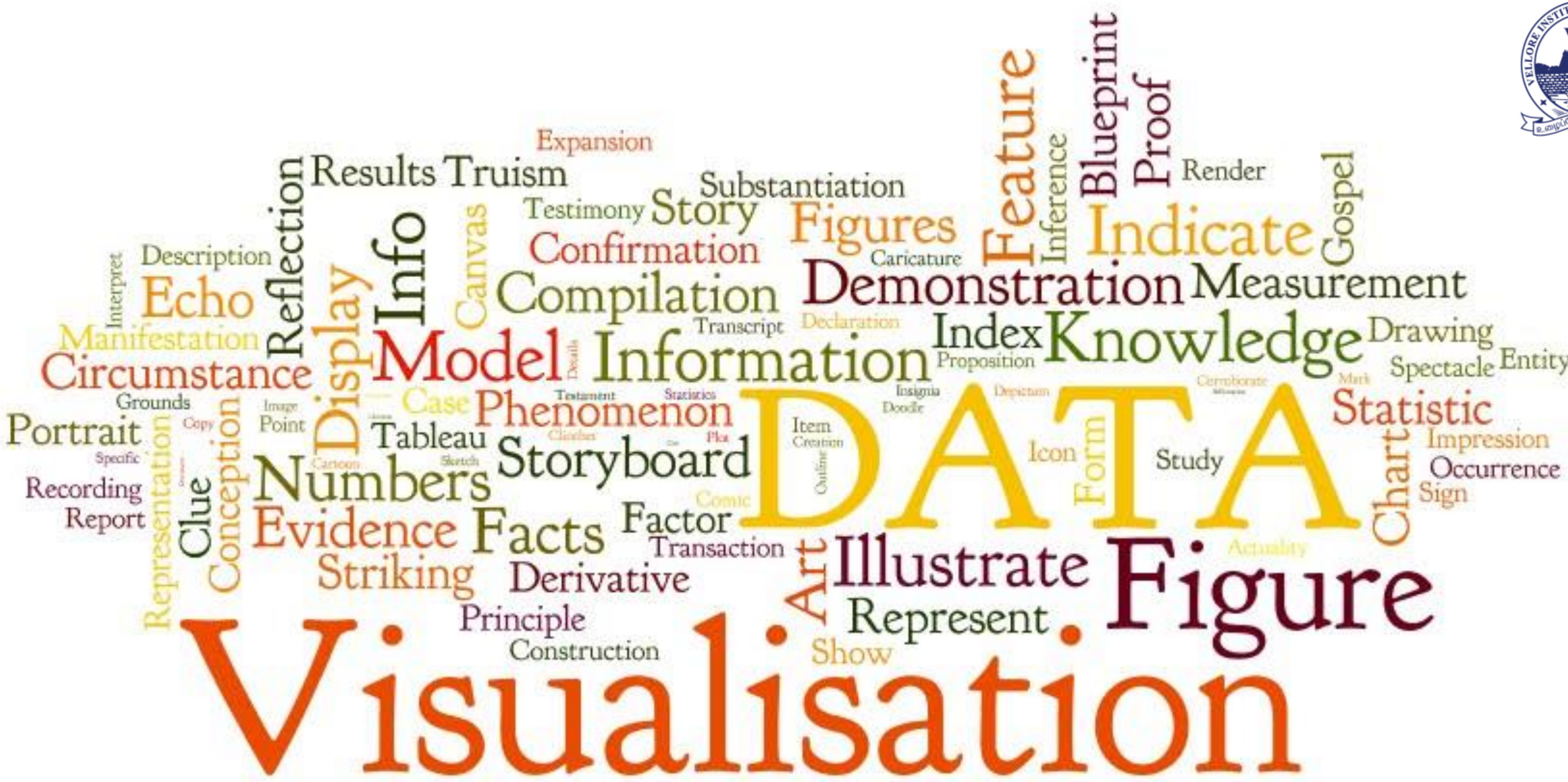


6.Scalar Visualization

Prof. Pattabiraman.V
SCSE, VIT, Chennai

Outline

- Introduction
- Color Mapping
- Designing Effective Colormaps
  - Guide Lines
  - Rainbow colormap
  - Different colormap Designs
- Contouring

Introduction

- Visualizing scalar data is frequently encountered in science, engineering, and medicine, but also in daily life.
- Scalar datasets, or scalar fields, represent functions $f:D \rightarrow R$, where D is usually a subset of R2 or R3.
- There exist many scalar visualization techniques, both for 2D and 3D datasets.

