Brief introduction in movies with gnuplot / colour schemes

Ingo Thies
@ Argelander-Institut für Astronomie
Bonn

Technical Seminar 2012-07-06

Part I: Colour basics

Colour can enhance graphs in several ways:

- transports additional information
- helps to distinguish small differences in value
- makes visualisations more attractive (any thus may help to advertise for your work!)

But too much or wrong colours may flip the best to the worst!

Q: Do you think that the ‘rainbow’ is a good scheme?

CIE 1931 Color space

Given a spectral intensity distribution $I(\lambda)$. Then

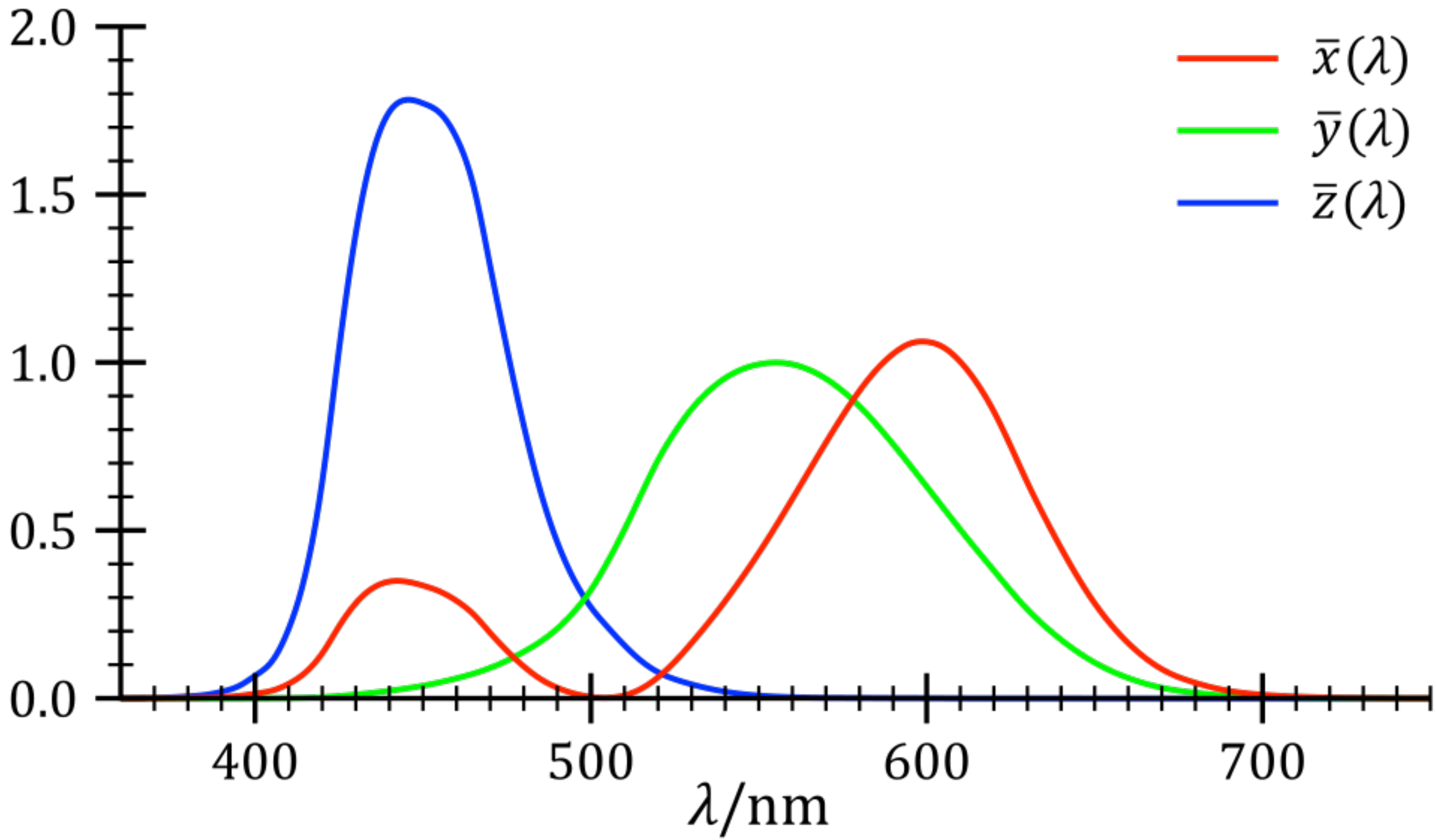
$$X = \int_0^\infty I(\lambda)\,\overline{x}(\lambda)\,d\lambda \quad \text{(same for Y, Z)}$$

