# Diverging Color Maps for Scientific Visualization (Expanded)

Kenneth Moreland

Sandia National Laboratories

**Abstract.** One of the most fundamental features of scientific visualization is the process of mapping scalar values to colors. This process allows us to view scalar fields by coloring surfaces and volumes. Unfortunately, the majority of scientific visualization tools still use a color map that is famous for its ineffectiveness: the rainbow color map. This color map, which naïvely sweeps through the most saturated colors, is well known for its ability to obscure data, introduce artifacts, and confuse users. Although many alternate color maps have been proposed, none have achieved widespread adoption by the visualization community for scientific visualization. This paper explores the use of diverging color maps (sometimes also called ratio, bipolar, or double-ended color maps) for use in scientific visualization, provides a diverging color map that generally performs well in scientific visualization applications, and presents an algorithm that allows users to easily generate their own customized color maps.

## 1 Introduction

At its core, visualization is the process of providing a visual representation of data. One of the most fundamental and important aspects of this process is the mapping of numbers to colors. This mapping allows us to pseudocolor an image or object based on varying numerical data. Obviously, the choice of color map is important to allow the viewer to easily perform the reverse mapping back to scalar values.

**Fig. 1.** The rainbow color map. Know thy enemy.

By far, the most common color map used in scientific visualization is the rainbow color map, shown in Figure 1 which cycles through all of the most saturated colors. In a recent review on the use of color maps, Borland and Taylor [1] find that the rainbow color map was used as the default in 8 out of the 9 toolkits they examined. Borland and Taylor also find that in IEEE Visualization papers from 2001 to 2005 the rainbow color map is used 51 percent of the time.

Despite its popularity, the rainbow color map has been shown to be a poor choice for a color map in nearly all problem domains. This well-studied field of perception shows that the rainbow color map obfuscates, rather than clarifies, the display of data in a variety of ways [1]. The choice of a color map can be a complicated decision that depends on the visualization type and problem domain, but the rainbow color map is a poor choice for almost all of them.

One of the major contributors to the dominance of the rainbow color map is the lack of a clear alternative, especially in terms of scientific visualization. There are many publications that recommend very good choices for color maps [2–5]. However, each candidate has its features and flaws, and the choice of the "right" one is difficult. The conclusion of all these publications is to pick from a variety of color maps for the best choice for a domain-specific visualization. Although this is reasonable for the designer of a targeted visualization application, a general purpose application, designed for multiple problem domains, would have to push this decision to the end-user with a dizzying array of color map choices. In our experience the user, who seldom has the technical background to make an informed decision, usually chooses a rainbow color map.

This paper recommends a good default color map for general purpose scientific visualization. The color map derived here is an all-around good performer: it works well for low and high frequency data, orders the data, is perceptually linear, behaves well for observers with color-deficient vision, and has reasonably low impact on the shading of three-dimensional surfaces.

## 2 Previous Work

This previous work section is divided into two parts. First is a quick review on previously proposed color maps that lists the pros and cons of each. Second is a quick review on color spaces, which is relied upon in subsequent discussions.

### 2.1 Color Maps

As stated previously, the rainbow color map is the most dominate in scientific visualization tools. Based on the colors of light at different wavelengths, the rainbow color map's design has nothing to do with how humans perceive color. This results in multiple problems when humans try to do the reverse mapping from colors back to numbers.

The first problem is that the colors do not follow any natural perceived ordering. Perceptual experiments show that although a test subject with no prior training will always order grayscale colors in order of luminance (in one direction or the other), the test subjects will order rainbow colors in numerous different ways [5].

The second problem is that the perceptual changes in the colors are not uniform. The colors appear to change faster in the cyan and yellow regions, which can cause Mach bands in those regions. The colors appear to change more slowly in the blue, green, and red regions, which creates larger bands of color. These bands can hide important changes in the underlying data. Thus, nonuniform perceptual changes simultaneously introduce artifacts and obfuscate real data [1]. This is demonstrated in Figure 2.

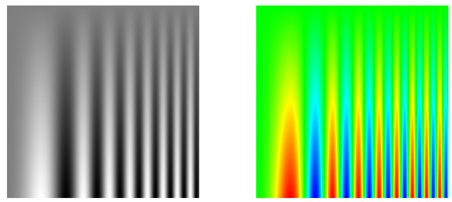


**Fig. 2.** A spatial contrast sensitivity function. The frequency of the function increases from left to right, and the contrast increases from top to bottom. Notice that the grayscale mapping (on the left) faithfully reproduces the function. The rainbow color mapping (on the right) hides the variation in the low contrast region and appears less smooth in the high-contrast, low-frequency region.

A third problem with the rainbow color map is that it is sensitive to deficiencies in vision. Roughly 5% of the population cannot distinguish between the red and green colors. Viewers with color deficiencies cannot distinguish many colors considered "far apart" in the rainbow color map [6].

**Fig. 3.** The grayscale color map.

Better color maps exist. A very simple one is the grayscale color map shown in Figure 3. Completely devoid of any chromaticity, this map relies entirely on luminance to demonstrate the numerical value. Although a very simple map to create and use, this map is surprisingly effective as the human visual system is most sensitive to changes in luminance [5, 7]. The grayscale color map is used heavily in the image processing and medical visualization fields.

The grayscale color map also has disadvantages. One problem is that a human's perception of brightness is subject to the brightness of the surrounding area. Thus, when asked to compare the luminance of two objects separated by distance and background, human subjects err up to 20% [8]. This effect, demonstrated in Figure 4, is called simultaneous contrast [9]. Adding a chromaticity shift helps, but does not fix the problem entirely. A chromatic shift also has the possitive side effect of increasing the dynamic range of the color map.

