The University of Manchester

PyZgoubi Tutorial

Sam Tygier
University of Manchester
Cockcroft Accelerator Group

2016-09-05

The Cockcroft Institute of Accelerator Science and Technology

Science & Technology Facilities Council

Zgoubi

- Zgoubi is a well establish tracking code
  - Long history, active development
  - Widely used for FFAGs
- Accurate stepwise raytracing
- Wide range of magnet models
  - Straight, sectors
  - Scaling, multipoles, maps
  - Edge angles, fringe fields

PyZgoubi

- Originally developed as an interface to Zgoubi
  - Replaces the Zgoubi text input format with a scripting language (Python)
  - Maths in the input file
  - Named variables
  - Also loops, branching, IO, etc.
- Analysis tools
  - Functions to read Zgoubi's output into an array
- Has grown into a design framework
  - Tools for finding properties
  - Parameter scanning
  - Optimisation
  - Graphics

Python

- Python is general purpose scripting language
- If you have programmed in any language it should look familiar
- Worth looking at `https://docs.python.org/2.7/tutorial/index.html`
- Main 'gotcha', uses indentation instead of parenthesis '{}'
- NumPy and SciPy give useful library of array and maths features
- Still using Python 2

Installation

- Requirements
  - Need Python 2.5-2.7, Numpy, SciPy, Matplotlib
  - For working with dev versions, GCC/GFortran, SVN, BZR
- New: install with pip (PyPI)

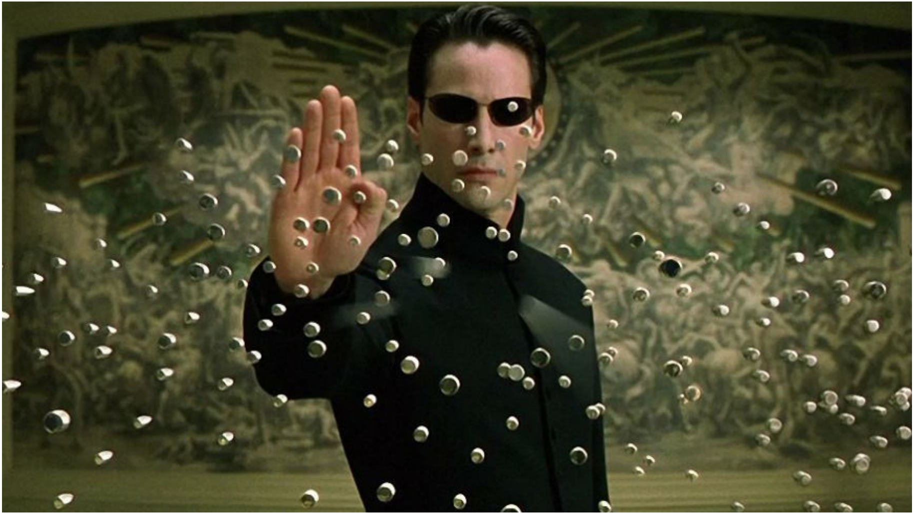
```
pip install --user pyzgoubi
```

- Or:
  - Download `http://www.hep.manchester.ac.uk/u/samt/pyzgoubi/`
  - Install instructions `http://www.hep.manchester.ac.uk/u/samt/pyzgoubi/doc/trunk/install.html`
- Documentation
  `http://www.hep.manchester.ac.uk/u/samt/pyzgoubi/doc/`

Translation

```
'QUADRUPO' foc
0
5.8782 3.7 -2.47715
0.0 0.0
0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0
0 0.0 0.0 0.0 0.0 0.0 0.0
1
1 0.0 0.0 0.0
```


ENTER



Translation

```
'QUADRUPO' foc
0
5.8782 3.7 -2.47715
0.0 0.0
0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0
0 0.0 0.0 0.0 0.0 0.0 0.0
1
1 0.0 0.0 0.0
```

```
qf = QUADRUPO('foc', XL=5.8782,
     R_0=3.7, B_0=-2.47715,
     XPAS=1, KPOS=1)
```

Translation

```
'QUADRUPO' foc
0
5.8782 3.7 -2.47715
0.0 0.0
0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0
0 0.0 0.0 0.0 0.0 0.0 0.0
1
1 0.0 0.0 0.0
```

```
fl = 58.782 *mm
fr = 37 * mm
fb = -6.695 * fr * T

qf = QUADRUPO('foc', XL=fl/cm,
     R_0=fr/cm, B_0=fb/kgauss,
     XPAS=1, KPOS=1)
```

- Tip: Always check units

Elements

- Each element in Zgoubi is a class in PyZgoubi
- Can manipulate elements
  - `qf.set(XL = 99)`
- Can generate the zgoubi .dat format
  - `print qf.output()`
- To see a list of elements implemented in PyZgoubi
  - `pyzgoubi --help elements`
- To see the paramenters of a given element
  - `pyzgoubi --help QUADRUPO`

