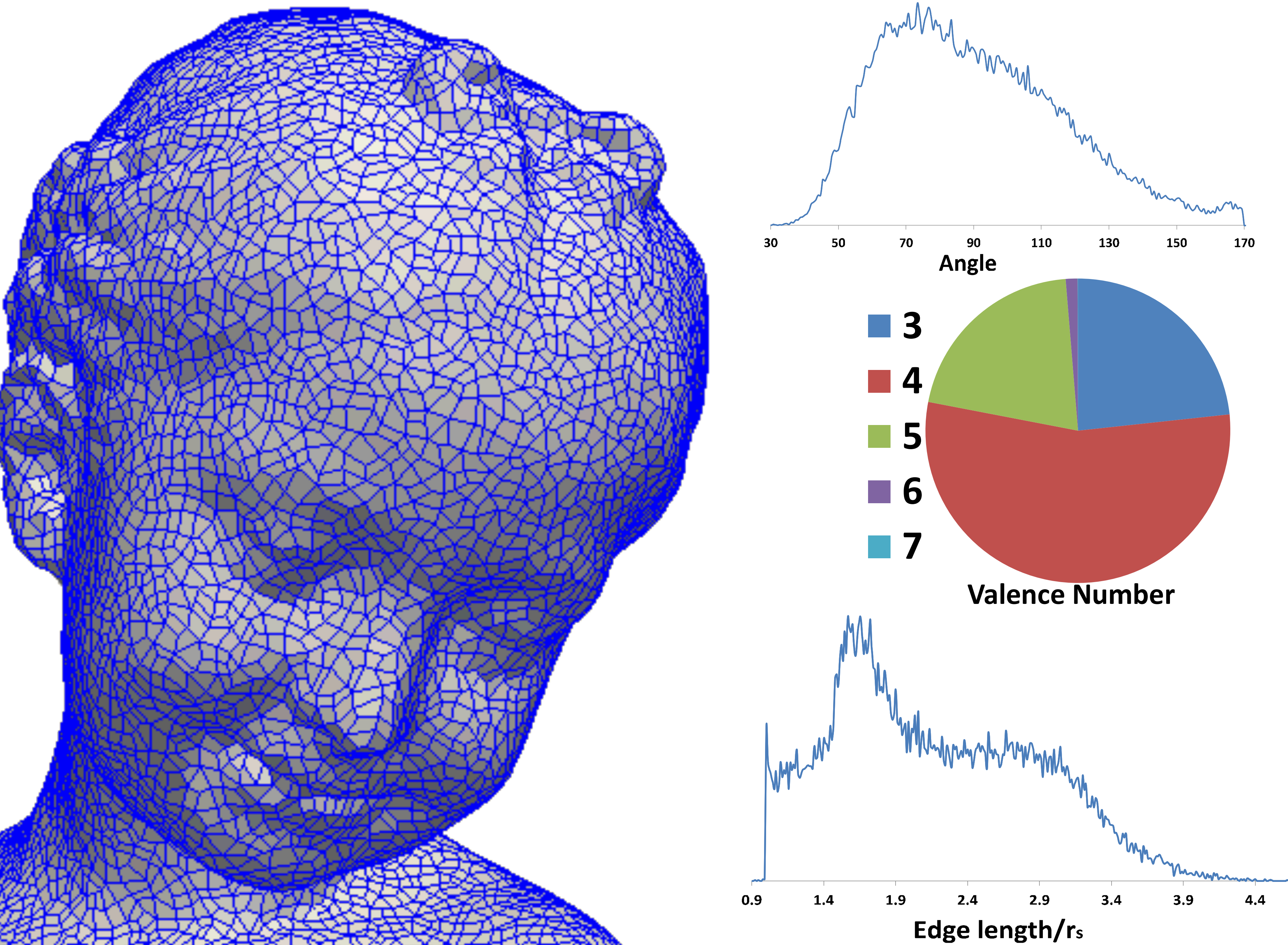
Adv. Front Wedge

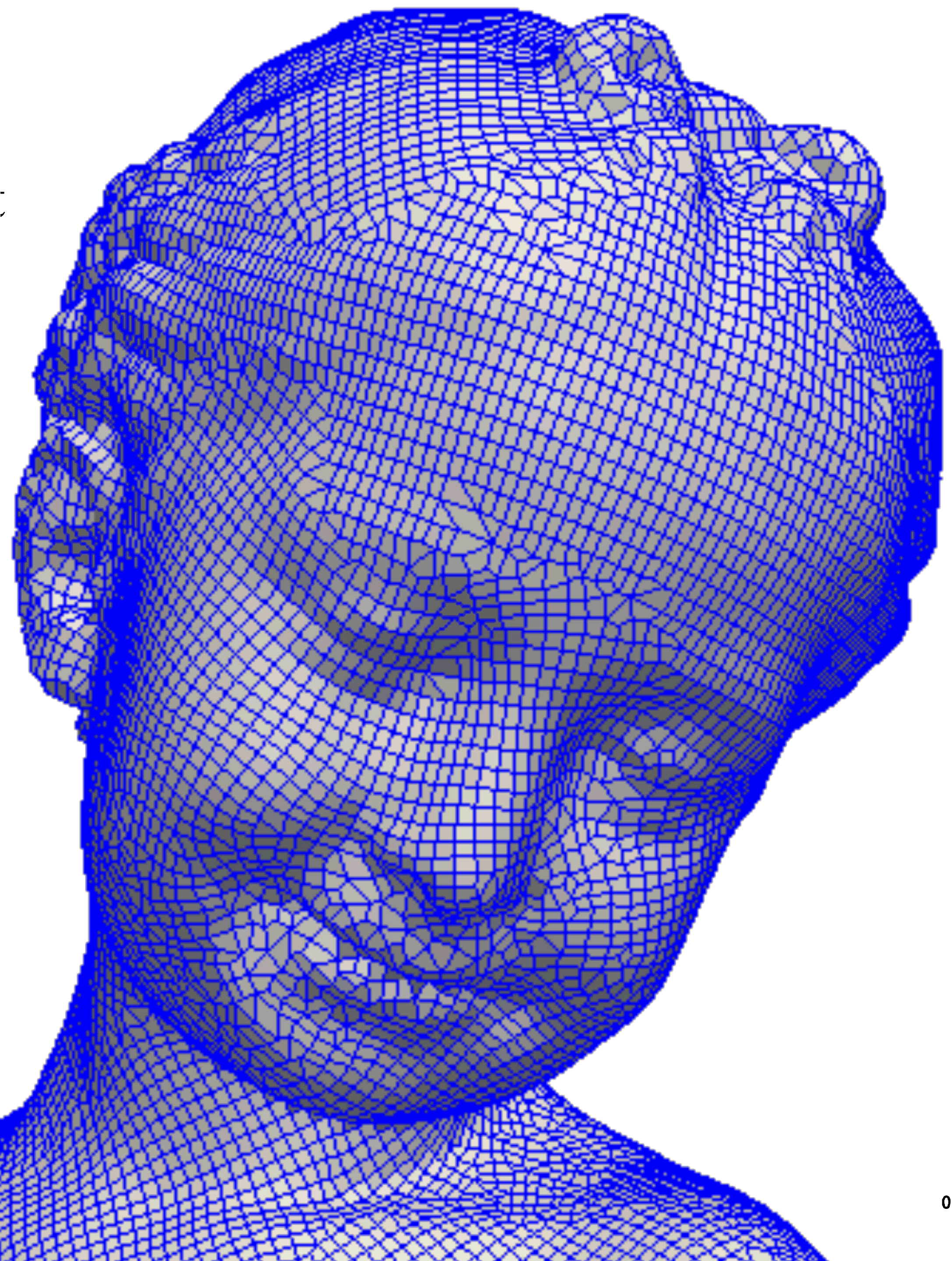


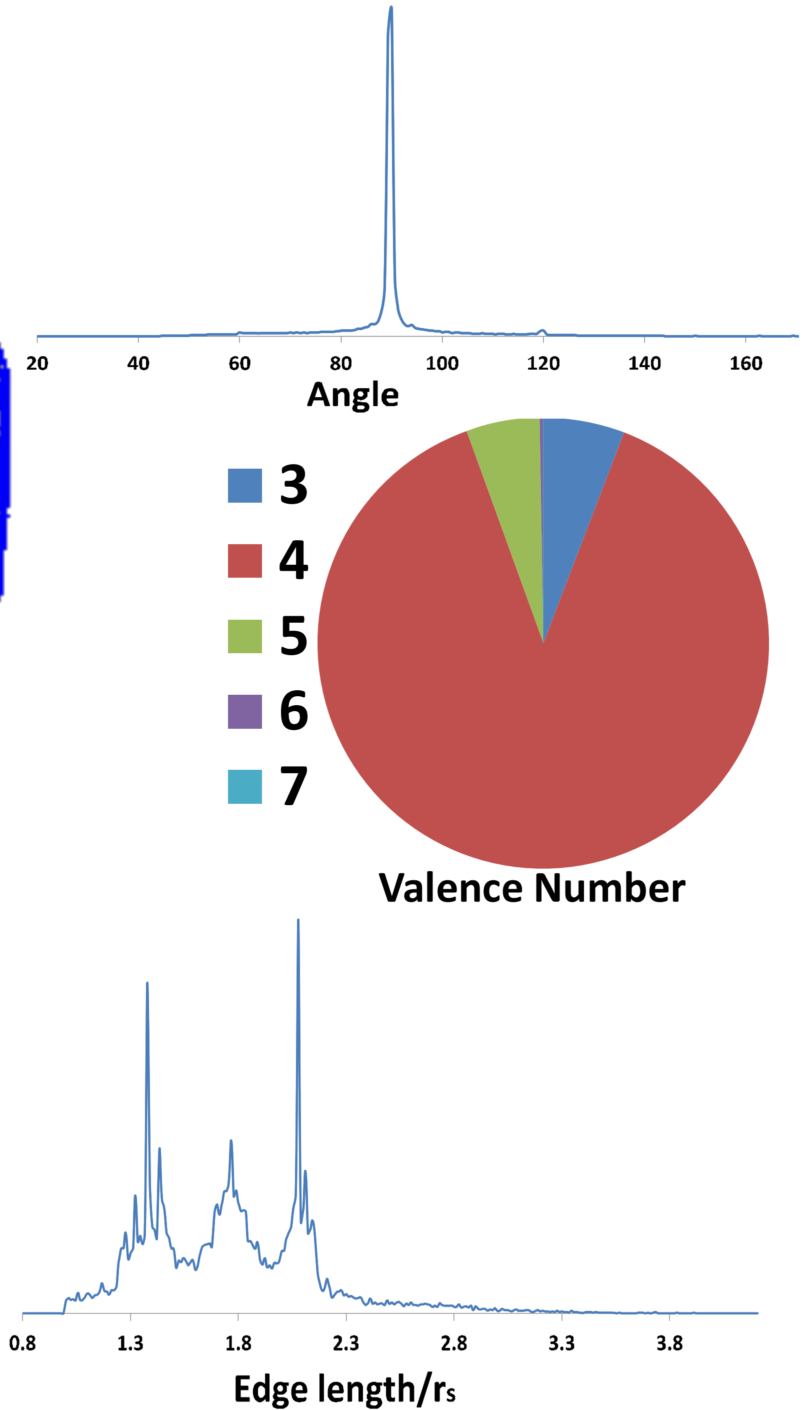




