Effective meshing in VSim

Sergey N. Averkin

TWSS 2019

September, 18 2019

Outline

- VSim meshing
  - Functional representation
  - Meshing of triangulated representation of surfaces of geometrical shapes
- Algorithm for meshing of triangulated surfaces
- Problems with CAD and surface triangulated mesh files
- Tips for getting good meshing

Meshing algorithms in VSIM

- Functional representation of geometrical objects
  - Currently available in text-based setup only
- Triangulated surfaces (aka STL files)
  - Default way of meshing in visual setup
  - For primitives in GUI we use openCASCADE to create STL files that are then meshed using TxSim related classes
  - For formats other than STL we create STL files first and then use our meshing algorithm

Functional representation

- How it works
  - We use functions whose value is equal to 1 inside the geometry object and to 0 outside
  - For these reasons we heavily rely on Heaviside step function

$$H(x) = \begin{cases} 0, & x < 0, \\ \frac{1}{2}, & x = 0, \\ 1, & x > 0. \end{cases}$$

  - For example
    - expression = H(z-(0.5))*H((0.4)-z)*H(0.4^2-x^2-y^2)

Surface triangulation

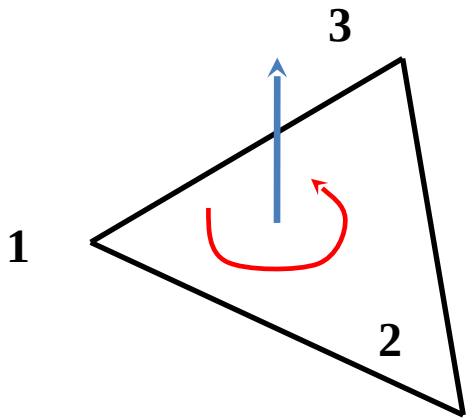
- We use many input formats that are finally translated into STL-like format using OpenCASCADE
- STL = an abbreviation of "stereolithography"
  - It describes surface triangulation of geometrical objects
  - Example:

```
solid TxsGridRgn_Polygons
  facet normal -6.929152621777728e-01 -2.227836250153121e-01 -6.857374832004920e-01
    outer loop
      vertex -4.978670775890350e-01 -4.957432746887207e-01 -7.049153413725920e-01
      vertex -7.006597394512872e-01 -4.957432746887207e-01 -5.000000000000000e-01
      vertex -6.394899460425015e-01 -2.478716373443604e-01 -6.423390550093767e-01
    endloop
  endfacet
endsolid
```

  - It has no connectivity information. Surface orientation is determined by the normals of the triangles.

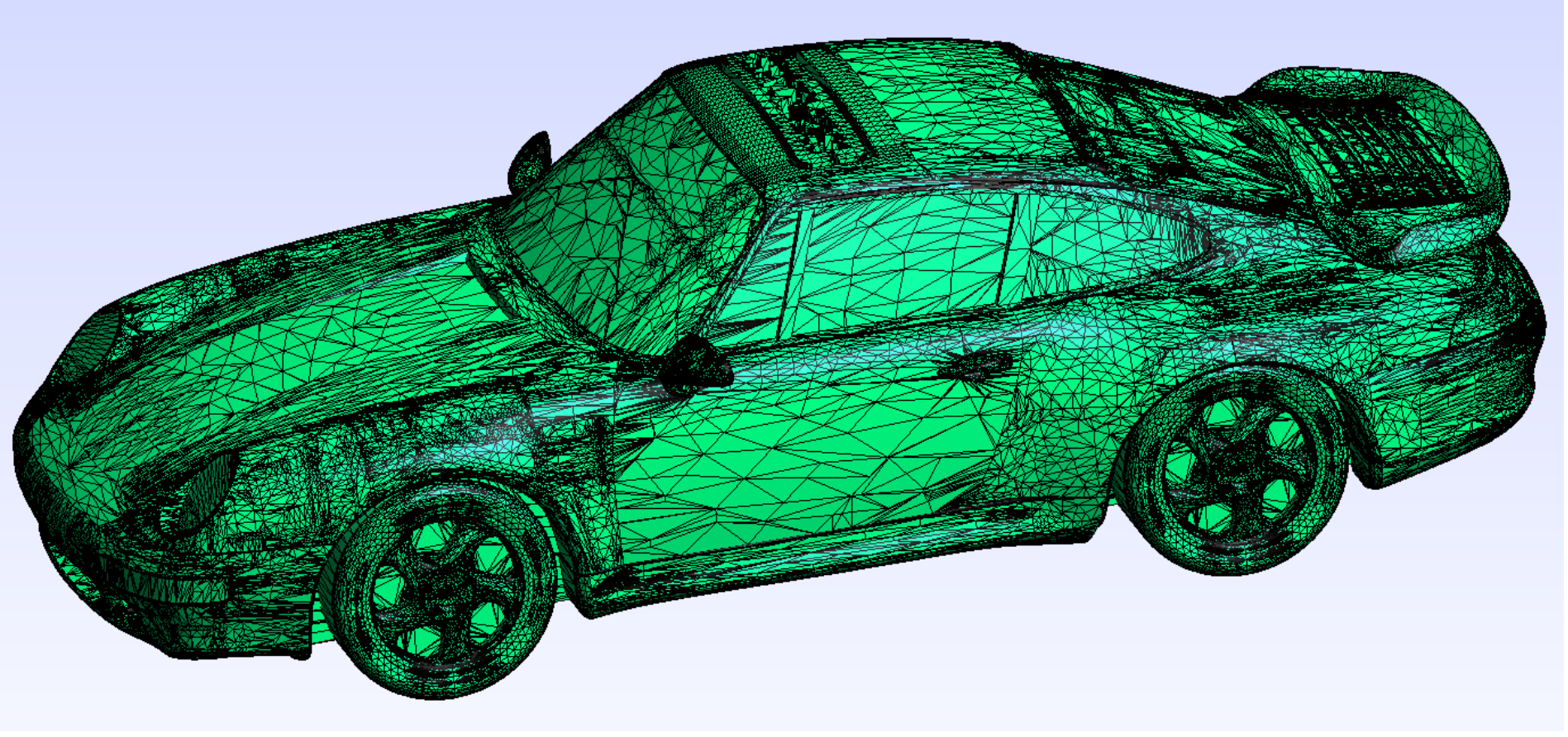
Surface triangulation

- Triangle normal is determined using two different ways:
  - Through **facet normal** in STL file
  - Through vertices ordering