Lines

- A Line holds all the elements that make up a lattice

```
fodo = Line("My fodo")
fodo.add(qf)
fodo.add(d1)
or
fodo.add(qf, d1, qd, qd, d1, qf)
```

- Can also hold control commands
- Can be nested

```
straight1.add(ms, fodo*4, -ms)
ring.add(arc, straight1, arc, striaght2)
```

Zgoubi style PyZgoubi

- Build a Line containing all required control commands
  - OBJET to create the beam
  - PARTICUL to set particle type (can use shortcuts ELECTRON, PROTON)
  - FAISCNL to beam store to fai file
  - REBELOTE to loop
- Run the line
  - `result = emma.run()`
- Display or save output files (plt, fai, res)
  - `print result.res()`
  - `result.save_fai("emma1.fai")`

Example 1 - emma

```
emma = Line('emma')
xpas = (10,10,10)

cells = 42
angle = 360/cells
d_offset = -34.048 * mm
f_offset = -7.514 * mm

 #lengths
ld = 210 * mm
sd = 50 * mm
fq = 58.782 * mm
dq = 75.699 * mm

 # quad radius
fr = 37 * mm
dr = 53 * mm

 #field
fb = -6.695 * fr * T
db = 4.704 * dr * T
```

```
ob = OBJET2()
rigidity = ke_to_rigidity(10e6, 0.51099892e6)
ob.set(BORO=-rigidity)
ob.add(Y=0.456, T=-38.1, D=1)
emma.add(ob)
emma.add(ELECTRON())
emma.add(DRIFT('ld', XL=ld/cm/2))
emma.add(CHANGREF(ALE=angle))
emma.add(CHANGREF(YCE=d_offset/cm))
emma.add(QUADRUPO('defoc', XL=dq/cm, R_0=dr/cm,
                B_0=db/kgauss, XPAS=xpas, KPOS=1))
emma.add(CHANGREF(YCE=-d_offset/cm))
emma.add(DRIFT('sd', XL=sd/cm))
emma.add(CHANGREF(YCE=f_offset/cm))
emma.add(QUADRUPO('foc', XL=fq/cm, R_0=fr/cm,
                B_0=fb/kgauss, XPAS=xpas, KPOS=1))
emma.add(CHANGREF(YCE=-f_offset/cm))
emma.add(DRIFT('ld', XL=ld/cm/2))
emma.add(FAISCNL(FNAME='zgoubi.fai'))
emma.add(REBELOTE(K=99, NPASS=10))
emma.add(FAISCEAU())
emma.add(END())
```

Example 1 - emma

```
print emma.output()
result = emma.run()
print result.res()
result.save_fai("emma1.fai")
```

- to run
  - `pyzgoubi example1_emma.py`
- Will show the Zgoubi input file
- Run Zgoubi
- Show the .res file
- Save the fai file

Example 2 - emma

- Now lets take some more advantage of PyZgoubi
- Make a line with just magnets

```
emma_cell = Line('emma')
emma_cell.add(DRIFT('ld', XL=ld/cm/2))
emma_cell.add(CHANGREF(ALE=angle))
...
```

- And a separate line to do the work

```
emma_line = Line("emma line")
ob = OBJET2()
rigidity = ke_to_rigidity(10e6, 0.51099892e6)
ob.set(BORO=-rigidity)
ob.add(Y=0.456, T=-38.1, D=1)
emma_line.add(ob)
emma_line.add(ELECTRON())
emma_line.add(emma_cell)
emma_line.add(FAISCNL(FNAME='zgoubi.fai'))
emma_line.add(REBELOTE(K=99, NPASS=100))
emma_line.add(END())
```

Example 2 - emma

- Now can run the line
- Use matplotlib to plot phase space

```
result = emma_line.run()
ftrack = result.get_all("fai")
print ftrack['Y']

pyplot.plot(ftrack['Y'], ftrack['T'], "b.")
pyplot.xlabel("Y (cm)")
pyplot.ylabel("T (mrad)")
pyplot.savefig("emma_phase_space.pdf")
```

