Multigrid Marching Tetra Algorithm

A adaptive refinement

Start: A single tetra covers the parts to be meshed.

Refine edges by midnode split (10 possible tetra patterns).

If nodes of one tetra belong to different parts and if B fails,
refine it on 6 edges.

Refine edges such that any tetra is fine
or its predecessor is fine*.

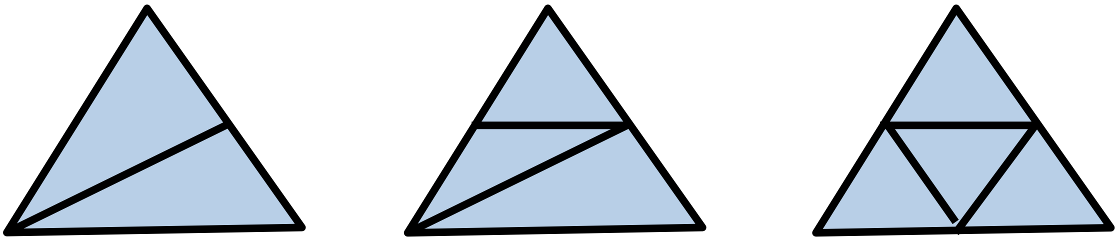
* A tetra is fine if all predecessors are refined by 6 edge splits.

B Geometric refinement of tetras when nodes belong to different parts

Refine edges by geometry splits: bisection, trisection
Refine triangles – finite number of patterns
Refine tetras – finite number of patterns

When the geometric refinement fails, refine this tetra in **A**
or if no refinement is left find a stopgap.

A midnode edge splits for adaptive refinement



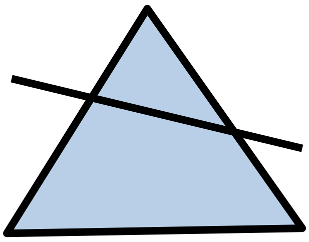
Edge splits	# tetra cases
1	1
2	2
3	3
4	2
5	1

