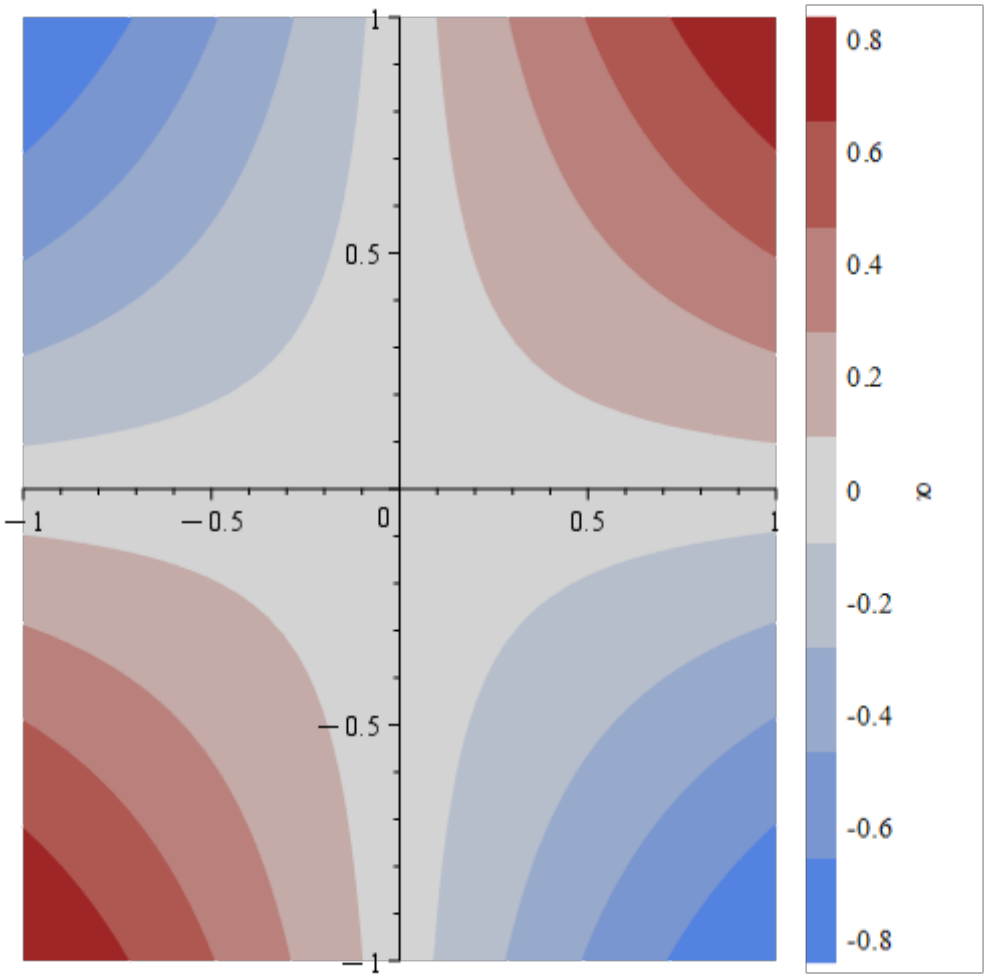
Visualization Updates in Maple 2024

Expanded support for color bars

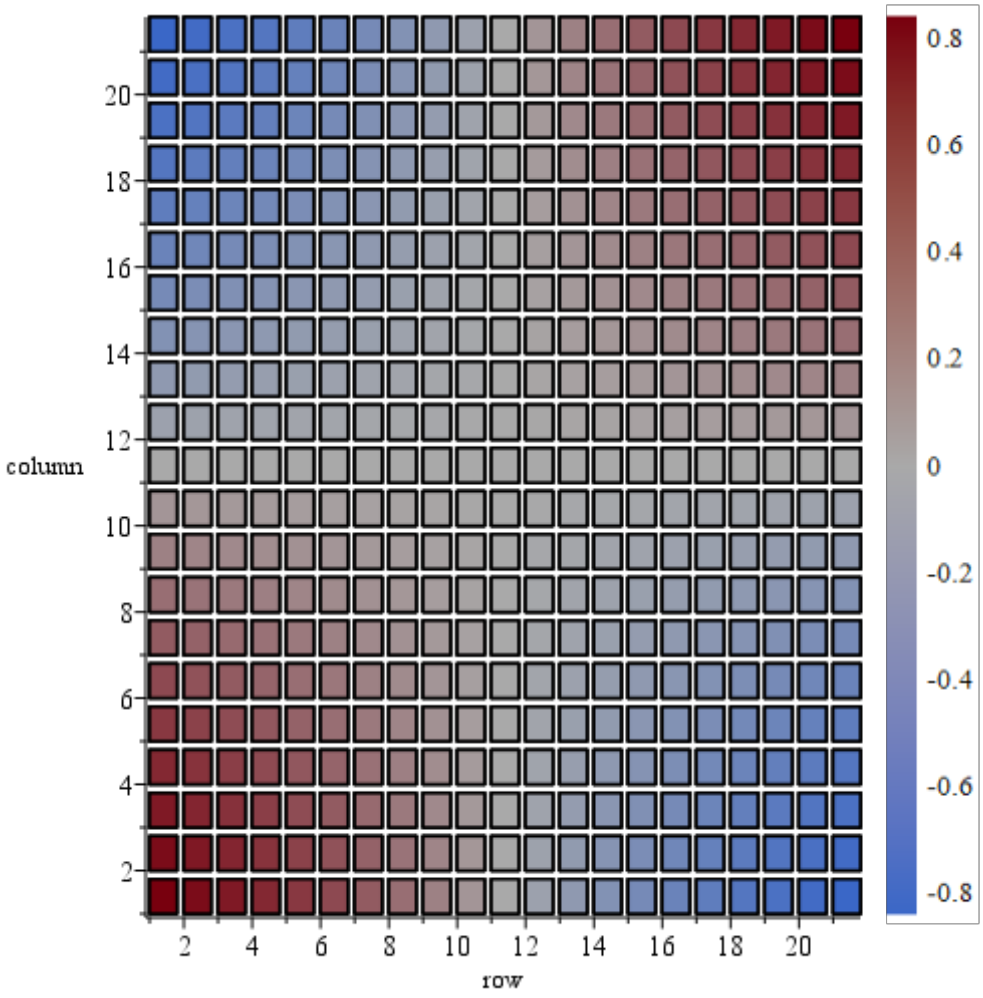
- The color bars added to plots:-densityplot and plots:-contourplot in Maple 2023 now have many new options to customize their display including changing the labels and fonts. A full list of new options can be found on the help page plot/colorbar. Additionally, several commands now support color bars and display them by default: plots:-matrixplot, plots:-surfdata, plots:-complexplot, plots:-complexplot3d, and SignalProcessing:-Spectrogram. In addition, every plot created with with plot3d that uses a custom gradient or coordinate colorscheme will display a matching color bar if the colorbar option in given.

```
> plots:-contourplot(sin(x*y), x = -1 .. 1, y = -1 .. 1, filled,
  colorbar = ['discrete', barcaption = typeset(alpha)]);
```

