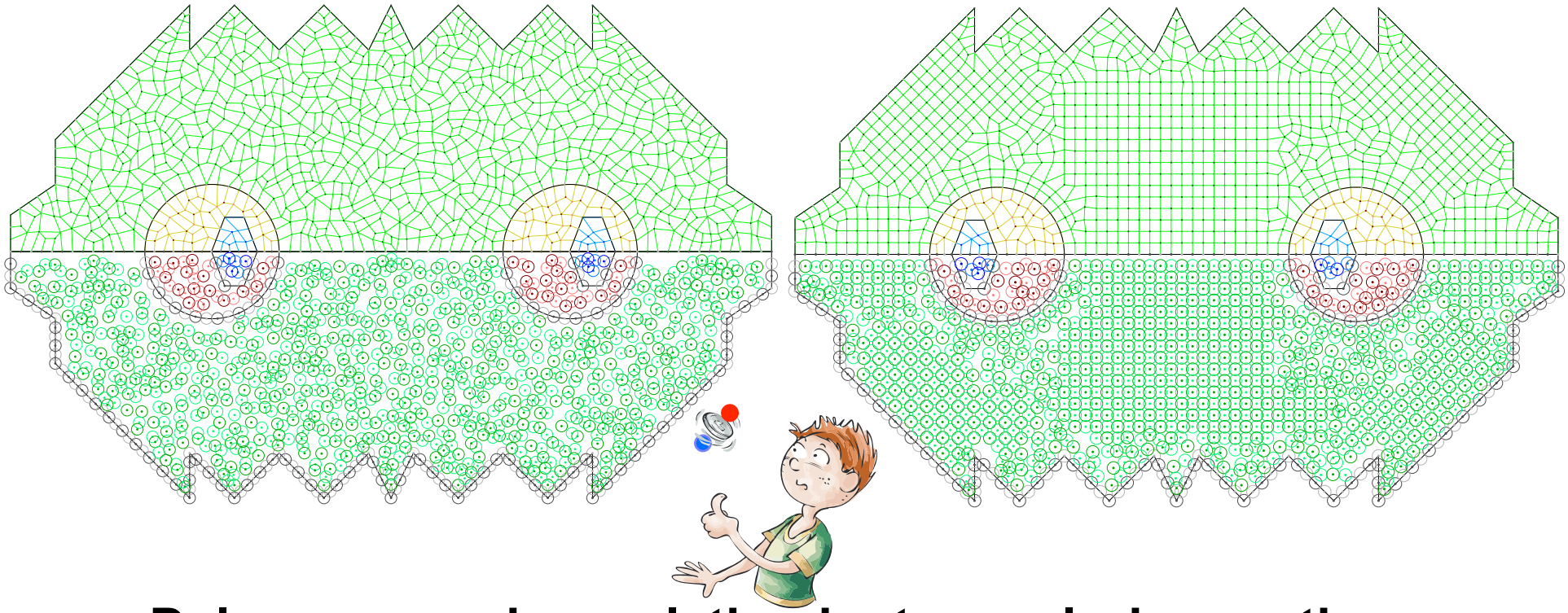


# Delaunay quadrangulation by two-coloring vertices

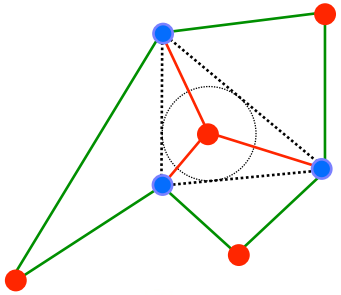
## or

## 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

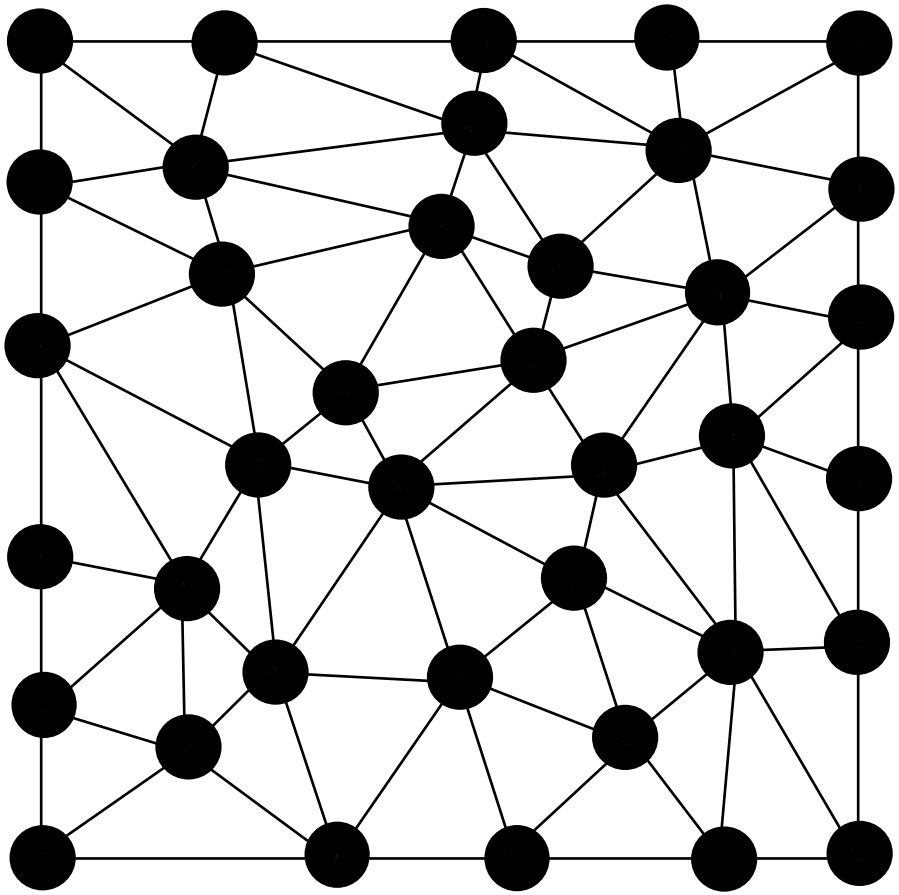
- Algorithm on one page
- Two-coloring idea
  - Contrast to Triangle Pairing
  - Even-sided polygons
  - Templates for provable quality (appendix on website)
    - Fixing by constrained incircle refinement
      - QTran
    - Constrained median refinement
- (Random Algorithm)
  - Well-spacedness properties, 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

Sandia National Laboratories

# 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

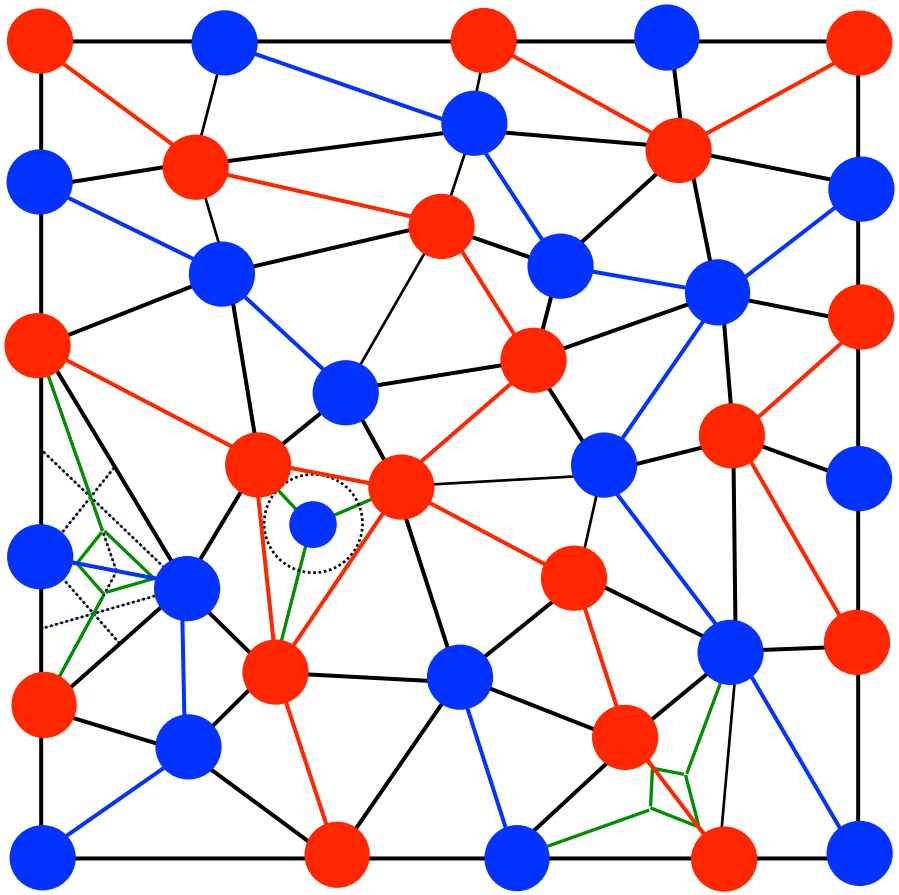


Sandia National Laboratories

# 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

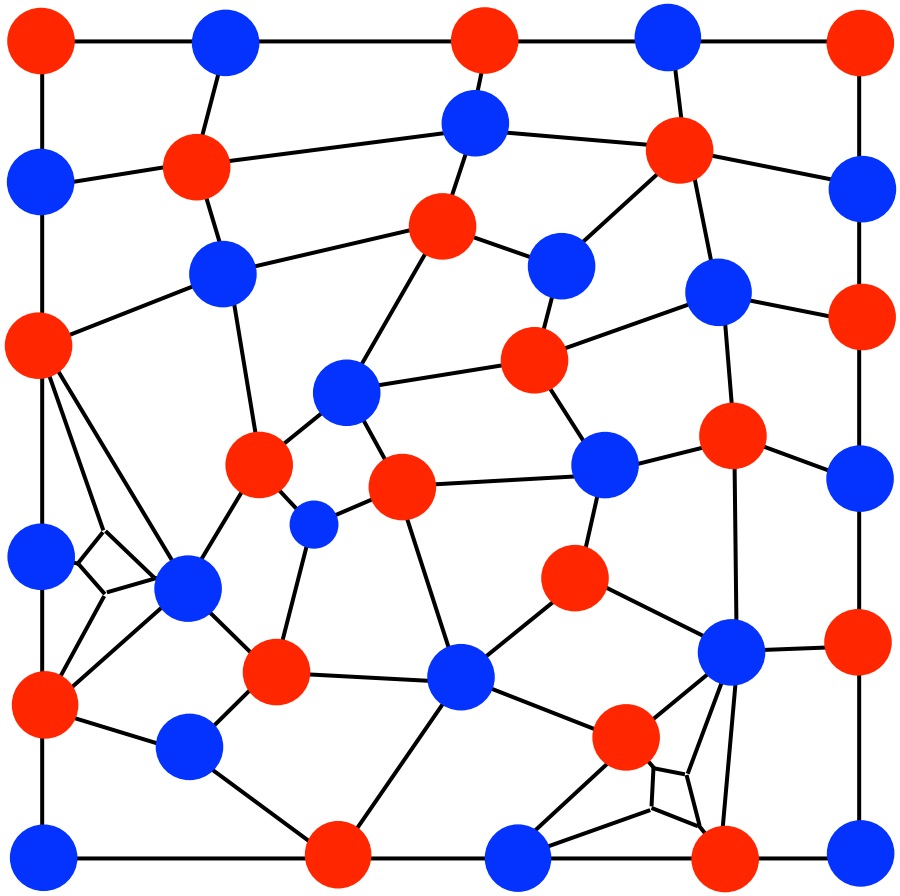


Sandia National Laboratories

# 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**

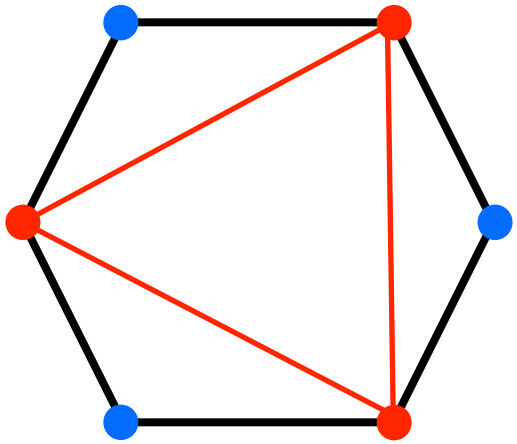
- **Coloring and position heuristics improve quality in practice**



Sandia National Laboratories

# 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 for all quads
- 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

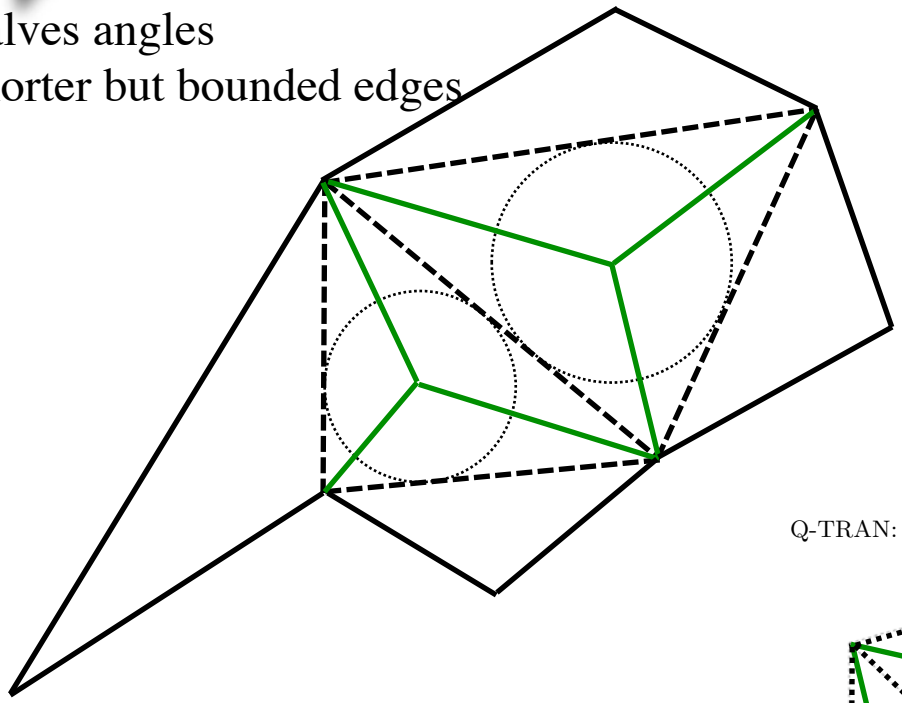


This is the only hexagon triangulation using only red edges, since every blue vertex must be in an ear

Sandia National Laboratories

# Incircle refinement details

halves angles
shorter but bounded edges



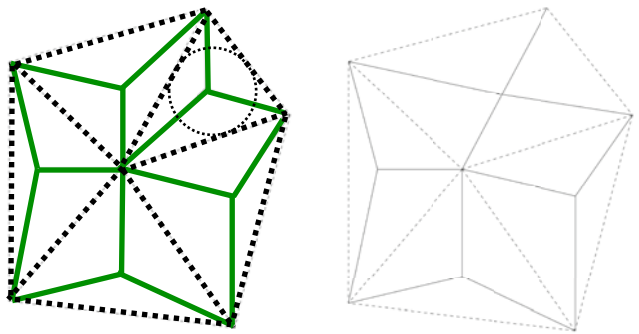
2n-gon has n-2 mono-triangles

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

adjacent mono-triangles OK
makes a red quad

Q-TRAN: Transform Triangular Meshes into Quadrilateral Meshes 27

Compare to Q-TRAN



(a) Input (b) Output

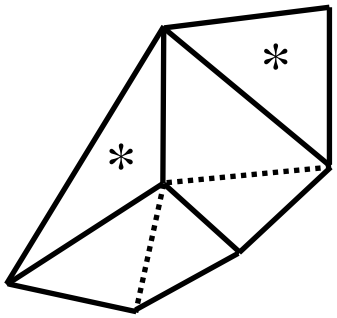
**Fig. 2.** Topology clean-up using face collapse to reduce the number of irregular vertices. A quadrilateral face is collapsed converting two irregular vertices into a regular one. The triangular tessellation is shown using dotted lines in both figures.