B Geometric edge splits

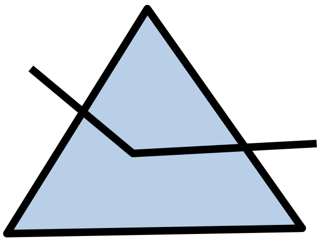
abc-d-e

a edges 0 cuts
b edges 1 cut
c edges 2 cuts
d corners 2D
e corners 3D

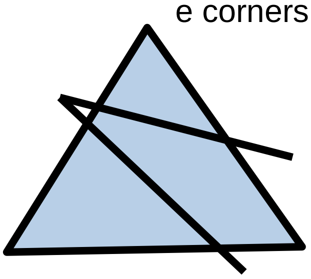
Edge bisection refinement patterns



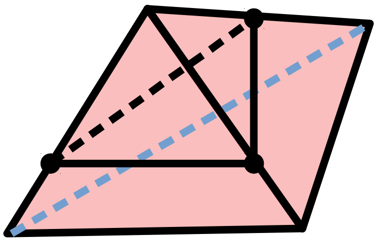
120-0



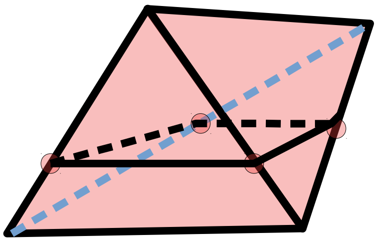
120-1



fails, but 021-0 helps

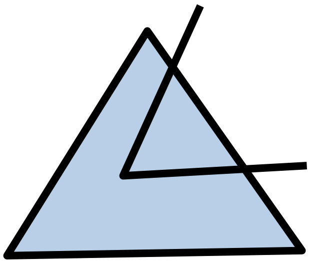


330-0-0

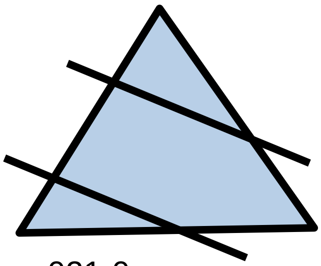


240-0-0

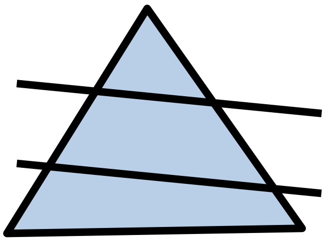
Edge trisection refinement patterns



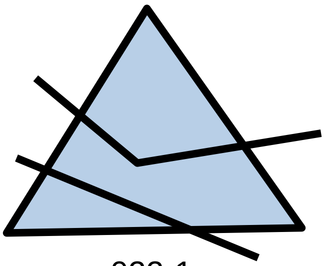
201-1



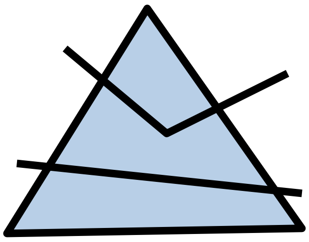
021-0



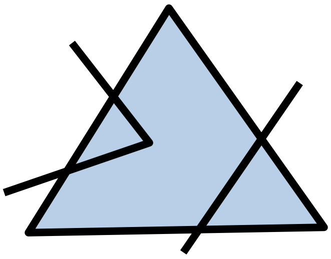
102-0



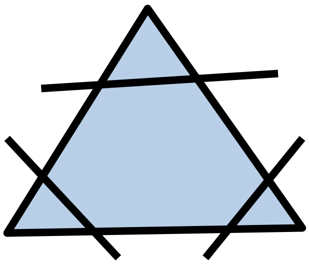
022-1



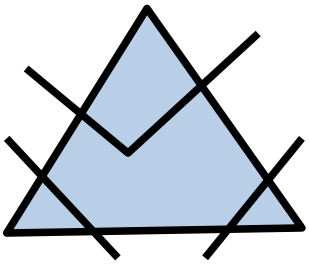
102-1



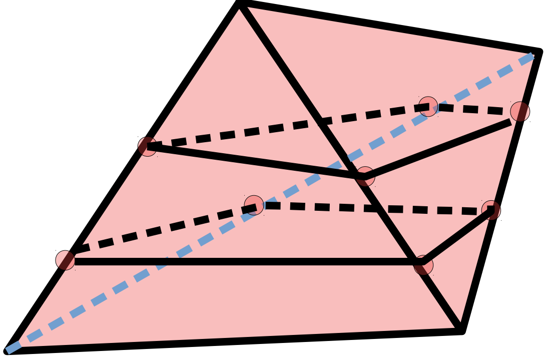
021-1



003-0

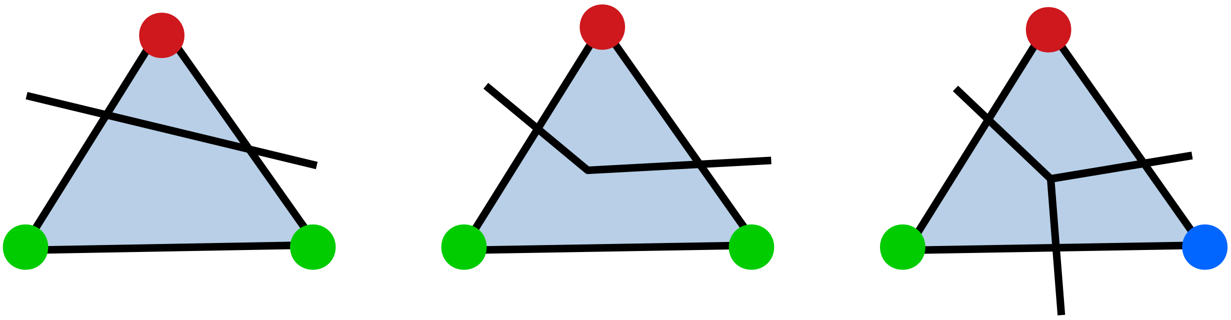


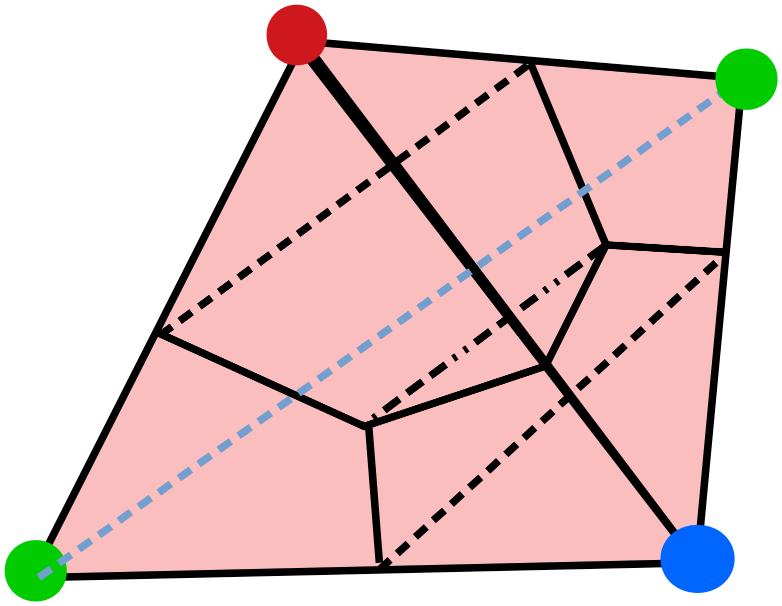
003-1



204-0–0-0

B Geometric edge splits, 3 parts





Geometric edge splits, 4 parts …

5 parts and 1 tetra cannot happen.
Any real geometric situation, where 5 different parts meet at one point will be approximated.