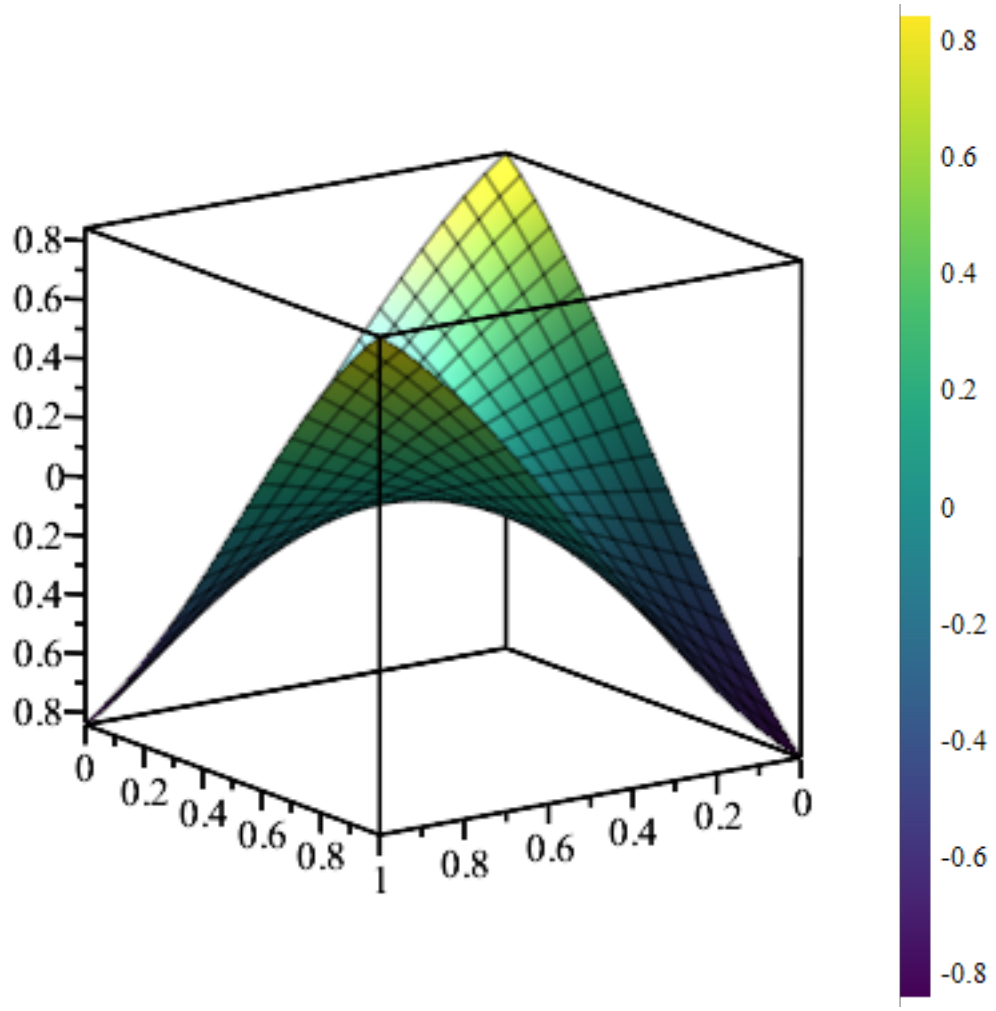



```
> A1 := Matrix(21, 21, (i, j) -> evalf(sin(1/100*(i - 11)*(j - 11))),
  datatype = float[8]):
```

```
> plots:-matrixplot(A1, 'dimension' = 2, 'gap' = 0.2);
```

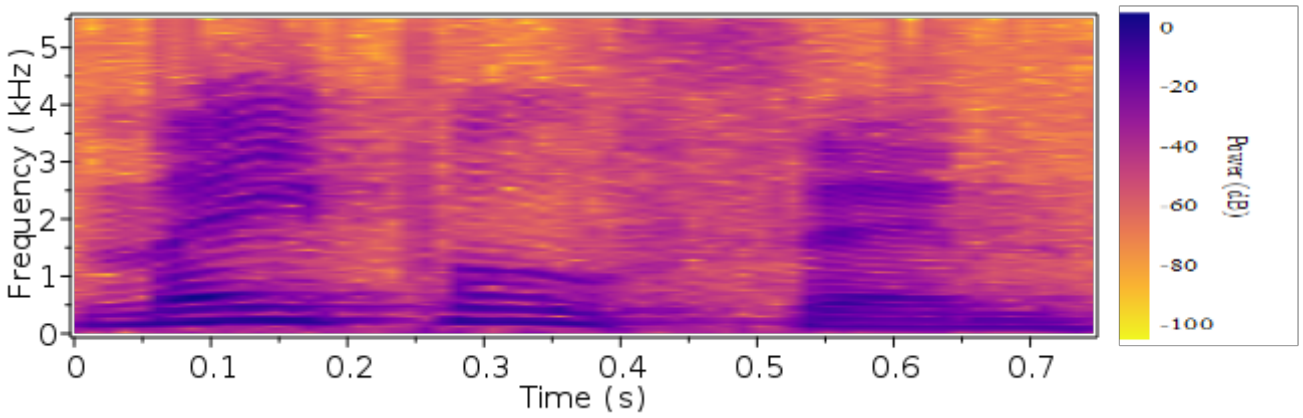


```
> plots:-surfdata(A1, colorscheme = "Viridis", 'dimension' = 3,
  'colorbar');
```

