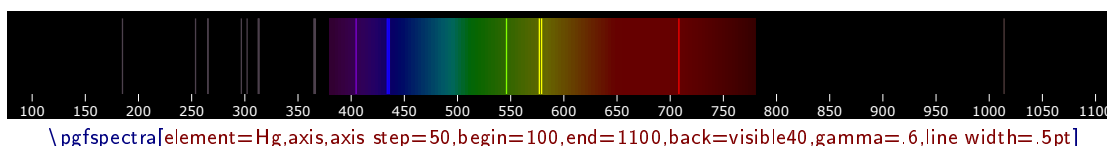


Manual for pgf-spectra 2.1.0

Hugo Gomes
hugo.parelho@gmail.com

16th May 2021



Abstract

The purpose of this package is to draw the spectrum of elements in a simple way. It's based on the package *pst-spectra*, but with some extra options. It relies on the pgf/TikZ to draw the desired spectrum, continuous or discrete. As in *pst-spectra* there are data available for the spectra of 98 elements and their ions. It also allows the user to draw a spectrum with their own personal data.

In version 2.0.0 the previous data of the visible region was extended to include lines from Extreme UV to Near IR ($10\text{ nm} \leq \lambda \leq 4000\text{ nm}$). See section *The lines data* below for more information.

Also in version 2.0.0 the possibility to redshift the lines of spectra was introduced, by entering directly the redshift value or the velocity and the angle to compute the redshift value (Doppler Redshift).

In this release – version 2.1.0 – a new color conversion (correlated color temperature), shadings for use with TikZ and/or PGFPLOTS and a color map for use with PGFPLOTS are available.

Contents

Installation and usage	1
What's new	2
In version 2.1.0	2
In version 2.0.0	3
The lines data	4
1 Data based on pst-spectra	4
2 Data based on NIST	4
The commands	5
\pgfspectra	5
\wcolor{wavelength}	5
\pgfspectraStyle[options]	6
\pgfspectraStyleReset	6
\tempercolor{Kelvin}	7
\pgfspectrashade[<h v>](start,end){name}	7
\pgfspectraplotshade[options]{name}	8
\pgfspectraplotmap[<I h>]{name}	9
\pgfspectrarainbow[<[tikz options]><(rainbow options)>]{radius}	10
The options for \pgfspectra	11
↪ use visible shading	11
↪ width	11
↪ height	11
↪ element	11
↪ charge	12
↪ Imin	12
↪ relative intensity	12
↪ relative intensity threshold	12
↪ line intensity	13
↪ gamma	14
↪ brightness	14
↪ back	15
↪ backIRUV	15
↪ IRcolor	15
↪ UVcolor	16
↪ lines	16
↪ line width	16
↪ begin	16
↪ end	17
↪ absorption	17
↪ axis	17
↪ axis step	18
↪ axis ticks	18
↪ axis unit	18
↪ axis unit precision	19
↪ axis color	19
↪ axis font	20
↪ axis font color	20
↪ label	20
↪ label position	20
↪ label font	20
↪ label font color	21

~ label before text	21
~ label after text	21
~ redshift	21
~ show redshift value	22
The options for <code>\pgfspectraplotshade</code>	23
~ shade end	23
~ shade opacity	23
~ shade opacity color	24
The options for <code>\pgfspectrarainbow</code>	24
~ rainbow start	25
~ rainbow knock out	25
~ rainbow fade	25
~ rainbow transparency	26
~ rainbow background	26
Examples	27
Using <code>\pgfspectra</code>	27
Using <code>\pgfspectrashade</code> in <code>TikZ</code>	30
Using <code>\pgfspectraplotshade</code> and <code>\pgfspectraplotmap</code> with <code>PGFPLOTS</code>	31
Using <code>\pgfspectrarainbow</code>	41
Alphabetical list of available options	44
Recommendations and known issues	46
The code	46

Installation and usage

pgf-spectra is placed under the terms of the L^AT_EX Project Public License, version 1.3 or later (<http://www.latex-project.org/lppl.txt>). pgf-spectra loads and only requires the package [TikZ](#).

You need to put the style file (pgf-spectra.sty) in a location where PDF^LA_TE_X can find them. According to the TDS conventions this may be a subdirectory named tex/latex/pgfspectra/ or tex/latex/misc/ in your (site specific) installation tree (insert your appropriate directory delimiter instead of /, if needed).

If you are using PDF^LA_TE_X, just can simply include the style file without any option via the `\usepackage` command, `\usepackage{pgf-spectra}`

It can also be loaded with *one option* to select the data source:
`\usepackage[option]{pgf-spectra}`

For more detailed information see section [The lines data](#).

What's new

► In version 2.1.0

- The continuous visible region is now drawn via TikZ shading, improving a little bit the speed of the whole process.
- Minor fix: the width of the emission/absorption lines are now correctly drawn.
- New keys for `\pgfspectra`:
 - use visible shading
 - backVIS
 - axis unit
 - axis unit precision
- New color conversion command, which converts a temperature in Kelvin to the correspondent rgb color, based on correlated color temperature:
 - `\tempercolor{temperature in Kelvin}`
- New commands that provides shadings to use in TikZ:
 - `\pgfspectrashade[<h|v>](start,end){name}`
 - `\pgfspectrainbow[<tikz options>]>[<rainbow options>]{radius}`

The TikZ keys that affect the rainbow are:

- * color
- * opacity
- * scope fading

The specific rainbow options are:

- * rainbow fade
- * rainbow start
- * rainbow knock out
- * rainbow background
- * rainbow transparency

- New command that provides a shading to use in PGFPLOTS:
 - `\pgfspectraplotshade[options]{name}` with the following specific keys
 - * shade end
 - * shade opacity
 - * shade opacity color
- New command that builds a color map to use in PGFPLOTS:
 - `\pgfspectraplotmap[<I|h>]{name}`

► In version 2.0.0

- The package can now be loaded with one of the following options:
 - `\usepackage[NIST]{pgf-spectra}` (**default**)
 - `\usepackage[LSE]{pgf-spectra}`
- Range of spectral window from 10 nm to 4000 nm (previous version from 380 nm to 780 nm).
- Added the lines data outside the visible range for the 98 elements.
- No more dependency on the package `ifthen` (code rewritten with the `\ifx` $\text{T}_{\text{E}}\text{X}$ primitive).
- Setting/disabling global options to draw the spectra's with the new commands:
 - `\pgfspectraStyle[options]`
 - `\pgfspectraStyleReset`
- New keys:
 - axis ticks
 - backIRUV (only for emission spectrum)
 - IRcolor (for emission lines and for background in absorption spectrum)
 - UVcolor (for emission lines and for background in absorption spectrum)
 - redshift
 - show redshift value
- The issues with the zooming of the pdf viewer sometimes introducing blank lines in the spectra have been fixed:



The rendition should now be working for every zoom (I hope!):

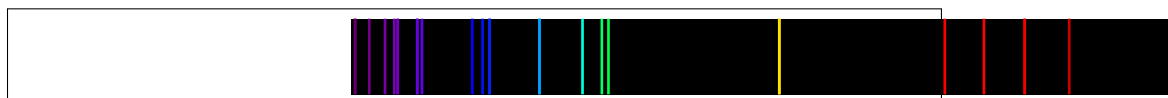


Many thanks to *Daniel García's* suggestion to solve this problem!

- Fixed the problem when putting the spectra inside any horizontal $\text{T}_{\text{E}}\text{X}$ box, like `\makebox`, `\mbox` or `\hbox`.

For instance, the code `\makebox[\textwidth][c]{\pgfspectra[element=He]}`:

- had as a result in the previous version (version 1.0):



- and will result in current version (2.0.0) at:



The lines data

There are two data sets available for drawing the spectra: one based in the previous version, whose data was initially obtained from the package `pst-spectra` and the other obtained from `NIST`.

In both cases are included the lines for 98 elements, from hydrogen ($Z = 1$) to einsteinium ($Z = 99$), except for francium ($Z = 87$). For each element there are lines between 10 nm and 4000 nm (obtained from the referred pages at February 2021).

1 Data based on `pst-spectra`

This set was obtained from <http://cdsarc.u-strasbg.fr/viz-bin/Cat?VI/16>

According to the information on the page the listed lines are based on "Line Spectra of the Elements", Joseph Reader and Charles H. Corliss CRC Handbook of Chemistry and Physics. This book refers that «The table contains the outstanding spectral lines of neutral (*I*) and singly ionized (*II*) atoms of the elements from hydrogen through plutonium ($Z = 1 - 94$); selected strong lines from doubly ionized (*III*), triply ionized (*IV*), and quadruply ionized (*V*) atoms are also included.»

Note: `pst-spectra` documentation refers "*Line Spectra of the Elements from the Astronomical Data Center of NASA*" as the source material, but I'm assuming the original source is "*Line Spectra of the Elements*", Joseph Reader and Charles H. Corliss CRC Handbook of Chemistry and Physics, obtained from <http://cdsarc.u-strasbg.fr/viz-bin/Cat?VI/16>.

To use this data set load the package `pgf-spectra` with the option `LSE` (acronym to Line Spectra of the Elements):

```
\usepackage[LSE]{pgf-spectra}
```

Number of lines provided: 46065 (see file `pgf-spectraDataLSE.pdf`)

2 Data based on `NIST`

This set was obtained from <https://physics.nist.gov/PhysRefData/Handbook/Tables/findinglist.htm>

According to the information on the page the listed lines «includes data for the neutral and singly-ionized atoms».

Note: **This set is loaded by default.** Because the data to search is slightly smaller (only neutral and singly-ionized atoms) the time consumption when building the spectra could be a bit lower.

To use this data set load the package `pgf-spectra` without options:

```
\usepackage{pgf-spectra}
```

Number of lines provided: 11980 (see file `pgf-spectraDataNIST.pdf`);

The commands

The four *main* commands, those related with this package itself, are:

- `\pgfspectra` or `\pgfspectra[options list]`
- `\wlcolor{wavelength}`
- `\pgfspectraStyle[options]`
- `\pgfspectraStyleReset`

There are other four commands to use with TikZ and/or PGFPLOTS:

- `\tempercolor{Kelvin}`
- `\pgfspectrashade[<h|v>](start,end){name}`
- `\pgfspectraplotshade[options]{name}`
- `\pgfspectraplotmap[<|h>]{name}`

And finally *just for fun* a command that draws a rainbow:

- `\pgfspectrarainbow[<[tikz options]> <(rainbow options)>]{radius}`

► Utilization of `\pgfspectra`

This command is used without options to draw the visible continuous spectrum:

```
\pgfspectra
```

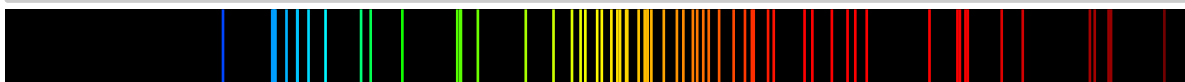


When using options, a continuous or discrete spectra in the visible region can be drawn, for instance:

```
\pgfspectra[width=.5\textwidth,height=1.25cm]
```



```
\pgfspectra[width=\textwidth,element=Ne]
```

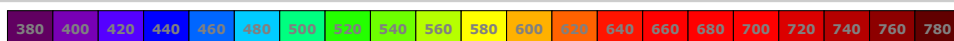


The options available for `\pgfspectra` are described in section [The options for \pgfspectra](#).

► Utilization of `\wlcolor{wavelength}`

A command to convert a wavelength from 380 to 780 nanometres (or other value in the range $10\text{ nm} \leq \lambda \leq 4000\text{ nm}$) to the respective color available as `'wlcolor'`:

```
\tikz{\foreach \x in {380,400,...,780}{\wlcolor{\x}
\draw[fill=wlcolor] (.03*\x,0) rectangle ++(.6,.5)
node[midway,font=\tiny\bfseries,text=black!50] {\x};}}
```




```
\tikz{\foreach \x/\y in {10/0,100/1,500/2,1000/3,2000/4,3000/5,4000/6}{\wlcolor{\x}
\draw[fill=wlcolor] (\y,0) rectangle ++(1,.5)
node[midway,font=\tiny\bfseries,text=black!50] {\x};}}
```



► Utilization of `\pgfspectraStyle[options]`

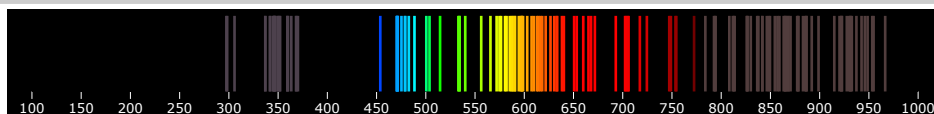
Use this command to set the global style of all the subsequent drawn spectra:

```
\pgfspectra[element=Ne] (before defining the global style)
```

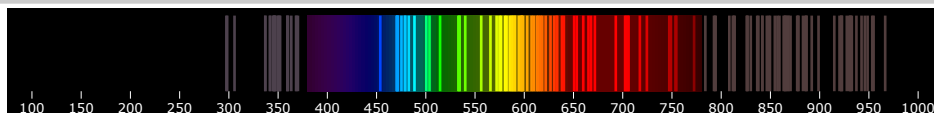


```
\pgfspectraStyle[width=.75\textwidth,axis,begin=100,end=1000,axis step=50]
```

```
\pgfspectra[element=Ne] (after defining the global style)
```

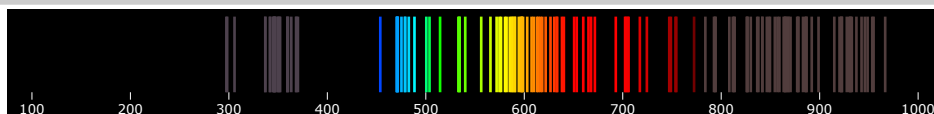


```
\pgfspectra[element=Ne,back=visible40,gamma=.6] (adding other options)
```



Note you can locally override the settings defined in the global style:

```
\pgfspectra[element=Ne,axis step=100] (overriding a global option)
```

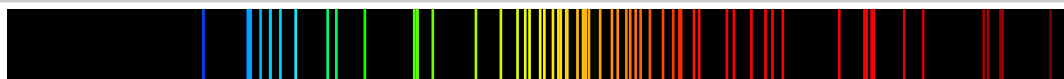


► Utilization of `\pgfspectraStyleReset`

Used to reset all the options of the spectra to their default values:

```
\pgfspectraStyleReset
```

```
\pgfspectra[element=Ne]
```

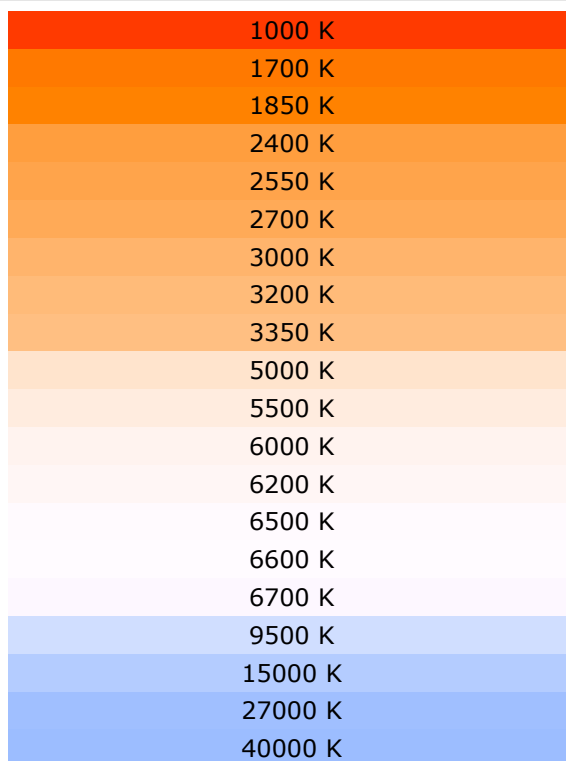


► Utilization of `\tempercolor{Kelvin}`

A command that uses the CIE 1964 10-degree color matching function to convert a given temperature, in Kelvin ($1000\text{ K} \leq T \leq 40000\text{ K}$), to the respective correlated color. For more information on the implemented algorithm, please see:

- <https://tannerhelland.com/2012/09/18/convert-temperature-rgb-algorithm-code.html>
- <https://www.zombieprototypes.com/?p=210>
- <https://github.com/neilbartlett/color-temperature>

```
\foreach \T in {1000,1700,1850,2400,2550,2700,3000,3200,%
  3350,5000,5500,6000,6200,6500,6600,6700,9500,15000,%
  27000,40000}
{
  \tempercolor{\T}\tikz{
    \fill[tempercolor,font=\small] (0,0) rectangle (7.5,.5) %
    node[midway] {\color{black}\T\ K};}\ [-1pt]%
  }%
```



► Utilization of `\pgfspectrashade[<h|v>](start,end){name}`

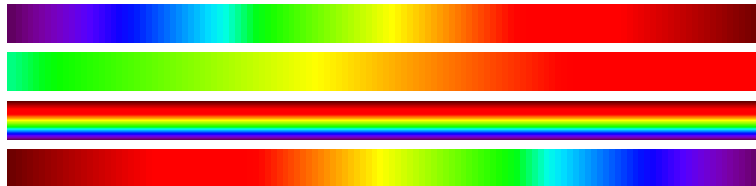
This command builds and makes available a **h**orizontal or a **v**ertical shading, between the '**start**' and '**end**' wavelengths (in nanometres), to use in TikZ pictures with the provided '**name**'.

Note that, in this command, the 'start' wavelength needs to be smaller than the 'end' wavelength and is in the visible region: $\lambda_{\text{start}} < \lambda_{\text{end}}$ and $380 \leq \lambda \leq 780$.

The optional parameter takes the value **h** or **v** and has the default value of **h**.

```
\pgfspectrashade(380,780){myShadeA}
\pgfspectrashade(500,700){myShadeB}
\pgfspectrashade[v](380,780){myShadeC}

\tikz{\fill[shading=myShadeA] (0,0) rectangle (10,.5);}
\\ [3pt]\tikz{\fill[shading=myShadeB] (0,0) rectangle (10,.5);}
\\ [3pt]\tikz{\fill[shading=myShadeC] (0,0) rectangle (10,.5);}
\\ [3pt]\tikz{\fill[shading=myShadeA,shading angle=180] (0,0) rectangle (10,.5);}
```

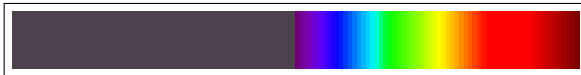


► Utilization of `\pgfspectraplotshade[options]{name}`

This command, without any options, builds and makes available a shading in the wavelength range from 0 nm to 780 nm to use in PGFPLOTS with the provided 'name'.

```
\pgfspectraplotshade{myPlotShadeA}

\fbbox{\tikz{\fill[shading=myPlotShadeA] (0,0) rectangle (7.5,.75);}}
```



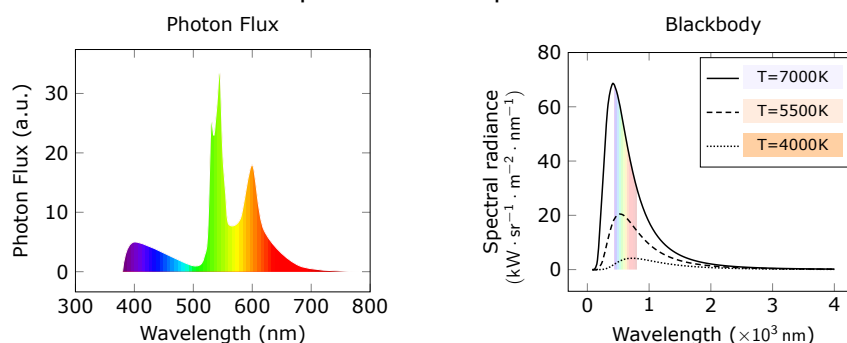
The optional argument can receive specific options for the shade or `\pgfspectra` options:

```
\pgfspectraplotshade[shade end=4000,IRcolor=white,UVcolor=white,
shade opacity=.2,gamma=.6]{myPlotShadeB}

\fbbox{\tikz{\fill[shading=myPlotShadeB] (0,0) rectangle (7.5,.75);}}
```



The specific options available are `shade end`, `shade opacity` and `shade opacity color`. See section [The options for \pgfspectraplotshade](#) for detailed information on using these options. When used in PGFPLOTS it's possible to do plots like:



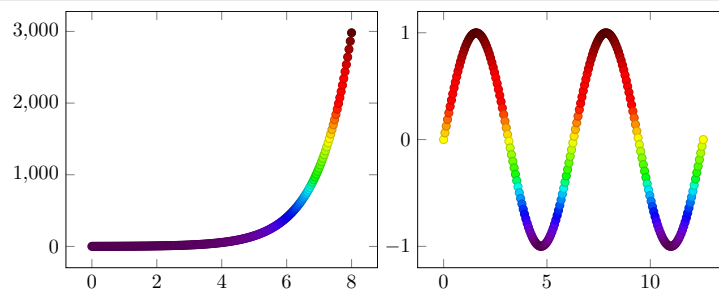
For these plots and other ones see [Using \pgfspectraplotshade and \pgfspectraplotmap with PGFPLOTS](#).

► Utilization of `\pgfspectraplotmap[<l|h>]{name}`

This command builds and makes available a **low** or **high resolution** color map in the wavelength range from 380 nm to 780 nm to use in PGFPLOTS with the provided 'name':

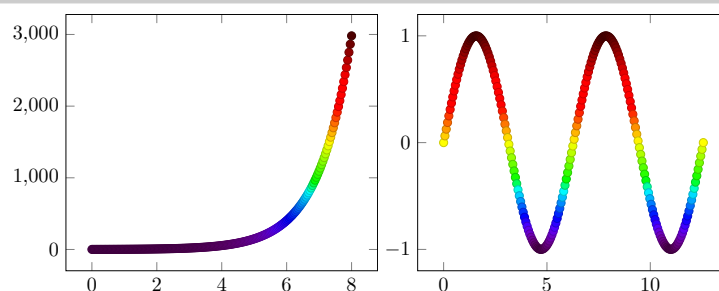
`\pgfspectraplotmap{myColorMap}`% low resolution (default value for optional parameter)

```
\begin{tikzpicture}
\begin{axis}[colormap name=myColorMap]
\addplot+[scatter,only marks,domain=0:8,samples=200] {exp(x)};
\end{axis}
\end{tikzpicture}
\begin{tikzpicture}
\begin{axis}[colormap name=myColorMap]
\addplot+[scatter,only marks,domain=0:4*pi,samples=200] {sin(deg(x))};
\end{axis}
\end{tikzpicture}
```



`\pgfspectraplotmap[h]{myColorMapH}`% high resolution ('h' value in optional parameter)

```
\begin{tikzpicture}
\begin{axis}[colormap name=myColorMapH]
\addplot+[scatter,only marks,domain=0:8,samples=200] {exp(x)};
\end{axis}
\end{tikzpicture}
\begin{tikzpicture}
\begin{axis}[colormap name=myColorMapH]
\addplot+[scatter,only marks,domain=0:4*pi,samples=200] {sin(deg(x))};
\end{axis}
\end{tikzpicture}
```



Actually using high or low resolution produces the same effect on plot. The difference resides on the number of colors available to the 'color of colormap' feature. For more information see [Using \pgfspectraplotshade](#) and [\pgfspectraplotmap with PGFPLOTS](#).

The above commands – `\pgfspectrashade`, `\pgfspectraplotshade` and `\pgfspectraplotmap` – were inspired in the [TeX - LaTeX Stack Exchange](#) questions, [Filling optical spectrum curve with color gradient](#) and [How to create a electromagnetic spectrum using pgfplots package \(together with colormaps\)](#), which were referred by Stefan Pinnow, as examples, in a features request for the `pgf-spectra` package.

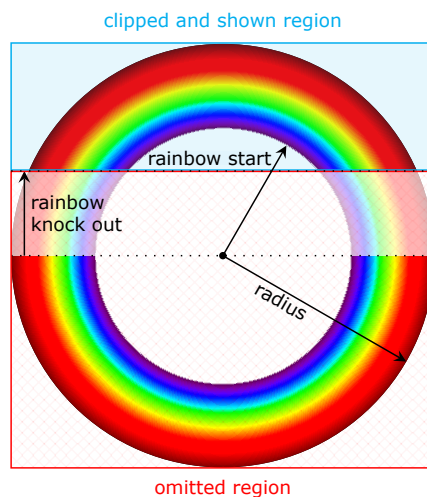
► **Utilization of `\pgfspectrarainbow<[tikz options]><(rainbow options)>\{radius\}`**

Without options this command draws a rainbow with the specified radius:

```
\pgfspectrarainbow{2cm}
```



The rainbow is designed with the following schema resulting in a clipped and shown region:



The options available could be specific options for the rainbow or *common* TikZ options:

- the rainbow specific options:
 - rainbow start
 - rainbow knock out
 - rainbow fade
 - rainbow transparency
 - rainbow background
- the TikZ options: any option known by TikZ and/or TikZ libraries.

For detailed information on using this command see [The options for `\pgfspectrarainbow`](#).

The options for `\pgfspectra`

For the commands `\pgfspectra` and `\pgfspectraStyle` there are a set of options available to draw the spectrum as described below.

The list of options is of the form `'key'` or `'key=value'` separated by commas.

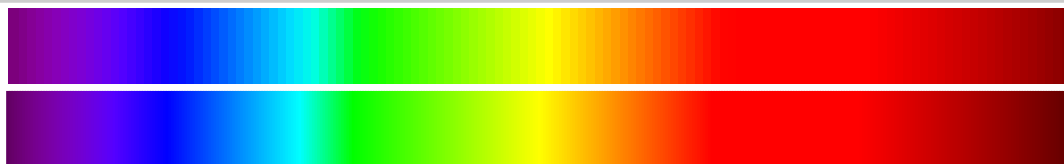
use visible shading

default: *true*

The visible region of the spectra is drawn using a *TikZ* shading instead of line by line, resulting in a faster drawing of that region. When set to `'false'` the visible region is drawn line by line: this value could be useful for printers that tend to be problematic when printing the shadings.

(new in v2.1.0)

```
\pgfspectra
\\ \pgfspectra[use visible shading=false]
```



width

default: *0.9\textwidth*

Sets the width of the spectrum.

```
\pgfspectra[width=10cm]
```



height

default: *1cm*

Sets the height of the spectrum.

```
\pgfspectra[height=40pt]
```



element

default: *NONE*

A single chemical symbol of an element or a list of chemical symbols.

```
\pgfspectra[element=H]
```



```
\pgfspectra[element={H,He}]
```



charge

default: 0

The charge of the *particle* to draw the spectrum. Use 'all' to get all available lines for the element, i.e, for the atom and all the positive ions that exist in the database.

```
\pgfspectra[element=He]
```



```
\pgfspectra[element=He,charge=1]
```



```
\pgfspectra[element=He,charge=2]
```

Element "He" with charge "2" have no lines to display.

```
\pgfspectra[element=He,charge=all]
```

**Imin**

default: 0

The minimum intensity of the lines to put in the spectrum. Value from 0 to 1.

```
\pgfspectra[element=He,Imin=.5]
```



```
\pgfspectra[element=He,Imin=.05]
```

**relative intensity**

default: false

Draws the lines respecting the intensity of the observed spectrum.

```
\pgfspectra[element=He,relative intensity]
```

**relative intensity threshold**

default: 0.25

Sets the minimum intensity for the lines in the spectrum when using relative intensities. When set to 0.25 a line with real intensity 0 will have a spectral intensity of 0.25 and a line with intensity equal to the max intensity observed in that spectrum will have an intensity in the computed spectrum of 1, assuming of course, an overall intensity in the range between 0 and 1.

```
\pgfspectra[element=He,relative intensity,relative intensity threshold=0]
```



```
\pgfspectra[element=He,relative intensity,relative intensity threshold=.25]
```



```
\pgfspectra[element=He,relative intensity,relative intensity threshold=.5]
```



```
\pgfspectra[element=He,relative intensity,relative intensity threshold=.75]
```



```
\pgfspectra[element=He,relative intensity,relative intensity threshold=1]
```



In fact, setting the relative intensity threshold to 1 is equivalent to the spectrum without relative intensities:

```
\pgfspectra[element=He]
```



line intensity

default: 100

Draws all the lines with the specified intensity between 0 and 100 (as a percentage of the maximum intensity).

```
\pgfspectra[element=He,line intensity=0]
```



```
\pgfspectra[element=He,line intensity=50]
```



```
\pgfspectra[element=He,line intensity=100]
```



```
\pgfspectra[element=He]
```



gamma

default: 0.8

Gamma color correction: any positive value.