Color Mapping

- Color mapping is probably the most widespread visualization method for scalar data. Color mapping associates a color with every scalar value.

- There are several ways to define such a scalar-to-color function.

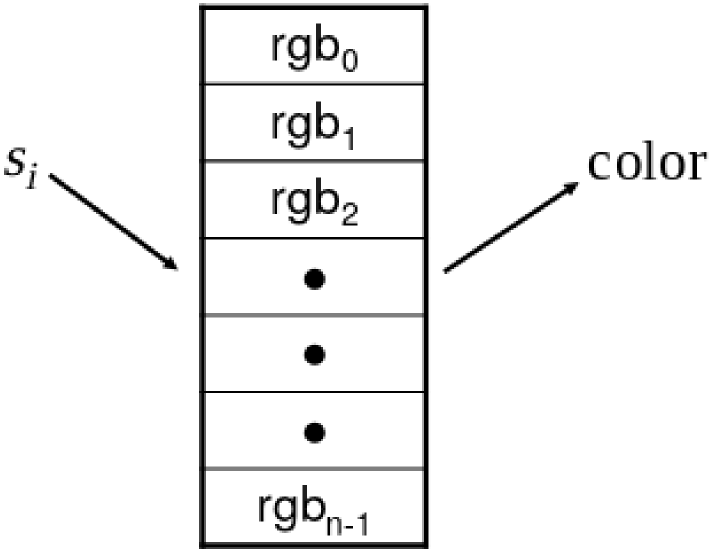
  - Color look-up tables

  - Color transfer functions

Color look-up tables

- Color look-up tables are the simplest way to implement color mapping.

$$
\begin{aligned}
s_i < \min \quad &: \quad i = 0 \\
s_i > \max \quad &: \quad i = n-1 \\
\text{otherwise:} \quad & i = n \cdot \left(\frac{s_i - \min}{\max - \min} \right)
\end{aligned}
$$

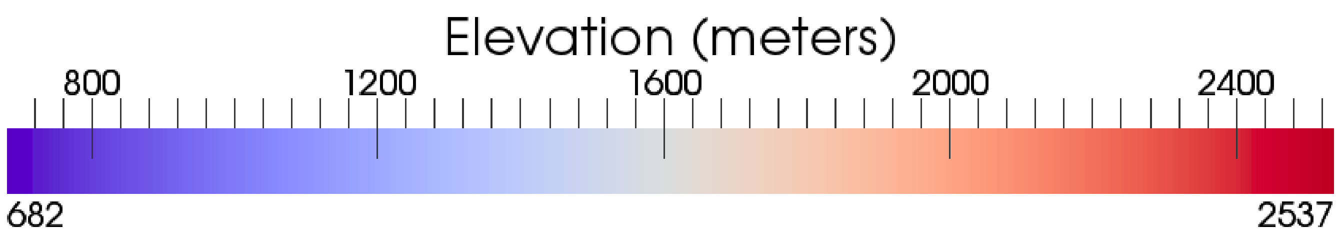


- Simply put, a color look-up table ***C***, also called a colormap, is a uniform sampling of the color-mapping function ***c***:

