Draw Polynomial Roots

Author: Thierry Monteil

Questions

Plot the complex roots of all polynomials of degree 13 with coefficients in $\{-1,1\}$.

Play around this idea: change the allowed coefficients, change the degree, make animations, supperpose pictures, add colors, ...

Hints

Polynomials

To be fast, we will represent the complex numbers as floating-point numbers, so we will use `CDF`. To create the polynomial ring with complex coefficients and with indeterminate `x`, namely $\mathbb{C}[x]$, we can do:

```
sage: R = PolynomialRing(CDF, 'x') ; R
```

Or more in a more compact way (but it is equivalent):

```
sage: R = CDF['x']
```

If `L` is a list and `R` a polynomial ring, you can use its entries as the coefficients of an element of `R` by calling `R(L)`:

```
sage: L = [1, 2, 3, 4]
sage: P = R(L)
sage: P
```

You can get the list of its roots (without multiplicities) as follows:

```
sage: P.roots(multiplicities=False)
```

Products of iterables

To make the product of an iterable `L` (typically a tuple, a list or a set) with itself `k` times, you can do:

```
sage: from itertools import product
sage: L = ['a', 'b']
sage: k = 3
sage: product(L, repeat=k)

sage: list(_)
```

Plot and animate

To plot a list of points, you can use the `points()` function. If you want each point to be a single pixel (not a larger ball), you can use the `size=1` option. You can have a look at the doc for further options.

To learn more about plotting, you can have a look at http://doc.sagemath.org/html/en/prep/Advanced-2DPlotting.html (sagenb live / jupyter live)

To get some animation examples, you can look at the doc and the source code of the `sage.plot.animate` module. Please note that animations do not work yet on jupyter notebook.

Note that the `plot` objects can be added with eachother, leadind to a superposition of the plots.