MANCHESTER 1824
The University of Manchester

nuSTORM: tracking and performance of an FFAG focussing ring

Sam Tygier, Rob Appleby, Jimmy Garland, Jaroslaw Pasternak, Jean-Baptiste Lagrange

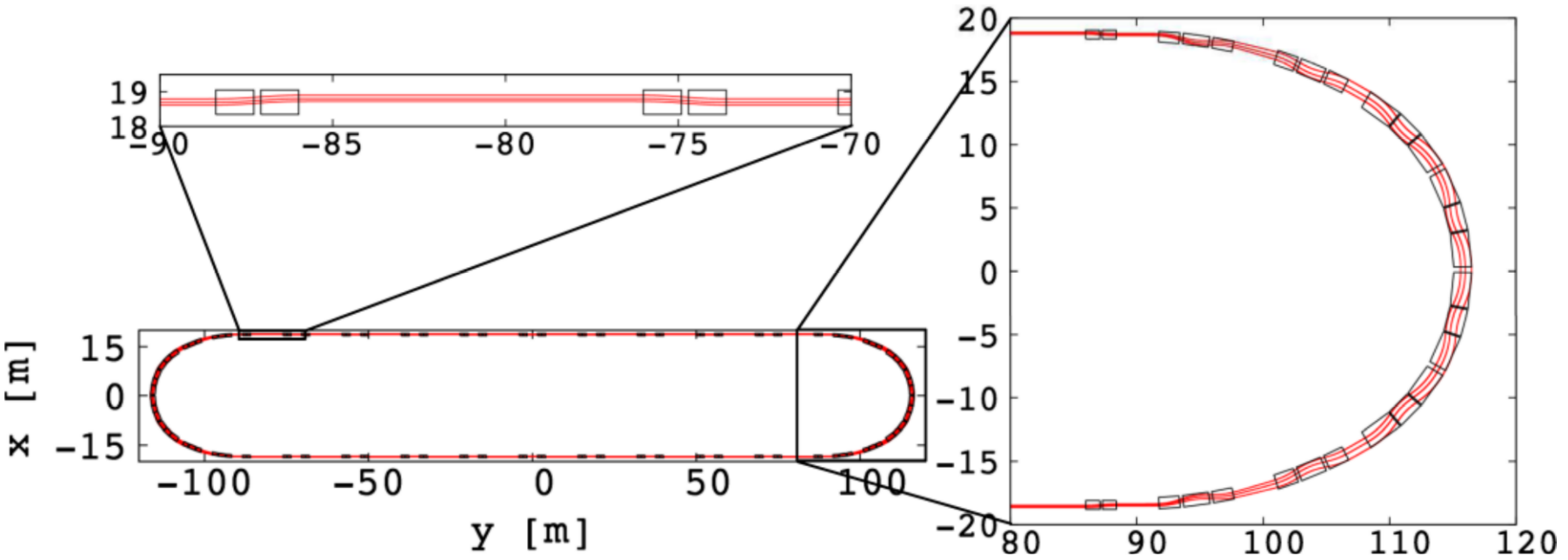
16 Feb 2017

The Cockcroft Institute of Accelerator Science and Technology

Science & Technology Facilities Council

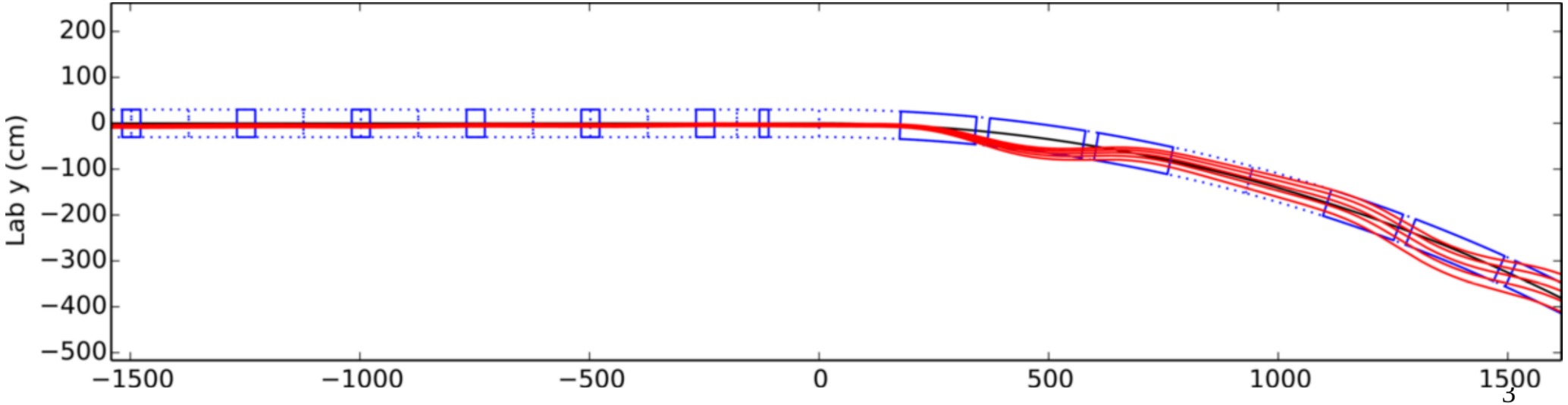
Overview

- nuSTORM
- Tracking codes
- Some results



NuSTORM FFAG lattice

- Novel FFAG lattice
- Wide energy acceptance
- Long straights with low scallop