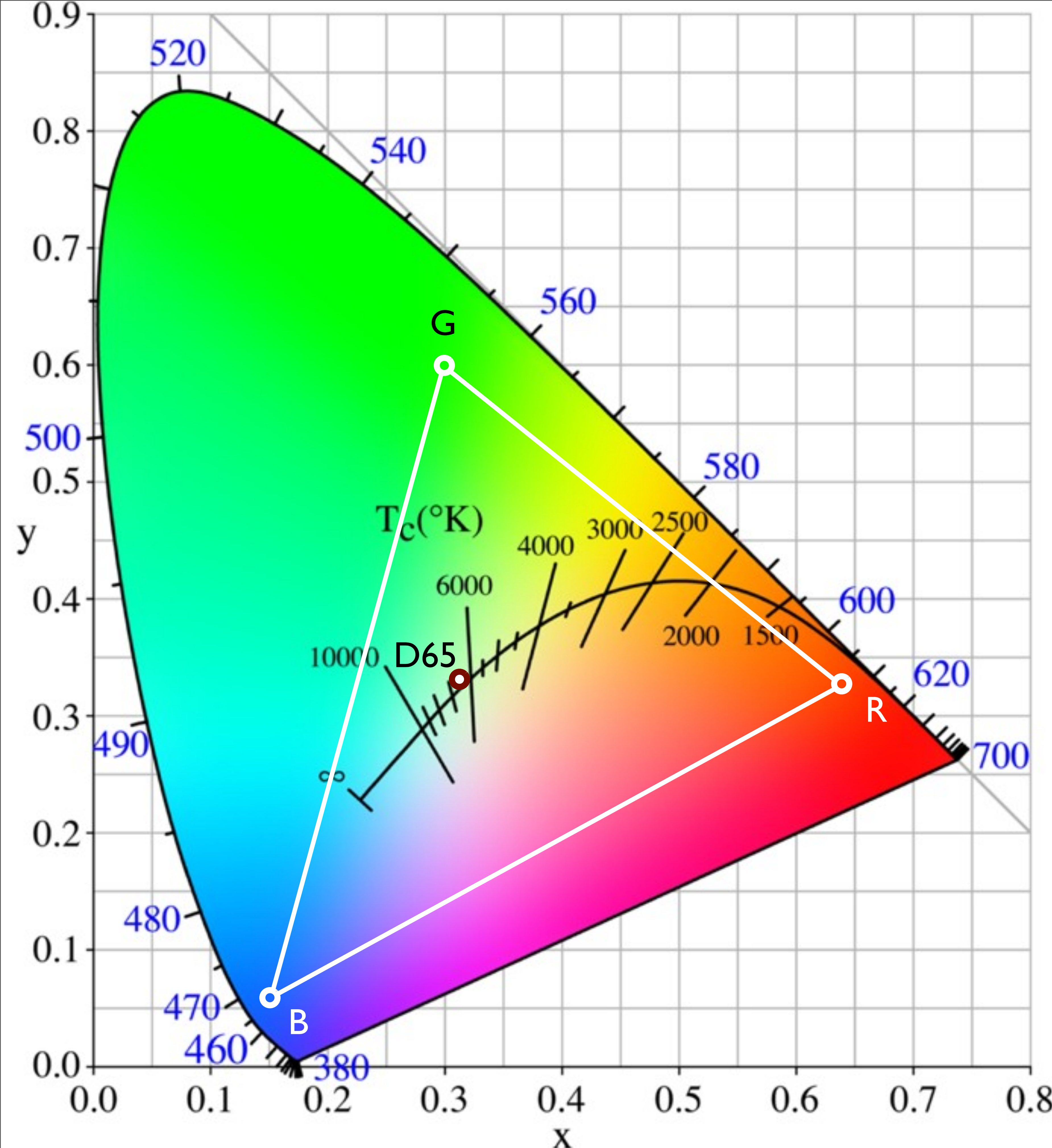
Chromaticity coordinates *x*, *y*, *z* defined by normalisation:

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z} = 1 - x - y$$

CIE 1931 Color matching functions





CIE 1931 chromaticity chart

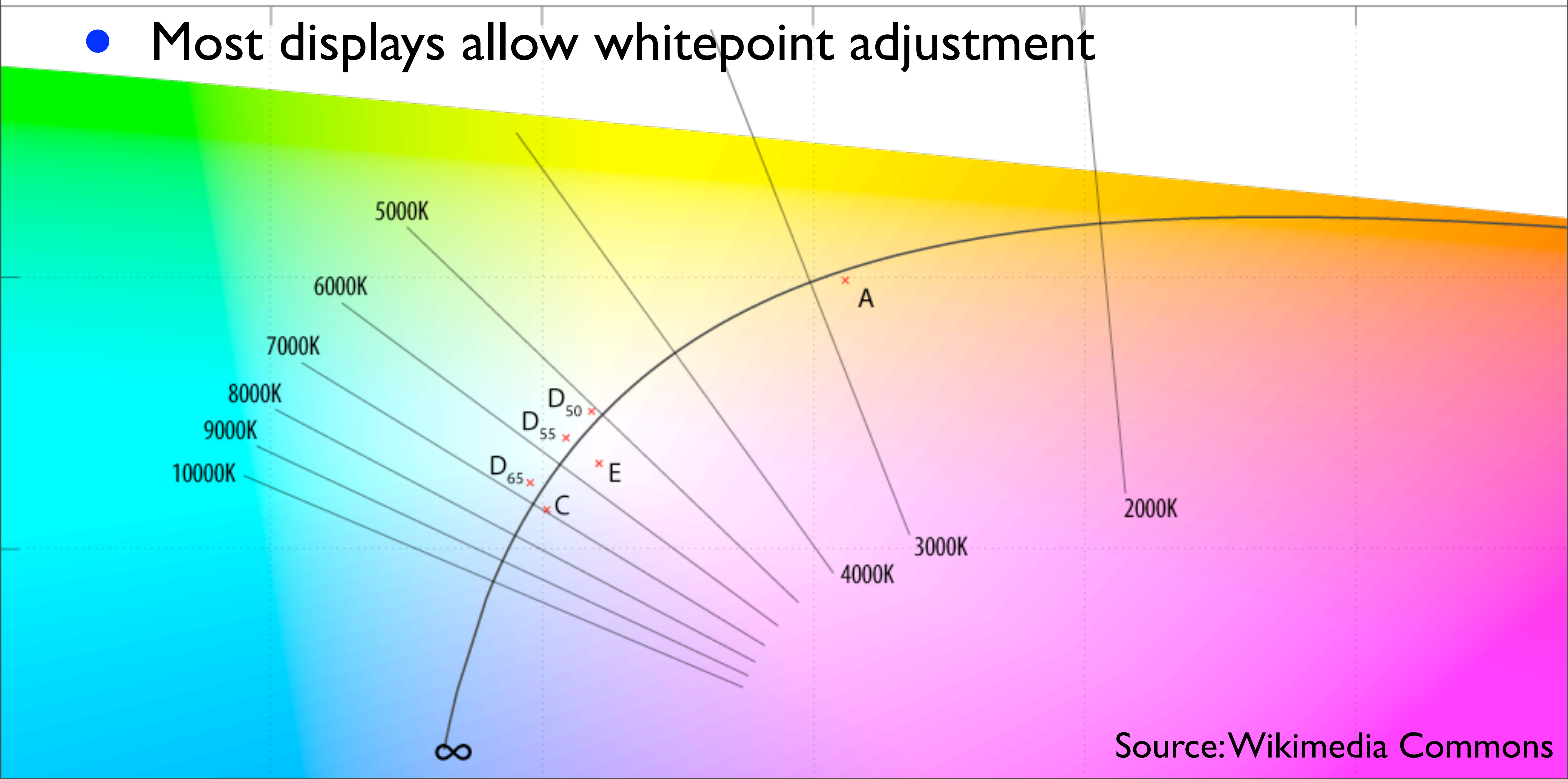
- spectral curve (border),
- Planckian locus (color temperature)
- s[tandard]RGB gamut (white triangle)
- Everything outside only approximation

