Vectorization and Cython

1. Executing a Python Function
   - a motivating example
   - timing the execution
2. Speeding Up Python
   - vectorization
   - Cython

MCS 320 Lecture 10
Introduction to Symbolic Computation
Jan Verschelde, 21 June 2024

Vectorization and Cython

1. Executing a Python Function
   - a motivating example
   - timing the execution
2. Speeding Up Python
   - vectorization
   - Cython

Approximating π

One way to approximate π is
to compute the area of the unit circle via numerical integration.

$$\frac{\pi}{4} = \int_0^1 \sqrt{1-x^2} \approx \frac{1}{n} \sum_{k=1}^{n} \sqrt{1-\left(\frac{k}{n}\right)^2}$$

This way is certainly not the best way to approximate π.

But the formulas are simple and computationally intensive,
which serves as a good motivating example to speedup code.

The n is the parameter to control the cost of this computation.

Vectorization and Cython

1. Executing a Python Function
   - a motivating example
   - timing the execution
2. Speeding Up Python
   - vectorization
   - Cython

Timing the Execution

We distinguish between small and large computations.

- For small computations, in the order of milliseconds, the timings should execute a loop of several runs.
- For large computations:
  1. mark the start of the computations,
  2. do the computation, and
  3. compute the elapsed CPU time.

Vectorization and Cython

1. Executing a Python Function
   - a motivating example
   - timing the execution

2. Speeding Up Python
   - vectorization
   - Cython

Vectorization

Lists are versatile data structures, but arrays (vectors)

1. allow for compact storage, as each item is of the same type,
2. can be accessed with fast index arithmetic.

Vectorization replaces arithmetical operations on vectors with functions that take vectors as arguments.

Applied to $\sum_{k=1}^{n} \sqrt{1 - \left(\frac{k}{n}\right)^2}$ we do the following:

1. Make a vector x of equidistant points in $[0, 1]$.
2. Apply the function $\sqrt{\ }$ to the entire vector x.
3. Apply the function `sum` to $\sqrt{x}$.

The functions on vectors are *optimized* and *compiled*.

Vectorization and Cython

1. Executing a Python Function
   - a motivating example
   - timing the execution

2. Speeding Up Python
   - vectorization
   - Cython

Cython

Cython is a variant of Python:

1. It adds type declarations.
2. Calls functions directly from the C standard library.
3. Is compiled.

Compared to vectorization,

- cythonizing code leaves the original control structure intact,
- while vectorizing often requires a significant reformulation.