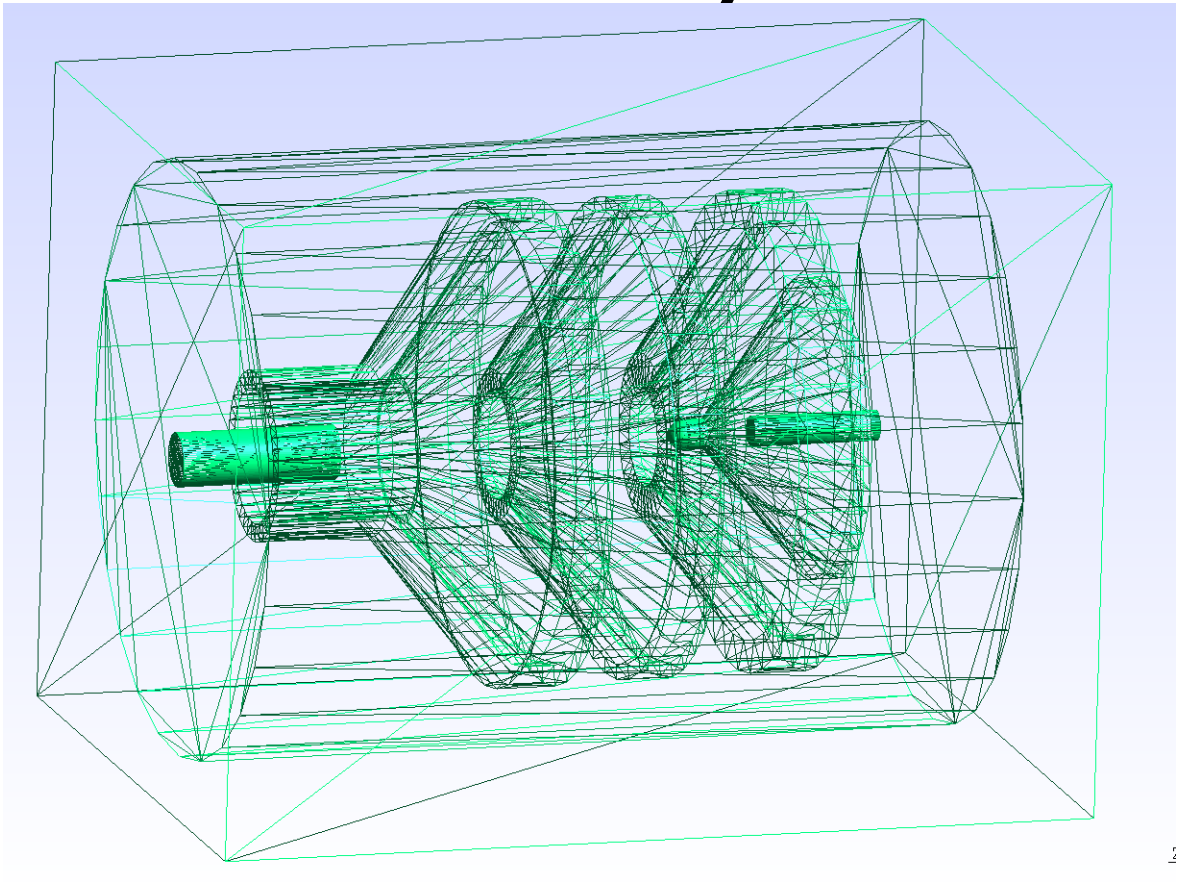


  - Some CAD software output garbage in the STL file for **facet normal**. Therefore, we always calculate these normals using vertices.

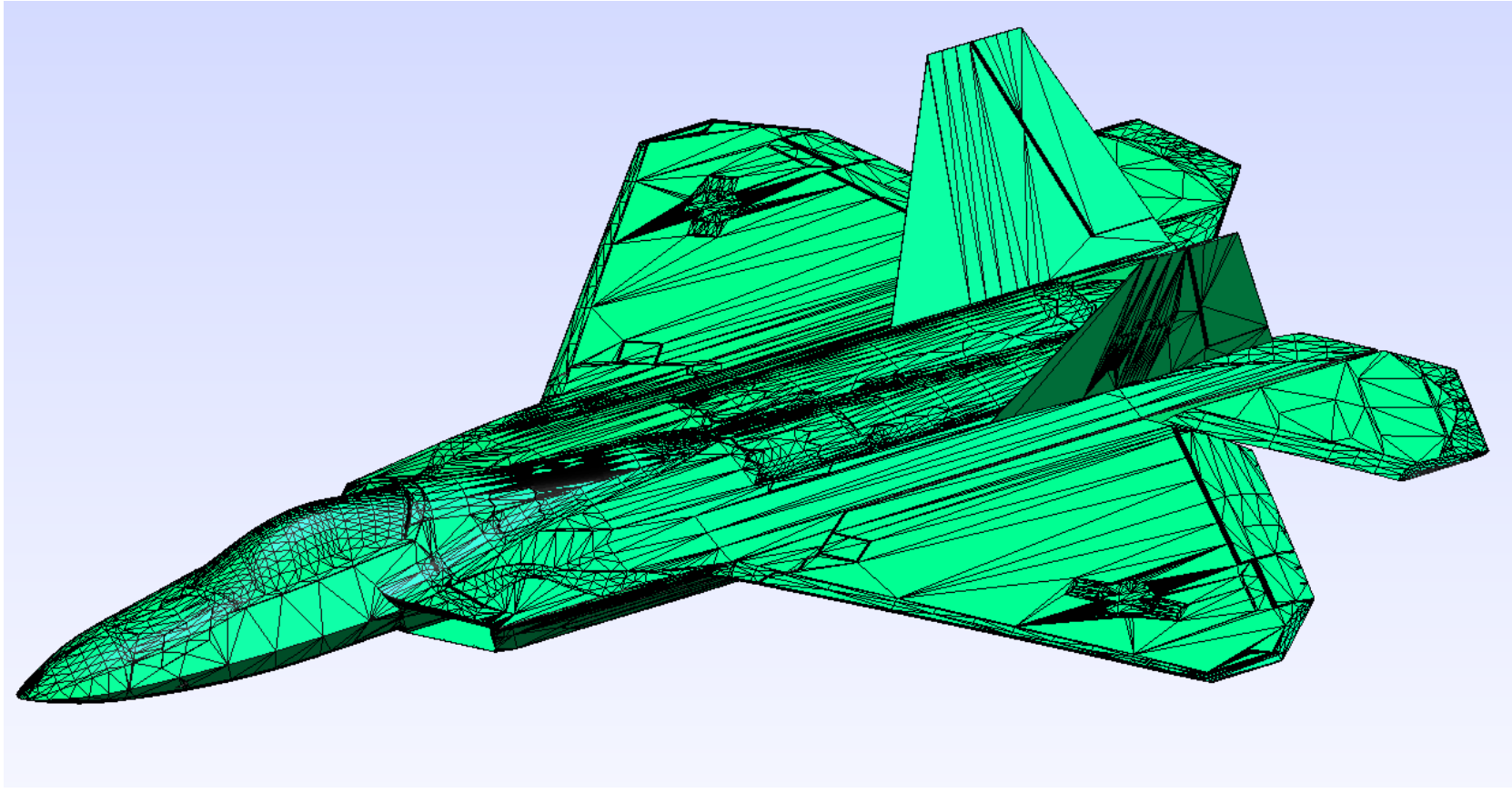
Surface triangulation examples (Porsche)



Surface triangulation examples (multistage Collector)



Surface triangulation examples (F-22 raptor)