- Race track with straight FFAG elements
- Matched in to scaling triplet arcs



Tracking Codes for FFAGs

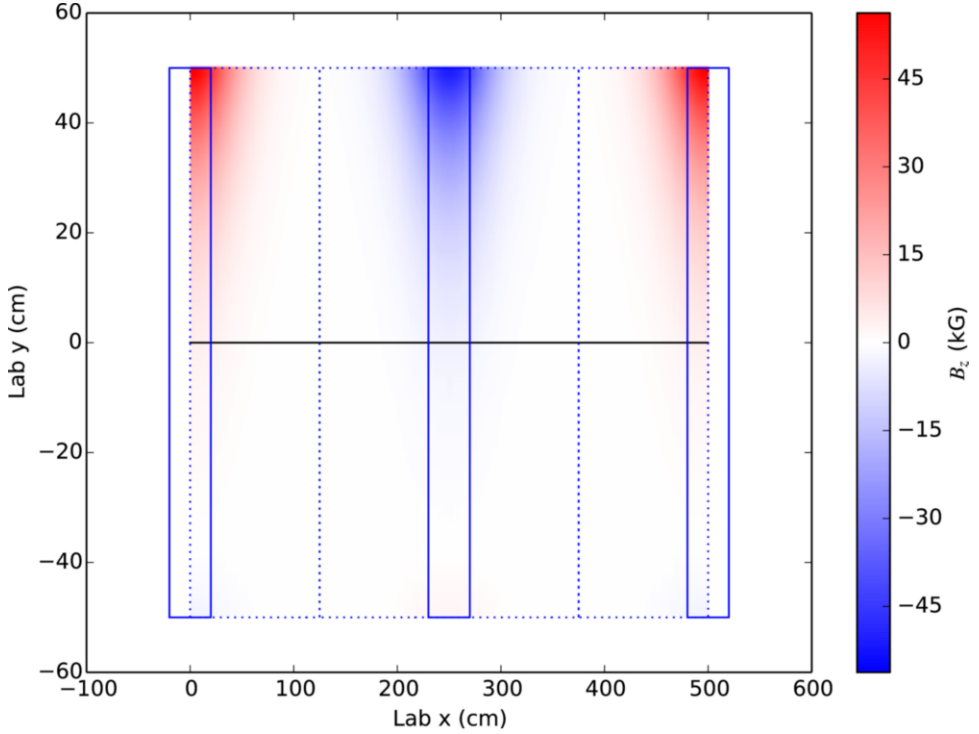
- FFAGs have big range of energies and trajectories through complex magnets
- Fringe fields very important
- Assumptions in most accelerator code make them unsuitable for FFAGs
- nuSTORM FFAG lattice was mostly studied using 2 codes
- JB's tracking code
  - Initial design, optimization, longterm tracking, acceptance, DA
- Zgoubi (pyZgoubi)
  - Validation, optimization, longterm tracking, DA

