Spatial Data

Cmpt 767 Visualization

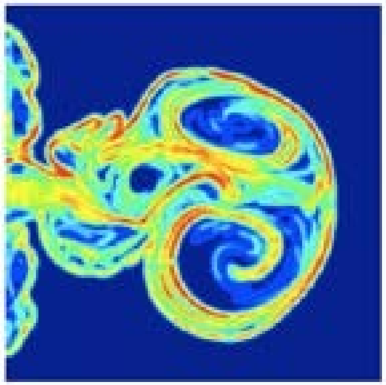
Steven Bergner
sbergner@sfu.ca

[Telea / Möller]

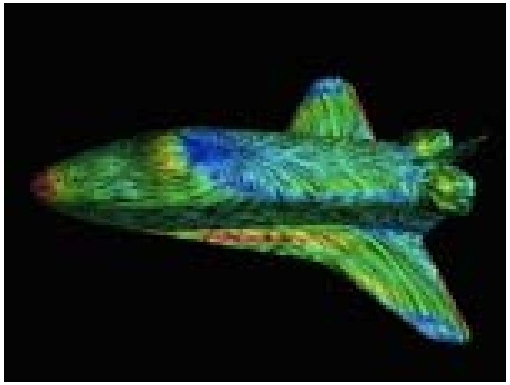
Overview

- Spatial data vis examples
- Function plots and height fields
- Isolines

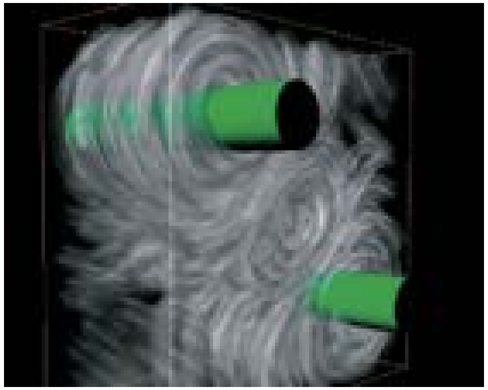
Visualization examples: Fluid flow



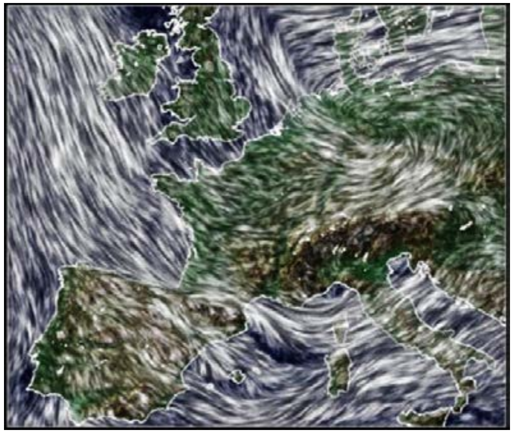
mixing of substances
(chemistry)



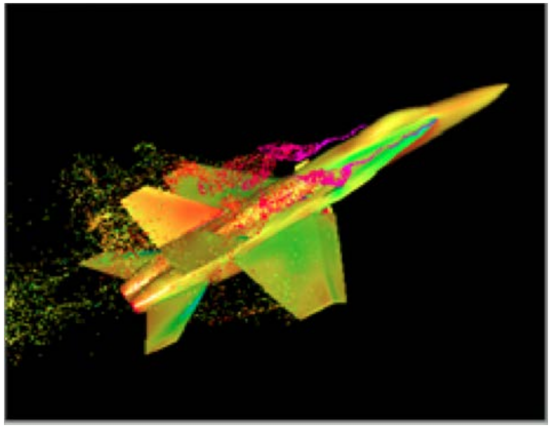
flow on surface
(aircraft design)



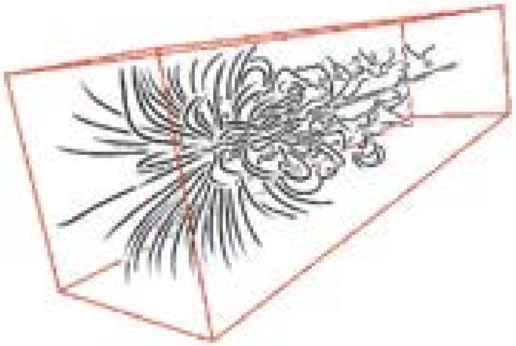
flow in volume
(engine design)



wind flow atop geo map
(weather forecast)

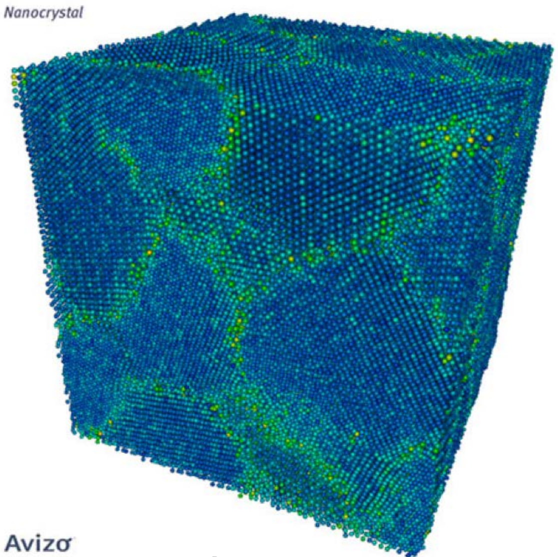


particle flow close to surface
(aircraft design 2)

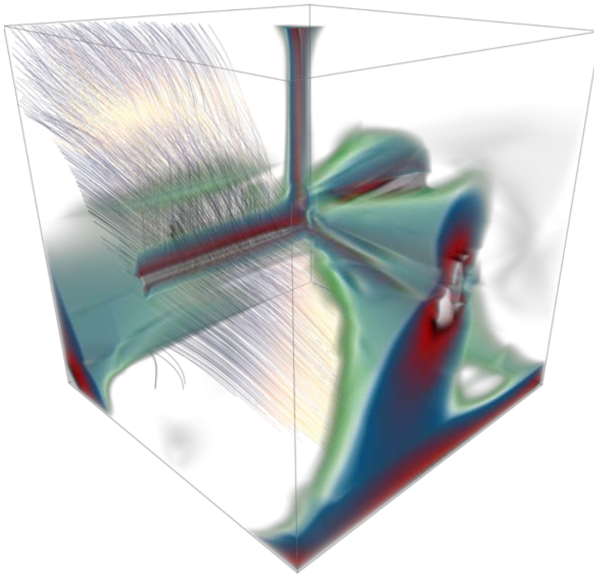


sketch of flow in volume
(illustrative/communication)