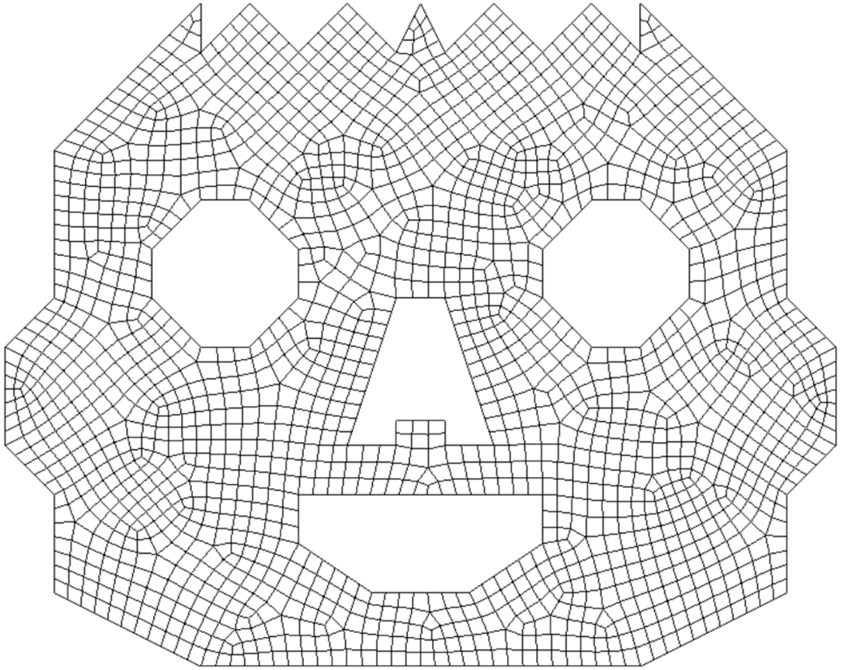
(example of a coarser mesh than the disk pictures)


30
50
70
90
110