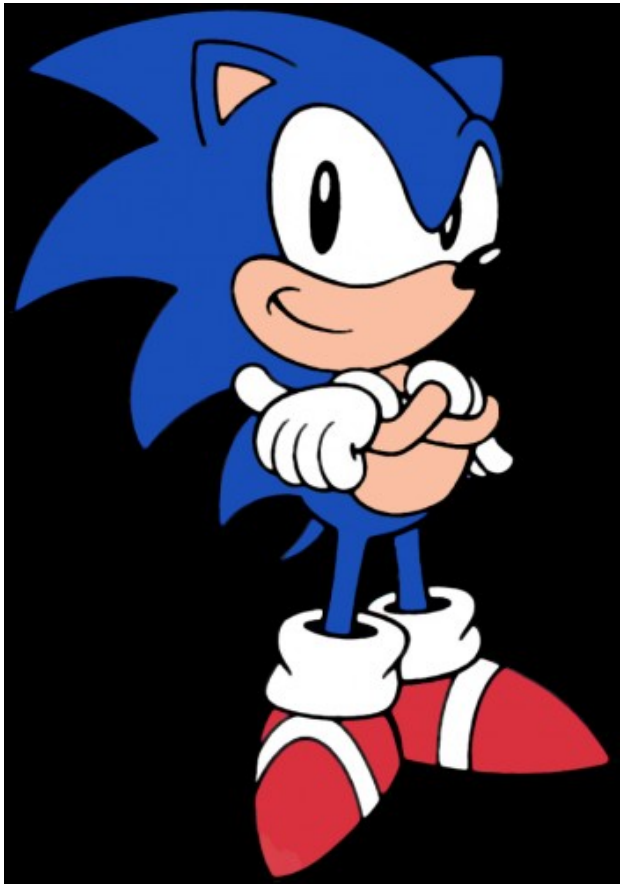


Meshing STL files

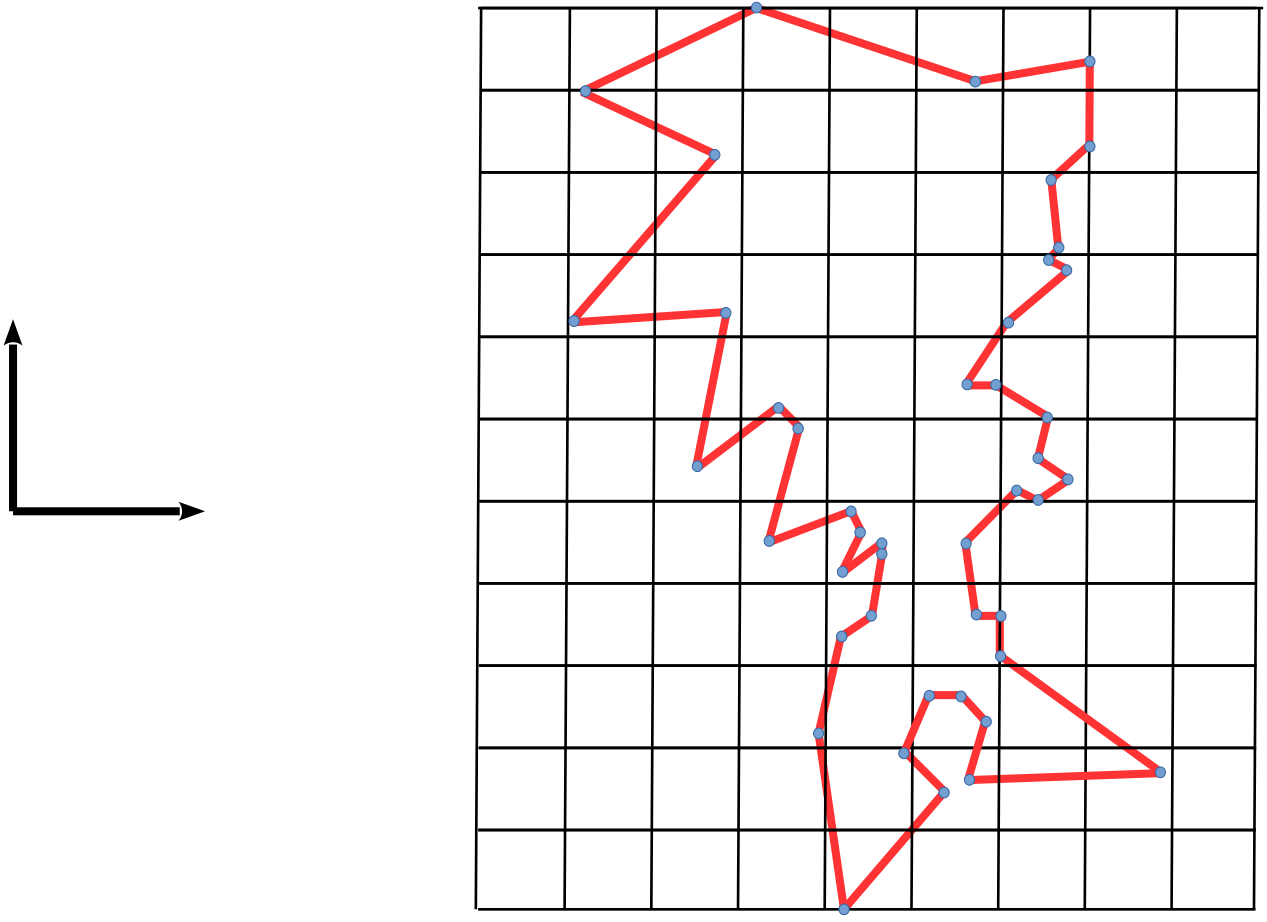
- Find edge cuts
- Process edge cuts
  - Eliminate repeated edge cuts using tolerance propagation analysis
  - Calculate consistent in/out flag for each node based on the signs of edge fractions
- Post-process edge cuts
  - For each edge cut use information from neighbor nodal in/out to check whether edge cut exists
  - If there are two different nodal in/out values and there is no edge cuts, we insert new one at the middle to keep consistency
- Reconstruct cut cells