Color look-up tables

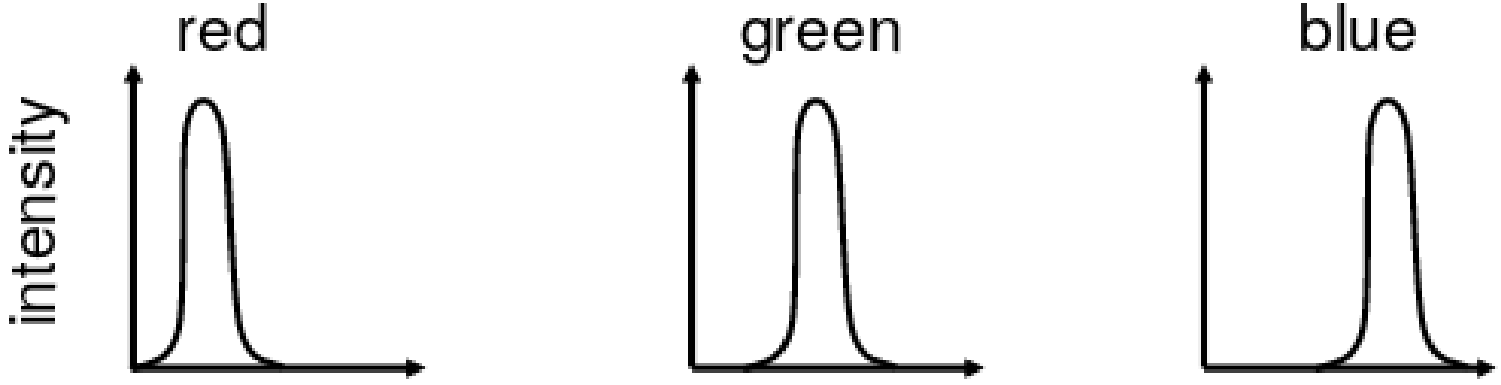
- Scalar values greater than the maximum are clamped to the maximum color, scalar values less than the minimum are clamped to the minimum value.

R	G	B	Result
0.0	0.0	0.0	black
1.0	0.0	0.0	red
0.0	1.0	0.0	green
1.0	1.0	0.0	yellow
0.0	0.0	1.0	blue
1.0	0.0	1.0	magenta
0.0	1.0	1.0	cyan
1.0	1.0	1.0	white



Color Transfer Functions

- A transfer function is any expression that maps scalar values into a color specification. For example, a function can be used to map scalar values into separate intensity values for the red, green, and blue components.



Color look-up tables vs Color Transfer Functions

- A lookup table is a discrete sampling of a transfer function. We can create a lookup table from any transfer function by sampling the transfer function at a set of discrete points.

Designing Effective Colormaps

- The main challenge for visualizations using color mapping is to design an effective colormap.
- A color-mapping visualization is effective if, by looking at the generated colors, we can easily and accurately make statements about the original scalar dataset that was color mapped.

Designing Effective Colormaps

- Different types of analysis goals require different types of colormaps.

1. *Absolute values*: Tell the absolute data values at all points in the displayed dataset.
2. *Value ordering*: Given two points in the displayed dataset, tell which of the corresponding two data values is greater.
3. *Value difference*: Given two points in the displayed dataset, tell what is the difference of data values at these points.

Designing Effective Colormaps

- Different types of analysis goals require different types of colormaps.

4. *Selected values*: Given a particular data value *f* interest , tell which points in the displayed data take the respective value *f* interest . A variation of this goal replaces *f* interest by a compact interval of data values.

5. *Value change*: Tell the speed of change, or first derivative, of the data values at given points in the displayed dataset.

Designing Effective Colormaps

- Different types of analysis goals require different types of colormaps.

4. *Selected values*: Given a particular data value *f* interest , tell which points in the displayed data take the respective value *f* interest . A variation of this goal replaces *f* interest by a compact interval of data values.

5. *Value change*: Tell the speed of change, or first derivative, of the data values at given points in the displayed dataset.

Designing Effective Colormaps

- Designing a colormap that achieves all above goals in equal measure is a very challenging task.
- Design decisions pertaining to the construction of colormaps, and relate them to the goals mentioned.

Designing Effective Colormaps

Color legends

– To achieve the first goal, we must be able to mentally invert the color-mapping function $\boldsymbol{c}$; that is, look at a color of some point in the visual domain $\boldsymbol{D_V}$ and tell its scalar value $\boldsymbol{f}$.

– In practice, this is achieved by drawing a so-called ***color legend***

Designing Effective Colormaps

Color legend mechanism has some conditions to succeed.

– First,the color-mapping function must be invertible.

– Every scalar value in the range [f min , f max] is associated with a unique color.

– The colors used must be unique in the eye of the beholder.

– It is not sufficient that the colors have different numerical (e.g., RGB) values.

– We must also be able to easily perceive them visually as being different if we want to be able to map them to scalars using the color legend.