You cannot display a spectrum properly in standard RGB!

Source: Wikimedia Commons

What is white?

- Naive: Neutral colour, i.e. without visible hue
- White points typically on or near Planckian curve between 5000 (afternoon Sun) and 6500 K (noon overcast sky)
- Standard RGB uses 6500 K
- Most displays allow whitepoint adjustment



Source: Wikimedia Commons

Standard RGB (sRGB)

Chromaticity	Red	Green	Blue	White
x	0.64	0.30	0.15	0.3127
y	0.33	0.60	0.06	0.3290
z	0.03	0.10	0.79	0.3583

- Whitepoint corresponds to standard illuminant D65 (6504 Kelvin, daylight standard, approx. overcast sky at noon)
- sRGB standard for many computer displays (CRT and LCD)
- Apple, of course, have their own (but similar) RGB space

XYZ to RGB

Given the X, Y, Z values we can calculate linear RGB:

$$\begin{pmatrix} R_{\text{linear}} \\ G_{\text{linear}} \\ B_{\text{linear}} \end{pmatrix} = \begin{pmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

After rescaling R,G,B to [0:1] apply the gamma correction to get the standard RGB values.
Let $C_{\text{srgb}} \in \{R_{\text{srgb}}, G_{\text{srgb}}, B_{\text{srgb}}\}$

$$C_{\text{srgb}} = \begin{cases} 12.92 C_{\text{linear}}, & C_{\text{linear}} \le 0.0031308 \\ (1+a) C_{\text{linear}}^{1/2.4} - a, & C_{\text{linear}} > 0.0031308 \end{cases}$$

Good colour schemes

Less is more

- Avoid saturated colours; vivid colours may "blind" the reader/audience (and they look wrong with most beamers anyway)
- Use simple colour pattern; just as many colours as required.

Colour scheme should emphasize your message

- If possible (and applicable) use colours that map well on grayscale (printing!)
- Colours should make intuitively sense (not trivial!)