VSim meshing example

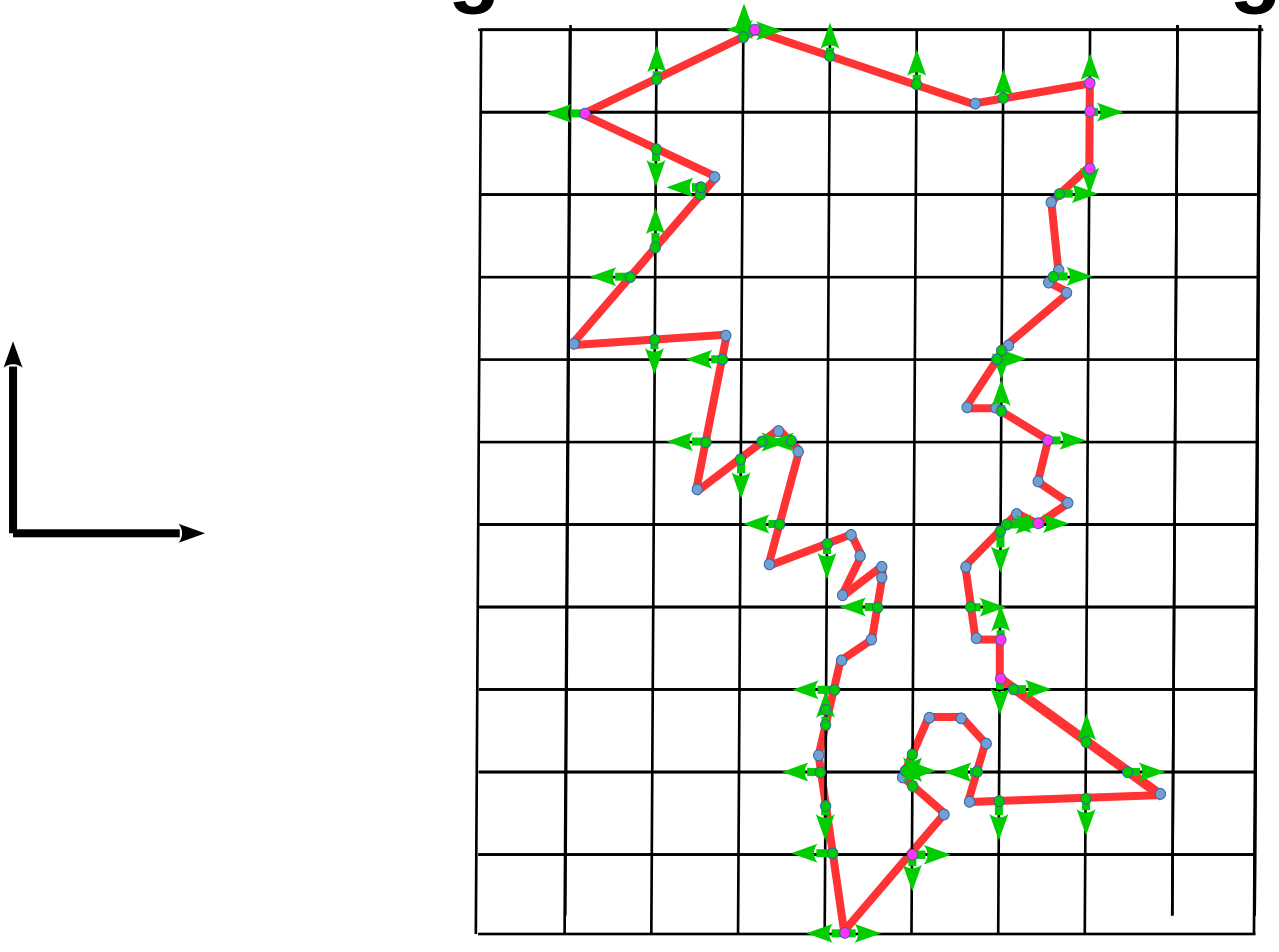
Meshing Sonic in 2D



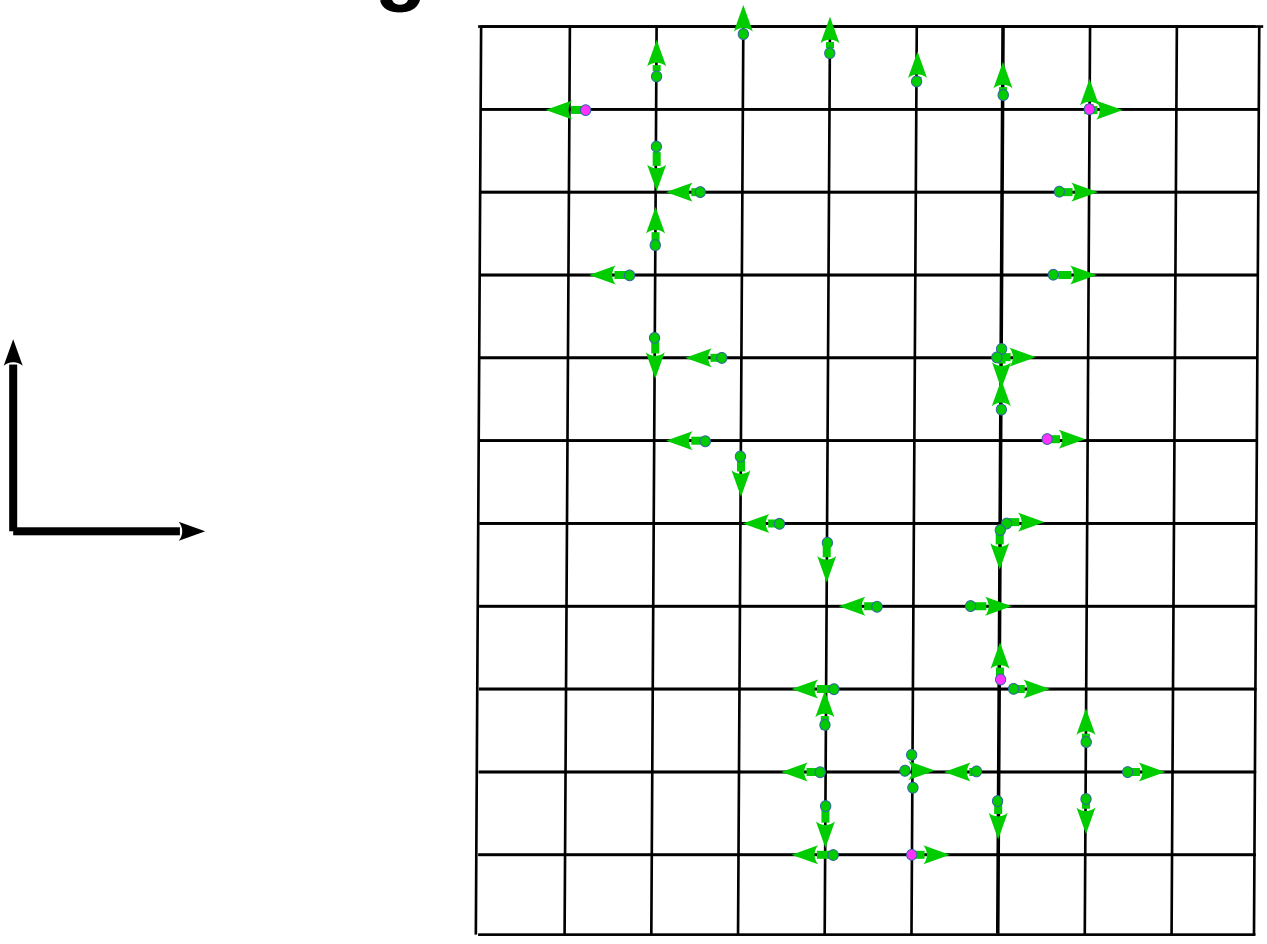
Meshing Sonic: Initial “STL” model



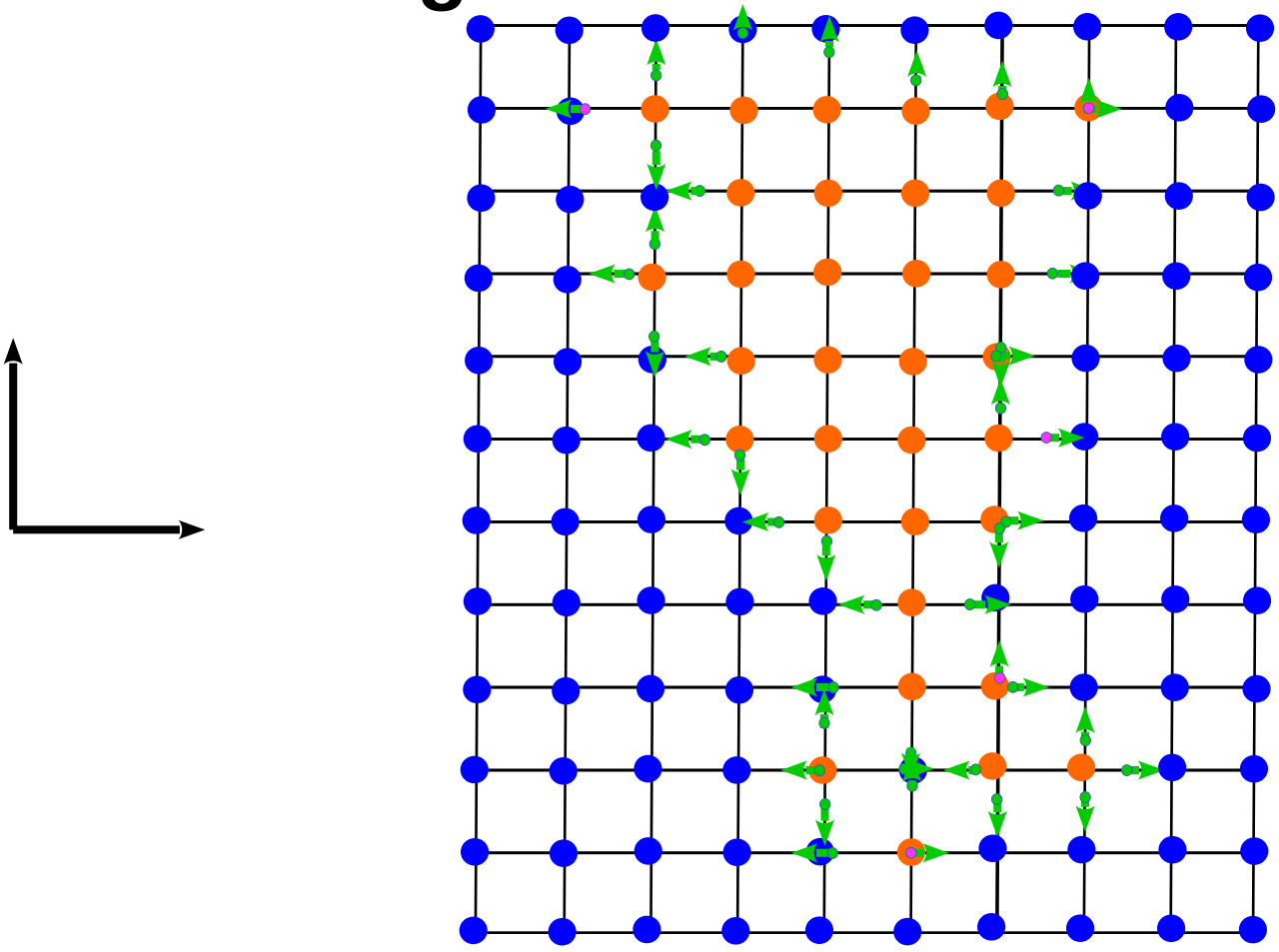
Meshing Sonic: Find all edge cuts



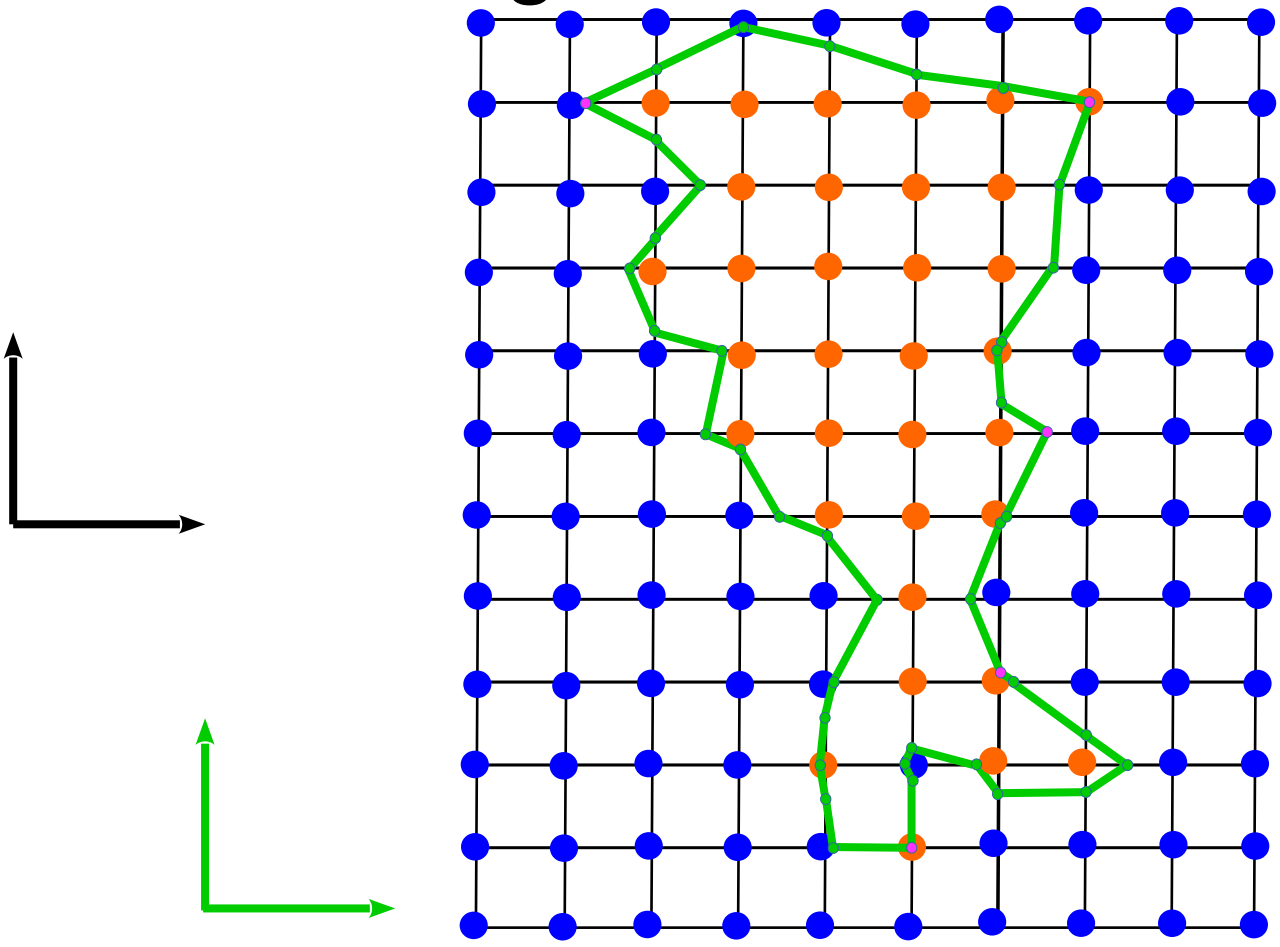
Meshing Sonic: Remove excessive cuts



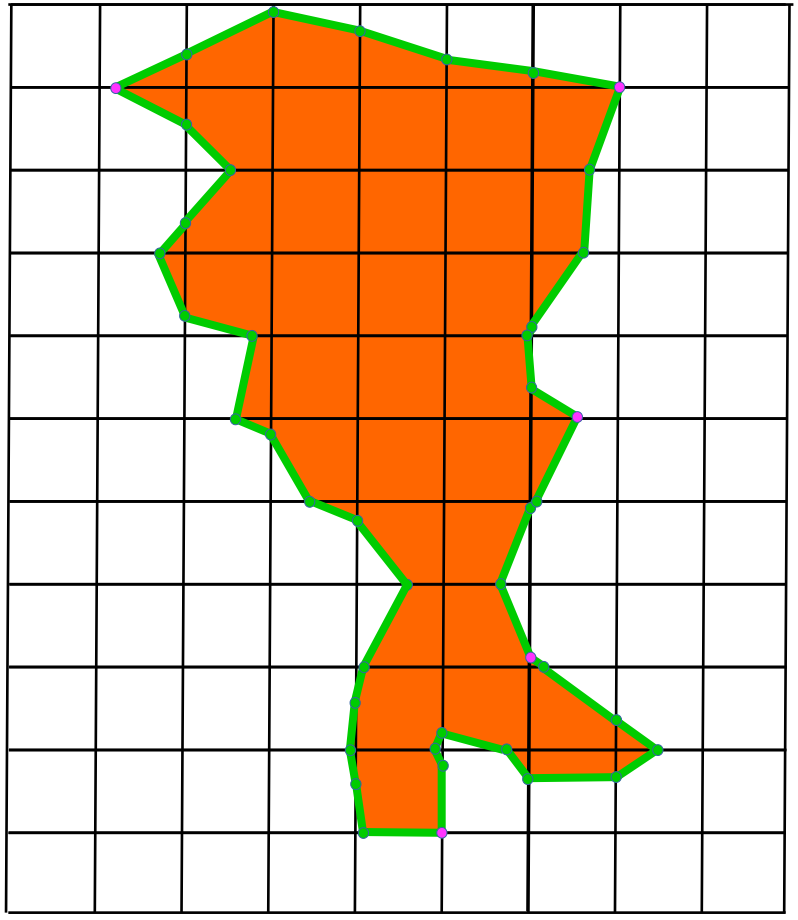
Meshing Sonic: Calculate in/out flags

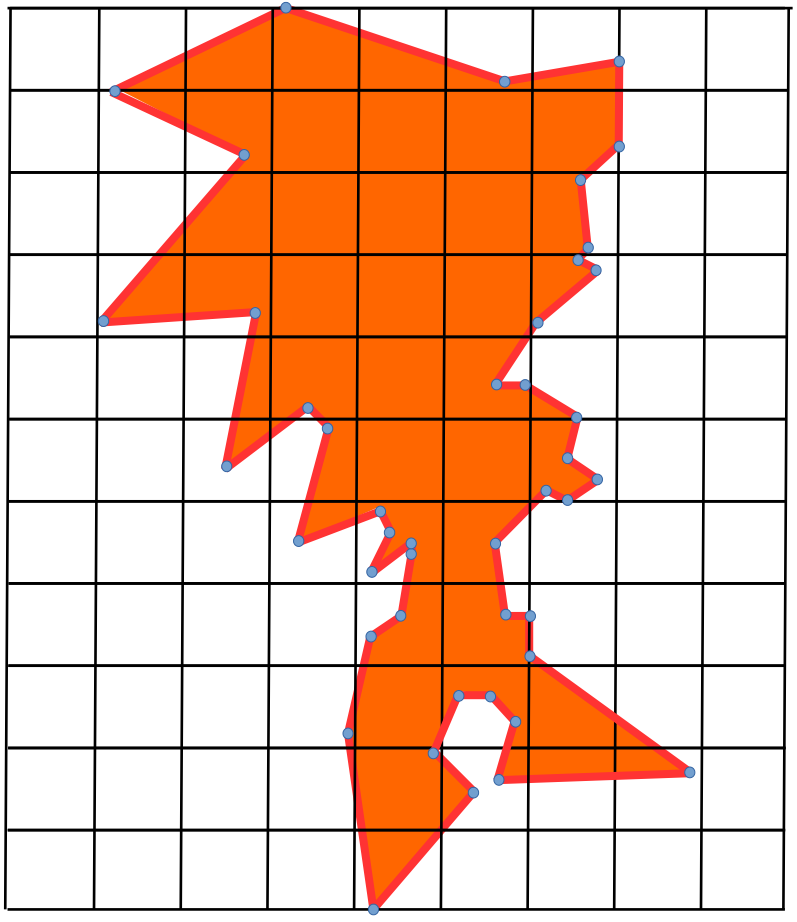


Meshing Sonic: Restore cut cells