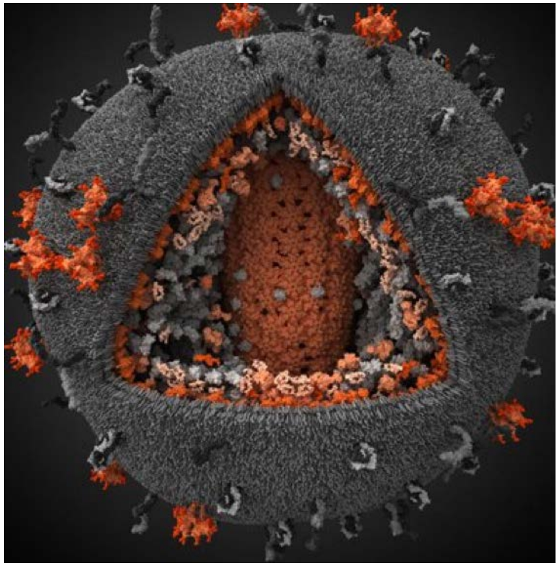
Visualization examples: Material/biosciences



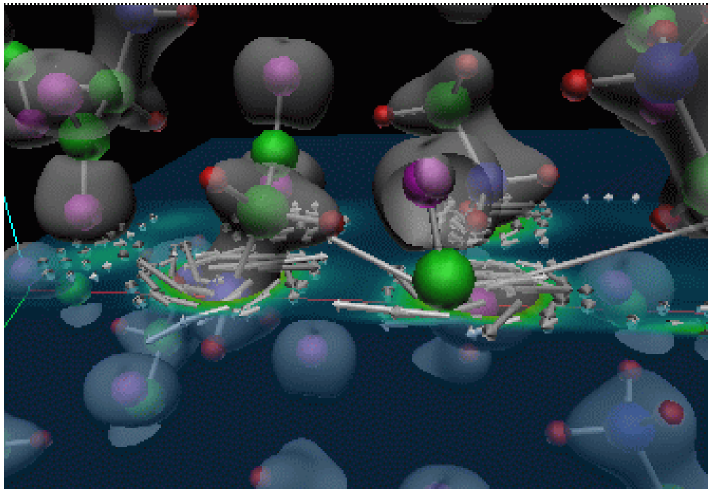
atoms in crystal
(crystallography)



