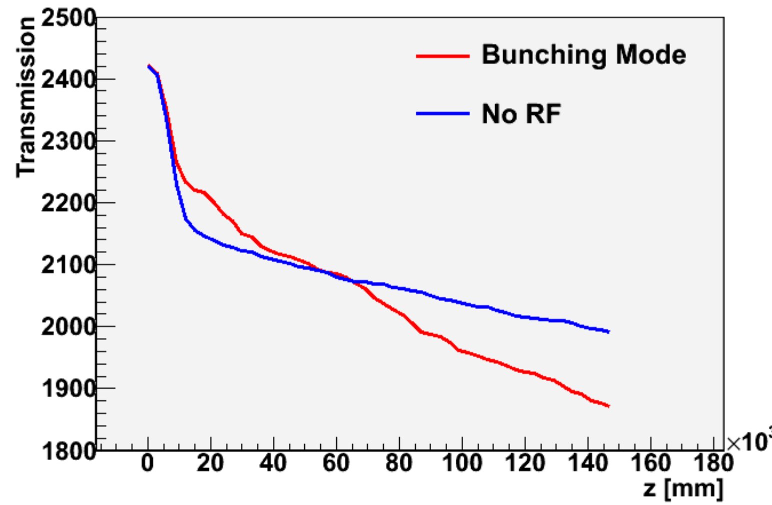
Non-Linear Solenoidal Optics

Chris Rogers,
ASTeC,
Rutherford Appleton Laboratory

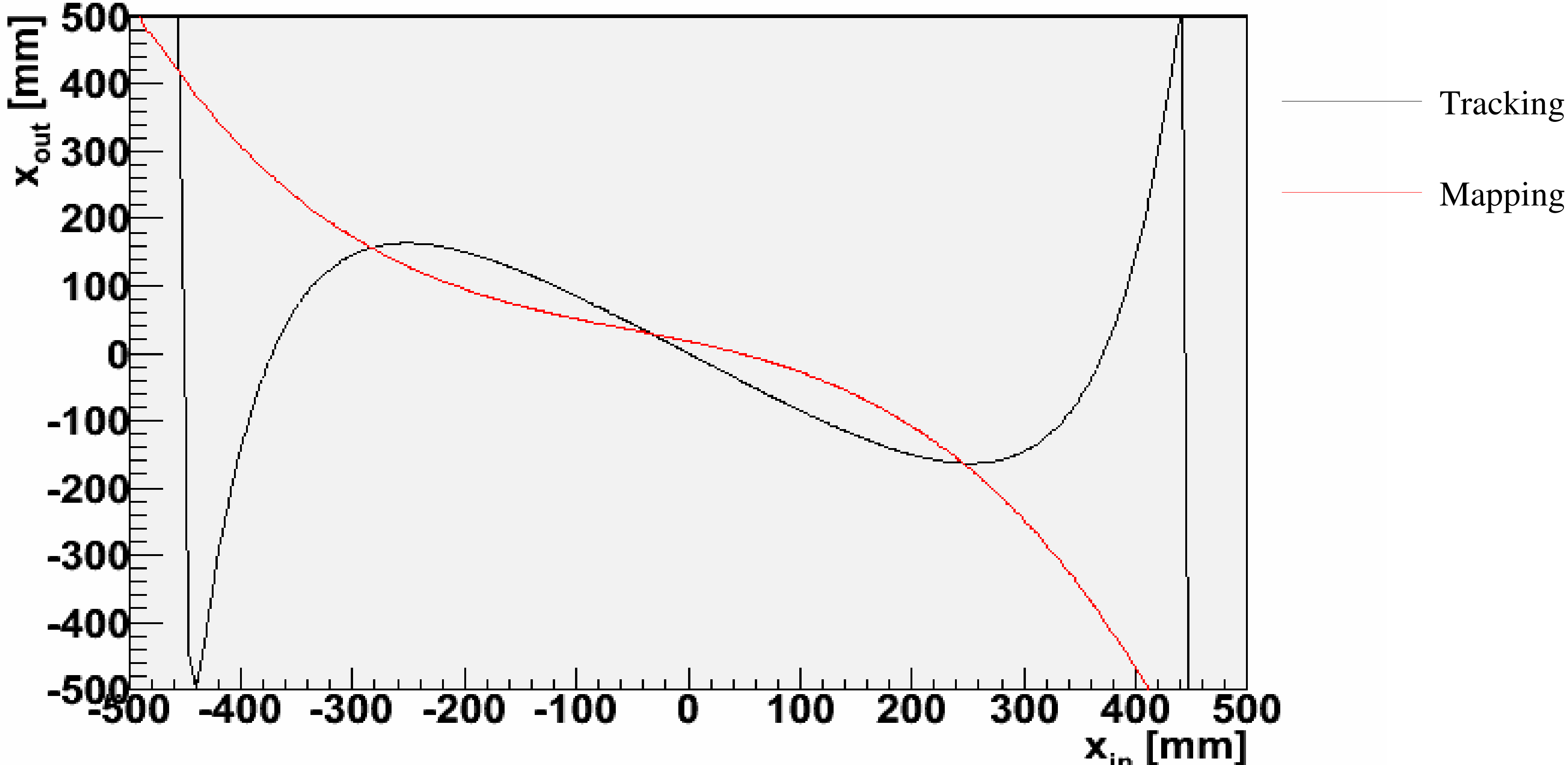
Emittance Growth and Beam Loss

- I get quite a bit of beam loss in my cooling channel
  - Significant even in the absence of absorbers
  - Exarcerbated by introduction of RF cavities
    - Nb “Bunching mode” here => 17.5 MV/m running at 0° phase
