pyAT: AT but for Python

Will Rogers
Diamond Light Source

Disclaimer 1:
I am more software engineer than physicist

Disclaimer 2:
I'm new to AT

Why Python?

- Python is great
- We do lots of things in Python at DLS
- Matlab can be a pain for writing applications

python™

Why AT?

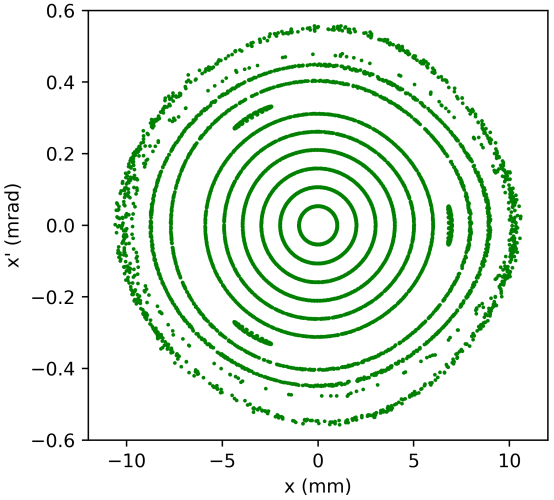
- I couldn't find an accelerator model that I could call from Python
- Boaz seemed keen
- DLS accelerator physicists like AT
- I could see a way to get started

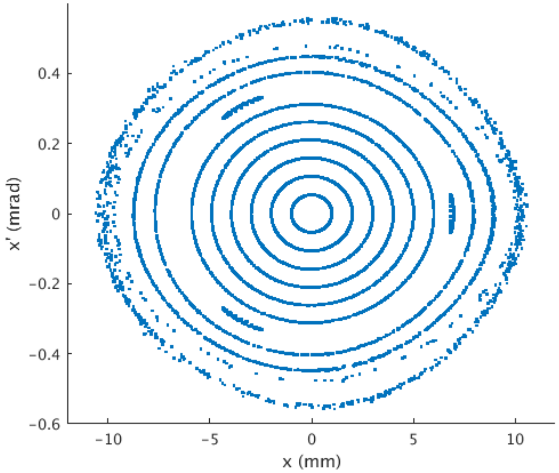
How?

- The AT core is written in C
- The physics was already separated from the Matlab-specific stuff
- We wrote the equivalent Python-specific stuff
  - Numpy for arrays
- It compiles on Linux, Mac, Windows
- Supports Python 2 and 3
- Laurent Farvacque, Nicola, and Boaz did a lot of work

Does it work?

- Yes – the numbers are identical since the underlying code is the same
- We've got a bunch of tests making sure we don't break things in future
- We can load lattices from .mat files





Is it done?

- No
- We have a tracking engine – C code
- In AT there are many physics functions – Matlab code
- We need a subset of these physics functions
- We can test to make sure they give the same answers