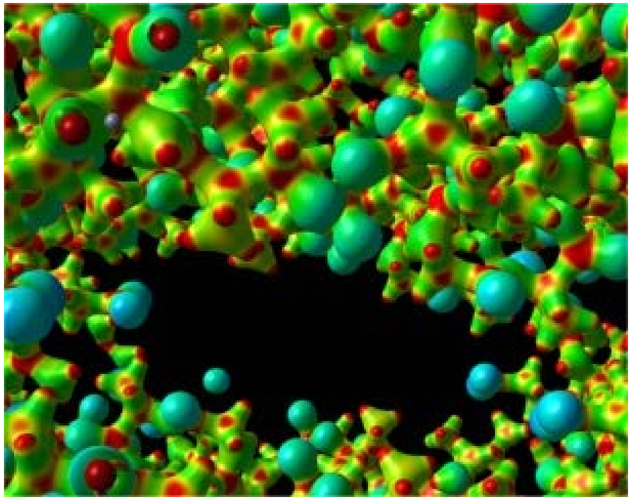
stress tensors



3D HIV model

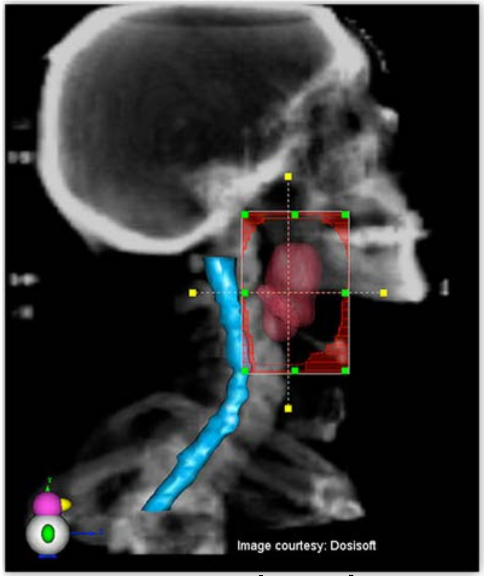


glycine crystal simulation

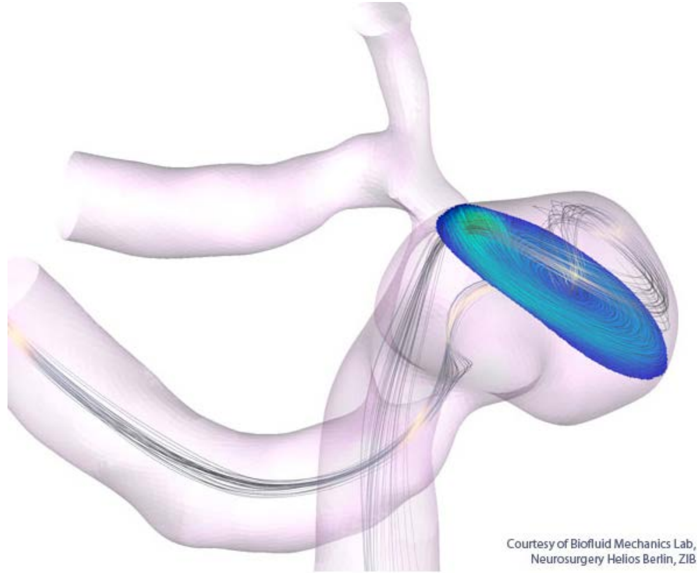


potential field in crystal structure

Visualization examples: Medical sciences

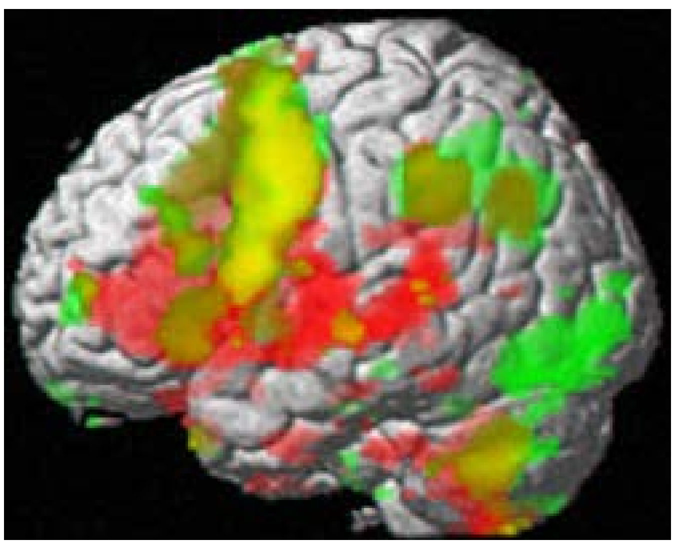


surgery planning

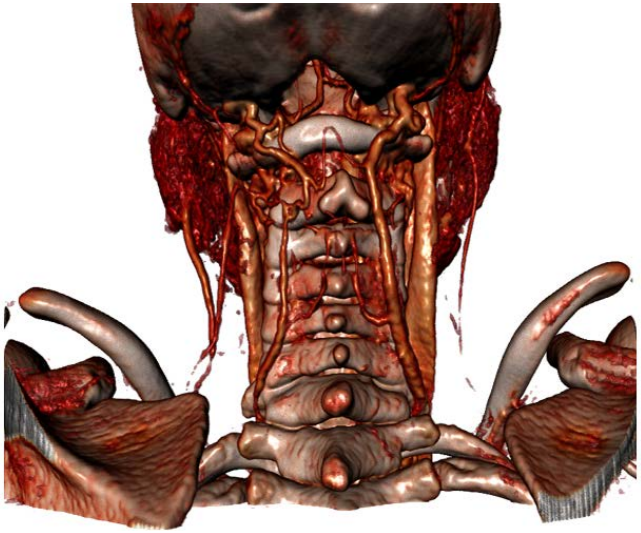


blood flow in aneurysm

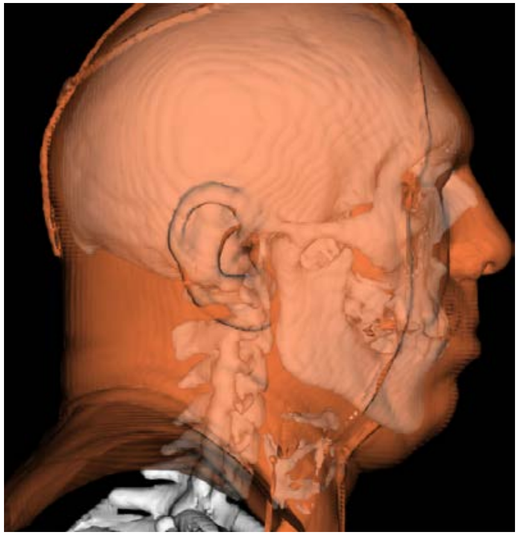
bone tissue density



brain activity (fMRI)

