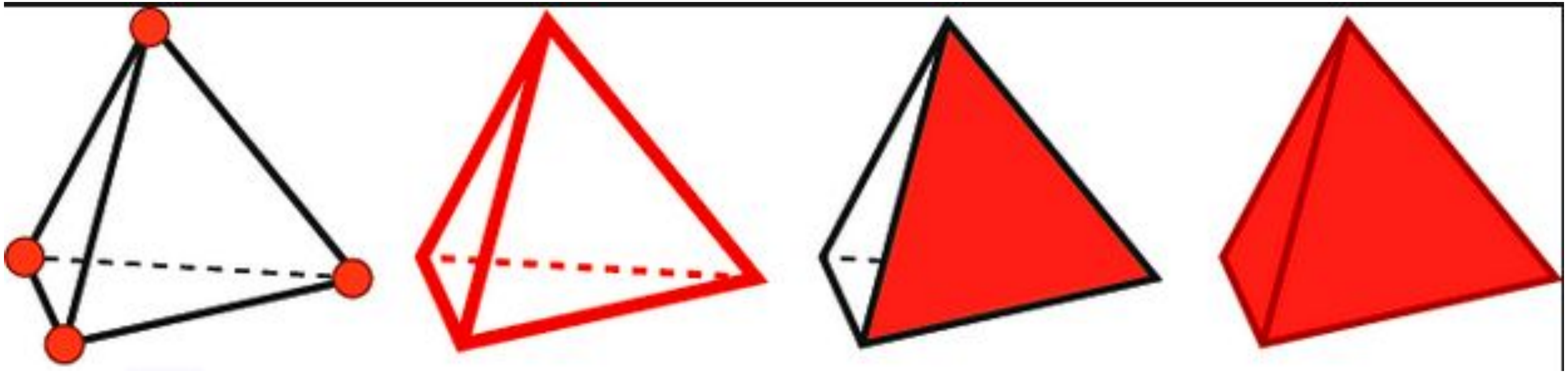
Scientific visualisation, algorithms in the real world

INSAlgo training session - Louis Gombert 05-30-23

$ whoami

- Louis Gombert
- 5 TC
- INSAlgo board 2020-2021
- Cod’INSA current president (did you enjoy this week-end ? :D)
- Intern at Kitware Europe
  - Working with the DoE’s Exascale Computing Project on VTK-m
  - Massive multithreading on heterogeneous architectures for mesh processing

3D Geometry basics



Vertices

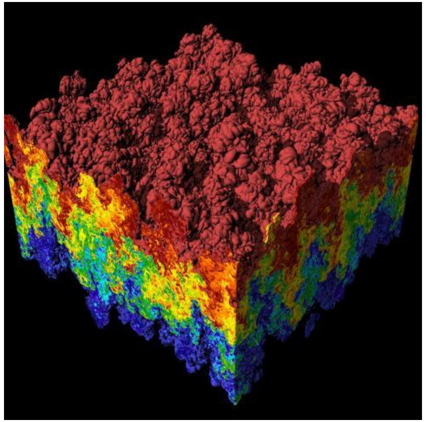
Edges

Face

Cell

Scientific visualization (sciviz) in a nutshell

- Find a way to visualize data from an experiment or a simulation
- In our case, data associated to 3D meshes
- A real use-case of plenty of smart algorithms, that need to be optimized
- How to represent 3D scalar data on a 2D screen ?
    - Slices
    - Isocontours
    - Streamlines
    - Many more…