Good colour schemes

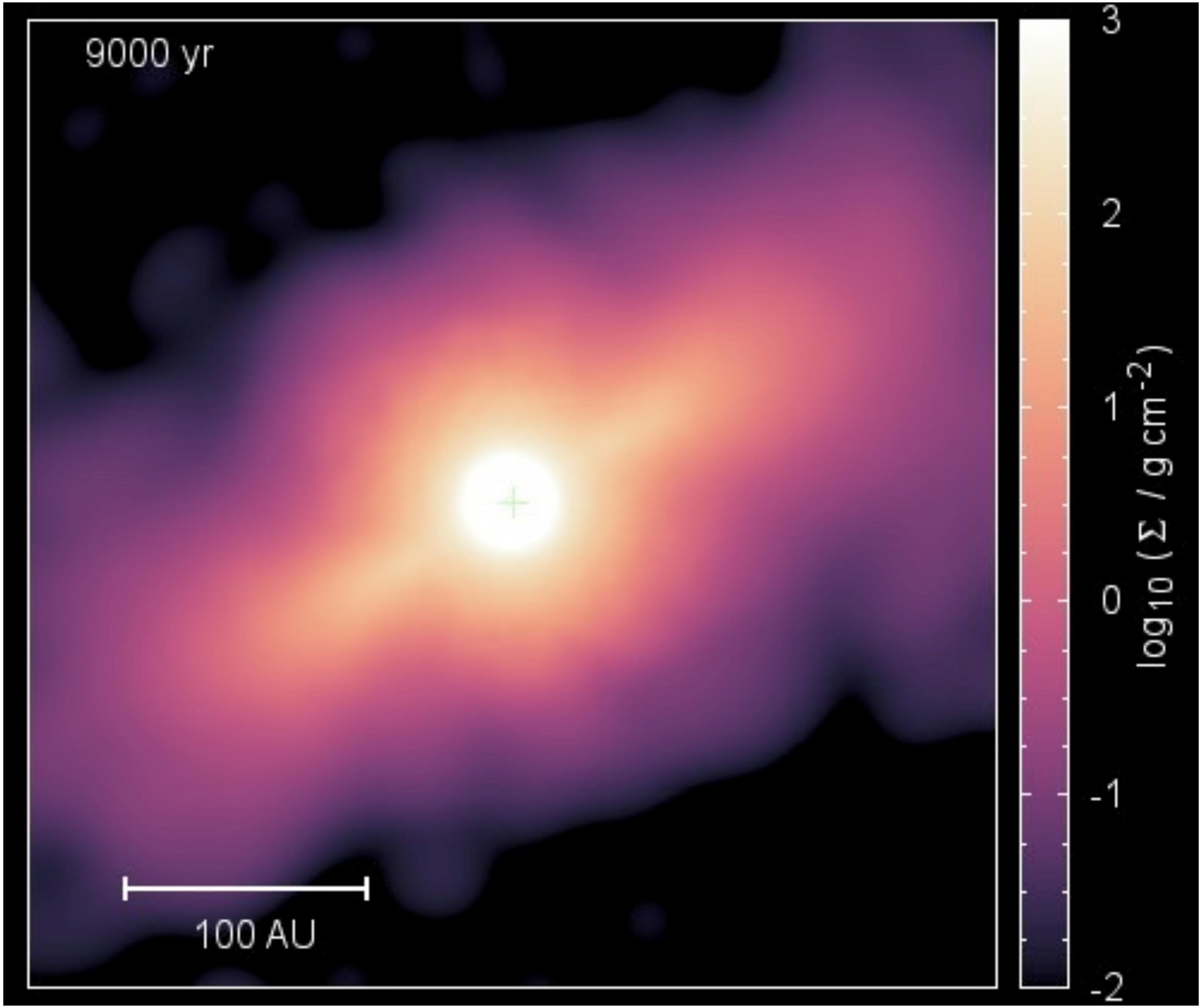
Luminance, hue and saturation

- Variation in luminance (“brightness”) for fine details
- Hue and saturation for large-scaled features
- Subtle changes in luminance and saturation often sufficient

Label your colours!

- Use the colorbox (default in gnuplot) feature unless your colour schemes are self-explaining (almost never) or for illustration only

Example of a "heat map"

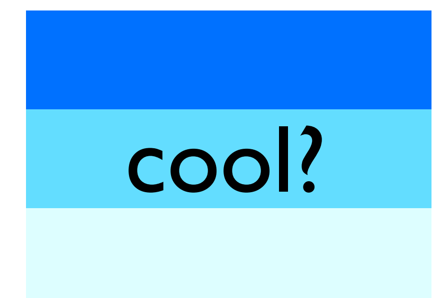


Thies & Kroupa (2011)

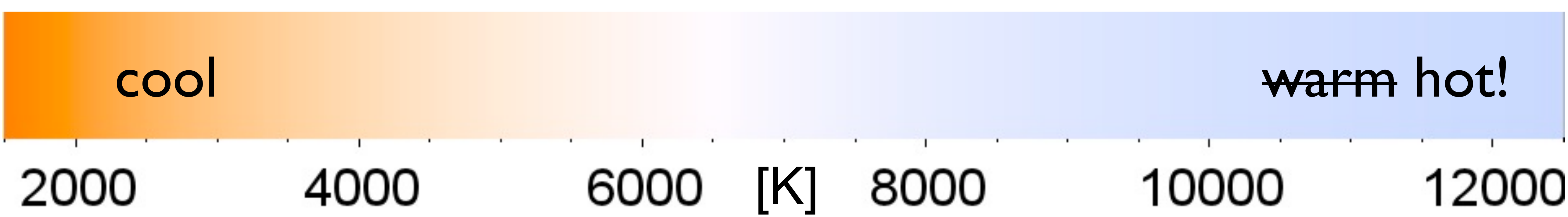
Good colour schemes

Colour-psychology: cool & warm colours

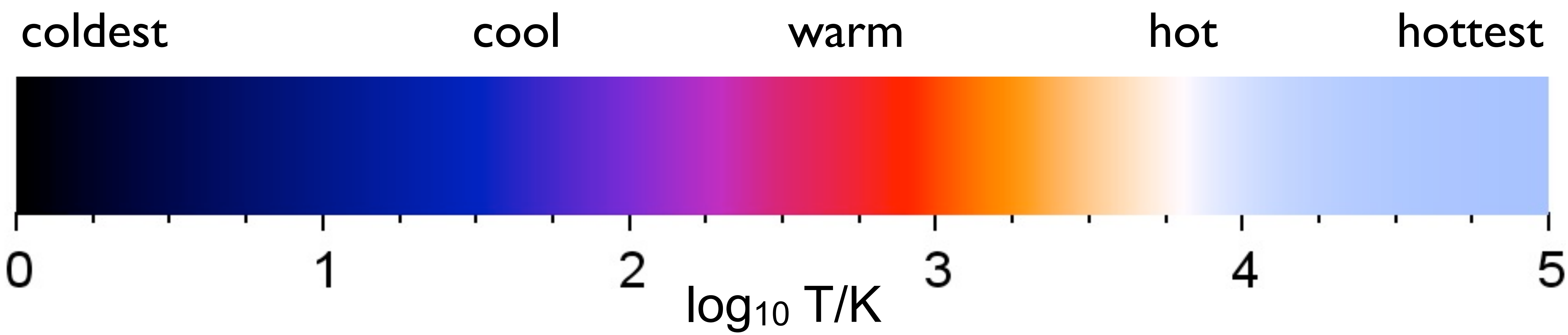
- In everyday's life, "warm" colours are yellowish to reddish hues (like fire)
- Blueish colours are generally considered "cool" (ice, the sea etc.)



But... physics is different!