Another problem with grayscale color maps that is of greater concern for general purpose scientific visualization is its interference with surface shading. The shading of 3D surfaces based on light sources is of utmost importance for perceiving surface shape. These shading cues are composed almost entirely of

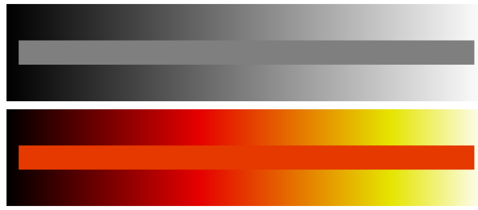


**Fig. 4.** Pixels of the same luminance may look different depending on the surrounding pixels.

luminance shifts. Thus the luminance shift of the grayscale color map masks the surface luminance shifts, especially in the darker part of the map, as demonstrated in Figure 5. The problem cannot be corrected without a major reduction in the range that the luminance shifts in the color map.

**Fig. 5.** Maps with big changes in luminance hide shading cues important for determining 3D structure (left image), whereas isoluminant maps minimize shading interference (right image).

**Fig. 6.** Isoluminant color maps. The green to red color map is popular because it uses a pair of opponent colors, but the cyan to mauve color map is much easier to see by individuals with deuteranope or protanopic vision.

A type of color map that is often suggested for use with 3D surfaces is an isoluminant color map such as those demonstrated in Figure 6. Somewhat opposite to the grayscale map, an isoluminant map maintains a constant (perceptual) luminance and relies entirely on chromatic shifts. An isoluminant color map is theoretically ideal for mapping onto shaded surfaces, as is demonstrated in Figure 5.

Isoluminant color maps are not without their flaws, however. Human perception is less sensitive to changes in saturation or hue than changes in luminance, especially for high frequency data [10]. Holding the luminance constant also restricts the colors that can be represented. Thus, the isoluminant color map will have a lower fidelity than one in which the luminance is allowed to change. Isoluminant color maps also tend to look dull and ugly, so casual users rarely choose one over a more vibrant color map such as the rainbow color map.

These color maps comprise those most commonly used in the literature and tools today. Other color maps are proposed by Ware [5] as well as several others. Most are similar in spirit to those here with uniform changes in luminance, saturation, hue, or some combination thereof.

## 2.2 Color Spaces

All color spaces are based on the tristimulus theory, which states that any perceived color can be uniquely represented by a 3-tuple [11]. This result is a side effect of the fact that there are exactly 3 different types of color receptors in the human eye. Limited space prevents more than a few applicable additive color spaces from being listed here. Any textbook on color will provide more spaces in more detail [11, 12].

The color space most frequently used in computer applications is the RGB color space. This color space is adopted by many graphics packages such as OpenGL and is presented to users by nearly every computer application that provides a color chooser. The three values in the RGB color space refer to the intensity output of each of the three light colors used in a monitor, television, or projector.

Although it is often convenient to use RGB to specify colors in terms of the output medium, the display may have nonlinearities that interfere with the blending and interpolation of colors [11]. When computing physical light effects, it is best to use a color space defined by the physical properties of light. XYZ is a widely used color space defined by physical light spectra. This conversion can be particularly difficult due to differences between displays that make the color definition somewhat ambiguous [13]. For the purposes of this paper, we will assume the RGB space conforms to the canonical monitor defined by the sRGB specification, a standard of the International Electrotechnical Commission (IEC 61966-2-1) that is widely used by many color management programs [11]. The conversion from the sRGB components to RGB components with physically linear properties is given in Equation 1.

$$
\begin{aligned}
R_{\text{Linear}} &= \begin{cases} ((R_{\text{sRGB}} + 0.055)/1.055)^{2.4} & \text{if } R_{\text{Linear}} > 0.04045 \\ R_{\text{sRGB}}/12.92 & \text{otherwise} \end{cases} \\
G_{\text{Linear}} &= \begin{cases} ((G_{\text{sRGB}} + 0.055)/1.055)^{2.4} & \text{if } G_{\text{Linear}} > 0.04045 \\ G_{\text{sRGB}}/12.92 & \text{otherwise} \end{cases} \\
B_{\text{Linear}} &= \begin{cases} ((B_{\text{sRGB}} + 0.055)/1.055)^{2.4} & \text{if } B_{\text{Linear}} > 0.04045 \\ B_{\text{sRGB}}/12.92 & \text{otherwise} \end{cases}
\end{aligned}
\tag{1}
$$

Given RGB values that are linear with respect to physical light intensity, conversion to XYZ space is a simple linear transformation. The transformation is dependent on the characteristics of the display for which the RGB space is defined, but the sRGB specification yeilds the one in Equation 2.

$$[X \quad Y \quad Z] = [R \quad G \quad B] \begin{bmatrix} 0.4124 & 0.2126 & 0.0193 \\ 0.3576 & 0.7152 & 0.1192 \\ 0.1805 & 0.0722 & 0.9505 \end{bmatrix} \tag{2}$$

There is a nonlinear relationship between light intensity and color perception. When defining a color map, we are more interested in how a color is perceived than how it is formed. In these cases, it is better to use a color map based on how humans perceive color. CIELAB and CIELUV are two common spaces. The choice between the two is fairly arbitrary; this paper uses CIELAB. The conversion from XYZ to CIELAB is given in Equation 3.

$$\begin{aligned} L* &= 116\,[\mathrm{f}(Y/Y_n) - 16/116] \\ a* &= 500\,[\mathrm{f}(X/X_n) - \mathrm{f}(Y/Y_n)] \\ b* &= 200\,[\mathrm{f}(Y/Y_n) - \mathrm{f}(Z/Z_n)] \\ \mathrm{f}(x) &\equiv \begin{cases} x^{1/3} & \text{if } x > 0.008856 \\ 7.787x + 16/116 & \text{if } x \le 0.008856 \end{cases} \\ [X_n \quad Y_n \quad Z_n] & \text{ is a reference white value} \end{aligned} \tag{3}$$

CIELAB is an approximation of how humans perceive light. The Euclidean distance between two points is the approximate perceived difference between the two colors. This Euclidean distance in CIELAB space is known as $\Delta E$ and makes a reasonable metric for comparing color differences [12]. This paper uses the notation $\Delta E\{\mathbf{c_1}, \mathbf{c_2}\}$ to denote the $\Delta E$ for the pair of colors $\mathbf{c_1}$ and $\mathbf{c_2}$.