\pgfspectra[gamma=.1]



\pgfspectra[gamma=.8]



\pgfspectra[gamma=1]



\pgfspectra[gamma=2]



\pgfspectra[gamma=5]



\pgfspectra[gamma=10]

**brightness**

default: 1

Brightness color correction as in the CMYK color model. Value between 0 and 1. Zero stands for black and one for the maximum bright. *This option only works for the continuous component of the spectra, to change the "brightness" of spectral lines use the option 'line intensity'.*

\pgfspectra[brightness=.1]



\pgfspectra[brightness=.5]



\pgfspectra[brightness=1]



backdefault: *black*

Sets the background color of the spectrum. Only useful when there are spectral lines. Some shorthand are defined to put the visible region in the background: `'visible5'`, `'visible10'`, `'visible15'`, ... , `'visible100'`.

Note: this labels combined with the `'brightness'` option makes it possible to achieve other values on the background, since the visible amount (5%,10%,...) is multiplied by the value of brightness.

```
\pgfspectra[element=He,back=white]
```



```
\pgfspectra[element=He,back=black!50]
```



```
\pgfspectra[element=He,back=visible50]
```



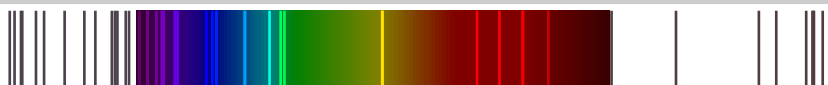
```
\pgfspectra[element=He,back=visible50,brightness=.26]
```

**backIRUV**default: *black*

Sets the background color, *only for the emission spectrum*, outside the visible region ($10nm \leq \lambda < 380nm$ and $780nm < \lambda \leq 4000nm$)

(new in v2.0.0)

```
\pgfspectra[element=He,back=visible50,begin=100,end=1000,backIRUV=white]
```

**IRcolor**default: *rgb(0.3157,0.2373,0.2373)*

Sets the color for emission lines and for background in absorption spectrum in the IR region ($780nm < \lambda \leq 4000nm$)

(new in v2.0.0)

```
\pgfspectra[element=He,back=visible50,begin=100,end=1000,IRcolor=white]
```



```
\pgfspectra[element=He,back=visible50,begin=100,end=1000,IRcolor=white,absorption]
```



UVcolordefault: *rgb(0.3,0.2568,0.3)*Sets the color for emission lines and for background in absorption spectrum in the UV region ($10\text{nm} \leq \lambda < 380\text{nm}$)*(new in v2.0.0)*

```
\pgfspectra[element=He,back=visible50,begin=100,end=1000,UVcolor=white]
```



```
\pgfspectra[element=He,back=visible50,begin=100,end=1000,UVcolor=white,absorption]
```

**lines**default: *{}*A comma separated list of wavelengths in the interval $[10; 4000]\text{nm}$.*(Interval updated in v2.0.0)*

```
\pgfspectra[lines={400,500,550,700}]
```



```
\pgfspectra[lines={200,205,400,500,550,700,850,950,980},begin=100,end=1000]
```

**line width**default: *1pt*

The width of each individual line in the spectrum.

```
\pgfspectra[line width=2pt]
```



```
\pgfspectra[line width=2pt,element=He]
```

**begin**default: *380*The starting wavelength in nanometres of the spectrum ($10 \leq \lambda \leq 4000$).*(Interval updated in v2.0.0)*

```
\pgfspectra[begin=500]
```



end

default: 740

The finishing wavelength in nanometres of the spectrum ($10 \leq \lambda \leq 4000$).*(Interval updated in v2.0.0)*`\pgfspectra[end=500]`

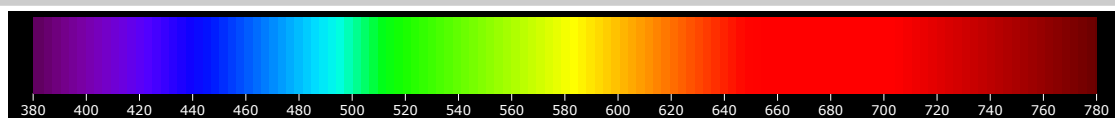
Remark: it's obviously possible to set 'begin' and 'end' at the same time and if desired change the order of the wavelengths.

`\pgfspectra[begin=500,end=700]``\pgfspectra[begin=700,end=500]``\pgfspectra[begin=780,end=380]``\pgfspectra[begin=780,end=380,element=He]`**absorption**default: *false*

Draws the absorption spectrum instead of the emission one.

`\pgfspectra[element=H,absorption]``\pgfspectra[element={H,He},absorption]`**axis**default: *false*

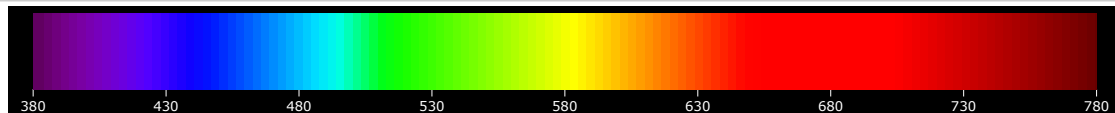
By default draws a nanometric axis below the spectrum.
In v2.1.0 is now possible to *change the unit* of the axis.

`\pgfspectra[axis]`

axis step

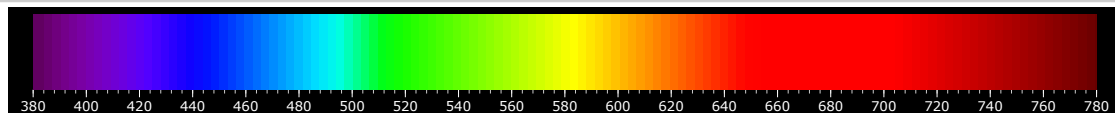
default: 20

The increment (in nanometres) to use in the axis scale.

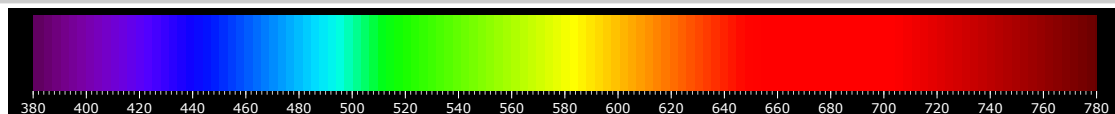
`\pgfspectra[axis,axis step=50]`**axis ticks**

default: 0

The number of minor ticks between two consecutive ticks in the axis.

(new in v2.0.0)`\pgfspectra[axis,axis ticks=4]`

Keep in mind, if you desire to divide two consecutive ticks into 10 equal parts set 'axis ticks=9':

`\pgfspectra[axis,axis ticks=9]`**axis unit**

default: nm

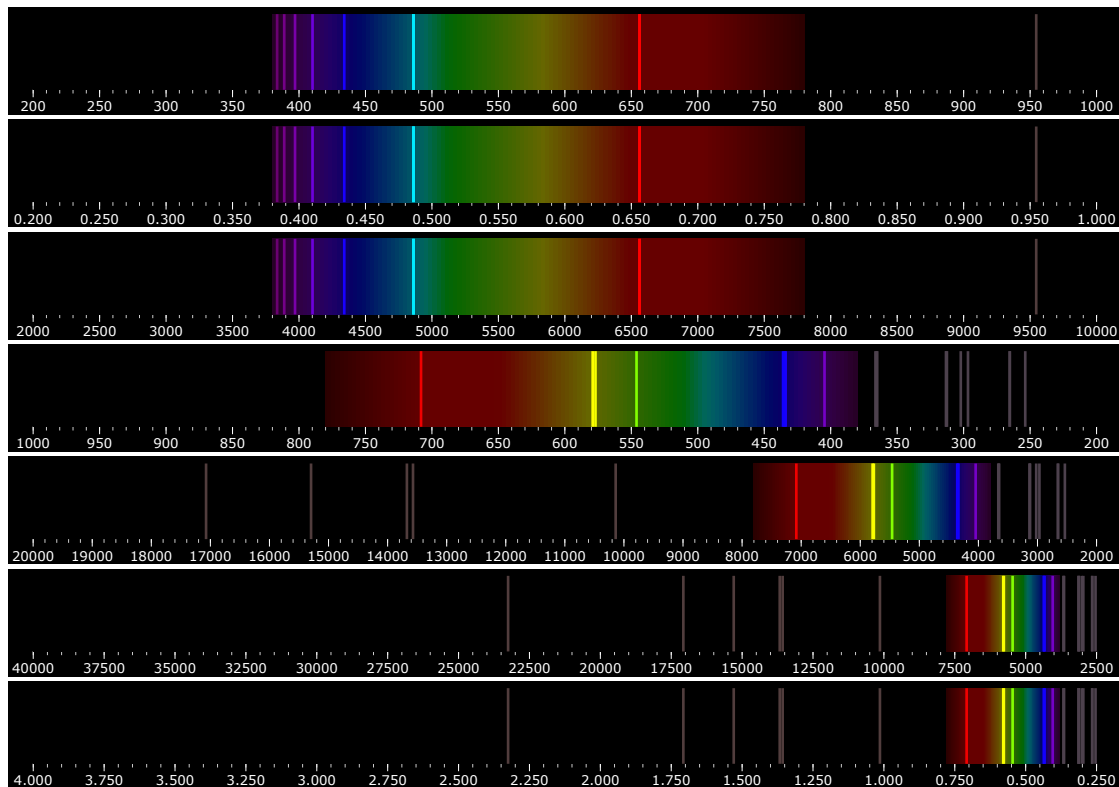
Sets the unit to use in the displayed values of wavelenghts in the axis below the spectrum.

Available units are:

- nanometre (nm): `axis unit=nm`
- micrometre (μm): `axis unit=micron`
- angstrom (\AA): `axis unit=A`

(new in v2.1.0)

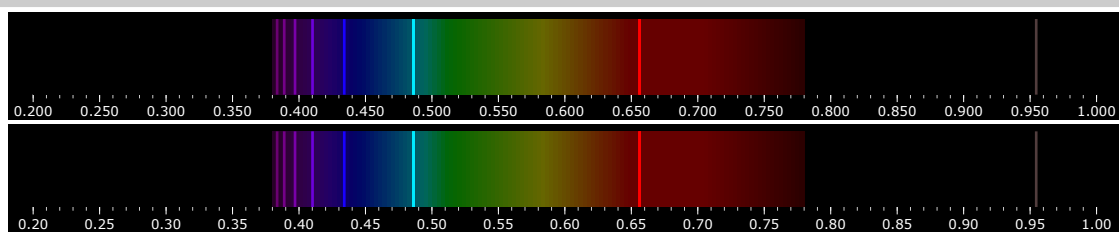
```
\pgfspectra[element=H,begin=200,end=1000,axis,axis step=50,axis
ticks=4,back=visible40]
\\ \pgfspectra[element=H,begin=200,end=1000,axis,axis step=50,axis
ticks=4,axis unit=micron,back=visible40]
\\ \pgfspectra[element=H,begin=200,end=1000,axis,axis step=50,axis
ticks=4,axis unit=A,back=visible40]
\\ \pgfspectra[element=Hg,begin=1000,end=200,axis,axis step=50,axis
ticks=4,back=visible40]
\\ \pgfspectra[element=Hg,begin=2000,end=200,axis,axis step=100,axis
ticks=4,axis unit=A,back=visible40]
\\ \pgfspectra[element=Hg,begin=4000,end=250,axis,axis step=250,axis
ticks=4,axis unit=A,back=visible40]
\\ \pgfspectra[element=Hg,begin=4000,end=250,axis,axis step=250,axis
ticks=4,axis unit=micron,back=visible40]
```

**axis unit precision**

default: 3

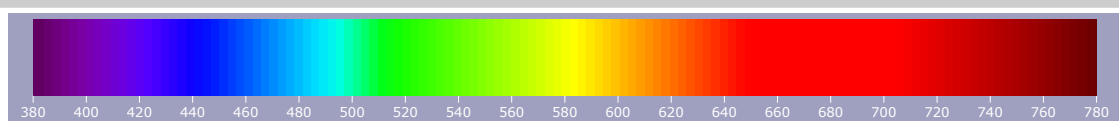
Sets the precision of the displayed wavelengths in the axis below the spectrum. *(new in v2.1.0)*

```
\pgfspectra[element=H,begin=200,end=1000,axis,axis step=50,axis ticks=4,axis
unit=micron,back=visible40]
\\ \pgfspectra[element=H,begin=200,end=1000,axis,axis step=50,axis
ticks=4,axis unit=micron,axis unit precision=2,back=visible40]
```

**axis color**default: *black*

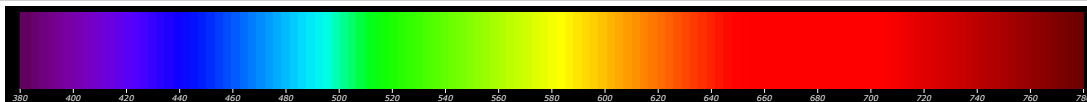
The color of the axis.

```
\pgfspectra[axis,axis color=red!50!green!50!blue!50]
```

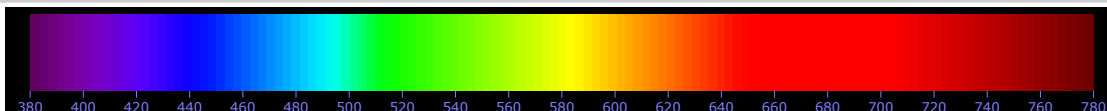


axis fontdefault: `\tiny`

The font specs to use in the axis.

`\pgfspectra[axis,axis font=\fontsize{3}{3}\itshape\selectfont]`**axis font color**default: `white`

The color of the font used in the axis.

`\pgfspectra[axis,axis font color=blue!50!white]`**label**default: `false`

Puts a label for the spectrum.

`\pgfspectra[label]``\pgfspectra[label,element=He]`**label position**default: `west`

Sets the position of the label according to:

north west	north	north east
west	<i>spectrum</i>	east
south west	south	south east

`\pgfspectra[label,label position=east,element=He]`**label font**default: `\bfseries\small`

The font specs for the label.

`\pgfspectra[label,label font=\footnotesize\itshape,element=He]`

label font colordefault: *black*

The color of the font used in the label.

`\pgfspectra[label,label font color=blue!50!white,element=He]`

He

**label before text**default: `{}`

Inserts text before the value stored in the label: if chemical symbols were provided, the label has them stored, otherwise it is empty.

`\pgfspectra[label,label before text=text\ ,element=He]`

text He

**Remark:** The `_` is to insert a space between the text entered by user and the text stored in label.**label after text**default: `{}`

Inserts text after the value stored in the label: if chemical symbols were provided, the label has them stored, otherwise it is empty.

`\pgfspectra[label,label after text=\ text,element=He]`

He text

**redshift**default: `{}`

Redshift (or blueshift) the spectral lines:

The redshift value (z) is *defined* as $1 + z = \lambda_{obs} / \lambda_E$ which leaves the observed wavelength to $\lambda_{obs} = (1 + z)\lambda_E$, given the emitted wavelength of the source (λ_E).

- Use `'redshift=<numeric value>'` to directly enter the redshift value
- or use `'redshift={D=<numeric value 1>/<numeric value 2>}'` to compute the Relativistic Doppler redshift with $\bar{v} = \text{<numeric value 1>}$ and $\theta = \text{<numeric value 2>}$.

The Relativistic Doppler redshift ($1 + z$) is calculated accordingly:

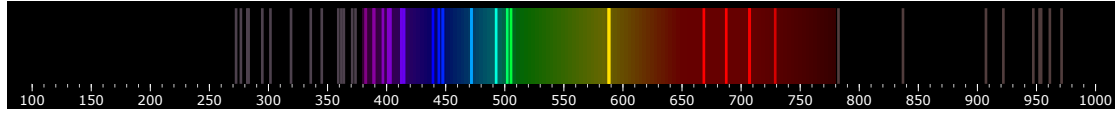
$$1 + z = \frac{1 + \bar{v} \cos \theta}{\sqrt{1 - \bar{v}^2}} \quad \bar{v} = \frac{v}{c}$$

where \bar{v} is the *normalized velocity* (in units of the speed of light in vacuum, c) of the emitter and θ is the angle between the direction of relative motion and the direction of emission in the observer's frame (zero angle is directly away from the observer). So, if the source of light is moving away from an observer, then redshift occurs ($z > 0$), but, if the source moves towards the observer, then blueshift occurs ($z < 0$).

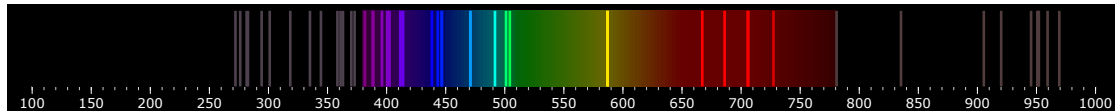


(new in v2.0.0)

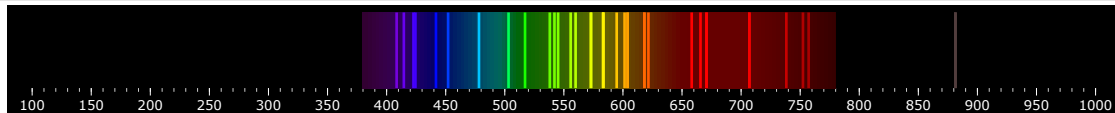

```
\pgfspectra[element=He,back=visible40,gamma=.6,axis,axis step=50,axis
ticks=4,begin=100,end=1000,redshift={D=.001/0}]
```



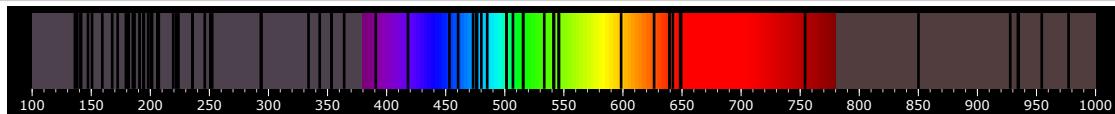
```
\pgfspectra[element=He,back=visible40,gamma=.6,axis,axis step=50,axis
ticks=4,begin=100,end=1000,redshift={D=.001/180}]
```



```
\pgfspectra[element=He,back=visible40,gamma=.6,axis,axis step=50,axis
ticks=4,begin=100,end=1000,redshift=.5]
```



```
\pgfspectra[element=He,absorption,gamma=.6,axis,axis step=50,axis
ticks=4,begin=100,end=1000,redshift=-.5]
```



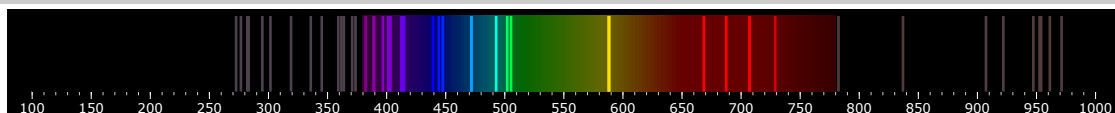
show redshift value

default: *false*

Writes the value of the redshift (left below the spectrum).

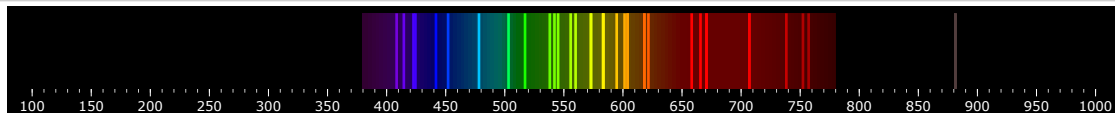
(*new in v2.0.0*)

```
\pgfspectra[element=He,back=visible40,gamma=.6,axis,axis step=50,axis
ticks=4,begin=100,end=1000,redshift={D=.001/0},show redshift value]
```



Relativistic Doppler redshift $z=0.001$ ($v=.001c$; $\theta=0^\circ$)

```
\pgfspectra[element=He,back=visible40,gamma=.6,axis,axis step=50,axis
ticks=4,begin=100,end=1000,redshift=.5,show redshift value]
```



redshift $z=.5$

The options for `\pgfspectraplotshade`

This command creates a shade to use with the `\addplot` command provided by the PGF-PLOTS package. The shade starts at 0 nm and finishes at `shade end`. The shading could be adjusted using the following options:

- `shade end`
- `shade opacity`
- `shade opacity color`

`shade end`

default: 780

This value determines the end wavelength of the computed shading. The end wavelength should be set equal to the superior limit of the domain provided to the plot (see the [PGFPLOTS package documentation](#) for more information) and should be greater or equal to 381 nm. *(new in v2.1.0)*

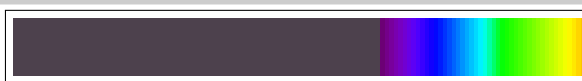
```
\pgfspectraplotshade{shadeDefault}
```

```
\fbox{\tikz{\fill[shading=shadeDefault] (0,0) rectangle (7.5,.75);}}
```



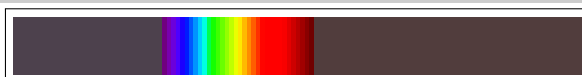
```
\pgfspectraplotshade[shade end=600]{shadeEnd600}
```

```
\fbox{\tikz{\fill[shading=shadeEnd600] (0,0) rectangle (7.5,.75);}}
```



```
\pgfspectraplotshade[shade end=1500]{shadeEnd1500}
```

```
\fbox{\tikz{\fill[shading=shadeEnd1500] (0,0) rectangle (7.5,.75);}}
```



`shade opacity`

default: 1

The opacity of the computed shade. '0' stands for 0% and the shading is totally transparent; '1' stands for 100% and the shading isn't transparent at all. *(new in v2.1.0)*

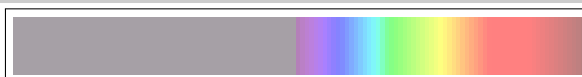
```
\pgfspectraplotshade{shadeDefault}
```

```
\fbox{\tikz{\fill[shading=shadeDefault] (0,0) rectangle (7.5,.75);}}
```



```
\pgfspectraplotshade[shade opacity=.5]{shadeOpacity50}
```

```
\fbox{\tikz{\fill[shading=shadeOpacity50] (0,0) rectangle (7.5,.75);}}
```



```
\pgfspectraplotshade[shade opacity=0]{shadeOpacity0}
```

```
\fbox{\tikz{\fill[shading=shadeOpacity0] (0,0) rectangle (7.5,.75);}}
```



shade opacity color

default: *white*

The background color of the computed shading. Only visible when **shade opacity** is lesser then 1. *(new in v2.1.0)*

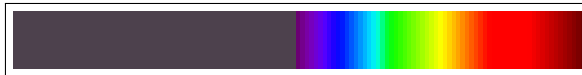
```
\pgfspectraplotshade{shadeDefault}
```

```
\fbox{\tikz{\fill[shading=shadeDefault] (0,0) rectangle (7.5,.75);}}
```



```
\pgfspectraplotshade[shade opacity color=black]{shadeOpacityBlack}
```

```
\fbox{\tikz{\fill[shading=shadeOpacityBlack] (0,0) rectangle (7.5,.75);}}
```



```
\pgfspectraplotshade[shade opacity color=black, shade  
opacity=.5]{shadeOpacityBlack50}
```

```
\fbox{\tikz{\fill[shading=shadeOpacityBlack50] (0,0) rectangle (7.5,.75);}}
```



```
\pgfspectraplotshade[shade opacity color=black, shade opac-  
ity=0]{shadeOpacityBlack100}
```

```
\fbox{\tikz{\fill[shading=shadeOpacityBlack100] (0,0) rectangle (7.5,.75);}}
```



The options for `\pgfspectrarainbow`

For the command `\pgfspectrarainbow` there are a set of options that control the rainbow drawn.

The specific rainbow options are:

- rainbow start
- rainbow knock out
- rainbow fade
- rainbow transparency
- rainbow background

Some TikZ keys that affect the rainbow are:

- `'color'`
- `opacity`
- `scope fading`

The default rainbow drawn is:

```
\pgfspectrarainbow{2cm}
```



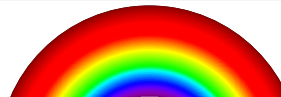
rainbow start

default: .6

The fraction from which the rainbow colors begin, relative to the center of a circle with radius 1. This value should be in the interval $[0,1]$.

(new in v2.1.0)

```
\pgfspectrarainbow(rainbow start=.8){2cm}% the rainbow colors starts at 1.6cm
\hspace{1cm}%
\pgfspectrarainbow(rainbow start=.4){2cm}% the rainbow colors starts at .8cm
```



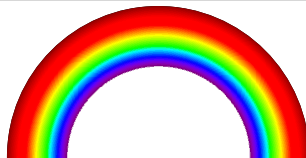
rainbow knock out

default: .4

The relative distance from the half-circle base to perform the clip. This value should be in the interval $[-1,1]$.

(new in v2.1.0)

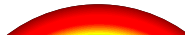
```
\pgfspectrarainbow(rainbow knock out=0){2cm}% the full half circle
```



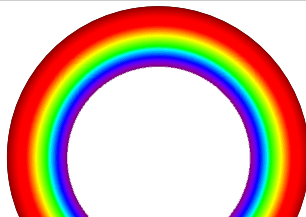
```
\pgfspectrarainbow(rainbow knock out=.4){2cm}% the default value
```



```
\pgfspectrarainbow(rainbow knock out=.8){2cm}% only 80% of the half circle is
shown
```



```
\pgfspectrarainbow(rainbow knock out=-.4){2cm}% «extending» the half-circle
```



rainbow fade

default: {}

Applies a scope fading in the clipped region (requires loading the TikZ fadings library). For more information about the fadings see the TikZ manual.

(new in v2.1.0)

```
%\usetikzlibrary{fadings}
...
\pgfspectrarainbow(rainbow fade=south){2cm}
\hspace{1cm}%
\pgfspectrarainbow(rainbow fade=west){2cm}
```

rainbow transparency

default: 0

The overall transparency of the rainbow. '0' (0%) stands for the fill colors in the rainbow without transparency ; '1' (100%) represents a totally transparent rainbow. *(new in v2.1.0)*

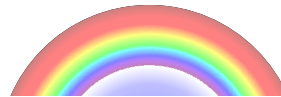
```
\pgfspectrarainbow(rainbow transparency=.5){2cm}
```

**rainbow background**

default: white

The background color below the rainbow (only visible with transparency). *(new in v2.1.0)*

```
\pgfspectrarainbow(rainbow background=blue){2cm}
\hspace{1cm}%
\pgfspectrarainbow(rainbow background=blue,rainbow transparency=.5){2cm}
```

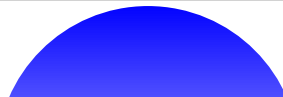


Some of the TikZ keys that affect the rainbow:

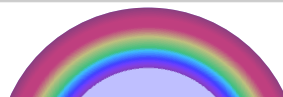
```
\pgfspectrarainbow[blue]{2cm}% Setting only the fill color only takes no effect
```



```
\pgfspectrarainbow[blue,scope fading=south]{2cm}
```



```
\pgfspectrarainbow[blue,opacity=.5]{2cm}
```



More examples in [Using `\pgfspectrarainbow`](#) `<[tikz options]><(rainbow options)>{radius}`.

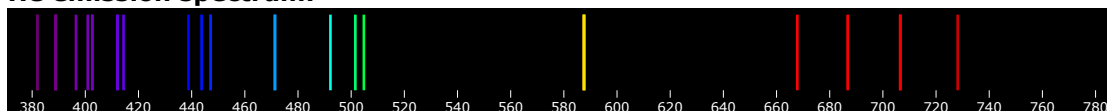
Examples

► Using \pgfspectra

Here are some examples for drawing some *eventually useful* spectra:

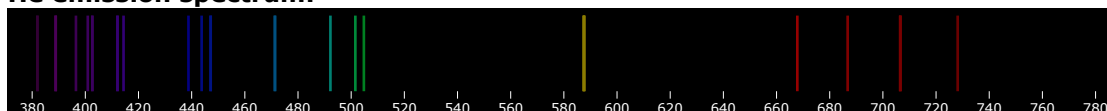
```
\pgfspectra[element=He,axis,label,label position=north west,
label after text=\ emission spectrum:]
```

He emission spectrum:



```
\pgfspectra[element=He,axis,label,label position=north west,label after text=
\ emission spectrum:,relative intensity,relative intensity threshold=.5]
```

He emission spectrum:

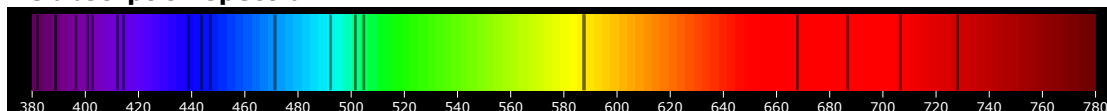


```
\pgfspectra[element=He,charge=all,line intensity=50,Imin=.05]
```



```
\pgfspectra[element=He,absorption,axis,label,label position=north west,label after
text=\ absorption spectrum:,relative intensity,relative intensity threshold=.5]
```

He absorption spectrum:



```
\pgfspectra[element=He,charge=all,absorption,line intensity=50]
```



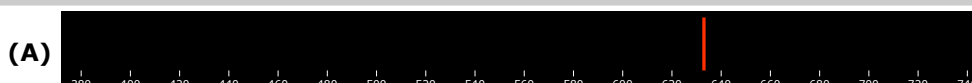
```
\pgfspectra[element=He,charge=all,relative intensity,back=visible75,gamma=2]
```



When the lines are manually inserted it's possible to use 'label before text' only with personalized text. In the next three examples 'label before text' is used to make labels for a multiple choice problem, omitting evidently the type of luminous font.