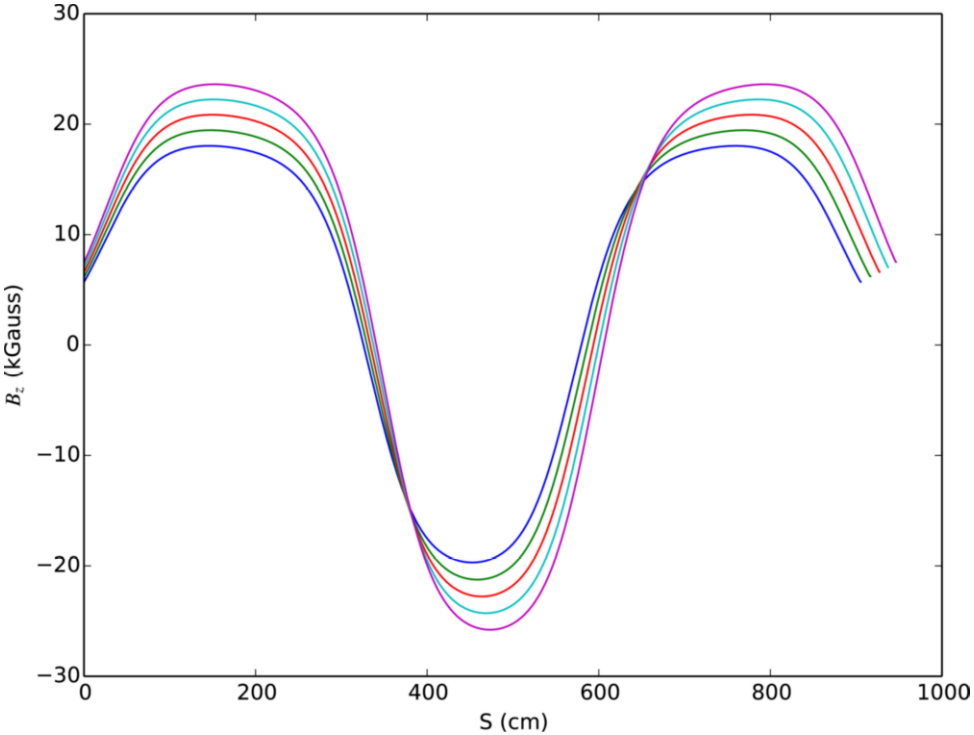
Tracking codes

- JB's Code
  - C++
  - Elements define fields
  - Stepwise tracker
    - Runge-Kutta
  - Supports all elements of lattice natively
- PyZgoubi
  - FORTRAN/Python
  - Elements define fields
  - Stepwise tracker
    - Taylor expanded lorentz force
  - Supports “normal” FFAGs
  - Widely used
  - Optimisation

nuSTORM modelling challenges

- Straight FFAG cell
- $B_y = B_0 e^{m(x-x_0)}$
- Multipole expansion in zgoubi
- Overlapping fringe fields
- Enge fringe falloff