## 3 Color Map Requirements

Our ultimate goal is to design a color map that works well for general-purpose scientific visualization and a wide variety of tasks and users. As such we have the following requirements. These criteria conform to many of those proposed previously [3, 6, 13].

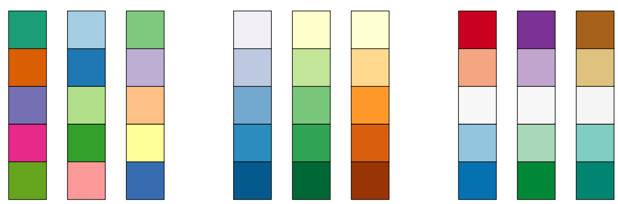
- The map yields images that are aesthetically pleasing.
- The map has a maximal perceptual resolution.
- Interference with the shading of 3D surfaces is minimal.
- The map is not sensitive to vision deficiencies.
- The order of the colors should be intuitively the same for all people.
- The perceptual interpolation matches the underlying scalars of the map.

The reasoning behind most of these requirements is self explanatory. The requirement that the color map be "pretty," however, is not one often found in the scientific literature. After all, the attractiveness of the color map, which is difficult to quantify in the first place, has little to do with its effectiveness in conveying information. Nevertheless, aesthetic appeal is important as users will use that as a criterion in selecting visualization products and generating images.

Several of these requirements are contradictory, making the choice of a general purpose color map difficult. All of the examples in Section 2.1 excel in some of the requirements, but fail completely in one or more of the others. It is impossible to have a color map that performs perfectly against all of the requirements. Our color map must be a compromise that works reasonably well in all areas.

## 4 Color Map Design

There are many color maps in existence today, but very few of them satisfy all of the requirements listed in Section 3. For inspiration, we look at the field of cartography. People have been making maps for thousands of years, and throughout this history there has been much focus on both the effectiveness of conveying information as well as the aesthetics of the design. Brewer [2] provides excellent advice for designing cartographic color maps and many well-designed examples.[1]



(a) Qualitative (b) Sequential (c) Diverging

**Fig. 7.** Examples of color maps from Brewer [2].

Brewer divides her color maps into three classes: qualitative, sequential, and diverging. Examples of these color maps are shown in Figure 7. The qualitative color maps (also known as nominal color maps [5]) are used to represent a collection of discrete, unordered classes. Since the colors have no ordering (by design), they are not appropriate for mapping a scalar variable.

The sequential color maps (also known as ordinal [5] or saturation [4] color maps) are (nearly) monochromatic. They range from a heavily saturated color to various levels of unsaturation. Luminance is also often increased as saturation is decreased so that the color map terminates in a color at or close to white. The monotonic nature of the saturation level maps well to a scalar value.

The diverging color maps (also known as ratio [5], bipolar [14], or double-ended [4]) have two major color components. The map transitions from one

[1] Brewer's color maps are also available on her web site: www.colorbrewer.org.

color component to the other by passing through an unsaturated color (white or yellow). Diverging color maps are typically used to represent a scalar with a significant value at or near the median. For example, a color map for elevation could put sea level at white with below sea level in blue and above sea level in tan [15]. The ordering of the colors is usually based on the context within which they are used.

Sequential color maps are clearly appropriate for scientific visualization. Their monotonic nature maps well to scalar values. Diverging color maps are a less obvious choice. We cannot expect there to be some significant median value that the diverging color map is designed to highlight.

However, diverging color maps can better satisfy the requirements given in Section 3 than their sequential counterparts. First, the more colorful nature of the diverging color map can be more aesthetically pleasing. Second, the diverging color map can have up to twice the perceptual resolution of the sequential color map without sacrificing the requirements of surface shading or losing viewers with dichromatic vision. Furthermore, the diverging color map visually divides scalar values into three logical regions: low, midrange, and high values. These regions provide more visual cues that are helpful for understanding data.

What diverging color maps lack in general is a natural ordering of colors. To impose a color ordering, we carefully chose two colors that most naturally have "low" and "high" connotations. We achieve this with the concept of "cool" and "warm" colors.

Studies show that people identify red and yellow colors as warm and blue and blue-green colors as cool across subjects, contexts, and cultures. Furthermore, people associate warmth with positive activation and coolness with negative activation [16]. Consequently, mapping cool blues to low values and warm reds to high values is natural [13].

### 4.1 Perceptual Uniformity

An important characteristic of any color map is that it is perceptually uniform throughout. For a discrete color map, perceptual uniformity means that all pairs of adjacent colors will look equally different from each other. That is, the $\Delta E$ for each adjacent pair is (roughly) the same.

For a continuous color map, we want the perceptual distance between two colors to be proportional to the distance between the scalars associated with each. If we characterize our color map with function $\mathbf{c}(x)$ that takes scalar value $x$ and returns a color vector, the color map is perceptually uniform if

$$\frac{\Delta E\{\mathbf{c}(x), \mathbf{c}(x + \Delta x)\}}{\Delta x} \tag{4}$$

is constant for all valid $x$.

Strictly speaking, we cannot satisfy Equation 4 for diverging color maps because the map necessarily passes through three points in CIELAB space that are not in a line. However, it is possible to ensure that the rate of change is constant. That is,

$$\lim_{\Delta x \to 0} \frac{\Delta E\{\mathbf{c}(x), \mathbf{c}(x + \Delta x)\}}{\Delta x} \tag{5}$$

is constant for all valid $x$. This relaxed property is sufficient for describing a perceptually linear color map so long as we make sure that the curve does not return to any set of colors.

