King Abdullah University of Science and Technology

KAUST

CS 247 – Scientific Visualization
Lecture 6: Scalar Fields, Pt. 2

Markus Hadwiger, KAUST

Reading Assignment #3 (until Feb 14)

Read (required):

- Data Visualization book, finish Chapter 3 (read starting with 3.6)
- Data Visualization book, Chapter 5 until 5.3 (inclusive)

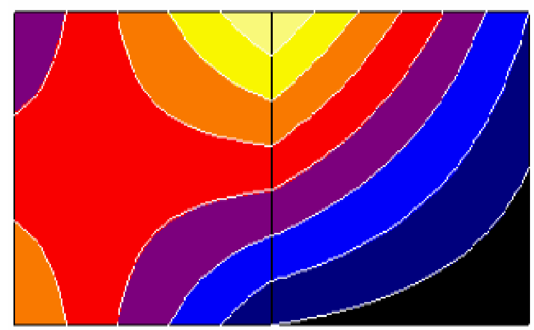
Scalar Fields

Contours

Set of points where the scalar field *f(x)* has a given value *c*:

$$S(c) := f^{-1}(c) \qquad S(c) := \{x \in \mathbb{R}^n : f(x) = c\}$$

bilinear interpolation

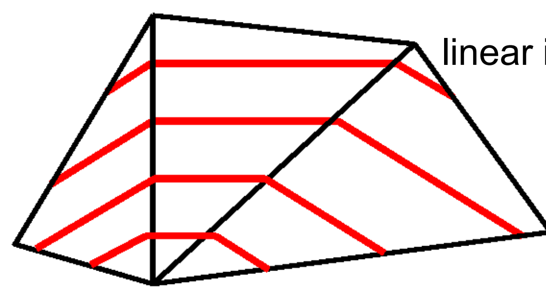


Common contouring algorithms

- 2D: marching squares, marching triangles
- 3D: marching cubes, marching tetrahedra

Implicit methods

- Point-on-contour test
- Isosurface ray-casting



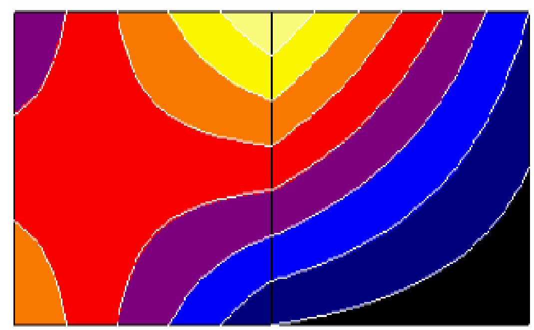
linear interpolation

Contours

Set of points where the scalar field *f(x)* has a given value *c*:

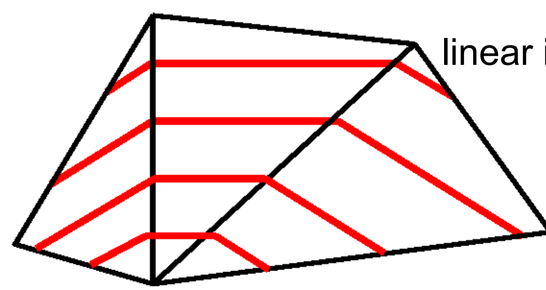
$$S(c) := f^{-1}(c) \qquad S(c) := \{x \in \mathbb{R}^2 : f(x) = c\}$$

bilinear interpolation



Common contouring algorithms

- 2D: marching squares, marching triangles
- 3D: marching cubes, marching tetrahedra



linear interpolation

Implicit methods

- Point-on-contour test
- Isosurface ray-casting

Contours

Set of points where the scalar field *f(x)* has a given value *c*:

$$S(c) := f^{-1}(c) \qquad S(c) := \{x \in \mathbb{R}^3 : f(x) = c\}$$

bilinear interpolation

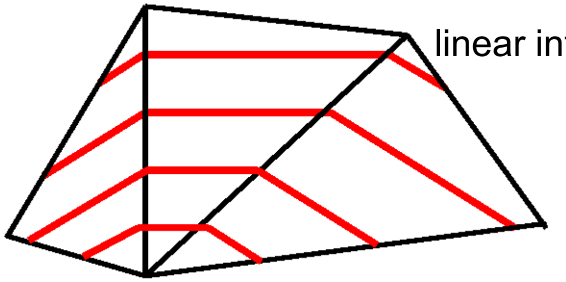
Common contouring algorithms

- 2D: marching squares, marching triangles
- 3D: marching cubes, marching tetrahedra

Implicit methods

- Point-on-contour test
- Isosurface ray-casting

linear interpolation



What are contours?