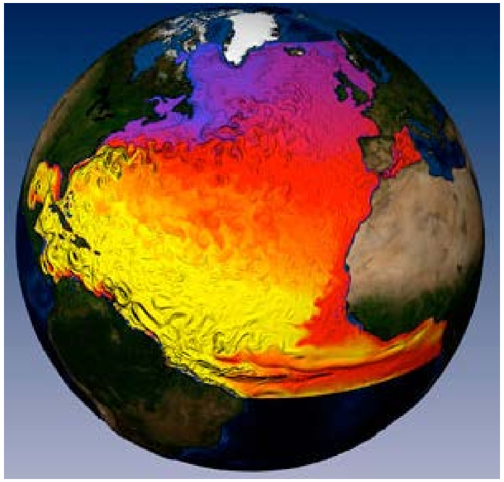


MRI scan - tissues

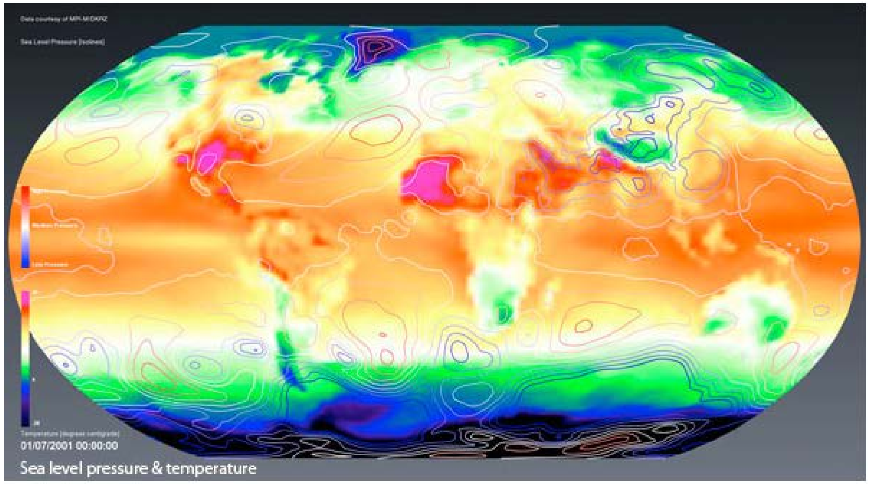


bone + skin surface

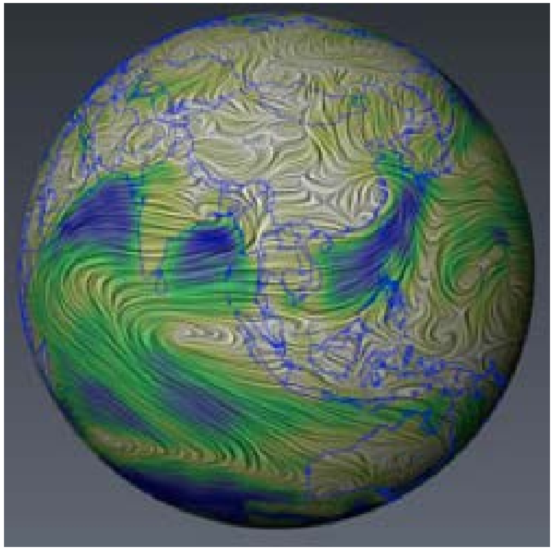
Visualization examples: Geosciences



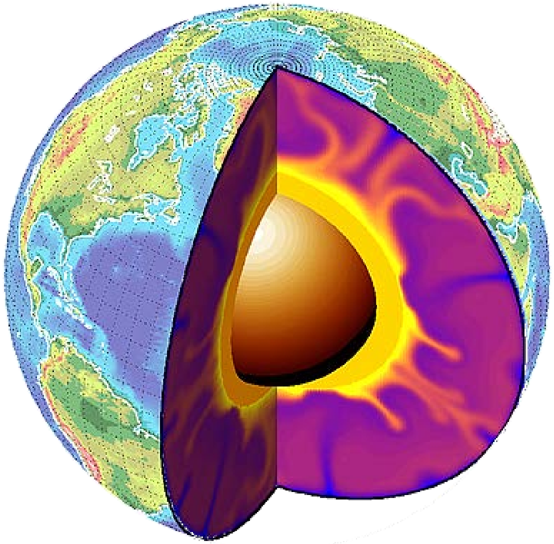
ocean velocity
and surface temperature



sea level pressure and temperature



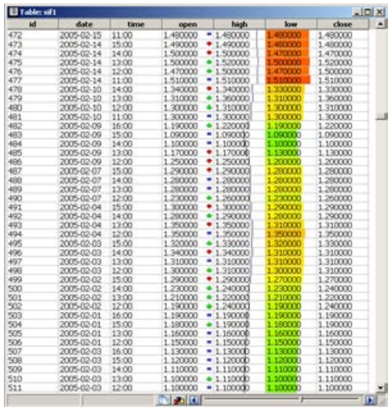
wind flow paths over
Earth's surface



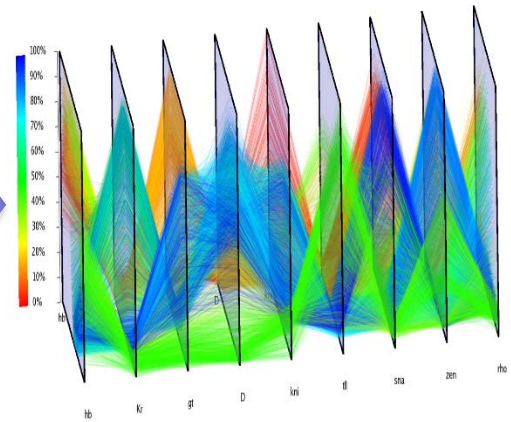
Earth surface and inner temperature

Visualization examples: Abstract data

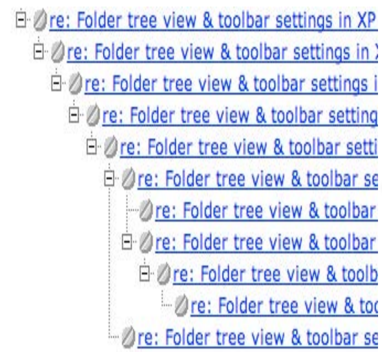
- mapping is not ‘neutral’ or natural, but reflects the problem/question to be solved



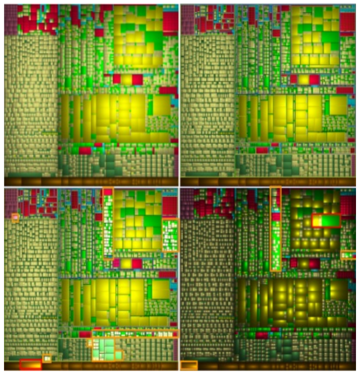
data table: classical view



data table: parallel coordinates view



tree: explorer view

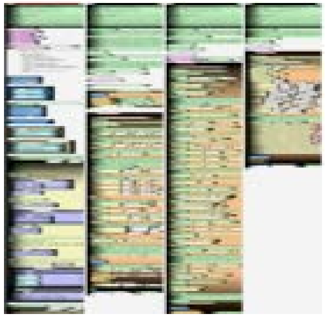


tree: cushion treemap view

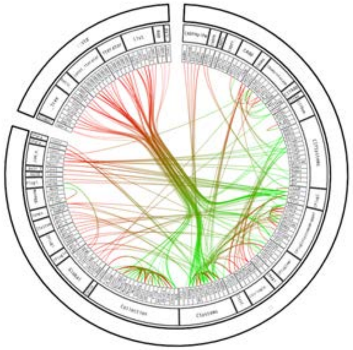
```
void ASTVisitor::traverse(ASTNode &obj)
{
  ASTNodeStack stack;
  static ASTNode sentinelNode(0); //put on the bottom of the s
  stack.push(StackItem(sentinelNode, SHOULD_IGNORE));
  stack.push(StackItem(obj, SHOULD_VISIT)); //the node that wi
  while(!stack.empty())
  {

    ASTNode &curNode(stack.top().astNode);
    if (stack.top().postVisit == SHOULD_IGNORE)
    {
      stack.pop();
    }
    else if (stack.top().postVisit == SHOULD_POSTVISIT)
    {
      const Visit visitResult(postVisitASTNode(curNode));
      if (visitResult == VISIT_STOP)
        return;
      stack.pop();
      if (visitResult == VISIT_POSTPARENT)
```

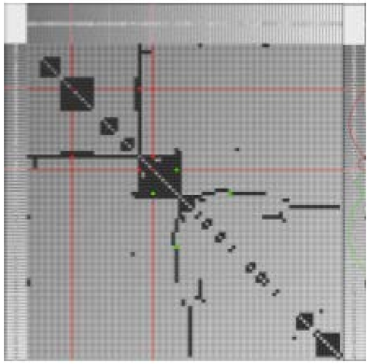
source code: classical view



source code: dense pixel view

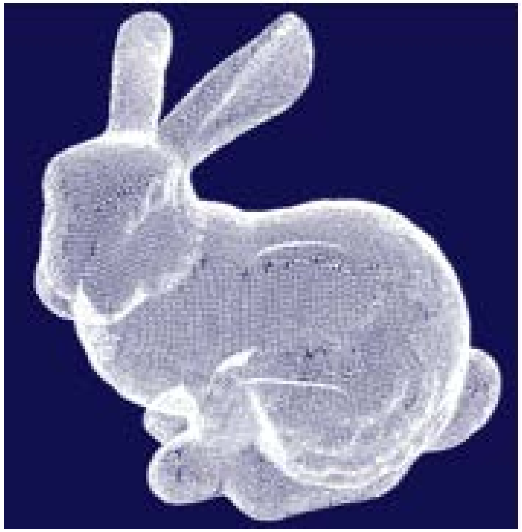


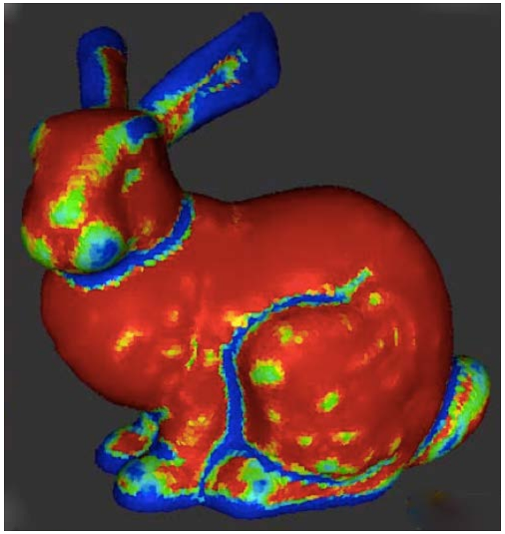
graph: bundled view



graph: adjacency matrix

Scientific Visualization – Basic data Characteristics





Stanford bunny 3D model
(70K triangles, 30K vertices)

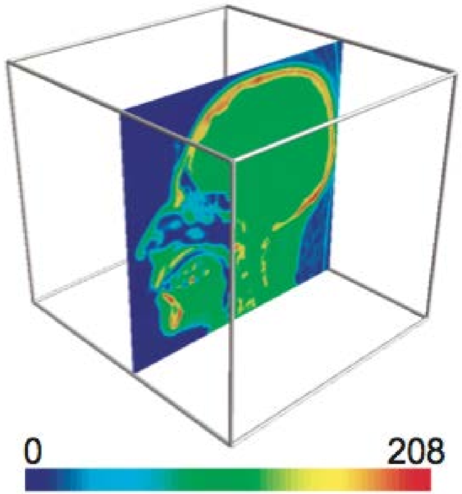
Right image shows surface curvature
(scalar dataset, red=flat, blue=curved)