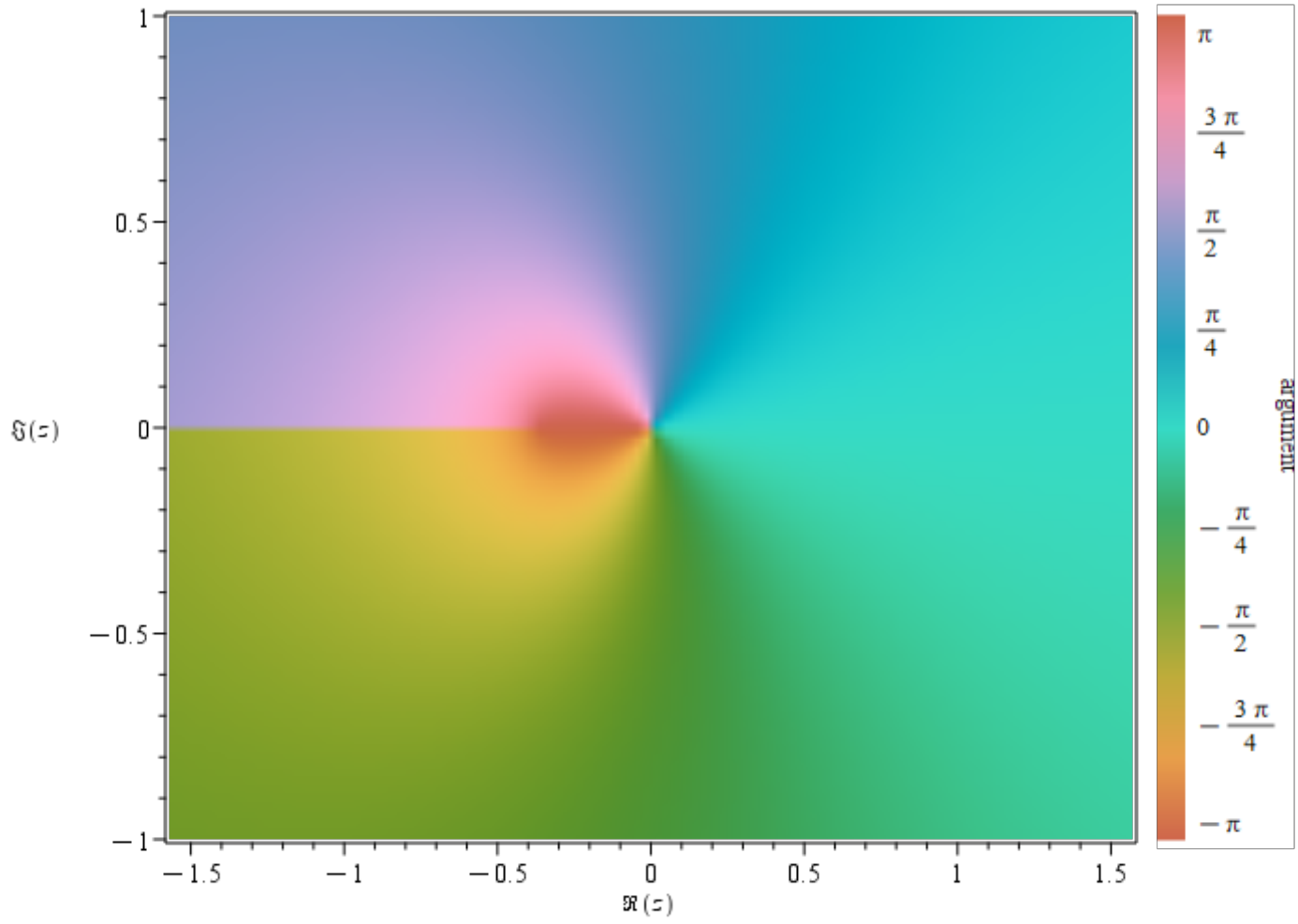



```
> audiofile := cat(kernelopts(datadir), "/audio/maplesim.wav"):
```

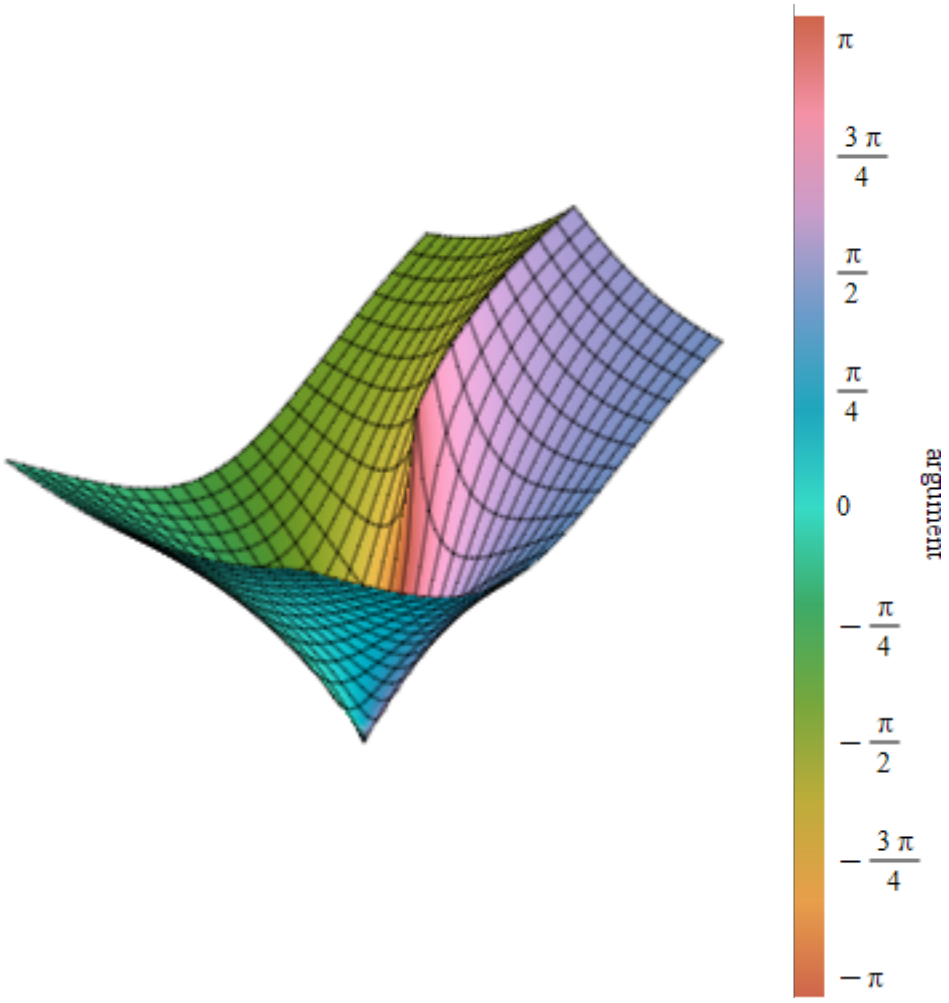
```
> SignalProcessing:-Spectrogram(audiofile, compactplot, size = [900,
  300], colorscheme = "Reverse Plasma");
```



