

# Cython Development

Robert Bradshaw

Sage Seminar, May 22, 2007



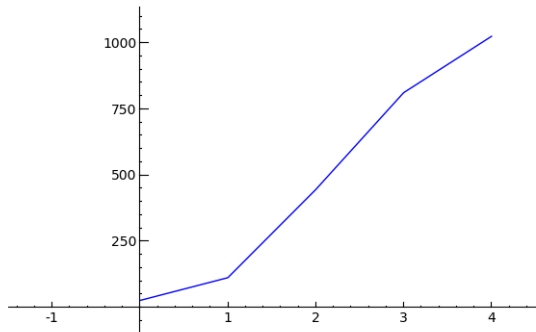
Cython's main advantage over Pyrex is that it has an *open* development model

Modeled after the Sage project:

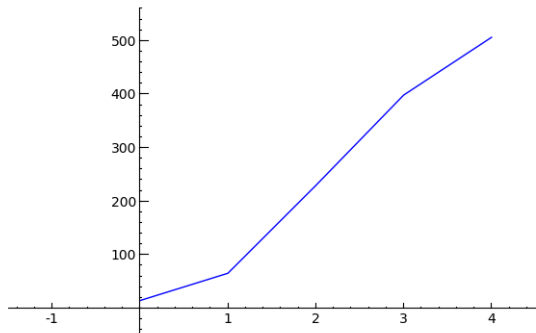
- ▶ Every copy of Cython comes with the revision control system
- ▶ Open bugs and features tracker  
[http://trac.cython.org/cython\\_trac/](http://trac.cython.org/cython_trac/)
- ▶ Heavy use of the Wiki
- ▶ Want automated testing

This will be a subject of discussion at Dev Days.

Total volume of emails



Total count of emails



- ▶ Support full Python syntax (2.x and 3.0)
  - ▶ Closures (inner functions, classes, and generators)
  - ▶ with statement
  - ▶ decorators
  - ▶ scoping particularities
- ▶ Optimized code *without* direct use of C/API
- ▶ Eventual Inclusion into Python

See <http://wiki.cython.org/enhancements>

*Not* trying to create a completely new language.

(Much of the difficulty with Cython lies in understanding the gap between Cython and Python, we want to reduce, not increase, this gap.)

The difference between Python and Cython is static typing.

Dag Sverre (and others) proposed to give more power in manipulating the parse tree.

Reduce the 1:1 correspondence between Python and C code.



# Concrete Example

- ▶ Fully understand feature (e.g. PEP)
- ▶ Modify Lexicon
- ▶ Modify Parser
- ▶ Create new Node
  - ▶ declarations
  - ▶ type analysis
  - ▶ code generation

- ▶ temporary variables
- ▶ `pxi` vs. `pxd`
- ▶ phase offsets
- ▶ declarator nodes