- data: numerical values defined on a spatial domain, say: $f : \mathbf{R}^3 \rightarrow \mathbf{R}$
- both domain and range are continuous spaces
- hence the following are easy or at least possible
- resampling / rescaling
- filtering
- reconstruction (from piecewise discrete representation e.g. triangles)
- visual interpretation (domain is a natural 3D shape)

Our input: Dataset examples

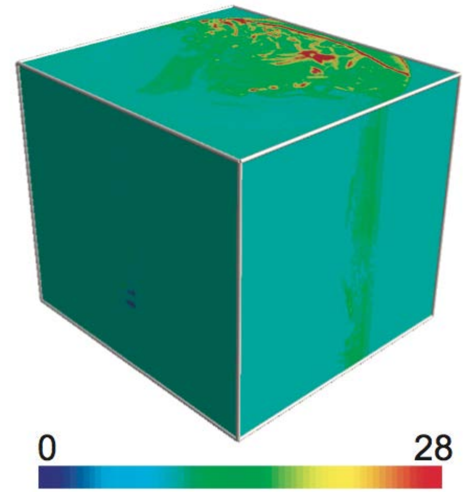
$f : \mathbf{R}^2 \rightarrow \mathbf{R}$

a planar slice



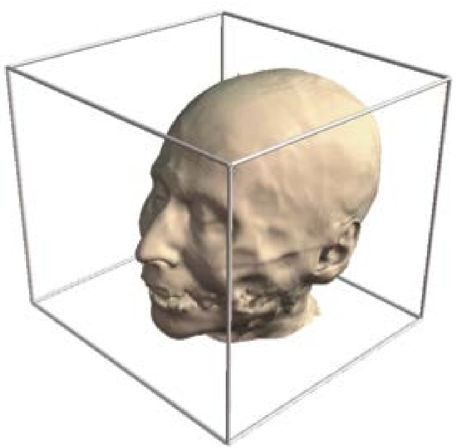
$f : \mathbf{R}^3 \rightarrow \mathbf{R}$

a volume



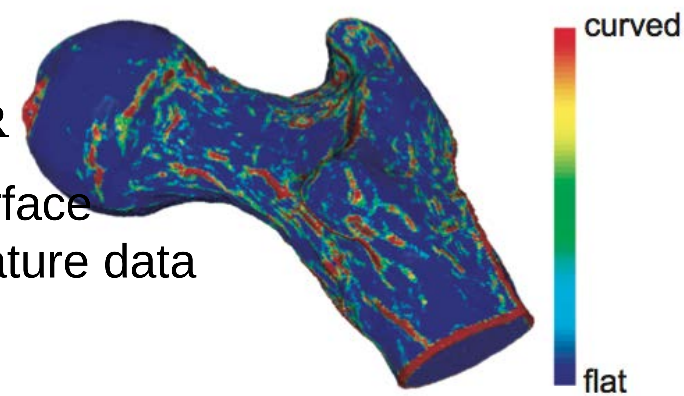
$f : \mathbf{R}^2 \rightarrow \mathbf{R}^0$

a surface



$f : \mathbf{R}^2 \rightarrow \mathbf{R}$

a surface
with curvature data



Basic Strategies

- Visualization of 1D, 2D, or 3D scalar fields
  - 1D scalar field: $\Omega \in R \rightarrow R$
  - 2D scalar field: $\Omega \in R^2 \rightarrow R$
  - 3D scalar field: $\Omega \in R^3 \rightarrow R$

    $\rightarrow$ **Volume visualization!**

Basic Strategies

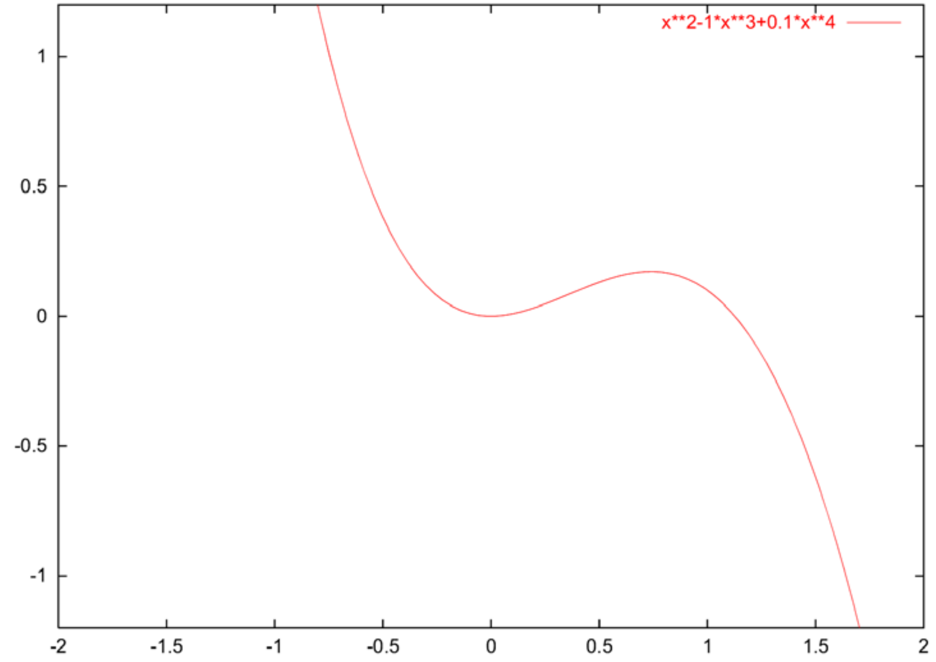
- Mapping to geometry
  - Function plots
  - Height fields
  - Isolines and isosurfaces
- Color coding
- Specific techniques for 3D data
  - Indirect volume visualization
  - Direct volume visualization
  - Slicing
- Visualization method depends heavily on dimensionality of domain