130
150
170
Angle
🟦 3
🟥 4
🟩 5
🟪 6
🟦 7
Valence Number
0.9
1.4
1.9
2.4
2.9
3.4
3.9
4.4
Edge length/r_s

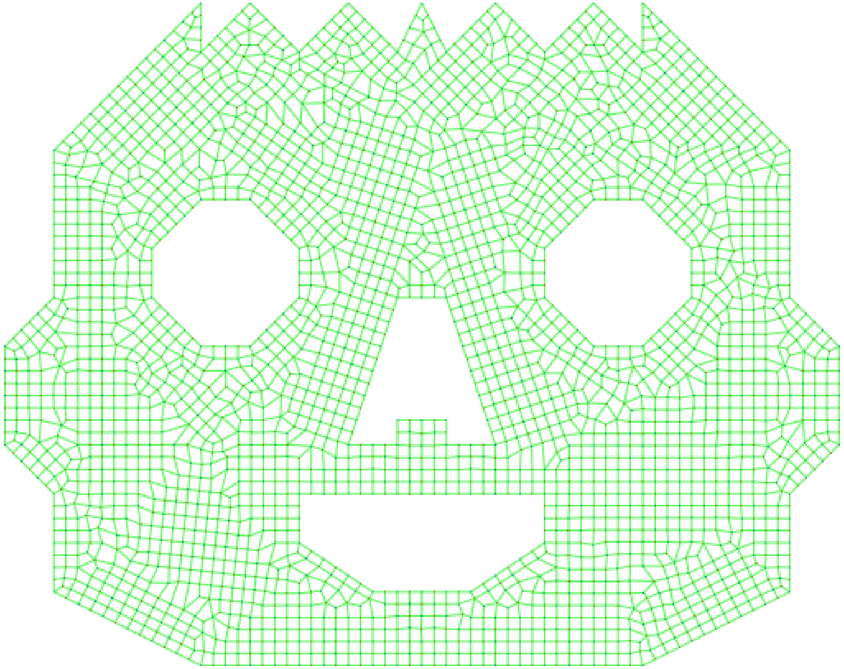



20
40
60
80
100
120
140
160
Angle
🟦 3
🟥 4
🟩 5
🟪 6
🟦 7
Valence Number
0.8
1.3
1.8
2.3
2.8
3.3
3.8
Edge length/rs