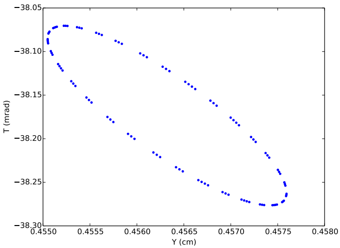
Example 2 - emma

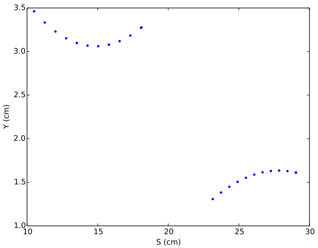
- Can reuse the line in the same script
- Turn on tracking (IL=2) and plot track

```
emma_line2 = Line("emma line2")
emma.full_tracking(True)
emma_line2.add(ob)
emma_line2.add(ELECTRON())
emma_line2.add(emma_cell)
emma_line2.add(END())


result = emma_line2.run()
ftrack = result.get_all("plt")
pyplot.clf()

pyplot.plot(ftrack['S'], ftrack['Y'], "b.")
pyplot.xlabel("S (cm)")
pyplot.ylabel("Y (cm)")
pyplot.savefig("emma_track.pdf")
```

Example 2 - emma

- Run
  - `pyzgoubi example2_emma.py`
- Will create 2 output files





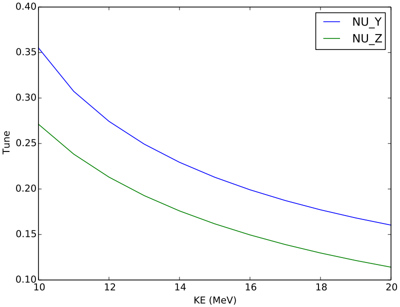
Cells

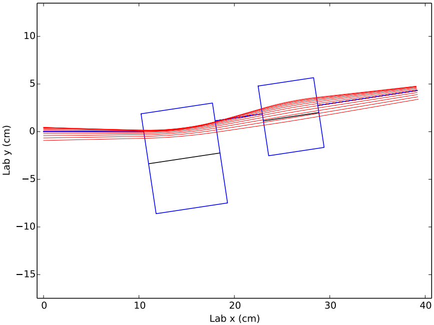
- For a lot of work we have a periodic cell and want to know
  - Closed orbit (`Y, T, Z, P`)
  - Time of flight (`tof`)
  - Tunes (`NU_Y, NU_Z`)
  - Max fields (`MAX_BY, MIN_BY, MAX_BZ, MIN_BZ`)
  - Transfer matrix (`matrix`)
  - DA (`DA`)
  - All for a range of energy or other parameters
- Developed GCP module (get cell properties) for these common tasks
  - No longer have to worry about which OBJET mode and which FAISCNL or MATRIX commands

Example 3 - emma

```
data = gcp.get_cell_properties(cell=emma_cell,
       min_ke=10e6, max_ke=20e6, ke_steps=11, particle='e')
print gcp.cell_properties_table(data, ["KE", "stable", "Y", "T", "NU_Y", "NU_Z"])

pyplot.plot(data['KE']/1e6, data['NU_Y'], label="NU_Y")
pyplot.plot(data['KE']/1e6, data['NU_Z'], label="NU_Z")
pyplot.xlabel("KE (MeV)")
pyplot.ylabel("Tune")
pyplot.legend()
pyplot.savefig("emma_tune.pdf")

gcp.plot_cell_tracks(cell=emma_cell, data=data, particle='e',
     output_file='emma_tracks.pdf')
```

Example 3 - emma





Example 4 - KURRI 150 MeV

- We have the Zgoubi input for the KURRI 105 MeV FFAG
- If I want to quickly get it into PyZgoubi I can use a fake element
- A bit messy but can get quick results

```
kurri_dat = """ 'FFAG'
20
5 30. 540.
0. 0.  -.6  0.1
6.3  03.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
...

kurri_cell = Line("KURRI")
kurri_cell.add(FAKE_ELEM(kurri_dat))
```

Example 4 - KURRI 150 MeV

```
data = gcp.get_cell_properties(cell=kurri_cell, min_ke=11e6, max_ke=150e6,
    ke_steps=11, particle='p')
```

- Finds no stable orbits
- Can give a hint, where to start search

```
closed_orbit_init_YTZP=[440,0,0,0]
```

- Or can check a range of starting points

```
closed_orbit_range=[20,0,0,0], closed_orbit_range_count=[20,0,0,0]
```