Function Plots and Height Fields

- Function plot for a 1D scalar field

$$\{(s, f(s))|s \in R\}$$

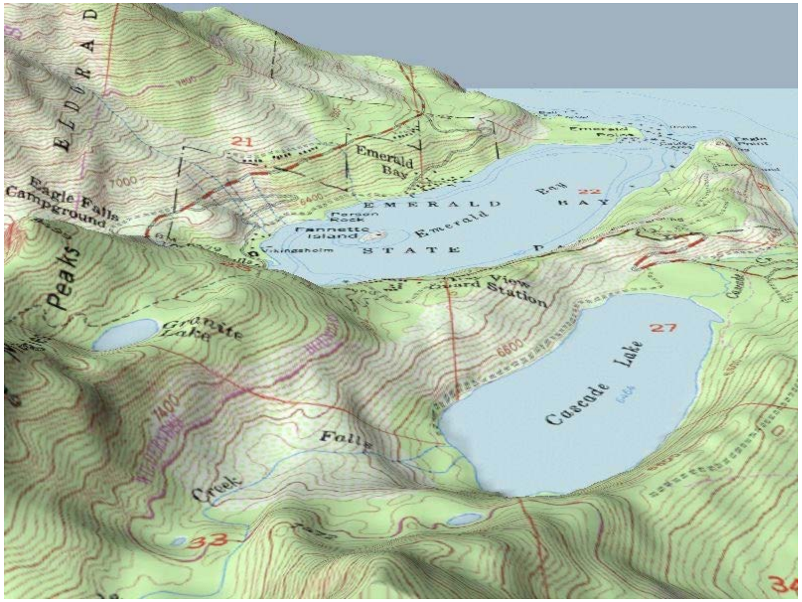
  - Points
  - 1D manifold: line
  - Error bars possible



Gnuplot example

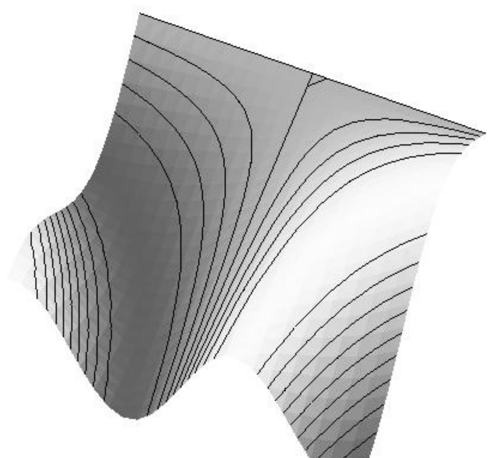
Function Plots and Height Fields

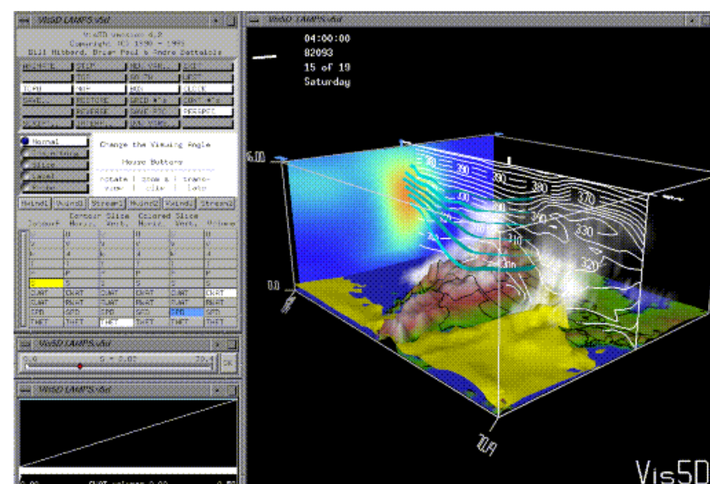
- Function plot for a 2D scalar field
$$\{(s,t,f(s,t))|(s,t) \in R^2\}$$
  - Points
  - 2D manifold: surface
- Surface representations
  - Wireframe
  - Hidden lines
  - Shaded surface



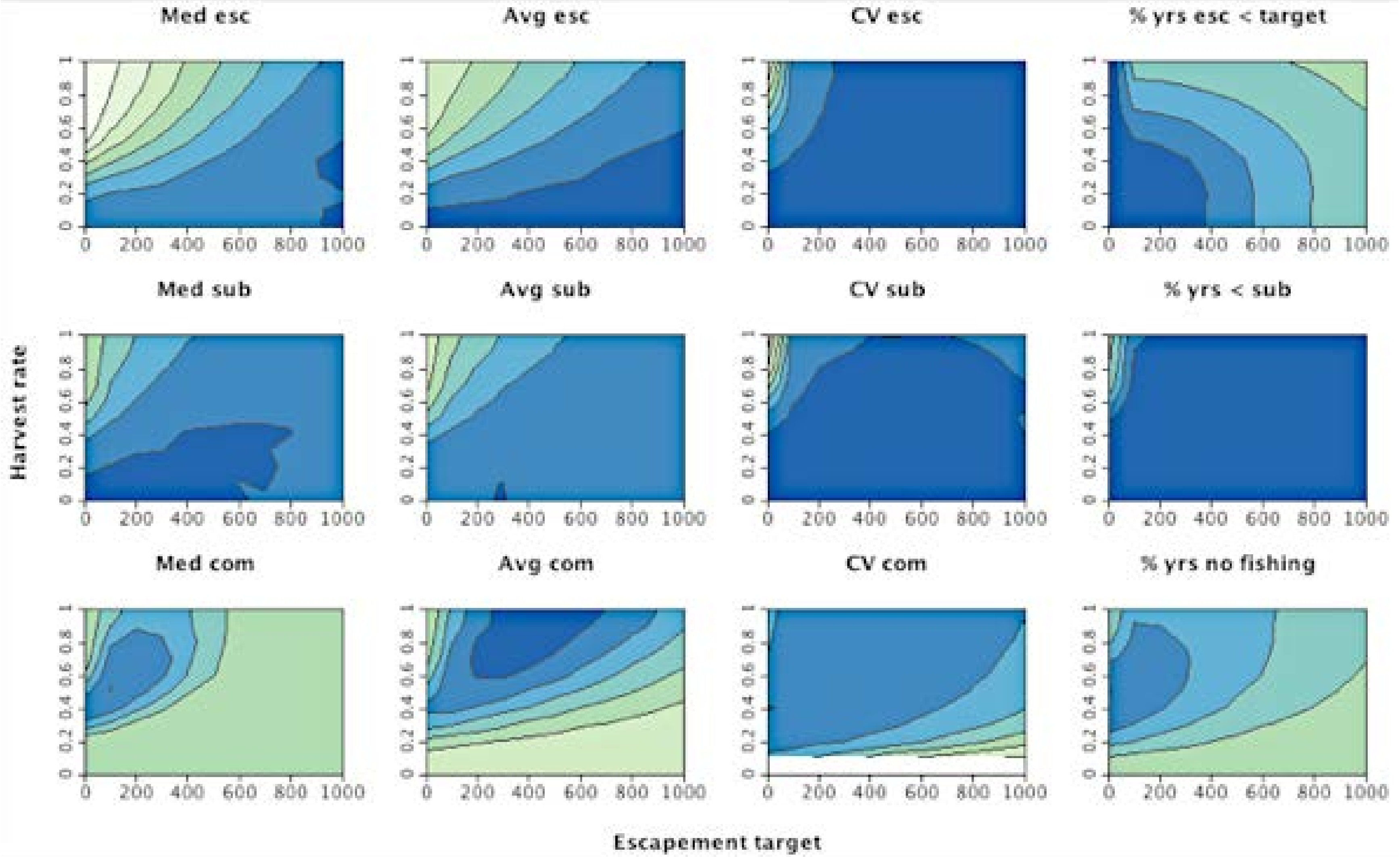
Isolines

- Visualization of 2D scalar fields
- Given a scalar function $f : \Omega \rightarrow R$ and a scalar value $c \in R$
- Isolines consist of points
  $$\{(x,y) | f(x,y) = c\}$$
- If ***f***() is differentiable and ***grad***(***f***) ≠ 0, then isolines are curves
- Contour lines