[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.

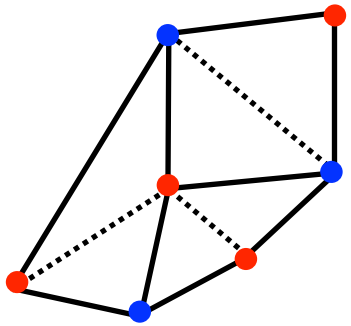
Sandia National Laboratories

# Pairing different than two-coloring
## (unless all-quad without refinement)

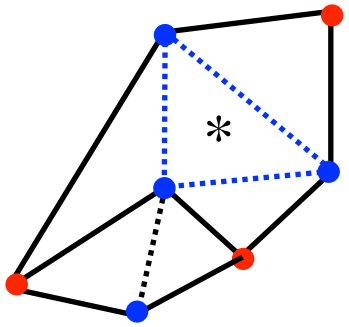
The stuck configurations are very different



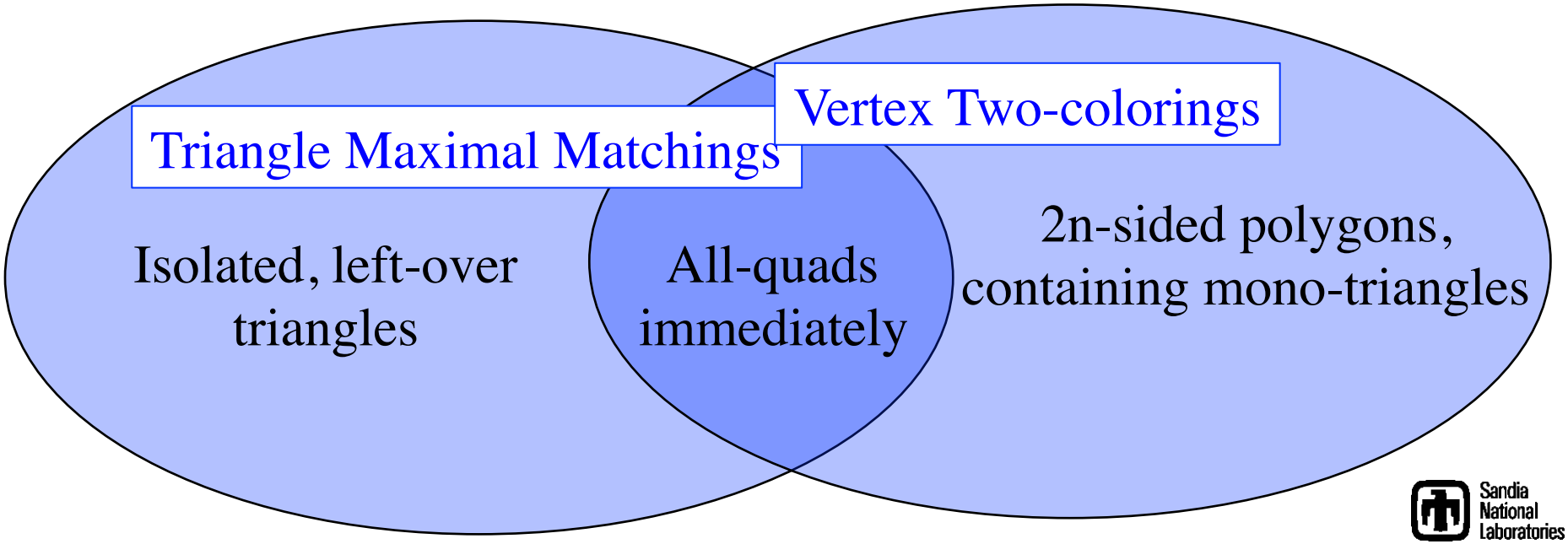

Two isolated
Unpaired triangles



All-quad
Perfect Matching <-> Two-coloring




6-sided polygon



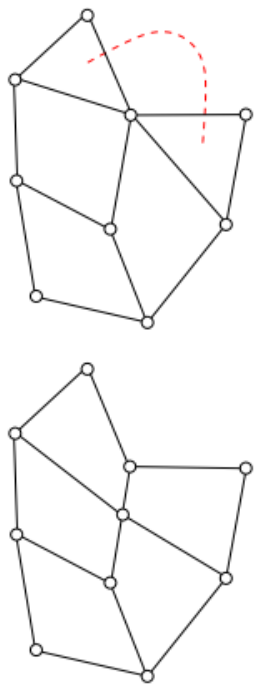

Sandia National Laboratories

OK, it works and is different. Why would you *want* to use it?

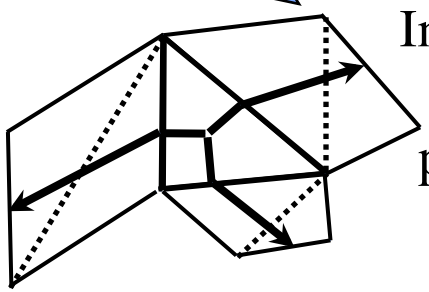
# Why two-coloring vertices?
# Why not matching triangles?

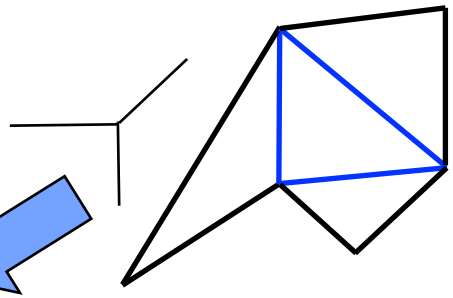
Locality!

must make a change



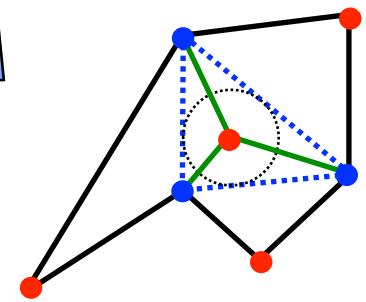
Pairing





Impossible topology
for
pairing or coloring

Coloring



local template
deterministic, provable quality

global refine chords
steering heuristics

Blossom-quad
local refinements

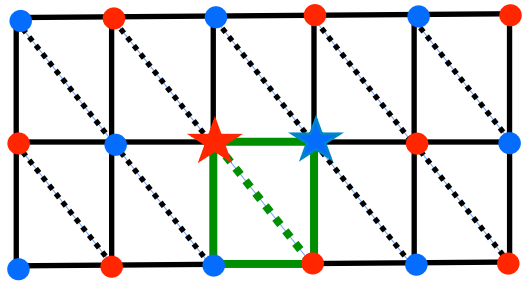
good (best?) triangle pairing algorithm

[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.

Sandia National Laboratories

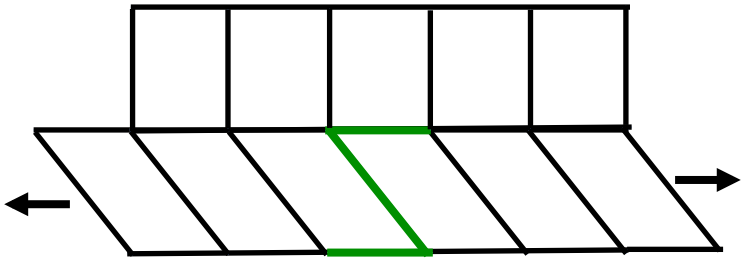
# Why two-coloring vertices?
# Why not matching triangles?

want a change



swap diagonal

Pairing

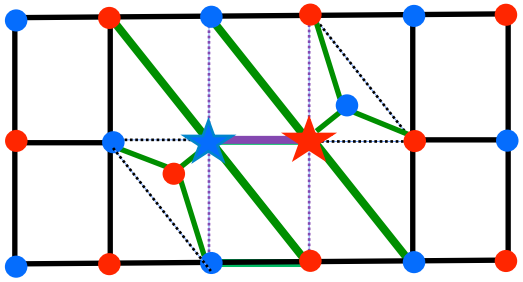


cascade of swaps

- **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**

- + match for quad quality
- – slow
  - - global matching alg
  - - quadratic runtime
- – rare isolated tri (unmatched)
  - tri 1->3 quad refine
    - + fixed vertices
    - - global propagation
  - + alternative local refine
    - (complicated, several rules)
- – local pair swap
  global cascade
- – global difficulties

**Two-coloring vertices**

- – colors don't measure quality
- + fast
  - + local coloring alg
  - + near linear runtime
- – rare isolated tri (monchromatic)
  - tri 1->3 tri refine
  - - adds vertices
  - + no propagation
- + local color flip
  local change
- + local difficulties

"-" means a negative feature
"+" means a positive feature

# Interval Assignment Interpretation

- Two-coloring is a local assignment
  - ensures even-number of boundary edges inherent global constraint for quad meshes
- Now let's use coloring for spatial positions

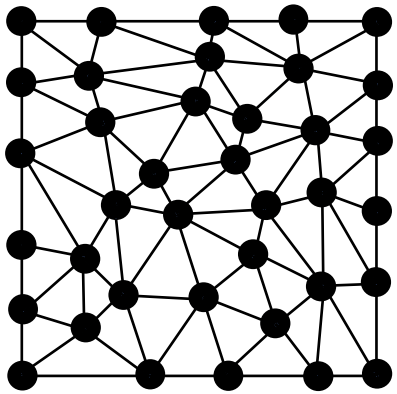
# Back to the algorithm

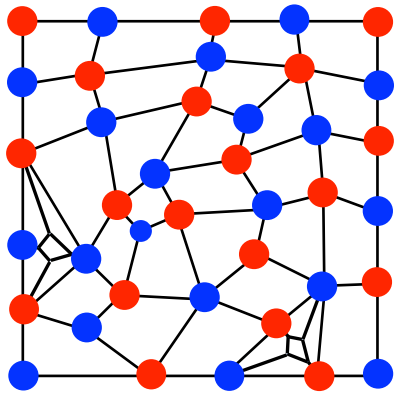
- **Generate (or given) well-spaced points**
- Delaunay triangulate
- Color points red or blue
  - intersperse colors
- Discard red-red and blue-blue edges
- Quads mostly
  - good quality, some large angles
- 6, 8, 10 sided polygons sometimes
  - constrained incircle refinement
  - median template for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

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

**We can take any Packing or 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)**

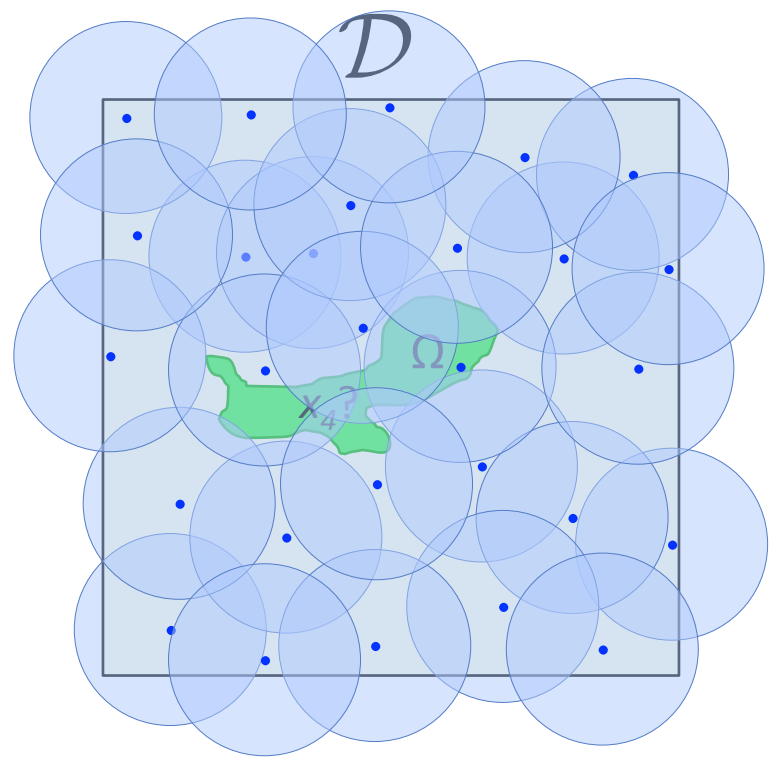
Empty disk: $\forall x_i, x_j \in X, x_i \neq x_j : ||x_i - x_j|| \geq r$

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**

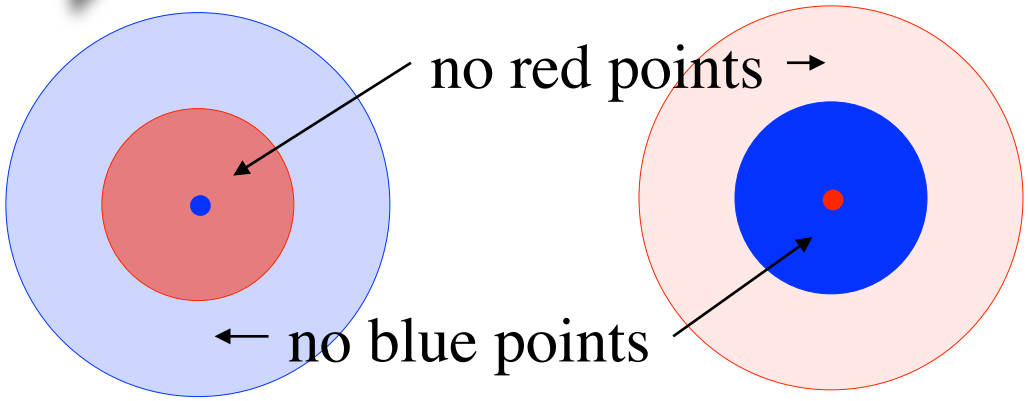
- **What is MPS? Sphere packing, output of**
  - **Insert random points**
    - **With "Poisson" process, and rejection**



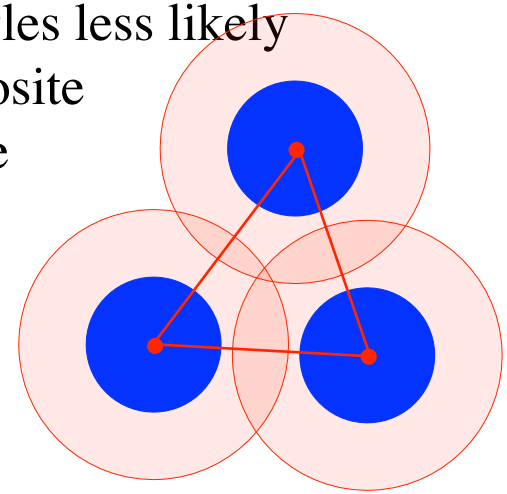

- **sphere packing algorithms are practical, our 2011-2013 work**
- **see Dafna Talmor thesis for quality theory**