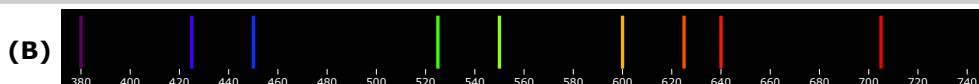
✓ Laser He-Ne

```
\pgfspectra[height=.7cm,end=740,lines={633},line
width=1.25pt,width=.75\linewidth,label,axis,label before text=(A),axis
font=\fontsize{4pt}{6pt}\selectfont]
```



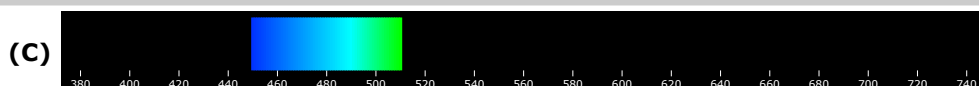
✓ Fluorescent lamp

```
\pgfspectra[height=.7cm,end=740,lines={380,425,450,525,550,600,625,640,705},
line width=1.25pt,width=.75\linewidth,label,axis,label before text=(B),axis
font=\fontsize{4pt}{6pt}\selectfont]
```



✓ Blue LED

```
\pgfspectra[height=.7cm,end=740,lines={450,451,452,453,454,455,456,457,458,459,
460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,
479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,
498,499,500,501,502,503,504,505,506,507,508,509,510},line width=1.25pt,
width=.75\linewidth,label,axis,label before text=(C),axis
font=\fontsize{4pt}{6pt}\selectfont]
```



✓ Sun like spectrum

```
\pgfspectra[element={H,Fe,Mg,Na},absorption,line intensity=40,Imin=.05]
```



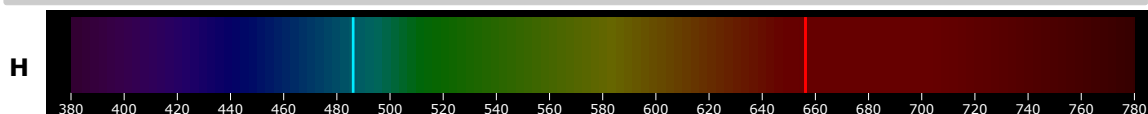
✓ Sirius like spectrum

```
\pgfspectra[element={H,He},absorption,line intensity=40,Imin=.05]
```

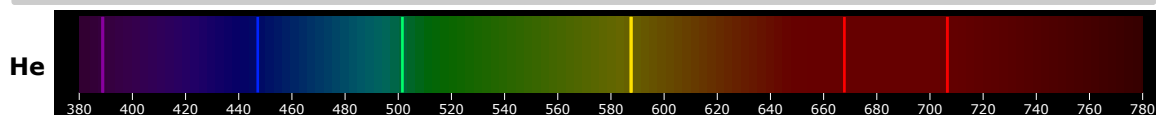


✓ "Classical" emission spectra of elements:

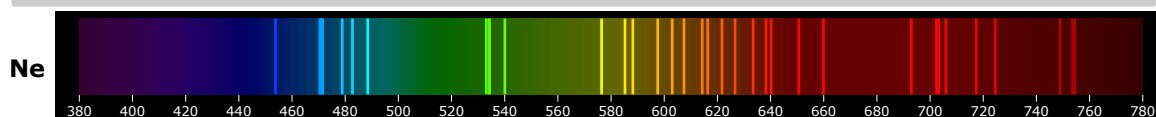
```
\pgfspectra[element=H,back=visible40,gamma=.6,label,axis,Imin=.05]
```



```
\pgfspectra[element=He,back=visible40,gamma=.6,label,axis,Imin=.05]
```

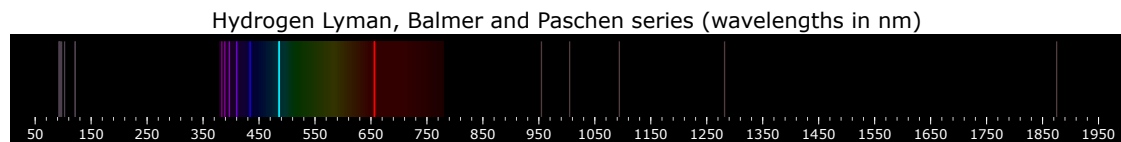


```
\pgfspectra[element=Ne,back=visible40,gamma=.6,label,axis,Imin=.05]
```



✓ Series of hydrogen:

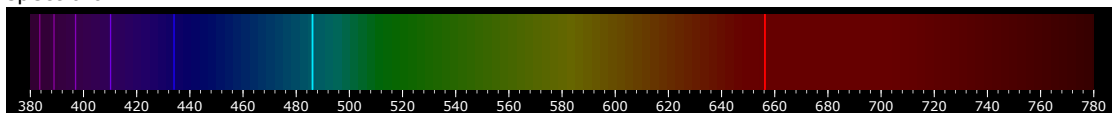
```
\pgfspectra[element=H,line width=.5pt,begin=50,end=1950,axis,axis
step=100,axis ticks=4,back=visible40,gamma=.6,brightness=.5,label,label
position=north,label font=\footnotesize,label after text={ydrogen Lyman, Balmer
and Paschen series (wavelengths in nm)}]
```



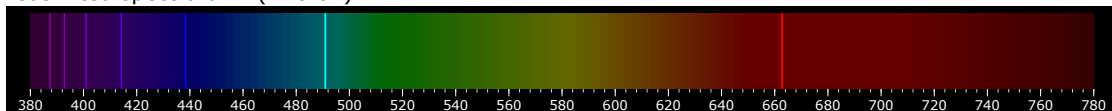
✓ Redshifted & Blueshifted lines of hydrogen using the \foreach statement:

```
\pgfspectraStyle[axis,axis ticks=4,back=visible40,gamma=.6,line width=.5pt]
\pgfspectra[element=H,label,label position=north west,label
font=\footnotesize,label before text={spectra of \ }]
\foreach \SQ/\z/\shift in {H/0.01/redshifted,H/-0.01/blueshifted}{
  \pgfspectra[element=\SQ,label,label position=north west,label
font=\footnotesize,label before text={\shift\ spectra of \ },label after
text={\ (z=\z)},redshift=\z]
}
\foreach \SQ/\z/\shift in {H/{D=0.01/0}/redshifted,H/{D=0.01/180}/blueshifted}{
  \pgfspectra[element=\SQ,label,label position=north west,label
font=\footnotesize,label before text={\shift\ spectra of \ },redshift=\z,show
redshift value]
}
```

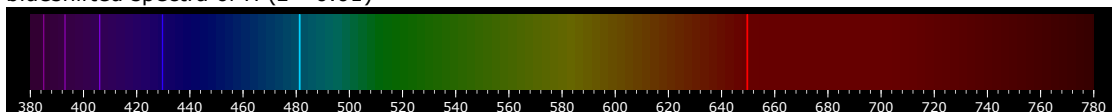
spectra of H



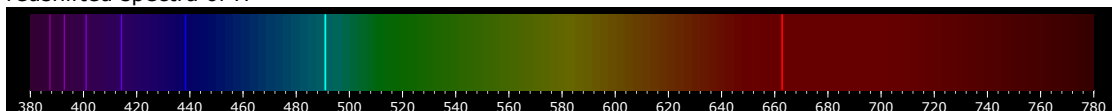
redshifted spectra of H (z=0.01)



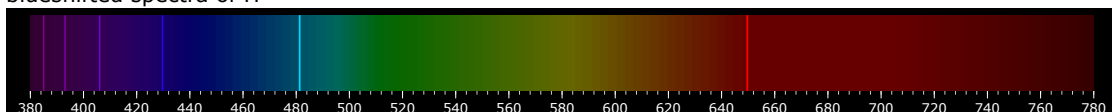
blueshifted spectra of H (z=-0.01)



redshifted spectra of H

Relativistic Doppler redshift z=0.01 ($v=0.01c$; $\theta=0^\circ$)

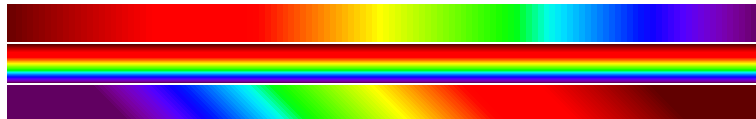
blueshifted spectra of H

Relativistic Doppler redshift z=-0.01 ($v=0.01c$; $\theta=180^\circ$)

► Using `\pgfspectrashade` in TikZ

Obviously, the *normal* TikZ keys used to control the shadings apply to the shading generated via `\pgfspectrashade`:

```
\pgfspectrashade(380,780){myShadeA}
\\ \tikz{\fill[0,0] (10,.5) rectangle (shading=myShadeA,shading angle=180);}
\\ \tikz{\fill[0,0] (10,.5) rectangle (shading=myShadeA,shading angle=90);}
\\ \tikz{\fill[0,0] (10,.5) rectangle (shading=myShadeA,shading angle=45);}
```



Providing an opacity to the drawing and applying a shade works well too:

```
\pgfspectrashade(380,780){myShadeA}
• on black background:
  \\ \tikz{\fill[0,0] (10,.5) rectangle (shading=myShadeA,opacity=.5);}
• on white background:
  \\ \tikz{\fill[0,0] (10,.5) rectangle (white,shading=myShadeA,opacity=.5);}
• on red background:
  \\ \tikz{\fill[0,0] (10,.5) rectangle (red,shading=myShadeA,opacity=.5);}
```

- on black background:



- on white background:



- on red background:



The gamma in the generated shade (via `\pgfspectrashade`) could be modified using the `'gamma'` key of `\pgfspectra` set by the command `\pgfspectraStyle`:

```
\pgfspectrashade(380,780){myShadeA}
\tikz{\fill[myShadeA] (0,0) rectangle (10,.5);}
```



```
\pgfspectraStyle[gamma=2]
\pgfspectrashade(380,780){myShadeGammaII}
\tikz{\fill[myShadeGammaII] (0,0) rectangle (10,.5);}
```



```
\pgfspectraStyle[gamma=10]
\pgfspectrashade(380,780){myShadeGammaX}
\tikz{\fill[myShadeGammaX] (0,0) rectangle (10,.5);}
\pgfspectraStyleReset
```

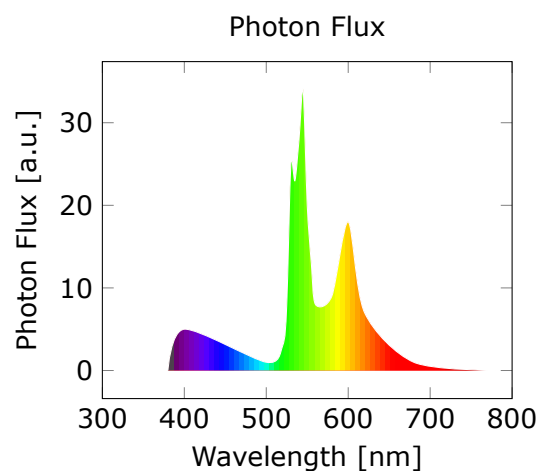


► Using `\pgfspectraplotshade` and `\pgfspectraplotmap` with PGFPLOTS

The command `\pgfspectraplotshade` is designed to build a shading to use with PGFPLOTS. Next examples show a few possibilities of how it could be used regarding two *sources*: a source of light and their photon flux and the blackbody spectral radiance.

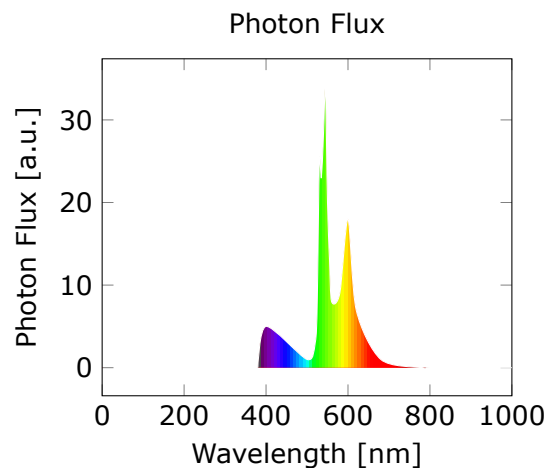
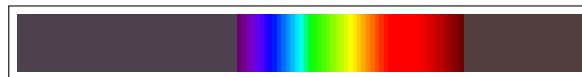
```
\pgfspectraplotshade{visiblespectrum}% default shading
```

```
\makebox[\linewidth][c]{%
\fbbox{\tikz{\fill[shading=visiblespectrum] (0,0) rectangle (7.5,.75);}}%
}%
\\ [10pt]\begin{tikzpicture}
\begin{axis}[%
title= Photon Flux,%
xlabel={Wavelength [nm]},%
ylabel={Photon Flux [a.u.]},%
xmin=300,%
xmax=800,%
]
\addplot[smooth, name path=spectrum,white] plot[] coordinates{%
( 380, 0 ) (400,5) ( 500, 1 ) ( 520, 3 ) ( 525, 8 ) ( 530, 25 )
( 535, 23 ) ( 540, 28 ) ( 545, 34 ) ( 550, 20 ) ( 555, 13 )
( 560, 8 ) ( 580, 9 ) ( 600, 18 ) ( 620, 7 ) ( 680, 1 ) ( 780, 0 )
};
\path[name path=axis] (axis cs:0,0) -- (axis cs:1,0);
\addplot+ [thick,shading=visiblespectrum]
fill between[of=spectrum and axis];
\end{axis}
\end{tikzpicture}%
```



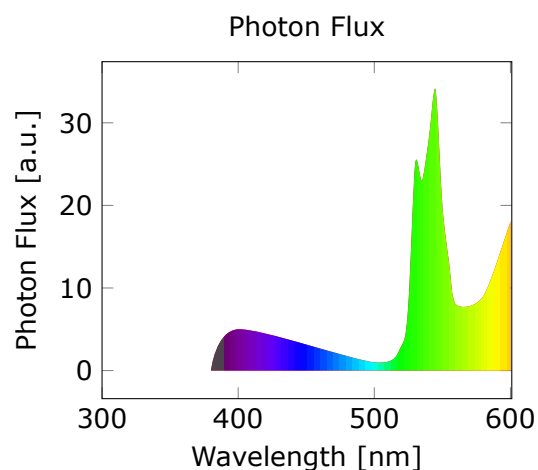
```
\pgfspectraplotshade[shade end=1000]{visiblespectrum}
```

```
\makebox[\linewidth][c]{%
\fbbox{\tikz{\fill[shading=visiblespectrum] (0,0) rectangle (7.5,.75);}}%
}%
\\ [10pt]\begin{tikzpicture}
\begin{axis}[%
title= Photon Flux,%
xlabel={Wavelength [nm]},%
ylabel={Photon Flux [a.u.]},%
xmin=0,%
xmax=1000,%
]
\addplot[smooth, name path=spectrum,white] plot[] coordinates{%
( 380, 0 ) (400,5) ( 500, 1 ) ( 520, 3 ) ( 525, 8 ) ( 530, 25 )
( 535, 23 ) ( 540, 28 ) ( 545, 34 ) ( 550, 20 ) ( 555, 13 )
( 560, 8 ) ( 580, 9 ) ( 600, 18 ) ( 620, 7 ) ( 680, 1 ) ( 780, 0 )
( 800, 0 ) ( 900, 0 ) ( 1000, 0 )
};
\path[name path=axis] (axis cs:0,0) -- (axis cs:1,0);
\addplot+ [thick,shading=visiblespectrum]
fill between[of=spectrum and axis];
\end{axis}
\end{tikzpicture}%
```



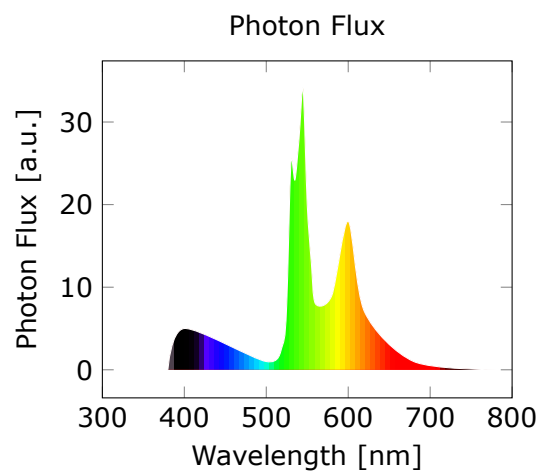
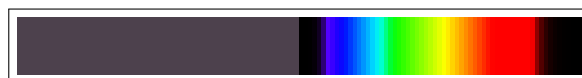
```
\pgfspectraplotshade[shade end=600]{visiblespectrum}
```

```
\makebox[\linewidth][c]{%
\fbbox{\tikz{\fill[shading=visiblespectrum] (0,0) rectangle (7.5,.75);}}%
}%
\\ [10pt]\\ \begin{tikzpicture}
\begin{axis}[%
title= Photon Flux,%
xlabel={Wavelength [nm]},%
ylabel={Photon Flux [a.u.]},%
xmin=300,%
xmax=601,%
]
\addplot[smooth, name path=spectrum,draw=none] plot[] coordinates{%
( 380, 0 ) (400,5) ( 500, 1 ) ( 520, 3 ) ( 525, 8 ) ( 530, 25 )
( 535, 23 ) ( 540, 28 ) ( 545, 34 ) ( 550, 20 ) ( 555, 13 )
( 560, 8 ) ( 580, 9 ) ( 600, 18 )
% to make the path smoothing to the horizontal axis -->
(600 ,17.9 ) (600 ,17 ) (600 ,15 ) (600 ,12 ) (600 ,6 )(600 ,0 )
};
\path[name path=axis] (axis cs:0,0) -- (axis cs:1,0);
\addplot+ [thick,shading=visiblespectrum]
fill between[of=spectrum and axis];
\end{axis}
\end{tikzpicture}%
```



```
\pgfspectraplotshade[gamma=10]{visiblespectrumGammaX}
```

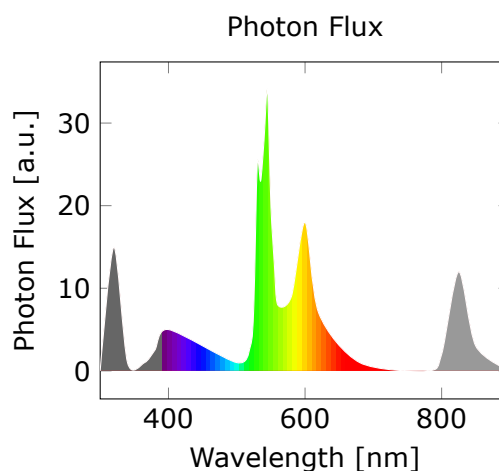
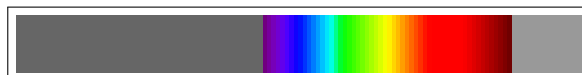
```
\makebox[\linewidth][c]{%
\fbbox{\tikz{\fill[shading=visiblespectrum] (0,0) rectangle (7.5,.75);}}%
}%
\\ [10pt]\\ \begin{tikzpicture}
\begin{axis}[%
title= Photon Flux,%
xlabel={Wavelength [nm]},%
ylabel={Photon Flux [a.u.]},%
xmin=300,%
xmax=800,%
]
\addplot[smooth, name path=spectrum,white] plot[] coordinates{%
( 380, 0 ) (400,5) ( 500, 1 ) ( 520, 3 ) ( 525, 8 ) ( 530, 25 )
( 535, 23 ) ( 540, 28 ) ( 545, 34 ) ( 550, 20 ) ( 555, 13 )
( 560, 8 ) ( 580, 9 ) ( 600, 18 ) ( 620, 7 ) ( 680, 1 ) ( 780, 0 )
};
\path[name path=axis] (axis cs:0,0) -- (axis cs:1,0);
\addplot+ [thick,shading=visiblespectrumGammaX]
fill between[of=spectrum and axis];
\end{axis}
\end{tikzpicture}%
```



Note: when setting the color for IR or UV within `\pgfspectraplotshade` make sure it doesn't end with '`!<number>`' like '`black!40`'; use '`black!40!white`' instead.

```
\pgfspectraplotshade[IRcolor=black!40!white,UVcolor=black!60!white,shade
end=900]{visiblespectrumIRUV}
```

```
\makebox[\linewidth][c]{%
\fbbox{\tikz{\fill[shading=visiblespectrumIRUV] (0,0) rectangle (7.5,.75);}}%
}%
\\ [10pt]\\ \begin{tikzpicture}
\begin{axis}[%
title= Photon Flux,%
xlabel={Wavelength [nm]},%
ylabel={Photon Flux [a.u.]},%
xmin=300,%
xmax=900,%
]%
\addplot[smooth, name path=spectrum,white] plot[] coordinates{%
( 300 , 0) (320, 15) (340, 1) (365, 1)
( 380, 2.5 ) (400,5) ( 500, 1 ) ( 520, 3 ) ( 525, 8 ) ( 530, 25 )
( 535, 23 ) ( 540, 28 ) ( 545, 34 ) ( 550, 20 ) ( 555, 13 )
( 560, 8 ) ( 580, 9 ) ( 600, 18 ) ( 620, 7 ) ( 680, 1 ) ( 780, 0 )
( 800, 2 ) (825, 12) (850, 3) ( 900, 0 )
};
\path[name path=axis] (axis cs:0,0) -- (axis cs:1,0);
\addplot+ [thick,shading=visiblespectrumIRUV]
fill between[of=spectrum and axis];
\end{axis}
\end{tikzpicture}%
```



For the *blackbody spectral radiance*, the Planck's distribution is used with the values:

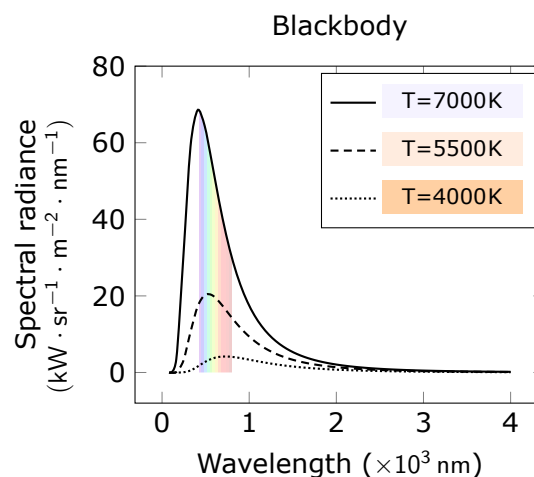
- $c = 3 \times 10^{14}$ microns \cdot s $^{-1}$ – speed of light
- $h = 6.626 \times 10^{22}$ kg \cdot microns $^2 \cdot$ s $^{-1}$ – Planck constant
- $k_B = 1.38 \times 10^{-11}$ kg \cdot microns $^2 \cdot$ s $^{-2} \cdot$ K $^{-1}$ – Boltzmann constant
- λ – wavelength (microns)
- T – temperature (K)
- Planck distribution: $B_\lambda = 2hc^2 \frac{1}{\lambda^5 e^{\frac{hc}{\lambda k_B T} - 1}}$ (kW \cdot sr $^{-1} \cdot$ m $^{-1} \cdot$ nm $^{-1}$)

The legend of the plots is created with the following definition:

```
\def\myentry#1{\tempercolor{#1}%
\tikz{\fill[tempercolor] (0,-.5pt) rectangle (40pt,.5pt)
node[midway,font=\footnotesize,anchor=mid] {\color{black} T=#1\hspace{.1ex}K}}}%
```

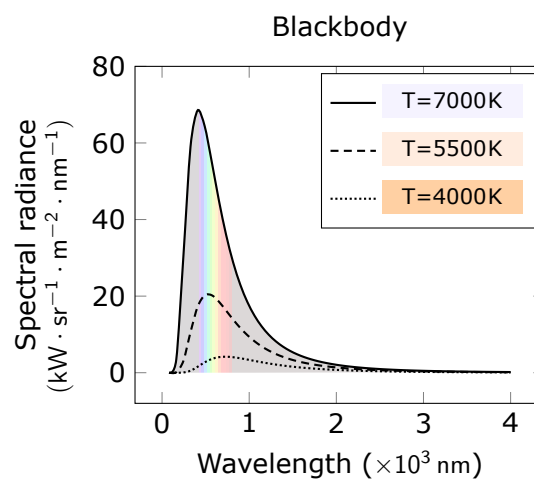
```
\pgfspectraplotshade[shade end=4000,IRcolor=white,UVcolor=white,gamma=.6,
shade opacity=.2]{BBody}
```

```
\makebox[\linewidth][c]{%
\fbbox{\tikz{\fill[shading=VISIBLESPECTRUM] (0,0) rectangle (7.5,.75)}}%
}%
\\ [10pt]\\ \begin{tikzpicture}
\begin{axis}[title=Blackbody,xlabel={Wavelength (\mathsf{\times 10^3 nm})},%
ylabel={\vbox{\hsize=120pt\vbox{\hfil Spectral radiance\hfil}&
\vbox{\mathsf{(kW \cdot sr^{-1} \cdot m^{-2} \cdot nm^{-1})}}}}},%
ymax=80,domain=0:4]%
\addplot[smooth, name path=spectrum,black,samples=50,thick] plot[]
{119.268/(x^5*(exp(14404/(x*7000))-1))};\addlegendentry{\myentry{7000}}%
\addplot[smooth,black,samples=50,densely dashed,thick] plot[]
{119.268/(x^5*(exp(14404/(x*5500))-1))};\addlegendentry{\myentry{5500}}%
\addplot[smooth,black,samples=50,densely dotted,thick] plot[]
{119.268/(x^5*(exp(14404/(x*4000))-1))};\addlegendentry{\myentry{4000}}%
\path[name path=axis] (axis cs:0,0) -- (axis cs:1,0);
\addplot+ [white,shading=BBody] fill between[of=spectrum and axis];
\end{axis}\end{tikzpicture}%
```



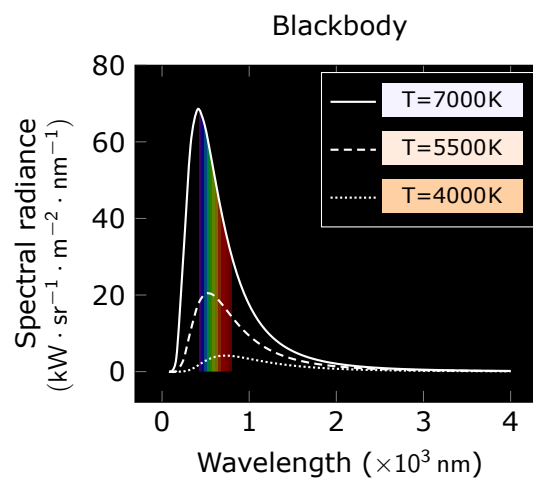
```
\pgfspectraplotshade[shade end=4000,shade opacity=.2,gamma=.6]{BBody}
```

```
\makebox[\linewidth][c]{%
\fbbox{\tikz{\fill[shading=VISIBLESPECTRUM] (0,0) rectangle (7.5,.75);}}%
}%
\\ [10pt]\\ \begin{tikzpicture}
\begin{axis}[title=Blackbody,xlabel={Wavelength ($\mathsf{\times 10^3 nm}$)},%
ylabel={\vbox{\hsize=120pt\vbox{\hfil Spectral radiance\hfil}&
\vbox{$\mathsf{(kW \cdot sr^{-1} \cdot m^{-2} \cdot nm^{-1})}$}}},%
ymax=80,domain=0:4]%
\addplot[smooth,name path=spectrum,black,samples=50,thick] plot[]
{119.268/(x^5*(exp(14404/(x*7000))-1))};\addlegendentry{\myentry{7000}}%
\addplot[smooth,black,samples=50,densely dashed,thick] plot[]
{119.268/(x^5*(exp(14404/(x*5500))-1))};\addlegendentry{\myentry{5500}}%
\addplot[smooth,black,samples=50,densely dotted,thick] plot[]
{119.268/(x^5*(exp(14404/(x*4000))-1))};\addlegendentry{\myentry{4000}}%
\path[name path=axis] (axis cs:0,0) -- (axis cs:1,0);
\addplot+ [white,shading=BBody] fill between[of=spectrum and axis];
\end{axis}\end{tikzpicture}%
```