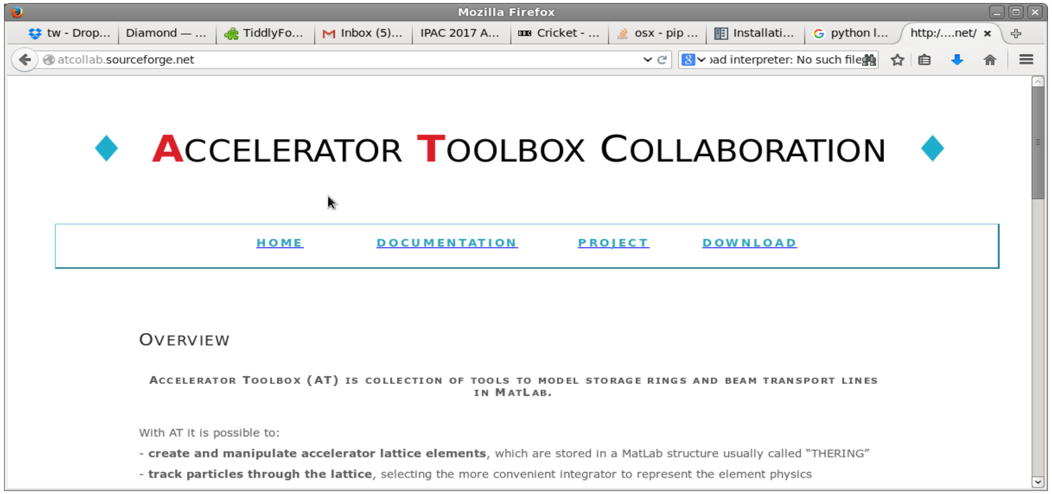
Distribution

- Upload to PyPI?
  - https://pypi.python.org/pypi
  - `pip install pyat`
  - (actually not pyat)
- Anaconda?
  - `conda install pyat`

ANACONDA®

Collaboration

- The ESRF people have been great
  - No forks, no conflicts
  - Put in lots of their time to help
  - Thanks!
- http://atcollab.sourceforge.net/



Accelerator Toolbox Collaboration

Overview

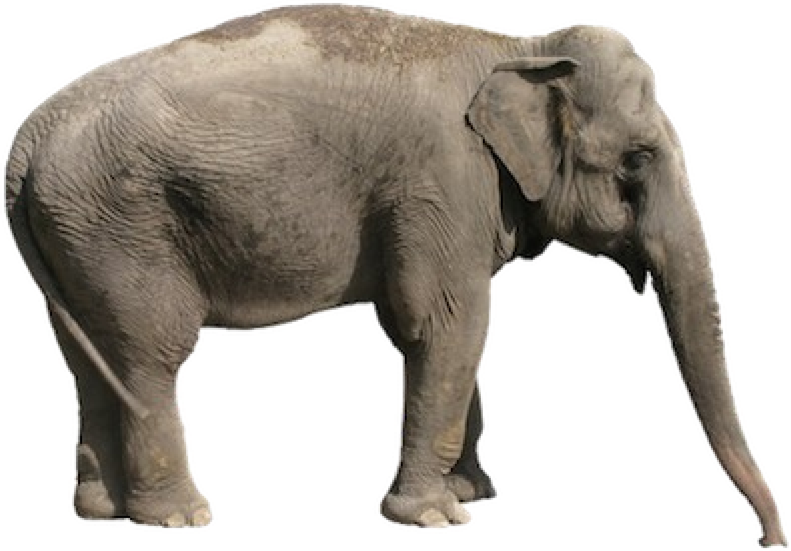
Accelerator Toolbox (AT) is collection of tools to model storage rings and beam transport lines in MatLab.

With AT it is possible to:

- **create and manipulate accelerator lattice elements**, which are stored in a MatLab structure usually called "THERING"
- **track particles through the lattice**, selecting the more convenient integrator to represent the element physics

Git ... ?

- Subversion is a hassle
- Git is more complicated
  - But you never want to go back
- https://github.com/willrogers/at
  - Also Github is great

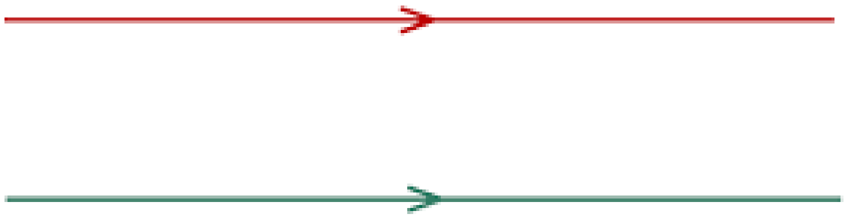


Licences

- For discussion?

Parallelisation

- You should be able to parallelise tracking on multiple particles using the `multiprocessing` module



MML / PML

- I have a student working on a basic MML-style application
- Useful for global machine software e.g. feedback systems
- pyAT could provide an on-line model

Anything else?