Isosurfaces

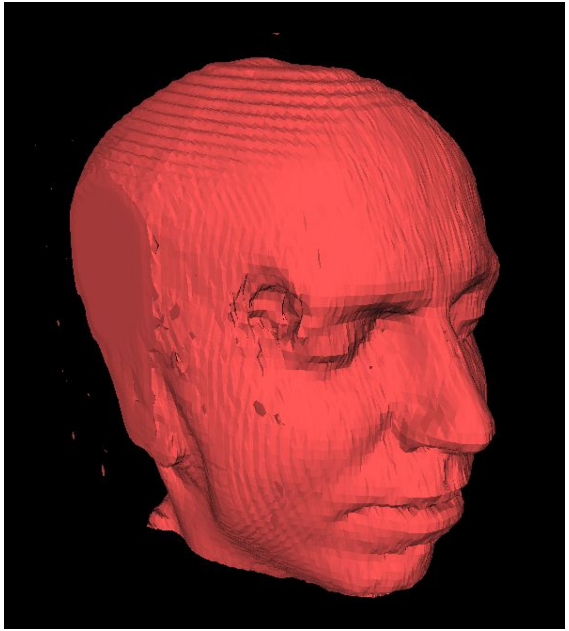
Surfaces where the variable attached to vertices is constant

= level set

Applications : CFD, medical imaging (CT), astrophysics…

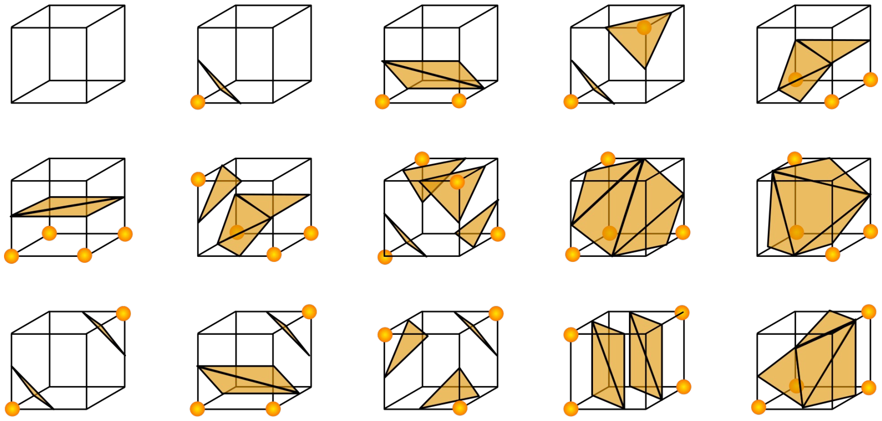
Usually we only use the 2-manifold, closed surface (contour)

Any idea how to compute them in a regular 3D grid ?



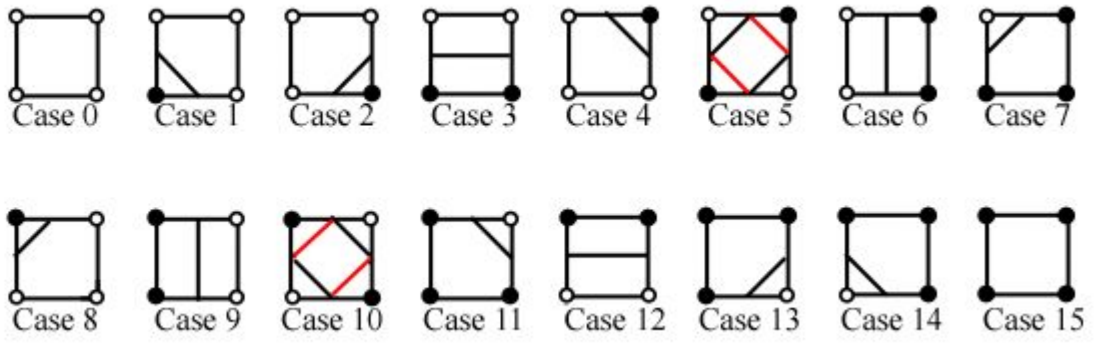
Marching cubes algorithm

Published in 1987 (patented) and improved over time



- Before : pre-calculate all 256 possibilities
- For each cube in the mesh :
  - Calculate which points are over the threshold
  - Generate the right triangles from pre-computed data
  - Interpolate coordinates
- Assemble the isosurface from the triangles
- Trim to only keep the closed surface

Implementation time ! (in 2D)



Fill all the TODOs : https://gist.github.com/Lgt2x/6395df29cff113a782f374a4abc502a4

Complete version : https://gist.github.com/Lgt2x/14ef956d5af2d5429b065332bc213401

Improving the algorithm...