Proposed temperature scheme



This is just an example for an overall color temperature scale

- Dark blue to purple for lowest temperatures (0 to a few hundred K, artist's scale)
- Red to yellow for intermediate temperatures (a few hundreds to a few thousands, Planck scale)
- white to blue for the highest temperatures (Planck scale)

gnuplot RGB palettes

gnuplot supports several methods to define the colour palette. Among them are:

- defined: explicit RGB values for (scalable) levels
- file: as "defined" but read from a file, e.g.

```
0       R0    G0    B0
1       R1    G1    B1
...
n       Rn    Gn    Bn
```

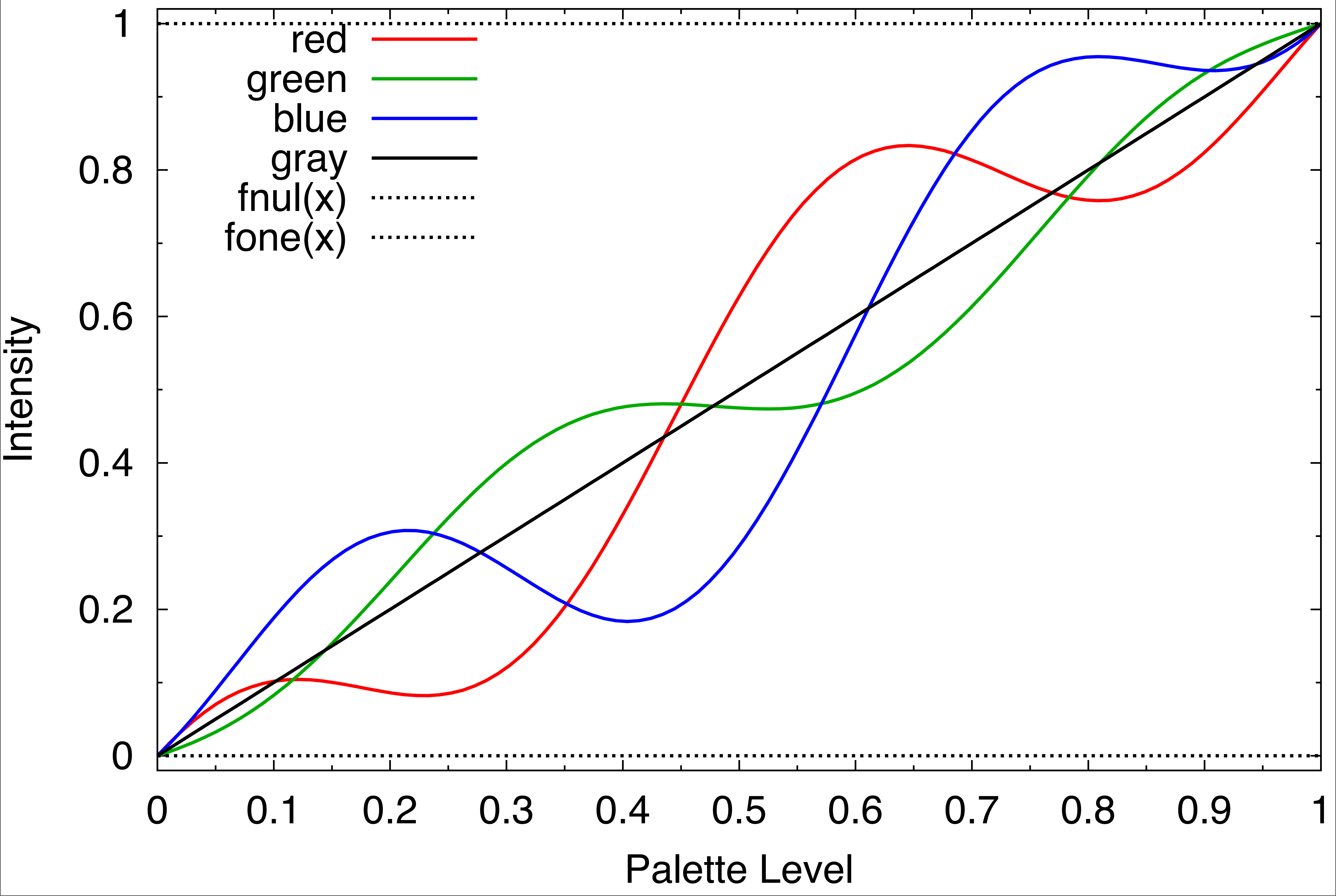
- function: Define R, G, B profiles as continuous functions
- cubehelix: Monotonic increasing luma with "orbiting" colour hue
- There are some more options available in gnuplot