Set of points where the scalar field *s* has a given value *c*:

$$S(c) := \{x \in \mathbb{R}^n : f(x) = c\}$$

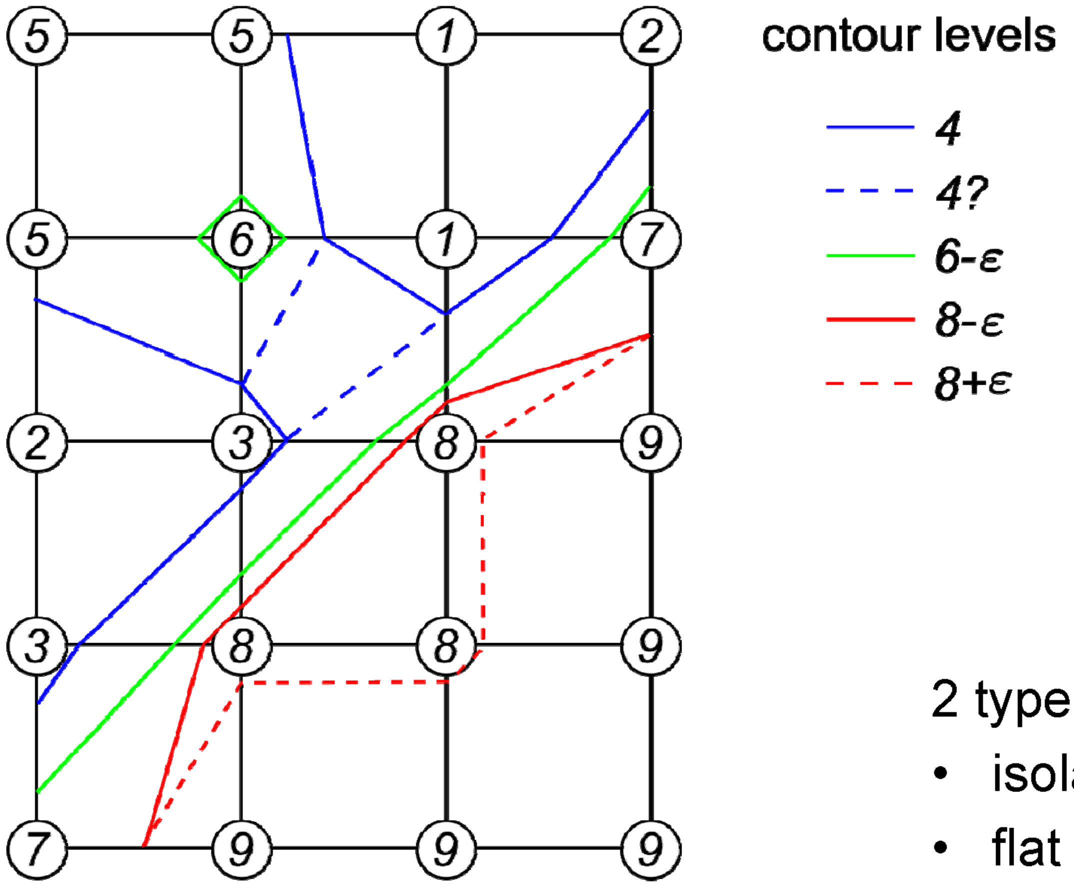
Examples in 2D:

- height contours on maps
- isobars on weather maps

Contouring algorithm:

- find intersection with grid edges
- connect points in each cell

Example



2 types of degeneracies:

- isolated points (c=6)
- flat regions (c=8)