Choropleth / Isopleth

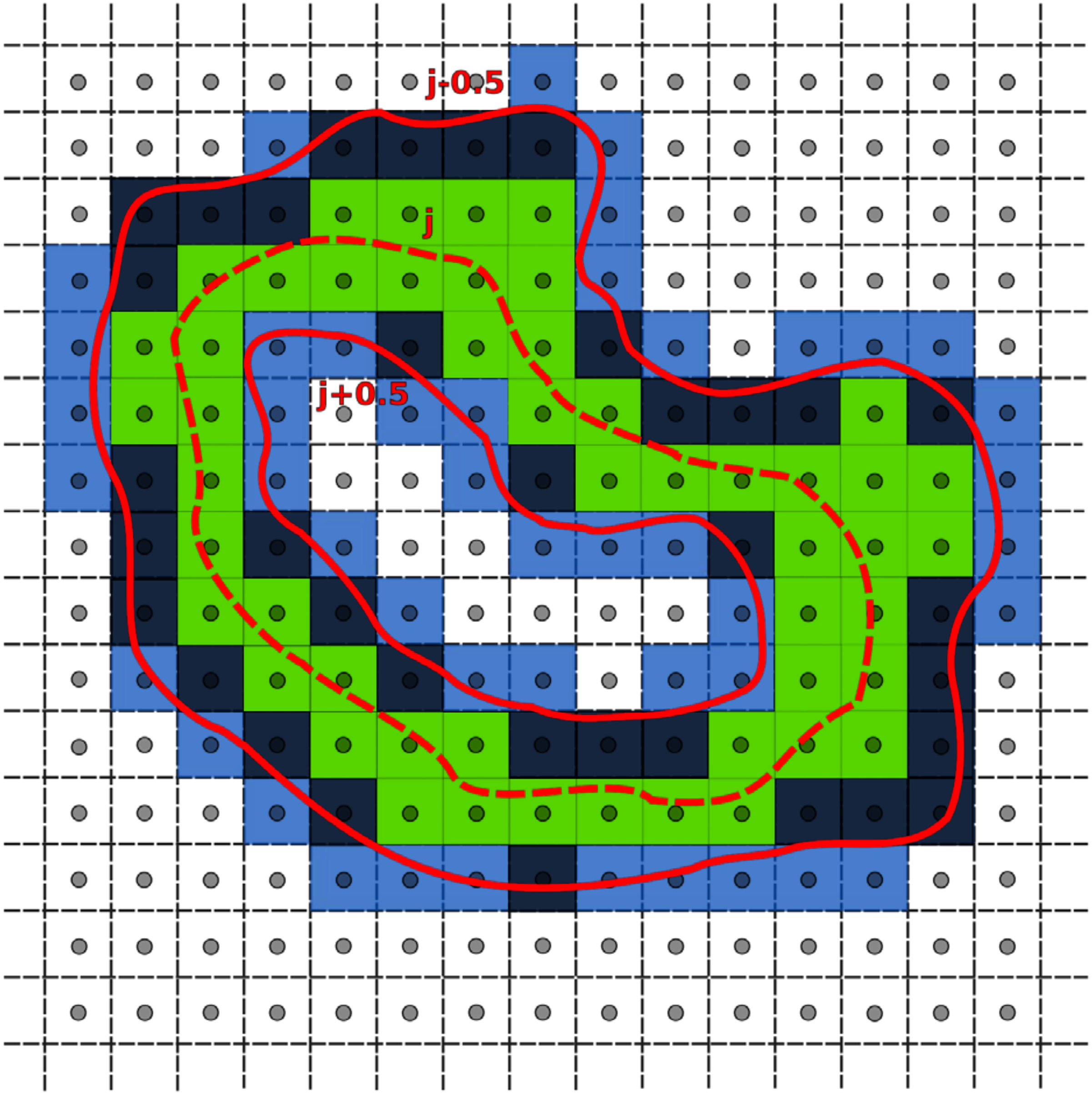


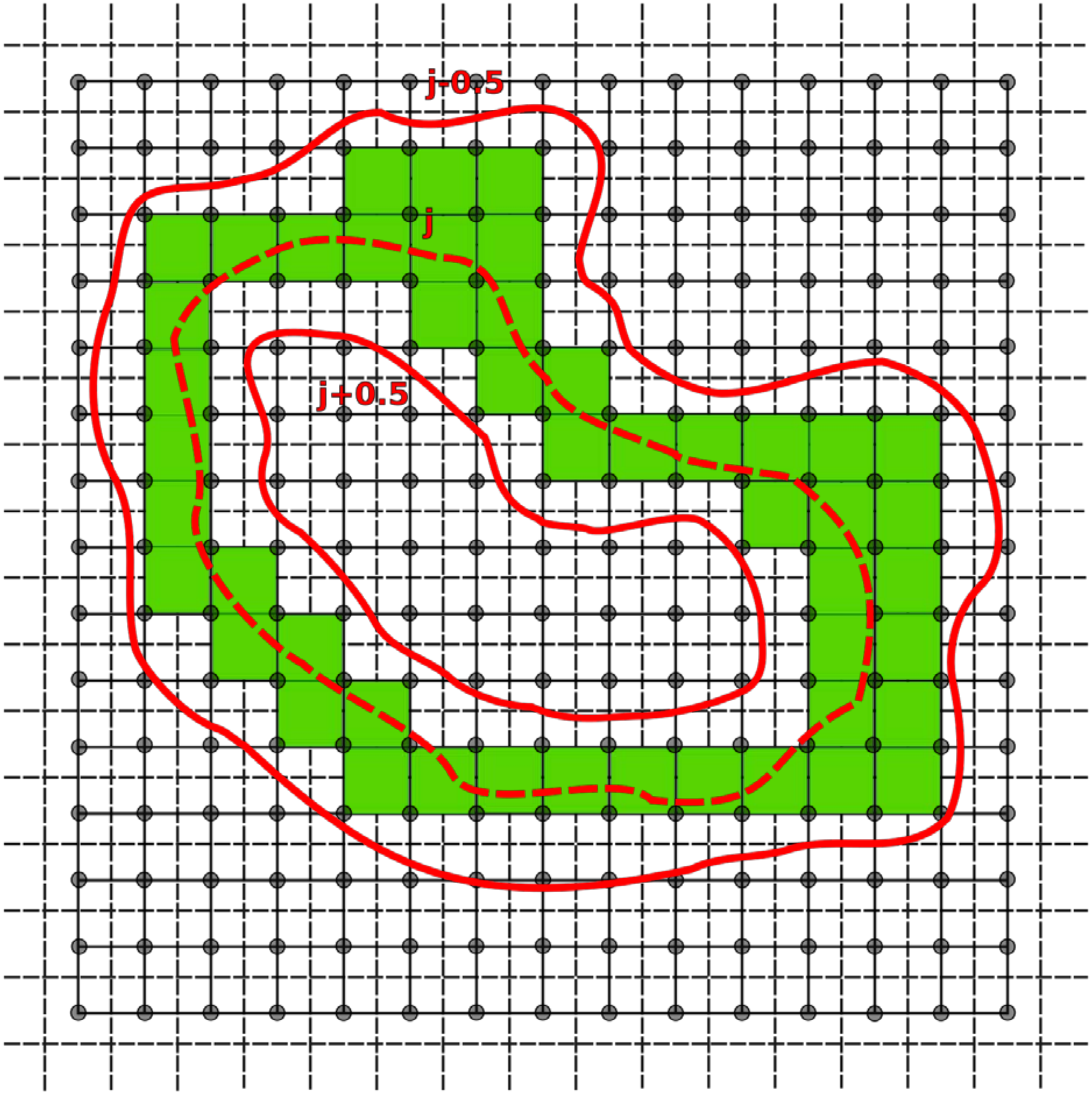
Isolines

- Pixel by pixel contouring
- Straightforward approach: scanning all pixels for equivalence with isovalue
- Input
  - $f : (1,\ldots,x_{max}) \times (1,\ldots,y_{max}) \rightarrow \mathbf{R}$
  - Isovalues $I_1,\ldots, I_n$ and isocolors $c_1,\ldots,c_n$
- Algorithm

```
for all (x,y) ∈ (1,...,x_max) x (1,...,y_max) do
    for all k∈ { 1,...,n } do
        if |f(x,y)-I_k| < ε then
            draw(x,y,c_k)
```

- Problem: Isoline can be missed if the gradient of *f()* is too large (despite range ε)


j-0.5
j
j+0.5

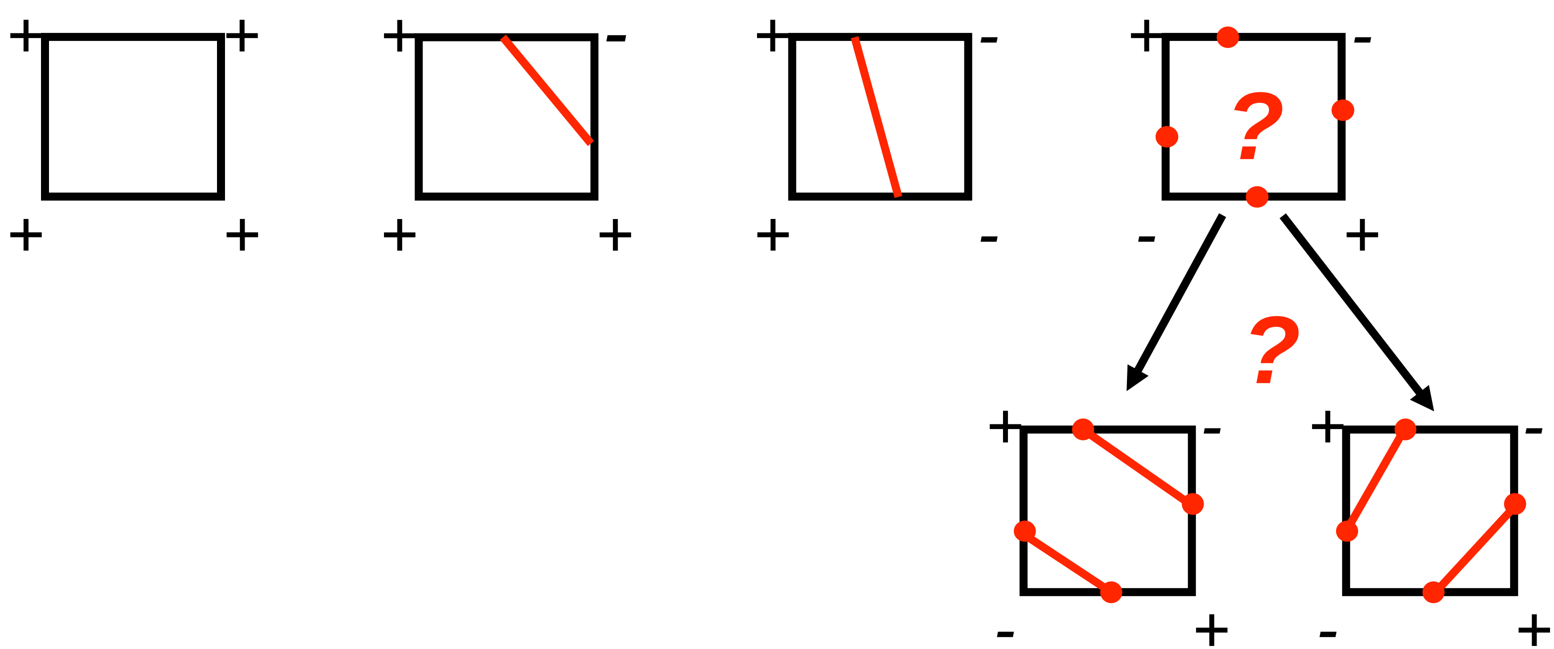

j-0.5
j
j+0.5

Marching Squares

- Representation of the scalar function on a rectilinear grid
- Scalar values are given at each vertex
- Take into account the interpolation within cells
- Isolines cannot be missed
- Divide and conquer: consider cells independently of each other

Marching Squares

- 4 different cases (classes) of combinations of signs
- Symmetries: rotation, reflection, change + $\boxed{\leftrightarrow}$ -
- Compute intersections between isoline and cell edge, based on linear interpolation along the cell edges

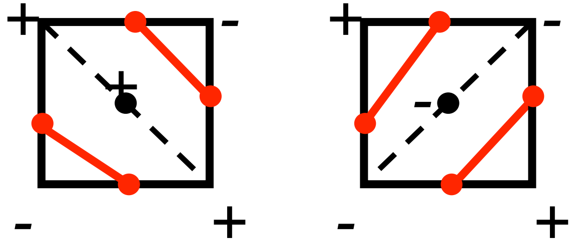


Isolines

- We can distinguish the cases by a decider
- Mid point decider
  - Interpolate the function value in the center

$$f_{\text{center}} = \frac{1}{4}\left(f_{i,j} + f_{i+1,j} + f_{i,j+1} + f_{i+1,j+1}\right)$$

  - If $f_{center} < c$ we chose the right case, otherwise we chose the left case



  - Not always correct solution