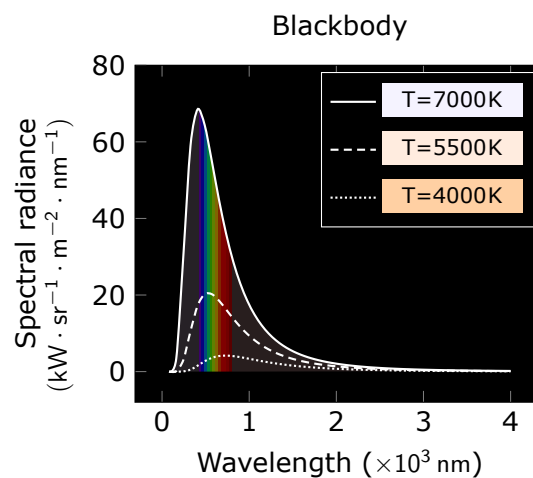

```
\pgfspectraplotshade[shade end=4000,IRcolor=black,UVcolor=black,gamma=.6,
shade opacity=.5,shade opacity color=black]{BBody}
```

```
\makebox[\linewidth][c]{%
\fbbox{\tikz{\fill[shading=VISIBLESPECTRUM] (0,0) rectangle (7.5,.75);}}%
}%
\\ [10pt]\\ \begin{tikzpicture}
\begin{axis}[title=Blackbody,xlabel={Wavelength ( $\times 10^3$  nm)},%
ylabel={\vbox{\hsize=120pt\vbox{\hfil Spectral radiance\hfil}&
\vbox{\mathsf{(kW\cdot sr^{-1}\cdot m^{-2}\cdot nm^{-1})}}}},%
ymax=80,domain=0:4]%
\addplot[smooth, name path=spectrum,black,samples=50,thick] plot[]
{119.268/(x^5*(exp(14404/(x*7000))-1))};\addlegendentry{\myentry{7000}}%
\addplot[smooth,black,samples=50,densely dashed,thick] plot[]
{119.268/(x^5*(exp(14404/(x*5500))-1))};\addlegendentry{\myentry{5500}}%
\addplot[smooth,black,samples=50,densely dotted,thick] plot[]
{119.268/(x^5*(exp(14404/(x*4000))-1))};\addlegendentry{\myentry{4000}}%
\path[name path=axis] (axis cs:0,0) -- (axis cs:4,0);
\addplot+ [white,shading=BBody] fill between[of=spectrum and axis];
\end{axis}\end{tikzpicture}%
```



```
\pgfspectraplotshade[shade end=4000,shade opacity=.5,gamma=.6,shade opacity
color=black]{BBody}
```

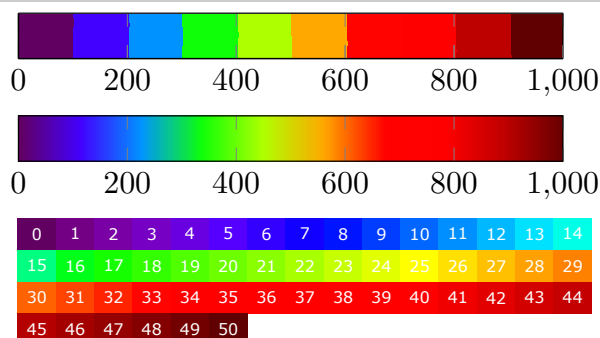
```
\makebox[\linewidth][c]{%
\fbbox{\tikz{\fill[shading=VISIBLESPECTRUM] (0,0) rectangle (7.5,.75);}}%
}%
\\ [10pt]\\ \begin{tikzpicture}
\begin{axis}[title=Blackbody,xlabel={Wavelength ($\mathsf{\times 10^3 nm}$)},%
ylabel={\vbox{\hsize=120pt\vbox{\hfil Spectral radiance\hfil}&
\vbox{$\mathsf{(kW \cdot sr^{-1} \cdot m^{-2} \cdot nm^{-1})}$}}},%
ymax=80,domain=0:4]%
\addplot[smooth, name path=spectrum,black,samples=50,thick] plot[]
{119.268/(x^5*(exp(14404/(x*7000))-1))};\addlegendentry{\myentry{7000}}%
\addplot[smooth,black,samples=50,densely dashed,thick] plot[]
{119.268/(x^5*(exp(14404/(x*5500))-1))};\addlegendentry{\myentry{5500}}%
\addplot[smooth,black,samples=50,densely dotted,thick] plot[]
{119.268/(x^5*(exp(14404/(x*4000))-1))};\addlegendentry{\myentry{4000}}%
\path[name path=axis] (axis cs:0,0) -- (axis cs:4,0);
\addplot+ [white,shading=BBody] fill between[of=spectrum and axis];
\end{axis}\end{tikzpicture}%
```



Next examples show possible usage of color maps feature of PGFPLOTS with the color map build with the `\pgfspectraplotmap` command:

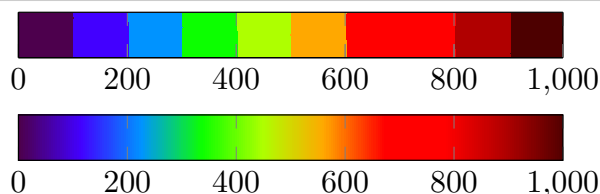
```
\pgfspectraplotmap{myColorMap}% default resolution (50 colors)
```

```
\pgfplotscolorbardrawstandalone[colormap={example}%
    {samples of colormap=(10 of myColorMap)},
colorbar horizontal,colormap access=const]
\\ \pgfplotscolorbardrawstandalone[colormap={example}%
    {samples of colormap=(10 of myColorMap)},
colorbar horizontal,colormap access=map]
\newdimen\y\y=0pt%
\\ \tikz{\foreach \i in {0,...,\pgfplotscolormaplastindexof{myColorMap}}{
    \pgfmathparse{int(mod(\i,15))}\edef\x{\pgfmathresult}%
    \fill[index of colormap={\i of myColorMap}]
        (12*\x pt,\y) rectangle ++(12pt,10pt)
        node[inner sep=0pt,midway,font=\tiny] {\color{white}\i};
    \ifnum\x=14\relax\global\advance\y by-10pt\fi}}
```



```
\pgfspectraplotmap[h]{myColorMap}% high resolution (400 colors)
% color(0) -> 380nm color(1) -> 381nm ... color(60) -> 380+60=440nm ...
% ... color(400) -> 780nm
```

```
\pgfplotscolorbardrawstandalone[colormap={example}%
    {samples of colormap=(10 of myColorMapH)},
colorbar horizontal,colormap access=const]
\\ \pgfplotscolorbardrawstandalone[colormap={example}%
    {samples of colormap=(10 of myColorMapH)},
colorbar horizontal,colormap access=map]
\newdimen\y\y=0pt%
\\ \tikz{\foreach \i in {0,...,\pgfplotscolormaplastindexof{myColorMap}}{
    \pgfmathparse{int(mod(\i,15))}\edef\x{\pgfmathresult}%
    \fill[index of colormap={\i of myColorMapH}]
        (12*\x pt,\y) rectangle ++(12pt,10pt)
        node[inner sep=0pt,midway,font=\tiny] {\color{white}\i};
    \ifnum\x=14\relax\global\advance\y by-10pt\fi}}
```



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
75	76	77	78	79	80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
135	136	137	138	139	140	141	142	143	144	145	146	147	148	149
150	151	152	153	154	155	156	157	158	159	160	161	162	163	164
165	166	167	168	169	170	171	172	173	174	175	176	177	178	179
180	181	182	183	184	185	186	187	188	189	190	191	192	193	194
195	196	197	198	199	200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254
255	256	257	258	259	260	261	262	263	264	265	266	267	268	269
270	271	272	273	274	275	276	277	278	279	280	281	282	283	284
285	286	287	288	289	290	291	292	293	294	295	296	297	298	299
300	301	302	303	304	305	306	307	308	309	310	311	312	313	314
315	316	317	318	319	320	321	322	323	324	325	326	327	328	329
330	331	332	333	334	335	336	337	338	339	340	341	342	343	344
345	346	347	348	349	350	351	352	353	354	355	356	357	358	359
360	361	362	363	364	365	366	367	368	369	370	371	372	373	374
375	376	377	378	379	380	381	382	383	384	385	386	387	388	389
390	391	392	393	394	395	396	397	398	399	400				

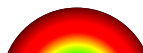
► Using `\pgfspectrarainbow`

Here are some examples of rainbows:

```
\pgfspectrarainbow{1cm}
```



```
\pgfspectrarainbow(rainbow start=0){1cm}
```



```
\pgfspectrarainbow(rainbow start=.4){1cm}
```



```
\pgfspectrarainbow(rainbow start=.8){1cm}
```



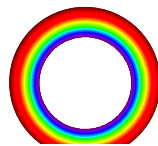
```
\pgfspectrarainbow(rainbow knock out=.8){1cm}
```



```
\pgfspectrarainbow(rainbow knock out=0){1cm}
```



```
\pgfspectrarainbow(rainbow knock out=-.8){1cm}
```



```
\pgfspectrarainbow(rainbow transparency=.5){1cm}
```



```
\pgfspectrarainbow(rainbow background=white){1cm}
```



```
\pgfspectrarainbow(rainbow background=blue,rainbow transparency=.5){1cm}
```



```
\pgfspectrarainbow(rainbow background=black,rainbow transparency=.5){1cm}
```



```
\pgfspectrarainbow(rainbow background=white,rainbow transparency=.5){1cm}
```



```
\pgfspectrarainbow(rainbow fade=south){1cm}
```

```
\pgfspectrarainbow(rainbow fade=north){1cm}
```

```
\pgfspectrarainbow[white,path fading=south]{1cm}
```



```
\pgfspectrarainbow[white](rainbow fade=south){1cm}
```

```
\pgfspectrarainbow[orange,path fading=west](rainbow fade=south){1cm}
```



```
\pgfspectrarainbow[blue,xslant=.1,opacity=.2]{1cm}
```



Alphabetical list of available options

\pgfspectra

key	description	type	default	value(s)
absorption	minimum intensity for the lines in the spectrum when using their relative intensities	boolean	false	{true, false}
axis	show or hide the axis	boolean	false	{true, false}
axis color	color of the axis	color	black	any named color or user defined color
axis font	font of the axis labels	font commands	{\tiny}	T _E X font commands
axis font color	color of the axis labels	color	white	any named color or user defined color
axis step	interval in nanometres between two major axis ticks	integer	20	[0; end-begin]nm
axis ticks	number of minor ticks	integer	0	{0,1,2,3,...}
axis unit	unit of the axis labels	text	nm	nm or micron or A
axis unit precision	number of significant digits (for values in nanometres) shown in axis labels	integer	3	{0,1,2,3,...}
back	spectrum background color	color	black	any named color or user defined color
backIRUV	IR and UV emission lines color in emission spectrum or background color of IR and UV regions in absorption spectrum	color	black	any named color or user defined color
backVIS	visible region background color in emission spectrum or emission lines color in absorption spectrum	color	black	any named color or user defined color
begin	first wavelength, in nanometres	integer	380	[10;4000]nm
brightness	brightness color correction as in the CMYK color model	decimal	1	[0;1]
charge	charge of the element(s)	integer	0	LSE Data: {0,1,2,3,4} NIST Data: {0,1}
element	chemical symbol of one element or comma sparated list of chemical symbols elements	text	NONE	H to Es except Fr
end	last wavelength, in nanometres	integer	780	[10;4000]nm
gamma	gamma color correction at the edges of the visible region	decimal	0.8	[0;1]
height	spectrum height	length	1cm	up to maximum T _E X dimension (16384pt)
Imin	minimum intensity of the lines	decimal	0	[0;1]
IRcolor	IR emission lines color in emission spectrum or background color of IR region in absorption spectrum	color	rgb(.3157,.2373,.2373)	any named color or user defined color
label	show or hide the axis labels	boolean	false	{true, false}
label after text	extra text to place after the label of the spectrum	text	{}	
label before text	extra text to place before the label of the spectrum	text	{}	
label font	font of the spectrum label	font commands	{\bfseries\small}	
label font color	color of the font of the spectrum label	color	black	any named color or user defined color
label position	position of the label of the spectrum	text	{west}	{west, north west, north, north east, east, south east, south, south west}
line intensity	draw all lines with the same intensity value	integer	100	{0,1,2,...,99,100}
line width	width of each line drawn in the spectrum	length	1pt	up to maximum T _E X dimension (16384pt)

\pgfspectra (continuation)

key	description	type	default	value(s)
lines	number or comma sparated list of numbers	integer or decimal	{}	[10;4000]nm
redshift	computes and draws the redshifted (or blueshifted) lines	text	{}	numeric value or {numeric value 1/numeric value 2}
relative intensity	draws the lines using their relative intensities	boolean	false	{true, false}
relative intensity threshold	all lines with intensity	decimal	0.25	[0;1]
show redshift value	show or hide the redshift value	boolean	false	{true, false}
use visible shading	visible region is drawn using a shading (instead of line by line)	boolean	true	{true, false}
UVcolor	UV emission lines color in emission spectrum or background color of UV region in absorption spectrum	color	rgb(.3,.2568,.3)	any named color or user defined color
width	spectrum width	length	{0.9\textwidth}	up to maximum \TeX dimension (16384pt)

\pgfspectraplotshade

key	description	type	default	value(s)
shade end	last wavelength, in nanometres	integer	780	[10;4000]nm
shade opacity	opacity of the computed shade	decimal	1	[0;1]
shade opacity color	background color of the computed shading	color	white	any named color or user defined color

\pgfspectrarainbow

key	description	type	default	value(s)
rainbow background	background color below the rainbow	color	white	any named color or user defined color
rainbow fade	scope fading in the clipped region	text	{}	any named $\text{\textit{TikZ}}$ fading or user defined fading
rainbow knock out	relative distance from the half-circle base to perform the clip	decimal	.4	[-1;1]
rainbow start	fraction from which the rainbow colors begin, relative to the center of a circle with radius 1	decimal	.6	[0;1]
rainbow transparency	overall transparency of the rainbow	decimal	0	[0;1]

Recommendations and known issues

The code could be a bit slow, so if there are many spectra to draw, the time consumption to get them could be high. In that case it's preferable to compile individual spectrum via the *preview* package, for later inclusion with `\includegraphics{<filename>.pdf}`:

```
% <filename>.tex
\documentclass{article}
\usepackage{pgf-spectra}
\usepackage[active,tightpage]{preview}
\PreviewEnvironment{tikzpicture}
\setlength\PreviewBorder{1pt}%
XXXXXXXXXXXXXXXXXXXX
\begin{document}
\pgfspectra[element=H,width=15cm]
\end{document}
```

Hint for T_EX 'limits':

If tex capacity exceeded when running...

«! TeX capacity exceeded, sorry [main memory size=2000001].»

just make a `\newpage` at the point of origin of the message (ejecting the page releases the T_EX memory!)

The code

```
1 % Hugo Gomes @ 15/04/2016 (v1.0)
2 % Hugo Gomes @ 15/03/2021 (v2.0.0)
3 % Hugo Gomes @ 12/05/2021 (v2.1.0)
4 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
5 \NeedsTeXFormat{LaTeX2e}%
6 \ProvidesPackage{pgf-spectra}[12/05/2021 pgf-spectra v2.1.0]%
7 \RequirePackage{tikz}%
8 \DeclareOption{LSE}{\input{spectra.data.LSE.tex}}%
9 \DeclareOption{NIST}{\input{spectra.data.NIST.tex}}%
10 \ExecuteOptions{NIST}%
11 \ProcessOptions\relax%
12 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
13 \definecolor{wlIRcolor}{rgb}{.3157,.2373,.2373}% NEW v2.0.0
14 \definecolor{wlUVcolor}{rgb}{.3,.2568,.3}% NEW v2.0.0
15 %
16 \newcount\wl@counta% tmp counter
17 \newcount\wl@countb% tmp counter
18 \newcount\wl@countc% tmp counter
19 \newcount\wl@countd% tmp counter
20 %
21 \newif\ifwl@absorption%
22 \newif\ifcur@elem@exist%
23 \newif\ifwl@drawaxis%
24 \newif\ifwl@axislabel%
25 \newif\ifwl@intensity%
26 \newif\ifwl@redshift% NEW v2.0.0
27 \newif\ifwl@RSvalue% NEW v2.0.0
28 \newif\ifwl@usevisiblesshade% NEW v2.1.0
29 % defining PGF keys
30 \pgfkeys{/wl/.cd,%
31 element/.get=\wl@element,%
32 element/.store in=\wl@element,%
33 element/.default=NONE,%
34 width/.get=\wl@width,%
35 width/.store in=\wl@width,%
```

```

36 width/.default={0.9\textwidth},%
37 height/.get=\wl@height,%
38 height/.store in=\wl@height,%
39 height/.default=1cm,%
40 back/.get=\wl@back,%
41 back/.store in=\wl@back,%
42 back/.default=black,%
43 backIRUV/.get=\wl@backnotvisible,% NEW v2.0.0
44 backIRUV/.store in=\wl@backnotvisible,% NEW v2.0.0
45 backIRUV/.default=black,% NEW v2.0.0
46 backVIS/.get=\wl@backvisible,% NEW v2.1.0
47 backVIS/.store in=\wl@backvisible,% NEW v2.1.0
48 backVIS/.default=black,% NEW v2.1.0
49 IRcolor/.get=\wl@IRcolor,% NEW v2.0.0
50 IRcolor/.store in=\wl@IRcolor,% NEW v2.0.0
51 IRcolor/.default=\wl@IRcolor,% NEW v2.0.0
52 UVcolor/.get=\wl@UVcolor,% NEW v2.0.0
53 UVcolor/.store in=\wl@UVcolor,% NEW v2.0.0
54 UVcolor/.default=\wl@UVcolor,% NEW v2.0.0
55 charge/.get=\wl@charge,%
56 charge/.store in=\wl@charge,%
57 charge/.default=0,%
58 Imin/.get=\wl@intmin,%
59 Imin/.store in=\wl@intmin,%
60 Imin/.default=0,%
61 lines/.get=\wl@lines,%
62 lines/.store in=\wl@lines,%
63 lines/.default={},%
64 line width/.get=\wl@linewidth,%
65 line width/.store in=\wl@linewidth,%
66 line width/.default=1pt,%
67 begin/.get=\wl@begin,%
68 begin/.store in=\wl@begin,%
69 begin/.default=380,%
70 end/.get=\wl@end,%
71 end/.store in=\wl@end,%
72 end/.default=780,%
73 axis step/.get=\wl@axisstep,%
74 axis step/.store in=\wl@axisstep,%
75 axis step/.default=20,%
76 axis unit/.get=\wl@axisunit,% NEW v2.1.0
77 axis unit/.store in=\wl@axisunit,% NEW v2.1.0
78 axis unit/.default=nm,% NEW v2.1.0
79 axis unit precision/.get=\wl@axisunitprecision,% NEW v2.1.0
80 axis unit precision/.store in=\wl@axisunitprecision,% NEW v2.1.0
81 axis unit precision/.default=3,% NEW v2.1.0
82 axis ticks/.get=\wl@axisticks,% NEW v2.0.0
83 axis ticks/.store in=\wl@axisticks,% NEW v2.0.0
84 axis ticks/.default=0,% NEW v2.0.0
85 axis color/.get=\wl@axiscolor,%
86 axis color/.store in=\wl@axiscolor,%
87 axis color/.default=black,%
88 axis font/.get=\wl@axisfont,%
89 axis font/.store in=\wl@axisfont,%
90 axis font/.default={\tiny},%
91 axis font color/.get=\wl@axisfontcolor,%
92 axis font color/.store in=\wl@axisfontcolor,%
93 axis font color/.default=white,%
94 label position/.get=\wl@labelposition,%
95 label position/.store in=\wl@labelposition,%
96 label position/.default={west},%
97 label before text/.get=\wl@labelbtext,%
98 label before text/.store in=\wl@labelbtext,%
99 label before text/.default={},%
100 label after text/.get=\wl@labelatext,%
101 label after text/.store in=\wl@labelatext,%
102 label after text/.default={},%
103 label font/.get=\wl@labelfont,%
104 label font/.store in=\wl@labelfont,%
105 label font/.default={\bfseries\small},%

```

```

106 label font color/.get=\wl@labelfontcolor,%
107 label font color/.store in=\wl@labelfontcolor,%
108 label font color/.default=black,%
109 gamma/.get=\wl@gamma,%
110 gamma/.store in=\wl@gamma,%
111 gamma/.default=0.8,%
112 brightness/.get=\wl@brightness,%
113 brightness/.store in=\wl@brightness,%
114 brightness/.default=1,%
115 line intensity/.get=\wl@lineint,%
116 line intensity/.store in=\wl@lineint,%
117 line intensity/.default=100,%
118 relative intensity threshold/.get=\wl@relintthresh,%
119 relative intensity threshold/.store in=\wl@relintthresh,%
120 relative intensity threshold/.default=0.25,%
121 absorption/.is if=\wl@absorption,%
122 axis/.is if=\wl@drawaxis,%
123 label/.is if=\wl@axislabel,%
124 relative intensity/.is if=\wl@intensity,%
125 redshift/.get=\wl@redshift,% NEW v2.0.0
126 redshift/.store in=\wl@redshift,% NEW v2.0.0
127 redshift/.default={},% NEW v2.0.0
128 show redshift value/.is if=\wl@RSvalue,% NEW v2.0.0
129 use visible shading/.is if=\wl@usevisibleshade% NEW v2.1.0
130 }%
131 % setting keys with default values
132 \pgfkeys{/wl/.cd,element,width,height,back,backIRUV,IRcolor,UVcolor,charge,Imin,lines,
    line width,begin,end,% NEW v2.0.0 -> backIRUV,IRcolor,UVcolor
133 axis color,axis font,axis font color,axis step,axis unit,axis unit precision,axis ticks
    ,label position,label before text,label after text,label font,label font color,gamma,
    brightness,line intensity,% NEW v2.1.0 -> axis unit,axis unit precision
134 relative intensity threshold,absorption=false,axis=false,label=false,relative intensity
    =false,redshift,show redshift value=false,% NEW v2.0.0 -> redshift, show redshift
    value
135 use visible shading,backVIS}% NEW v2.1.0 -> use visible shading
136 % strings for ifa tests
137 \def\wl@NONE{NONE}%
138 \def\wl@all{all}%
139 \def\wl@visible{visible}%
140 \def\wl@visible@list{visible,visible5,visible10,visible15,visible20,visible25,visible
    30,visible35,visible40,visible45,visible50,visible55,visible60,visible65,visible70,
    visible75,visible80,visible85,visible90,visible95,visible100}%
141 \def\wl@label@position@list{west,north west,north,north east,east,south east,south,
    south west}%
142 \def\wl@redshift@D{D}% NEW v2.0.0
143 \def\wl@axisunit@nm{nm}% NEW v2.1.0
144 \def\wl@axisunit@um{micron}% NEW v2.1.0
145 \def\wl@axisunit@A{A}% NEW v2.1.0
146 %%% COMMANDS
----->
147 % commands #####
148 \newif\ifpgfspectra@StyleIsDef\pgfspectra@StyleIsDeffalse% NEW v2.0.0
149 % \pgfspectra@Style[options]% NEW v2.0.0
150 \def\pgfspectra@Style[#1]{\pgfspectra@StyleReset\pgfspectra@StyleIsDeftrue\relax\tikzset
    {/wl/.cd,#1}%
151 \def\pgfspectra@DoStyle{\tikzset{/wl/.cd,#1}% applies storing user style for future
    use
152 }% NEW v2.0.0
153 % \pgfspectra@StyleReset% NEW v2.0.0
154 \def\pgfspectra@StyleReset{\pgfspectra@StyleIsDeffalse\tikzset{/wl/.cd,%
155 element=NONE,width=0.9\textwidth,height=1cm,back=black,backIRUV=black,IRcolor=\wl@IRcolor
    ,UVcolor=\wl@UVcolor,%
156 charge=0,Imin=0,lines={},line width=1pt,begin=380,end=780,axis color=black,axis font=\
    tiny,axis font color=white,axis step=20,axis unit=nm,axis unit precision=3,axis ticks
    =0,label position=west,% NEW v2.1.0 -> axis unit,axis unit precision
157 label before text={},label after text={},label font=\bfseries\small,label font color=
    black,gamma=0.8,brightness=1,line intensity=100,%
158 relative intensity threshold=0.25,absorption=false,axis=false,label=false,relative
    intensity=false,redshift={},show redshift value=false,% NEW v2.0.0 -> redshift, show
    redshift value

```

```

159 use visible shading,backVIS=black}}% NEW v2.1.0 -> use visible shading
160 % ----- The main command to draw the spectra
-----
161 % |pgfspectra[options]
162 \def\pgfspectra{\@ifnextchar[\wl@pgfspectra@withoptions{\wl@pgfspectra@nooptions}}%
163 \def\wl@pgfspectra@nooptions{\wl@pgfspectra@continuous(0.9\textwidth,1cm)}%
164 % #####
165 \def\wl@pgfspectra@continuous(#1,#2){\ignorespaces%
166 \ifwl@usevisibleshade% NEW v2.1.0
167 \pgfspectrashade(380,780){\wl@visibleshade}%
168 \tikz{\fill[shading=\wl@visibleshade] (0,0) rectangle (#1,#2);}%
169 \else%
170 \begin{tikzpicture}%
171 \pgfmathparse{#1/400}\edef\xscale{\pgfmathresult}
172 \pgfmathparse{1.4*\xscale+.09*\linewidth/\wl@width}\edef\wl@linewidth{\pgfmathresult}%
173 NEW v2.0.0 {\xscale} -> {1.4*\xscale+.09*\linewidth/\wl@width}
174 \foreach \x in {380,...,780}%
175 {%
176 \wlcolor{\x}%
177 \pgfmathparse{((x-380)*\xscale)}\edef\wl@currentx{\pgfmathresult pt}%
178 \draw[\wl@temp,line width=\wl@linewidth] (\wl@currentx,0) -- ++(0,#2);%
179 }%
180 \end{tikzpicture}%
181 \fi%
182 }% #####
183 \def\wl@pgfspectra@withoptions[#1]{\ignorespaces%
184 % setting default values or user style
185 \ifpgfspectra@StyleIsDef\pgfspectraStyleReset\pgfspectra@DoStyle\
186 pgfspectra@StyleIsDeftrue\relax\else\pgfspectraStyleReset\relax\fi% NEW v2.0.0
187 % process options (key values)
188 \pgfkeys{/wl/.cd,#1}%
189 % axis height
190 \setbox0=\hbox{\wl@axisfont\selectfont380}\edef\wl@axis@height{\the\ht0}%
191 % process visible background (visible+opacity)
192 \wl@counta=0%
193 \wl@countb=-1%
194 \@for\@myarg:=\wl@visible@list\do{%
195 \ifx\wl@back\@myarg\wl@countb=\wl@counta\fi%
196 \advance\wl@counta by1%
197 }%
198 \ifnum\wl@countb=-1\edef\@visible@opacity{1}\else% NEW v2.0.0
199 \ifnum\wl@countb=0\let\wl@back\wl@visible\edef\@visible@opacity{.5}\else%
200 \ifnum\wl@countb>0\let\wl@back\wl@visible\pgfmathparse{.05*\wl@countb}\edef\
201 @visible@opacity{\pgfmathresult}\fi\fi\fi%
202 %
-----
203 % check limits... % NEW v2.0.0
204 \ifnum\wl@end<10\relax\def\wldeez{10}\let\wl@end\wldeez\fi%
205 \ifnum\wl@end>4000\relax\def\wlquatomil{4000}\let\wl@end\wlquatomil\fi%
206 \ifnum\wl@begin<10\relax\def\wldeez{10}\let\wl@begin\wldeez\fi%
207 \ifnum\wl@begin>4000\relax\def\wlquatomil{4000}\let\wl@begin\wlquatomil\fi%
208 %
-----
209 % verifying redshift key
210 \ifx\wl@redshift\@empty\relax%
211 \wl@redshiftfalse%
212 \else%
213 \wl@processredshiftkey\wl@redshift\relax%
214 \fi%
215 %
-----
216 % if no element provided draws continuous spectrum with options or user list of lines
217 \ifx\wl@element\wl@NE%no element by the user
218 \ifx\wl@elt@chemsym\undefined\else\let\wl@elt@chemsym\undefined\fi%
219 \ifx\wl@lines\@empty%no lines by the user => continuous spectrum
220 % draws the continuous spectrum width options (default or by the user)