Meshing Sonic: Final mesh and STL



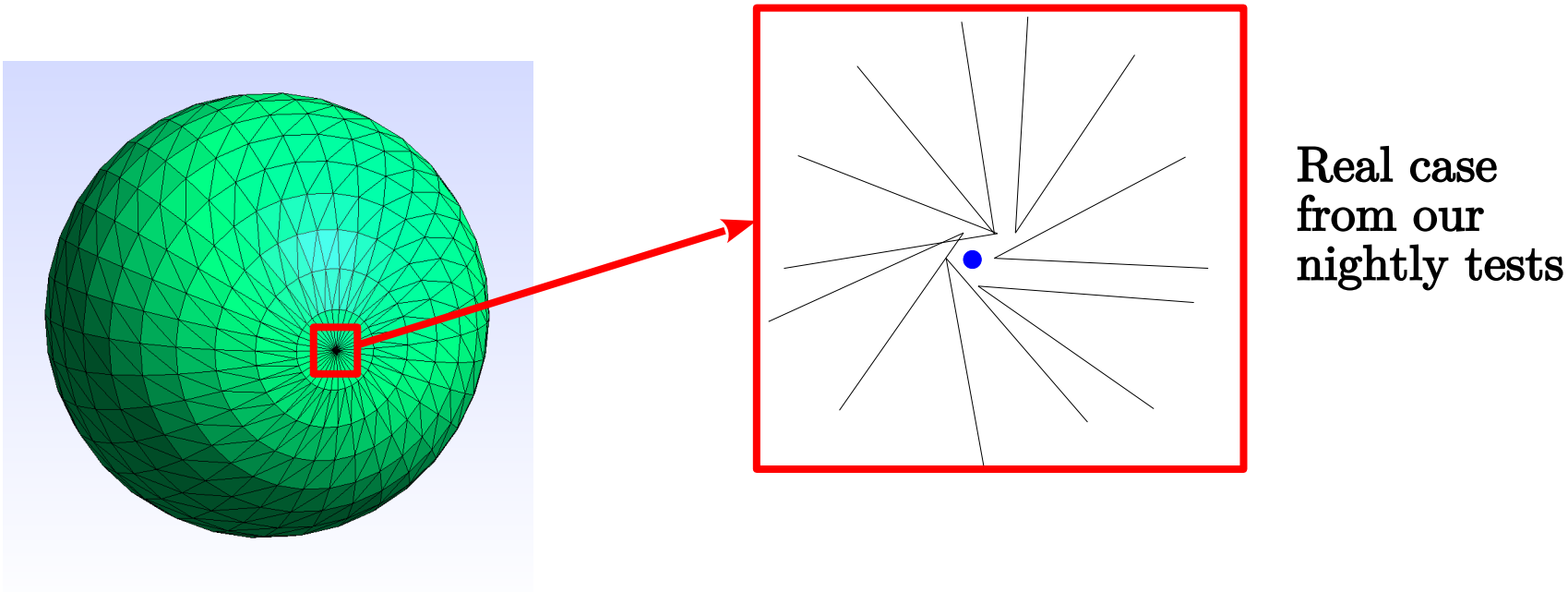


What might go wrong?

- Valid STL
  - Precision problems
  - Complicated nodal intersections
  - Step-like geometrical objects
  - Bugs in meshing
- Invalid STL files
  - Problems with surface normals
  - Holes
  - Intersections

Precision problems with STL files

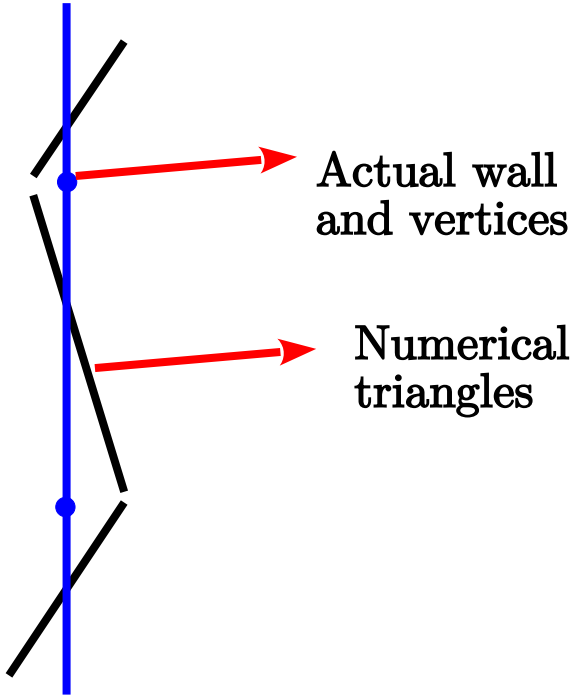
- Binary STL format specifies that all vertices are in **float** format
- In some cases STL files have round-off problems
  - The same vertices that belong to different triangles have different coordinates



Real case from our nightly tests

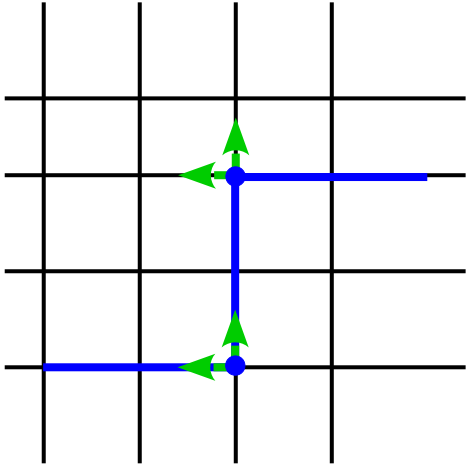
Precision problems with STL files (cont'd)

- Flat surface with round-off triangles could be non-flat



Step-like geometries

- If we have a step-like geometry aligned with grid planes

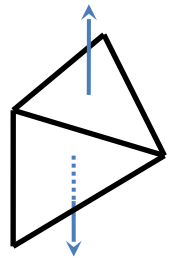


Repeated edge cut signs
are the same as for incorrect
STL files with self-intersections