      - Want to look at optical losses in absence of absorber material
    - Rather larger bucket than in normal running
    - Expect these optics-induced losses to be worse in normal running



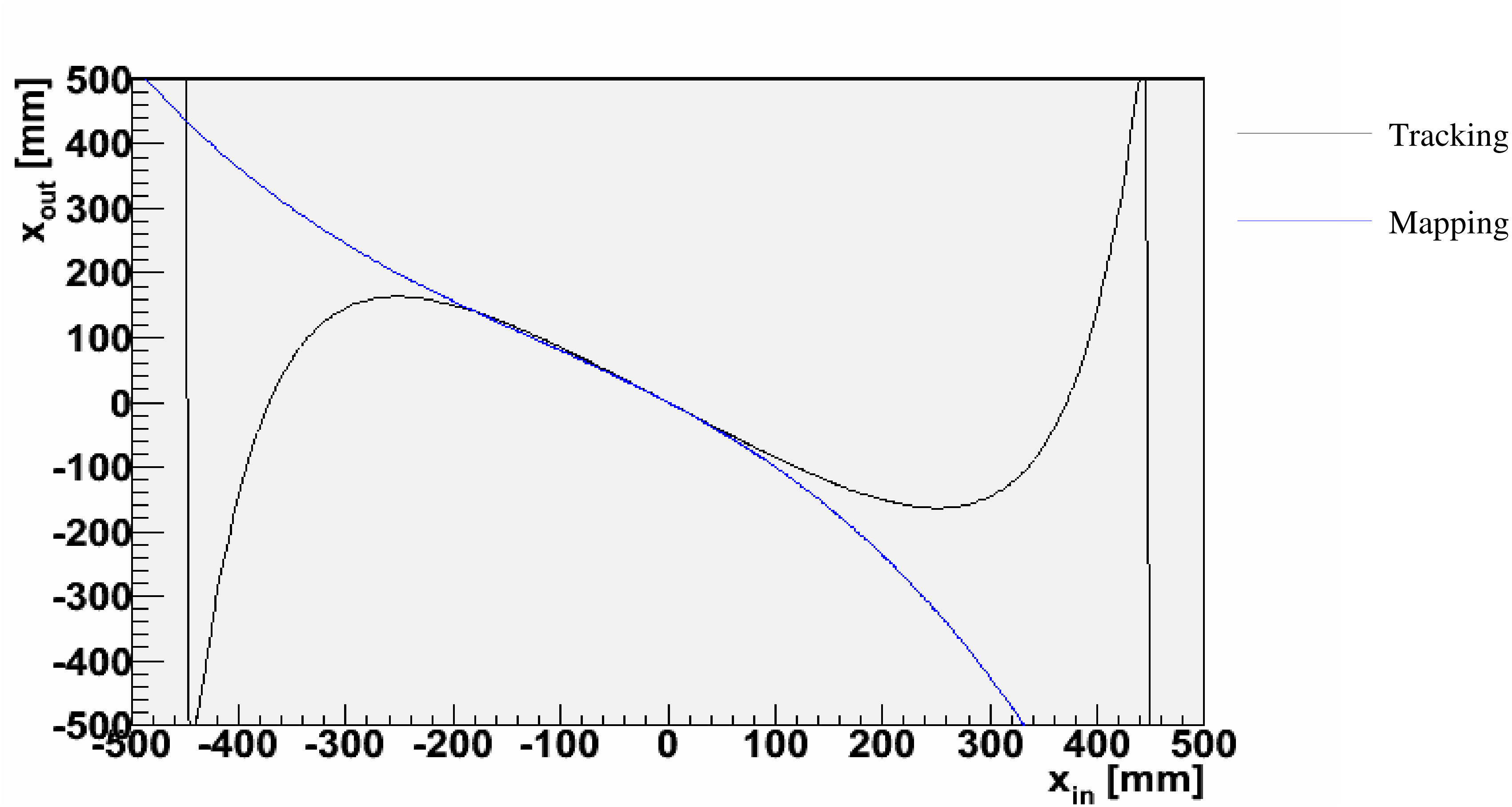
- Try using mapping technique to study optical heating
  - Expand (x,y,t;px,py,E) as a mapping across a cooling cell
  - Ignore material effects
- Focus in this talk on mapping techniques
  - Not so much about lattice development here