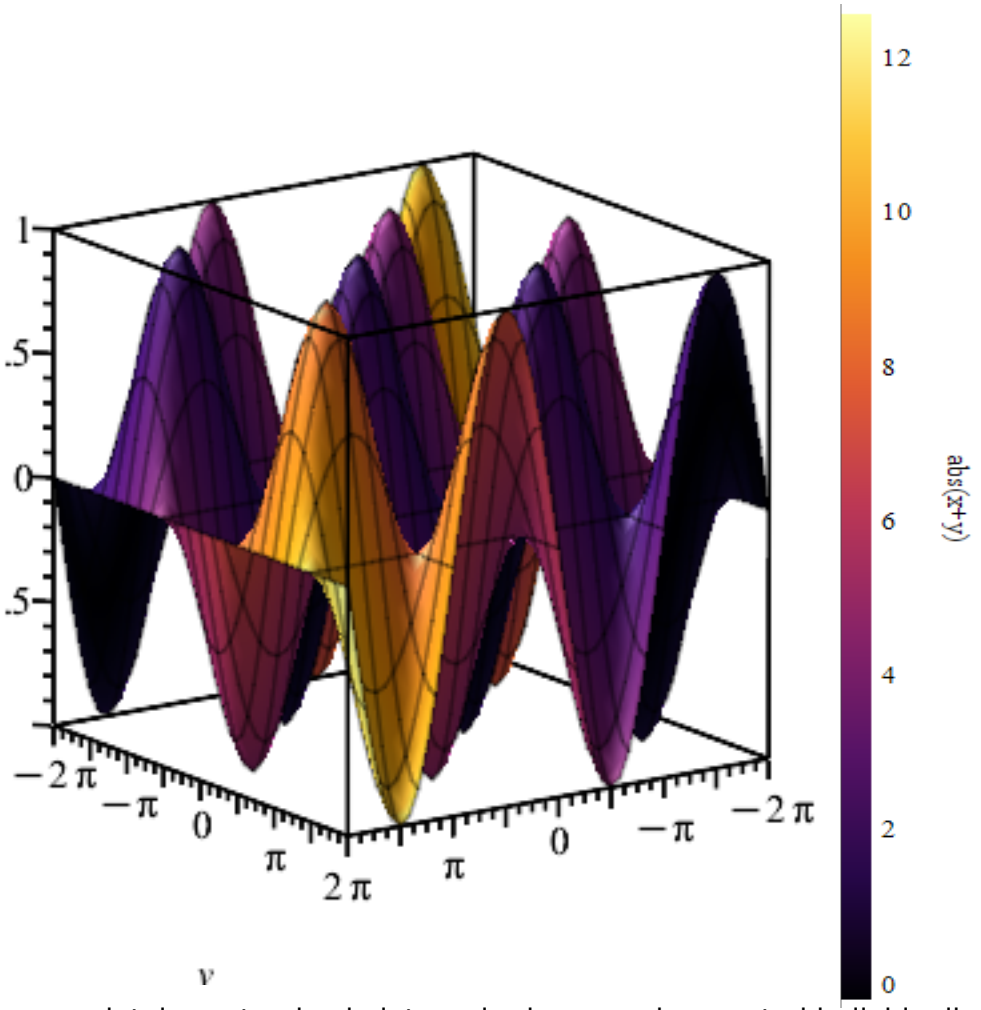

```
> plots:-complexplot(LambertW(z), z = -Pi/2 - I .. Pi/2 + I, size =
  [700, 500], axes = boxed);
```



```
> plots:-complexplot3d(LambertW(z), z = -Pi/2 - I .. Pi/2 + I, size =
  [700, 500], axes = none);
```



```
> plot3d(sin(x) * cos(y), x=-2*Pi..2*Pi, y=-2*Pi..2*Pi, colorscheme=
  ["xyzcoloring", (x,y,z)->abs(x+y), palette="Inferno"], colorbar=
  [barcaption="abs(x+y)"], size=[800, 600]);
```

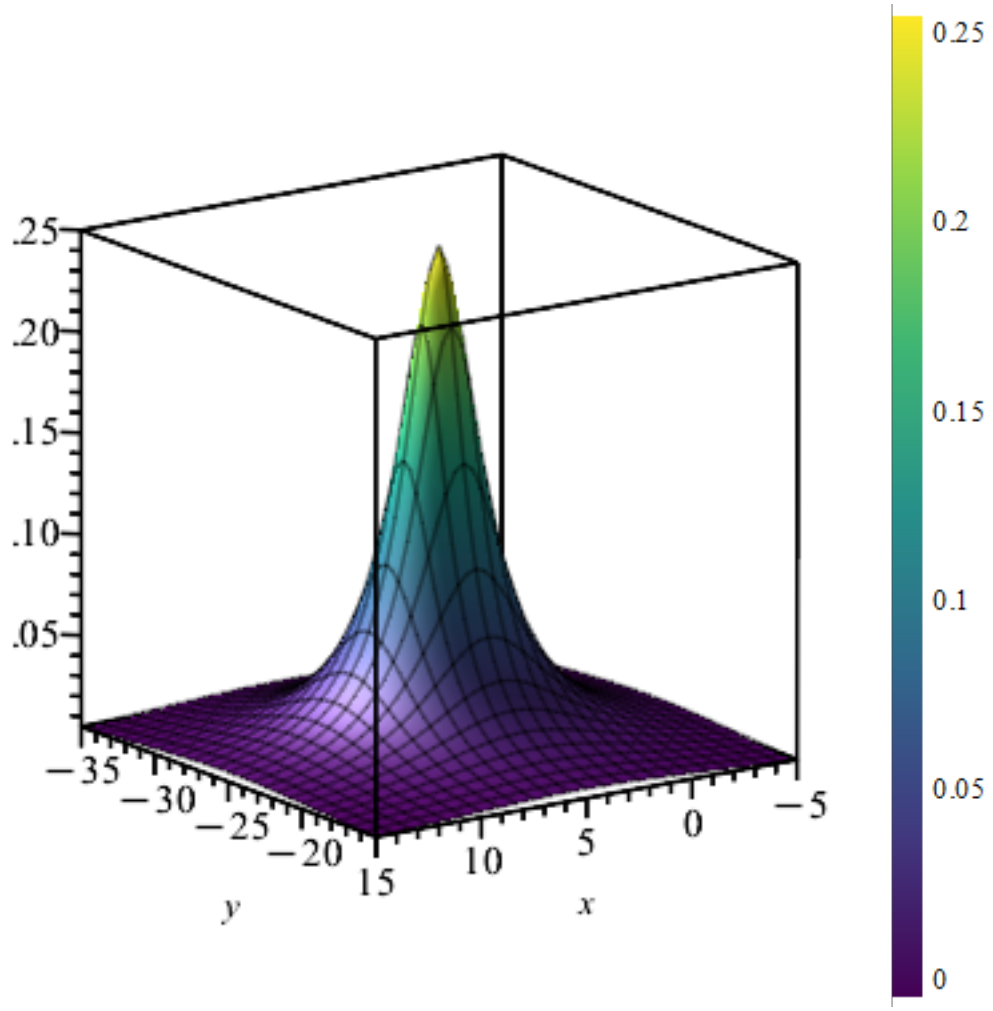


- Finally, for completely customized plots, color bars can be created individually with the new command plottools:-colorbar. Those bars can and then added any plot with the plots:-display command. If a plot already has a color bar this new bar will replace the existing one since multiple color bars are currently not supported.

```
> P := plot3d(1/((x-5)^2 + (y + 25)^2 + 4), colorscheme="Viridis"):
```

```
> cb := plottools:-colorbar(0..1/4, "Viridis"):
```