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

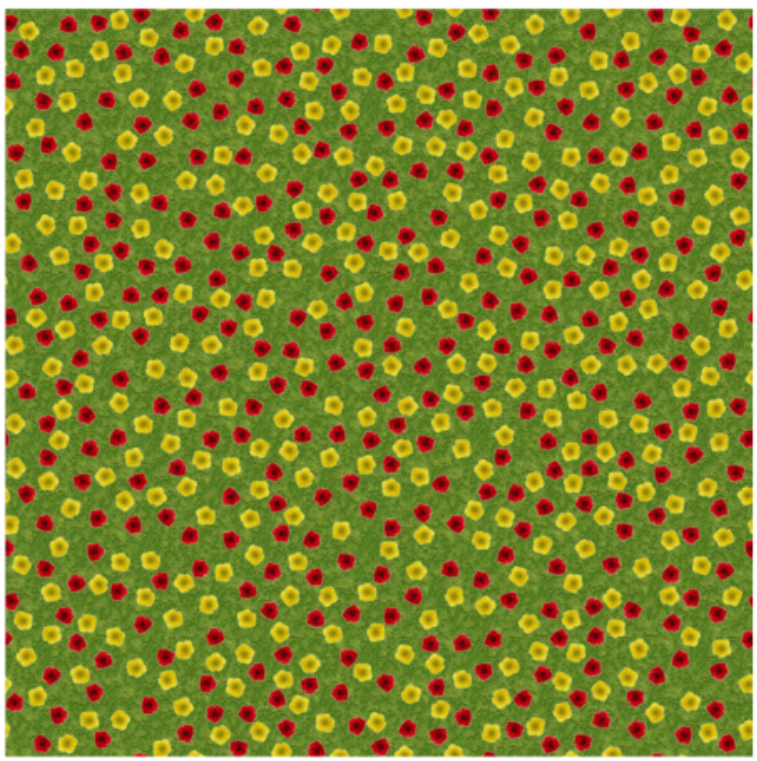
[17] L.-Y. Wei, Multi-class blue noise sampling, ACM Trans. Graph. 29 (2010) 79:1–79:8.

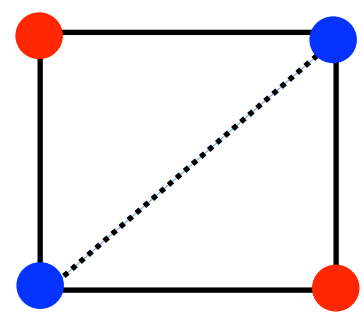



Blue-blue farther than blue-red
Mono-triangles less likely
can fit opposite
color inside



2 classes of objects

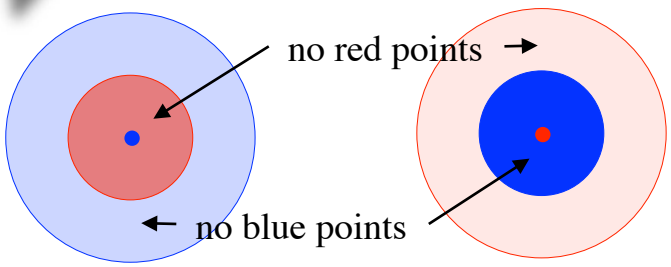




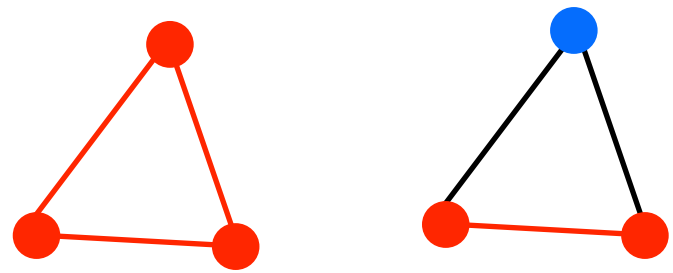
discard long diagonals
for square-like quads

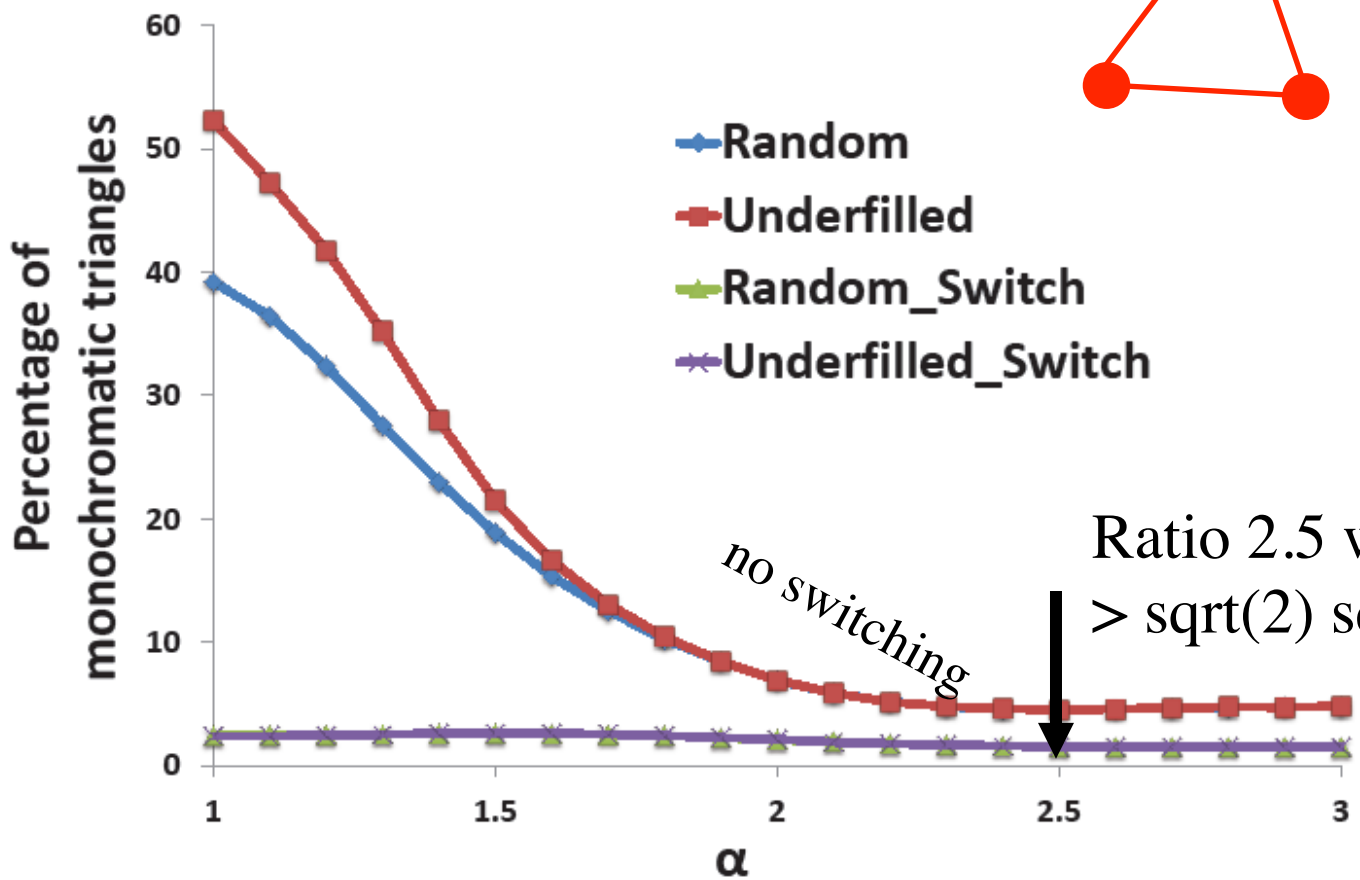
Sandia National Laboratories

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




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



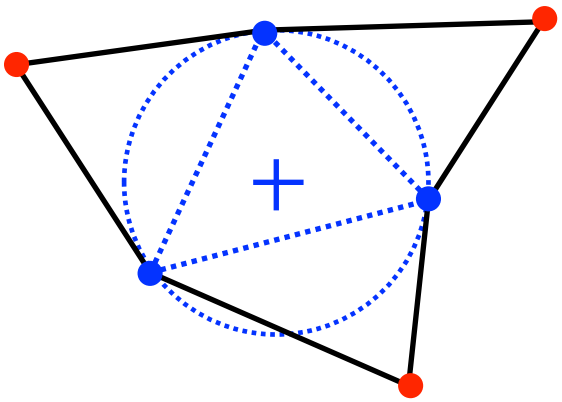


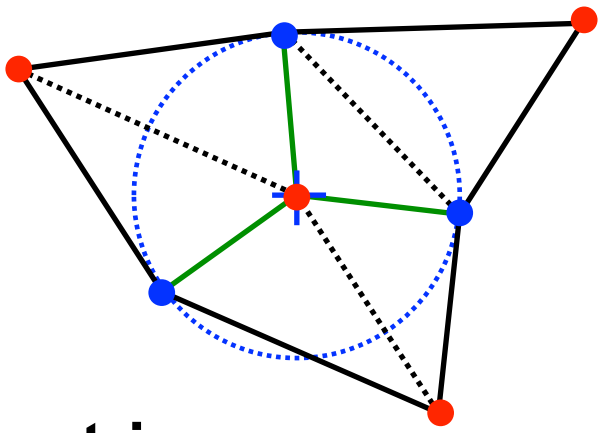

Ratio 2.5 works well,
> sqrt(2) square diagonal

switching, stubborn 2%

# Resolving Stubborn Mono-tris

## Circumcircle Delaunay Refinement



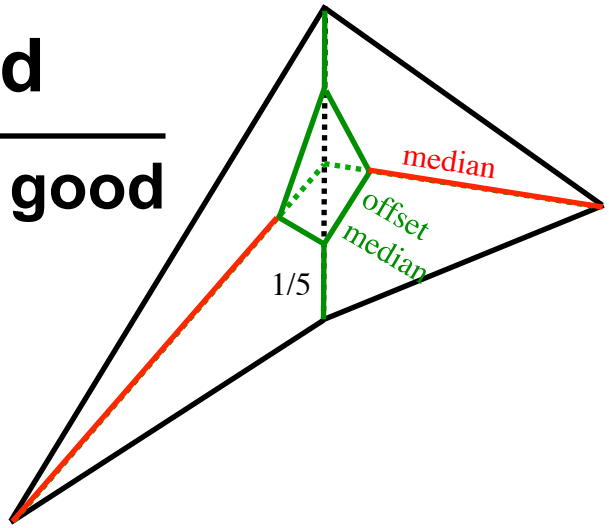


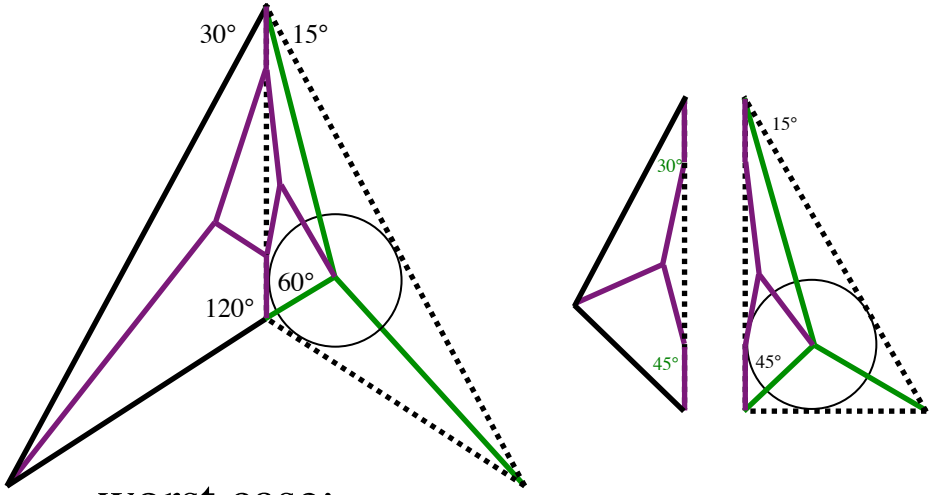
**Removes the mono-tri**

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

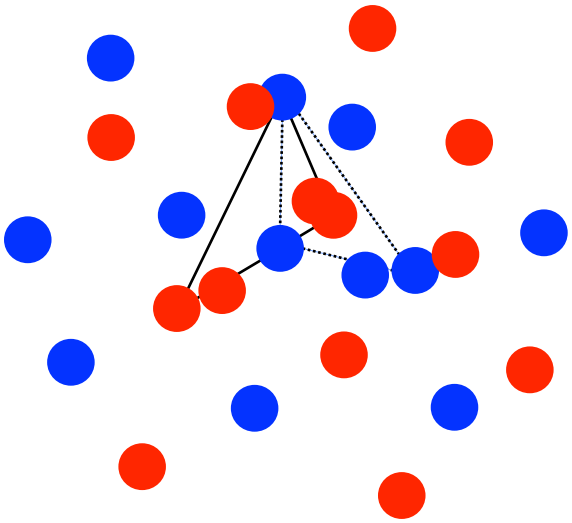
# Avoiding Reflex Quad

- **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

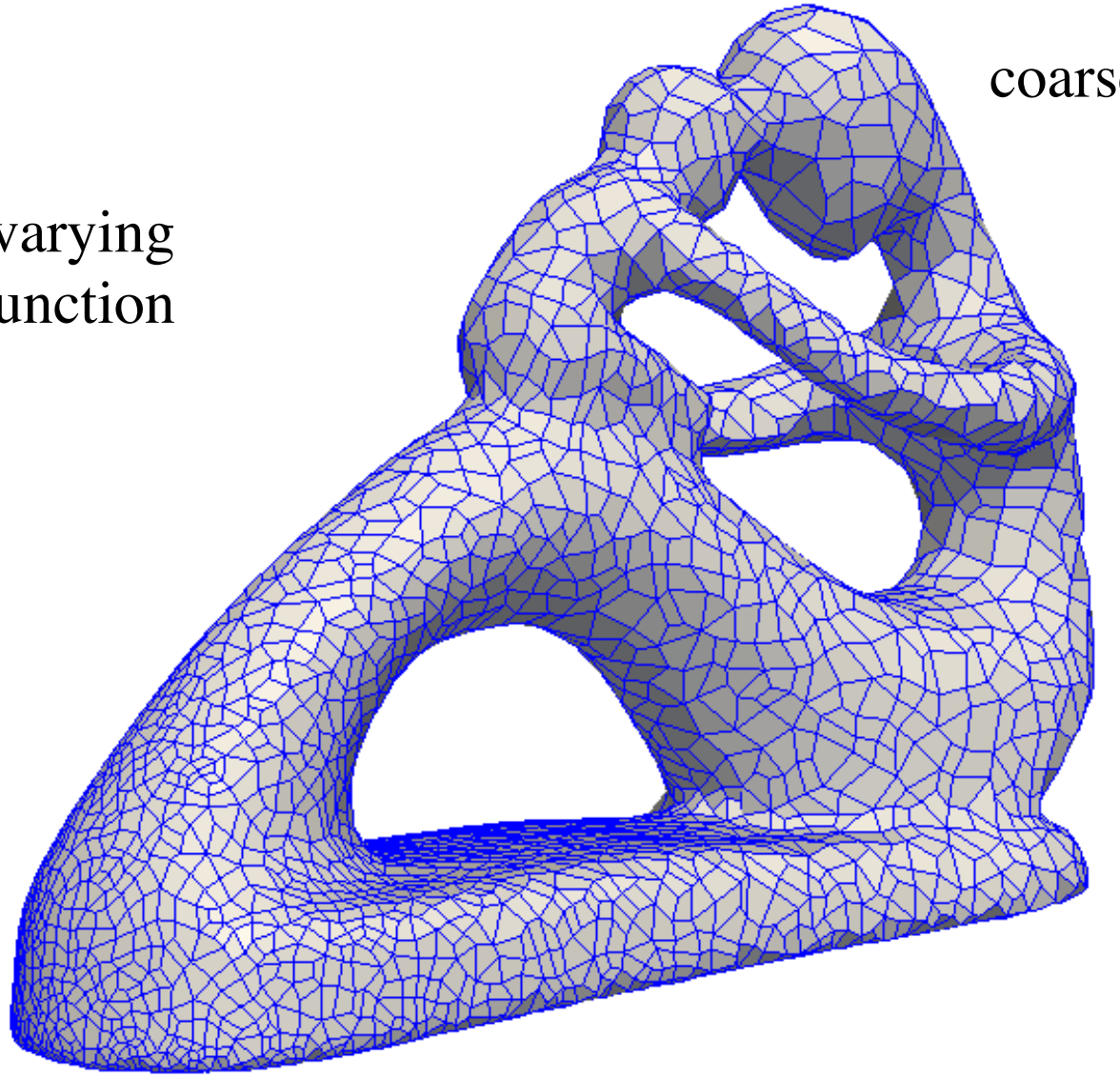
- Reduce frequency of mono-tris
  - Two-color multiclass sampling with radii ratio 2.5
  - Color switching
- Resolve mono-tris
  - Delaunay incircle refinement
- Avoid reflex quads
  - Local resampling

