Cython

Python Module of the week, INM-6

Mar 17th 2023 I Javed Lindner, Kirsten Fischer

What is Cython?

A python module

What is the idea behind Cython?

JÜLICH
Forschungszentrum

What is the idea behind Cython?

- **Short** answer: Write C without writing C

JÜLICH Forschungszentrum

What is the idea behind Cython?

- **Short** answer: Write C without writing C
- **Better** answer:
  - Rid (parts) of Python code which are slow using advantages from C (static types etc.)

Example: Loops in Python

Rule 1 in Python: Don’t write loops if you can avoid it.

-> But why are Python loops slow? Part of the reason: Dynamic types

Example: Loops in Python

Rule 1 in Python: Don’t write loops if you can avoid it.

-> But why are Python loops slow? Part of the reason: Dynamic types

Cython: Solve problem by static typing variables (such as iterators)

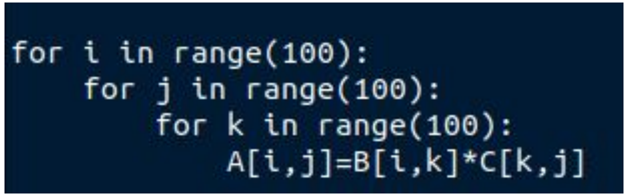
How is Cython better?

- compiled code
- static typing
- profiling options for code

Cython: Static types

Declare type when variable occurs

Short examples:

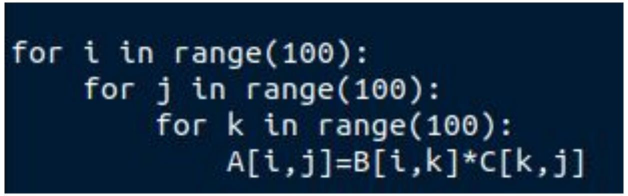


```
for i in range(100):
    for j in range(100):
        for k in range(100):
            A[i,j]=B[i,k]*C[k,j]
```

Cython: Static types

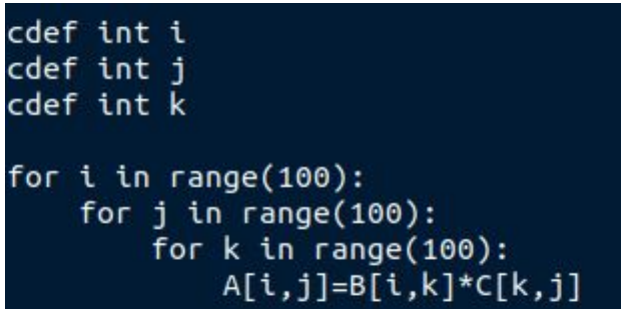
Declare type when variable occurs

Short examples:



```python
for i in range(100):
    for j in range(100):
        for k in range(100):
            A[i,j]=B[i,k]*C[k,j]
```

$\longrightarrow$

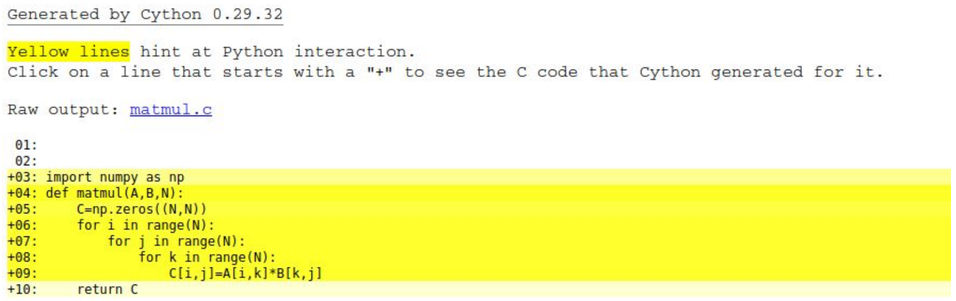


```cython
cdef int i
cdef int j
cdef int k

for i in range(100):
    for j in range(100):
        for k in range(100):
            A[i,j]=B[i,k]*C[k,j]
```

Cython: Annotation

Useful to see which code needs cythonization



Generated by Cython 0.29.32

Yellow lines hint at Python interaction.
Click on a line that starts with a "+" to see the C code that Cython generated for it.

Raw output: [matmul.c]

```
 01:
 02:
+03: import numpy as np
+04: def matmul(A,B,N):
+05:     C=np.zeros((N,N))
+06:     for i in range(N):
+07:         for j in range(N):
+08:             for k in range(N):
+09:                 C[i,j]=A[i,k]*B[k,j]
+10:     return C
```

Cython: Workflow and what you want to avoid

1.) Time code (using `timeit` or `time`) -> Sufficiently fast?
2.) Profile code (using `cProfile/lineProfiler`) -> Which parts can be optimized?
3.) Start easy, introduce static types and compile cython code
4.) Still not fast enough? Use `annotate` to identify problematic areas/bottlenecks

Cython: Workflow and what you want to avoid

1.) Time code (using `timeit` or `time`) -> Sufficiently fast?
2.) Profile code (using `cProfile/lineProfiler`) -> Which parts can be optimized?
3.) Start easy, introduce static types and compile cython code
4.) Still not fast enough? Use `annotate` to identify problematic areas/bottlenecks

$\downarrow$

Avoid rabbithole of code optimization.

Cython: Workflow and what you want to avoid

1.) Time code (using `timeit` or `time`) -> Sufficiently fast?
2.) Profile code (using `cProfile/lineProfiler`) -> Which parts can be optimized?
3.) Start easy, introduce static types and compile cython code
4.) Still not fast enough? Use `annotate` to identify problematic areas/bottlenecks

Avoid rabbithole of code optimization.

If it is not worth it to rewrite a whole program in C/C++--> Use **Cython**

Cython: Workflow and what you want to avoid

1.) Time code (using `timeit` or `time`) -> Sufficiently fast?
2.) Profile code (using `cProfile/lineProfiler`) -> Which parts can be optimized?
3.) Start easy, introduce static types and compile cython code
4.) Still not fast enough? Use `annotate` to identify problematic areas/bottlenecks

Avoid rabbithole of code optimization.

If it is not worth it to rewrite a whole program in C/C++--> Use **Cython**

Alternatives include: C2py, F2py

Links and Sources

Overview

https://pythonprogramming.net/introduction-and-basics-cython-tutorial/

https://nyu-cds.github.io/python-cython/02-executing/

https://www.peterbaumgartner.com/blog/intro-to-just-enough-cython-to-be-useful/

https://events.prace-ri.eu/event/1147/contributions/1184/attachments/1445/3026/Cython.pdf

JÜLICH Forschungszentrum