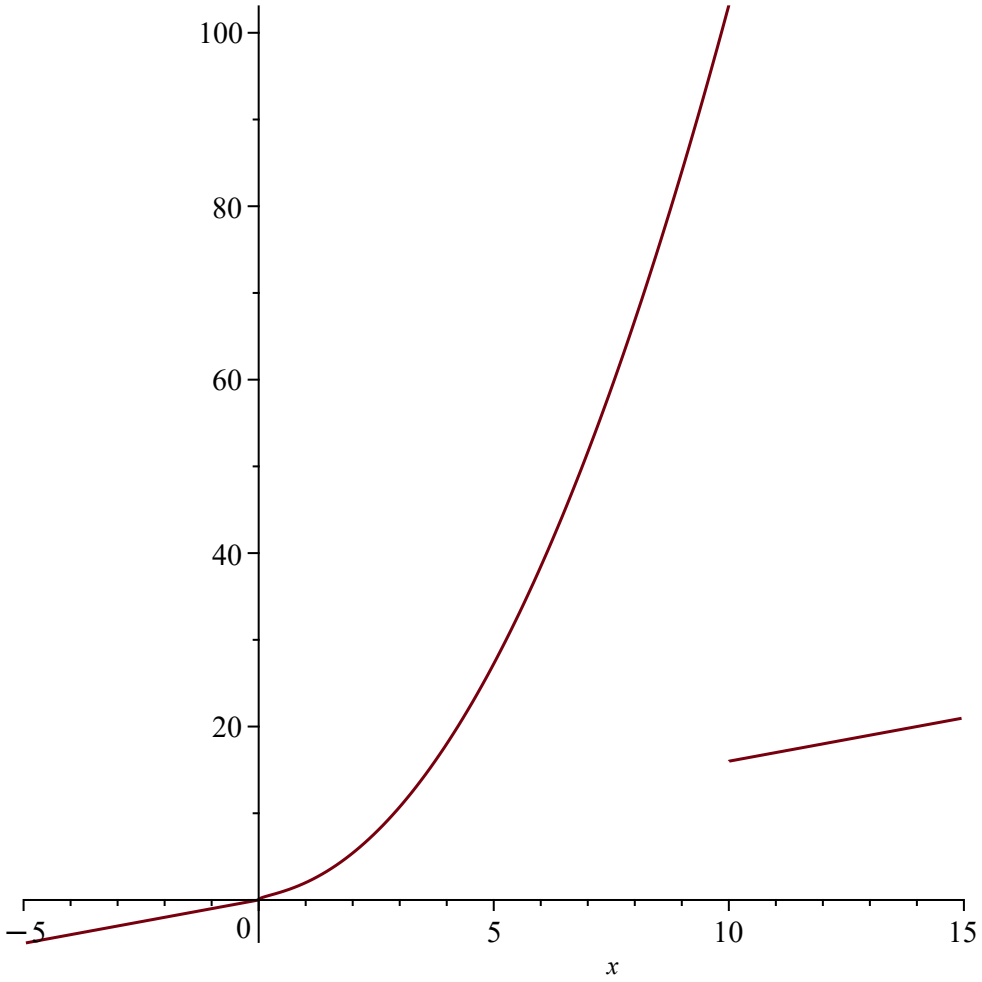
```
> plots:-display(P, cb);
```



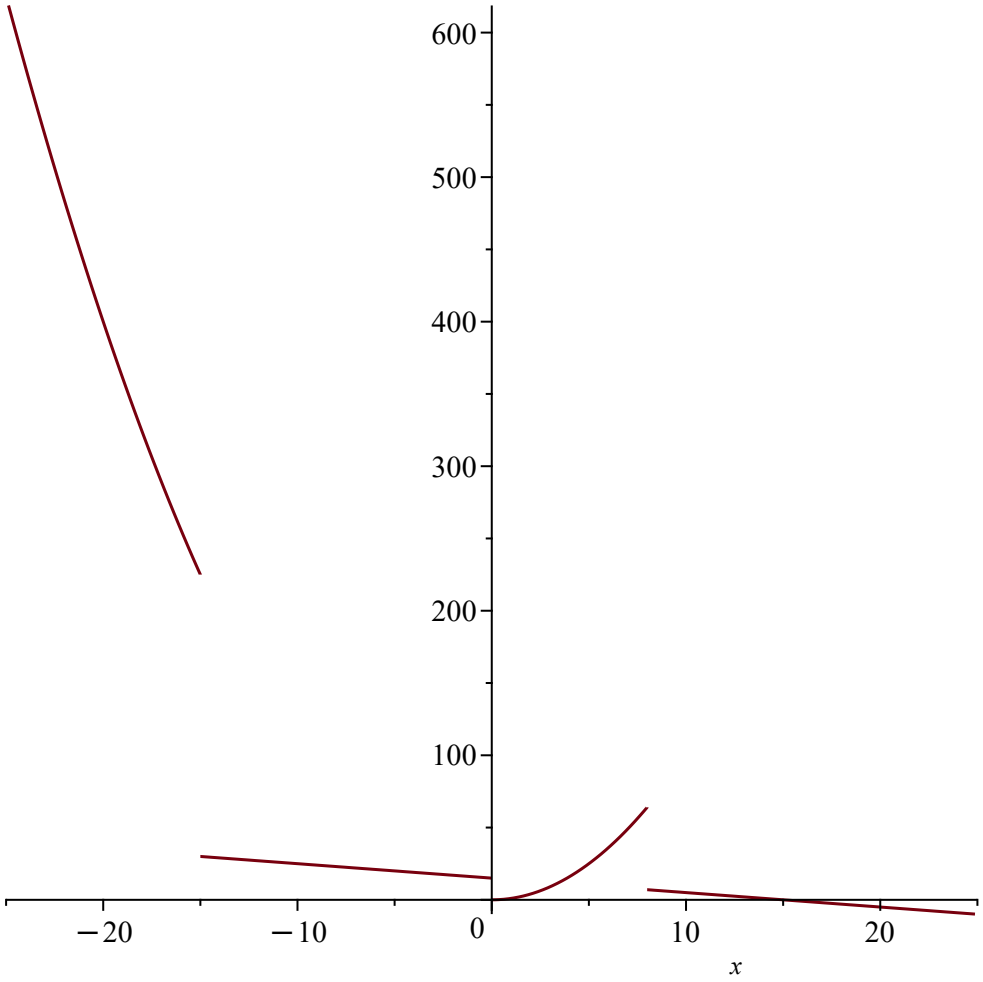
Piecewise plots

- Plots of many piecewise functions now show discontinuities by default (that is, without having to use the discont option).

```
> plot( piecewise( x> 10, x+6, And(x>0, x<=10), x^2+sqrt(x), x<0, x ),
  x = -5 .. 15 );
```

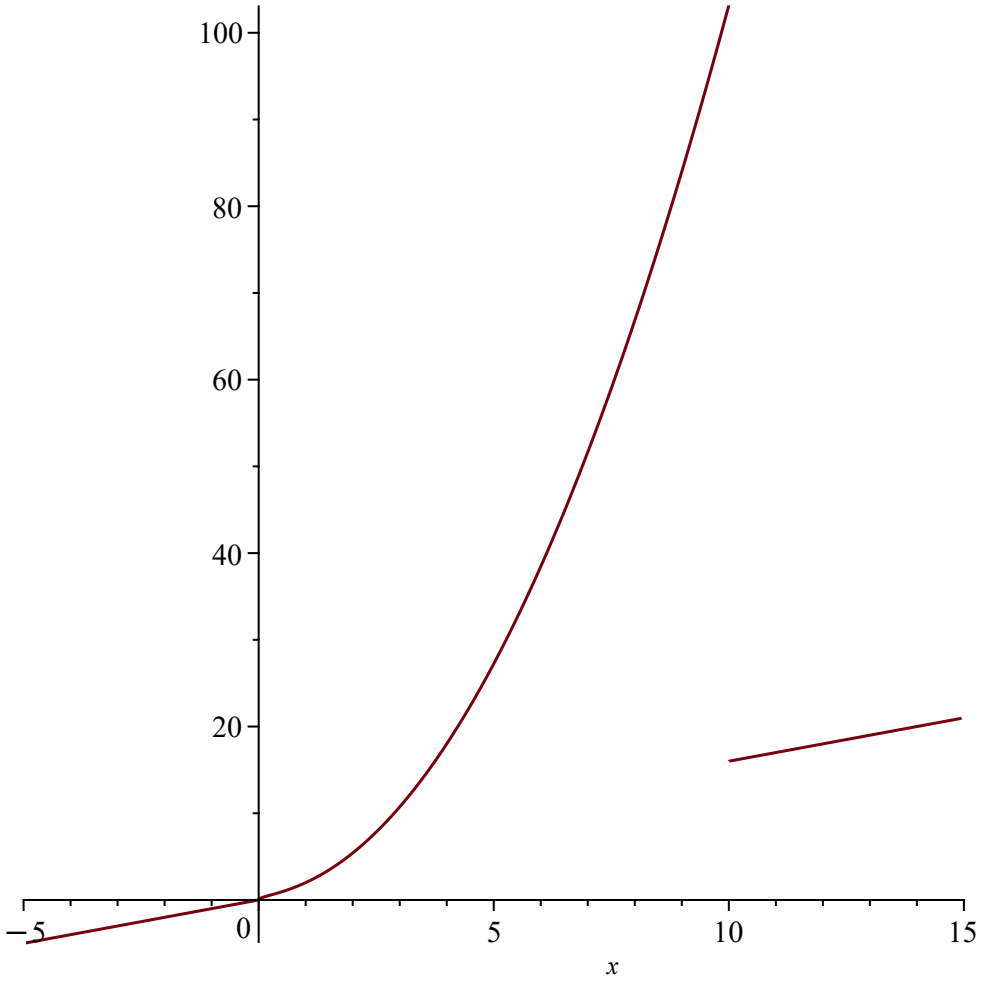


```
> plot( piecewise( x^3 + 7*x^2 - 120*x  < 0, x^2, 15 - x ), x = -25 ..
  25 );
```

