24. Two dimensional plots

Plotting is one of the main strengths of computer algebra. See the package **plots**:

```
[> with(plots);
```

24.1 Plotting Functions and Formulas

For example, suppose we wished to plot $f(x) = e^{-x^2} \sin(\pi x^3)$ over the interval $[-2, 2]$:

```
[> f := x -> exp(-x^2)*sin(Pi*x^3);
[> plot(f,-2..2);
```

When plotting a function, it suffices to give the range of the independent variable. For a formula, we must indicate the variable for the range, e.g.: `x = -2..2`, instead of `-2..2`. For example:

```
[> plot(f(z),z=-2..2);
```

Observe that with `f(z)` we obtained the formula defined by the function `f` evaluated at the symbol `z`.

24.2 Displaying and Animating Plots

We can assign plots to variables and save them for later display. Suppose we wish to add the amplitude e^{-x^2} to the plot above:

```
[> amp_plus := plot(exp(-x^2),x=-2..2,color=blue):
[> curve_plot := plot(f,-2..2):
[> amp_minus := plot(-exp(-x^2),x=-2..2,color=black):
[> display(amp_plus,curve_plot,amp_minus);
```

We can play with the frequency, adding an additional parameter k to f:

```
[> fxk := exp(-x^2)*sin(k*Pi*x^3);
[> movie := animate(fxk,x=-2..2,k=1..40,numpoints=200):
[> display(movie,insequence=true);
```

To play the movie, we click on the picture and we see a new toolbar appear. From this toolbar we select the play button. We can also let the movie repeat itself indefinitely.

24.3 Plotting around Singularities

To plot algebraic curves, we can use the command implicitplot:

```
[> implicitplot(x^3+y^3-5*x*y+1/5,x=-3..3,y=-3..3);
```

Not all curves will plot that easily:

```
[> f := (x^2+y^2)^3 + 5.12*(x^2+y^2)^2 - 5.15*(x^4-y^4) - 14.7456*y^2;
[> implicitplot(f,x=-3..3,y=-3..3);
```

As we adjust the range, Maple displays confusing pictures:

```
[> implicitplot(f,x=-11..11,y=-11..11);
```

This plot is made with hindsight (see below), but does still not look good:

```
[> implicitplot(f,x=-1.2..1.2,y=-1.2..1.2);
```

On this latest plot we see that the function does not seem to get drawn around the origin. The problem is the singularity at the origin $(0,0)$. In particular, it is easy to see that $\frac{\partial f}{\partial x}(0,0) = 0$ and $\frac{\partial f}{\partial y}(0,0) = 0$. To visualize the curve properly, we must convert to polar coordinates:

```
[> sf := subs({x=r*cos(t),y=r*sin(t)},f);
```

Since r=0 is just one point: the origin, we can divide out the common factor r^2:

```
[> nsf := normal(sf/r^2);
```

And now we solve for r:

```
[> sols := solve(nsf,r);
```

In this case, it suffices to take the first solution, to see the entire curve:

```
[> polarplot(sols[1]);
```

This curve is one of the curves "invented" by James Watt.

24.4 Drawing Sparse Matrices

Below we visualize a random matrix with entries consisting of zeros and ones.

```
[> randomize():                    # reset seed for random number generators
[> m := matrix(32,32,x->rand() mod 2):
[> plots[sparsematrixplot](m,axes=none,scaling=constrained,symbolsize=6);
```

With this facility we can make systematic patterns.

24.5 Exporting Plots

Once we have a nice picture, we may want to include it in a paper, or send it separately to the printer.

```
[> plotsetup(ps);                  # all plots will become postscript files
[> plotsetup(default);             # reset to the default
```

With the options of `plotsetup`, we can also set the height and the width of the pictures.

24.6 Assignments

1. The "neoid" is defined by $r = at + b$ in polar coordinates. Give the Maple commands
   (a) to make a plot for $a = 0.2$ and $b = 0.5$, for $t = 0 \ldots 6\pi$; and
   (b) to produce an animation of 10 frames, for $a = 0.2$ and for b going from 0.1 to 1 (also for $t = 0 \ldots 6\pi$).
2. Do `f := 4*x*y^2 + 2*x^3 - x^2;` and give Maple commands
   (a) to convert the formula for `f` into an equation in polar coordinates;
   (b) to solve the equation obtained in (a) for the radius;
   (c) to make the plot using the solution(s) obtained in (b).
3. The (i, j)-th entry Pascal matrix is defined by $\left(\begin{array}{c} i + j - 2 \\ j - 1 \end{array} \right)$.

   Create a sparse 32-by-32 matrix, setting the even entries in the Pascal matrix to zero, and the odd entries to one. The sparse matrix plot should show the Sierpinski gasket.