- 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

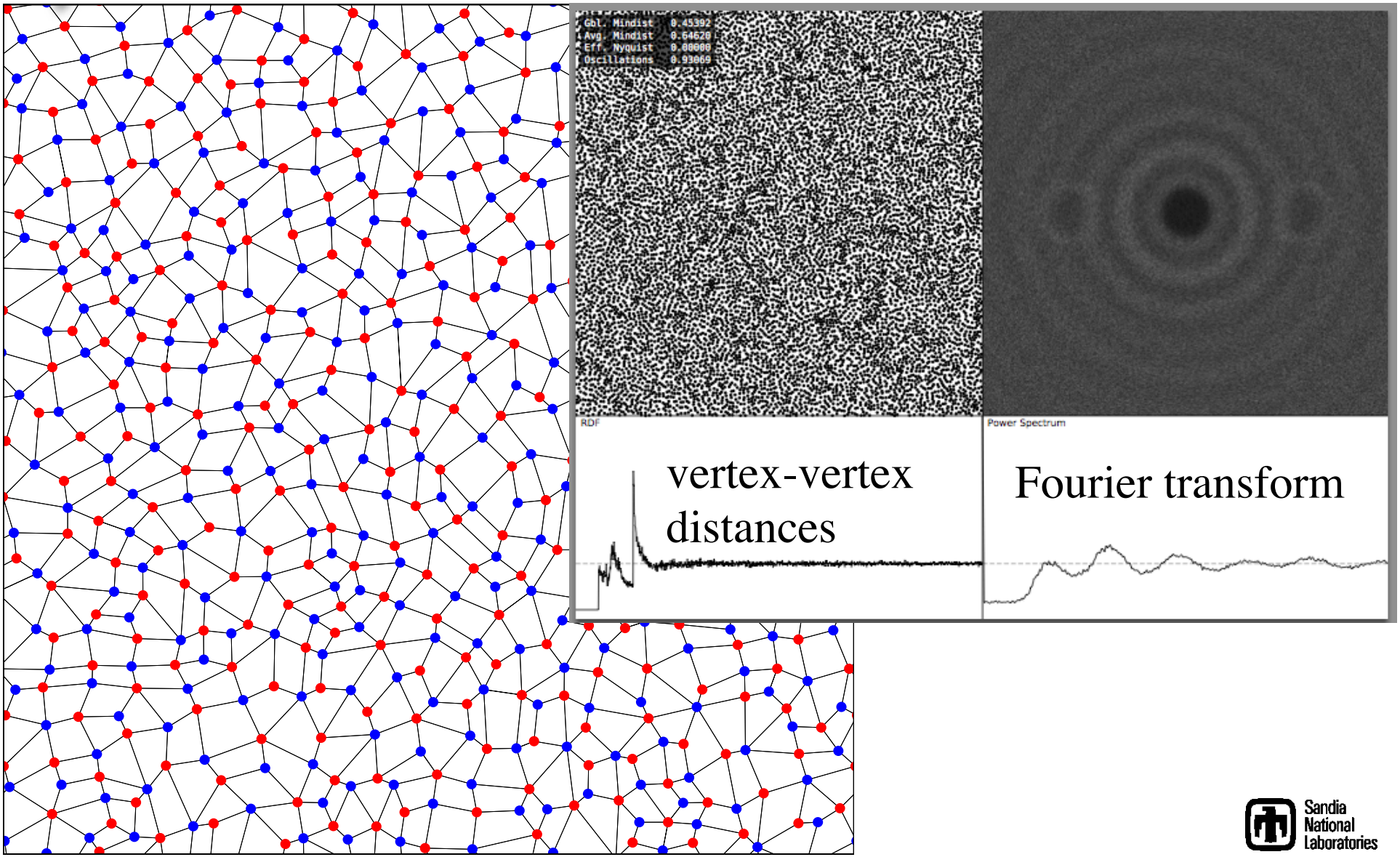


fine

Sandia National Laboratories

# Fourier Spectrum

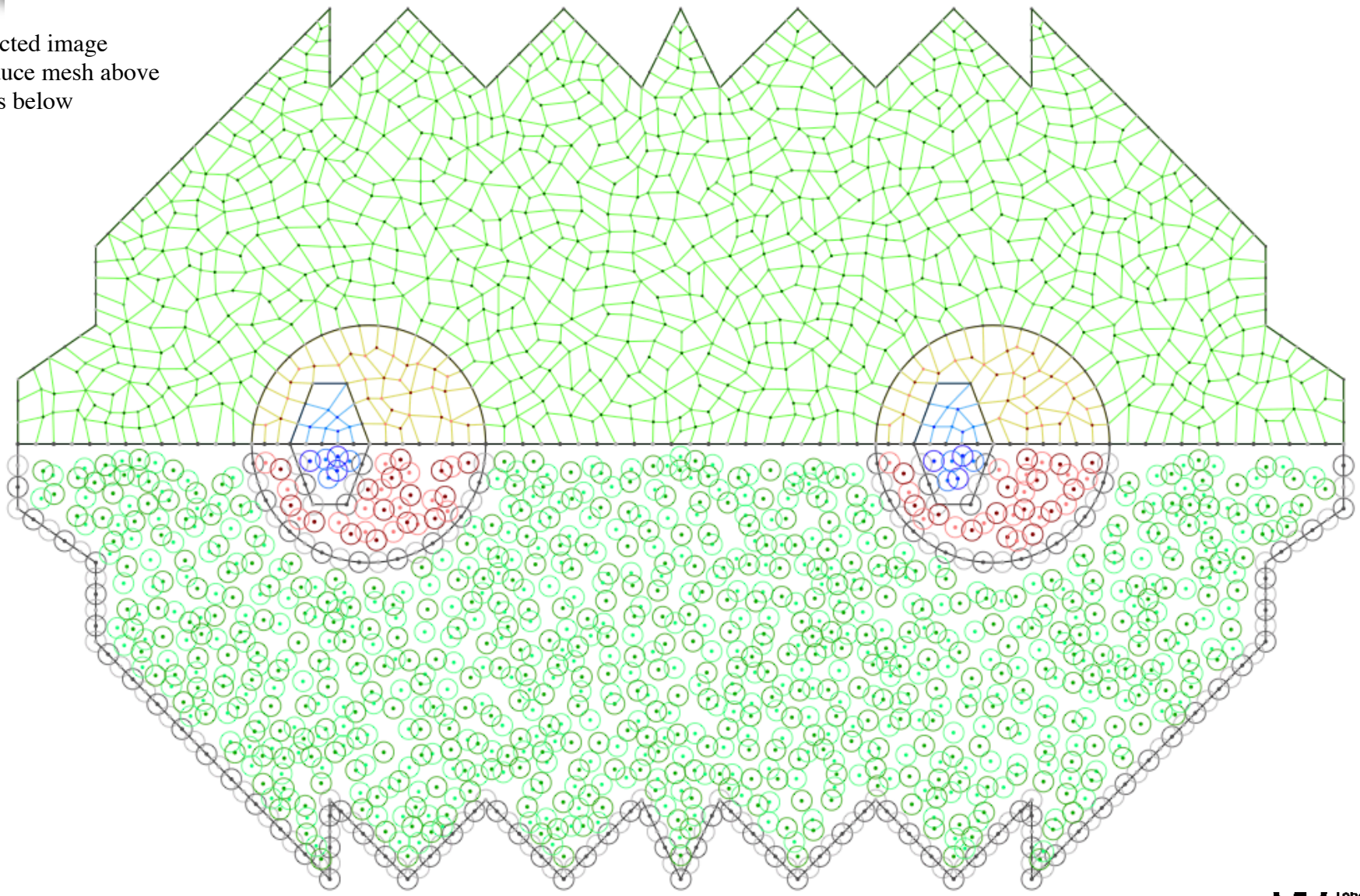
Some graphics applications rely on random positions to avoid artifacts




Sandia National Laboratories

# Random

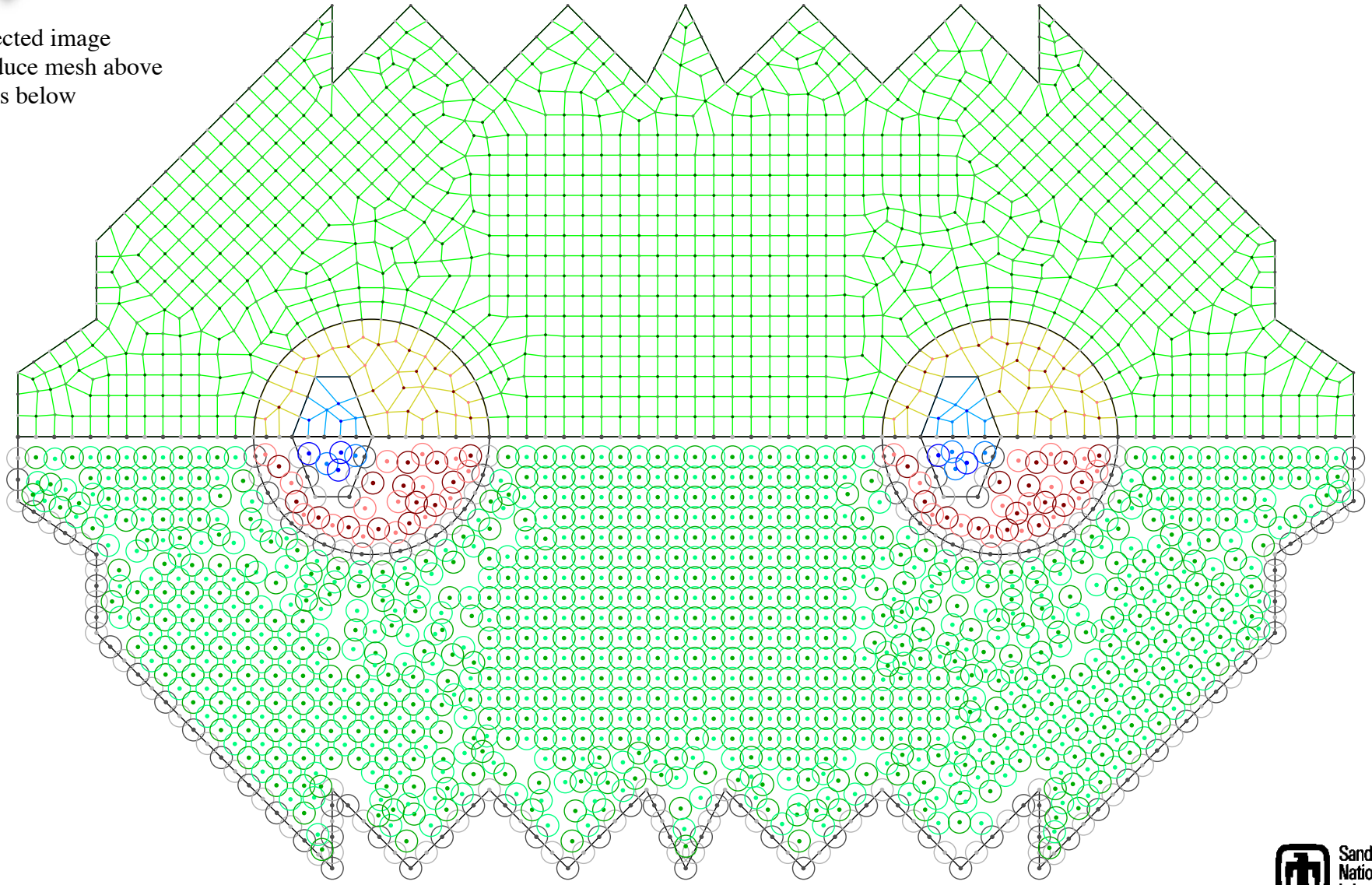
reflected image
produce mesh above
disks below



dia
onal
Laboratories

# Structured

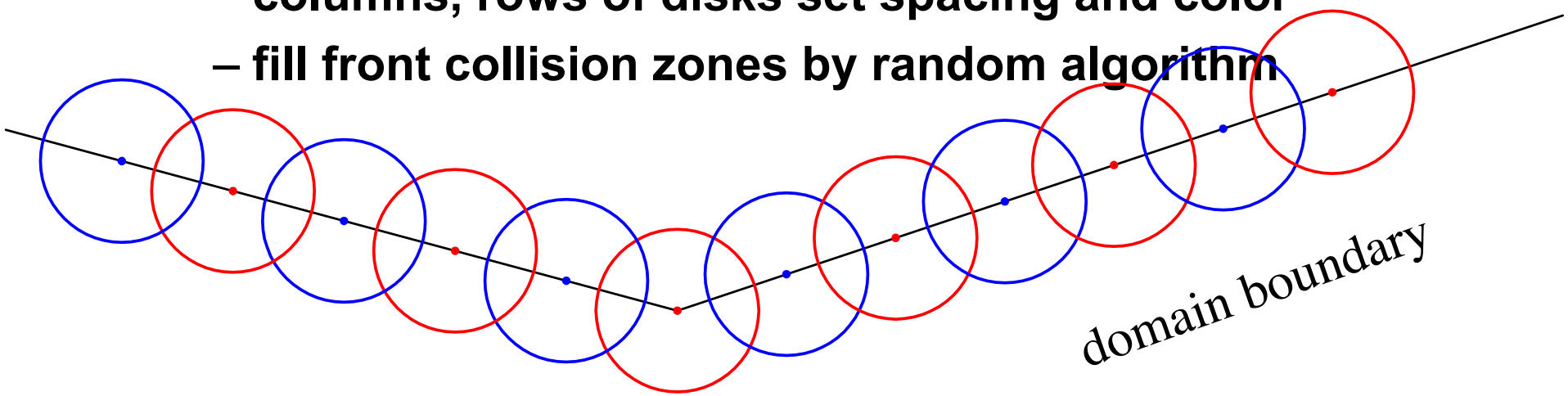
reflected image
produce mesh above
disks below



