

Plotting with Octave and Gnuplot

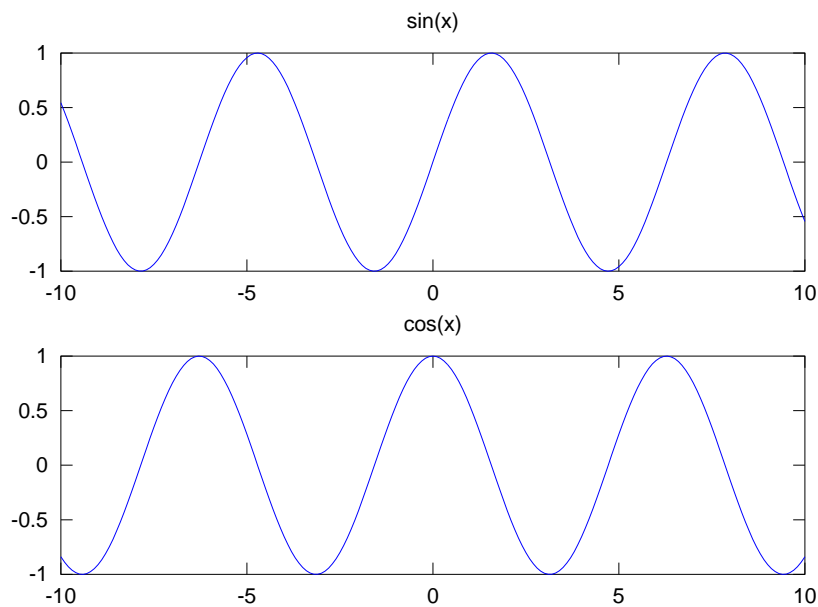
A few simple plots which illustrate how to work with Pure, Octave and Gnuplot in $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$. This is a selection of the some of the plots in the `plot_demo.pure` script of the `pure-octave` package.

```
> using gnuplot, math; using namespace gnuplot;
```

```
> /* Set some global defaults. */  
set (0, "defaultlinewidth", 1.0, "defaulttextfontname", "Helvetica",  
"defaultaxesfontname", "Helvetica", "defaultaxesfontsize", 8.0);
```

```
()
```

```
> /* Two simple function plots on one page */  
let x = -10:-9.9..10;  
let subplot (2, 1, 1), plot (x, map sin x), title ("sin(x)");  
let subplot (2.0, 1.0, 2.0), plot (x, map cos x), title ("cos(x)");  
let psplot();
```