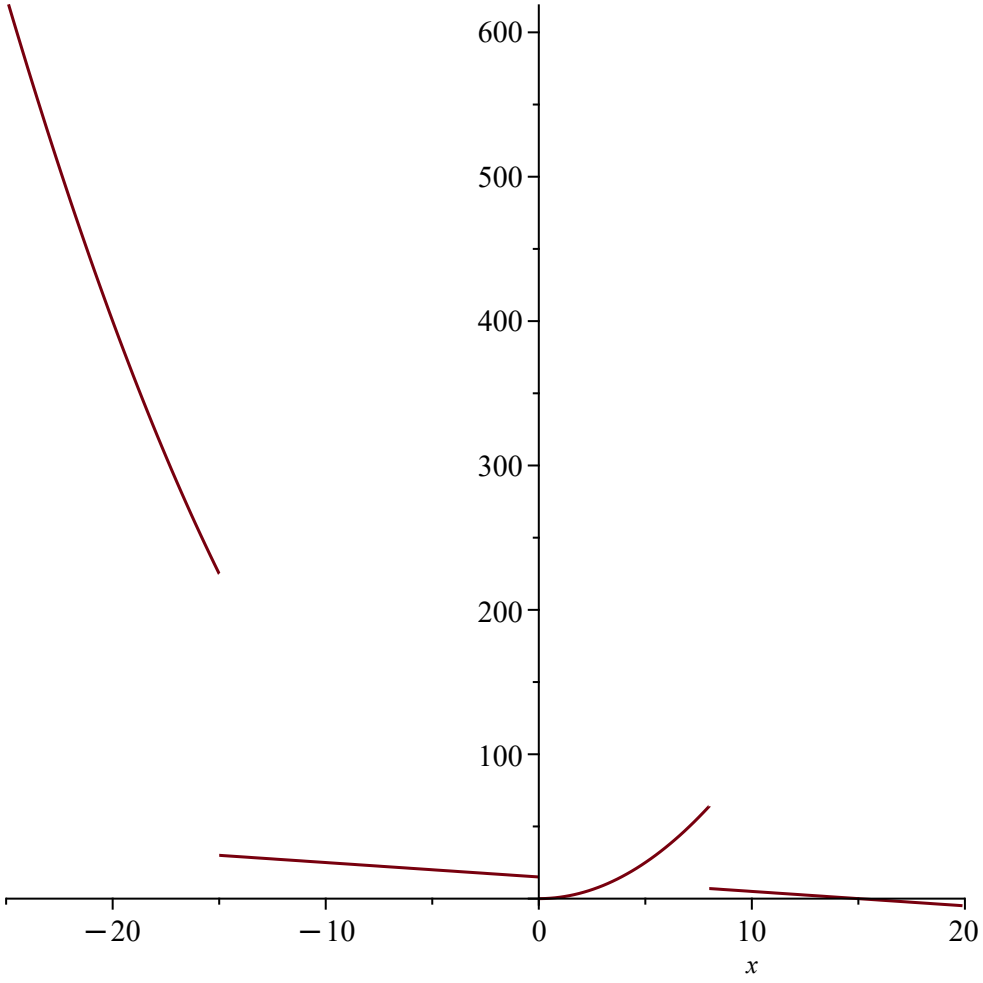


- In Maple, for most types of plots, you don't need to specify a domain for a plot explicitly. Maple then tries to find a domain that includes most of the interesting features of the plot. This functionality was improved for Maple 2024; most notably for piecewise functions.

```
> plot( piecewise( x> 10, x+6, And(x>0, x<=10), x^2+sqrt(x), x<0, x )
  );
```



```
> plot( piecewise( x^3 + 7*x^2 - 120*x  < 0, x^2, 15 - x ) );
```



Handling of the viewing ranges in Explore

A new option adaptview has been added to the Explore command, which controls the handling of the viewing range of interactive plots. The purpose is to allow for a more usefully consistent viewing range to be used without having to specify the view option.

With a default value of true, the adaptview option allows the viewing range to grow, as the parameter values are changed. As any Slider or controller is changed (or set back to an earlier value) then the largest view attained so far is retained.