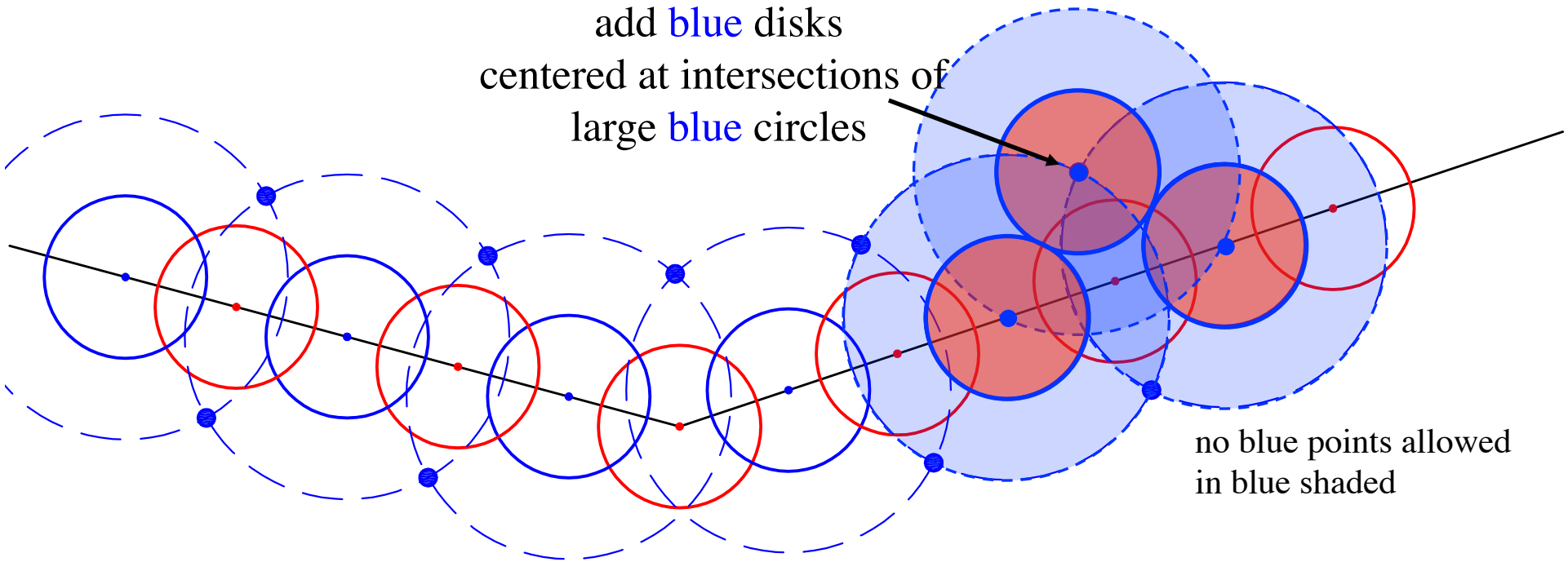
Sandia National Laboratories

# Advancing Front for Structured

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



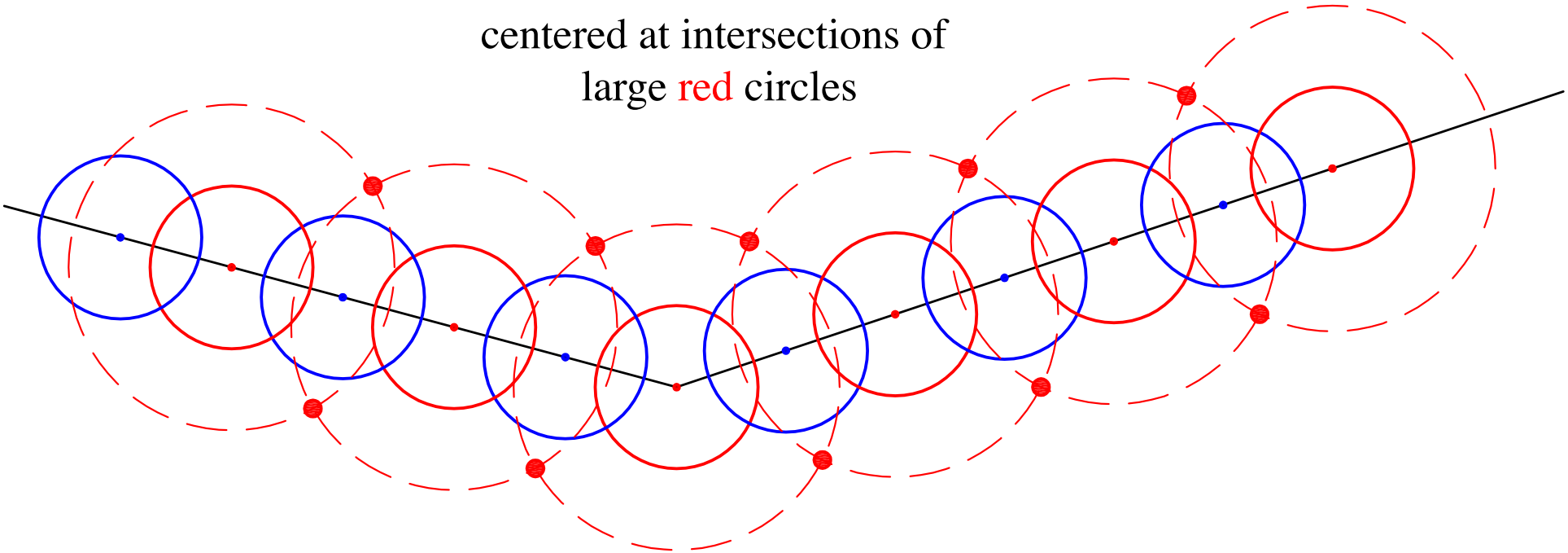
# Advancing Front for Structured



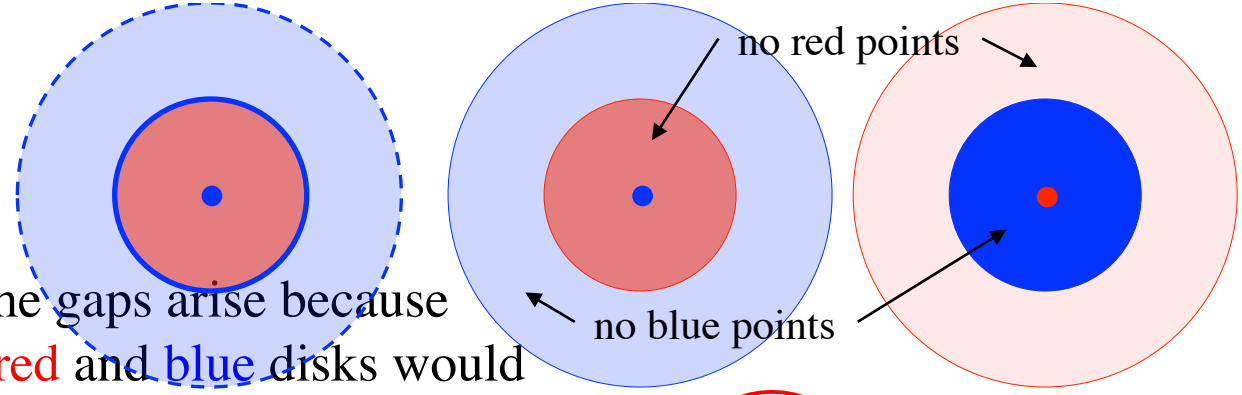

This is the closest we can place blue disks
and not violate the blue-blue separation distance

# Advancing Front for Structured

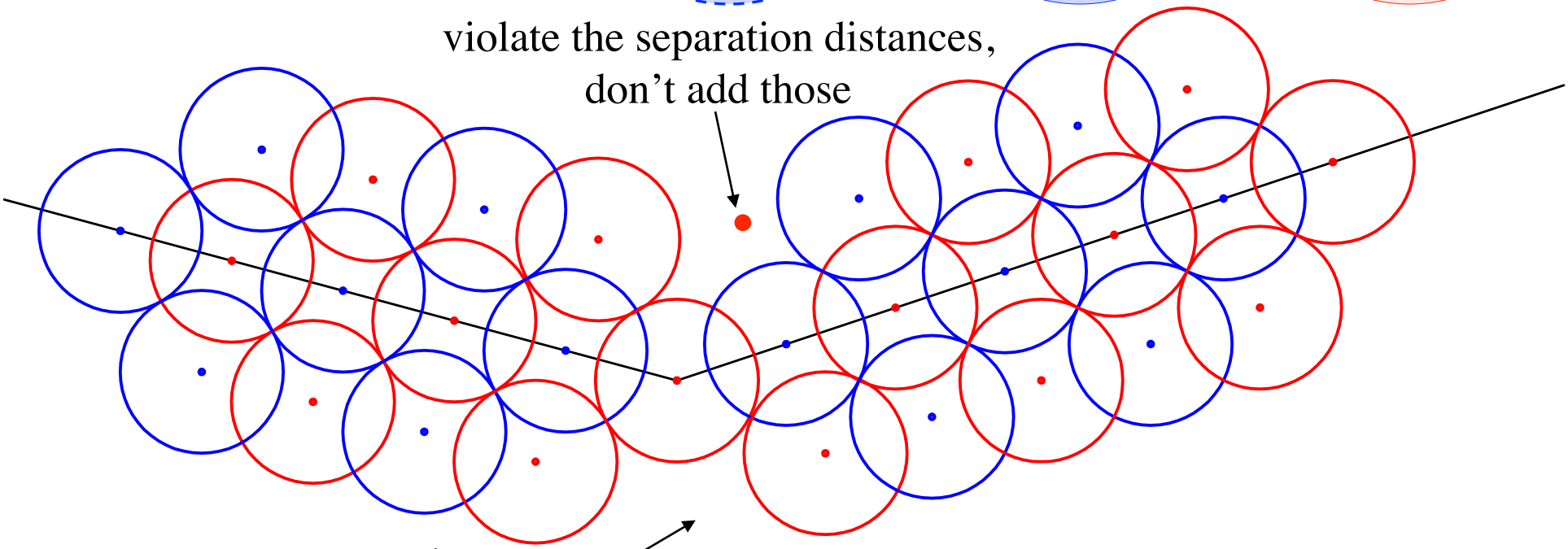
add red disks
centered at intersections of
large red circles