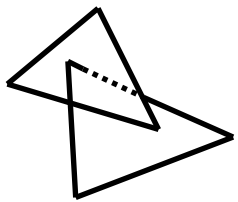
Incorrect STL files

Common problems in STL files leading to mesh corruptions

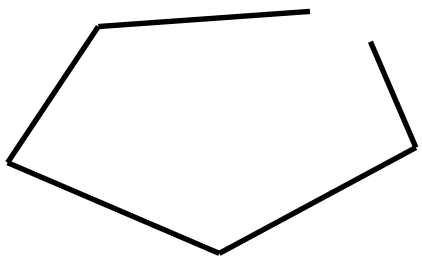
- Mismatched surface normals



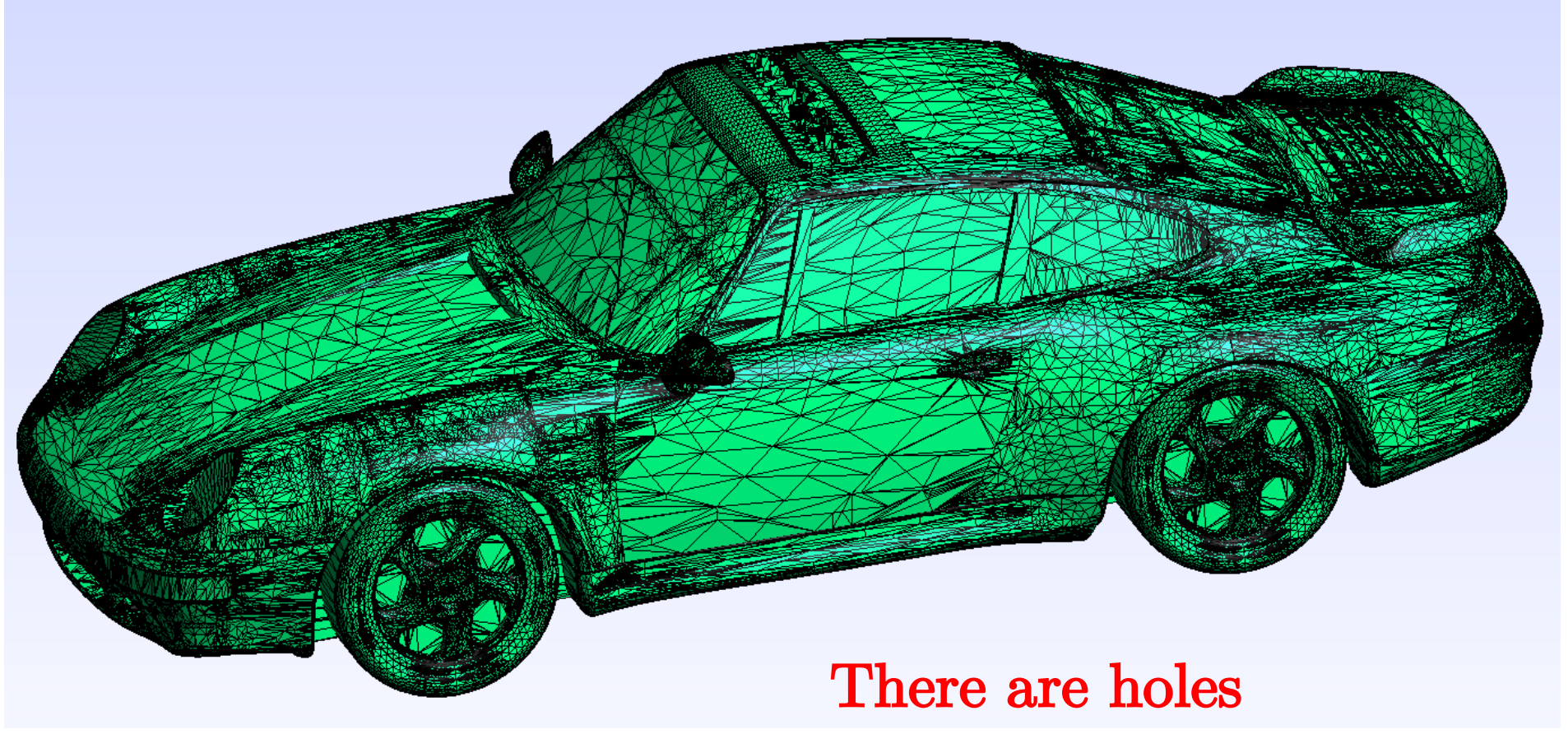
- Surfaces intersections



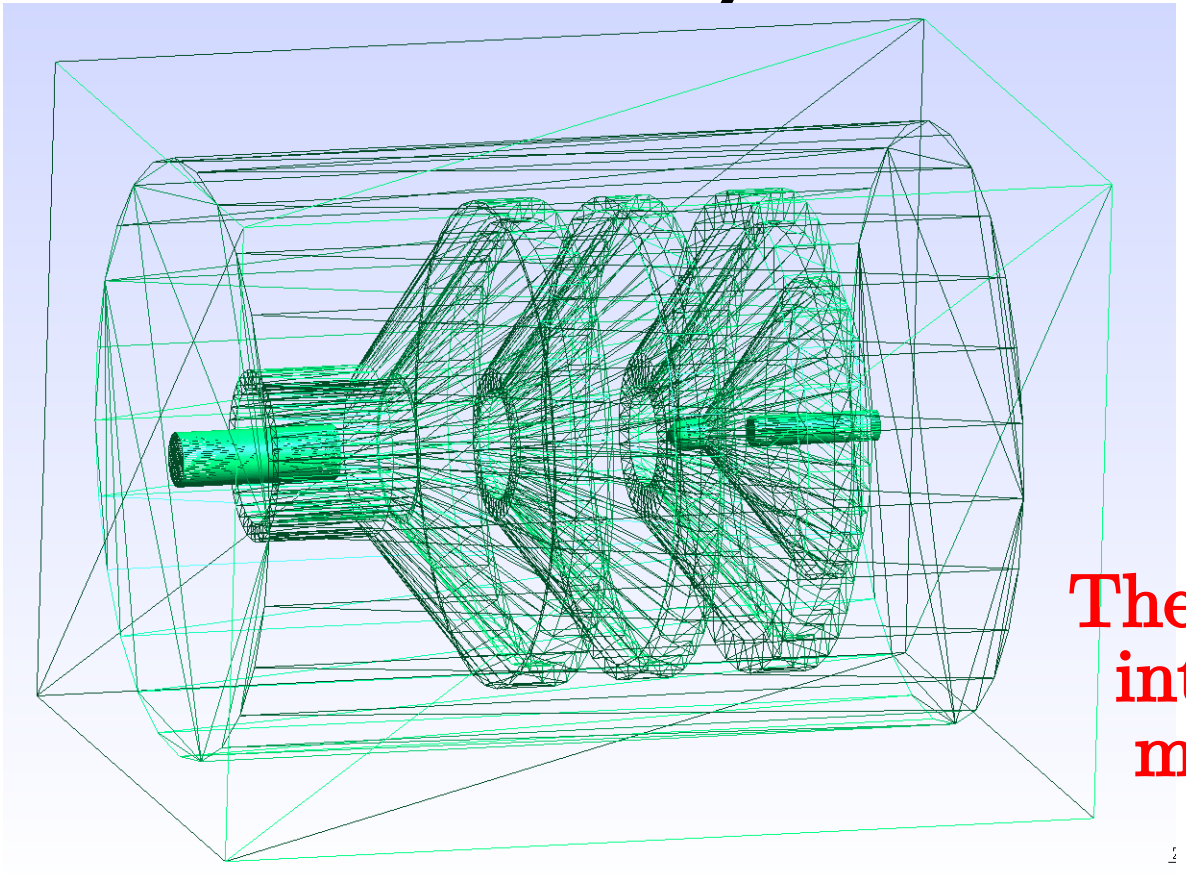
- Holes in surfaces



Surface triangulation examples (Porsche)