Designing Effective Colormaps

- Second, the spatial resolution of the visualized dataset must be high enough as compared to the speed of variation of the scalar data ***f*** that we are able to visually distinguish separate regions having different colors.

- Color legends are required for any application of color mapping where we require to map a color to a data-related quantity. Note that this does not apply only to goal 1 of our taxonomy.

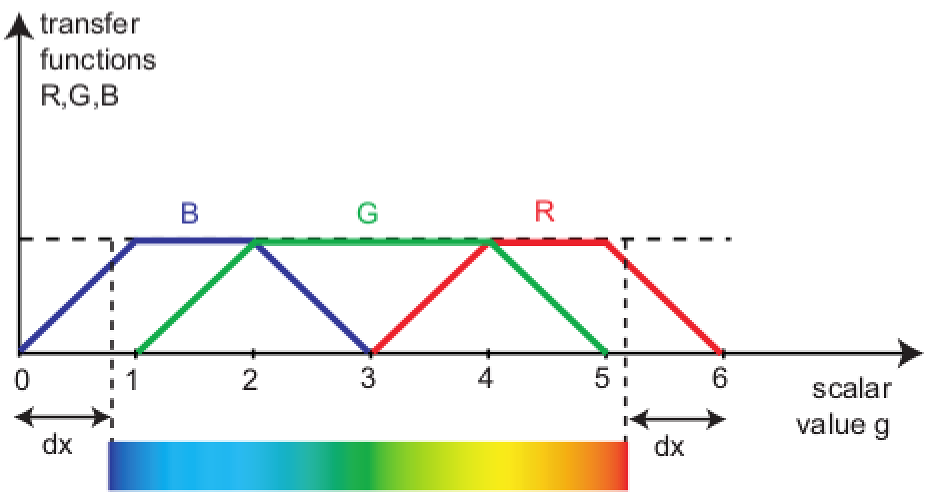
- Indeed, if we want to make quantitative judgments about the ordering of data values (goal 2), a color legend is required to tell how colors are ordered with respect to the ordering of the data values.

Designing Effective Colormaps

- If we want to compare distances between data values (goal-3), we need a color legend to show which differences are small and which are large.
- If we want to tell which data points take a given value of interest (goal-4), we need a color legend to tell which is the color showing the value of interest.
- If we want to tell the speed of data variation (goal-5), and reduce this task to color mapping the magnitude of the gradient ∇f of our scalar signal f , we then need a color legend for $\|\nabla f\|$.

Designing Effective Colormaps

- Rainbow colormap
  - Many engineering and weather forecast applications use a blue-to-red colormap, often called the rainbow colormap.
  - This colormap is based on the intuition that blue, a “cold” color, suggests low values, whereas red, a “hot” color, suggests high values.