Qualitative comparison to a production code

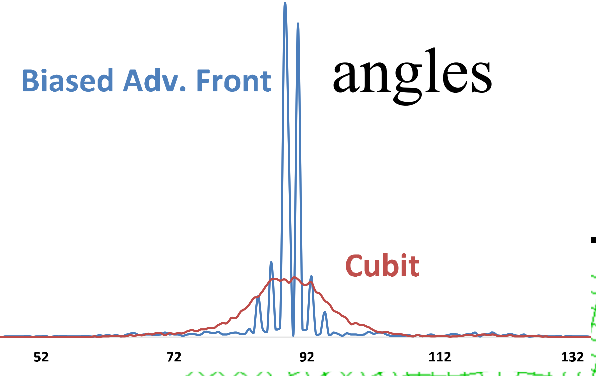


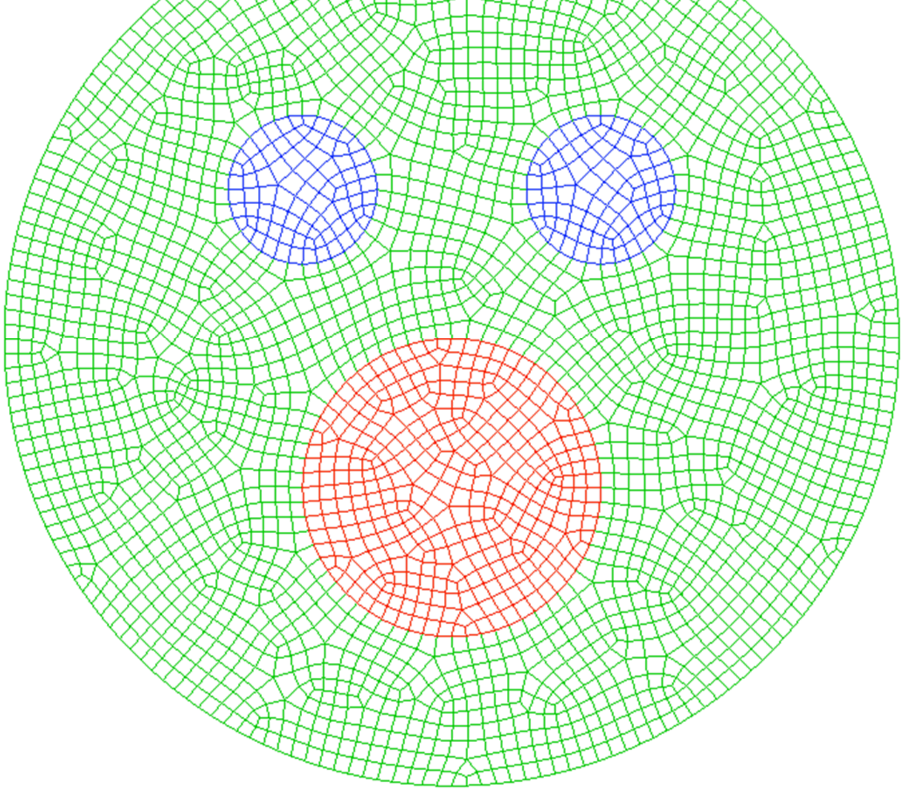
Cubit paver



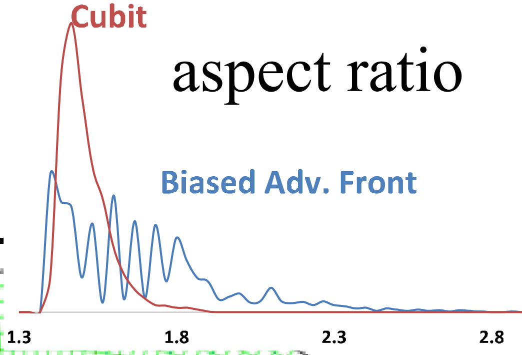
Two-color adv. front no cleanup