```

```

219 \begin{tikzpicture}%
220 \pgfkeys{/wl/.cd,#1}% NEW v2.0.0
221 \pgfmathparse{\wl@width/(abs(\wl@end-\wl@begin))}\edef\xscale{\pgfmathresult}%
222 \ifwl@drawaxis%draws the axis
223 \wl@utils@draw@axis%
224 \fi%|ifwl@drawaxis
225 \ifwl@axislabel%put the label
226 \wl@utils@put@label%
227 \fi%|ifwl@axislabel
228 \let\wl@back\wl@visible%
229 \let\wl@background@UVcolor\wl@backnotvisible\let\wl@background@IRcolor\
    wl@backnotvisible\relax%
230 \wl@utils@drawbackground{\@visible@opacity*\wl@brightness}%
231 \end{tikzpicture}%
232 \let\wl@list@{}%empty%
233 \else% lines by the user
234 \edef\wl@list@{\wl@lines}%
235 \let\wl@background@UVcolor\wl@backnotvisible\let\wl@background@IRcolor\
    wl@backnotvisible\relax%
236 \fi%|wl@lines|empty
237 \else%|wl@element|wlNONE
238 % else get element(s) data
239 \wl@countc=0%
240 \wl@countd=1%
241 \@for\@myarg:=\wl@element\do{\advance\wl@countc by1}%count number of elements
242 \wl@addt@list{\}%
243 \@for\@myarg:=\wl@element\do{%
244 \cur@elem@existtrue%
245 \def\wl@elt@chemsym{NOT FOUND!}%
246 \def\@search@result@err{NOT FOUND!}%
247 \wl@elt@data{\@myarg}\relax%
248 % check if element provided exists
249 \ifx\@search@result@err\wl@elt@chemsym Element\ '@\@myarg' with charge '\
    wl@charge' not found!\cur@elem@existfalse\else%
250 % if exists, set the wavelength's list
251 \wl@set@element@list{\wl@elt@elemdata}{\wl@elt@lmax}%
252 \fi%|@search@result@err|wl@elt@chemsym
253 \ifcur@elem@exist\ifnum\wl@countd<\wl@countc\wl@addt@list{\wl@list@}{,}\fi
    \fi%
254 \advance\wl@countd by1%
255 }%end do
256 \fi%|wl@element|wlNONE
257 % check if there are lines to draw and make the spectrum
258 \ifx\wl@list@{}%empty\ifx\wl@element\wlNONE\else Element\ '@\wl@element' with
    charge '\wl@charge' have no lines to display.\fi\else%
259 \ifwl@absorption%absorption spectrum
260 \begin{tikzpicture}%
261 \pgfkeys{/wl/.cd,#1}% NEW v2.0.0
262 \pgfmathparse{\wl@width/(abs(\wl@end-\wl@begin))}\edef\xscale{\
    pgfmathresult}%
263 \ifwl@drawaxis%draws the axis
264 \wl@utils@draw@axis%
265 \fi%|ifwl@drawaxis
266 \ifwl@axislabel%put the label
267 \wl@utils@put@label%
268 \fi%|ifwl@axislabel
269 \let\wl@back\wl@visible%
270 \let\wl@background@UVcolor\wl@UVcolor\let\wl@background@IRcolor\
    wl@IRcolor\relax%
271 \wl@utils@drawbackground{\wl@brightness}%
272 % draws the lines
273 \wl@utils@drawabsorptionlines%
274 \end{tikzpicture}%
275 \else%emission spectrum
276 % draws the spectrum
277 \ifx\wl@back\wl@visible%visible background
278 \begin{tikzpicture}%
279 \pgfkeys{/wl/.cd,#1}% NEW v2.0.0
280 \pgfmathparse{\wl@width/(abs(\wl@end-\wl@begin))}\edef\xscale{\
    pgfmathresult}%

```

```

281         \ifwl@drawaxis%draws the axis
282         \wl@utils@draw@axis%
283         \fi%|ifwl@drawaxis
284         \ifwl@axislabel%put the label
285         \wl@utils@put@label%
286         \fi%|ifwl@axislabel
287         \let\wl@background@UVcolor\wl@backnotvisible\let\wl@background@IRcolor\
          \wl@backnotvisible\relax%
288         \wl@utils@drawbackground{\@visible@opacity*\wl@brightness}%
289         \wl@utils@drawemissionlines% emission lines
290     \end{tikzpicture}%
291     \else%without visible background
292     \begin{tikzpicture}%
293         \pgfkeys{/wl/.cd,#1}% NEW v2.0.0
294         \pgfmathparse{\wl@width/(abs(\wl@end-\wl@begin))}\edef\xscale{\
          \pgfmathresult}%
295         \ifwl@drawaxis%draws the axis
296         \wl@utils@draw@axis%
297         \fi%|ifwl@drawaxis
298         \ifwl@axislabel%put the label
299         \wl@utils@put@label%
300         \fi%|ifwl@axislabel
301         \let\wl@background@UVcolor\wl@backnotvisible\let\wl@background@IRcolor\
          \wl@backnotvisible\relax%
302         \wl@utils@drawbackground{0}% dummy argument
303         \wl@utils@drawemissionlines% emission lines
304     \end{tikzpicture}%
305     \fi%|wl@back|\@visible
306     \fi%|ifwl@absorption
307     \fi% |wl@list@@|\@empty
308 }%
309 % #####
310 %XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
311 % get individual line data from one element of the array data
312 \def\wl@get@line@info[#1 #2 #3]{%
313 \def\@currentline@wl{#1}% return
314 \def\@currentline@int{#2}% return
315 \def\@currentline@charge{#3}% return
316 }%
317 % #####
318 % ##### |wl@set@element@list #####
319 % #####
320 %
321 % |wl@set@element@list{|wl@elt@elemdata}{|wl@elt@Imax}
322 % RETURN: |wl@list@@ -> (wl1,wl2,...)
323 % or if relative intensity true (between 0 and 1)
324 % |wl@list@@ -> (wl1/int1,wl2/int2,...)
325 %
326 \newif\ifwl@first% for first occurrence of Imin
327 \def\wl@set@element@list#1#2{\ignorespaces% |wl@elt@Imax
328 \wl@firsttrue%
329 \wl@counta=0%
330 \wl@countb=1%
331 \pgfmathparse{int(\wl@intmin*100)}\edef\@wl@intmin{\pgfmathresult}%intensity percentage
332 \ifnum\@wl@intmin=0% include all intensities
333 \ifx\wl@ll\wl@charge%ALL lines
334 \@for\@myarg:=#1\do{\advance\wl@counta by1}%count all entries
335     \ifwl@intensity%
336     \@for\@myarg:=#1%
337     \do{%
338     \expandafter\wl@get@line@info\@myarg%
339     \pgfmathparse{\wl@relintthresh+(1-\wl@relintthresh)*\@currentline@int/#2}\edef\
          \wl@intensity@to@list{\pgfmathresult}%
340     \ifnum\wl@countb<\wl@counta\wl@addt@list{\wl@list@@}{\@currentline@wl/\
          \wl@intensity@to@list,}%else%
341     \wl@addt@list{\wl@list@@}{\@currentline@wl/\wl@intensity@to@list}\fi%
342     \advance\wl@countb by1%
343     }%END do
344     \else%
345     \@for\@myarg:=#1%

```

```

346         \do{%
347         \expandafter\wl@get@line@info\@myarg%
348         \ifnum\wl@countb<\wl@counta\wl@addt@list{\wl@list@@}{\@currentline@wl,}\else%
349         \wl@addt@list{\wl@list@@}{\@currentline@wl}\fi%
350         \advance\wl@countb by 1%
351         }%END do
352         \fi%
353 \else% lines for one specific charge
354 \@for\@myarg:=#1\do{\expandafter\wl@get@line@info\@myarg\ifx\@currentline@charge\
    wl@charge\advance\wl@counta by 1\fi}%count only if is the desired charge
355     \ifwl@intensity%
356     \@for\@myarg:=#1%
357     \do{%
358     \expandafter\wl@get@line@info\@myarg%
359     \pgfmathparse{\wl@relinthresh+(1-\wl@relinthresh)*\@currentline@int/#2}\edef\
        wl@intensity@to@list{\pgfmathresult}%
360     \ifx\@currentline@charge\wl@charge%add to list if is the desired charge
361     \ifnum\wl@countb<\wl@counta\wl@addt@list{\wl@list@@}{\@currentline@wl/
        wl@intensity@to@list,}\else%
362     \wl@addt@list{\wl@list@@}{\@currentline@wl/\wl@intensity@to@list}\fi%
363     \advance\wl@countb by 1%
364     \fi%
365     }%END do
366     \else%
367     \@for\@myarg:=#1%
368     \do{%
369     \expandafter\wl@get@line@info\@myarg%
370     \ifx\@currentline@charge\wl@charge%add to list if is the desired charge
371     \ifnum\wl@countb<\wl@counta\wl@addt@list{\wl@list@@}{\@currentline@wl,}\
        else%
372     \wl@addt@list{\wl@list@@}{\@currentline@wl}\fi%
373     \advance\wl@countb by 1%
374     \fi%
375     }%END do
376     \fi%
377 \fi%
378 \else%|wl@intmin>0 & |wl@intmin<1
379 \ifnum\@wl@intmin>100\else%
380 \pgfmathparse{\wl@intmin*#2}\edef\wl@actual@int{\pgfmathresult}%
381 \ifx\wl@ll\wl@charge%ALL lines
382 \@for\@myarg:=#1\do{\advance\wl@counta by 1}%count all entries
383     \ifwl@intensity%
384     \@for\@myarg:=#1%
385     \do{%
386     \expandafter\wl@get@line@info\@myarg%
387     \pgfmathparse{notless(\@currentline@int,\wl@actual@int)}\relax\edef\
        wl@int@result{\pgfmathresult}%
388     \ifnum\wl@int@result=1%
389     \pgfmathparse{\wl@relinthresh+(1-\wl@relinthresh)*\@currentline@int/#2}\
        edef\wl@intensity@to@list{\pgfmathresult}%
390     \ifwl@first\wl@addt@list{\wl@list@@}{\@currentline@wl/\wl@intensity@to@list
        }\else%
391     \wl@addt@list{\wl@list@@}{,\@currentline@wl/\wl@intensity@to@list}\fi%
392     \ifwl@first\wl@firstfalse\fi%
393     \fi%
394     \advance\wl@countb by 1%
395     }%END do
396     \else%
397     \@for\@myarg:=#1%
398     \do{%
399     \expandafter\wl@get@line@info\@myarg%
400     \pgfmathparse{notless(\@currentline@int,\wl@actual@int)}\relax\edef\
        wl@int@result{\pgfmathresult}%
401     \ifnum\wl@int@result=1%
402     \ifwl@first\wl@addt@list{\wl@list@@}{\@currentline@wl}\else%
403     \wl@addt@list{\wl@list@@}{,\@currentline@wl}\fi%
404     \ifwl@first\wl@firstfalse\fi%
405     \fi%
406     \advance\wl@countb by 1%
407     }%END do

```

```

408     \fi%
409 \else% lines for one specific charge
410 \@for\@myarg:=#1\do{\expandafter\wl@get@line@info\@myarg\ifx\@currentline@charge\
    wl@charge\advance\wl@counta by 1\fi}%count only if is the desired charge
411     \ifwl@intensity%
412     \@for\@myarg:=#1%
413     \do{%
414     \expandafter\wl@get@line@info\@myarg%
415     \ifx\@currentline@charge\wl@charge%add to list if is the desired charge
416     \pgfmathparse{notless(\@currentline@int,\wl@actual@int)}\edef\wl@int@result
        {\pgfmathresult}%
417     \ifnum\wl@int@result=1%
418     \pgfmathparse{\wl@relintthresh+(1-\wl@relintthresh)*\@currentline@int/#2}\
        \edef\wl@intensity@to@list{\pgfmathresult}%
419     \ifwl@first\wl@addt@list{\wl@list@@}{\@currentline@wl/\wl@intensity@to@list
        }\else%
420     \wl@addt@list{\wl@list@@}{,\@currentline@wl/\wl@intensity@to@list}\fi%
421     \ifwl@first\wl@firstfalse\fi%
422     \fi%
423     \advance\wl@countb by 1%
424     \fi%
425 }%END do
426 \else%
427 \@for\@myarg:=#1%
428 \do{%
429 \expandafter\wl@get@line@info\@myarg%
430 \ifx\@currentline@charge\wl@charge%add to list if is the desired charge
431 \pgfmathparse{notless(\@currentline@int,\wl@actual@int)}\edef\wl@int@result
    {\pgfmathresult}%
432 \ifnum\wl@int@result=1%
433 \ifwl@first\wl@addt@list{\wl@list@@}{\@currentline@wl}\else%
434 \wl@addt@list{\wl@list@@}{,\@currentline@wl}\fi%
435 \ifwl@first\wl@firstfalse\fi%
436 \fi%
437 \advance\wl@countb by 1%
438 \fi%
439 }%END do
440 \fi%
441 \fi%
442 \fi%
443 \fi%
444 }%
445 % add to list
446 \def\wl@addt@list#1#2{\edef\wl@list@@{#1#2}}%
447 % internal utils
448 % internal utils
449 % internal utils
450 \def\wl@utils@draw@axis{\ignorespaces%
451     % axis unit -> NEW v2.1.0
452     \pgfkeys{/pgf/number format/.cd,fixed,precision=\wl@axisunitprecision,
        set thousands separator={},assume math mode=true}
453     \ifx\wl@axisunit\wl@axisunit@nm\relax%
454     \def\wl@axisunit@scale{1}\def\wl@axisunit@addzeros{}%
455     \else\ifx\wl@axisunit\wl@axisunit@um\relax%
456     \def\wl@axisunit@scale{1000}\def\wl@axisunit@addzeros{}\pgfkeys{/
        pgf/number format/.cd,fixed zerofill}%
457     \else\ifx\wl@axisunit\wl@axisunit@A\relax%
458     \def\wl@axisunit@scale{1}\def\wl@axisunit@addzeros{0}% probably not
        the best solution to overcome the TeX dimension limit values
        (16384pt)... but works!
459     \fi\fi\fi%
460     \ifnum\wl@begin>\wl@end%
461     % New xshift={-2.5*\@wl@axis@height} to hold bigger numbers, e.g. 2500
462     \draw[draw=none,fill=\wl@axiscolor] ([xshift={2.5*\@wl@axis@height}]0,\
        wl@height+2.5pt) rectangle ([xshift={-2.5*\@wl@axis@height}]-\
        wl@width,-2.5*\@wl@axis@height);%
463     % minor ticks -> NEW v2.0.0
464     \ifnum\wl@axisticks>0\relax%
465     \pgfmathparse{\wl@end+\wl@axisstep/(\wl@axisticks+1)}\
        \pgfmathparse{int(\pgfmathresult)}%

```



```

466         \edef\@axis@list{\wl@end,\pgfmathresult,...,\wl@begin}%
467         \foreach \x in \@axis@list%
468         {%
469             \pgfmathparse{(\wl@end-\x)*\xscale}\edef\wl@currentx{\
                pgfmathresult pt}%
470             \draw[\wl@axisfontcolor!80!transparent,line width=.25pt] (\
                wl@currentx,-.375*\@wl@axis@height) -- ++(0,.375*\
                @wl@axis@height);%
471         }%
472         \fi%
473     \pgfmathparse{\wl@end+\wl@axisstep}\pgfmathparse{int(\pgfmathresult)}%
474     \edef\@axis@list{\wl@end,\pgfmathresult,...,\wl@begin}%
475     \foreach \x in \@axis@list%
476     {%
477         \pgfmathparse{(\wl@end-\x)*\xscale}\edef\wl@currentx{\pgfmathresult pt}%
478         \draw[\wl@axisfontcolor,line width=.25pt] (\wl@currentx,-.75*\
            @wl@axis@height) -- ++(0,.75*\@wl@axis@height);%
479         \pgfmathparse{\x/\wl@axisunit@scale}\edef\wl@xscaledvalue{\
            pgfmathresult}% NEW v2.1.0
480         \node[\wl@axisfontcolor,font=\wl@axisfont,above,inner sep=0pt] at (\
            wl@currentx,-2.25*\@wl@axis@height) {\pgfmathprintnumber{\
            wl@xscaledvalue}\wl@axisunit@addzeros};%
481     }%
482     \else%
483     \draw[draw=none,fill=\wl@axiscolor] ([xshift={-2.5*\@wl@axis@height
        }]0,\wl@height+2.5pt) rectangle ([xshift={2.5*\@wl@axis@height}]\
        wl@width,-2.5*\@wl@axis@height);%
484         % minor ticks -> NEW v2.0.0
485         \ifnum\wl@axisticks>0\relax%
486             \pgfmathparse{\wl@begin+\wl@axisstep/(\wl@axisticks+1)}\
                pgfmathparse{int(\pgfmathresult)}%
487             \edef\@axis@list{\wl@begin,\pgfmathresult,...,\wl@end}%
488             \foreach \x in \@axis@list%
489             {%
490                 \pgfmathparse{(\x-\wl@begin)*\xscale}\edef\wl@currentx{\
                    pgfmathresult pt}%
491                 \draw[\wl@axisfontcolor!80!transparent,line width=.25pt] (\
                    wl@currentx,-.375*\@wl@axis@height) -- ++(0,.375*\
                    @wl@axis@height);%
492             }%
493             \fi%
494             \pgfmathparse{\wl@begin+\wl@axisstep}\pgfmathparse{int(\pgfmathresult)}%
495             \edef\@axis@list{\wl@begin,\pgfmathresult,...,\wl@end}%
496             \foreach \x in \@axis@list%
497             {%
498                 \pgfmathparse{(\x-\wl@begin)*\xscale}\edef\wl@currentx{\pgfmathresult
                    pt}%
499                 \draw[\wl@axisfontcolor,line width=.25pt] (\wl@currentx,-.75*\
                    @wl@axis@height) -- ++(0,.75*\@wl@axis@height);%
500                 \pgfmathparse{\x/\wl@axisunit@scale}\edef\wl@xscaledvalue{\
                    pgfmathresult}% NEW v2.1.0
501                 \node[\wl@axisfontcolor,font=\wl@axisfont,above,inner sep=0pt] at (\
                    wl@currentx,-2.25*\@wl@axis@height) {\pgfmathprintnumber{\
                    wl@xscaledvalue}\wl@axisunit@addzeros};%
502             }%
503             \fi%
504     }%
505     \def\wl@utils@put@label{\ignorespaces%
506         \ifx\wl@elt@chemsym\undefined\def\wl@elt@chemsym{}\fi%
507         \wl@get@label@position%
508         \ifnum\wl@begin>\wl@end%
509             \ifcase\wl@label@position%
510                 % west
511                 \ifwl@drawaxis%\ifwl@axislabel%
512                 \node[\wl@labelfontcolor,font=\wl@labelfont,left,minimum
                    width=2em,align=right] at (-2.5*\@wl@axis@height-\
                    wl@width,0.5*\wl@height-1.25*\@wl@axis@height) {\
                    wl@labelbtext\wl@elt@chemsym\wl@labelatext};%

```

```

513         \else%
514         \node[\wl@labelfontcolor,font=\wl@labelfont,left,minimum
           width=2em,align=right] at (-\wl@width,0.5*\wl@height) {\
           \wl@labelbtext\wl@elt@chemsym\wl@labelatext};%
515     \fi%
516 \or%north west
517     \ifwl@drawaxis%
518     \node[\wl@labelfontcolor,font=\wl@labelfont,above right,
           inner xsep=0pt] at (-2.5*\wl@axis@height-\wl@width,\
           \wl@height) {\wl@labelbtext\wl@elt@chemsym\wl@labelatext};
           %
519     \else%
520     \node[\wl@labelfontcolor,font=\wl@labelfont,above right,
           inner xsep=0pt] at (-\wl@width,\wl@height) {\
           \wl@labelbtext\wl@elt@chemsym\wl@labelatext};%
521     \fi%
522 \or%north
523     \node[\wl@labelfontcolor,font=\wl@labelfont,above] at (-0.5*\
           \wl@width,\wl@height) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
524 \or%north east
525     \ifwl@drawaxis%
526     \node[\wl@labelfontcolor,font=\wl@labelfont,above left,inner
           xsep=0pt] at (2.5*\wl@axis@height,\wl@height) {\
           \wl@labelbtext\wl@elt@chemsym\wl@labelatext};%
527     \else%
528     \node[\wl@labelfontcolor,font=\wl@labelfont,above left,inner
           xsep=0pt] at (0,\wl@height) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
529     \fi
530 \or%east
531     \ifwl@drawaxis%
532     \node[\wl@labelfontcolor,font=\wl@labelfont,right] at
           (2.5*\wl@axis@height,0.5*\wl@height-1.25*\
           \wl@axis@height) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
533     \else%
534     \node[\wl@labelfontcolor,font=\wl@labelfont,right] at
           (0,0.5*\wl@height) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
535     \fi%
536 \or%south east
537     \ifwl@drawaxis%
538     \node[\wl@labelfontcolor,font=\wl@labelfont,below left,
           inner xsep=0pt] at (2.5*\wl@axis@height,-2.5*\
           \wl@axis@height) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
539     \else%
540     \node[\wl@labelfontcolor,font=\wl@labelfont,below left,
           inner xsep=0pt] at (0,0) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
541     \fi%
542 \or%south
543     \ifwl@drawaxis%
544     \node[\wl@labelfontcolor,font=\wl@labelfont,below] at
           (-0.5*\wl@width,-2.5*\wl@axis@height) {\wl@labelbtext\
           \wl@elt@chemsym\wl@labelatext};%
545     \else%
546     \node[\wl@labelfontcolor,font=\wl@labelfont,below] at
           (-0.5*\wl@width,0) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
547     \fi%
548 \or%south west
549     \ifwl@drawaxis%
550     \node[\wl@labelfontcolor,font=\wl@labelfont,below right,
           inner xsep=0pt] at (-2.5*\wl@axis@height-\wl@width
           ,-2.5*\wl@axis@height) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
551     \else%

```

```

552         \node[\wl@labelfontcolor,font=\wl@labelfont,below right,
           inner xsep=0pt] at (-\wl@width,0) {\wl@labelbtext\
           \wl@elt@chemsym\wl@labelatext};%
553     \fi%
554 \fi%
555 \else%
556     \ifcase\wl@label@position%
557     %west
558         \ifwl@drawaxis%
559         \node[\wl@labelfontcolor,font=\wl@labelfont,left,minimum
           width=2em,align=right] at (-2.5*\wl@axis@height,0.5*\
           \wl@height-1.25*\wl@axis@height) {\wl@labelbtext\
           \wl@elt@chemsym\wl@labelatext};%
560         \else%
561         \node[\wl@labelfontcolor,font=\wl@labelfont,left,minimum
           width=2em,align=right] at (0,0.5*\wl@height) {\
           \wl@labelbtext\wl@elt@chemsym\wl@labelatext};%
562         \fi%
563     \or%north west
564         \ifwl@drawaxis%
565         \node[\wl@labelfontcolor,font=\wl@labelfont,above right,
           inner xsep=0pt] at (-2.5*\wl@axis@height,\wl@height) {\
           \wl@labelbtext\wl@elt@chemsym\wl@labelatext};%
566         \else%
567         \node[\wl@labelfontcolor,font=\wl@labelfont,above right,
           inner xsep=0pt] at (0,\wl@height) {\wl@labelbtext\
           \wl@elt@chemsym\wl@labelatext};%
568         \fi%
569     \or%north
570     \node[\wl@labelfontcolor,font=\wl@labelfont,above] at (0.5*\
           \wl@width,\wl@height) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
571     \or%north east
572     \ifwl@drawaxis%
573     \node[\wl@labelfontcolor,font=\wl@labelfont,above left,
           inner xsep=0pt] at (\wl@width+2.5*\wl@axis@height,\
           \wl@height) {\wl@labelbtext\wl@elt@chemsym\wl@labelatext};
           %
574     \else%
575     \node[\wl@labelfontcolor,font=\wl@labelfont,above left,
           inner xsep=0pt] at (\wl@width,\wl@height) {\wl@labelbtext
           \wl@elt@chemsym\wl@labelatext};%
576     \fi%
577     \or%east
578     \ifwl@drawaxis%
579     \node[\wl@labelfontcolor,font=\wl@labelfont,right] at ([
           xshift={2.5*\wl@axis@height}]\wl@width,0.5*\wl@height
           -1.25*\wl@axis@height) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
580     \else%
581     \node[\wl@labelfontcolor,font=\wl@labelfont,right] at (\
           \wl@width,0.5*\wl@height) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
582     \fi%
583     \or%south east
584     \ifwl@drawaxis%
585     \node[\wl@labelfontcolor,font=\wl@labelfont,below left,
           inner xsep=0pt] at (\wl@width+2.5*\wl@axis@height,-2.5*\
           \wl@axis@height) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
586     \else%
587     \node[\wl@labelfontcolor,font=\wl@labelfont,below left,
           inner xsep=0pt] at (\wl@width,0) {\wl@labelbtext\
           \wl@elt@chemsym\wl@labelatext};%
588     \fi%
589     \or%south
590     \ifwl@drawaxis%
591     \node[\wl@labelfontcolor,font=\wl@labelfont,below] at
           (0.5*\wl@width,-2.5*\wl@axis@height) {\wl@labelbtext\
           \wl@elt@chemsym\wl@labelatext};%

```

```

592         \else%
593         \node[\wl@labelfontcolor,font=\wl@labelfont,below] at
           (0.5*\wl@width,0) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
594         \fi%
595     \or%south west
596     \ifwl@drawaxis%
597     \node[\wl@labelfontcolor,font=\wl@labelfont,below right,
           inner xsep=0pt] at (-2.5*\wl@axis@height,-2.5*\
           \wl@axis@height) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
598     \else%
599     \node[\wl@labelfontcolor,font=\wl@labelfont,below right,
           inner xsep=0pt] at (0,0) {\wl@labelbtext\wl@elt@chemsym\
           \wl@labelatext};%
600     \fi%
601     \fi%
602 \fi%
603 }%
604 \def\wl@utils@drawbackground#1{\ignorespaces% NEW v2.0.0 - in this version draws from
           UV, VIS to IV -> replaces \wl@utils@visiblespectrum#1
605 % reprocess visible background (only visible) -> needed because of the override in keys
606     \wl@counta=0%
607     \wl@countb=-1%
608     \@for\@myarg:=\wl@visible@list\do{%
609         \ifx\wl@back\@myarg\wl@countb=\wl@counta\fi%
610         \advance\wl@counta by1%
611     }%
612     \ifnum\wl@countb>-1\let\wl@back\wl@visible\fi%
613     \ifwl@usevisibleshade\relax% NEW v2.1.0
614         \pgfmathparse{int(#1*100)}\edef\wl@bright{\pgfmathresult}%
615         \ifnum\wl@begin>\wl@end% 0
616             \ifnum\wl@end<380\relax% 1
617                 \pgfmathparse{(\wl@end-380)*\xscale}\edef\wl@pointA{\
pgfmathresult pt}%
618             \ifnum\wl@begin>780\relax% 2
619                 \pgfmathparse{(\wl@end-780)*\xscale}\edef\wl@pointB{\
pgfmathresult pt}%
620                 \draw[draw=none,fill=\wl@background@UVcolor] (0,0)
           rectangle (\wl@pointA,\wl@height);%
621                 \draw[draw=none,fill=\wl@background@IRcolor] (\wl@pointB,0)
           rectangle (-\wl@width,\wl@height);%
622                 \ifx\wl@back\wl@visible% 3 visible background
623                     \pgfspectrashade(380,780){\wl@visibleshade}%
624                     \fill[shading=\wl@visibleshade,shading angle=180] (\
           \wl@pointA,0) rectangle (\wl@pointB,\wl@height);%
625                 \else% 3
626                     \draw[draw=none,fill=\wl@back] (\wl@pointA,0)
           rectangle (\wl@pointB,\wl@height);%
627                 \fi% 3
628             \else% 2
629                 \pgfmathparse{(\wl@end-\wl@begin)*\xscale}\edef\wl@pointB{\
pgfmathresult pt}%
630                 \draw[draw=none,fill=\wl@background@UVcolor] (0,0)
           rectangle (\wl@pointA,\wl@height);%
631                 \ifx\wl@back\wl@visible% 3 visible background
632                     \pgfspectrashade(380,\wl@begin){\wl@visibleshade}%
633                     \fill[shading=\wl@visibleshade,shading angle=180] (\
           \wl@pointA,0) rectangle (\wl@pointB,\wl@height);%
634                 \else% 3
635                     \draw[draw=none,fill=\wl@back] (\wl@pointA,0)
           rectangle (\wl@pointB,\wl@height);%
636                 \fi% 3
637             \fi% 2
638         \else% 1
639             \ifnum\wl@begin>780\relax% 2
640                 \pgfmathparse{(\wl@end-780)*\xscale}\edef\wl@pointB{\
pgfmathresult pt}%
641                 \draw[draw=none,fill=\wl@background@IRcolor] (\wl@pointB,0)
           rectangle (-\wl@width,\wl@height);%