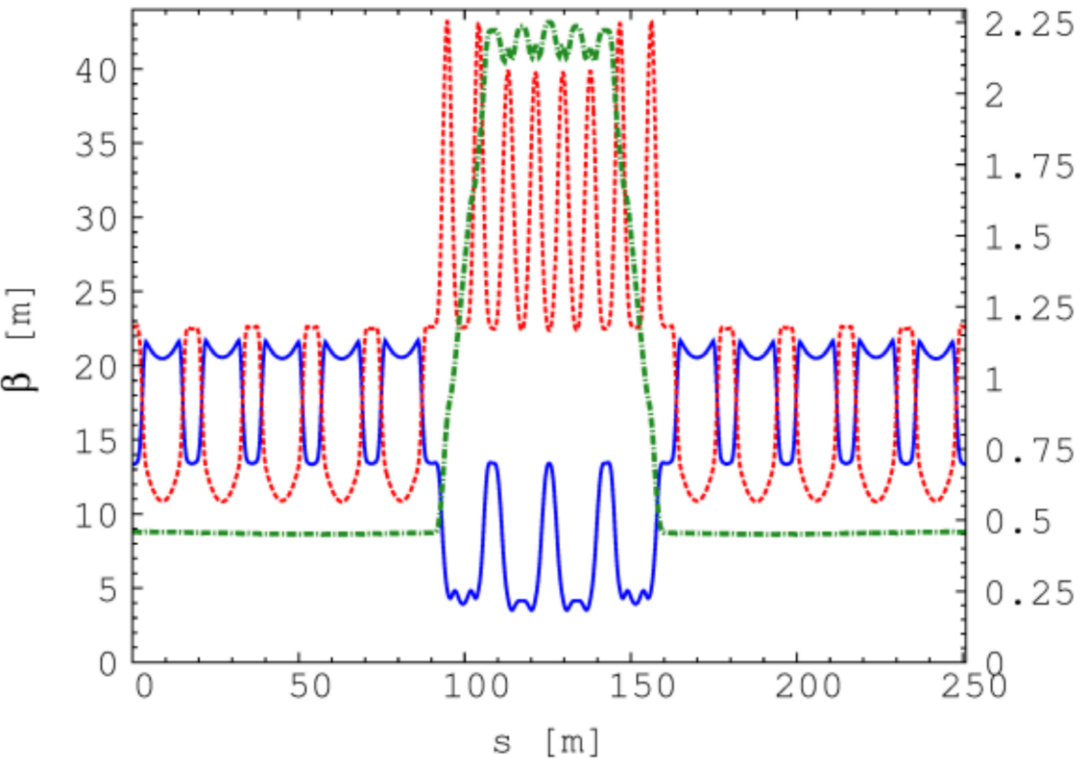


Validation with PyZgoubi

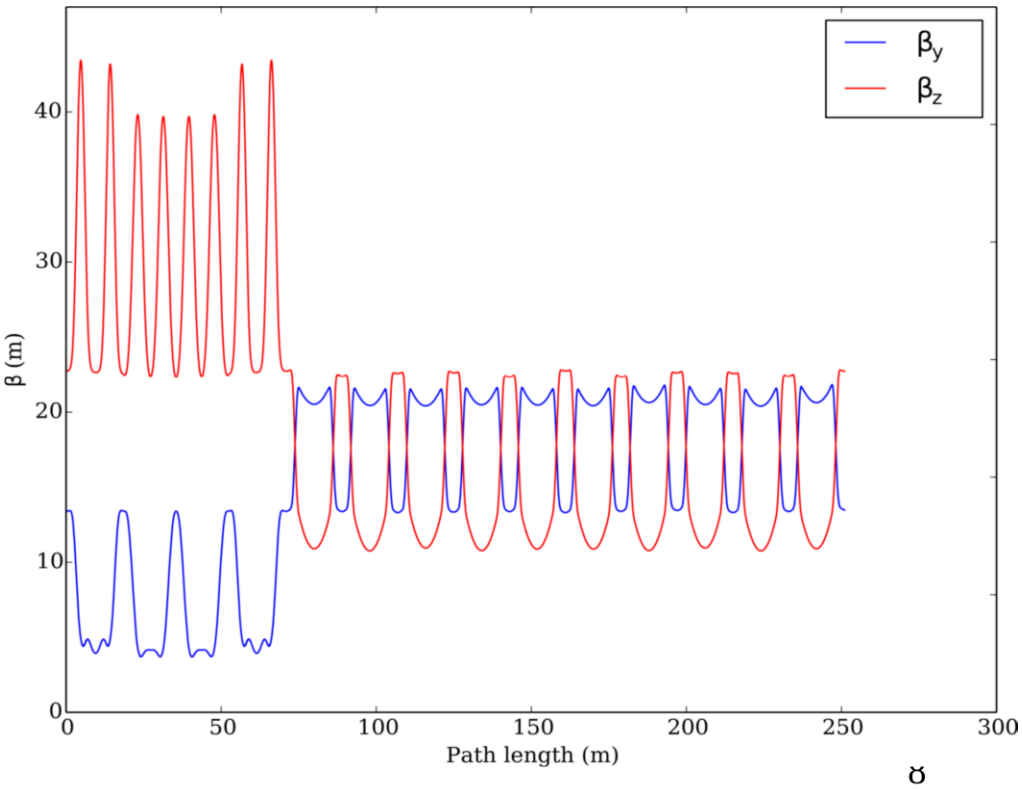
- Useful to validate a design in a second code
- Zgoubi is widely used for FFAGs (also synchrotrons, spectrometers, cyclotrons)
- Does not have a built in straight FFAG element, but it can be modelled using multipole expansion in the DIPOLES element
- PyZgoubi is python framework around Zgoubi
- PyZgoubi makes it easy to read lattice parameters from a text file and assemble a lattice
- Initial design work at ICL
- Validation at Manchester