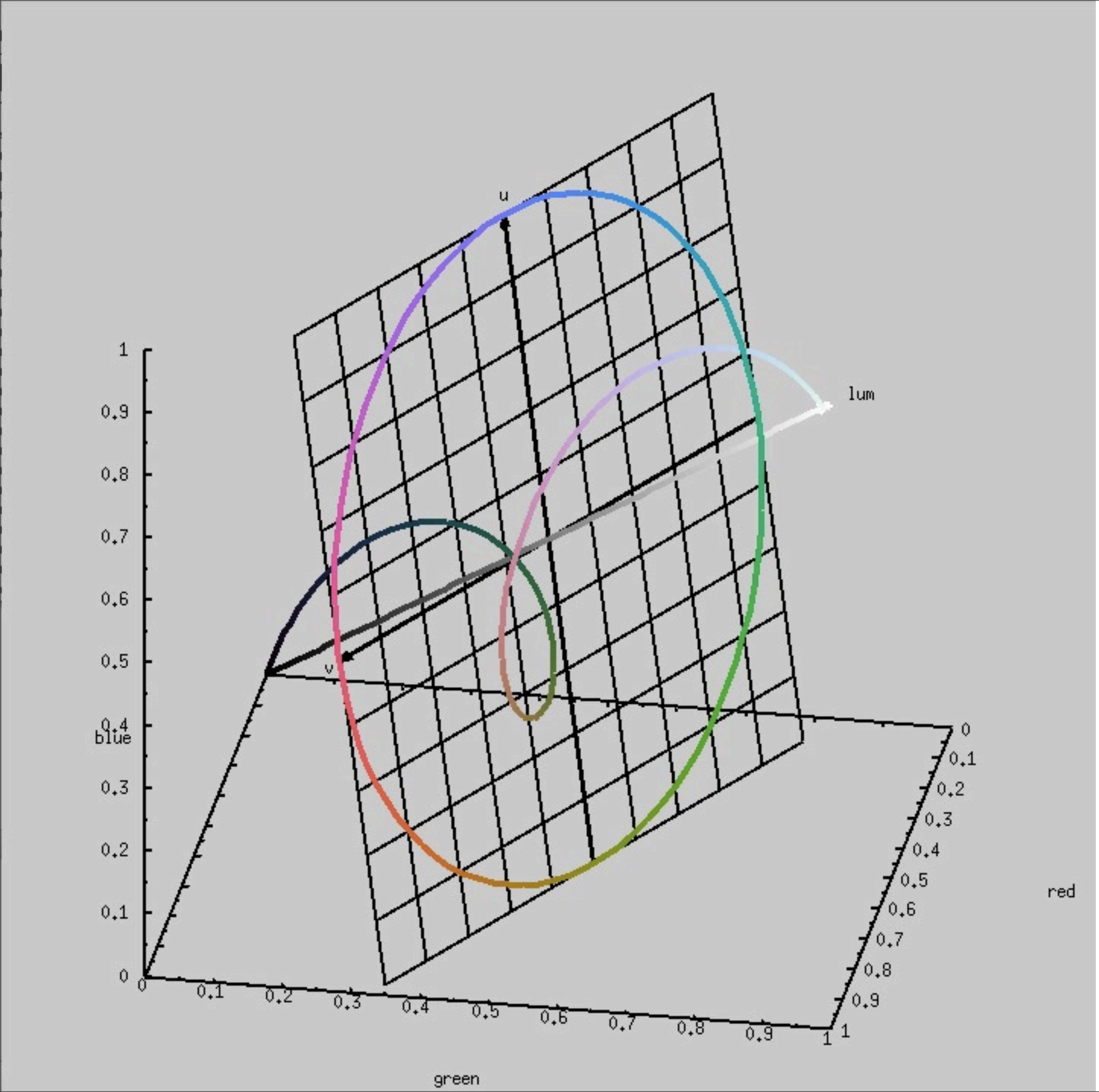
Cubehelix palettes

- Published by Dave Green (2011; arXiv:1108.5083)
- Later implemented in gnuplot 4.6
- Goal: Coloured palette with (almost) linearly rising luma (~printing brightness)
- Easy to define by start colour, spin direction and number of spins

```
set palette cubehelix start 0.5 cycles -1.5 saturation 1
set palette gamma 1.5
```

Cubehelix colour functions





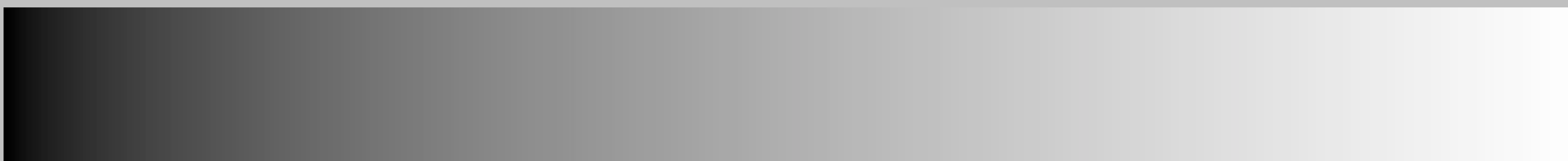
Cubehelix

- Luma linear rising
- RGB "spins" around grey axis
- tilt of color plane corresponding to R,G,B luma values

Examples

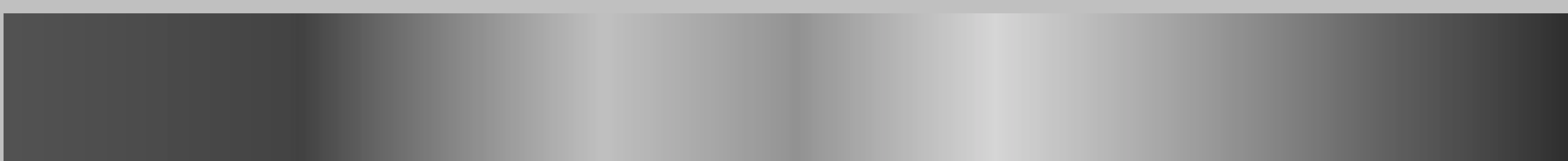
Cubehelix default

same, as greyscale



“standard” rainbow

same, as greyscale



Colorhelix program

- Written be I. Thies (2011, before cubehelix came to gnuplot)
- Supports original cubehelix and CIE 1976 (L*a*b*) version
- L*a*b* sometimes produces smoother palettes, but more difficult to avoid over-saturation
- available at .../~ithies/gnuplot/colortools/colorhelix.tar.bz2

XYZ to L*a*b* equations:

$$L^\star = 116 f(Y/Y_n) - 16$$
$$a^\star = 500\,[f(X/X_n) - f(Y/Y_n)]$$
$$b^\star = 200\,[f(Y/Y_n) - f(Z/Z_n)]$$

where X_n, Y_n and Z_n refer to the white point, and

$$f(t) = \begin{cases} t^{1/3} & \text{if } t > (\frac{6}{29})^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29} & \text{otherwise} \end{cases}$$

RGB vs. L*a*b*

Cubehelix default: startcolor 0.5, spins -1.5

Cubehelix default in L*a*b*

Cubehelix: start 2.0, spins -0.5, opt. saturation

same in L*a*b*: start -0.75, spins +0.5

More examples

Cubehelix: start 0.45, spins -0.4, opt. saturation

Cubehelix: start 0.45, spins 0.4, opt. saturation

Cubehelix: start 0.75, spins 1.0 (L*a*b*)

