2014-05-12.sagews

May 12, 2014

Contents

1 Math 480b Sage Course

1.1 A Tour of More Advanced Cython Functionality

1.2 May 12, 2014

Screencast: REMIND ME!

Plan

- Questions
- Homework:
  - I collected 6, redistributed grading, etc. Let me know if youre missing something.
  - Go over hw7, which is about Cython and your projects.
- Cython, part 2:
  - Creating a standalone Cython module
  - Cython support for numpy it doesnt seem to work at all in Sage right now due to incompatibilities, so were skipping it. Sigh.
  - Editing Cython code in the Sage library
  - Cython support for C++

Note: article about GO for student doing a project on it: `http://www.wired.com/2014/05/the-world-of-computer-go/`

1.3 Creating a standalone Cython module

We will discuss this example: `https://github.com/cython/cython/wiki/examples-mandelbrot`

Definition: The Mandelbrot set is the set of $c \in \mathbf{C}$ such that the sequence of complex numbers given by $z_0 = 0$ and $z_{n+1} = z_n^2 + c$ remains bounded.

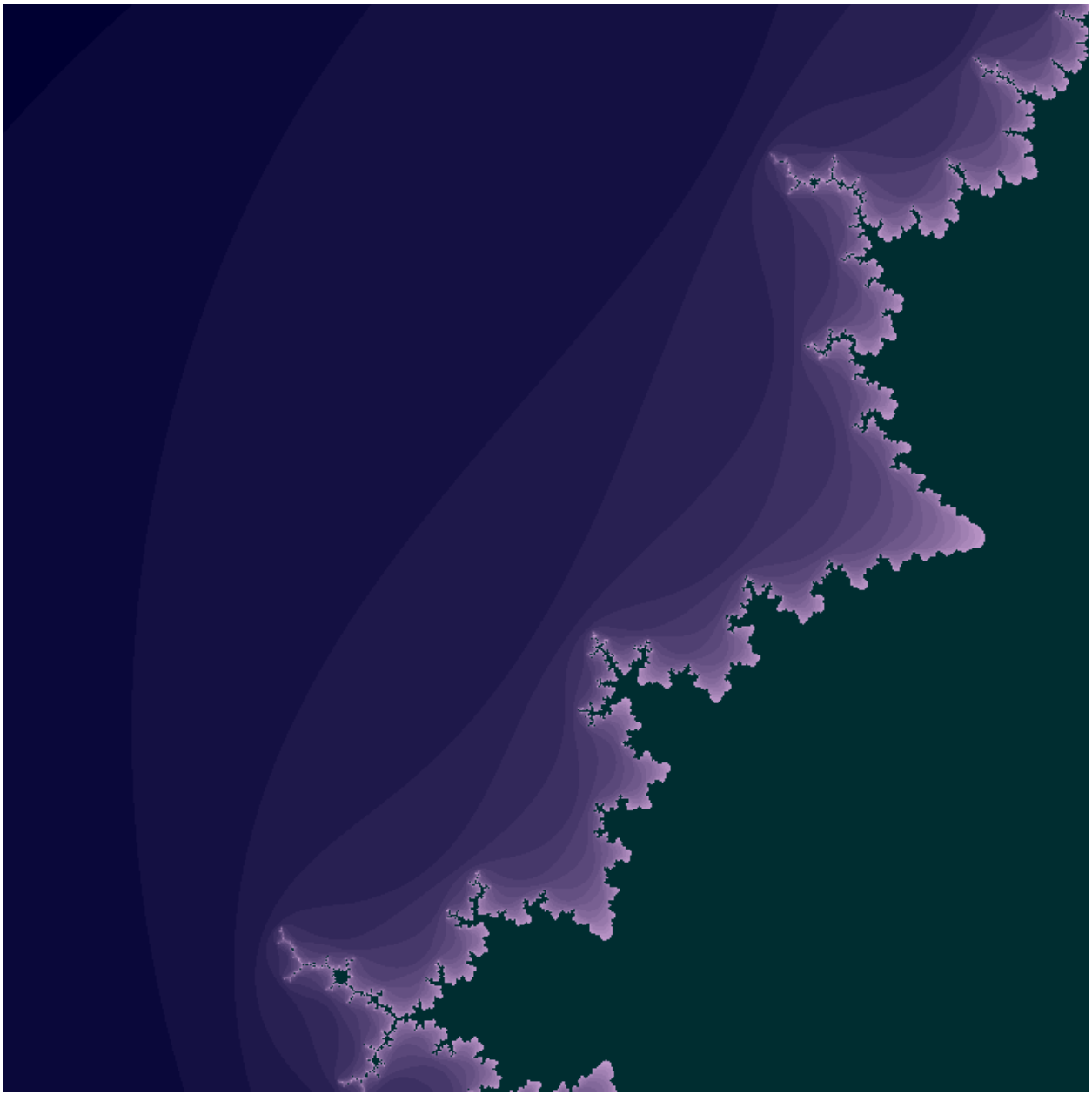
- Look at mandelcy1.pyx code
- Look at setup.py also look at `http://docs.cython.org/src/quickstart/build.html`
- Build the code
- Run it below to create an image (which we display)

```
# BIG WARNING: though this appears to work, once you import *once* you \
   can never import/reload again until you
# completely restart the worksheet (or to randomly name the cython module\
   ).
sys.path.append(".")
import mandelcy1
```

```
mandelcy1.demo()
```

```
it took 0.305397 seconds to run
```

```
salvus.file("mandelbrot.png")
```



1.4 Editing and building Cython code in the Sage library

- Edit the code as normal. But if the file ends in .pyx, it is Cython code, so you have to be aware of the implications.
- Build the code as usual: sage -br

```
# do an example involving integer.pyx
```

1.5 Cython support for external library code

A key feature of Cython is that you can use it to make functionality from existing C/C++ code available to Python (and Cython) code.

```
%cython

# very simple example -- use the C library log directly
# See http://docs.cython.org/src/userguide/external_C_code.html for much \
   more

cdef extern from "math.h":
    double log(double)

cdef class DoubleList:
    cdef double* v
    cdef int n
    def __init__(self, v):
        self.n = len(v)
        self.v = <double*> sage_malloc(sizeof(double)*self.n)
        cdef int i
        for i in range(self.n):
            self.v[i] = v[i]

    def __del__(self):
        sage_free(self.v)

    def log(self):
        import math
        return sum(math.log(self.v[i]) for i in range(self.n))

    def log1(self):
        cdef double ans=0
        cdef int i
        for i in range(self.n):
            ans += log(self.v[i])
        return ans
```

https://cloud.sagemath.com/blobs/cdefs.html?uuid=77a6033e-4539-4f1b-b9e4-8888a4e59f4bShowauto-generated

```
v =DoubleList(range(1,100000))
```

```
%timeit v.log()
```

```
5 loops, best of 3: 39.4 ms per loop
```

```
%timeit v.log1()
```

```
25 loops, best of 3: 9.07 ms per loop
```