Contours in a quadrangle cell

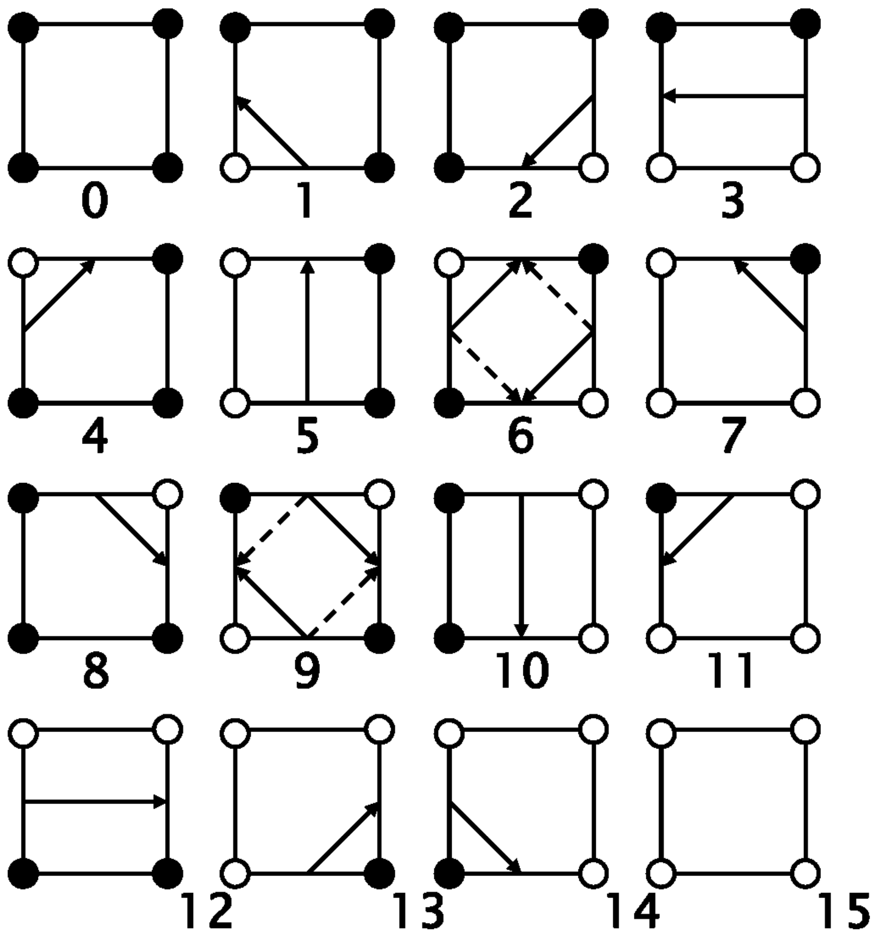
Basic contouring algorithms:

- cell-by-cell algorithms: simple structure, but generate disconnected segments, require post-processing
- contour propagation methods: more complicated, but generate connected contours

"Marching squares" algorithm (systematic cell-by-cell):

- process nodes in ccw order, denoted here as x_0, x_1, x_2, x_3
- compute at each node $\mathbf{x}_i$ the reduced field $\tilde{f}(x_i) = f(x_i) - (c - \varepsilon)$ (which is forced to be nonzero)
- take its sign as the i^{th} bit of a 4-bit integer
- use this as an index for lookup table containing the connectivity information:

Contours in a quadrangle cell



- ● $\tilde{f}(x_i) < 0$
- ○ $\tilde{f}(x_i) > 0$

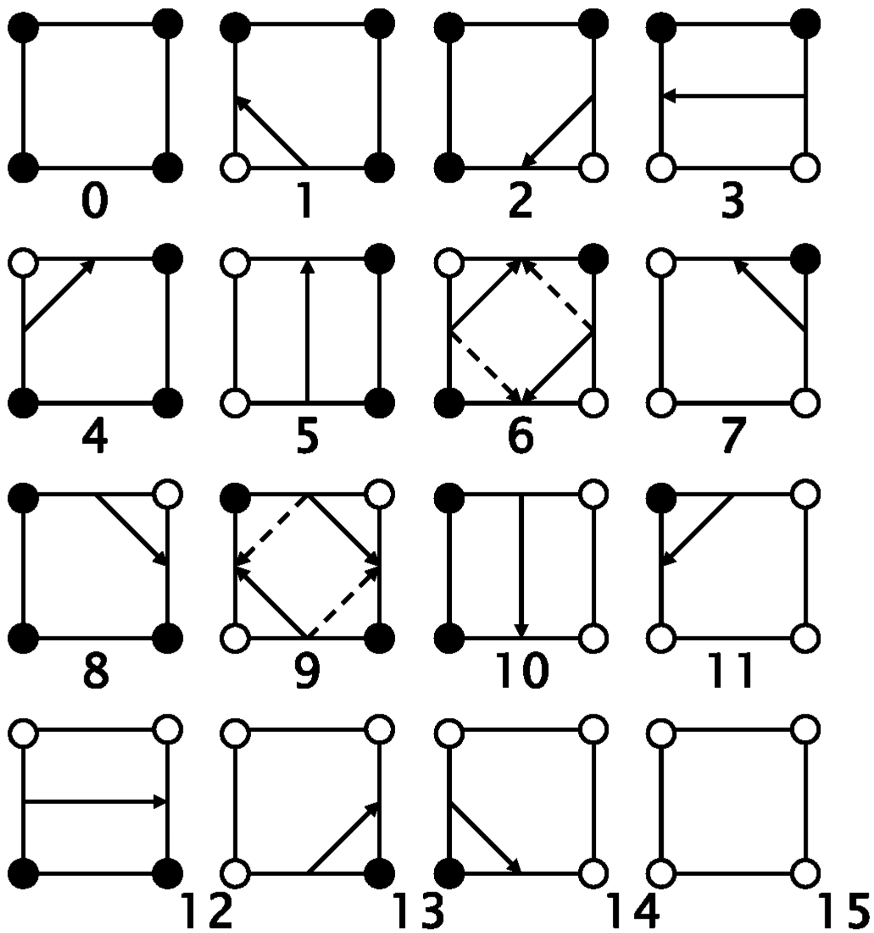
Alternating signs exist in cases 6 and 9.