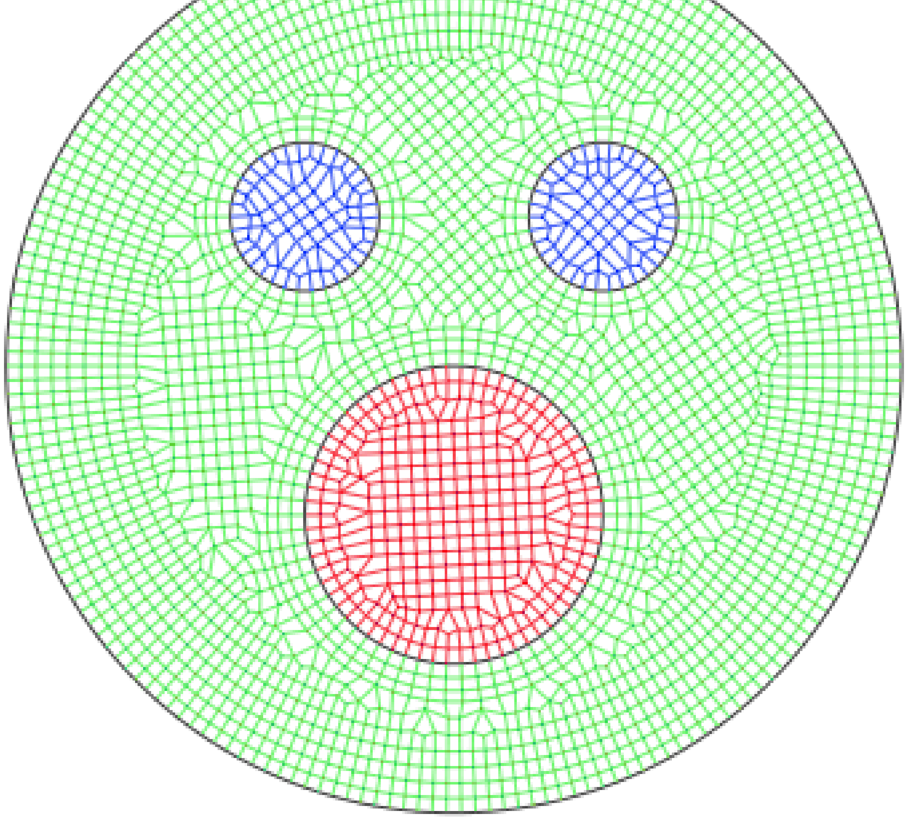
Qualitative comparison to a production code





Cubit paver



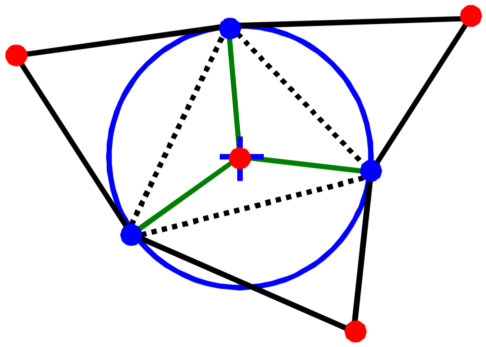


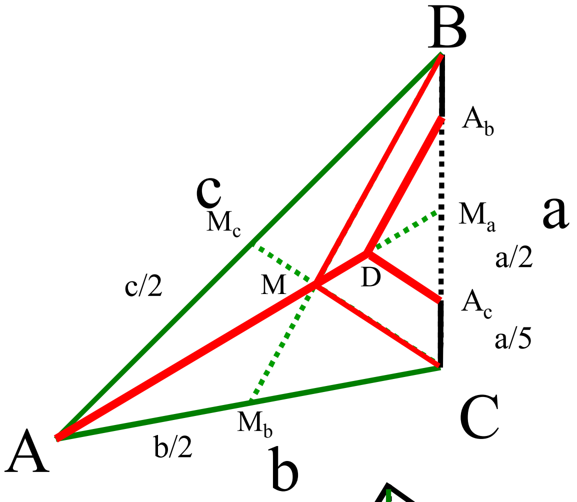
Two-color “biased” adv. front
no cleanup