Linear Least Squares



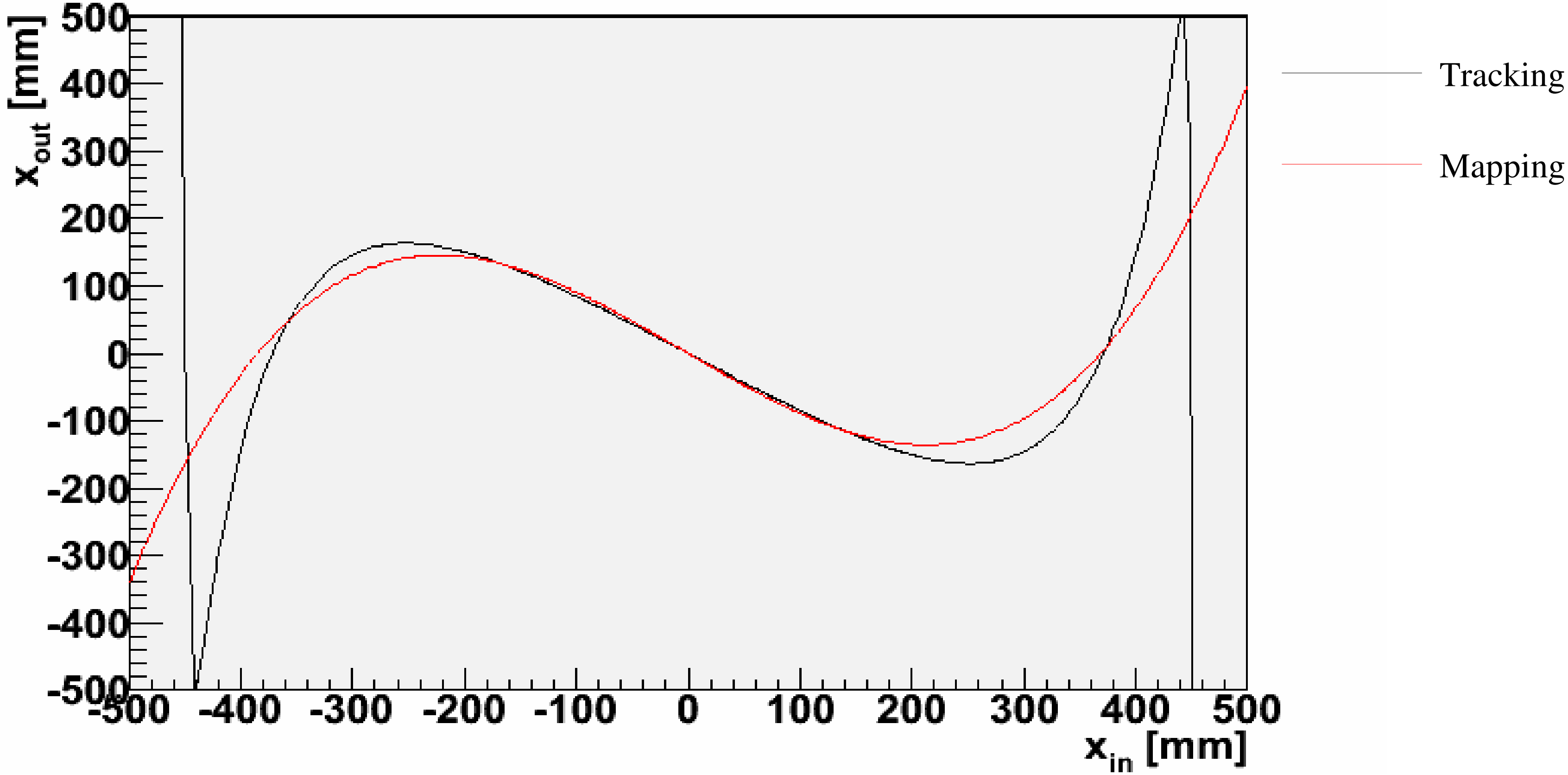
- First consider algorithms mapping 1D space
  - e.g. x_{in} -> x_{out} as a 4th order polynomial
- Consider Linear Least Squares fit of input particle data to output particle data after tracking through G4MICE