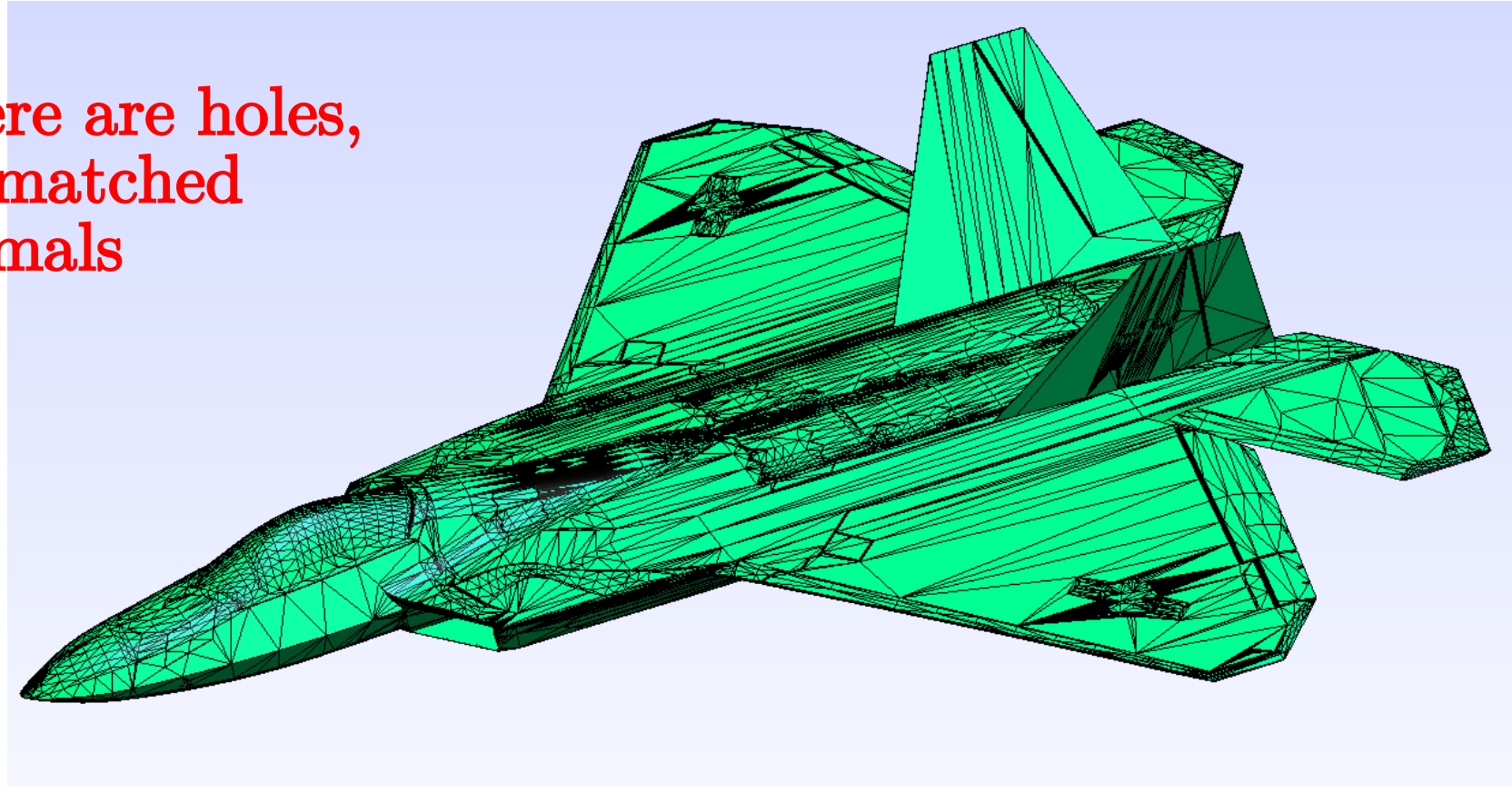
Surface triangulation examples (multistage Collector)



There are holes, intersections, mismatched normals

Surface triangulation examples (F-22 raptor)

There are holes, mismatched normals



How to detect problems with STL files?

- GMSH (free meshing tool)
  - Visual inspection of STL file
    - Check normals to see if there are any mismatches
    - Check surfaces if there are intersections
    - Check gmsh output to see if there are duplicate triangles
- Tech-X Software
  - We have a number of tools that allow to detect certain problems. We do not yet distribute them. We can identify
    - Holes
    - Mismatches surfaces (they are treated as holes)
    - Precision problems
    - Output Euler characteristic of the surfaces (their “openness”)

gmsh

- Free, supports multiple OSes
- Can detect duplicate triangles
- Can visualize normals
- Limitations
  - Doesn't detect holes automatically but they can be found visually
  - Doesn't detect surfaces intersections but they can be found visually
  - Normals mismatch can only be found visually

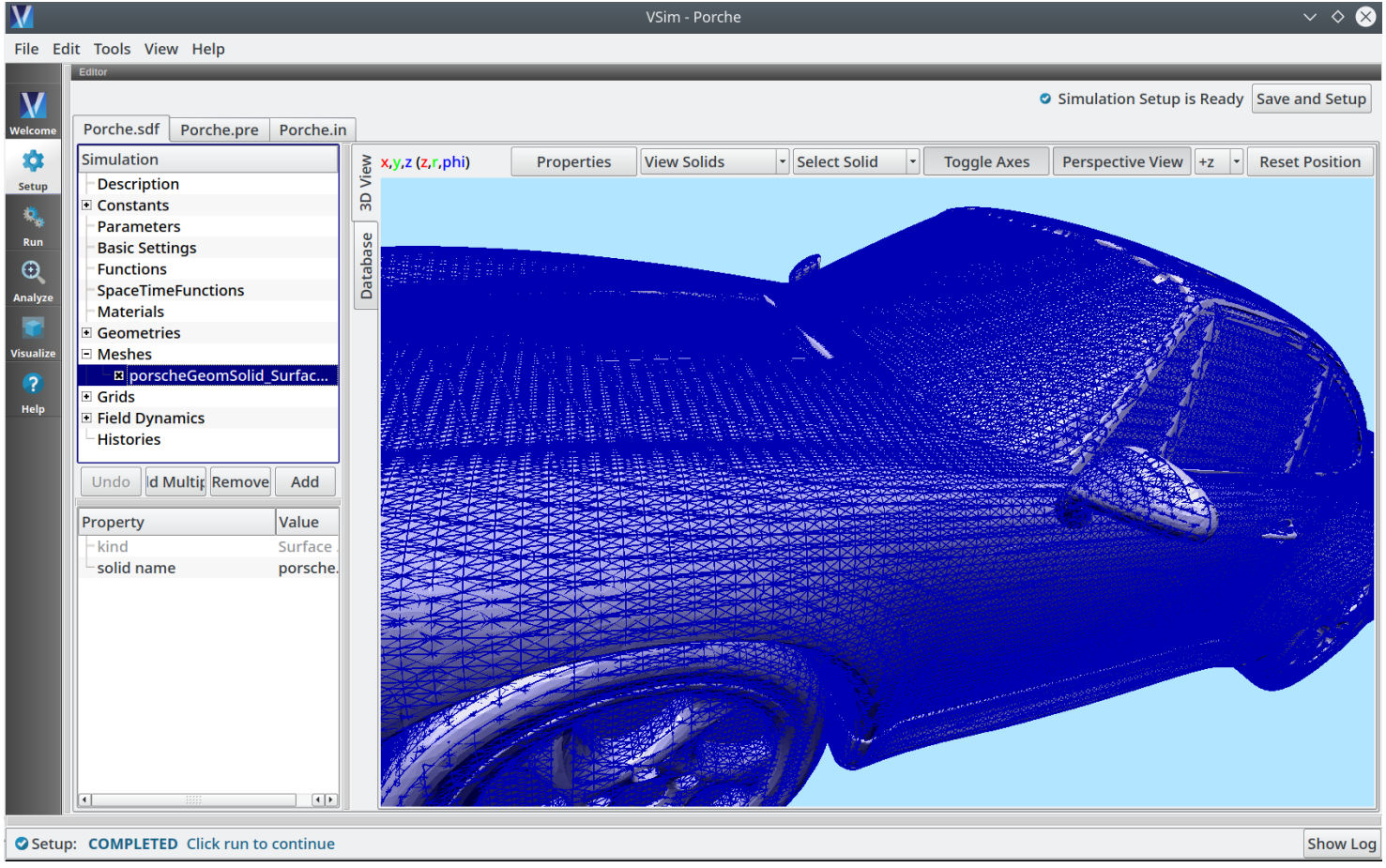
Reparing STL files

- Mismatched normals
  - Need to decompose STL file into a set of surfaces, fix normals using gmsh and merge them back
- Intersecting surfaces
  - Need to decompose STL file into a set of surfaces and use union set operation in order to create a union of files. Union can be done in VSim.
- Holes
  - In simple cases the algorithm in Vsim can take care about them but it is not guaranteed since we rely on the assumption of valid STL files. In more complicated cases it can be done in something like blender

General meshing tricks

- Avoid aligning geometrical objects with grid lines or planes by jiggling geometry
- If there is a problem
  - Check STL files
    - If there are open surfaces then this is most likely incorrect STL file and we need to either fix it or create a new one
  - Merge close points to avoid precision problems
  - If a problem occurs when meshing in parallel (like an isolated island)
    - the problem is potentially with empty domain
    - Try changing decomp or running in parallel
      - If it fixes it then file a bug
- If possible use functional representation

Meshing in Visual Setup (surface mesh)



Questions?