Closing Thoughts

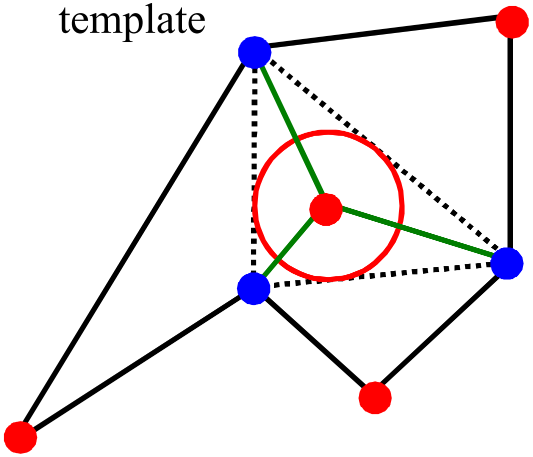
- **Three centers: Circumcenter, Incenter, Centroid**

Circumcenter for mono-tris heuristic

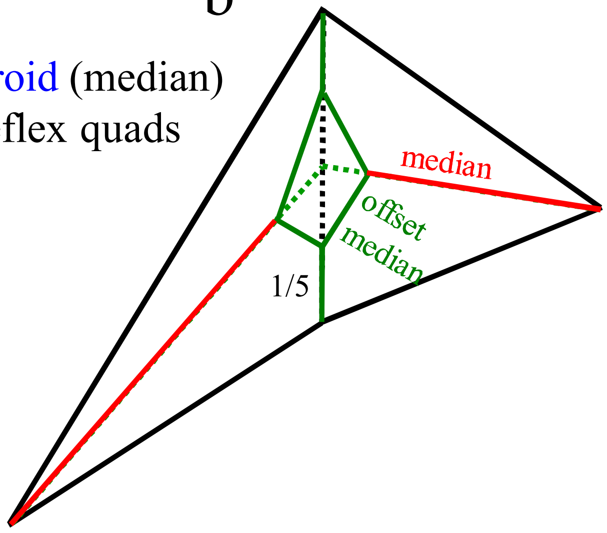




Incenter for mono-tris template



Centroid (median) for reflex quads

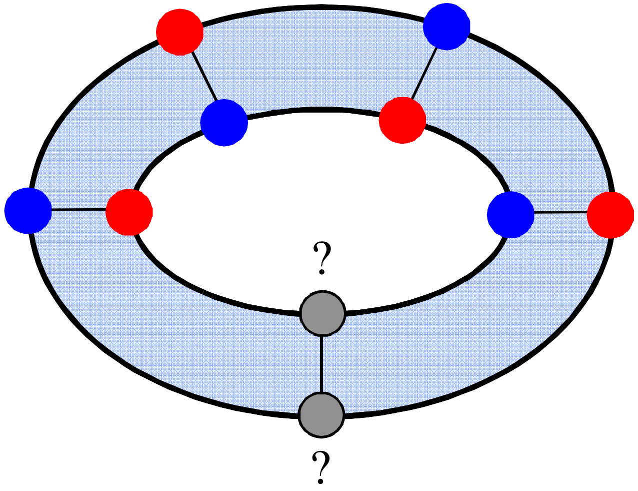


Orthocenter feeling left out

deviantArt Free Ai Vector Smiley Pack by MathieuBerenguer

Closing Thoughts

- We create two-colorable quad meshes
  - Some given quad meshes are not two-colorable



odd-cycle of edges
non-disk domain topology

Closing thoughts

– Can we create hex meshes by coloring?

no.

See graphics community literature on 3d cross-fields and connecting hex-duals

Summary

- Questions
  - Varying-size advancing front?
  - How fast can we vary the size?
    - Bounds for two-radii DT known
- Features
  - Robust
  - Simple
  - Local
  - Provable quality
  - Heuristics for good quality in practice
  - What more do you want?
- Quality is reasonable before cleanup
  - someone could build a production tool based on this
  - add traditional cleanup

Sandia National Laboratories