Choose the solid or dashed line?

Both are possible for topological consistency.

This allows to have a fixed table of 16 cases.

Contours in a quadrangle cell



- ● $f(x_i) < c$
- ○ $f(x_i) \geq c$

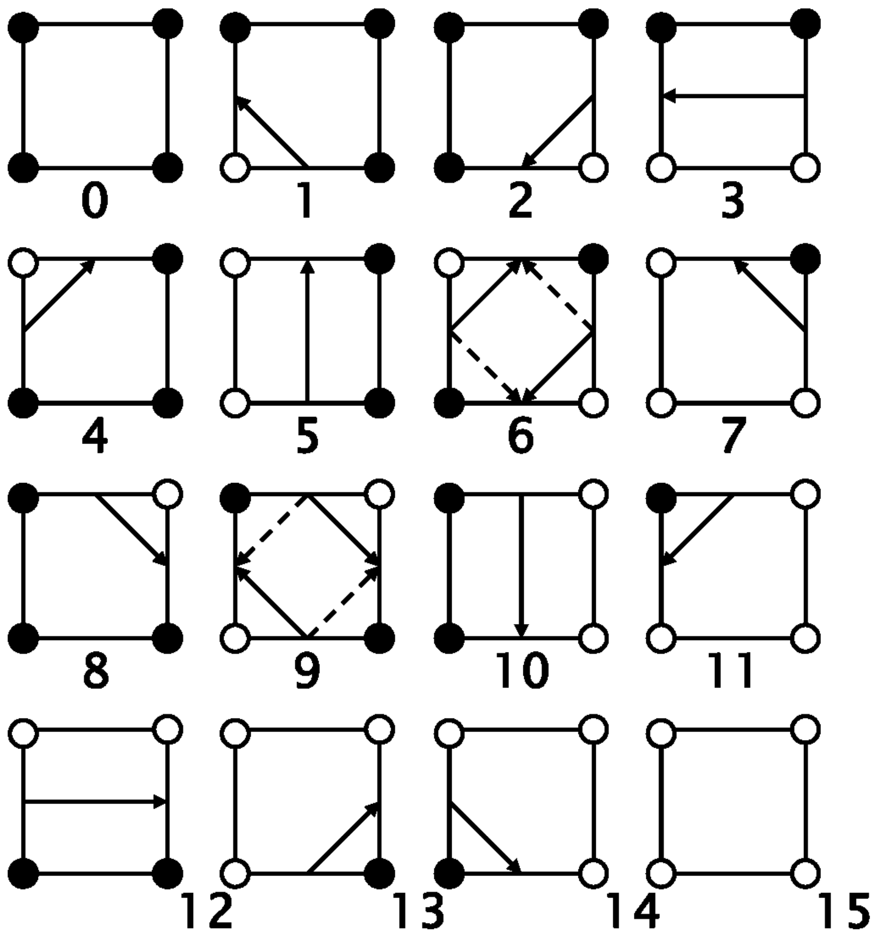
Alternating signs exist in cases 6 and 9.

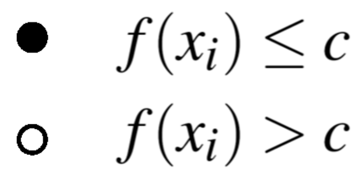
Choose the solid or dashed line?

Both are possible for topological consistency.

This allows to have a fixed table of 16 cases.

Contours in a quadrangle cell





Alternating signs exist in cases 6 and 9.

Choose the solid or dashed line?