Comparison

- Codes give good agreement
  - (once parameter communications were resolved)
- Both find C-S parameters by tracking



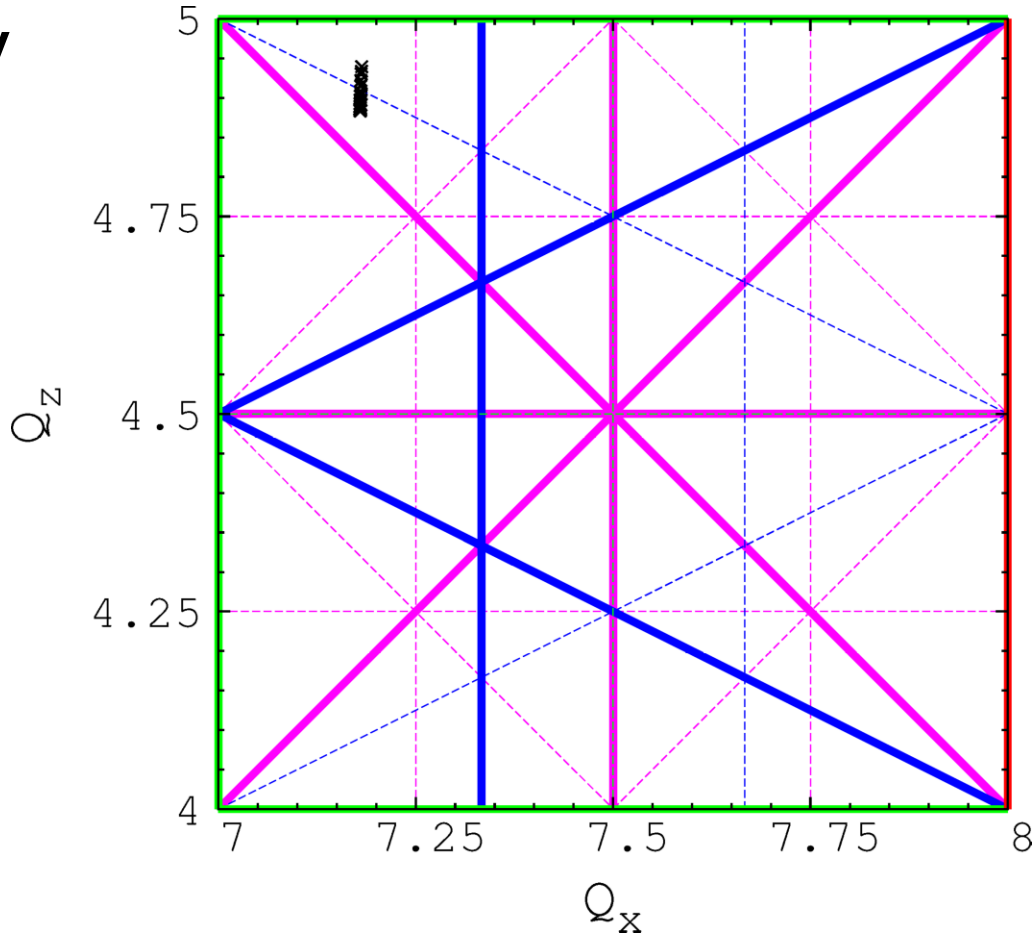
JB



pyZgoubi

Tune flatness

- For stability over large momentum range (±19%) need a low tune variation
- Scaling FFAG arcs and straight FFAG straights
- Keeps beam away from resonances
- Good agreement between codes
- Should give large dynamic aperture