Linear Least Squares - Numerical Diff for 1st term



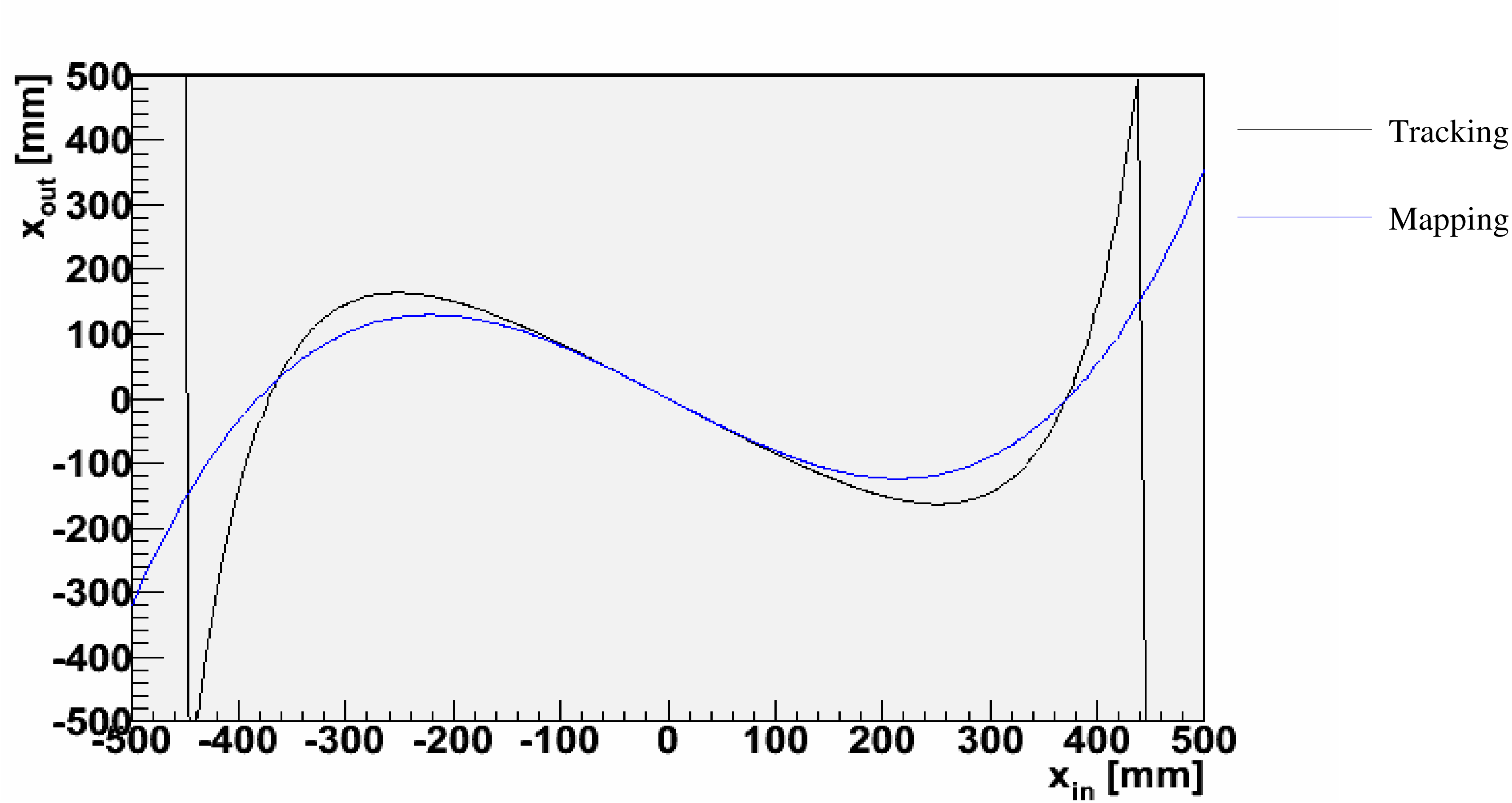
- Can improve the fit by numerically differentiating at x=0
  - Force the polynomial 1st derivative here