Example 5 - KURRI 150 MeV

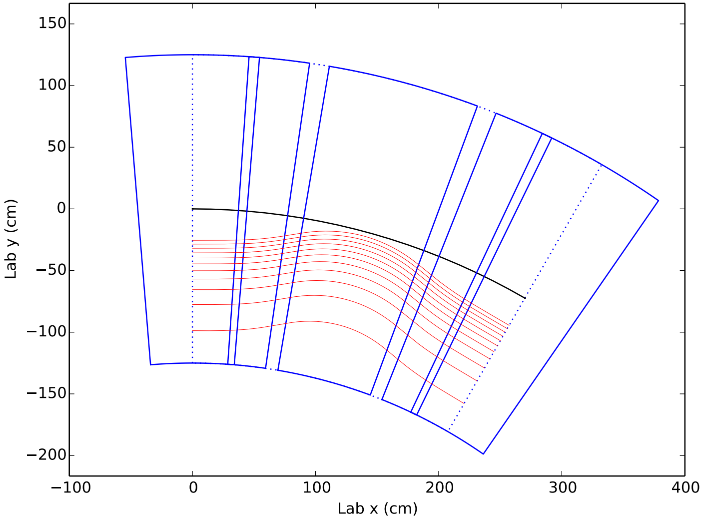
- Now can do the pyzgoubi version properly

```
kurri_mag = FFAG("kurri", IL=20,
                 N=5, AT=30, RM=540,
                 KIRD=0, RESOL=2,
                 XPAS=0.25,
                 KPOS=2,
                 RE=540, RS=540)

kurri_mag.add(ACN=0, DELTA_RM=0, BZ_0=-.6, K=0.1,
              G0_E=g0, KAPPA_E=kappa,
              NCE=nc, CE_0=c0, CE_1=c1, CE_2=c2, CE_3=c3,
              OMEGA_E=omega0_e,
              R1_E=big, U1_E=-big, U2_E=big, R2_E=big,
              G0_S=g0, KAPPA_S=kappa,
              NCS=nc, CS_0=c0, CS_1=c1, CS_2=c2, CS_3=c3,
              OMEGA_S=omega0_s,
              R1_S=big, U1_S=-big, U2_S=big, R2_S=big,
              KAPPA_L=-1)
...
```

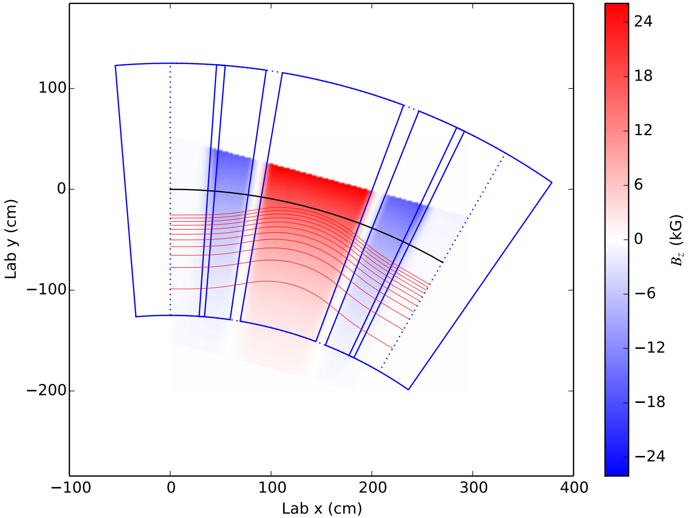
Example 5 - KURRI 150 MeV

```
data = gcp.get_cell_properties(cell=kurri_cell, min_ke=11e6, max_ke=150e6,
      ke_steps=11, particle='p', closed_orbit_init_YTZP=[-100,0,0,0],
      closed_orbit_range=[50,0,0,0], closed_orbit_range_count=[40,0,0,0])
print gcp.cell_properties_table(data, ["KE", "stable", "Y", "T", "NU_Y", "NU_Z"])

gcp.plot_cell_tracks(cell=kurri_cell, data=data, particle='p',
             output_file='kurri_cell.pdf', sector_width=200)

gcp.plot_cell_tracks(cell=kurri_cell, data=data, particle='p',
          output_file='kurri_cell_mag.pdf', sector_width=200,
        draw_field_midplane=True, min_y=-150, max_y=50, y_steps=100,
        angle=-radians(15)*1000)
```

Example 5 - KURRI 150 MeV



Example 5 - KURRI 150 MeV



Dynamic Aperture

- Lets do a quick DA simulation
- Only 10 passes (actually 10 cells)
- Quick mode, 1 particle

```
data = gcp.get_cell_properties(cell=kurri_cell, min_ke=11e6, max_ke=150e6,
        ke_steps=1, particle='p', closed_orbit_init_YTZP=[-100,0,0,0],
       closed_orbit_range=[50,0,0,0], closed_orbit_range_count=[40,0,0,0])
gcp.get_dynamic_aperture(cell=kurri_cell, data=data, particle='p', npass=10,
       nangles=3, tol=0.01, quick_mode="+y+z")

print "geo", data[0]['DA']*1e6, "mm mrad"
```

Further info

- Docs
- `http://www.hep.manchester.ac.uk/u/samt/pyzgoubi/doc/trunk/index.html`
- Email me
- sam.tygier@manchester.ac.uk

FIN

Learn French With Zgoubi

Aimant

Faisceau

Particule

Rebelote

Apprendre le Français avec Zgoubi

Objet

Champ de fuite

Photos from wikipedia