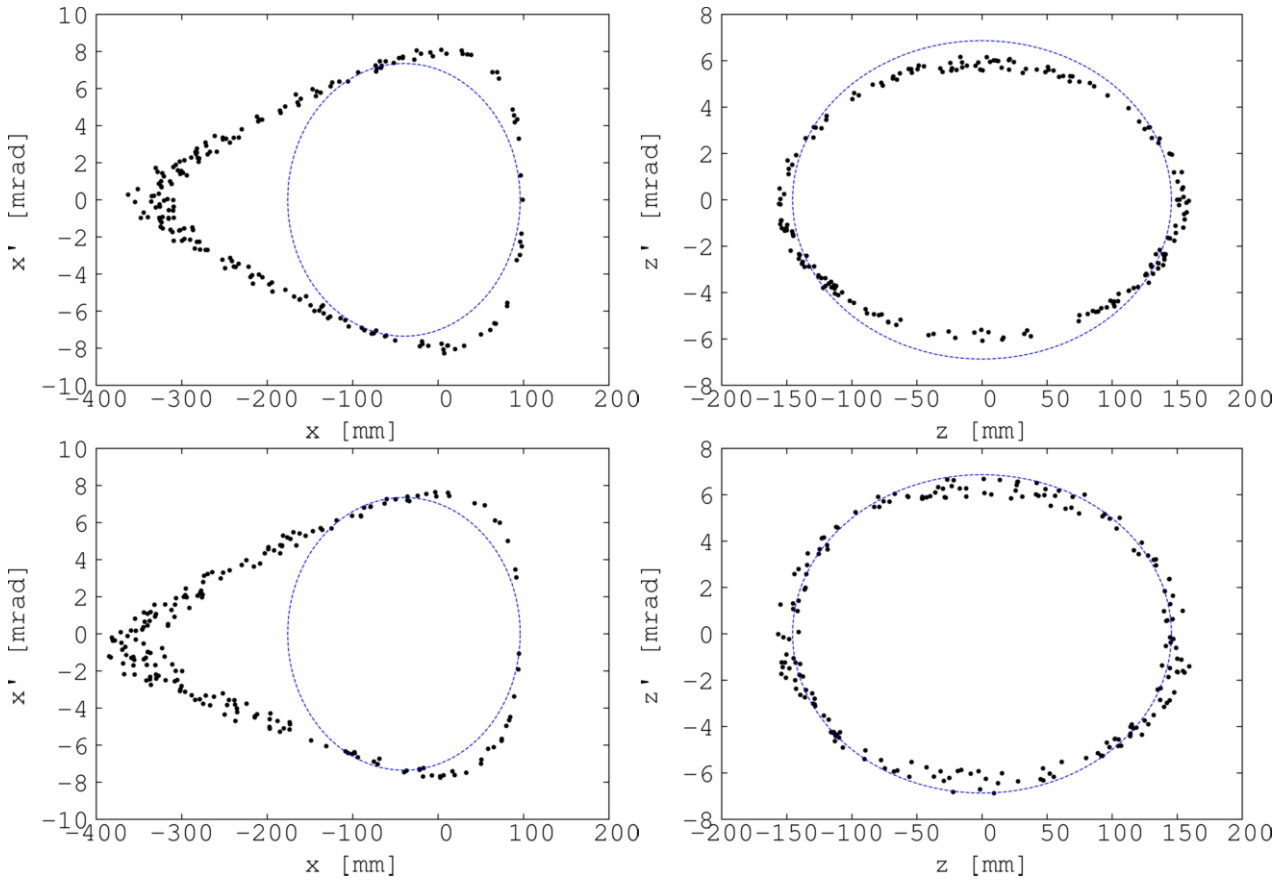


Long term tracking

- Muons have lifetime of order 100 turns
- Need a large DA in order to accept and hold large fraction of injected beam
- DA has been simulated in both codes with good agreement
  - This is a very tough test of tracking agreement