```

```

642         \ifx\wl@back\wl@visible% 3 visible background
643         \pgfspectrashade(\wl@end,780){\wl@visibleshade}%
644         \fill[shading=\wl@visibleshade,shading angle=180]
645             (0,0) rectangle (\wl@pointB,\wl@height);%
646     \else% 3
647         \draw[draw=none,fill=\wl@back] (0,0) rectangle (\wl@pointB,\wl@height);%
648     \fi% 3
649 \else% 2
650     \pgfmathparse{(\wl@end-\wl@begin)*\xscale}\edef\wl@pointB{\pgfmathresult pt}%
651     \ifx\wl@back\wl@visible% 3 visible background
652     \pgfspectrashade(\wl@end,\wl@begin){\wl@visibleshade}%
653     \fill[shading=\wl@visibleshade,shading angle=180]
654         (0,0) rectangle (\wl@pointB,\wl@height);%
655     \else% 3
656     \draw[draw=none,fill=\wl@back] (0,0) rectangle (\wl@pointB,\wl@height);%
657     \fi% 3
658 \fi% 2
659 \else% 1
660     \ifnum\wl@begin<380\relax% 1
661     \pgfmathparse{(380-\wl@begin)*\xscale}\edef\wl@pointA{\pgfmathresult pt}%
662     \ifnum\wl@end>780\relax% 2
663     \pgfmathparse{(780-\wl@begin)*\xscale}\edef\wl@pointB{\pgfmathresult pt}%
664     \draw[draw=none,fill=\wl@background@UVcolor] (0,0) rectangle (\wl@pointA,\wl@height);%
665     \draw[draw=none,fill=\wl@background@IRcolor] (\wl@pointB,0) rectangle (\wl@width,\wl@height);%
666     \ifx\wl@back\wl@visible% 3 visible background
667     \pgfspectrashade(380,780){\wl@visibleshade}%
668     \fill[shading=\wl@visibleshade] (\wl@pointA,0) rectangle (\wl@pointB,\wl@height);%
669     \else% 3
670     \draw[draw=none,fill=\wl@back] (\wl@pointA,0) rectangle (\wl@pointB,\wl@height);%
671     \fi% 3
672 \else% 2
673     \pgfmathparse{(\wl@end-\wl@begin)*\xscale}\edef\wl@pointB{\pgfmathresult pt}%
674     \draw[draw=none,fill=\wl@background@UVcolor] (0,0) rectangle (\wl@pointA,\wl@height);%
675     \ifx\wl@back\wl@visible% 3 visible background
676     \pgfspectrashade(380,\wl@end){\wl@visibleshade}%
677     \fill[shading=\wl@visibleshade] (\wl@pointA,0) rectangle (\wl@pointB,\wl@height);%
678     \else% 3
679     \draw[draw=none,fill=\wl@back] (\wl@pointA,0) rectangle (\wl@pointB,\wl@height);%
680     \fi% 3
681 \fi% 2
682 \else% 1
683     \ifnum\wl@end>780\relax% 2
684     \pgfmathparse{(780-\wl@begin)*\xscale}\edef\wl@pointB{\pgfmathresult pt}%
685     \draw[draw=none,fill=\wl@background@IRcolor] (\wl@pointB,0) rectangle (\wl@width,\wl@height);%
686     \ifx\wl@back\wl@visible% 3 visible background
687     \pgfspectrashade(\wl@begin,780){\wl@visibleshade}%
688     \fill[shading=\wl@visibleshade] (0,0) rectangle (\wl@pointB,\wl@height);%
689     \else% 3
690     \draw[draw=none,fill=\wl@back] (0,0) rectangle (\wl@pointB,\wl@height);%
691     \fi% 3
692 \else% 2

```

```

692 \pgfmathparse{(\wl@end-\wl@begin)*\xscale}\edef\wl@pointB{\
pgfmathresult pt}%
693 \ifx\wl@back\wl@visible% 3 visible background
694 \pgfspectrashade(\wl@begin,\wl@end){\wl@visibleshade
}%
695 \fill[shading=\wl@visibleshade] (0,0) rectangle (\
wl@pointB,\wl@height);%
696 \else% 3
697 \draw[draw=none,fill=\wl@back] (0,0) rectangle (\
wl@pointB,\wl@height);%
698 \fi% 3
699 \fi% 2
700 \fi% 1
701 \fi% 0
702 \else% \wl@usevisibleshade (don't use shading, draw line by line...)
703 \pgfmathparse{int(#1*100)}\edef\wl@bright{\pgfmathresult}%
704 \pgfmathparse{1.4*\xscale+.09*\linewidth/\wl@width}\edef\wl@linewidth
{\pgfmathresult}% NEW v2.0.0 {\xscale} -> {1.4*\xscale+.09*\linewidth
/\wl@width}
705 \ifnum\wl@begin>\wl@end% 0
706 \ifnum\wl@end<380\relax% 1
707 \pgfmathparse{(\wl@end-380)*\xscale}\edef\wl@pointA{\
pgfmathresult pt}%
708 \ifnum\wl@begin>780\relax% 2
709 \pgfmathparse{(\wl@end-780)*\xscale}\edef\wl@pointB{\
pgfmathresult pt}%
710 \draw[draw=none,fill=\wl@background@UVcolor] (0,0)
rectangle (\wl@pointA,\wl@height);%
711 \draw[draw=none,fill=\wl@background@IRcolor] (\wl@pointB,0)
rectangle (-\wl@width,\wl@height);%
712 \ifx\wl@back\wl@visible% 3 visible background
713 \foreach \x in {780,...,380}{%
714 \wlcolor{\x}%
715 \colorlet{\wlcolor}{\wl@temp!\wl@bright!\
wl@backvisible}% CHANGED v2.1.0
716 \pgfmathparse{\wl@pointB+(780-\x)*\xscale}\edef\
wl@currentx{\pgfmathresult pt}%
717 \draw[\wlcolor,line width=\wl@linewidth] (\
wl@currentx,0) -- ++(0,\wl@height);}%
718 \else% 3
719 \draw[draw=none,fill=\wl@back] (\wl@pointA,0)
rectangle (\wl@pointB,\wl@height);%
720 \fi% 3
721 \else% 2
722 \pgfmathparse{(\wl@end-\wl@begin)*\xscale}\edef\wl@pointB{\
pgfmathresult pt}%
723 \draw[draw=none,fill=\wl@background@UVcolor] (0,0)
rectangle (\wl@pointA,\wl@height);%
724 \ifx\wl@back\wl@visible% 3 visible background
725 \foreach \x in {\wl@begin,...,380}{%
726 \wlcolor{\x}%
727 \colorlet{\wlcolor}{\wl@temp!\wl@bright!\
wl@backvisible}% CHANGED v2.1.0
728 \pgfmathparse{\wl@pointB+(\wl@begin-\x)*\xscale}\
edef\wl@currentx{\pgfmathresult pt}%
729 \draw[\wlcolor,line width=\wl@linewidth] (\
wl@currentx,0) -- ++(0,\wl@height);}%
730 \else% 3
731 \draw[draw=none,fill=\wl@back] (\wl@pointA,0)
rectangle (\wl@pointB,\wl@height);%
732 \fi% 3
733 \fi% 2
734 \else% 1
735 \ifnum\wl@begin>780\relax% 2
736 \pgfmathparse{(\wl@end-780)*\xscale}\edef\wl@pointB{\
pgfmathresult pt}%
737 \draw[draw=none,fill=\wl@background@IRcolor] (\wl@pointB,0)
rectangle (-\wl@width,\wl@height);%
738 \ifx\wl@back\wl@visible% 3 visible background
739 \foreach \x in {780,...,\wl@end}{%

```

```

740         \wlcolor{\x}%
741         \colorlet{wlcolor}{wl@temp!\wl@bright!\
              wl@backvisible}% CHANGED v2.1.0
742         \pgfmathparse{\wl@pointB+(780-\x)*\xscale}\edef\
              wl@currentx{\pgfmathresult pt}%
743         \draw[wlcolor,line width=\wl@linewidth] (\
              wl@currentx,0) -- ++(0,\wl@height);}%
744     \else% 3
745         \draw[draw=none,fill=\wl@back] (0,0) rectangle (\
              wl@pointB,\wl@height);%
746     \fi% 3
747 \else% 2
748     \pgfmathparse{(\wl@end-\wl@begin)*\xscale}\edef\wl@pointB{\
              pgfmathresult pt}%
749     \ifx\wl@back\wl@visible% 3 visible background
750         \foreach \x in {\wl@begin,...,\wl@end}{%
751             \wlcolor{\x}%
752             \colorlet{wlcolor}{wl@temp!\wl@bright!\
                  wl@backvisible}% CHANGED v2.1.0
753             \pgfmathparse{\wl@pointB+(\wl@begin-\x)*\xscale}\
                  \edef\wl@currentx{\pgfmathresult pt}%
754             \draw[wlcolor,line width=\wl@linewidth] (\
                  wl@currentx,0) -- ++(0,\wl@height);}%
755         \else% 3
756             \draw[draw=none,fill=\wl@back] (0,0) rectangle (\
                  wl@pointB,\wl@height);%
757         \fi% 3
758     \fi% 2
759 \fi% 1
760 \else% 0
761     \ifnum\wl@begin<380\relax% 1
762         \pgfmathparse{(380-\wl@begin)*\xscale}\edef\wl@pointA{\
              pgfmathresult pt}%
763         \ifnum\wl@end>780\relax% 2
764             \pgfmathparse{(780-\wl@begin)*\xscale}\edef\wl@pointB{\
              pgfmathresult pt}%
765             \draw[draw=none,fill=\wl@background@UVcolor] (0,0)
              rectangle (\wl@pointA,\wl@height);%
766             \draw[draw=none,fill=\wl@background@IRcolor] (\wl@pointB,0)
              rectangle (\wl@width,\wl@height);%
767             \ifx\wl@back\wl@visible% 3 visible background
768                 \foreach \x in {380,...,780}{%
769                     \wlcolor{\x}%
770                     \colorlet{wlcolor}{wl@temp!\wl@bright!\
                          wl@backvisible}% CHANGED v2.1.0
771                     \pgfmathparse{\wl@pointB-(780-\x)*\xscale}\edef\
                          wl@currentx{\pgfmathresult pt}%
772                     \draw[wlcolor,line width=\wl@linewidth] (\
                          wl@currentx,0) -- ++(0,\wl@height);}%
773                 \else% 3
774                     \draw[draw=none,fill=\wl@back] (\wl@pointA,0)
                          rectangle (\wl@pointB,\wl@height);%
775                 \fi% 3
776             \else% 2
777                 \pgfmathparse{(\wl@end-\wl@begin)*\xscale}\edef\wl@pointB{\
              pgfmathresult pt}%
778                 \draw[draw=none,fill=\wl@background@UVcolor] (0,0)
              rectangle (\wl@pointA,\wl@height);%
779                 \ifx\wl@back\wl@visible% 3 visible background
780                     \foreach \x in {\wl@end,...,380}{%
781                         \wlcolor{\x}%
782                         \colorlet{wlcolor}{wl@temp!\wl@bright!\
                              wl@backvisible}% CHANGED v2.1.0
783                         \pgfmathparse{\wl@pointB-(\wl@end-\x)*\xscale}\edef\
                              \wl@currentx{\pgfmathresult pt}%
784                         \draw[wlcolor,line width=\wl@linewidth] (\
                              wl@currentx,0) -- ++(0,\wl@height);}%
785                     \else% 3
786                         \draw[draw=none,fill=\wl@back] (\wl@pointA,0)
                              rectangle (\wl@pointB,\wl@height);%

```

```

787         \fi% 3
788     \fi% 2
789 \else% 1
790     \ifnum\wl@end>780\relax% 2
791         \pgfmathparse{(780-\wl@begin)*\xscale}\edef\wl@pointB{\pgfmathresult pt}%
792         \draw[draw=none,fill=\wl@background@IRcolor] (\wl@pointB,0)
            rectangle (\wl@width,\wl@height);%
793         \ifx\wl@back\wl@visible% 3 visible background
794             \foreach \x in {\wl@begin,...,780}{%
795                 \wlcolor{\x}%
796                 \colorlet{\wlcolor}{\wl@temp!\wl@bright!\wl@backvisible}% CHANGED v2.1.0
797                 \pgfmathparse{(\x-\wl@begin)*\xscale}\edef\wl@currentx{\pgfmathresult pt}%
798                 \draw[\wlcolor,line width=\wl@linewidth] (\wl@currentx,0) -- ++(0,\wl@height);}%
799         \else% 3
800             \draw[draw=none,fill=\wl@back] (0,0) rectangle (\wl@pointB,\wl@height);%
801         \fi% 3
802     \else% 2
803         \pgfmathparse{(\wl@end-\wl@begin)*\xscale}\edef\wl@pointB{\pgfmathresult pt}%
804         \ifx\wl@back\wl@visible% 3 visible background
805             \foreach \x in {\wl@begin,...,\wl@end}{%
806                 \wlcolor{\x}%
807                 \colorlet{\wlcolor}{\wl@temp!\wl@bright!\wl@backvisible}% CHANGED v2.1.0
808                 \pgfmathparse{\x-\wl@begin)*\xscale}\edef\wl@currentx{\pgfmathresult pt}%
809                 \draw[\wlcolor,line width=\wl@linewidth] (\wl@currentx,0) -- ++(0,\wl@height);}%
810         \else% 3
811             \draw[draw=none,fill=\wl@back] (0,0) rectangle (\wl@pointB,\wl@height);%
812         \fi% 3
813     \fi% 2
814 \fi% 1
815 \fi% 0
816 \fi% \wl@usevisibleshade
817 }%
818 \def\wl@utils@drawabsorptionlines{\ignorespaces%
819     \ifnum\wl@begin>\wl@end%
820         \if\wl@intensity%
821             \if\wl@redshift\wl@utils@redshift\fi% NEW v2.0.0
822             \foreach \x/\y in \wl@list%%
823                 {%
824                     \pgfmathparse{notless(\x,\wl@end)}\edef\wl@x@nl{\pgfmathresult}%
825                     \pgfmathparse{notgreater(\x,\wl@begin)}\edef\wl@x@ng{\pgfmathresult}%
826                     \pgfmathparse{and(\wl@x@nl,\wl@x@ng)}\edef\wl@plot@point{\pgfmathresult}%
827                     \ifnum\wl@plot@point=1%
828                         \pgfmathparse{(\wl@end-\x)*\xscale}\edef\wl@currentx{\pgfmathresult pt}%
829                         \pgfmathparse{int(\y*100)}\edef\wl@black{\pgfmathresult}%
830                         \wlcolor{\x}%
831                         \colorlet{\wlcolor}{black!\wl@black!\wl@temp}%
832                         \draw[\wlcolor,line width=\wl@linewidth] (\wl@currentx,0) -- ++(0,\wl@height);%
833                     \fi%
834                 }%
835         \else%
836             \if\wl@redshift\wl@utils@redshift\fi% NEW v2.0.0
837             \foreach \x in \wl@list%%
838                 {%
839                     \pgfmathparse{notless(\x,\wl@end)}\edef\wl@x@nl{\pgfmathresult}%

```



```

840 \pgfmathparse{notgreater(\x,\wl@begin)}\edef\wl@x@ng{\
      pgfmathresult}%
841 \pgfmathparse{and(\wl@x@nl,\wl@x@ng)}\edef\wl@plot@point{\
      pgfmathresult}%
842 \ifnum\wl@plot@point=1%
843 \pgfmathparse{(\wl@end-\x)*\xscale}\edef\wl@currentx{\
      pgfmathresult pt}%
844 \wlcolor{\x}%
845 \colorlet{\wlcolor}{black!\wl@lineint!\wl@temp}%
846 \draw[\wlcolor,line width=\wl@linewidth] (\wl@currentx,0) --
      ++(0,\wl@height);%
847 \fi%
848 }%
849 \fi%
850 \else%
851 \ifwl@intensity%
852 \ifwl@redshift\wl@utils@redshift\fi% NEW v2.0.0
853 \foreach \x/\y in \wl@list@%
854 {%
855 \pgfmathparse{notless(\x,\wl@begin)}\edef\wl@x@nl{\
      pgfmathresult}%
856 \pgfmathparse{notgreater(\x,\wl@end)}\edef\wl@x@ng{\
      pgfmathresult}%
857 \pgfmathparse{and(\wl@x@nl,\wl@x@ng)}\edef\wl@plot@point{\
      pgfmathresult}%
858 \ifnum\wl@plot@point=1%
859 \pgfmathparse{(\x-\wl@begin)*\xscale}\edef\wl@currentx{\
      pgfmathresult pt}%
860 \pgfmathparse{int(\y*100)}\edef\wl@black{\pgfmathresult}%
861 \wlcolor{\x}%
862 \colorlet{\wlcolor}{black!\wl@black!\wl@temp}%
863 \draw[\wlcolor,line width=\wl@linewidth] (\wl@currentx,0) --
      ++(0,\wl@height);%
864 \fi%
865 }%
866 \else%
867 \ifwl@redshift\wl@utils@redshift\fi% NEW v2.0.0
868 \foreach \x in \wl@list@%
869 {%
870 \pgfmathparse{notless(\x,\wl@begin)}\edef\wl@x@nl{\
      pgfmathresult}%
871 \pgfmathparse{notgreater(\x,\wl@end)}\edef\wl@x@ng{\
      pgfmathresult}%
872 \pgfmathparse{and(\wl@x@nl,\wl@x@ng)}\edef\wl@plot@point{\
      pgfmathresult}%
873 \ifnum\wl@plot@point=1%
874 \pgfmathparse{(\x-\wl@begin)*\xscale}\edef\wl@currentx{\
      pgfmathresult pt}%
875 \wlcolor{\x}%
876 \colorlet{\wlcolor}{black!\wl@lineint!\wl@temp}%
877 \draw[\wlcolor,line width=\wl@linewidth] (\wl@currentx,0) --
      ++(0,\wl@height);%
878 \fi%
879 }%
880 \fi%
881 \fi%
882 \ifwl@RSvalue% NEW v2.0.0
883 \ifnum\wl@begin>\wl@end%
884 \ifwl@drawaxis\pgfmathparse{-\wl@width-2.5*\wl@axis@height}\edef\wl@redshiftinfo@x{\
      pgfmathresult pt}%
885 \else\pgfmathparse{-\wl@width}\edef\wl@redshiftinfo@x{\pgfmathresult pt}\fi%
886 \else%
887 \ifwl@drawaxis\pgfmathparse{-2.5*\wl@axis@height}\edef\wl@redshiftinfo@x{\
      pgfmathresult pt}%
888 \else\edef\wl@redshiftinfo@x{0pt}\fi%
889 \fi%
890 \ifwl@drawaxis\pgfmathparse{-.75*\wl@axis@height-1.3*\ht0-2pt}\edef\wl@redshiftinfo@y
      {\pgfmathresult pt}\else\edef\wl@redshiftinfo@y{0pt}\fi%
891 \node[below right,inner xsep=0pt,font=\wl@axisfont] at (\wl@redshiftinfo@x,\
      \wl@redshiftinfo@y) {\wl@redshiftinfo};%

```

```

892 \fi% NEW v2.0.0
893 }%
894 \def\wl@utils@drawemissionlines{\ignorespaces%
895     \ifnum\wl@begin>\wl@end%
896     \ifwl@intensity%
897         \ifwl@redshift\wl@utils@redshift\fi% NEW v2.0.0
898         \foreach \x/\y in \wl@list@@%
899             {%
900                 \wlcolor{\x}%
901                 \pgfmathparse{notless(\x,\wl@end)}\edef\wl@x@nl{\pgfmathresult}%
902                 \pgfmathparse{notgreater(\x,\wl@begin)}\edef\wl@x@ng{\pgfmathresult}%
903                 \pgfmathparse{and(\wl@x@nl,\wl@x@ng)}\edef\wl@plot@point{\pgfmathresult}%
904                 \ifnum\wl@plot@point=1%
905                     \pgfmathparse{(\wl@end-\x)*\xscale}\edef\wl@currentx{\pgfmathresult pt}%
906                     \pgfmathparse{int(\y*100)}\edef\wl@black{\pgfmathresult}%
907                     \colorlet{\wlcolor}{\wl@temp!\wl@black!black}%
908                     \draw[\wlcolor,line width=\wl@linewidth] (\wl@currentx,0) --
909                         ++(0,\wl@height);%
910                     \fi%
911                 }%
912             \else%
913                 \ifwl@redshift\wl@utils@redshift\fi% NEW v2.0.0
914                 \foreach \x in \wl@list@@%
915                     {%
916                         \wlcolor{\x}%
917                         \pgfmathparse{notless(\x,\wl@end)}\edef\wl@x@nl{\pgfmathresult}%
918                         \pgfmathparse{notgreater(\x,\wl@begin)}\edef\wl@x@ng{\pgfmathresult}%
919                         \pgfmathparse{and(\wl@x@nl,\wl@x@ng)}\edef\wl@plot@point{\pgfmathresult}%
920                         \ifnum\wl@plot@point=1%
921                             \pgfmathparse{(\wl@end-\x)*\xscale}\edef\wl@currentx{\pgfmathresult pt}%
922                             \colorlet{\wlcolor}{\wl@temp!\wl@lineint!black}%
923                             \draw[\wlcolor,line width=\wl@linewidth] (\wl@currentx,0) --
924                                 ++(0,\wl@height);%
925                             \fi%
926                         }%
927                     \else%
928                         \ifwl@intensity%
929                             \ifwl@redshift\wl@utils@redshift\fi% NEW v2.0.0
930                             \foreach \x/\y in \wl@list@@%
931                                 {%
932                                     \wlcolor{\x}%
933                                     \pgfmathparse{notless(\x,\wl@begin)}\edef\wl@x@nl{\pgfmathresult}%
934                                     \pgfmathparse{notgreater(\x,\wl@end)}\edef\wl@x@ng{\pgfmathresult}%
935                                     \pgfmathparse{and(\wl@x@nl,\wl@x@ng)}\edef\wl@plot@point{\pgfmathresult}%
936                                     \ifnum\wl@plot@point=1%
937                                         \pgfmathparse{(\x-\wl@begin)*\xscale}\edef\wl@currentx{\pgfmathresult pt}%
938                                         \pgfmathparse{int(\y*100)}\edef\wl@black{\pgfmathresult}%
939                                         \colorlet{\wlcolor}{\wl@temp!\wl@black!black}%
940                                         \draw[\wlcolor,line width=\wl@linewidth] (\wl@currentx,0) --
941                                             ++(0,\wl@height);%
942                                         \fi%
943                                     }%
944                                 \else%
945                                     \ifwl@redshift\wl@utils@redshift\fi% NEW v2.0.0
946                                     \foreach \x in \wl@list@@%
947                                         {%
948                                             \wlcolor{\x}%

```

```

947 \pgfmathparse{notless(\x,\wl@begin)}\edef\wl@x@n1{\
pgfmathresult}%
948 \pgfmathparse{notgreater(\x,\wl@end)}\edef\wl@x@ng{\
pgfmathresult}%
949 \pgfmathparse{and(\wl@x@n1,\wl@x@ng)}\edef\wl@plot@point{\
pgfmathresult}%
950 \ifnum\wl@plot@point=1%
951 \pgfmathparse{(\x-\wl@begin)*\xscale}\edef\wl@currentx{\
pgfmathresult pt}%
952 \colorlet{wlcolor}{wl@temp!\wl@lineint!black}%
953 \draw[wlcolor,line width=\wl@linewidth] (\wl@currentx,0) --
++(0,\wl@height);%
954 \fi%
955 }%
956 \fi%
957 \fi%
958 \ifwl@RSvalue% NEW v2.0.0
959 \ifnum\wl@begin>\wl@end%
960 \ifwl@drawaxis\pgfmathparse{-\wl@width-2.5*\wl@axis@height}\edef\wl@redshiftinfo@x{\
pgfmathresult pt}%
961 \else\pgfmathparse{-\wl@width}\edef\wl@redshiftinfo@x{\pgfmathresult pt}\fi%
962 \else%
963 \ifwl@drawaxis\pgfmathparse{-2.5*\wl@axis@height}\edef\wl@redshiftinfo@x{\
pgfmathresult pt}%
964 \else\edef\wl@redshiftinfo@x{0pt}\fi%
965 \fi%
966 \ifwl@drawaxis\pgfmathparse{-.75*\wl@axis@height-1.3*\ht0-2pt}\edef\wl@redshiftinfo@y
{\pgfmathresult pt}\else\edef\wl@redshiftinfo@y{0pt}\fi%
967 \node[below right,inner xsep=0pt,font=\wl@axisfont] at (\wl@redshiftinfo@x,\
wl@redshiftinfo@y) {\wl@redshiftinfo};%
968 \fi% NEW v2.0.0
969 }%
970 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
971 % return: integer with position (e.g. '0' for west, ... )
972 \def\wl@get@label@position{\ignorespaces%
973 \wl@countc=0%
974 \@for\@mylabel:=\wl@label@position@list%
975 \do{%
976 \ifx\@mylabel\wl@labelposition\edef\wl@label@position{\the\wl@countc}\fi%
977 \advance\wl@countc by1%
978 }%
979 }%
980 % NEW v2.0.0 -----
981 % redshift:  $\lambda(\text{obs}) = \lambda(\text{emit}) * \{(1+v/c * \cos(\theta)) / \sqrt{1-v^2/c^2}\}$  --->
Relativistic Doppler
982 % vbar -> v/c -> normalized velocity of the source (e.g. '0.9' for v=0.9c)
983 % theta -> angle between the direction of relative motion of the source and the
direction of emission in the observer's frame (zero angle is directly away from the
observer)
984 \def\wl@processredshiftkey#1{\ignorespaces%
985 \wl@redshiftfalse%
986 \edef\wl@redshiftkey@expand{#1}% to expand a value passed by a macro (not necessary if
the user provided a number...)
987 \expandafter\wl@redshiftkey@firstchar\wl@redshiftkey@expand\relax%
988 \ifcat1\wl@RedShift\relax%
989 \pgfmathparse{1+#1}\relax%
990 \edef\wl@UMMAISZ{\pgfmathresult}\wl@redshifttrue%
991 \edef\wl@redshiftinfo{redshift z=#1}%
992 \else%
993 \edef\wl@redshiftkey@expand{#1}%
994 \expandafter\wl@process@redshift\wl@redshiftkey@expand\relax%
995 \fi%
996 }%
997 \def\wl@redshiftkey@firstchar#1#2\relax{\edef\wl@RedShift{#1}}%
998 \def\wl@process@redshift#1=#2/#3\relax{%
999 \edef\wl@redshifttest{#1}%
1000 \ifx\wl@redshifttest\wl@redshift@D\relax%
1001 \pgfmathparse{(1+#2*cos(#3))/sqrt(1-#2*#2)}%
1002 \edef\wl@UMMAISZ{\pgfmathresult}\wl@redshifttrue%
1003 \pgfmathparse{#2*cos(#3)/sqrt(1-#2*#2)}%