We can resolve Equation 5 a bit by applying the $\Delta E$ operation and splitting up the $\mathbf{c}$ function into its components ($\Delta E\{\mathbf{c_1}, \mathbf{c_2}\} = \|\mathbf{c_1} - \mathbf{c_2}\| = \sqrt{\sum_i (c_{1i} - c_{2i})^2}$).

$$\lim_{\Delta x \to 0} \frac{\|\mathbf{c}(x + \Delta x) - \mathbf{c}(x)\|}{\Delta x} \tag{6}$$

$$\lim_{\Delta x \to 0} \left\| \frac{\mathbf{c}(x + \Delta x) - \mathbf{c}(x)}{\Delta x} \right\|$$

$$\lim_{\Delta x \to 0} \sqrt{\sum_i \left( \frac{\mathrm{c}_i(x + \Delta x) - \mathrm{c}_i(x)}{\Delta x} \right)^2}$$

$$\sqrt{\sum_i \left( \lim_{\Delta x \to 0} \frac{\mathrm{c}_i(x + \Delta x) - \mathrm{c}_i(x)}{\Delta x} \right)^2} \tag{7}$$

In the final form of Equation 7, we can clearly see that the limit is the definition of a derivative. So replacing the limit with a derivative, we get the following.

$$\sqrt{\sum_i (\mathrm{c}'_i(x))^2} \tag{8}$$

With some abuse of notation, let us declare $\mathbf{c}'(x)$ as the piecewise derivative of $\mathbf{c}(x)$. Using this notation, the constant rate of change requirement reduces to the following.

$$\|\mathbf{c}'(x)\| \tag{9}$$

The easiest way to ensure that Equation 9 is constant is to linearly interpolate colors in the CIELAB color space. However, that is not entirely possible to do for diverging color maps. Lines from red to blue will not go through white. A piecewise linear interpolation is mostly effective, but can create an artificial Mach band at white where the luminance sharply transitions from increasing to decreasing as demonstrated in Figure 8.

Having this sharp transition is fine, perhaps even desirable, when the white value has special significance, but to use the divergent color map in general situations we require a "leveling off" of the luminance as the color map approaches white. To compensate, the chromaticity must change more dramatically in this part of the color map. A method for designing this type of color map is defined in the next section.

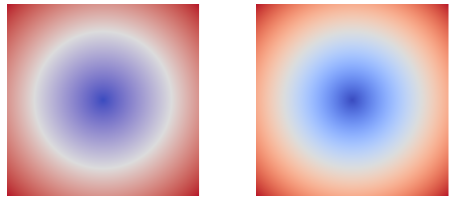


**Fig. 8.** Using piecewise linear interpolations in CIELAB color space causes Mach bands in the white part of diverging color maps (left image). The transition can be softened by interpolating in Msh space (right image).

### 4.2 Msh Color Space

To simplify the design of continuous, diverging color maps, we derive a new color space called Msh. Msh is basically a polar form of the CIELAB color space. $M$ is the magnitude of the vector, $s$ (the saturation) is the angle away from the $L*$ axis, and $h$ (the hue) is the angle of the vector's projection in the $a*$-$b*$ plane. Conversion between the two color spaces is straightforward.

$$\begin{aligned} M &= \sqrt{L*^2 + a*^2 + b*^2} \\ s &= \arccos \frac{L*}{M} \\ h &= \arctan \frac{b*}{a*} \end{aligned} \qquad \begin{aligned} L* &= M \cos s \\ a* &= M \sin s \cos h \\ b* &= M \sin s \sin h \end{aligned} \tag{10}$$

Note that Msh, like all polar coordinates, has a pole in which one of the coordinates is ill defined. Specifically, when $s = 0$ (the color is on the $L*$ axis), $h$ has no effect. This pole was chosen because it coincides with a singularity in human vision. When saturation is low, the color has no hue. It is therefore possible to make a discontinuous jump in the hue while still maintaining perceptual continuance.

Piecewise linear interpolations in Msh space behave very well for diverging color maps. As $s$ linearly approaches zero, the luminance naturally levels out while the chromaticity changes faster.

An ideal way to build a diverging color map in Msh space is to start at one color, linearly reduce $s$ to 0 (to get white), flip $h$ to the appropriate value for the last color, and then linearly increase $s$ to the desired value. In fact, we can show that if $s$ changes linearly while $M$ and $h$ are held constant, Equation 9 is constant, which is our criterion for a uniform color map. We can characterize a $\mathbf{c}(x)$ that behaves in this way in CIELAB space as

$$\mathbf{c}(x) = [M \cos \mathrm{s}(x) \quad M \sin \mathrm{s}(x) \cos h \quad M \sin \mathrm{s}(x) \sin h] \tag{11}$$

where $M$ and $h$ are constant and $\mathrm{s}(x)$ is a linear function of slope $s_m$.

To show that linear saturation changes in Msh are perceptually linear, we plug Equation 11 into Equation 9 and resolve to show that perceptual changes are indeed constant.

$$\begin{aligned}
\|\mathbf{c}'(x)\| &= \|M s_m \sin \mathrm{s}(x) \quad M s_m \cos \mathrm{s}(x) \cos h \quad M s_m \cos \mathrm{s}(x) \sin(h)\| \\
&= \sqrt{M^2 s_m^2 \left(\sin^2 \mathrm{s}(x) + \cos^2 \mathrm{s}(x) \cos^2 h + \cos^2 \mathrm{s}(x) \sin^2 h\right)} \\
&= \sqrt{M^2 s_m^2 \left(\sin^2 \mathrm{s}(x) + \cos^2 \mathrm{s}(x) \left(\cos^2 h + \sin^2 h\right)\right)} \\
&= \sqrt{M^2 s_m^2 \left(\sin^2 \mathrm{s}(x) + \cos^2 \mathrm{s}(x)\right)} \\
&= M s_m \qquad (12)
\end{aligned}$$

Clearly Equation 12 resolves to a constant and therefore meets our criterion for a "uniform" color space. There is still a discontinuity when we flip $h$. However, because this discontinuous change of hue occurs when there is no saturation, it is not noticeable. And unlike the piecewise linear interpolation in CIELAB space, this piecewise linear interpolation in Msh space results in a smooth change in luminance throughout the entire color map, as evidenced in Figure 9.