JB

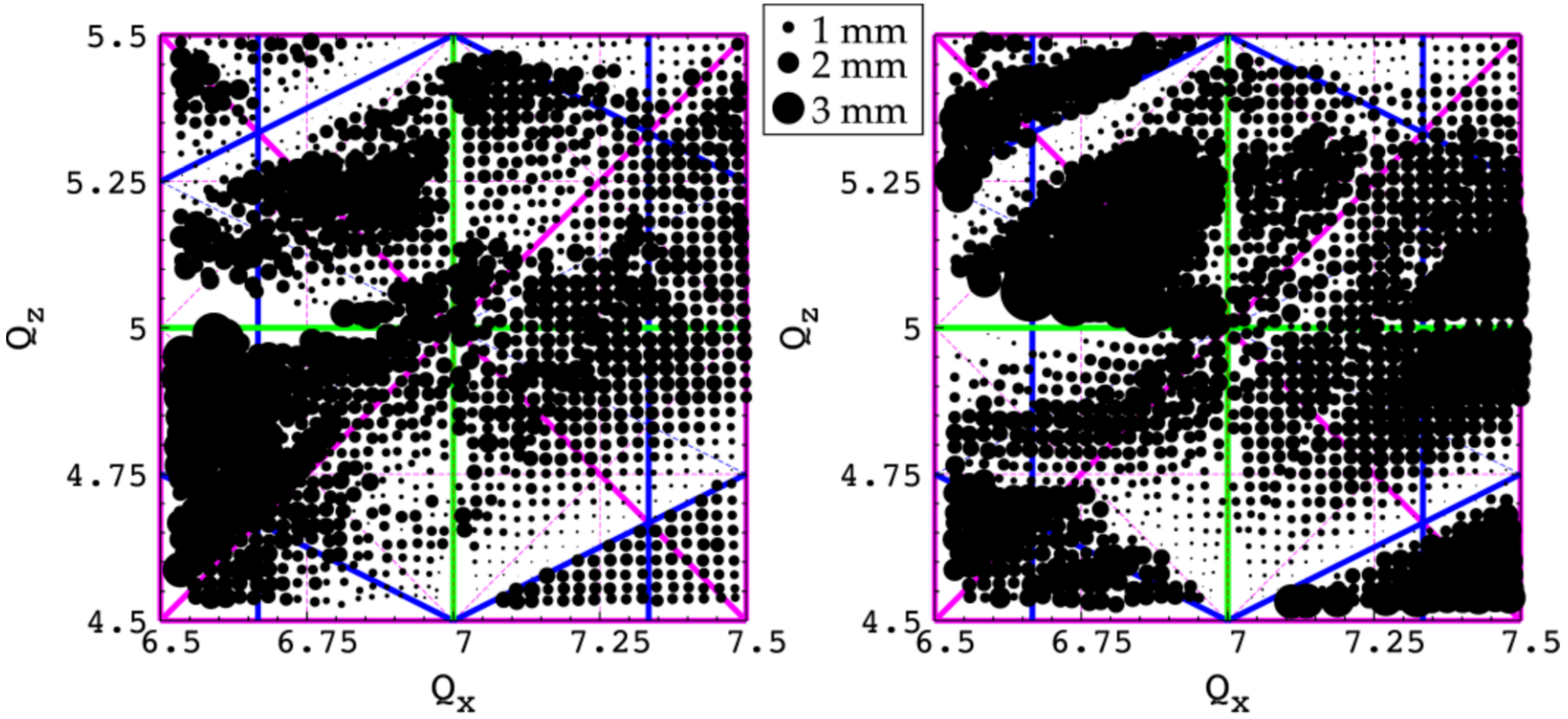
PyZgoubi



*Triplet lattice

DA optimisation

- DA is very sensitive to tune
- Can optimize working point to improve DA
- Can achieve DA of 2 mmrad H, 1 mmrad V