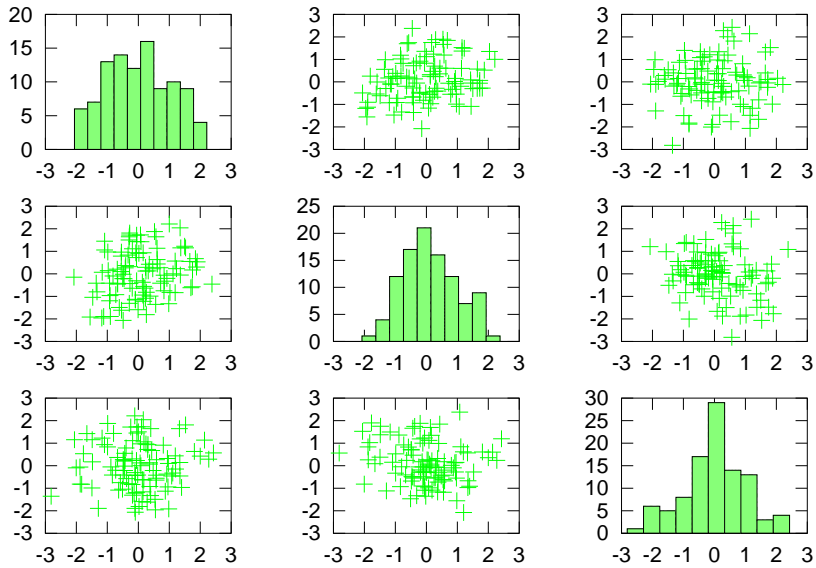


```
> /* A little animation. This pops up in a gnuplot window. */  
let () when  
  t = 0:0.2..20;  
  s = [sin(t)*exp(-t/5) | t = t];  
  a = [0, 20, -1, 1];  
  n = 8;  
  clf(); popup();  
  do frame $ linspace(0,n*pi,200) with  
    frame k = () when  
      plot(t, map (sin(k)*) s);  
      axis(a);  
      drawnow();  
    end;  
  end;  
  popdn();  
end;
```

```

> /* A collection of scatter plots of the columns of one matrix
    against another (histograms in the diagonal). */
let plotmatrix (randn (100, 3), "g+");
let psplot();

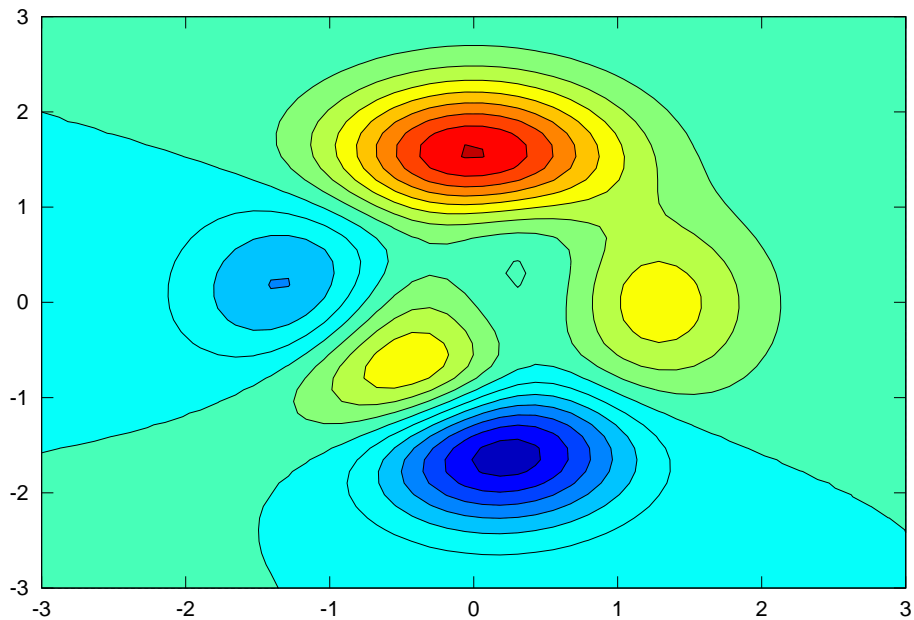
```



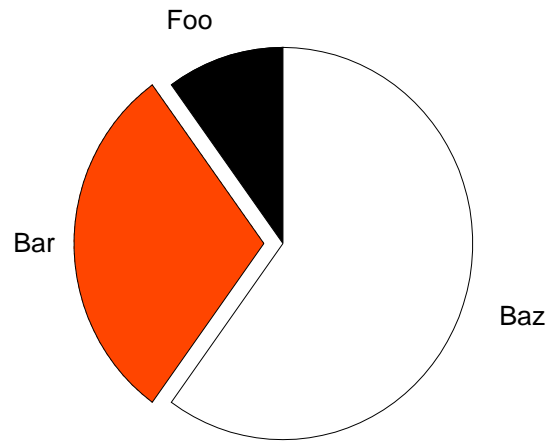
```

> /* A filled contour plot. */
let x, y, z = peaks3 50; let clf(), contourf (x, y, z, -7..9);
let psplot();

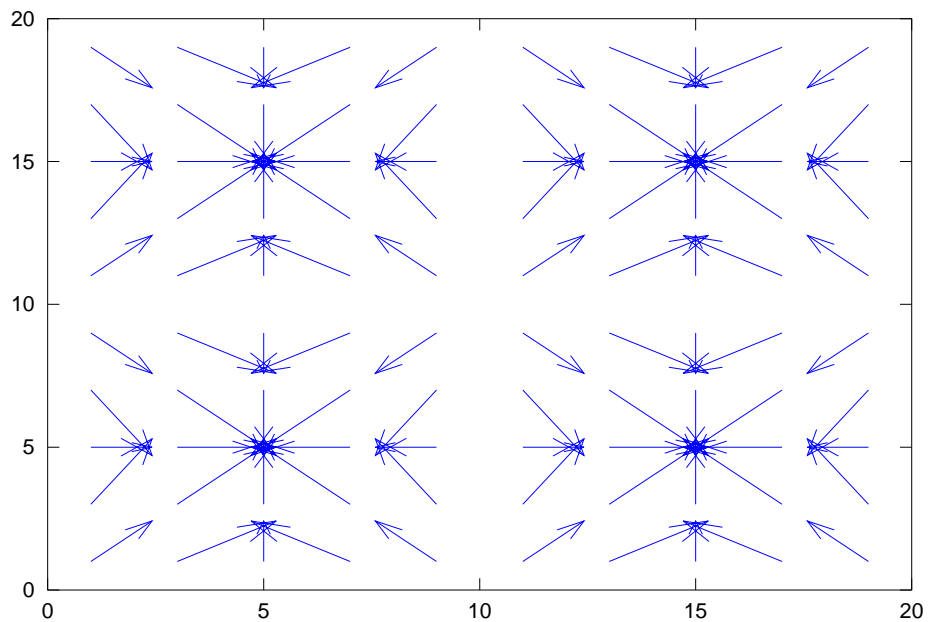
```



```
> /* A pie chart with "exploded" and labelled slices. */  
let pie([0.1,0.3,0.6], [0,1,0], ["Foo","Bar","Baz"]);  
let colormap "hot", psplot(), colormap "default";
```