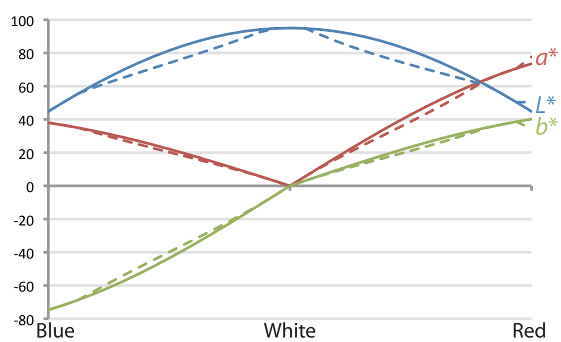



**Fig. 9.** Plot of $L*$, $a*$, and $b*$ coordinates when interpolating $s$ linearly in Msh space. Dashed lines show effect when colors are "clipped" to the gamut of a typical monitor.

A common problem we run into with interpolating in Msh space is that the interpolated colors often leave the gamut of colors displayable by a video monitor. When trying to display many of these colors, you must "clip" to what can be represented. This clipping can lead to noticable artifacts in the color map.

We have two techniques for picking interpolation points. The first is to uniformly reduce the $M$ of each points. Dropping $M$ will bring all the interpolated colors toward the gamut of displayable colors.

Although you will always be able to pull all the colors within the display gamut by reducing $M$, it usually results in colors that are too dim. Thus, a second technique we can do is to allow $M$ to be smaller for the endpoints than for the middle white color. This breaks the uniformity of the color map because a smaller $M$ will mean that a change in $s$ will have a smaller effect. We can restore the uniformity of the color map again by adding some "spin" to the hue. Even though $h$ is interpolated linearly, the changes have a greater effect on the color when $s$ is larger, which can counterbalance the growing $M$ (although a large change can still cause a noticeable pointing of the luminance). The next section describes how to choose an appropriate hue change.

### 4.3 Choosing a Hue Spin

Let us consider the transition from a saturated color, $\mathbf{c}_s = (M_s, s_s, h_s)$, at an end of the color map to an unsaturated "white" color, $\mathbf{c}_u = (M_u, 0, h_u)$, at the middle of the color map. As the color map moves from $\mathbf{c}_s$ to $\mathbf{c}_u$, the $M$, $s$, and $h$ coordinates are varied linearly. The slope of these coordinates can be characterized as $M_m = M_u - M_s$, $s_m = -s_s$, and $h_m = h_u - h_s$. (Note that $h_u$ has no effect on the unsaturated color, but is provided to conveniently define the rate of change.)

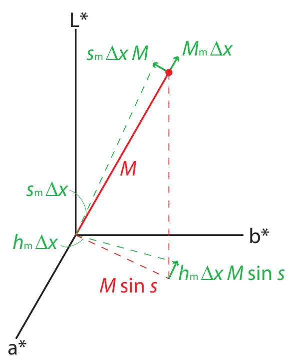



**Fig. 10.** A small linear movement in Msh space. The three axes, $L*$, $a*$, and $b*$, refer to the three dimensions in CIELAB space. Linear movements in Msh space (a polar version of CIELAB) result in nonlinear movements of the CIELAB coordinates.

Figure 10 shows how a small movement in this linear Msh function behaves in CIELAB space. The distance measurements take advantage of the property that if you rotate a vector of radius $r$ by some small angle $\Delta\alpha$, then the change in the vector is $\lim_{\Delta\alpha \to 0} r\Delta\alpha$. Clearly the $\Delta E$, the magnitude of change in CIELAB space, is

$$\sqrt{(M_m \Delta x)^2 + (s_m \Delta x M)^2 + (h_m \Delta x M \sin s)^2} \tag{13}$$

Equation 13 will not be constant unless $M_m$ and $h_m$ are zero, which, as described in the previous section, is unacceptable. However we can get pretty close to constant by choosing $h_u$ so that Equation 13 is equal for $\mathbf{c}_s$ and $\mathbf{c}_u$.

$$\sqrt{(M_m \varDelta x)^2 + (s_m \varDelta x M_s)^2 + (h_m \varDelta x M_s \sin s_s)^2} = \sqrt{(M_m \varDelta x)^2 + (s_m \varDelta x M_u)^2} \tag{14}$$

Note that the right side of Equation 14 is missing a term because it evaluates to 0 for the unsaturated color. We can safely get rid of the square roots because there is a sum of square real numbers inside them both.

$$\begin{aligned}
(M_m \varDelta x)^2 + (s_m M_s)^2 + (h_m M_s \sin s_s)^2 &= (M_m \varDelta x)^2 + (s_m M_u)^2 \\
h_m^2 M_s^2 \sin^2 s_s &= s_m^2 (M_u^2 - M_s^2) \\
h_m^2 &= \frac{s_m^2 (M_u^2 - M_s^2)}{M_s^2 \sin^2 s_s} \\
h_m &= \pm \frac{s_m \sqrt{M_u^2 - M_s^2}}{M_s \sin s_s} \qquad (15)
\end{aligned}$$

Remember that $s_m = -s_s$. We can use Equation 15 to determine a good hue to use for the white point (from the given side).

$$h_u = h_s \pm \frac{s_s \sqrt{M_u^2 - M_s^2}}{M_s \sin s_s} \tag{16}$$

Note that Equation 16 will most certainly yield a different value for each of the saturated colors used in the diverging color map. The direction in which the hue is "spun" is unimportant with regard to perception. The examples here adjust the hue to be away from 0 (except in the purple hues) because it provides slightly more aesthetically pleasing results. Figure 11 gives a simple algorithm for adjusting the hue.