Conclusion

- FFAG nuSTORM lattice gives large longitudinal and transverse acceptance
- Lattice has been modelled in detail
  - Scaling and Straight FFAG magnets
  - Fringe fields and overlaps
- Good agreement
  - Lattice parameters
  - Tunes
  - Large dynamic acceptance confirmed in both codes

Extra slides

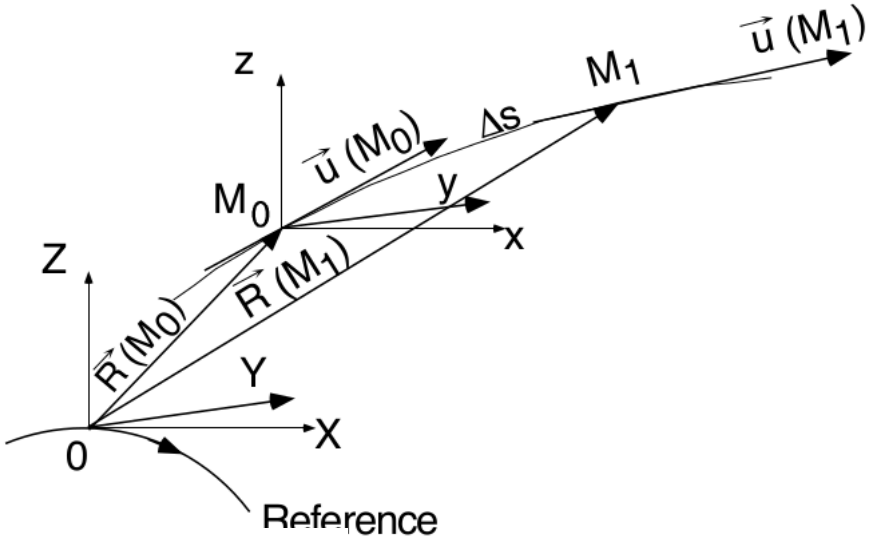
Zgoubi tracking

Lorentz force

$$\frac{d(m\vec{v})}{dt} = q\,(\vec{e} + \vec{v} \times \vec{b})$$

Split momentum and direction

$$(B\rho)'\,\vec{u} + B\rho\,\vec{u}' = \frac{\vec{e}}{v} + \vec{u} \times \vec{b}$$



Expand changes

$$\vec{R}(M_1) \approx \vec{R}(M_0) + \vec{u}(M_0)\,\Delta s + \vec{u}'(M_0)\,\frac{\Delta s^2}{2!} + \ldots + \vec{u}''''(M_0)\,\frac{\Delta s^6}{6!}$$

$$\vec{u}(M_1) \approx \vec{u}(M_0) + \vec{u}'(M_0)\,\Delta s + \vec{u}''(M_0)\,\frac{\Delta s^2}{2!} + \ldots + \vec{u}''''(M_0)\,\frac{\Delta s^5}{5!}$$

Derivatives of fields

$$\vec{u}' = \vec{u} \times \vec{B}$$

$$\vec{u}'' = \vec{u}' \times \vec{B} + \vec{u} \times \vec{B}'$$

$$\vec{u}''' = \vec{u}'' \times \vec{B} + 2\vec{u}' \times \vec{B}' + \vec{u} \times \vec{B}''$$

$$\vec{u}'''' = \vec{u}''' \times \vec{B} + 3\vec{u}'' \times \vec{B}' + 3\vec{u}' \times \vec{B}'' + \vec{u} \times \vec{B}'''$$

$$\vec{u}''''' = \vec{u}'''' \times \vec{B} + 4\vec{u}''' \times \vec{B}' + 6\vec{u}'' \times \vec{B}'' + 4\vec{u}' \times \vec{B}''' + \vec{u} \times \vec{B}''''$$

See zgoubi manual for full details