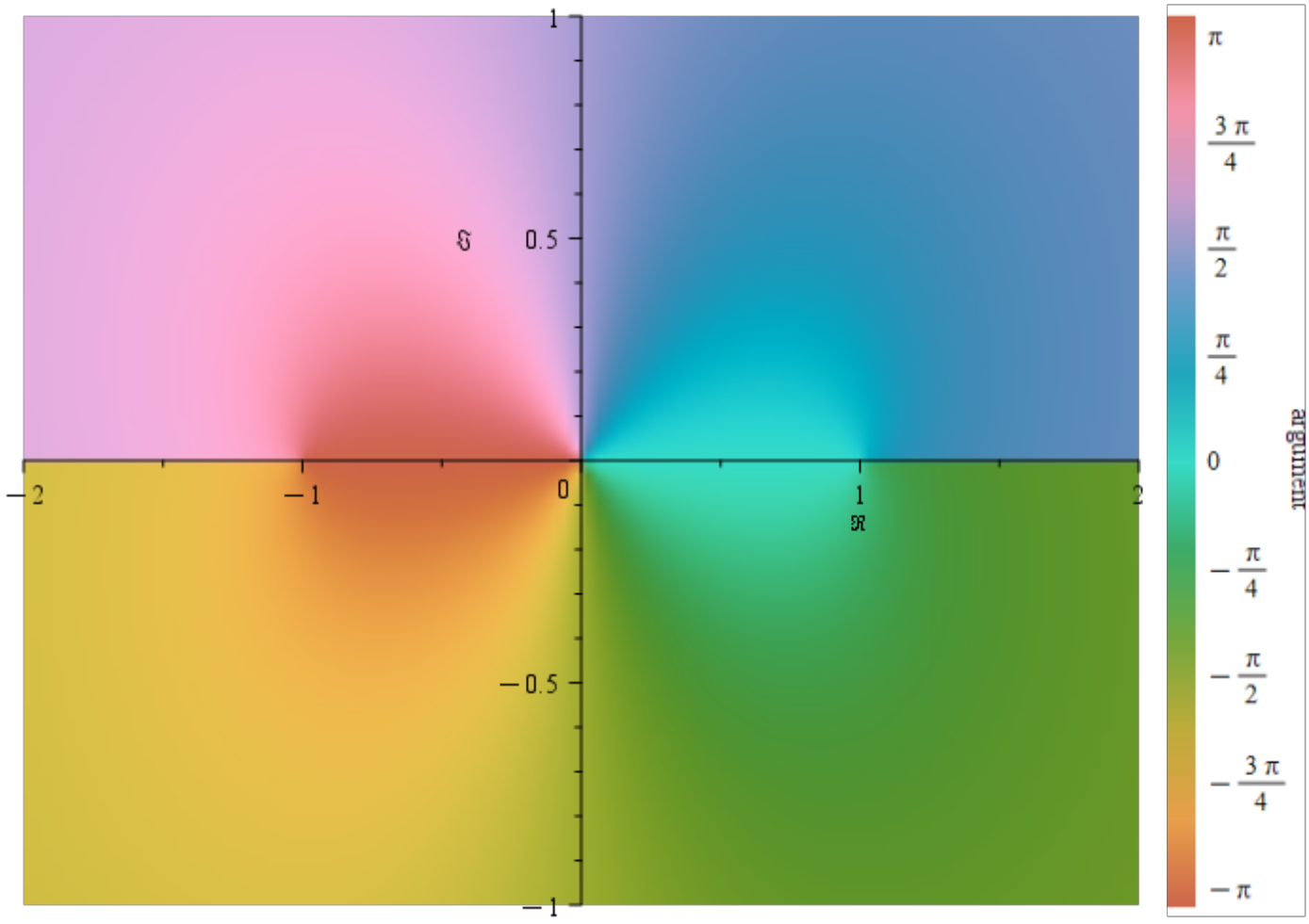
An additional feature is that the upper and lower parameter values are sampled in advance, and the larger range used for the first frame, which can often result in a larger viewing range being used up front.

The previous behavior, where all plotting frames get their own independent viewing range, is available by specifying adaptview=false.

Complex plot command improvements

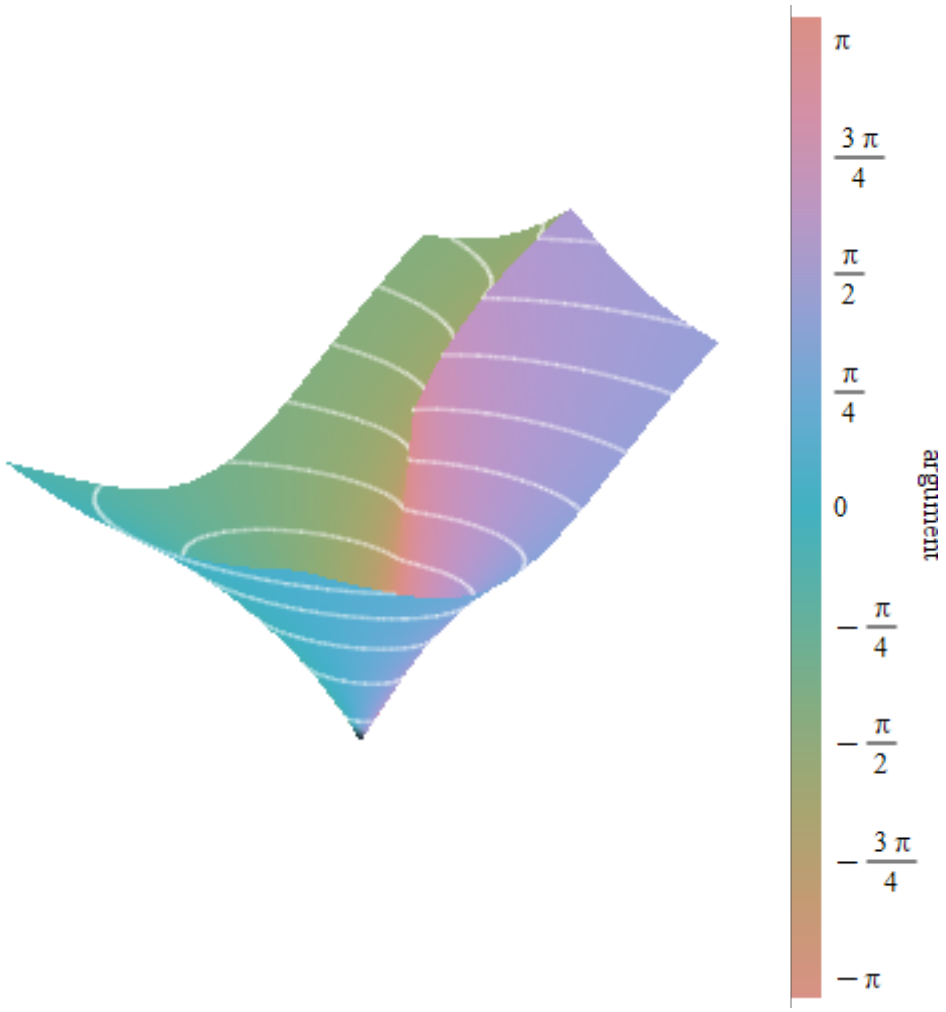
- In addition to the color bar support mentioned above, both complex plotting commands plots:-complexplot and plots:-complexplot3d have new options.
- The 2-D complexplot command now can create phase plots when given a complex range. That is a densityplot of the argument of the function over the complex plane.

```
> plots:-complexplot(arctanh, -2-I..2+I, size=[700,500]);
```



- The 3-D complexplot3d command now supports the colorscheme option to select custom colormaps to represent the argument of the function being plotted.

```
> plots:-complexplot3d(LambertW(z), z = -Pi/2 - I .. Pi/2 + I, size =
  [700, 500], axes = none, colorscheme="Isocircle", style=
  surfacecontour);
```