Cubehelix: start 0.75, spins -1.0 (L*a*b*)

Part II: Making movies with gnuplot

Two philosophies

Animated GIFs

- Readily available with gnuplot *(sic! lowercase)*
- But large files and only 256 colours

Postprocessing separate plots

- Flexible & true colour (e.g. MPEG from JPEG)
- But requires a bit more work...

gnuplot → image

Plotting an image with pixel (x,y) coordinates and explicit RGB values:

```
plot 'foo.dat' u 1:2:3:4:5 w rgbimage
```

where ‘foo.dat’ has the following structure

```
xmin   ymin     R  G  B # R,G,B from 0 to 255
...
xmax   ymin     R  G  B
       <blank line separating the blocks>
xmin   ymin+1   R  G  B
...
xmax   ymin+1   R  G  B
...
...
xmax   ymax     R  G  B
```

gnuplot → image

Plotting "heat map" with pixel (x,y) coordinates and scalar value v (mapped to colour through palette)

```
plot 'foo.dat' u 1:2:3 w image
```

where 'foo.dat' has the following structure

```
xmin   ymin      v(xmin,ymin)
...
xmax   ymin      v(xmax,ymin)
        <blank line separating the blocks>
xmin   ymin+1    v(xmin,ymin+1)
...
xmax   ymin+1    v(xmax,ymin)
...
...
xmax   ymax      v(xmax,ymax)
```

JPEG to MPEG/AVI

Sample bash code to convert numbered files sph*.jpg to MPEG4-coded movie file sph.avi.

```
mencoder "mf://sph*.jpg" -mf fps=25 -o sph.avi
-ovc lavc -lavcopts vcodec=mpeg4
```

(sorry for line break; command should be one line)

JPEG to MPEG/MP4

Sampe bash code to convert numbered files sph*.jpg to MPEG4-coded movie file sph.mp4.

```
ffmpeg -y -r 25 -sameq -i sph_%04d.jpg sph.mp4
```

fps indexed files output

AVI and MP4 are just different containers for the MPEG movie.

Part III: Plot panels with gnuplot

Basic syntax

```
set multiplot #don't use intrinsic layout here
set lmargin screen lmg
set rmargin screen rmg
set bmargin screen bmg
set tmargin screen tmg
```

Values *mg stand for decimal numbers between 0 (left/bottom margin) and 1 (opposite margin).