Designing Effective Colormaps

- Rainbow colormap

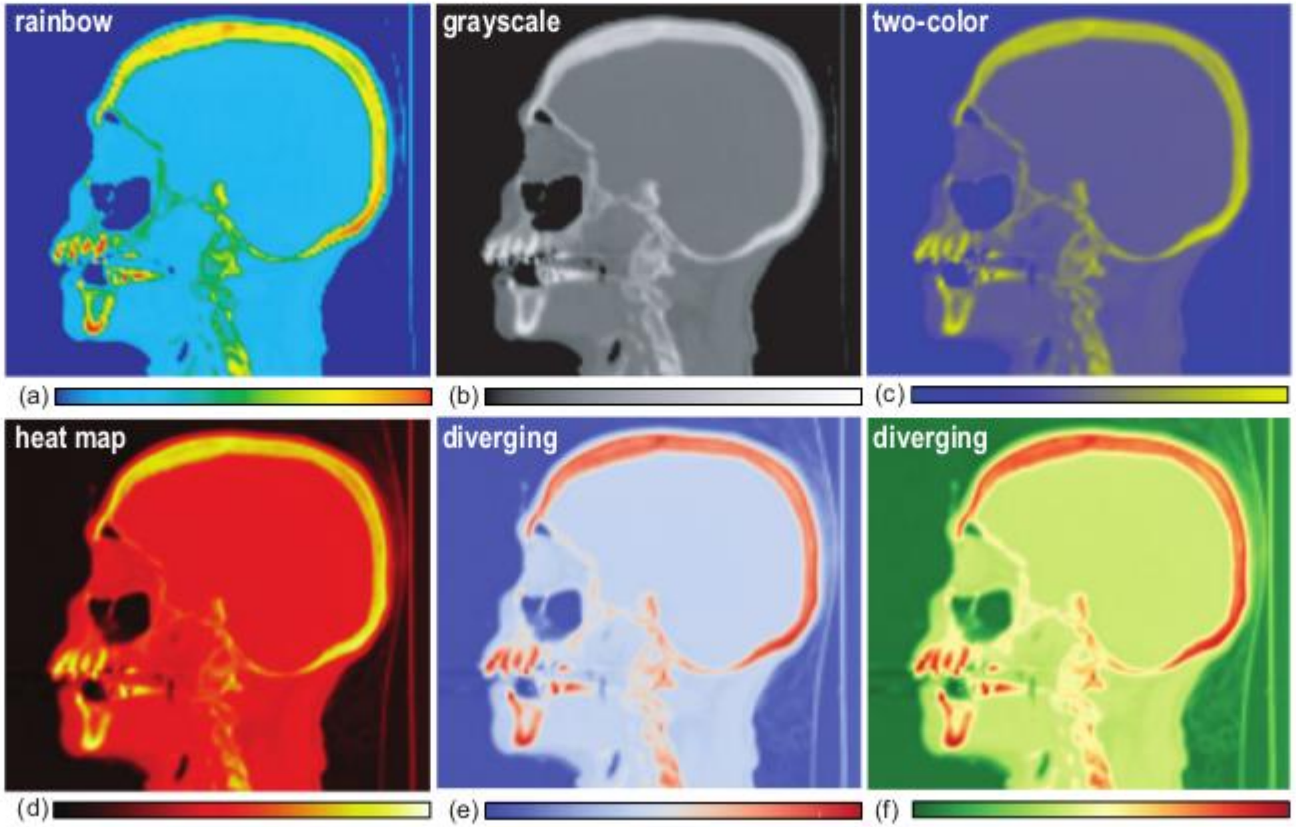
```
void c( float f , float& R, float& G, float& B)
{
const float dx = 0 . 8 ;
f = ( f <0)? 0 : ( f >1)? 1 : f ; //clamp f in [0, 1]
g = (6 −2∗ dx ) ∗ f + dx ; // scale f to [dx, 6 − dx]
R = max(0 ,(3 − f a b s ( g−4)− f a b s ( g − 5 ) ) / 2 ) ;
G = max(0 ,(4 − f a b s ( g−2)− f a b s ( g − 4 ) ) / 2 ) ;
B = max(0 ,(3 − f a b s ( g−1)− f a b s ( g − 2 ) ) / 2 ) ;
}
```

Designing Effective Colormaps

- Rainbow colormap has limitations:
  - **Focus:** Perceptually, warm colors arguably attract attention more than cold colors.
  - **Luminances:** of the rainbow colormap entries vary non-monotonically. This leads to users being potentially attracted more to certain colors than to others.
  - **Context:** Hues can have application-dependent semantics.
  - **Ordering:** we cannot assume that any user will order hues in this particular manner

Designing Effective Colormaps

- Other colormap designs



Designing Effective Colormaps

- Other colormap designs
- **Grayscale:** it maps data values ***f*** linearly to luminance, or gray value, with ***f*** min corresponding to black and ***f*** max corresponding to white.
- Most medical specialists, would agree that the grayscale produces a much easier-to-follow, less-confusing visualization on which details are easier to spot

Designing Effective Colormaps

- Other colormap designs
- **Grayscale:** it maps data values ***f*** linearly to luminance, or gray value, with ***f*** min corresponding to black and ***f*** max corresponding to white.
- Most medical specialists, would agree that the grayscale produces a much easier-to-follow, less-confusing visualization on which details are easier to spot

Designing Effective Colormaps

- Other colormap designs

- **Two-hue:** The two-hue colormap can be seen as a generalization of the grayscale colormap, where we interpolate between two colors, rather than between black and white.