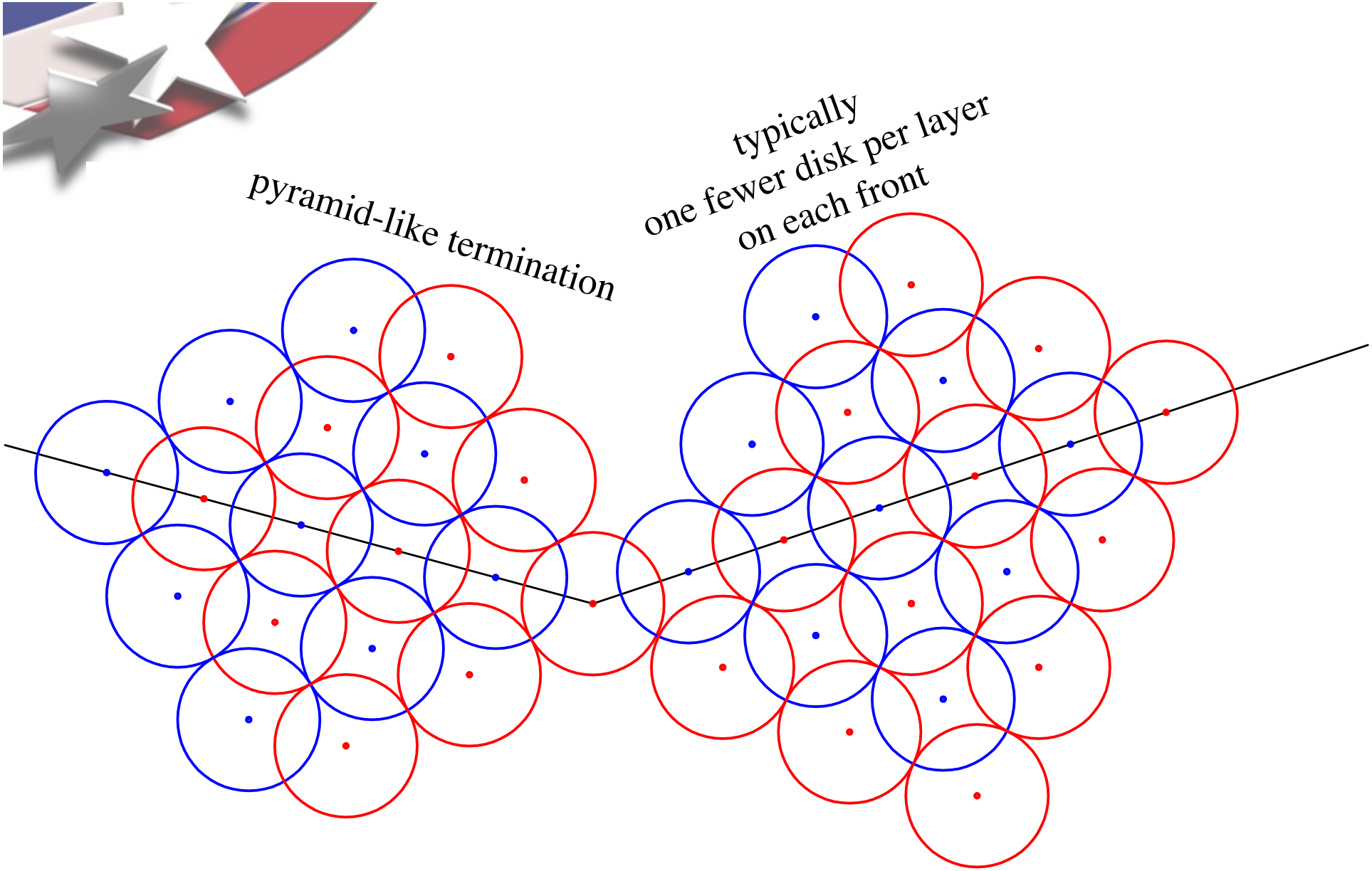
Sandia National Laboratories



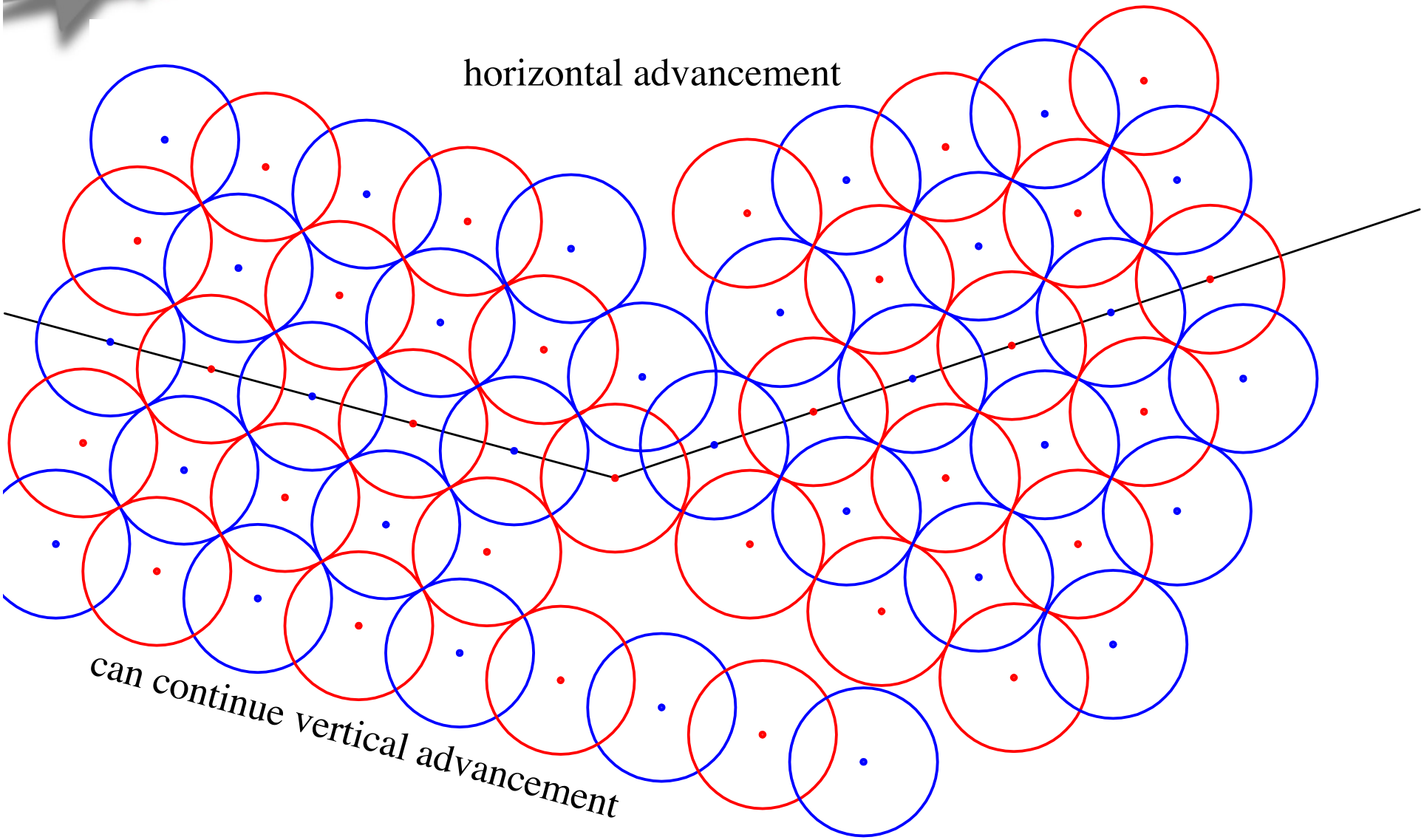

some gaps arise because
some red and blue disks would
violate the separation distances,
don't add those



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

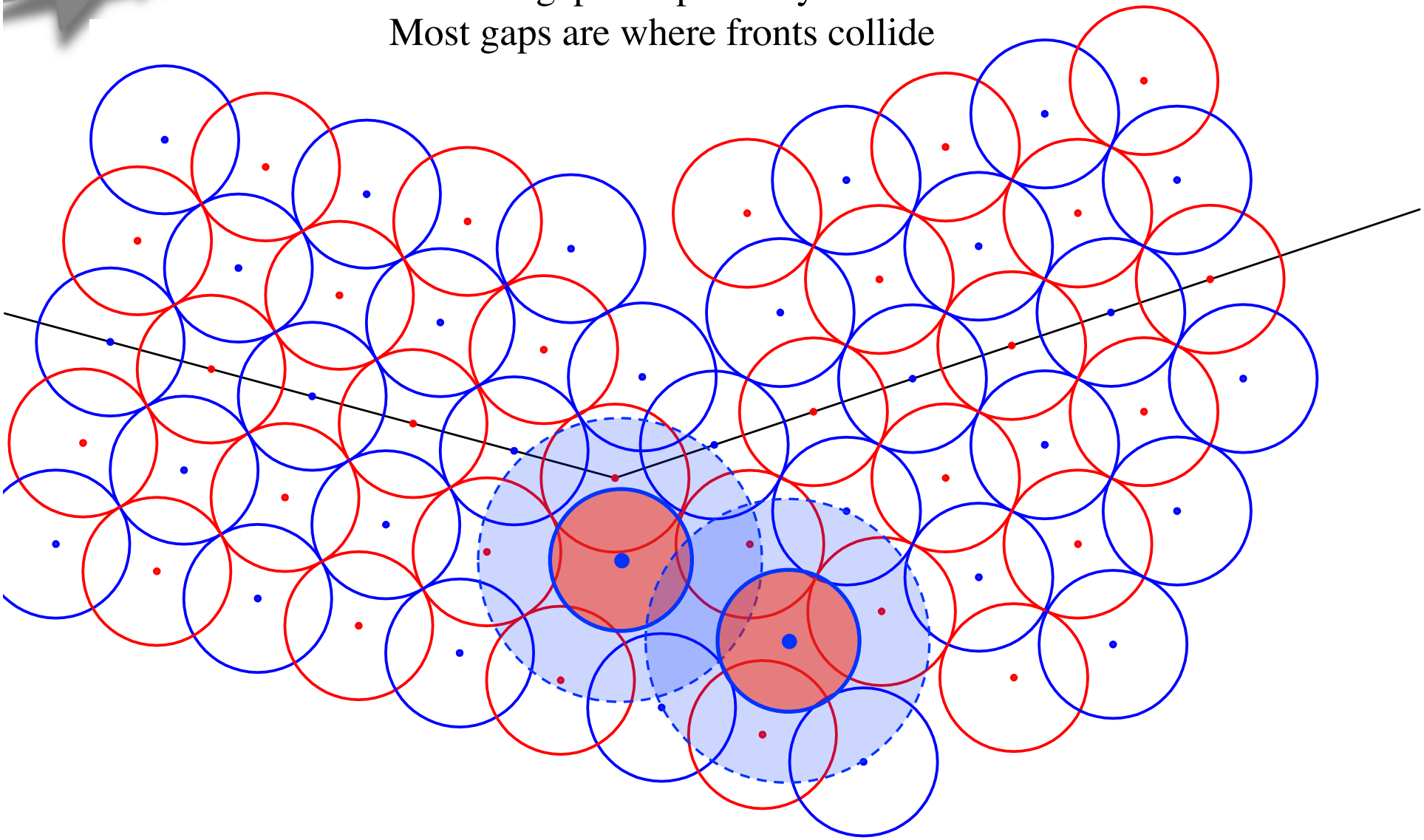



Sandia National Laboratories



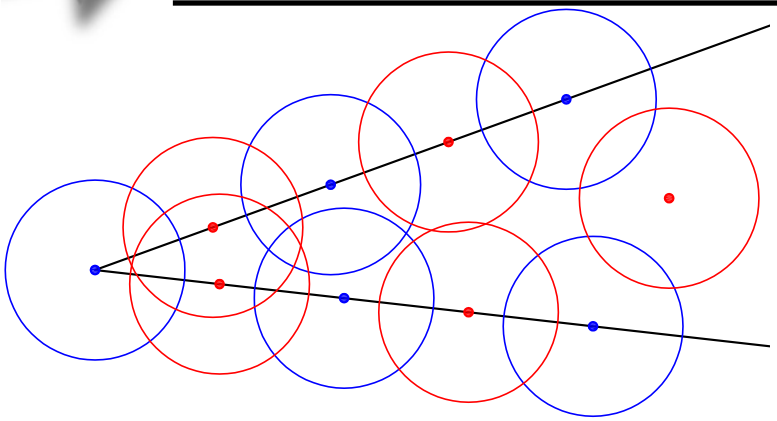

Sandia National Laboratories

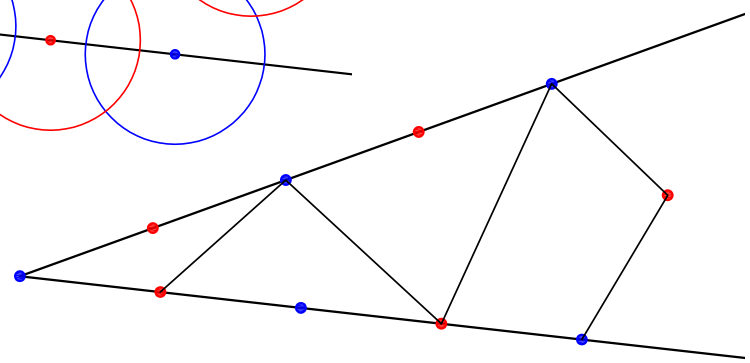
fill large gaps with random algorithm
small gaps are provably OK
Most gaps are where fronts collide