For full-canvas panels try lmg,bmg=0, rmg,tmg=1

Easy layout possible with mplayout script from

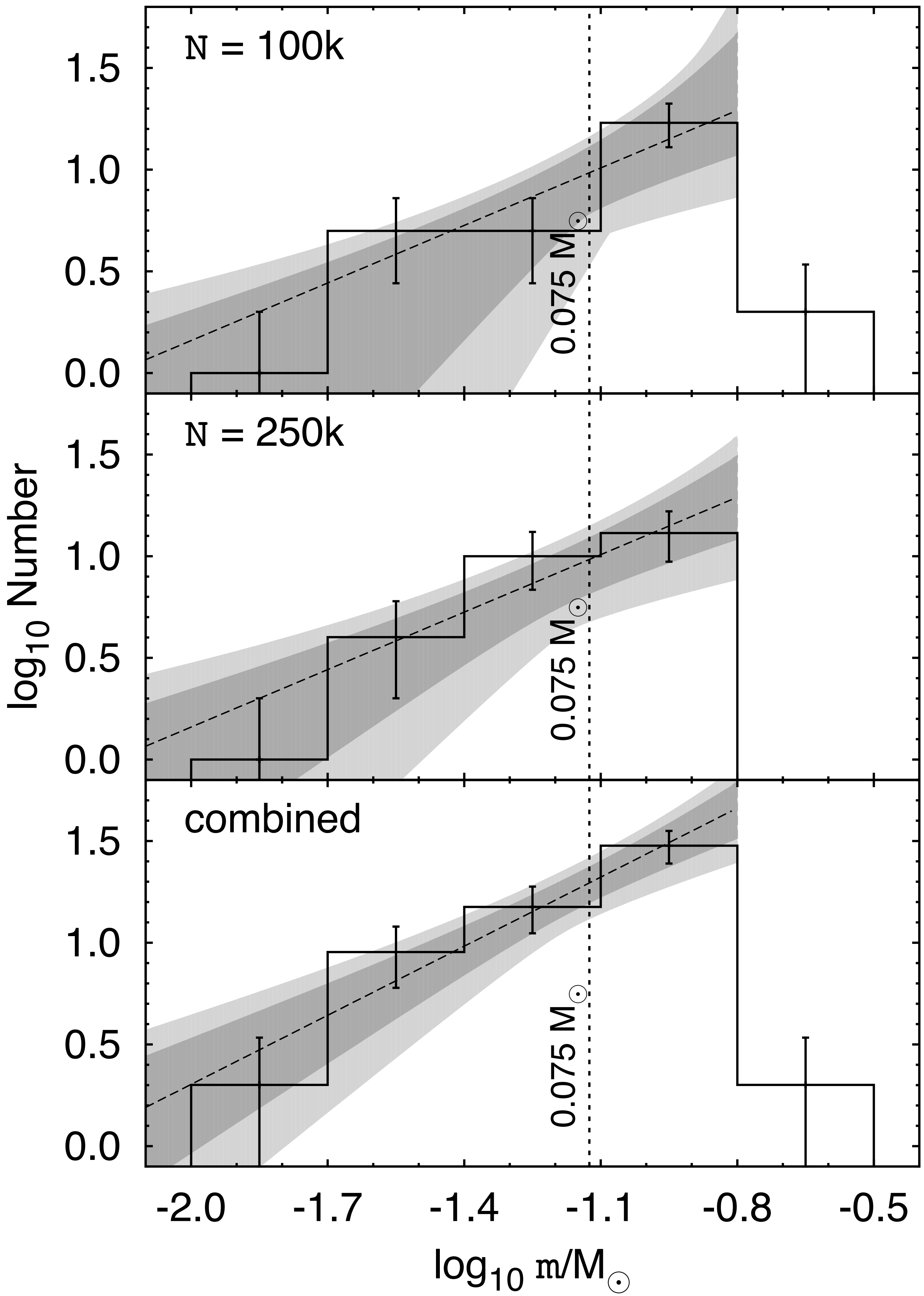
- .../~ithies/gnuplot/mplayout

MPlayout usage

General structure of calling script

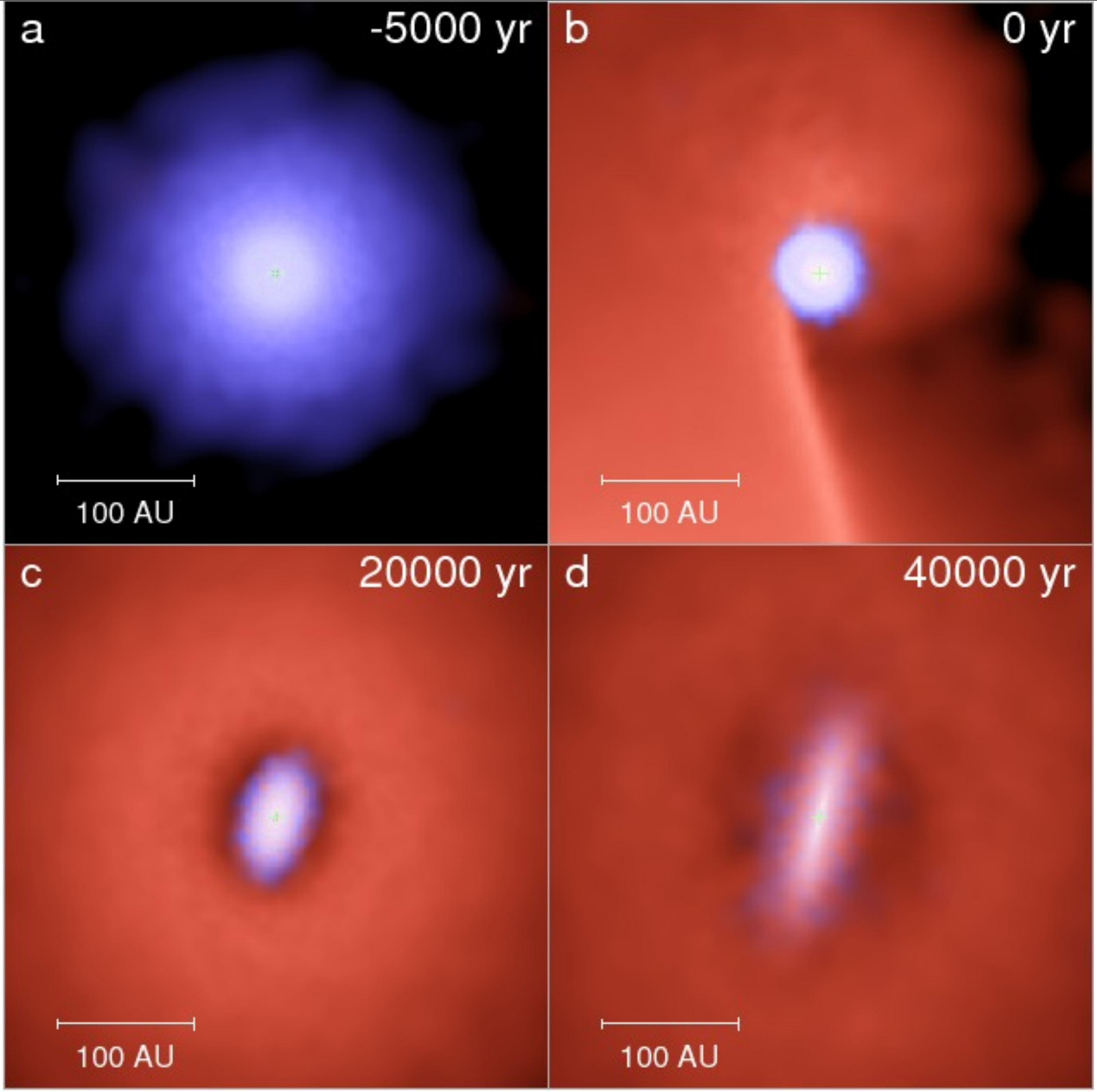
```
<define some mplayout variables here>
load ‘mplayout_base.gp’

<define terminal, labels, x/yranges etc.>

set multiplot

load 'mplayout_frame.gp'
plot <1st plot data> ...

load 'mplayout_frame.gp'
plot <2nd plot data> ...
...
```

MPlayout



- Easy multiplot layout with gnuplot
- Works with 4.4+ (probably also 4.2+)
- Common labels and tics are also possible
- You may also add margins to each frame

Thies & Kroupa (2011)



Thies & Kroupa (2011)

MPlayout

- Frames keep their shape (aspect ratio)
- Automatic enumeration (alphabetic & numeric)
- Enjoy!

References

- Gnuplot 4.6 Manual
- Philipp K. Janert, Gnuplot In Action, Manning 2010
- Wikipedia pages on CIE 1931, sRGB and others
- Dave Green (2011; arXiv:1108.5083)