- **Heat Map:**The intuition behind its colors is that they represent the color of an object heated at increasing temperature values, with black corresponding to low data values, red-orange hues for intermediate data ranges, and yellow-white hues for the high data values respectively

Designing Effective Colormaps

- Other colormap designs
- **Diverging:** Diverging colormaps are constructed starting from two typically isoluminant hues, just as the isoluminant two-hue colormaps. However, rather than interpolating between the end colors $\boldsymbol{c}_{\boldsymbol{min}}$ and $\boldsymbol{c}_{\boldsymbol{max}}$, we now add a third color $\boldsymbol{c}_{\boldsymbol{mid}}$

  - c_{min} = blue, c_{max} = red, and c_{mid} = white
  - c_{min} = green, c_{max} = red, and c_{mid} = bright yellow

Designing Effective Colormaps

- Selecting an optimal colormap is influenced by

– Geometry: Not all colors are equally strongly perceived when displayed on surfaces that have the same area. Perceiving a color accurately is strongly influenced by the colors displayed in neighboring areas.

– User group: While most users would feel comfortable with colormaps containing a large variety of hues, 6 to 10 percent of all men would not be able to correctly separate red from green hues

Designing Effective Colormaps

- Selecting an optimal colormap is influenced by

  – Medium: Another factor in colormap usage is the medium used to present the visualization. Computer screens (CRT or LCD), printed materials (matte or glossy), and projectors have quite different ways of reproducing and displaying color.

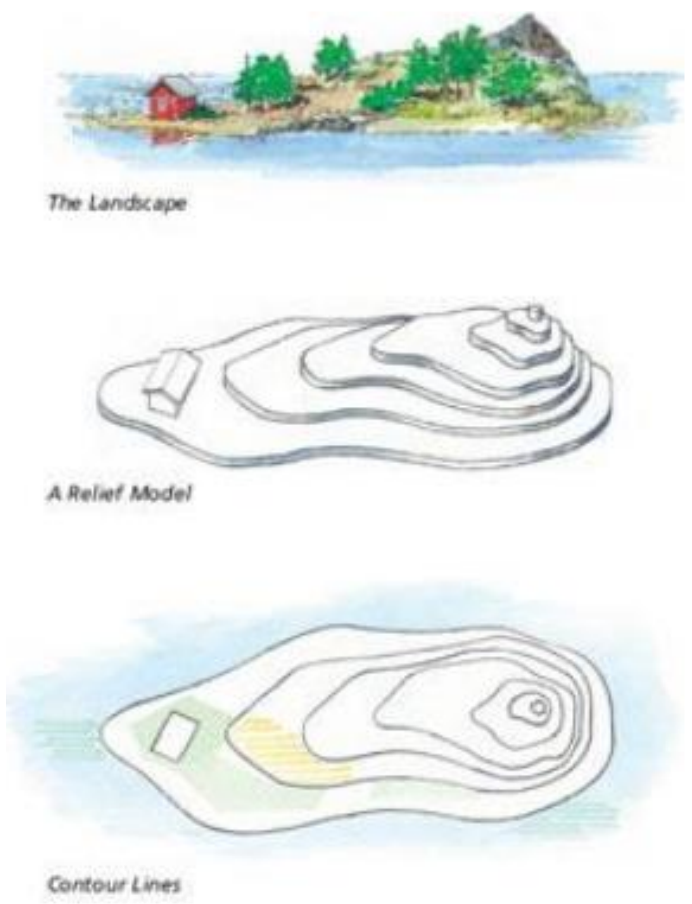
  – User group: While most users would feel comfortable with colormaps containing a large variety of hues, 6 to 10 percent of all men would not be able to correctly separate red from green hues