Both are possible for topological consistency.

This allows to have a fixed table of 16 cases.

Orientability (1-manifold embedded in 2D)

Orientability of 1-manifold:

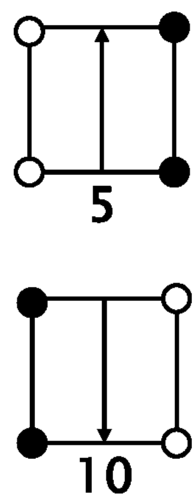
Possible to assign consistent left/right orientation

not orientable

Moebius strip
(only one side!)

Iso-contours

- Consistent side for scalar values…
  - greater than iso-value (e.g, *left* side)
  - less than iso-value (e.g., *right* side)
- Use consistent ordering of vertices
  (e.g., larger vertex index is "tip" of arrow;
  if (0,1) points "up", "left" is left, …)



● $\tilde{f}(x_i) < 0$

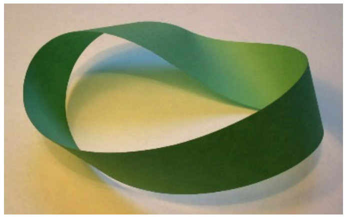
○ $\tilde{f}(x_i) > 0$

Orientability (2-manifold embedded in 3D)

Orientability of 2-manifold:

Possible to assign consistent normal vector orientation

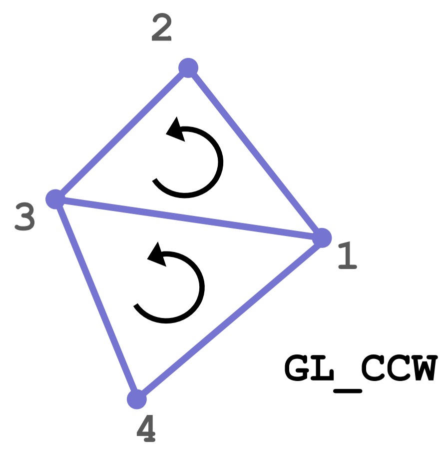
not orientable



Moebius strip
(only one side!)

Triangle meshes

- Edges
  - Consistent ordering of vertices: CCW (counter-clockwise) or CW (clockwise) (e.g., (3,1,2) on one side of edge, (1,3,4) on the other side)
- Triangles
  - Consistent front side vs. back side
  - Normal vector; or ordering of vertices (CCW/CW)
  - See also: “right-hand rule”



Topological consistency

To avoid degeneracies, use symbolic perturbations:

If level c is found as a node value, set the level to $c-\varepsilon$ where ε is a symbolic infinitesimal.

Then:

- contours intersect edges at some (possibly infinitesimal) distance from end points
- flat regions can be visualized by pair of contours at $c-\varepsilon$ and $c+\varepsilon$
- contours are topologically consistent, meaning:

Contours are closed, orientable, nonintersecting lines.

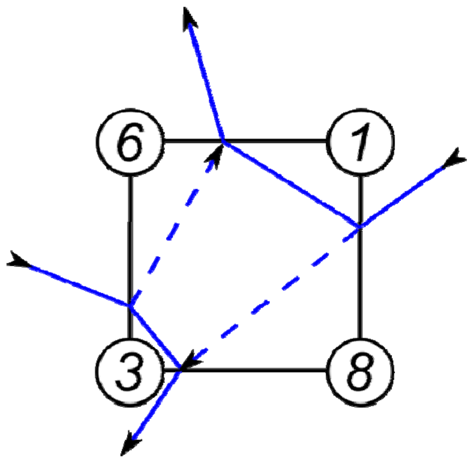
(except where the boundary is hit)

Ambiguities of contours

What is the correct contour of c=4?

Two possibilities, both are orientable:

- connect high values
- connect low values



Answer: correctness depends on interior values of $f(x)$.

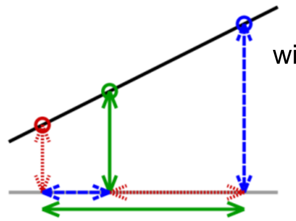
But: different interpolation schemes are possible.

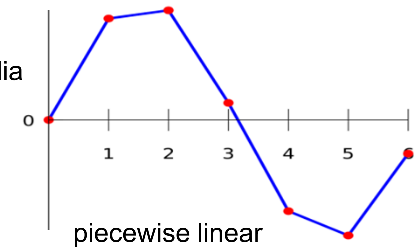
Better question: What is the correct contour with respect to bilinear interpolation?