AdjustHue($\{M_{\text{sat}}, s_{\text{sat}}, h_{\text{sat}}\}, M_{\text{unsat}}$)

**if** $M_{\text{sat}} \geq M_{\text{unsat}}$
  **then return** $h_{\text{sat}}$ ▷ Best we can do
  **else** $hSpin \leftarrow \frac{s_{\text{sat}} \sqrt{M_{\text{unsat}}^2 - M_{\text{sat}}^2}}{M_{\text{sat}} \sin(s_{\text{sat}})}$
    **if** $h_{\text{sat}} > -\frac{\pi}{3}$ ▷ Spin away from purple
      **then return** $h_{\text{sat}} + hSpin$
      **else return** $h_{\text{sat}} - hSpin$

**Fig. 11.** Function to provide an adjusted hue when interpolating to an unsaturated color in Msh space.

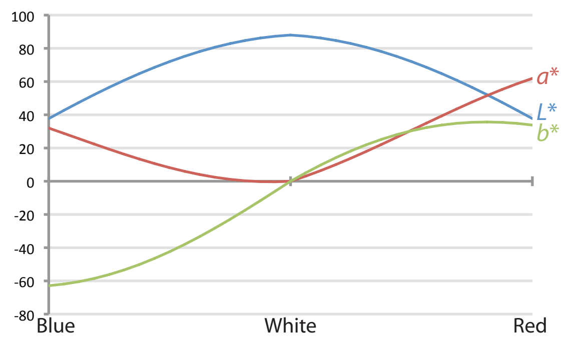



**Fig. 12.** Plot of $L*$, $a*$, and $b*$ coordinates when interpolating piecewise linearly in Msh space and applying a hue spin. Notice the amplified bend in the $a*$ and $b*$ curves, which determine hue.

Figure 12 shows the effects of applying a hue spin. The plot is smooth, perceptually uniform, and comparable to the plot given in Figure 9. In addition, the hue spin allows the entire curve to remain in the gamut of displayable colors.

### 4.4 Interpolating Control Points

The InterpolateColor algorithm in Figure 13 combines the techniques described previously in this section. The algorithm simplifies the process of building a continuous diverging color map by allowing a user to define colors at control points. InterpolateColor accepts two colors and an interpolation factor between 0 and 1. InterpolateColor takes colors in RGB space to make it easier for users to define.

The InterpolateColor algorithm works as follows. The colors are first converted to Msh.[2] To enforce a diverging color map, white is added between the two control points (lines 3–9). The middle white points is not added if either color is already unsaturated (indicating that there is already a control point for the white part of the diverging color map) or if the angular difference between the two hue orientations (computed by RadDiff) is small (which would mean that both sides of the diverging color map would be roughly the same color).[3] If either control point is unsaturated, its hue is adjusted (lines 10–13) using the AdjustHue function in Figure 11. Finally, the two Msh colors are linearly interpolated (line 14). The result is converted back to RGB and returned.

---

[2] The implementations of RGB2Msh and Msh2RGB can be derived from Equations 2, 3, and 10.

[3] The parameters for specifying a low saturation (less than 0.05 radians) and similar hue angles (less than $\frac{\pi}{3}$ radians) is somewhat arbitrary, but the values provided here work well in practice.

InterpolateColor($\{r_1, g_1, b_1\}, \{r_2, g_2, b_2\}, interp$)

$\{M_1, s_1, h_1\} \leftarrow$ RGB2Msh($\{r_1, g_1, b_1\}$)
$\{M_2, s_2, h_2\} \leftarrow$ RGB2Msh($\{r_2, g_2, b_2\}$)
▷ If points saturated and distinct, place white in middle
**if** $(s_1 > 0.05) \cap (s_2 > 0.05) \cap (\text{RadDiff}(h_1, h_2) > \frac{\pi}{3})$
  **then** $M_{\text{mid}} \leftarrow \max(M_1, M_2, 88)$
    **if** $interp < \frac{1}{2}$
      **then** $M_2 \leftarrow M_{\text{mid}}, s_2 \leftarrow 0, h_2 \leftarrow 0$
        $interp \leftarrow 2\, interp$
      **else** $M_1 \leftarrow M_{\text{mid}}, s_1 \leftarrow 0, h_1 \leftarrow 0$
        $interp \leftarrow 2\, interp - 1$
▷ Adjust hue of unsaturated colors
**if** $(s_1 < 0.05) \cap (s_2 > 0.05)$
  **then** $h_1 \leftarrow$ AdjustHue($\{M_2, s_2, h_2\}, M_1$)
**elseif** $(s_2 < 0.05) \cap (s_1 > 0.05)$
  **then** $h_2 \leftarrow$ AdjustHue($\{M_1, s_1, h_1\}, M_2$)
▷ Linear interpolation on adjusted control points
$\{M_{\text{mid}}, s_{\text{mid}}, h_{\text{mid}}\}$
  $\leftarrow (1 - interp)\{M_1, s_1, h_1\} + interp\{M_2, s_2, h_2\}$
**return** Msh2RGB($\{M_{\text{mid}}, s_{\text{mid}}, h_{\text{mid}}\}$)

**Fig. 13.** Interpolation algorithm to automatically create continuous diverging color maps.

The InterpolateColor algorithm makes it easy for users to build and modify continuous diverging color maps using RGB control points as demonstrated in Figure 14.

(a) Simply pick two endpoint colors to create a diverging color map in between them.

(b) Creating a white or gray control point allows you to define the intensity and location of the "middle" of the diverging color map.