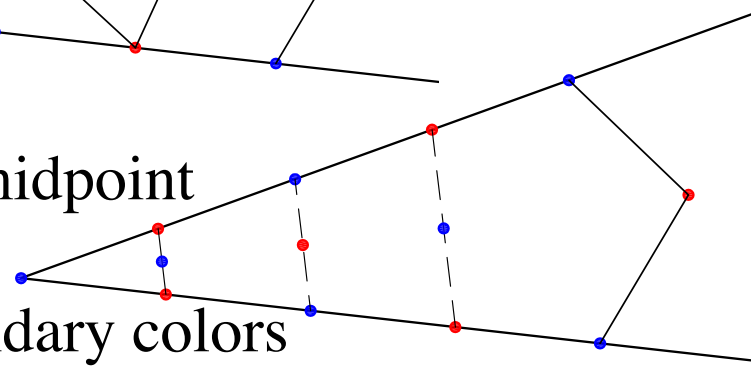


Sandia National Laboratories

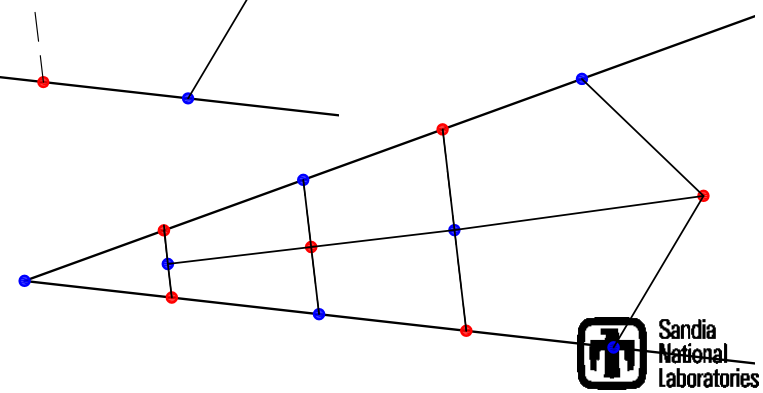
# Sharp Corners



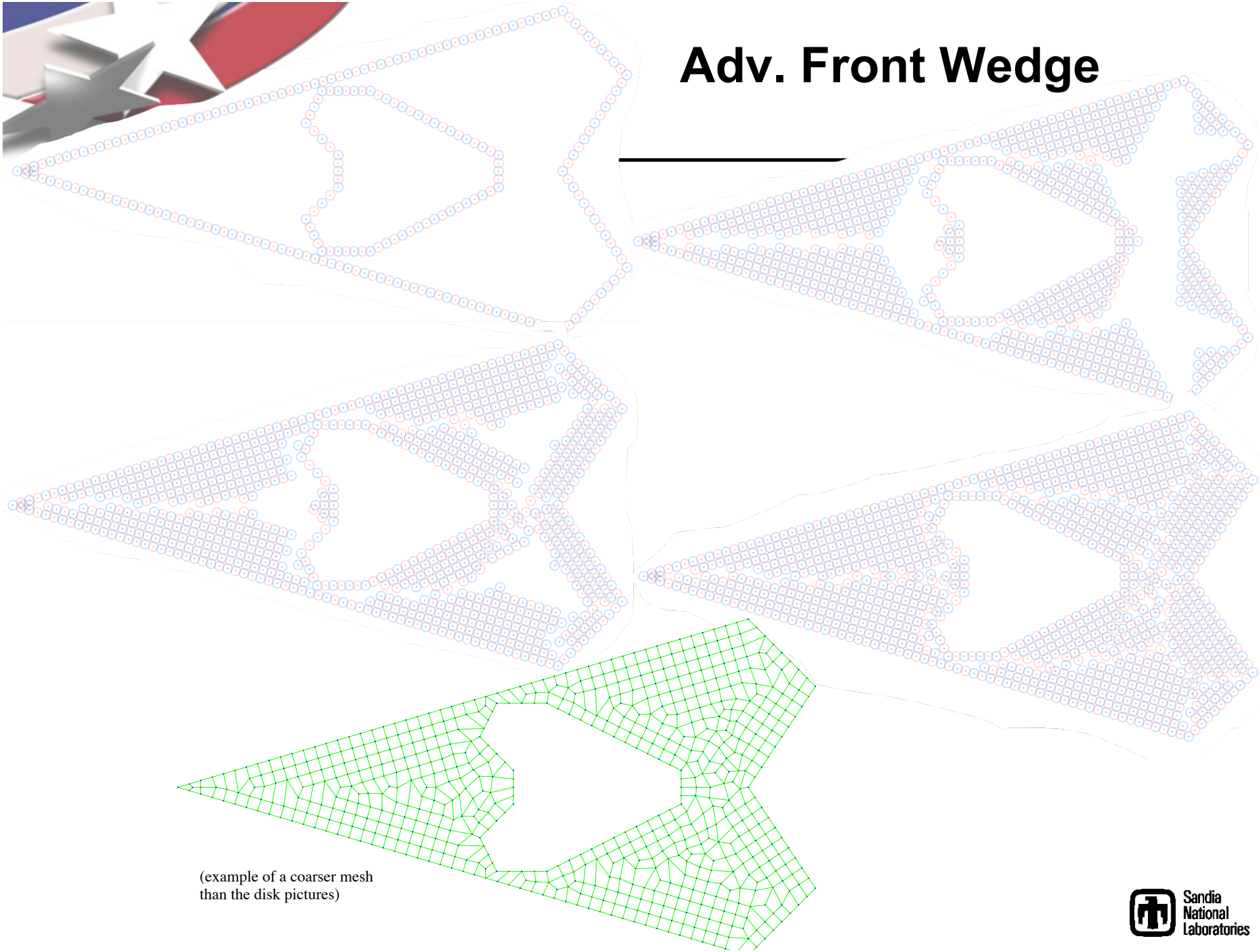




Add opposite-colored midpoint
on mono-edges
exploit alternating boundary colors

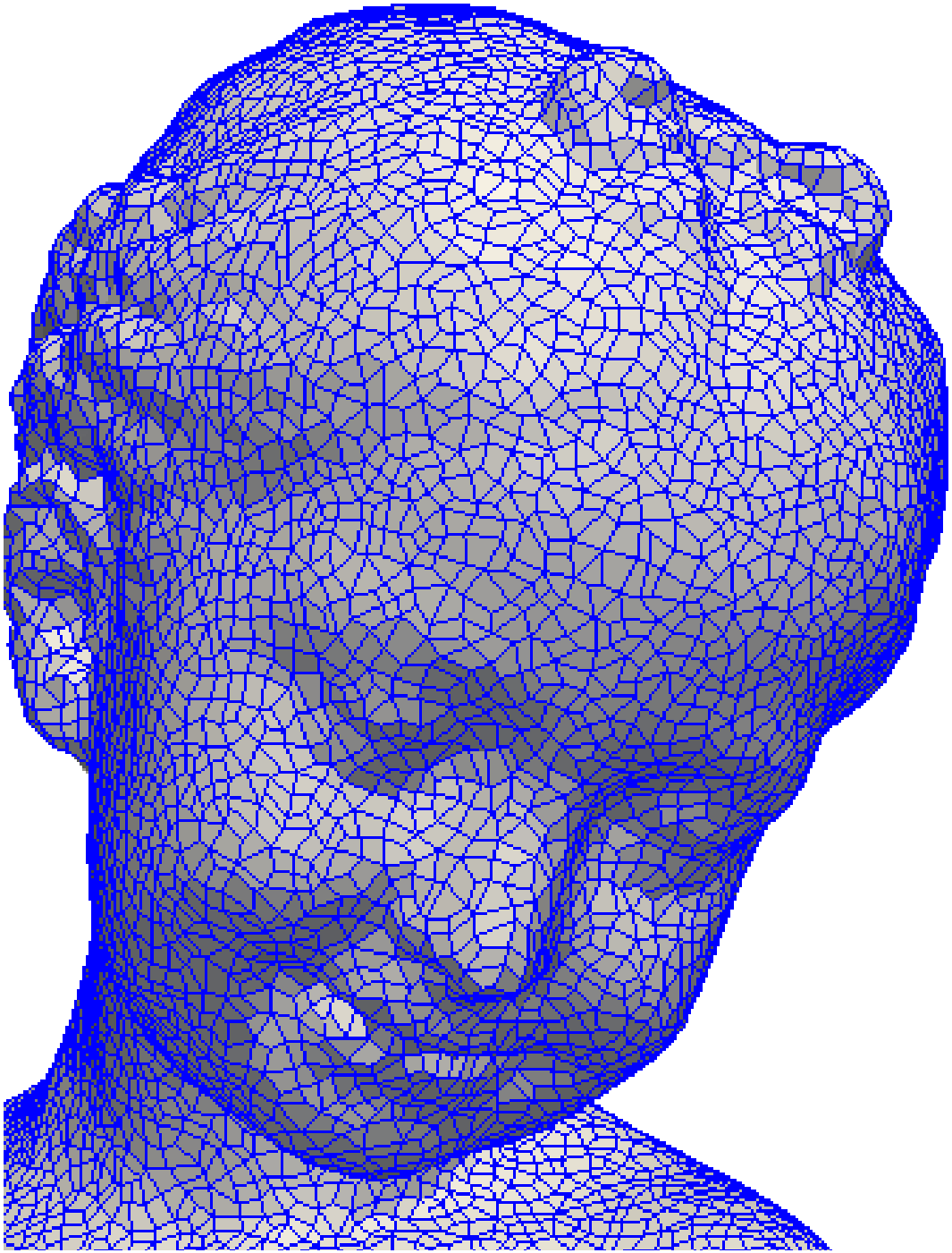


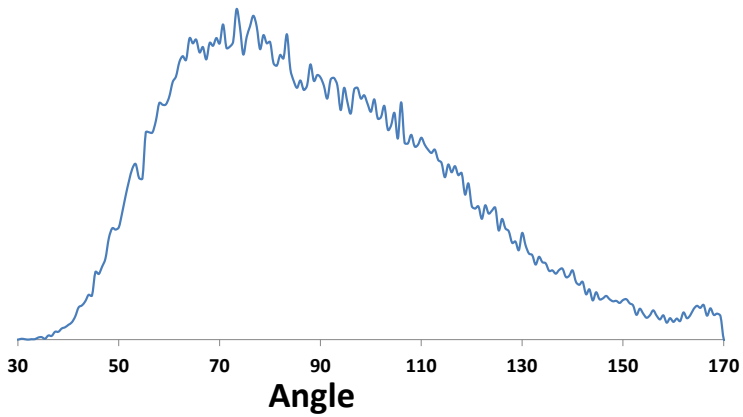
# Adv. Front Wedge

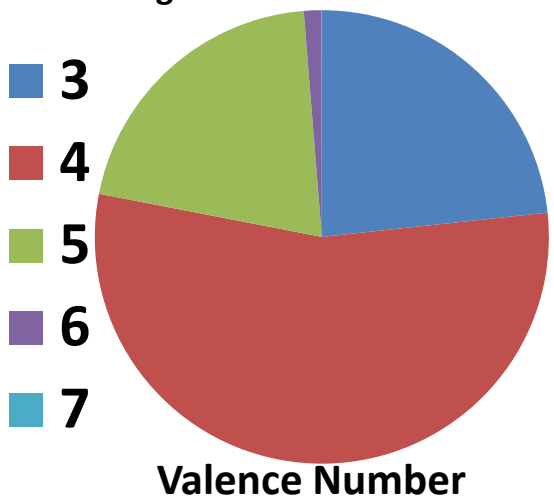


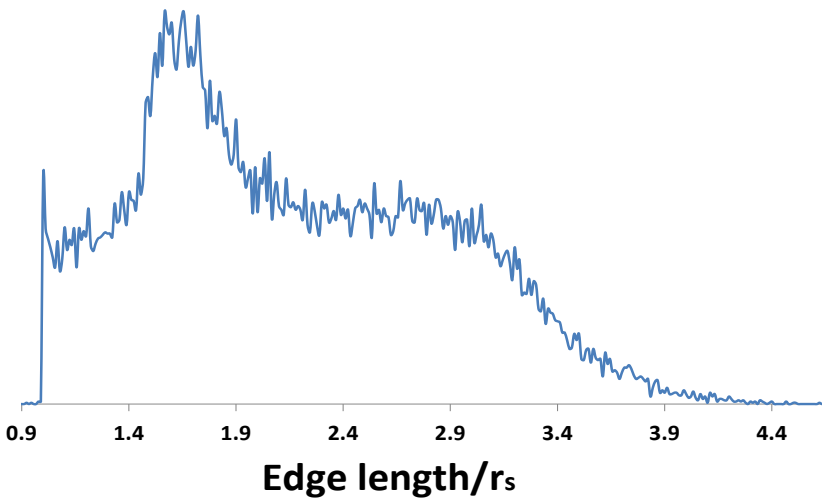
(example of a coarser mesh than the disk pictures)