Linear Interpolation / Convex Combinations

Linear interpolation in 1D:

$$f(\alpha) = (1-\alpha)v_1 + \alpha v_2$$





Line embedded in 2D (linear interpolation of vertex coordinates/attributes):

$$f(\alpha_1, \alpha_2) = \alpha_1 v_1 + \alpha_2 v_2 \qquad f(\alpha) = v_1 + \alpha(v_2 - v_1)$$

$$\alpha_1 + \alpha_2 = 1 \qquad \alpha = \alpha_2$$

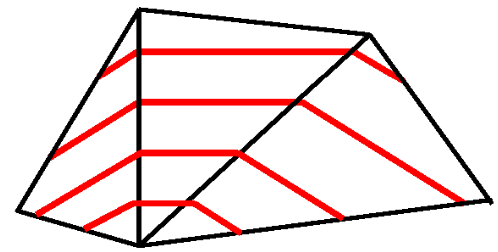
Line segment: $\alpha_1, \alpha_2 \geq 0$ (→ convex combination)

Compare to line parameterization
with parameter t:

$$v(t) = v_1 + t(v_2 - v_1)$$

Contours in triangle/tetrahedral cells

Linear interpolation of cells implies piece-wise linear contours.



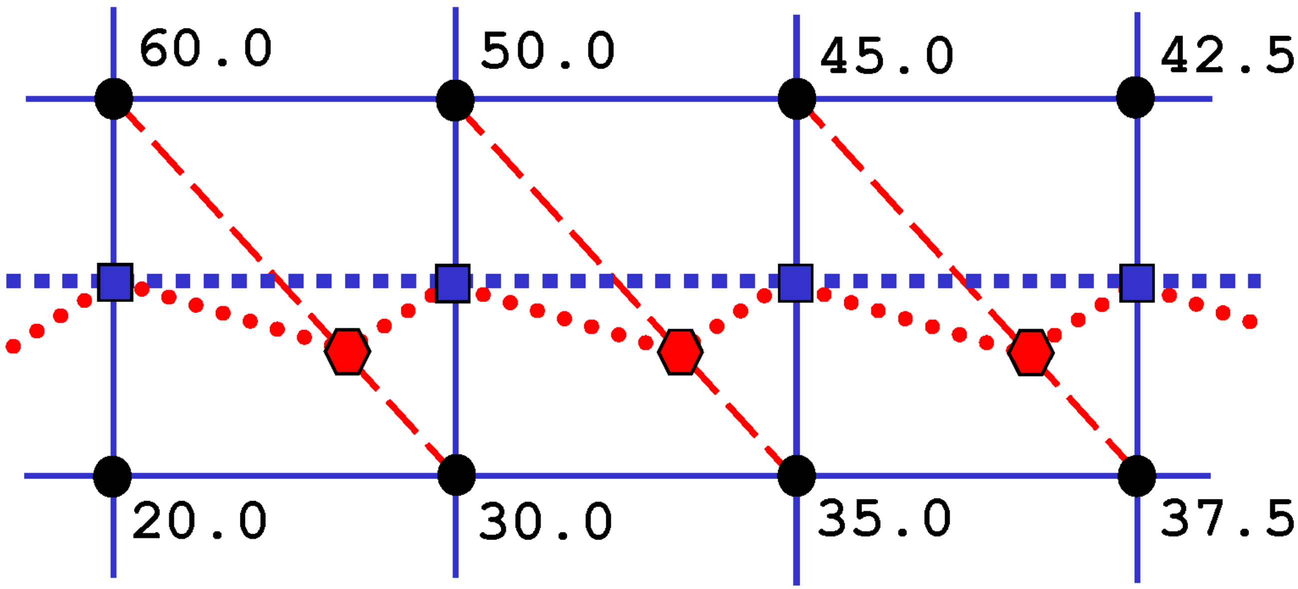
Contours are unambiguous, making "marching triangles" even simpler than "marching squares".

Question: Why not split quadrangles into two triangles (and hexahedra into five or six tetrahedra) and use marching triangles (tetrahedra)?

Answer: This can introduce periodic artifacts!

Contours in triangle/tetrahedral cells

Illustrative example: Find contour at level *c*=40.0 !



original quad grid, yielding vertices and contour

triangulated grid, yielding vertices and contour

Thank you.

Thanks for material

- Helwig Hauser
- Eduard Gröller
- Daniel Weiskopf
- Torsten Möller
- Ronny Peikert
- Philipp Muigg
- Christof Rezk-Salama