(c) Adding a control point in a colored area allows you to stretch and compress regions.

**Fig. 14.** Interacting with color maps in the Msh color space.

## 5 Results

**Fig. 15.** A continuous diverging color map well suited to scientific visualization.

Applying the design described in Section 4, we can build the cool to warm color map shown in Figure 15. The control points, to be interpolated in Msh space, are given in Table 1. Table 2 gives example interpolated RGB values for this color map. RGB values are computed from CIELAB values using a D65 white point.

**Table 1.** Cool to warm color map control points.

| Color | M | s | h |
|---|---|---|---|
| Red | 80 | 1.08 | 0.5 |
| White | 88 | 0 | 1.061/-1.661 |
| Blue | 80 | 1.08 | -1.1 |

**Table 2.** Cool to warm color map RGB values.

| Scalar | Red | Green | Blue | Scalar | Red | Green | Blue |
|---|---|---|---|---|---|---|---|
| 0.0 | 59 | 76 | 192 | 0.53125 | 229 | 216 | 209 |
| 0.03125 | 68 | 90 | 204 | 0.5625 | 236 | 211 | 197 |
| 0.0625 | 77 | 104 | 215 | 0.59375 | 241 | 204 | 185 |
| 0.09375 | 87 | 117 | 225 | 0.625 | 245 | 196 | 173 |
| 0.125 | 98 | 130 | 234 | 0.65625 | 247 | 187 | 160 |
| 0.15625 | 108 | 142 | 241 | 0.6875 | 247 | 177 | 148 |
| 0.1875 | 119 | 154 | 247 | 0.71875 | 247 | 166 | 135 |
| 0.21875 | 130 | 165 | 251 | 0.75 | 244 | 154 | 123 |
| 0.25 | 141 | 176 | 254 | 0.78125 | 241 | 141 | 111 |
| 0.28125 | 152 | 185 | 255 | 0.8125 | 236 | 127 | 99 |
| 0.3125 | 163 | 194 | 255 | 0.84375 | 229 | 112 | 88 |
| 0.34375 | 174 | 201 | 253 | 0.875 | 222 | 96 | 77 |
| 0.375 | 184 | 208 | 249 | 0.90625 | 213 | 80 | 66 |
| 0.40625 | 194 | 213 | 244 | 0.9375 | 203 | 62 | 56 |
| 0.4375 | 204 | 217 | 238 | 0.96875 | 192 | 40 | 47 |
| 0.46875 | 213 | 219 | 230 | 1.0 | 180 | 4 | 38 |
| 0.5 | 221 | 221 | 221 | . | | | |

This diverging color map works admirably for all of our requirements outlined in Section 3. The colors are aesthetically pleasing, the order of the colors is natural, the rate of change is perceptually linear, and the colors are still easily distinguished by those with dichromatic vision. The map also has a good perceptual range and minimally interferes with shading.

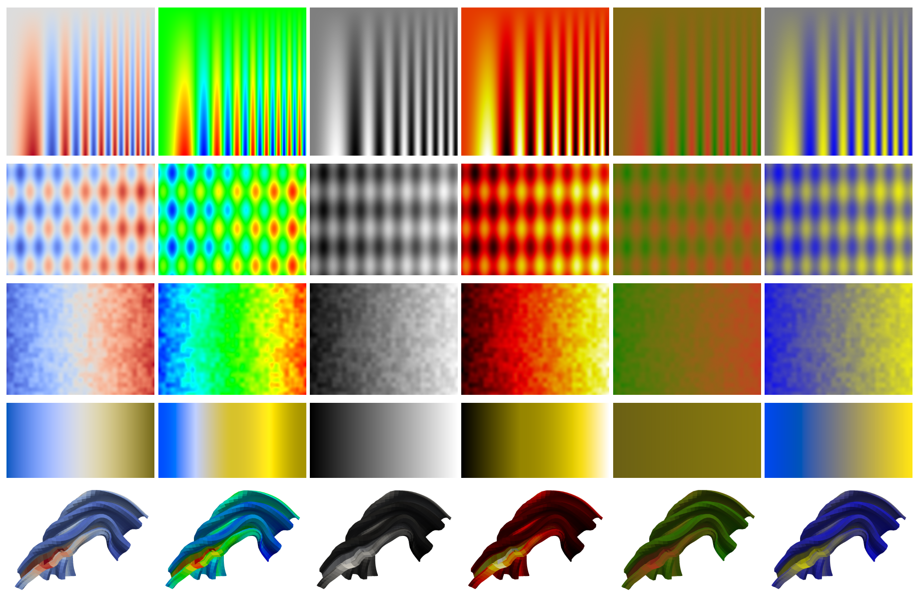


**Fig. 16.** Comparison of color map effectiveness. The color maps are, from left to right, cool-warm, rainbow, grayscale, heated body, isoluminant, and blue-yellow. The demonstrations are, from top to bottom, a spatial contrast sensitivity function, a low-frequency sensitivity function, high-frequency noise, an approximation of the color map viewed by someone with deuteranope color-deficient vision (computed with Vischeck), and 3D shading.

Figure 16 compares the cool-warm color map to some common alternatives as well as some recommended by Rheingans [4] and Ware [5]. The cool-warm color map works well in all the cases demonstrated here. The rainbow color map exhibits problems with irregular perception and sensitivity to color deficiencies. The grayscale and heated-body color maps work poorly in conjunction with 3D shaded surfaces. The isoluminant color map has a low dynamic range and performs particularly poorly with high frequency data. The common choice of greed-red isoluminant color maps is also useless to most people with color-deficient vision. The blue-yellow map works reasonably well in all these cases, but has a lower resolution than the cool-warm map, which yields poorer results with low contrast.

In addition, despite having a relatively large perceptual response, the color map still allows for a significant amount of annotation or visual components to be added, as shown in Figure 17.