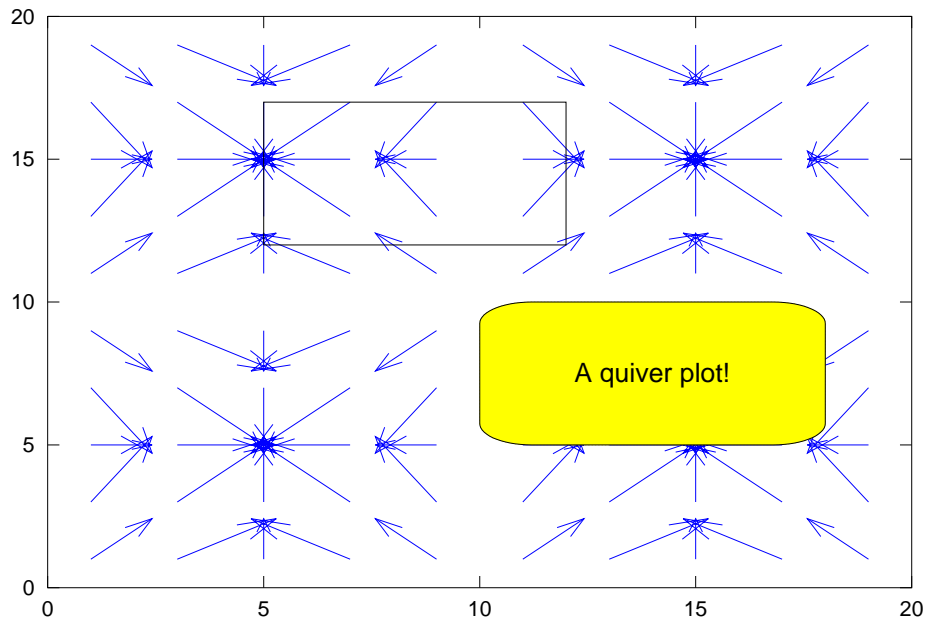
```
> /* A quiver (vector field) plot. */  
let x, y = meshgrid2 (1:3..20);  
let h = quiver (x, y, {sin (2*pi*x/10) | x = x}, {sin (2*pi*y/10) | y = y});  
let set (h, "maxheadsize", 0.33), psplot();
```



```

> /* Add some patches to the previous plot. */
let rectangle("Position", {5.0,12.0,7.0,5.0});
let rectangle("Position", {10.0,5.0,8.0,5.0}, "Curvature", {0.3,0.3},
"FaceColor", "yellow");
let text(14.0,7.5, "A quiver plot!", "horizontalalignment", "center");
let psplot();

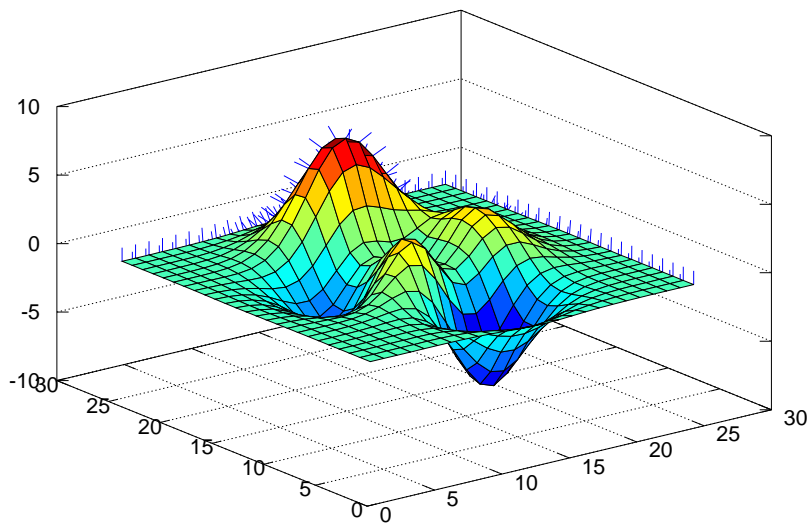
```



```

> /* A surface with normals. */
let surfnorm (peaks1 25), psplot();

