Apollonius' Circle Problem on a Canvas

Jasmine Otto: A D3 interface to phcpy in Jupyter

Python

Fast blackbox algebraic solver via polynomial homotopy continuation.

```
from phcpy.solver import solve
```

Solves systems of polynomials by continuous map from a system solved by roots of unity. Can filter for real solutions of plausible residual.

```
''' // 6 equations
x^2 + y^2 = (r ± 1)^2
(x-c2x)^2 + y^2 = (r ± r2)^2
(x-c3x)^2 + (y-c3y)^2 = (r ± r3)^2
'''
```

Yields real-time answers (~10fps) to small-enough problems given only a high-level specification.

Caveats

PHCpack requires GNU Ada to compile.

requireJS is finicky, sometimes must be serial.

```
requirejs.config({paths: ...});
```

JS namespace gets polluted. May compile JS using cell magic from TypeScript, etc?

Input events collide with pagescroll on mobile.

IPython kernel.execute() is still undocumented, interface flags etc. are probably unstable.

Notebook

Bidirectional Python / JS comms.

```
// Arbitrary kernels?

var ker = IPython.notebook.kernel;
ker.execute("",
  {shell: {reply: ...}},
  {user_expressions:
   {output: ...}}
  )
```

Kernel will eventually evaluate our (programmatically defined) code and ping our callback.

```
function handleReply(out) {
  ans = mungeReply(out)
  output.selectAll('circle')
     .data(ans)
}
function mungeReply(py_out) {
  res = out.content
          .user_expressions.output
  return JSON.parse(
           res.replace(', "))
}
var output_code = "solve(poly("
  + input.select(...)
         .attributes[...].value+','
  + ...
```

Want to cut down boilerplate until generated html is a resuable widget.

Javascript

Data-driven frontend - pictures in, pictures out.

```
var svg = d3.select('#id')
  .append('g')
  .selectAll('circle')
```

d3 selections accept data entry, attribute binding, and behaviors.

```
selection.filter(...).call(
  d3.behavior.drag()
   .origin(...).on(...)
);
```

Request kernel update when input state changes.

References

Jan Verschelde: Modernizing PHCpack through phcpy. In Proceedings of EuroSciPy 2013 p71-76, 2014. arxiv.org/abs/1310.0056

Tanya Schlusser: Callbacks: Python + mpld3 + Jupyter 4. gist.github.com/tanyaschlusser/047148b1411ba4e05bb7

The jupyter and d3 communities on stackexchange, etc.

Inspo from Processing, a tool for art (or the cybernetics of classical geometry)