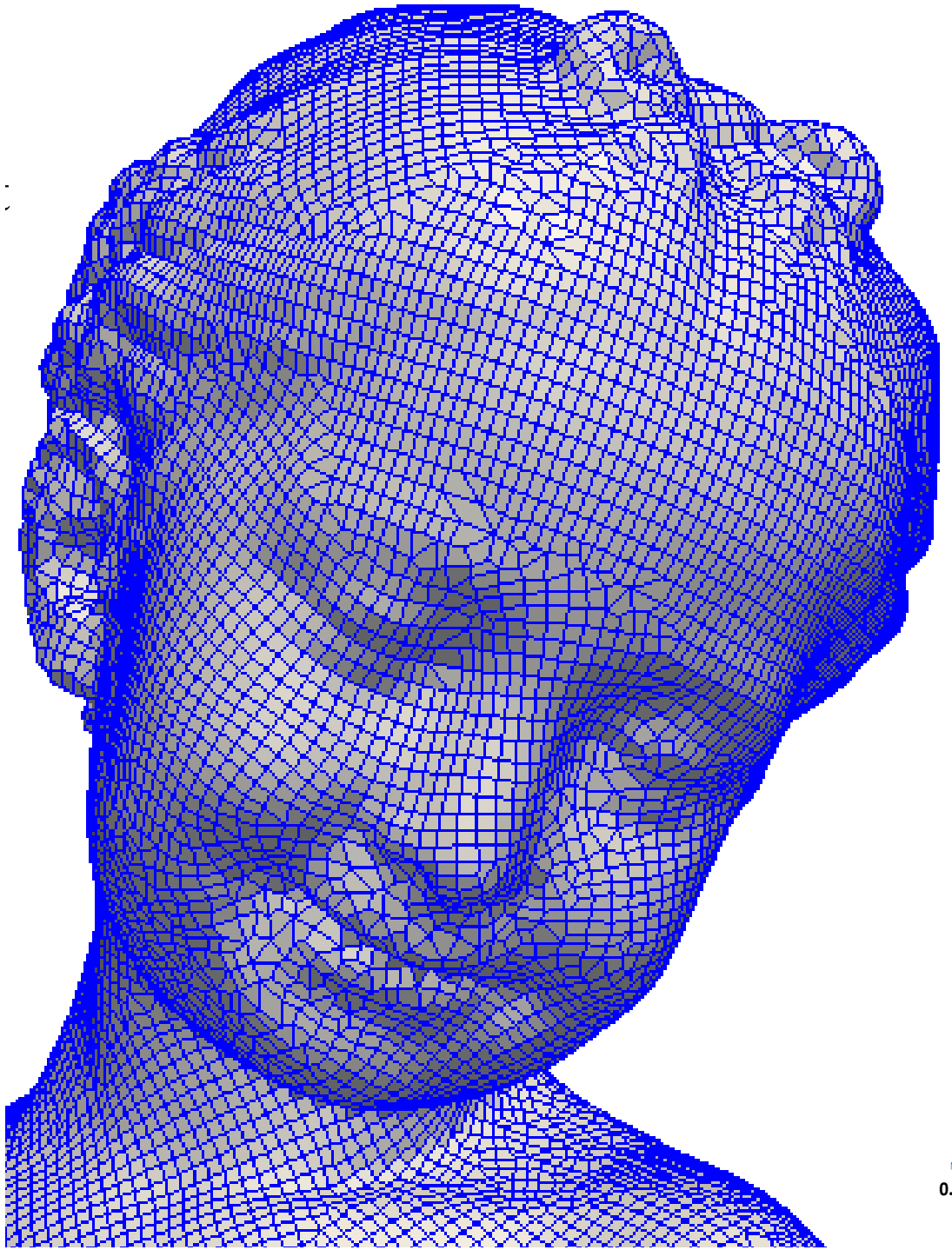
Sandia National Laboratories

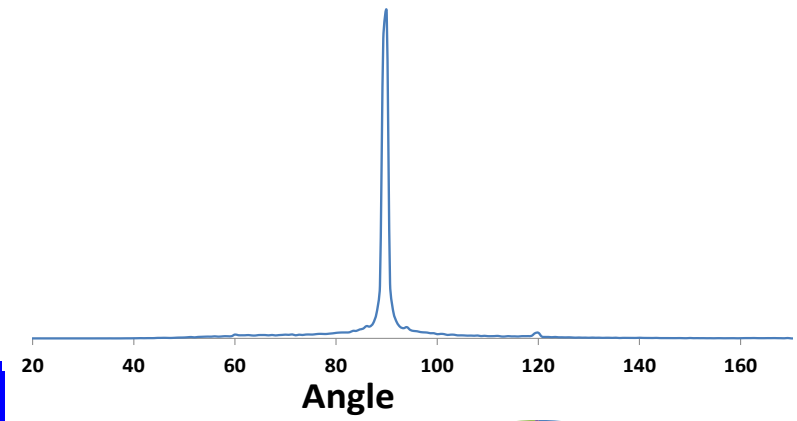


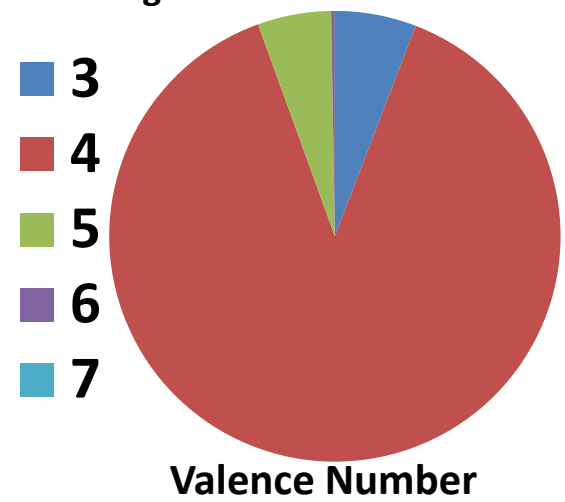

30
50
70
90
110
130
150
170
Angle

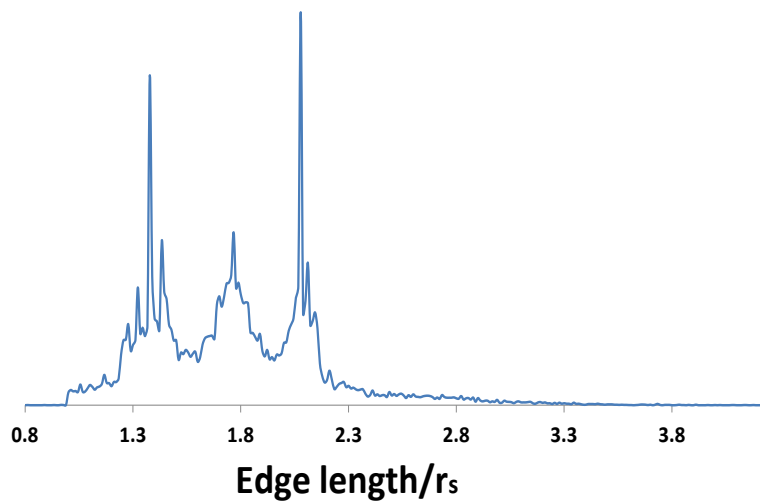

3
4
5
6
7
Valence Number


0.9
1.4
1.9
2.4
2.9
3.4
3.9
4.4
Edge length/rs

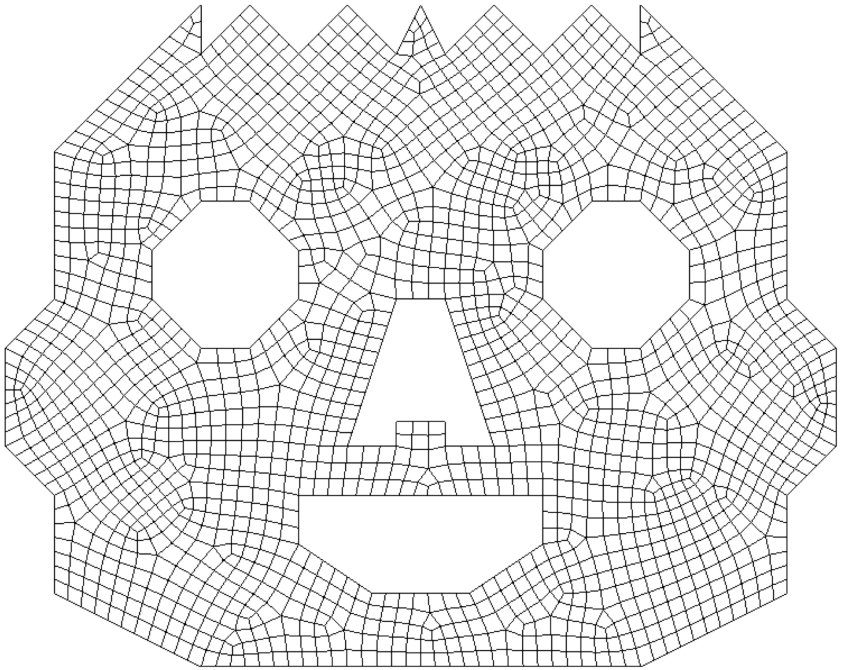



20
40
60
80
100
120
140
160
Angle

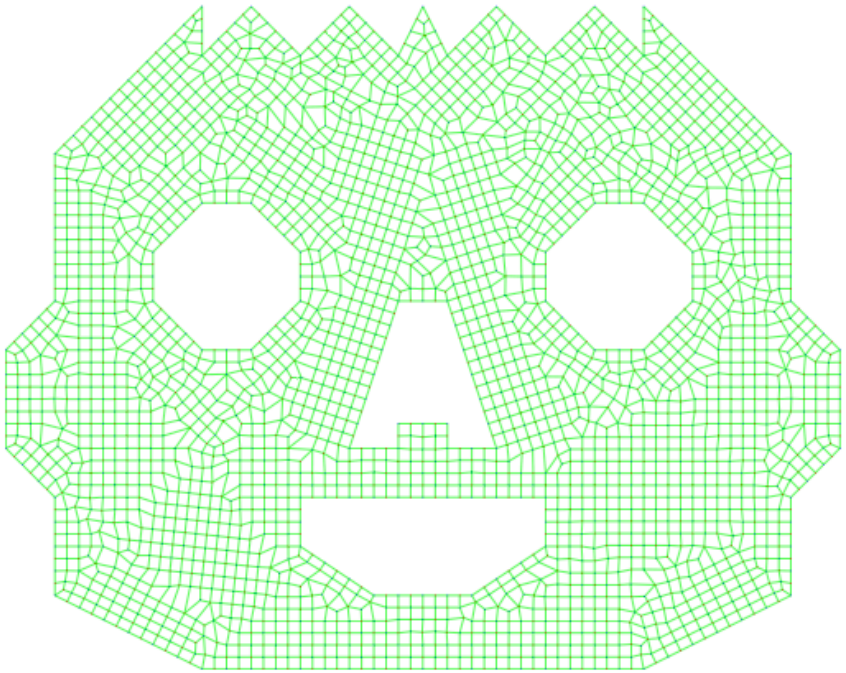

3
4
5
6
7
Valence Number


0.8
1.3
1.8
2.3
2.8
3.3
3.8
Edge length/rs

# Qualitative comparison to a production code



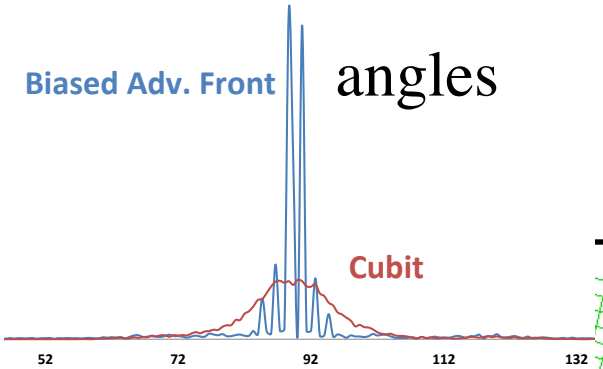
Cubit paver

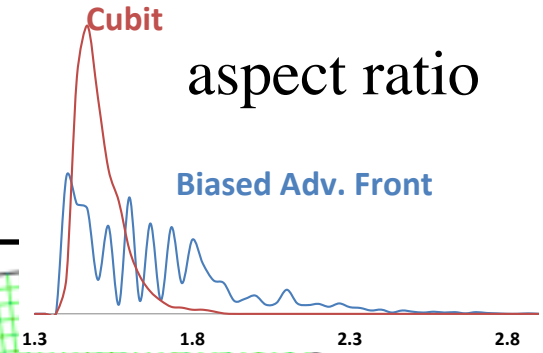


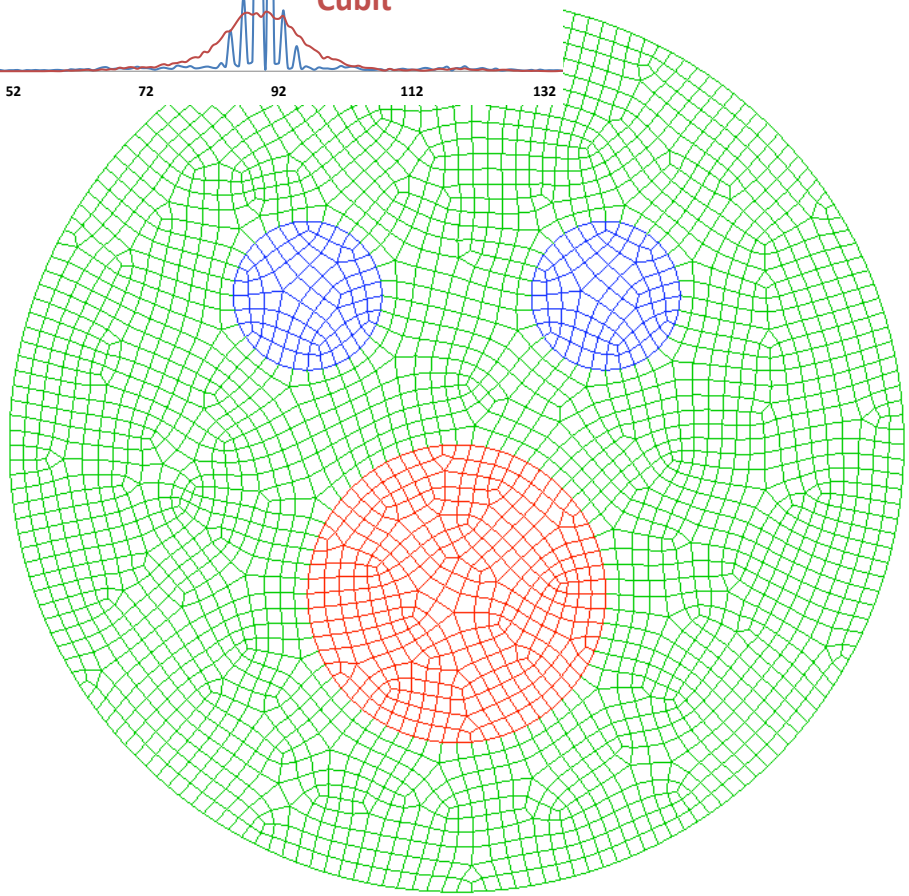
Two-color adv. front no cleanup

Sandia National Laboratories

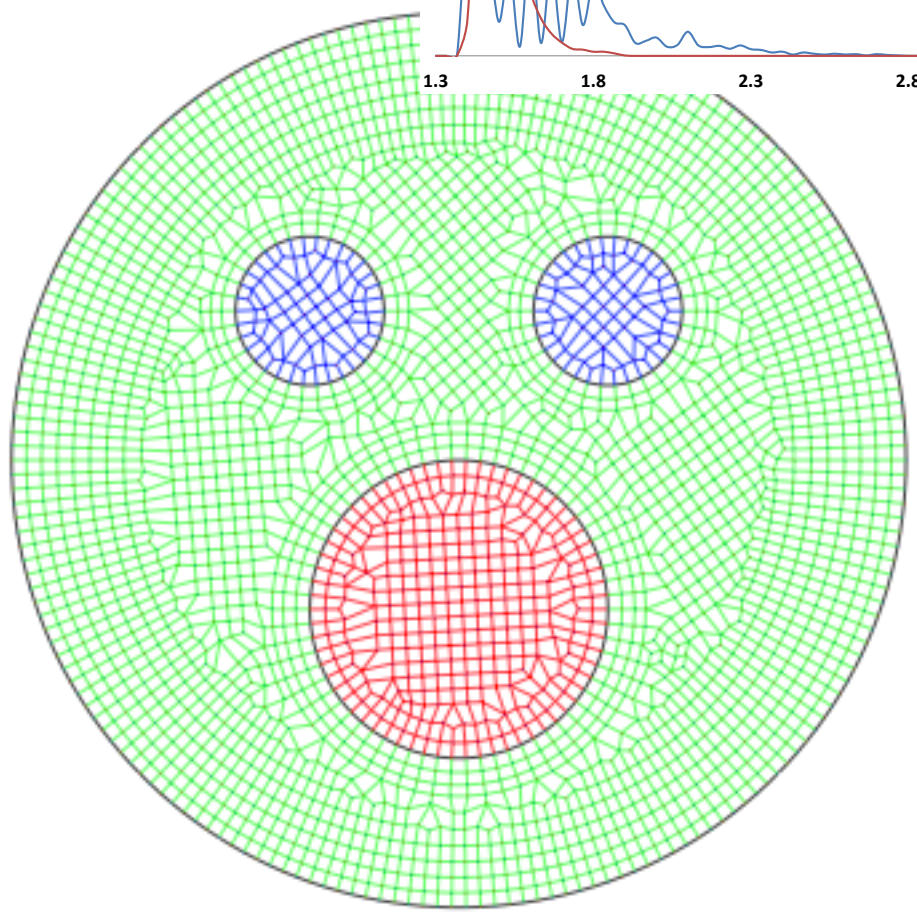
# Qualitative comparison to a production code



**Cubit paver**



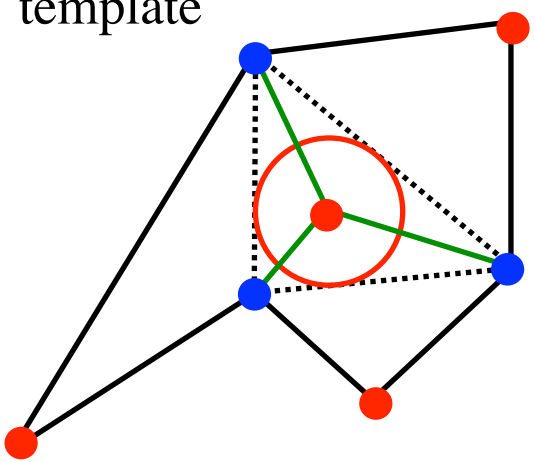
**Two-color "biased" adv. front no cleanup**

Sandia National Laboratories

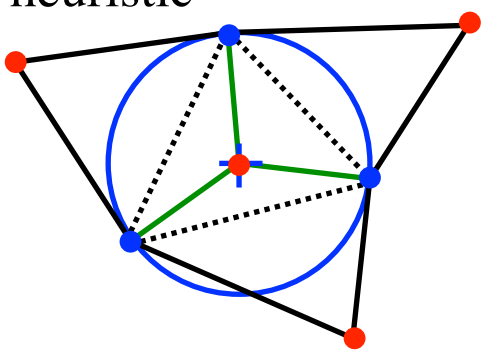
# Closing Thoughts

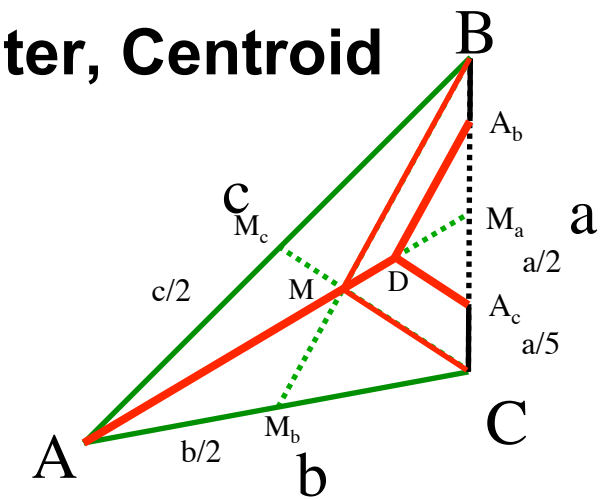
- **Three centers: Circumcenter, Incenter, Centroid**

Incenter for mono-tris template

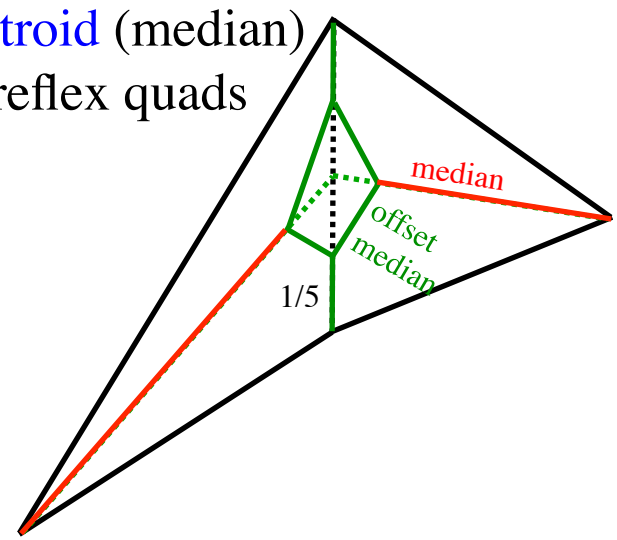


Circumcenter for mono-tris heuristic






Centroid (median) for reflex quads




Orthocenter feeling left out

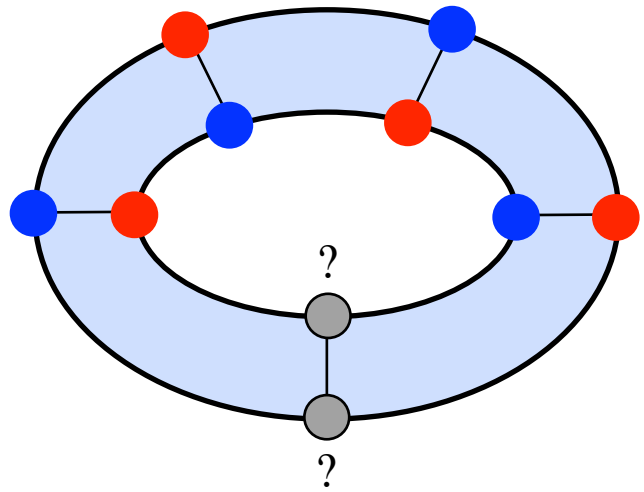
deviantArt Free Ai Vector Smiley Pack
by MathieuBerenguer

Sandia National Laboratories

# Closing Thoughts

- **We create two-colorable quad meshes**
  - **For non-disk domain topologies, 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.

2d, a chain of edges of even length -> 2n-gon

3d, a chain of edges of even length
!= edge cycles bounding hexes

Pairings: combine tets to form hexes?

Closest idea: graphics and meshing literature on 3d cross-fields and connecting hex-duals

# Reasonable Extensions

- Questions
  - Varying-size advancing front?
  - How fast can we vary the size?
    - Bounds for two-radii DT known

# Summary

- Three key ideas
  - Graph theory, two-coloring of vertices
  - Random sphere-packings (better control than Delaunay refine)
  - Advancing front

- 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