```



```

> /* An isosurface example, adapted from the Octave manual. */

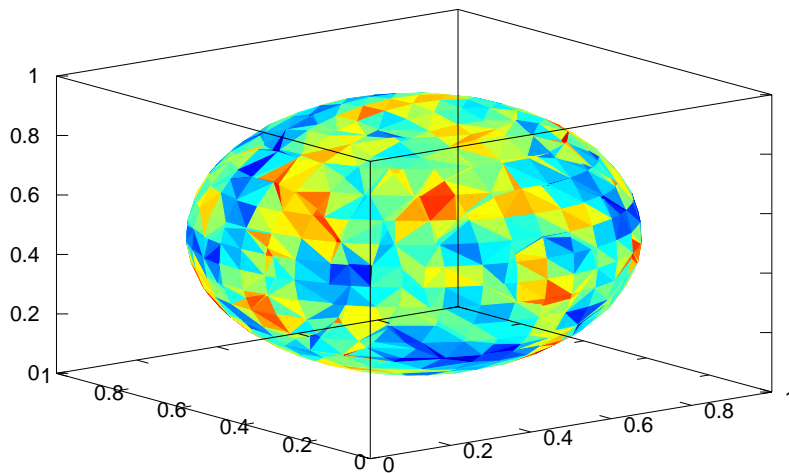
zipv3 = zipwith3.zipwith3;
mapv3 = map.map;

isofinish p = () when
  set (gca(),
      "PlotBoxAspectRatioMode", "manual",
      "PlotBoxAspectRatio", {1,1,1});
  set (p, "FaceColor", "interp");
  view (-38, 20);
end;

let n = 30; /* number of vertices in each direction */
let iso = 0.2; /* try different isovalues */
let lin = linspace (0, 2, n);
let x, y, z = meshgrid (lin, lin, lin);
let c = zipv3 (\x y z->abs ((x-0.5)^2 + (y-0.5)^2 + (z-0.5)^2)) x y z;
let f, v = isosurface2 (x, y, z, c, iso);

let clf();
let p = patch ("Faces", f, "Vertices", v, "EdgeColor", "none");
let cdat = rand (size c); /* random colors */
let isocolors (x, y, z, cdat, p), isofinish p, psplot();

```

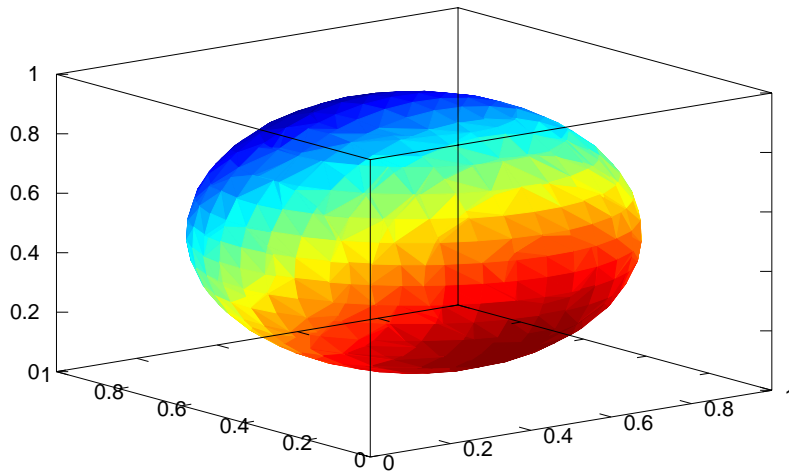


```
warning: division by zero
```

```

> let clf();
let p = patch ("Faces", f, "Vertices", v, "EdgeColor", "none");
let r, g, b = meshgrid (lin, map (2-) lin, map (2-) lin); /* gradients */
let isocolors (r, g, b, c, p), isofinish p, psplot();

```



warning: division by zero

```
> /* A mesh plot (the sombrero). */
let u = linspace (-8, 8, 41);
let x,y = meshgrid2 u;
let z = {sin r/r | r = {sqrt (x^2+y^2)+eps() | x,y = zip x y}};
let mesh (u, u, z), psplot();
```