Parametric exploration in the Plot Builder

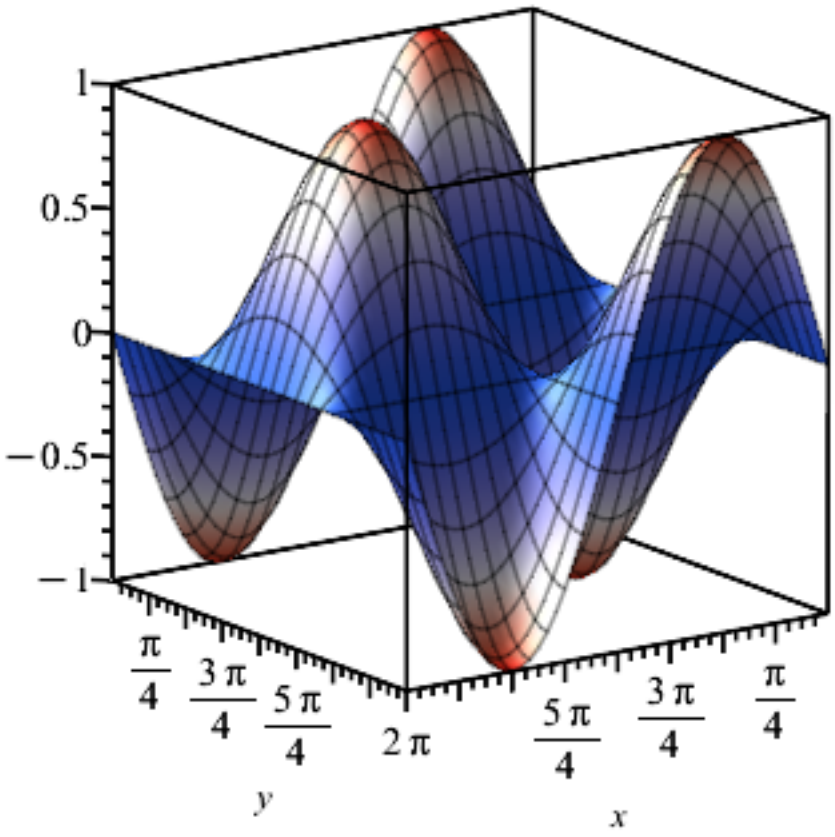
Various improvements have been made to the Plot Builder.

- The entry box for view ranges has been made into two entry fields, for upper and lower values.
- In the case of interactive plot building, where there are additional named parameters, names beginning with lower or uppercase x, y, or z are given preference as the initial choices for the first, second and third plotting variables respectively.
- Miscellaneous improvements have been made to the robustness of handling of extra parameters and interactive plot building.

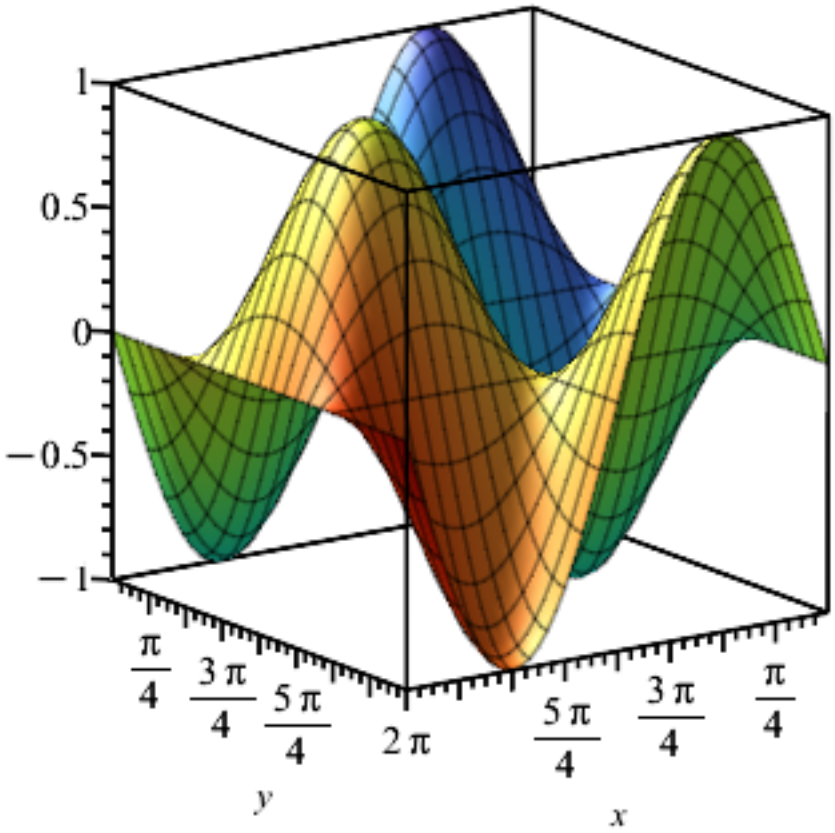
New coordinate colorscheme option

- The coordinate-based color schemes (see plot,colorscheme,coordinates) now support custom colormaps everywhere hue-based colorings were used before. This makes it easy to use the many built-in colormaps in ColorTools,ColorCollections in custom coloring schemes. To use a colormap instead of hue coloring, the option palette=P can be added to the colorscheme options. This works for both zcoloring and xyzcoloring.

```
> plot3d(sin(x)*cos(y), x=0..2*Pi, y=0..2*Pi, colorscheme=
  ["zcoloring", z->z^2, palette="CoolWarm"]);
```



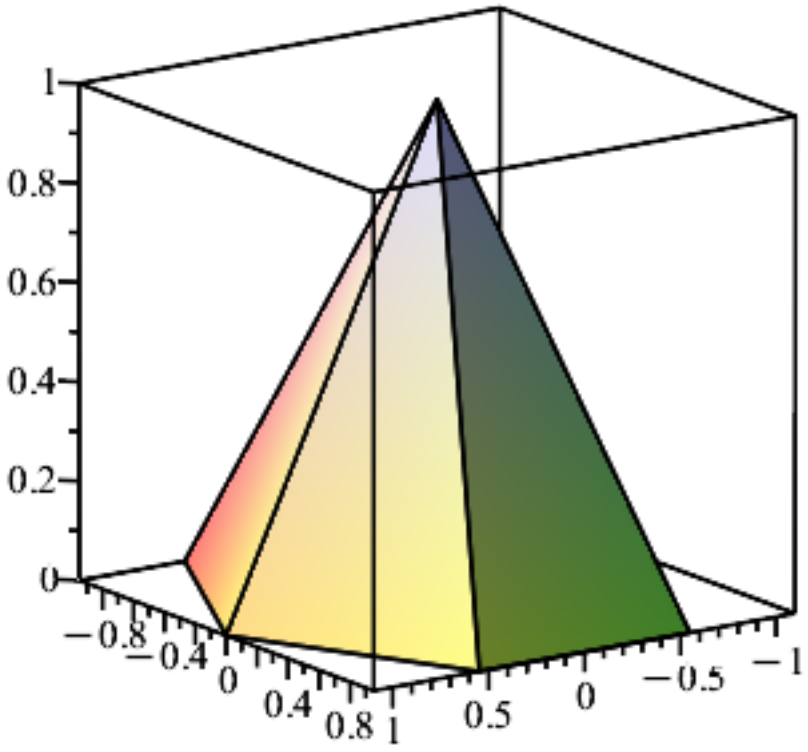
```
> plot3d(sin(x)*cos(y), x=0..2*Pi, y=0..2*Pi, colorscheme=
  ["xyzcoloring", (x, y, z)->x+y-z^2, palette="Turbo"]);
```



▼ New command for creating pyramids

- The command plottools:-pyramid can be used to create pyramids with a base of any shape known by plottools:-polygonbyname or with a two-dimensional polygon created by other plot commands.

```
> plots:-display(plottools:-pyramid("hexagon"));
```



```
> P := plottools:-polygon([[-1,1], [0,1], [-1,-1]]):
> plots:-display(plottools:-pyramid(P, 'base'=-1, 'height'=2));
```