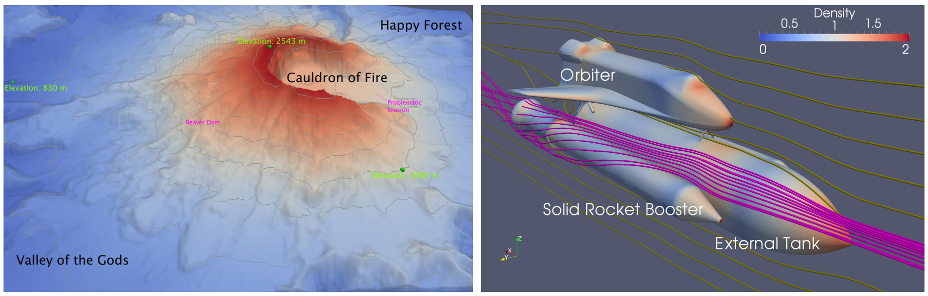



**Fig. 17.** Examples of using the color map in conjunction with multiple other forms of annotation.

Using the techniques described in Section 4, we can also design continuous diverging color maps with different colors. Such color maps may be useful in domain-specific situations when colors have specific meaning. Some examples are given in Figure 18.

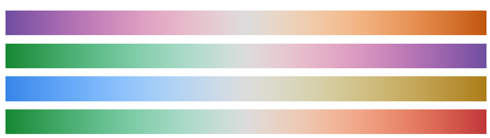


**Fig. 18.** Further examples of color maps defined in Msh space.

An implementation of using the Msh color space to create diverging maps has been added to the `vtkColorTransferFunction` class in the Visualization Toolkit (VTK), a free, open-source scientific visualization library.[4] Any developers or users of scientific visualization software are encouraged to use these color map building tools for their own needs.

This diverging color map interpolation has also been added to ParaView, a free, open-source end-user scientific visualization application,[5] and was first featured in the 3.4 release in October 2008. Although ParaView does let users change the color map and there is no way to track who does so, in our experience few users actually do this. In the nearly 3000 messages on the ParaView users'

[4] www.vtk.org
[5] www.paraview.org

mailing list from October 2008 to July 2009, there was no mention of the change of color map from rainbow to cool-warm diverging. Users seem to have accepted the change with little notice despite most users' affinity for rainbow color maps.

## 6 Discussion

This paper provides a color map that is a good all-around performer for scientific visualization. The map is an effective way to communicate data through colors. Because its endpoints match those of the rainbow color map most often currently used, it can be used as a drop-in replacement.

Diverging color maps have not traditionally been considered for most scientific computing due to their design of a "central" point, which was originally intended to have some significance. However, with the addition of the Msh color space, the central point becomes a smooth neutral color between two other colors. The middle point serves as much to highlight the two extremes as it does to highlight itself. In effect, the divergent color map allows us to quickly identify whether values are near extrema and which extrema they are near.

This paper also provides an algorithm to generate new continuous diverging color maps. This interaction is useful for applying colors with domain specific meaning or for modifying the scaling of the colors.

Although we have not been able to do user studies, the design of this color map is based on well established theories on color perception. This map is a clear improvement over what is commonly used today, and I hope that many will follow in adopting it.

## 7 Acknowledgements

Thanks to Russell M. Taylor II for his help on color space and color map design. Thanks to Patricia Crossno, Brian Wylie, Timothy Shead, and the ParaView development team for their critical comments and general willingness to be guinea pigs.

This work was done at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

## References

1. Borland, D., Taylor II, R.M.: Rainbow color map (still) considered harmful. IEEE Computer Graphics and Applications **27** (2007) 14–17
2. Brewer, C.A.: Designing better MAPS: A Guide for GIS Users. ESRI Press (2005) ISBN 1-58948-089-9.
3. Levkowitz, H., Herman, G.T.: Color scales for image data. IEEE Computer Graphics and Applications **12** (1992) 72–80

4. Rheingans, P.: Task-based color scale design. In: Proceedings of Applied Image and Pattern Recognition '99. (1999) 35–43
5. Ware, C.: Information Visualization: Perception for Design. 2nd edn. Morgan Kaufmann (2004) ISBN 1-55860-819-2.
6. Light, A., Bartlein, P.: The end of the rainbow? Color schemes for improved data graphics. EOS, Transactions, American Geophysical Union **85** (2004) 385, 391
7. Mullen, K.T.: The contrast sensitivity of human colour vision to red–green and blue–yellow chromatic gratings. The Journal of Physiology **359** (1985) 381–400
8. Ware, C.: Color sequences for univariate maps: Theory, experiments, and principles. IEEE Computer Graphics and Applications **8** (1988) 41–49
9. Stone, M.C.: Representing colors as three numbers. IEEE Computer Graphics and Applications **25** (2005) 78–85
10. Rogowitz, B.E., Treinish, L.A., Bryson, S.: How not to lie with visualization. Computers in Physics **10** (1996) 268–273
11. Stone, M.C.: A Field Guide to Digital Color. A K Peters (2003) 1-56881-161-6.
12. Wyszecki, G., Stiles, W.: Color Science: Concepts and Methods, Quantitative Data and Formulae. John Wiley & Sons, Inc. (1982) ISBN 0-471-02106-7.
13. Fortner, B., Meyer, T.E.: Number by Colors: a Guide to Using Color to Understand Technical Data. Springer-Verlag (1997) ISBN 0-387-94685-3.
14. Spence, I., Efendov, A.: Target detection in scientific visualization. Journal of Experimental Psychology: Applied **7** (2001) 13–26
15. Tufte, E.R.: Visual Explanations. Graphics Press (1997) ISBN 0-9613921-2-6.
16. Hardin, C., Maffi, L., eds.: Color categories in thought and language. Cambridge University Press (1997) ISBN 0-521-49800-7.