Contouring

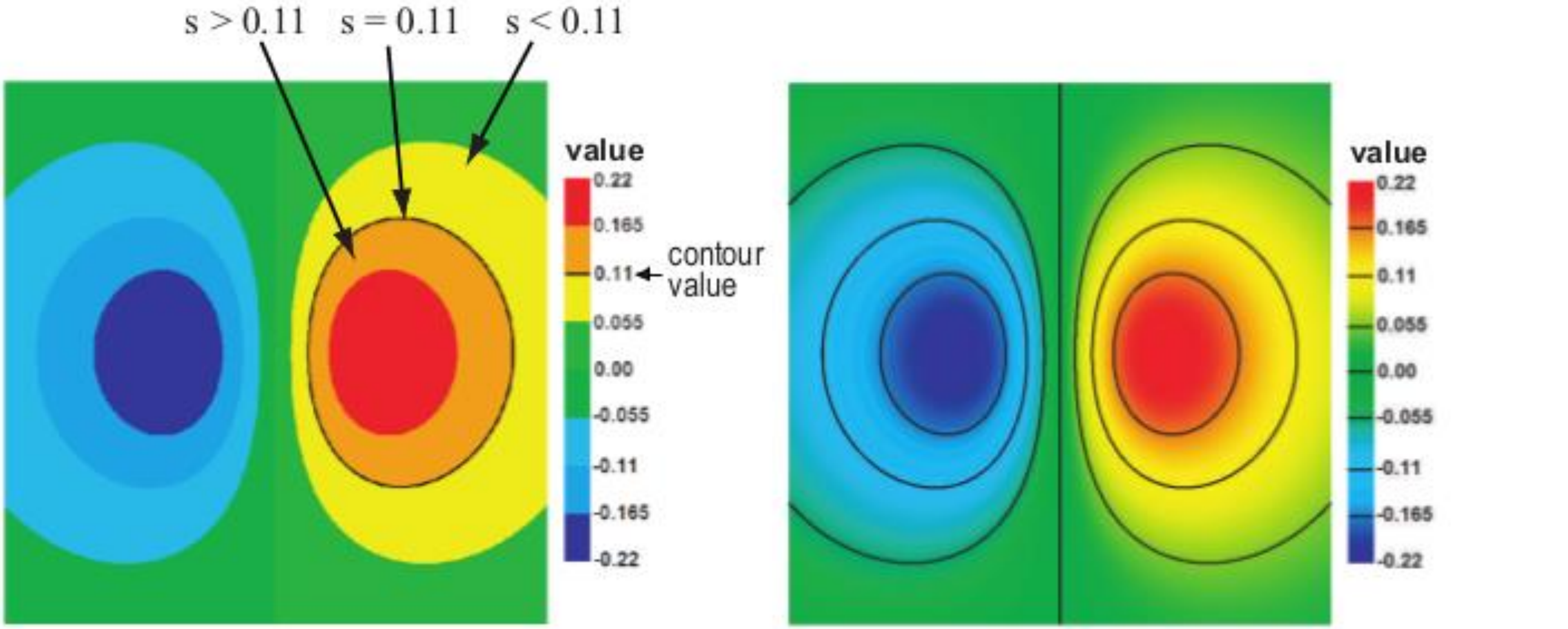
A contour line C is defined as all points p in a dataset D that have the same scalar value, or isovalue.

For 2D dataset, a contour line is called an isoline.

For 3-D dataset, a contour is a 2-D surface, called isosurface.