```

```

1004 \edef\wl@redshiftinfo{Relativistic Doppler redshift z=\pgfmathresult\ (\mbox{v\hskip.1
      ex=\hskip.1ex\#2\hskip.1exc\hskip.5ex;\hskip.5ex\ensuremath{\theta}\hskip.1ex=\hskip.1
      ex\#3\ensuremath{\sim\circ}})}%
1005 \fi%
1006 }%
1007 % |wl@utils@redshift
-----
1008 % returns the wllist with the shift computed
1009 \def\wl@utils@redshift{\ignorespaces%
1010   \let\wt@backlist@@\wl@list@@\relax%
1011   \let\wl@list@@@empty\relax%
1012   \wl@firsttrue\relax%
1013   \ifwl@intensity% list (lambda/intensity)
1014     \foreach \x/\y in \wt@backlist@@{%
1015       \pgfmathparse{\x*\wl@UMMAISZ}\edef\@currentline@wl{\pgfmathresult}%
1016       \ifwl@first\global\wl@addt@list{\wl@list@@}{\@currentline@wl/\y}\else%
1017         \global\wl@addt@list{\wl@list@@}{,\@currentline@wl/\y}\fi%
1018       \ifwl@first\global\wl@firstfalse\fi%
1019     }%
1020   \else% list (lambda)
1021     \foreach \x in \wt@backlist@@{%
1022       \pgfmathparse{\x*\wl@UMMAISZ}\edef\@currentline@wl{\pgfmathresult}%
1023       \ifwl@first\global\wl@addt@list{\wl@list@@}{\@currentline@wl}\else%
1024         \global\wl@addt@list{\wl@list@@}{,\@currentline@wl}\fi%
1025       \ifwl@first\global\wl@firstfalse\fi%
1026     }%
1027   \fi%
1028 }%
1029 %XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1030 % NEW v2.1.0 ----->
1031 %XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1032 % |pgfspectrashade[type=<h/v>](start,end){name} ---> type: h-horizontal (default) / v-
      vertical
1033 \def\pgfspectrashade{\ignorespaces\@ifnextchar[\wl@pgfspectrashade{\wl@pgfspectrashade[
      h]}%
1034 \def\wl@pgfspectrashade[#1](#2,#3)#4{\ignorespaces%
1035 \ifnum#2>#3\relax\PackageError{pgf-spectra}%
1036 {in \textbackslash pgfspectrashade the starting wavelength (#2) must be lesser then the
      finishing wavelength (#3)}%
1037 {Try typing the wavelengths in the correct order: \textbackslash pgfspectrashade(#3,#2)
      ...}\else%
1038 \ifnum#3<381\relax\PackageError{pgf-spectra}{in \textbackslash pgfspectrashade the
      finishing wavelength (#3) must be at least 381...}%
1039 {Please type a finishing wavelength greater then 380...}\else%
1040 \ifnum#2<380\relax\def\wl@shade@begin{380}\else\pgfmathparse{int(#2)}\edef\
      wl@shade@begin{\pgfmathresult}\fi%
1041 \ifnum#3>780\relax\def\wl@shade@end{780}\else\pgfmathparse{int(#3)}\edef\wl@shade@end{\
      pgfmathresult}\fi%
1042 \pgfmathparse{(\wl@shade@end-\wl@shade@begin)/50}\edef\wl@shadecolor@step{\
      pgfmathresult}%
1043 %\foreach \n in {1,...,51}{%
1044 \for\n
      :={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36
      }do{%
1045 \pgfmathparse{\wl@shade@begin+(\n-1)*\wl@shadecolor@step}\edef\wl@currentwl{\
      pgfmathresult}%
1046 \wlcolor{\wl@currentwl}%
1047 \edef\wl@colorname{wlshcol\@Roman{n}}\relax\colorlet{\wl@colorname}{\wlcolor}%
1048 }%
1049 \def\wl@test{#1}\def\wl@v{v}\def\wl@h{h}%
1050 \ifx\wl@test\wl@h\relax%
1051 % the horizontal pgfshading
1052 \pgfdeclarehorizontalshading{#4}{100bp}{color(0bp)=(wlshcolI); color(25bp)=(wlshcolI);
      color(26bp)=(wlshcolII); color(27bp)=(wlshcolIII); color(28bp)=(wlshcolIV); color(29
      bp)=(wlshcolV); color(30bp)=(wlshcolVI); color(31bp)=(wlshcolVII); color(32bp)=(
      wlshcolVIII); color(33bp)=(wlshcolIX); color(34bp)=(wlshcolX); color(35bp)=(wlshcolXI
      ); color(36bp)=(wlshcolXII); color(37bp)=(wlshcolXIII); color(38bp)=(wlshcolXIV);
      color(39bp)=(wlshcolXV); color(40bp)=(wlshcolXVI); color(41bp)=(wlshcolXVII); color
      (42bp)=(wlshcolXVIII); color(43bp)=(wlshcolXIX); color(44bp)=(wlshcolXX); color(45bp)
      =(wlshcolXXI); color(46bp)=(wlshcolXXII); color(47bp)=(wlshcolXXIII); color(48bp)=(

```

```

    wlshcolXXIV); color(49bp)=(wlshcolXXV); color(50bp)=(wlshcolXXVI); color(51bp)=(
    wlshcolXXVII); color(52bp)=(wlshcolXXVIII); color(53bp)=(wlshcolXXIX); color(54bp)=(
    wlshcolXXX); color(55bp)=(wlshcolXXXI); color(56bp)=(wlshcolXXXII); color(57bp)=(
    wlshcolXXXIII); color(58bp)=(wlshcolXXXIV); color(59bp)=(wlshcolXXXV); color(60bp)=(
    wlshcolXXXVI); color(61bp)=(wlshcolXXXVII); color(62bp)=(wlshcolXXXVIII); color(63bp)=
    (wlshcolXXXIX); color(64bp)=(wlshcolXL); color(65bp)=(wlshcolXLI); color(66bp)=(
    wlshcolXLII); color(67bp)=(wlshcolXLIII); color(68bp)=(wlshcolXLIV); color(69bp)=(
    wlshcolXLV); color(70bp)=(wlshcolXLVI); color(71bp)=(wlshcolXLVII); color(72bp)=(
    wlshcolXLVIII); color(73bp)=(wlshcolXLIX); color(74bp)=(wlshcolL); color(75bp)=(
    wlshcolLI); color(100bp)=(wlshcolLI)}%
1053 \else\ifx\wl@test\wl@v\relax%
1054 % the vertical pgfshading
1055 \pgfdeclareverticalshading{#4}{100bp}{color(0bp)=(wlshcolI); color(25bp)=(wlshcolI);
    color(26bp)=(wlshcolII); color(27bp)=(wlshcolIII); color(28bp)=(wlshcolIV); color(29
    bp)=(wlshcolV); color(30bp)=(wlshcolVI); color(31bp)=(wlshcolVII); color(32bp)=(
    wlshcolVIII); color(33bp)=(wlshcolIX); color(34bp)=(wlshcolX); color(35bp)=(wlshcolXI
    ); color(36bp)=(wlshcolXII); color(37bp)=(wlshcolXIII); color(38bp)=(wlshcolXIV);
    color(39bp)=(wlshcolXV); color(40bp)=(wlshcolXVI); color(41bp)=(wlshcolXVII); color
    (42bp)=(wlshcolXVIII); color(43bp)=(wlshcolXIX); color(44bp)=(wlshcolXX); color(45bp)=
    (wlshcolXXI); color(46bp)=(wlshcolXXII); color(47bp)=(wlshcolXXIII); color(48bp)=(
    wlshcolXXIV); color(49bp)=(wlshcolXXV); color(50bp)=(wlshcolXXVI); color(51bp)=(
    wlshcolXXVII); color(52bp)=(wlshcolXXVIII); color(53bp)=(wlshcolXXIX); color(54bp)=(
    wlshcolXXX); color(55bp)=(wlshcolXXXI); color(56bp)=(wlshcolXXXII); color(57bp)=(
    wlshcolXXXIII); color(58bp)=(wlshcolXXXIV); color(59bp)=(wlshcolXXXV); color(60bp)=(
    wlshcolXXXVI); color(61bp)=(wlshcolXXXVII); color(62bp)=(wlshcolXXXVIII); color(63bp)=
    (wlshcolXXXIX); color(64bp)=(wlshcolXL); color(65bp)=(wlshcolXLI); color(66bp)=(
    wlshcolXLII); color(67bp)=(wlshcolXLIII); color(68bp)=(wlshcolXLIV); color(69bp)=(
    wlshcolXLV); color(70bp)=(wlshcolXLVI); color(71bp)=(wlshcolXLVII); color(72bp)=(
    wlshcolXLVIII); color(73bp)=(wlshcolXLIX); color(74bp)=(wlshcolL); color(75bp)=(
    wlshcolLI); color(100bp)=(wlshcolLI)}%
1056 \fi\fi%
1057 \fi\fi%
1058 }%
1059 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1060 \pgfkeys{/wl/.cd,%
1061 shade end/.get=\wl@shadeend,%
1062 shade end/.store in=\wl@shadeend,%
1063 shade end/.default=780,%
1064 shade opacity/.get=\wl@shade@opacity,%
1065 shade opacity/.store in=\wl@shade@opacity,%
1066 shade opacity/.default=1,%
1067 shade opacity color/.get=\wl@shade@opacitycolor,%
1068 shade opacity color/.store in=\wl@shade@opacitycolor,%
1069 shade opacity color/.default=white%
1070 }%
1071 \pgfkeys{/wl/.cd,shade end,shade opacity,shade opacity color}%
1072 % |pgfspectraplotshade{name} or
1073 % |pgfspectraplotshade[options]{name}
1074 \def\pgfspectraplotshade{\@ifnextchar{\wl@pgfspectraplotshade{\wl@pgfspectraplotshade
    []}}%
1075 \def\wl@pgfspectraplotshade[#1]#2{\ignorespaces%
1076 \pgfkeys{/wl/.cd,shade end=780,shade opacity=1,shade opacity color=white}%
1077 \pgfkeys{/wl/.cd,UVcolor=\wlUVcolor,IRcolor=\wlIRcolor,gamma=.8}%
1078 \pgfkeys{/wl/.cd,#1}%
1079 \pgfmathparse{100*\wl@shade@opacity}\edef\@wl@shade@opacity{\pgfmathresult}%
1080 %
1081 %  $\lambda$ [bp]=25+(380+8*(\ln-1))*(50/\wl@shadeend)
1082 %
1083 \edef\wl@shadeend@pt{\wl@shadeend pt}%
1084 \ifdim\wl@shadeend@pt<381pt\relax%
1085 \PackageError{pgf-spectra}{shade end must be greater then 380}{Type a wavelength
    greater then 380}%
1086 \else%
1087 \pgfkeys{/pgf/number format/.cd,fixed,precision=4,set thousands separator={},assume
    math mode=true}%
1088 \pgfmathparse{25+379.9/\wl@shadeend*50}%
1089 \pgfmathprintnumberto{\pgfmathresult}{\wl@X@tmp}%
1090 \edef\x0{\wl@X@tmp bp}%
1091 \ifdim\wl@shadeend@pt<780pt\relax%

```

```

1092 \pgfkeys{/pgf/number format/.cd,fixed,precision=3,set thousands separator={},assume
      math mode=true}%
1093 \for\n
      :={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36}
      do{%
1094 \pgfmathparse{380+(\n-1)*(\wl@shadeend/50-7.6)}\edef\wl@currentwl{\pgfmathresult}%
1095 \wcolor{\wl@currentwl}%
1096 \colorlet{\wcolor}{\wl@temp!\@wl@shade@opacity!\wl@shade@opacitycolor}%
1097 \edef\wl@colorname{\wlshcol\@Roman\n}\relax\colorlet{\wl@colorname}{\wcolor}%%
1098 \pgfmathparse{24+380/\wl@shadeend*50+\n-380/\wl@shadeend*\n+380/\wl@shadeend}%
1099 \pgfmathprintnumberto{\pgfmathresult}{\wl@X@tmp}%
1100 \expandafter\edef\csname x\@Roman\n\endcsname{\wl@X@tmp bp}%
1101 }%
1102 \pgfkeys{/pgf/number format/.cd,fixed,precision=4,set thousands separator={},assume
      math mode=true}%
1103 \pgfmathparse{25+(\wl@shadeend+.1)/\wl@shadeend*50}%
1104 \pgfmathprintnumberto{\pgfmathresult}{\wl@X@tmp}%
1105 \edef\xF{\wl@X@tmp bp}%
1106 \else%
1107 \pgfkeys{/pgf/number format/.cd,fixed,precision=3,set thousands separator={},assume
      math mode=true}%
1108 \for\n
      :={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36}
      do{%
1109 \pgfmathparse{380+(\n-1)*8}\edef\wl@currentwl{\pgfmathresult}%
1110 \wcolor{\wl@currentwl}%
1111 \colorlet{\wcolor}{\wl@temp!\@wl@shade@opacity!\wl@shade@opacitycolor}%
1112 \edef\wl@colorname{\wlshcol\@Roman\n}\relax\colorlet{\wl@colorname}{\wcolor}%%
1113 \pgfmathparse{25+380/\wl@shadeend*50+400/\wl@shadeend*\n-400/\wl@shadeend}%
1114 \pgfmathprintnumberto{\pgfmathresult}{\wl@X@tmp}%
1115 \expandafter\edef\csname x\@Roman\n\endcsname{\wl@X@tmp bp}%
1116 }%
1117 \pgfkeys{/pgf/number format/.cd,fixed,precision=2,set thousands separator={},assume
      math mode=true}%
1118 \pgfmathparse{25+780.1/\wl@shadeend*50}%
1119 \pgfmathprintnumberto{\pgfmathresult}{\wl@X@tmp}%
1120 \edef\xF{\wl@X@tmp bp}%
1121 \fi% ifdim<780pt
1122 % the horizontal shading
1123 \pgfdeclarehorizontalshading[#2]{100bp}{color(0bp)=(\wl@UVcolor!\@wl@shade@opacity!\
      wl@shade@opacitycolor);color(\x0)=(\wl@UVcolor!\@wl@shade@opacity!\
      wl@shade@opacitycolor);%
1124 color(\xI)=(wlshcolI);color(\xII)=(wlshcolII);color(\xIII)=(wlshcolIII);color(\xIV)=(
      wlshcolIV);color(\xV)=(wlshcolV);color(\xVI)=(wlshcolVI);color(\xVII)=(wlshcolVII);
      color(\xVIII)=(wlshcolVIII);color(\xIX)=(wlshcolIX);color(\xX)=(wlshcolX);color(\xXI)
      =(wlshcolXI);color(\xXII)=(wlshcolXII);color(\xXIII)=(wlshcolXIII);color(\xXIV)=(
      wlshcolXIV);color(\xXV)=(wlshcolXV);color(\xXVI)=(wlshcolXVI);color(\xXVII)=(
      wlshcolXVII);color(\xXVIII)=(wlshcolXVIII);color(\xXIX)=(wlshcolXIX);color(\xXX)=(
      wlshcolXX);color(\xXXI)=(wlshcolXXI);color(\xXXII)=(wlshcolXXII);color(\xXXIII)=(
      wlshcolXXIII);color(\xXXIV)=(wlshcolXXIV);color(\xXXV)=(wlshcolXXV);color(\xXXVI)=(
      wlshcolXXVI);color(\xXXVII)=(wlshcolXXVII);color(\xXXVIII)=(wlshcolXXVIII);color(\
      xXXIX)=(wlshcolXXIX);color(\xXXX)=(wlshcolXXX);color(\xXXXI)=(wlshcolXXXI);color(\
      XXXII)=(wlshcolXXXII);color(\xXXXIII)=(wlshcolXXXIII);color(\xXXXIV)=(wlshcolXXXIV);
      color(\xXXXV)=(wlshcolXXXV);color(\xXXXVI)=(wlshcolXXXVI);color(\xXXXVII)=(
      wlshcolXXXVII);color(\xXXXVIII)=(wlshcolXXXVIII);color(\xXXXIX)=(wlshcolXXXIX);color
      (\xXL)=(wlshcolXL);color(\xXLI)=(wlshcolXLI);color(\xXLII)=(wlshcolXLII);color(\
      xXLIII)=(wlshcolXLIII);color(\xXLIV)=(wlshcolXLIV);color(\xXLV)=(wlshcolXLV);color(\
      xXLVI)=(wlshcolXLVI);color(\xXLVII)=(wlshcolXLVII);color(\xXLVIII)=(wlshcolXLVIII);
      color(\xXLIX)=(wlshcolXLIX);color(\xL)=(wlshcolL);color(\xLI)=(wlshcolLI);%
1125 color(\xF)=(\wl@IRcolor!\@wl@shade@opacity!\wl@shade@opacitycolor);color(100bp)=(\
      wl@IRcolor!\@wl@shade@opacity!\wl@shade@opacitycolor)}
1126 \fi% ifdim<381pt
1127 }%
1128 %XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1129 % |pgfspectraplotmap[<l/h>]{name}
1130 % l -> low resolution (51 colors -> 380 to 780 nm ; step 8 nm) ! Default
1131 % h -> high resolution (401 colors -> 380 to 780 nm ; step 1 nm)
1132 \def\pgfspectraplotmap{\ignorespaces\ifnextchar[\wl@pgfspectraplotmap{\
      wl@pgfspectraplotmap[1]}}%
1133 \def\wl@pgfspectraplotmap[#1]#2{\ignorespaces%

```

```

1134 \def\wl@test{#1}\def\wl@l{1}\def\wl@h{h}%
1135 \ifx\wl@test\wl@l\relax%
1136 \@for\n
      :={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36}
      do{%
1137 \pgfmathparse{380+(\n-1)*8}\edef\wl@currentwl{\pgfmathresult}%
1138 \wcolor{\wl@currentwl}%
1139 \edef\wl@colname{\wlshcol\@Roman\n}\relax\colorlet{\wl@colname}{\wlcolor}%%
1140 }%
1141 % the pgfplots colormap
1142 \pgfplotsset{/pgfplots/colormap={#2}{color(0bp)=(wlshcolI); color(1bp)=(wlshcolII);
      color(2bp)=(wlshcolIII); color(3bp)=(wlshcolIV); color(4bp)=(wlshcolV); color(5bp)=(
      wlshcolVI); color(6bp)=(wlshcolVII); color(7bp)=(wlshcolVIII); color(8bp)=(wlshcolIX)
      ; color(9bp)=(wlshcolX); color(10bp)=(wlshcolXI); color(11bp)=(wlshcolXII); color(12
      bp)=(wlshcolXIII); color(13bp)=(wlshcolXIV); color(14bp)=(wlshcolXV); color(15bp)=(
      wlshcolXVI); color(16bp)=(wlshcolXVII); color(17bp)=(wlshcolXVIII); color(18bp)=(
      wlshcolXIX); color(19bp)=(wlshcolXX); color(20bp)=(wlshcolXXI); color(21bp)=(
      wlshcolXXII); color(22bp)=(wlshcolXXIII); color(23bp)=(wlshcolXXIV); color(24bp)=(
      wlshcolXXV); color(25bp)=(wlshcolXXVI); color(26bp)=(wlshcolXXVII); color(27bp)=(
      wlshcolXXVIII); color(28bp)=(wlshcolXXIX); color(29bp)=(wlshcolXXX); color(30bp)=(
      wlshcolXXXI); color(31bp)=(wlshcolXXXII); color(32bp)=(wlshcolXXXIII); color(33bp)=(
      wlshcolXXXIV); color(34bp)=(wlshcolXXXV); color(35bp)=(wlshcolXXXVI); color(36bp)=(
      wlshcolXXXVII); color(37bp)=(wlshcolXXXVIII); color(38bp)=(wlshcolXXXIX); color(39bp)
      =(wlshcolXL); color(40bp)=(wlshcolXLI); color(41bp)=(wlshcolXLII); color(42bp)=(
      wlshcolXLIII); color(43bp)=(wlshcolXLIV); color(44bp)=(wlshcolXLV); color(45bp)=(
      wlshcolXLVI); color(46bp)=(wlshcolXLVII); color(47bp)=(wlshcolXLVIII); color(48bp)=(
      wlshcolXLIX); color(49bp)=(wlshcolL); color(50bp)=(wlshcolLI)}}%
1143 \else\ifx\wl@test\wl@h\relax%%
1144 \pgfplotsset{/pgfplots/colormap={#2}{rgb(0bp)=(.3,0,.3); rgb(1bp)=(.3122,0,.3175); rgb
      (2bp)=(.3238,0,.335); rgb(3bp)=(.3349,0,.3525); rgb(4bp)=(.3453,0,.37); rgb(5bp)
      =(0.3552,0,.3875); rgb(6bp)=(.3645,0,.405); rgb(7bp)=(.3732,0,.4225); rgb(8bp)
      =(0.3813,0,.44); rgb(9bp)=(.3889,0,.4575); rgb(10bp)=(.3958,0,.475); rgb(11bp)
      =(0.4022,0,.4925); rgb(12bp)=(.408,0,.51); rgb(13bp)=(.4132,0,.5275); rgb(14bp)
      =(0.4178,0,.545); rgb(15bp)=(.4219,0,.5625); rgb(16bp)=(.4253,0,.58); rgb(17bp)
      =(0.4282,0,.5975); rgb(18bp)=(.4305,0,.615); rgb(19bp)=(.4322,0,.6325); rgb(20bp)
      =(0.4333,0,.65); rgb(21bp)=(.4339,0,.6675); rgb(22bp)=(.4338,0,.685); rgb(23bp)
      =(0.4332,0,.7025); rgb(24bp)=(.432,0,.72); rgb(25bp)=(.4302,0,.7375); rgb(26bp)
      =(0.4278,0,.755); rgb(27bp)=(.4249,0,.7725); rgb(28bp)=(.4213,0,.79); rgb(29bp)
      =(0.4172,0,.8075); rgb(30bp)=(.4125,0,.825); rgb(31bp)=(.4072,0,.8425); rgb(32bp)
      =(0.4013,0,.86); rgb(33bp)=(.3949,0,.8775); rgb(34bp)=(.3878,0,.895); rgb(35bp)
      =(0.3802,0,.9125); rgb(36bp)=(.372,0,.93); rgb(37bp)=(.3632,0,.9475); rgb(38bp)
      =(0.3538,0,.965); rgb(39bp)=(.3439,0,.9825); rgb(40bp)=(.3333,0,1); rgb(41bp)
      =(0.3167,0,1); rgb(42bp)=(.3,0,1); rgb(43bp)=(.2833,0,1); rgb(44bp)=(.2667,0,1); rgb
      (45bp)=(.25,0,1); rgb(46bp)=(.2333,0,1); rgb(47bp)=(.2167,0,1); rgb(48bp)=(.2,0,1);
      rgb(49bp)=(.1833,0,1); rgb(50bp)=(.1667,0,1); %
1145 rgb(51bp)=(.15,0,1); rgb(52bp)=(.1333,0,1); rgb(53bp)=(.1167,0,1); rgb(54bp)=(.1,0,1);
      rgb(55bp)=(.0833,0,1); rgb(56bp)=(.0667,0,1); rgb(57bp)=(.05,0,1); rgb(58bp)
      =(0.0333,0,1); rgb(59bp)=(.0167,0,1); rgb(60bp)=(0,0,1); rgb(61bp)=(0,.02,1); rgb(62bp)
      =(0,.04,1); rgb(63bp)=(0,.06,1); rgb(64bp)=(0,.08,1); rgb(65bp)=(0,.1,1); rgb(66bp)
      =(0,.12,1); rgb(67bp)=(0,.14,1); rgb(68bp)=(0,.16,1); rgb(69bp)=(0,.18,1); rgb(70bp)
      =(0,.2,1); rgb(71bp)=(0,.22,1); rgb(72bp)=(0,.24,1); rgb(73bp)=(0,.26,1); rgb(74bp)
      =(0,.28,1); rgb(75bp)=(0,.3,1); rgb(76bp)=(0,.32,1); rgb(77bp)=(0,.34,1); rgb(78bp)
      =(0,.36,1); rgb(79bp)=(0,.38,1); rgb(80bp)=(0,.4,1); rgb(81bp)=(0,.42,1); rgb(82bp)
      =(0,.44,1); rgb(83bp)=(0,.46,1); rgb(84bp)=(0,.48,1); rgb(85bp)=(0,.5,1); rgb(86bp)
      =(0,.52,1); rgb(87bp)=(0,.54,1); rgb(88bp)=(0,.56,1); rgb(89bp)=(0,.58,1); rgb(90bp)
      =(0,.6,1); rgb(91bp)=(0,.62,1); rgb(92bp)=(0,.64,1); rgb(93bp)=(0,.66,1); rgb(94bp)
      =(0,.68,1); rgb(95bp)=(0,.7,1); rgb(96bp)=(0,.72,1); rgb(97bp)=(0,.74,1); rgb(98bp)
      =(0,.76,1); rgb(99bp)=(0,.78,1); rgb(100bp)=(0,.8,1); %
1146 rgb(101bp)=(0,.82,1); rgb(102bp)=(0,.84,1); rgb(103bp)=(0,.86,1); rgb(104bp)=(0,.88,1);
      rgb(105bp)=(0,.9,1); rgb(106bp)=(0,.92,1); rgb(107bp)=(0,.94,1); rgb(108bp)
      =(0,.96,1); rgb(109bp)=(0,.98,1); rgb(110bp)=(0,1,1); rgb(111bp)=(0,1,.95); rgb(112bp)
      =(0,1,.9); rgb(113bp)=(0,1,.85); rgb(114bp)=(0,1,.8); rgb(115bp)=(0,1,.75); rgb(116
      bp)=(0,1,.7); rgb(117bp)=(0,1,.65); rgb(118bp)=(0,1,.6); rgb(119bp)=(0,1,.55); rgb
      (120bp)=(0,1,.5); rgb(121bp)=(0,1,.45); rgb(122bp)=(0,1,.4); rgb(123bp)=(0,1,.35);
      rgb(124bp)=(0,1,.3); rgb(125bp)=(0,1,.25); rgb(126bp)=(0,1,.2); rgb(127bp)=(0,1,.15);
      rgb(128bp)=(0,1,.1); rgb(129bp)=(0,1,.05); rgb(130bp)=(0,1,0); rgb(131bp)
      =(0.0143,1,0); rgb(132bp)=(.0286,1,0); rgb(133bp)=(.0429,1,0); rgb(134bp)=(.0571,1,0);
      rgb(135bp)=(.0714,1,0); rgb(136bp)=(.0857,1,0); rgb(137bp)=(.1,1,0); rgb(138bp)
      =(0.1143,1,0); rgb(139bp)=(.1286,1,0); rgb(140bp)=(.1429,1,0); rgb(141bp)=(.1571,1,0);
      rgb(142bp)=(.1714,1,0); rgb(143bp)=(.1857,1,0); rgb(144bp)=(.2,1,0); rgb(145bp)