- What if we don’t have a structured, regular mesh ?
  - Marching Tetrahedron
- What about parallelism ?
  - Marching cubes is **embarrassingly parallel** : the computation of each cell is **independent**
  - Remember Amdahl’s law ? The speedup is limited by the serial portion
  - But how to allocate the memory ? We may need several passes
- How many times do we go through each point ?
  - 8 times (for an inner point)
  - Can we do less ? (yes : see Flying Edges)

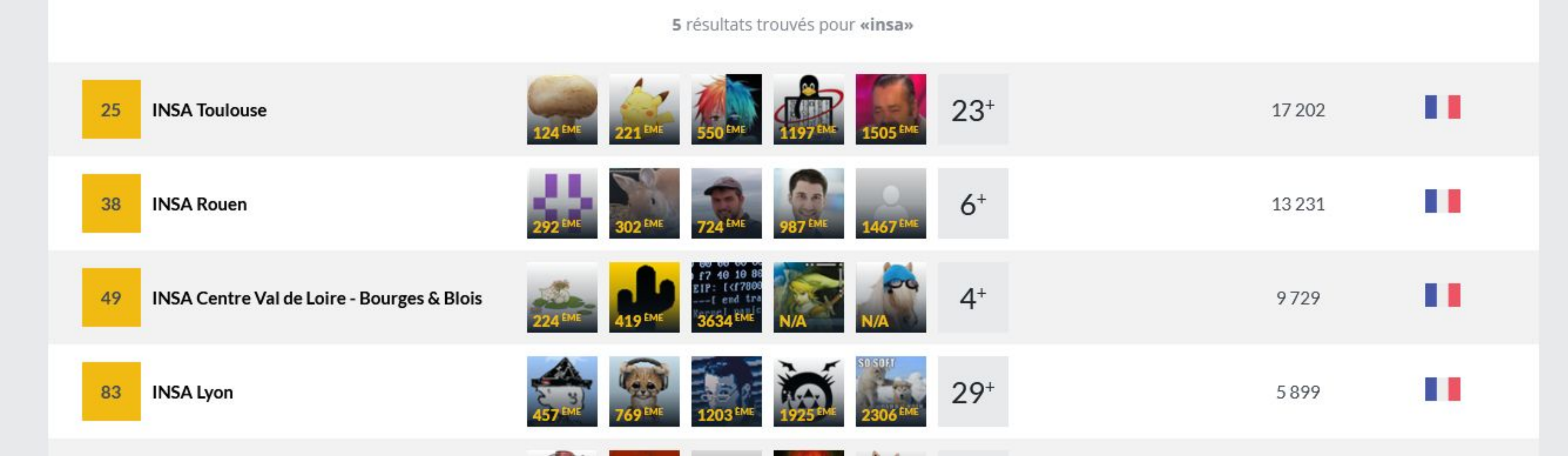
How does a real implementation look like ?

VTK is an open-source visualisation library, one of the first to feature Marching Tet (and Flying Edges)

Let’s look at the (2D) implementation together !

https://gitlab.kitware.com/vtk/vtk/-/blob/master/Filters/Core/vtkMarchingSquares.cxx

Homework - Beat the other INSA at the Codingame Spring challenge !



5 résultats trouvés pour **«insa»**

25	INSA Toulouse	124 ÈME	221 ÈME	550 ÈME	1197 ÈME	1505 ÈME	23^{+}	17 202	🇫🇷
38	INSA Rouen	292 ÈME	302 ÈME	724 ÈME	987 ÈME	1467 ÈME	6^{+}	13 231	🇫🇷
49	INSA Centre Val de Loire - Bourges & Blois	224 ÈME	419 ÈME	3634 ÈME	N/A	N/A	4^{+}	9 729	🇫🇷
83	INSA Lyon	457 ÈME	769 ÈME	1203 ÈME	1925 ÈME	2306 ÈME	29^{+}	5 899	🇫🇷