Contouring

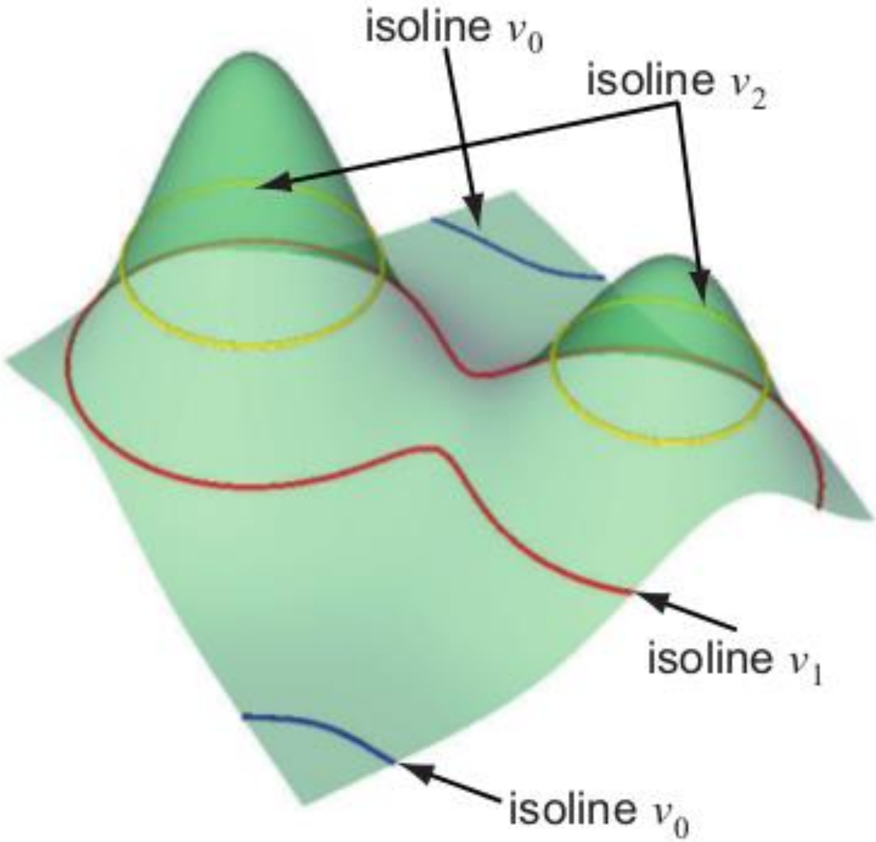


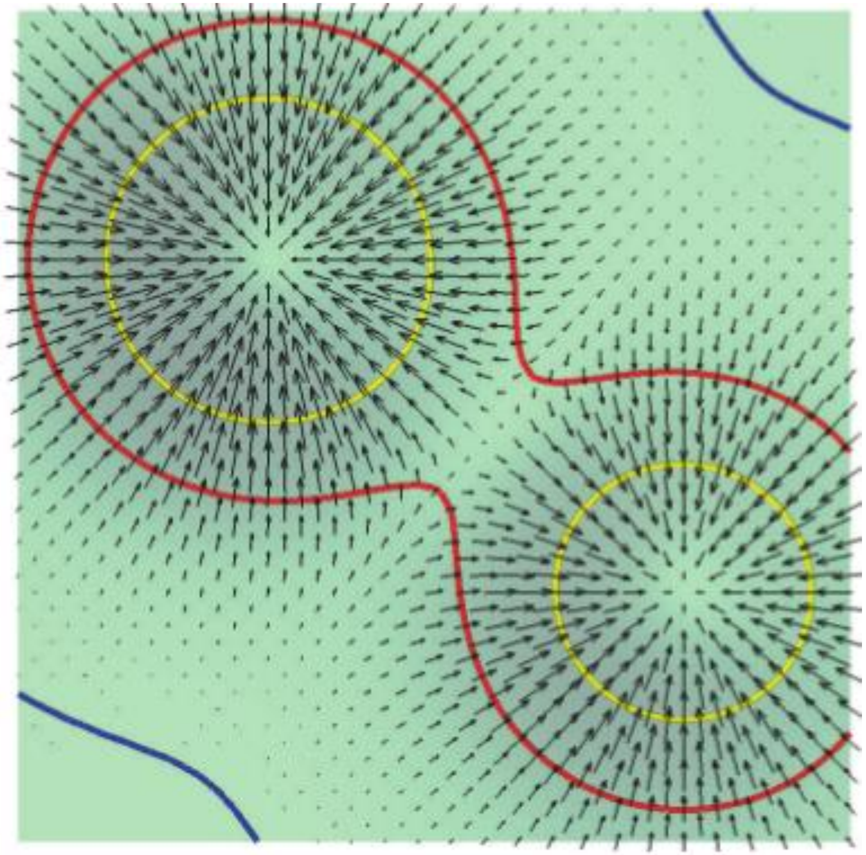
(a) one contour ($s=0.11$) (b) 7 contours ($s \in [-0.165, 0.165]$)

Properties of Contours

- Indicating specific values of interest.
- In the height-plot, a contour line corresponds with the interaction of the graph with a horizontal plane of ***s*** value.
- The tangent to a contour line is the direction of the function’s minimal (zero) variation
- The perpendicular to a contour line is the direction of the function’s maximum variation: the **gradient**
-

Properties of Contours





Thank You