```

```

    =(.2143,1,0); rgb(146bp)=(.2286,1,0); rgb(147bp)=(.2429,1,0); rgb(148bp)=(.2571,1,0);
    rgb(149bp)=(.2714,1,0); rgb(150bp)=(.2857,1,0); %
1147 rgb(151bp)=(.3,1,0); rgb(152bp)=(.3143,1,0); rgb(153bp)=(.3286,1,0); rgb(154bp)
    =(.3429,1,0); rgb(155bp)=(.3571,1,0); rgb(156bp)=(.3714,1,0); rgb(157bp)=(.3857,1,0);
    rgb(158bp)=(.4,1,0); rgb(159bp)=(.4143,1,0); rgb(160bp)=(.4286,1,0); rgb(161bp)
    =(.4429,1,0); rgb(162bp)=(.4571,1,0); rgb(163bp)=(.4714,1,0); rgb(164bp)=(.4857,1,0);
    rgb(165bp)=(.5,1,0); rgb(166bp)=(.5143,1,0); rgb(167bp)=(.5286,1,0); rgb(168bp)
    =(.5429,1,0); rgb(169bp)=(.5571,1,0); rgb(170bp)=(.5714,1,0); rgb(171bp)=(.5857,1,0);
    rgb(172bp)=(.6,1,0); rgb(173bp)=(.6143,1,0); rgb(174bp)=(.6286,1,0); rgb(175bp)
    =(.6429,1,0); rgb(176bp)=(.6571,1,0); rgb(177bp)=(.6714,1,0); rgb(178bp)=(.6857,1,0);
    rgb(179bp)=(.7,1,0); rgb(180bp)=(.7143,1,0); rgb(181bp)=(.7286,1,0); rgb(182bp)
    =(.7429,1,0); rgb(183bp)=(.7571,1,0); rgb(184bp)=(.7714,1,0); rgb(185bp)=(.7857,1,0);
    rgb(186bp)=(.8,1,0); rgb(187bp)=(.8143,1,0); rgb(188bp)=(.8286,1,0); rgb(189bp)
    =(.8429,1,0); rgb(190bp)=(.8571,1,0); rgb(191bp)=(.8714,1,0); rgb(192bp)=(.8857,1,0);
    rgb(193bp)=(.9,1,0); rgb(194bp)=(.9143,1,0); rgb(195bp)=(.9286,1,0); rgb(196bp)
    =(.9429,1,0); rgb(197bp)=(.9571,1,0); rgb(198bp)=(.9714,1,0); rgb(199bp)=(.9857,1,0);
    rgb(200bp)=(1,1,0); %
1148 rgb(201bp)=(1,.9846,0); rgb(202bp)=(1,.9692,0); rgb(203bp)=(1,.9538,0); rgb(204bp)
    =(1,.9385,0); rgb(205bp)=(1,.9231,0); rgb(206bp)=(1,.9077,0); rgb(207bp)=(1,.8923,0);
    rgb(208bp)=(1,.8769,0); rgb(209bp)=(1,.8615,0); rgb(210bp)=(1,.8462,0); rgb(211bp)
    =(1,.8308,0); rgb(212bp)=(1,.8154,0); rgb(213bp)=(1,.8,0); rgb(214bp)=(1,.7846,0);
    rgb(215bp)=(1,.7692,0); rgb(216bp)=(1,.7538,0); rgb(217bp)=(1,.7385,0); rgb(218bp)
    =(1,.7231,0); rgb(219bp)=(1,.7077,0); rgb(220bp)=(1,.6923,0); rgb(221bp)=(1,.6769,0);
    rgb(222bp)=(1,.6615,0); rgb(223bp)=(1,.6462,0); rgb(224bp)=(1,.6308,0); rgb(225bp)
    =(1,.6154,0); rgb(226bp)=(1,.6,0); rgb(227bp)=(1,.5846,0); rgb(228bp)=(1,.5692,0);
    rgb(229bp)=(1,.5538,0); rgb(230bp)=(1,.5385,0); rgb(231bp)=(1,.5231,0); rgb(232bp)
    =(1,.5077,0); rgb(233bp)=(1,.4923,0); rgb(234bp)=(1,.4769,0); rgb(235bp)=(1,.4615,0);
    rgb(236bp)=(1,.4462,0); rgb(237bp)=(1,.4308,0); rgb(238bp)=(1,.4154,0); rgb(239bp)
    =(1,.4,0); rgb(240bp)=(1,.3846,0); rgb(241bp)=(1,.3692,0); rgb(242bp)=(1,.3538,0);
    rgb(243bp)=(1,.3385,0); rgb(244bp)=(1,.3231,0); rgb(245bp)=(1,.3077,0); rgb(246bp)
    =(1,.2923,0); rgb(247bp)=(1,.2769,0); rgb(248bp)=(1,.2615,0); rgb(249bp)=(1,.2462,0);
    rgb(250bp)=(1,.2308,0); %
1149 rgb(251bp)=(1,.2154,0); rgb(252bp)=(1,.2,0); rgb(253bp)=(1,.1846,0); rgb(254bp)
    =(1,.1692,0); rgb(255bp)=(1,.1538,0); rgb(256bp)=(1,.1385,0); rgb(257bp)=(1,.1231,0);
    rgb(258bp)=(1,.1077,0); rgb(259bp)=(1,.0923,0); rgb(260bp)=(1,.0769,0); rgb(261bp)
    =(1,.0615,0); rgb(262bp)=(1,.0462,0); rgb(263bp)=(1,.0308,0); rgb(264bp)=(1,.0154,0);
    rgb(265bp)=(1,0,0); rgb(266bp)=(1,0,0); rgb(267bp)=(1,0,0); rgb(268bp)=(1,0,0); rgb
    (269bp)=(1,0,0); rgb(270bp)=(1,0,0); rgb(271bp)=(1,0,0); rgb(272bp)=(1,0,0); rgb(273
    bp)=(1,0,0); rgb(274bp)=(1,0,0); rgb(275bp)=(1,0,0); rgb(276bp)=(1,0,0); rgb(277bp)
    =(1,0,0); rgb(278bp)=(1,0,0); rgb(279bp)=(1,0,0); rgb(280bp)=(1,0,0); rgb(281bp)
    =(1,0,0); rgb(282bp)=(1,0,0); rgb(283bp)=(1,0,0); rgb(284bp)=(1,0,0); rgb(285bp)
    =(1,0,0); rgb(286bp)=(1,0,0); rgb(287bp)=(1,0,0); rgb(288bp)=(1,0,0); rgb(289bp)
    =(1,0,0); rgb(290bp)=(1,0,0); rgb(291bp)=(1,0,0); rgb(292bp)=(1,0,0); rgb(293bp)
    =(1,0,0); rgb(294bp)=(1,0,0); rgb(295bp)=(1,0,0); rgb(296bp)=(1,0,0); rgb(297bp)
    =(1,0,0); rgb(298bp)=(1,0,0); rgb(299bp)=(1,0,0); rgb(300bp)=(1,0,0); %
1150 rgb(301bp)=(1,0,0); rgb(302bp)=(1,0,0); rgb(303bp)=(1,0,0); rgb(304bp)=(1,0,0); rgb(305
    bp)=(1,0,0); rgb(306bp)=(1,0,0); rgb(307bp)=(1,0,0); rgb(308bp)=(1,0,0); rgb(309bp)
    =(1,0,0); rgb(310bp)=(1,0,0); rgb(311bp)=(1,0,0); rgb(312bp)=(1,0,0); rgb(313bp)
    =(1,0,0); rgb(314bp)=(1,0,0); rgb(315bp)=(1,0,0); rgb(316bp)=(1,0,0); rgb(317bp)
    =(1,0,0); rgb(318bp)=(1,0,0); rgb(319bp)=(1,0,0); rgb(320bp)=(1,0,0); rgb(321bp)
    =(.9913,0,0); rgb(322bp)=(.9825,0,0); rgb(323bp)=(.9738,0,0); rgb(324bp)=(.965,0,0);
    rgb(325bp)=(.9563,0,0); rgb(326bp)=(.9475,0,0); rgb(327bp)=(.9388,0,0); rgb(328bp)
    =(.93,0,0); rgb(329bp)=(.9213,0,0); rgb(330bp)=(.9125,0,0); rgb(331bp)=(.9038,0,0);
    rgb(332bp)=(.895,0,0); rgb(333bp)=(.8863,0,0); rgb(334bp)=(.8775,0,0); rgb(335bp)
    =(.8688,0,0); rgb(336bp)=(.86,0,0); rgb(337bp)=(.8513,0,0); rgb(338bp)=(.8425,0,0);
    rgb(339bp)=(.8338,0,0); rgb(340bp)=(.825,0,0); rgb(341bp)=(.8163,0,0); rgb(342bp)
    =(.8075,0,0); rgb(343bp)=(.7988,0,0); rgb(344bp)=(.79,0,0); rgb(345bp)=(.7813,0,0);
    rgb(346bp)=(.7725,0,0); rgb(347bp)=(.7638,0,0); rgb(348bp)=(.755,0,0); rgb(349bp)
    =(.7463,0,0); rgb(350bp)=(.7375,0,0); %
1151 rgb(351bp)=(.7288,0,0); rgb(352bp)=(.72,0,0); rgb(353bp)=(.7113,0,0); rgb(354bp)
    =(.7025,0,0); rgb(355bp)=(.6938,0,0); rgb(356bp)=(.685,0,0); rgb(357bp)=(.6763,0,0);
    rgb(358bp)=(.6675,0,0); rgb(359bp)=(.6588,0,0); rgb(360bp)=(.65,0,0); rgb(361bp)
    =(.6413,0,0); rgb(362bp)=(.6325,0,0); rgb(363bp)=(.6238,0,0); rgb(364bp)=(.615,0,0);
    rgb(365bp)=(.6063,0,0); rgb(366bp)=(.5975,0,0); rgb(367bp)=(.5888,0,0); rgb(368bp)
    =(.58,0,0); rgb(369bp)=(.5713,0,0); rgb(370bp)=(.5625,0,0); rgb(371bp)=(.5538,0,0);
    rgb(372bp)=(.545,0,0); rgb(373bp)=(.5363,0,0); rgb(374bp)=(.5275,0,0); rgb(375bp)
    =(.5188,0,0); rgb(376bp)=(.51,0,0); rgb(377bp)=(.5013,0,0); rgb(378bp)=(.4925,0,0);
    rgb(379bp)=(.4838,0,0); rgb(380bp)=(.475,0,0); rgb(381bp)=(.4663,0,0); rgb(382bp)
    =(.4575,0,0); rgb(383bp)=(.4488,0,0); rgb(384bp)=(.44,0,0); rgb(385bp)=(.4313,0,0);
    rgb(386bp)=(.4225,0,0); rgb(387bp)=(.4138,0,0); rgb(388bp)=(.405,0,0); rgb(389bp)

```



```

= (.3963,0,0); rgb(390bp)=(.3875,0,0); rgb(391bp)=(.3788,0,0); rgb(392bp)=(.37,0,0);
rgb(393bp)=(.3613,0,0); rgb(394bp)=(.3525,0,0); rgb(395bp)=(.3438,0,0); rgb(396bp)
= (.335,0,0); rgb(397bp)=(.3263,0,0); rgb(398bp)=(.3175,0,0); rgb(399bp)=(.3088,0,0);
rgb(400bp)=(.3,0,0); %
1152 }}%
1153 \fi\fi%
1154 }%
1155 %XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1156 % |pgfspectrarainbow[<tikz options>](<rainbow fade>,<rainbow start>,<rainbow knock out
    >,<rainbow background>,<rainbow transparency>){radius}
1157 % tikz options -> color, opacity, scope fading
1158 % rainbow clip -> applies a scope fading in clipped region
1159 % ...
1160 \pgfkeys{/wl/.cd,%
1161 rainbow fade/.get=\wl@rainbowfade,%
1162 rainbow fade/.store in=\wl@rainbowfade,%
1163 rainbow fade/.default={},%
1164 rainbow start/.get=\wl@rainbow@start,%
1165 rainbow start/.store in=\wl@rainbow@start,%
1166 rainbow start/.default=.6,% -> 60%
1167 rainbow knock out/.get=\wl@rainbow@KO,%
1168 rainbow knock out/.store in=\wl@rainbow@KO,%
1169 rainbow knock out/.default=.4,% -> 40%
1170 rainbow background/.get=\wl@rainbowback,%
1171 rainbow background/.store in=\wl@rainbowback,%
1172 rainbow background/.default=white,%
1173 rainbow transparency/.get=\wl@rainbowtransp,%
1174 rainbow transparency/.store in=\wl@rainbowtransp,%
1175 rainbow transparency/.default=0}% -> 0%
1176 %
1177 \def\pgfspectrarainbow{\ignorespaces\@ifnextchar[\pgf@spectrarainbow{\
    pgf@spectrarainbow[]}}%
1178 \def\pgf@spectrarainbow[#1]{\ignorespaces\@ifnextchar({\pgf@spectra@rainbow{#1}}{\
    pgf@spectra@rainbow{#1}())}%
1179 %
1180 \def\pgf@spectra@rainbow#1(#2)#3{\ignorespaces%
1181 \pgfkeys{/wl/.cd,rainbow fade,rainbow start,rainbow knock out,rainbow background,
    rainbow transparency}%
1182 \pgfkeys{/wl/.cd,#2}%
1183 \pgfmathparse{100-\wl@rainbowtransp*100}\edef\wl@rainbow@transp{\pgfmathresult}%
1184 \pgfmathparse{\wl@rainbowtransp*100}\edef\wl@rainbow@transp@w{\pgfmathresult}%
1185 \edef\wl@rainbowend{.8875cm}\pgfmathparse{\wl@rainbow@start*\wl@rainbowend/1cm}\edef\
    wl@rainbowstart{\pgfmathresult cm}%
1186 \pgfmathparse{\wl@rainbow@KO*#3/1cm}\edef\wl@rainbowKO{\pgfmathresult cm}%%\edef\
    wl@rainbowKO{\wl@rainbow@KO cm}%
1187 \ifdim\wl@rainbowstart<\wl@rainbowend\relax%
1188 \ifdim\wl@rainbowstart<.0175cm\relax\edef\wl@rainbowstart{.0175cm}\fi% ensuring there
    is no error in radial shading
1189 \pgfkeys{/pgf/number format/.cd,fixed,precision=3,set thousands separator={},assume
    math mode=true}%
1190 \pgfmathparse{\wl@rainbowstart-1/50*(\wl@rainbowend-\wl@rainbowstart)}%
1191 \pgfmathprintnumberto{\pgfmathresult}{\wl@rainbowresult}%
1192 \edef\r0{\wl@rainbowresult pt}%
1193 \@for\n
    :={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36}
    \do{%
1194 \pgfmathparse{380+(\n-1)*8}\edef\wl@currentwl{\pgfmathresult}%
1195 \wlcolor{\wl@currentwl}%
1196 \edef\wl@colorname{\wlshcol\@Roman\n}\relax\colorlet{\wl@colorname}{\wlcolor!100!
    transparent!\wl@rainbow@transp}%
1197 \pgfmathparse{\wl@rainbowstart+1/50*(\n-1)*(\wl@rainbowend-\wl@rainbowstart)}%
1198 \pgfmathprintnumberto{\pgfmathresult}{\wl@rainbowresult}%
1199 \expandafter\edef\csname r\@Roman\n\endcsname{\wl@rainbowresult pt}%
1200 }%
1201 \pgfdeclareradialshading[pgfspectrarainbow]{\pgfpoint{0pt}{0pt}}{\pgfpoint{0pt}{0pt}}%
1202 color(0cm)=(\wl@rainbowback!\wl@rainbow@transp@w!white); color(4/5*\r0)=(\wl@rainbowback
    !\wl@rainbow@transp@w!white!50); color(\r0)=(white);%
1203 color(\rI)=(\wlshcolI); color(\rII)=(\wlshcolII); color(\rIII)=(\wlshcolIII); color(\rIV)=(
    \wlshcolIV); color(\rV)=(\wlshcolV); color(\rVI)=(\wlshcolVI); color(\rVII)=(\wlshcolVII);
    color(\rVIII)=(\wlshcolVIII); color(\rIX)=(\wlshcolIX); color(\rX)=(\wlshcolX);%

```

```

1204 color(\rXI)=(wlshcolXI);color(\rXII)=(wlshcolXII);color(\rXIII)=(wlshcolXIII);color(\
rXIV)=(wlshcolXIV);color(\rXV)=(wlshcolXV);color(\rXVI)=(wlshcolXVI);color(\rXVII)=(
wlshcolXVII);color(\rXVIII)=(wlshcolXVIII);color(\rXIX)=(wlshcolXIX);color(\rXX)=(
wlshcolXX);%
1205 color(\rXXI)=(wlshcolXXI);color(\rXXII)=(wlshcolXXII);color(\rXXIII)=(wlshcolXXIII);
color(\rXXIV)=(wlshcolXXIV);color(\rXXV)=(wlshcolXXV);color(\rXXVI)=(wlshcolXXVI);
color(\rXXVII)=(wlshcolXXVII);color(\rXXVIII)=(wlshcolXXVIII);color(\rXXIX)=(
wlshcolXXIX);color(\rXXX)=(wlshcolXXX);%
1206 color(\rXXXI)=(wlshcolXXXI);color(\rXXXII)=(wlshcolXXXII);color(\rXXXIII)=(
wlshcolXXXIII);color(\rXXXIV)=(wlshcolXXXIV);color(\rXXXV)=(wlshcolXXXV);color(\
rXXXVI)=(wlshcolXXXVI);color(\rXXXVII)=(wlshcolXXXVII);color(\rXXXVIII)=(
wlshcolXXXVIII);color(\rXXXIX)=(wlshcolXXXIX);color(\rXL)=(wlshcolXL);%
1207 color(\rXLI)=(wlshcolXLI);color(\rXLII)=(wlshcolXLII);color(\rXLIII)=(wlshcolXLIII);
color(\rXLIV)=(wlshcolXLIV);color(\rXLV)=(wlshcolXLV);color(\rXLVI)=(wlshcolXLVI);
color(\rXLVII)=(wlshcolXLVII);color(\rXLVIII)=(wlshcolXLVIII);color(\rXLIX)=(
wlshcolXLIX);color(\rL)=(wlshcolL);color(\rLI)=(wlshcolLI);%
1208 color(.95cm)=(wlshcolLI)%
1209 }%
1210 \ifx\wl@rainbowfade\@empty\relax%
1211 \tikz{\clip(-#3,\wl@rainbowK0) rectangle ++(2*#3,#3-\wl@rainbowK0);%
1212 \fill[#1,shading=pgfspectrarrainbow] (0,0) circle(#3);}%
1213 \else%
1214 \tikz{\clip[scope fading=\wl@rainbowfade] (-#3,\wl@rainbowK0) rectangle ++(2*#3,#3-\
wl@rainbowK0);%
1215 \fill[shading=pgfspectrarrainbow,\wl@rainbowback,#1] (0,0) circle(#3);}%
1216 \fi%
1217 \else\PackageError{pgf-spectra}{invalid 'rainbow start' value (rainbow start=\
wl@rainbow@start). The rainbow start should be greater or equal then 0 and lower then
1.}{Don't forget that 'rainbow start' value is the fraction from witch the colors
begin, relative to the center of a circle with radius 1...}%
1218 \fi
1219 }%
1220 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1221 % |tempercolor -> Convert a temperature in Kelvin to rgb color
1222 %
1223 % The original algorithm used in this macro can be found at
1224 %
1225 % https://github.com/neilbartlett/color-temperature
1226 %
1227 % More information at...
1228 %
1229 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1230 %
-----
1231 % https://tannerhelland.com/2012/09/18/convert-temperature-rgb-algorithm-code.html
1232 %
1233 % «Start with a temperature, in Kelvin, somewhere between 1000 and 40000. (Other
values may work,
1234 % but I can't make any promises about the quality of the algorithm's estimates above
40000 K.)»
1235 %
Tanner Helland
1236 % https://www.zombieprototypes.com/?p=210
1237 %
-----
1238 % and the improved algorithm at...
1239 %
-----
1240 % https://github.com/neilbartlett/color-temperature
1241 %
1242 % neilbartlett/color-temperature is licensed under the
1243 %
1244 % MIT License
1245 % A short and simple permissive license with conditions only requiring preservation of
copyright and license notices.

```

```

1246 % Licensed works, modifications, and larger works may be distributed under different
1247 % terms and without source code.
1248 %
1249 % Permissions
1250 % Commercial use, Modification, Distribution, Private use
1251 % Limitations
1252 % Liability, Warranty
1253 %
1254 % The MIT License (MIT)
1255 %
1256 % Copyright (c) 2015 Neil Bartlett
1257 %
1258 % Permission is hereby granted, free of charge, to any person obtaining a copy
1259 % of this software and associated documentation files (the "Software"), to deal
1260 % in the Software without restriction, including without limitation the rights
1261 % to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
1262 % copies of the Software, and to permit persons to whom the Software is
1263 % furnished to do so, subject to the following conditions:
1264 %
1265 % The above copyright notice and this permission notice shall be included in
1266 % all copies or substantial portions of the Software.
1267 %
1268 % THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
1269 % IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
1270 % FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
1271 % AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
1272 % LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
1273 % OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
1274 % THE SOFTWARE.
1275 % © 2021 GitHub, Inc.
1276 %
1277 %
1278 \definecolor{tempercolor}{rgb}{0,0,0}%
1279 %
1280 % |tempercolor{Kelvin}
1281 \def\tempercolor#1{\ignorespaces%
1282 %Set Temperature = Temperature | 100
1283 \edef\wl@temperatura{#1sp}%
1284 \ifdim\wl@temperatura>16380sp\relax% for supporting large values of T
1285 \pgfkeys{/pgf/number format/.cd,fixed,precision=2,fixed zerofill,set thousands
1286 separator={},assume math mode=true}%
1287 \pgfmathparse{#1sp/100sp*1.0026931+.002}% correction factor for the division made in sp
1288 \pgfmathprintnumberto{\pgfmathresult}{\wl@T@tmp}%
1289 \edef\wl@temperatura{\wl@T@tmp}%
1290 \edef\wl@temperatura@pt{\pgfmathresult pt}%
1291 \else%
1292 \pgfmathparse{#1/100}%
1293 \edef\wl@temperatura{\pgfmathresult}%
1294 \edef\wl@temperatura@pt{\pgfmathresult pt}%
1295 \fi%
1296 \ifdim\wl@temperatura@pt<10pt\relax%
1297 \PackageError{pgf-spectra}{tempercolor: the temperature must be at least 1000K}{Type a
1298 greater temperature...}%
1299 \else\ifdim\wl@temperatura@pt>400pt\relax%
1300 \PackageError{pgf-spectra}{tempercolor: the temperature must be at most 40000K}{Type a
1301 smaller temperature...}%
1302 \else%
1303 %Calculate Red:
1304 \ifdim\wl@temperatura@pt>66pt\relax%
1305 \pgfmathparse{\wl@temperatura-55}\edef\wl@tempercolor@r{\pgfmathresult}%
1306 \pgfmathparse{351.976906+ 0.114207*\wl@tempercolor@r-40.253663*ln(\wl@tempercolor@r
1307 )}%
1308 \edef\wl@tempercolor@r{\pgfmathresult}%
1309 \edef\wl@tempercolor@r@pt{\pgfmathresult pt}%
1310 \ifdim\wl@tempercolor@r@pt<0pt\edef\wl@tempercolor@r{0}\relax\fi%
1311 \ifdim\wl@tempercolor@r@pt>255pt\edef\wl@tempercolor@r{255}\relax\fi%
1312 \pgfmathparse{\wl@tempercolor@r/255}\edef\wl@tempercolor@r{\pgfmathresult}%
1313 \else%
1314 \edef\wl@tempercolor@r{1}%

```

```

1311 \fi%
1312 % Calculate Green:
1313 \ifdim\wl@temperatura@pt>66pt\relax%
1314 \pgfmathparse{\wl@temperatura-50}\edef\wl@tempercolor@g{\pgfmathresult}%
1315 \pgfmathparse{325.449413+0.079435*\wl@tempercolor@g-28.085296*ln(\wl@tempercolor@g
1316 )}%
1317 \edef\wl@tempercolor@g{\pgfmathresult}%
1318 \edef\wl@tempercolor@g@pt{\pgfmathresult pt}%
1319 \ifdim\wl@tempercolor@g@pt<0pt\edef\wl@tempercolor@g{0}\relax\fi%
1320 \ifdim\wl@tempercolor@g@pt>255pt\edef\wl@tempercolor@g{255}\relax\fi%
1321 \pgfmathparse{\wl@tempercolor@g/255}\edef\wl@tempercolor@g{\pgfmathresult}%
1322 \else%
1323 \pgfmathparse{\wl@temperatura-2}\edef\wl@tempercolor@g{\pgfmathresult}%
1324 \pgfmathparse{-155.254856-0.445970*\wl@tempercolor@g+104.492162*ln(\
1325 \wl@tempercolor@g)}%
1326 \edef\wl@tempercolor@g{\pgfmathresult}%
1327 \edef\wl@tempercolor@g@pt{\pgfmathresult pt}%
1328 \ifdim\wl@tempercolor@g@pt<0pt\edef\wl@tempercolor@g{0}\relax\fi%
1329 \ifdim\wl@tempercolor@g@pt>255pt\edef\wl@tempercolor@g{255}\relax\fi%
1330 \pgfmathparse{\wl@tempercolor@g/255}\edef\wl@tempercolor@g{\pgfmathresult}%
1331 \fi%
1332 % Calculate Blue:
1333 \ifdim\wl@temperatura@pt<66pt\relax%
1334 \ifdim\wl@temperatura@pt>20pt\relax%
1335 \pgfmathparse{\wl@temperatura-10}\edef\wl@tempercolor@b{\pgfmathresult}%
1336 \pgfmathparse{-254.769352+0.827410*\wl@tempercolor@b+115.679944*ln(\
1337 \wl@tempercolor@b)}%
1338 \edef\wl@tempercolor@b{\pgfmathresult}%
1339 \edef\wl@tempercolor@b@pt{\pgfmathresult pt}%
1340 \ifdim\wl@tempercolor@b@pt<0pt\edef\wl@tempercolor@b{0}\relax\fi%
1341 \ifdim\wl@tempercolor@b@pt>255pt\edef\wl@tempercolor@b{255}\relax\fi%
1342 \pgfmathparse{\wl@tempercolor@b/255}\edef\wl@tempercolor@b{\pgfmathresult}%
1343 \else%
1344 \edef\wl@tempercolor@b{0}%
1345 \fi%
1346 \else%
1347 \edef\wl@tempercolor@b{1}%
1348 \fi%
1349 \definecolor{wl@tempercolor}{rgb}{\wl@tempercolor@r,\wl@tempercolor@g,\
1350 \wl@tempercolor@b}%
1351 \colorlet{tempercolor}{wl@tempercolor}%
1352 \fi\fi% check limits 1000K<T<40000K
1353 }%
1354 % <--- NEW v2.1.0 -----
1355 % nm2rgb convert nanometre wavelength to rgb
1356 % (380 <= Lambda <= 780 ) -> r,g,b on stack
1357 %
1358 % BASED on FORTRAN Code
1359 % RGB VALUES FOR VISIBLE WAVELENGTHS by Dan Bruton (astro@tamu.edu)
1360 % This program can be found at
1361 % http://www.physics.sfasu.edu/astro/color.html
1362 % and was last updated on February 20, 1996.
1363 % The spectrum is generated using approximate RGB values for visible
1364 % wavelengths between 380 nm and 780 nm.
1365 % The red, green and blue values (RGB) are
1366 % assumed to vary linearly with wavelength (for GAMMA=1).
1367 %
1368 \newdimen\wl#wavelength
1369 \newdimen\wl@i%intensity
1370 \newdimen\wl@gamma%gamma
1371 \newdimen\wlc@lorr%red (0. - 1)
1372 \newdimen\wlc@lorg%green (0. - 1)
1373 \newdimen\wlc@lorb%blue (0. - 1) % wavelength to rgb values
1374 %
1375 %
1376 % |wlcolor{wavelength}

```

```

1377 \def\wlc@color#1{\ignorespaces%
1378 \ifx\wlc@bright\undefined\relax\def\wlc@bright{100}\fi% NEW v2.1.0
1379 \wlc=#1pt%
1380 \wlc@gamma=\wlc@gamma pt%
1381 % compute the rgb components
1382 \ifdim\wlc<10pt\relax\PackageWarning{pgf-spectra}{(#1nm) wavelength out of range ignored
. The wavelength must be greater or equal to 10nm (EUV)...}\else% NEW v2.0.0
1383 \ifdim\wlc<379.99999pt\relax\else%% NEW v2.0.0
1384 \ifdim\wlc<440pt\wlc@lorr=440pt\advance\wlc@lorr by-\wlc\divide\wlc@lorr by60\wlc@lorg=0
pt\wlc@lorb=1pt\else%
1385 \ifdim\wlc<490pt\wlc@lorr=0pt\wlc@lorg=\wlc\advance\wlc@lorg by-440pt\divide\wlc@lorg by
50\wlc@lorb=1pt\else%
1386 \ifdim\wlc<510pt\wlc@lorr=0pt\wlc@lorg=1pt\wlc@lorb=510pt\advance\wlc@lorb by-\wlc\divide
\wlc@lorb by20\else%
1387 \ifdim\wlc<580pt\wlc@lorr=\wlc\advance\wlc@lorr by-510pt\divide\wlc@lorr by70\wlc@lorg=1
pt\wlc@lorb=0pt\else%
1388 \ifdim\wlc<645pt\wlc@lorr=1pt\wlc@lorg=645pt\advance\wlc@lorg by-\wlc\divide\wlc@lorg by
65\wlc@lorb=0pt\else%
1389 \ifdim\wlc<780.00001pt\wlc@lorr=1pt\wlc@lorg=0pt\wlc@lorb=0pt\else%
1390 \ifdim\wlc>4000pt\relax\PackageWarning{pgf-spectra}{invalid wavelength (#1nm). The
wavelength must be lesser or equal to 4000nm (NIR)...}% NEW v2.0.0
1391 \relax%
1392 \fi\fi\fi\fi\fi\fi\fi\fi\fi%
1393 % intensity correction at vision limits
1394 \ifdim\wlc>700pt\ifdim\wlc<780.00001pt\wlc@i=780pt\advance\wlc@i by-\wlc\divide\wlc@i by80\
multiply\wlc@i by7\advance\wlc@i by3pt\divide\wlc@i by10\fi\else% NEW
1395 \ifdim\wlc<420pt\ifdim\wlc>379.99999pt\wlc@i=\wlc\advance\wlc@i by-380pt\divide\wlc@i by40\
multiply\wlc@i by7\advance\wlc@i by3pt\divide\wlc@i by10\fi\else%
1396 \wlc@i=1pt%
1397 \fi\fi%
1398 % apply intensity at vision limits correction and gamma
1399 \ifdim\wlc<380pt\colorlet\wlc@temp{\wlc@UVcolor}\else\ifdim\wlc>780pt\colorlet\wlc@temp{\
wlc@IRcolor}\else% NEW v2.0.0
1400 \pgfmathparse{\wlc@lorr*\wlc@i~\wlc@gamma}\edef\wlc@red{\pgfmathresult}%
1401 \pgfmathparse{\wlc@lorg*\wlc@i~\wlc@gamma}\edef\wlc@green{\pgfmathresult}%
1402 \pgfmathparse{\wlc@lorb*\wlc@i~\wlc@gamma}\edef\wlc@blue{\pgfmathresult}%
1403 \definecolor\wlc@temp{rgb}{\wlc@red,\wlc@green,\wlc@blue}%
1404 \fi\fi% NEW v2.0.0
1405 \ifwlc@usevisiblesshade\relax% NEW v2.1.0
1406 \colorlet\wlc@color{\wlc@temp!\wlc@bright!\wlc@backvisible}%
1407 \else%
1408 \colorlet\wlc@color{\wlc@temp}%
1409 \fi%
1410 }%
1411 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1412 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1413 \def\wlc@elt@search#1#2#3#4{\ignorespaces%
1414 % #1 Chemical Symbol, entered by USER
1415 % #2 Chemical Symbol to compare to, e.g. Na
1416 % #3 Emission Lines Data (or error message)
1417 % #4 Imax
1418 \edef\wlc@CS@user{#1}\edef\wlc@CS@comp{#2}\relax% New
1419 \ifx\wlc@CS@user\wlc@CS@comp\relax% New -> was \iftthenelse{...
1420 \def\wlc@elt@chemsym{#2}% set chemical symbol
1421 \def\wlc@elt@elemdata{#3}% set element lines data
1422 \def\wlc@elt@Imax{#4}% set element Imax
1423 \fi%
1424 }%
1425 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1426 \endinput

```