Linear Least Squares - Chi2 Cut



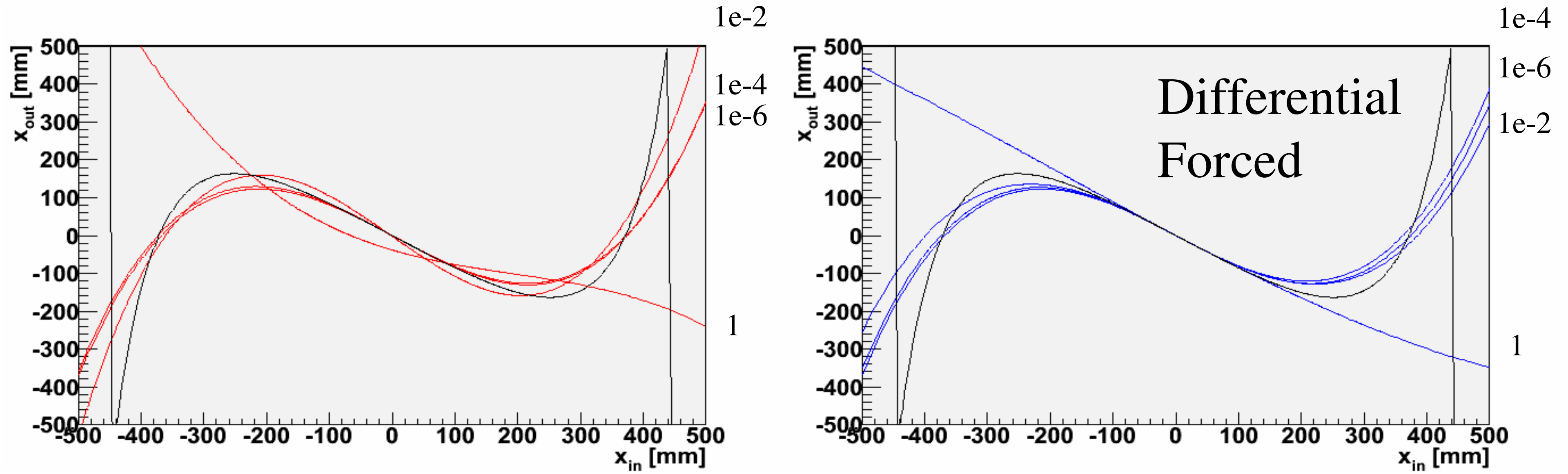
- Alternatively try applying a cut on particles with large amplitudes
  - Reduce the size of the amplitude acceptance until difference between fit and true data is small

Linear Least Squares - Numerical Diff for 1st term and Chi2 Cut



- Now force 1st derivative and take chi2 cut

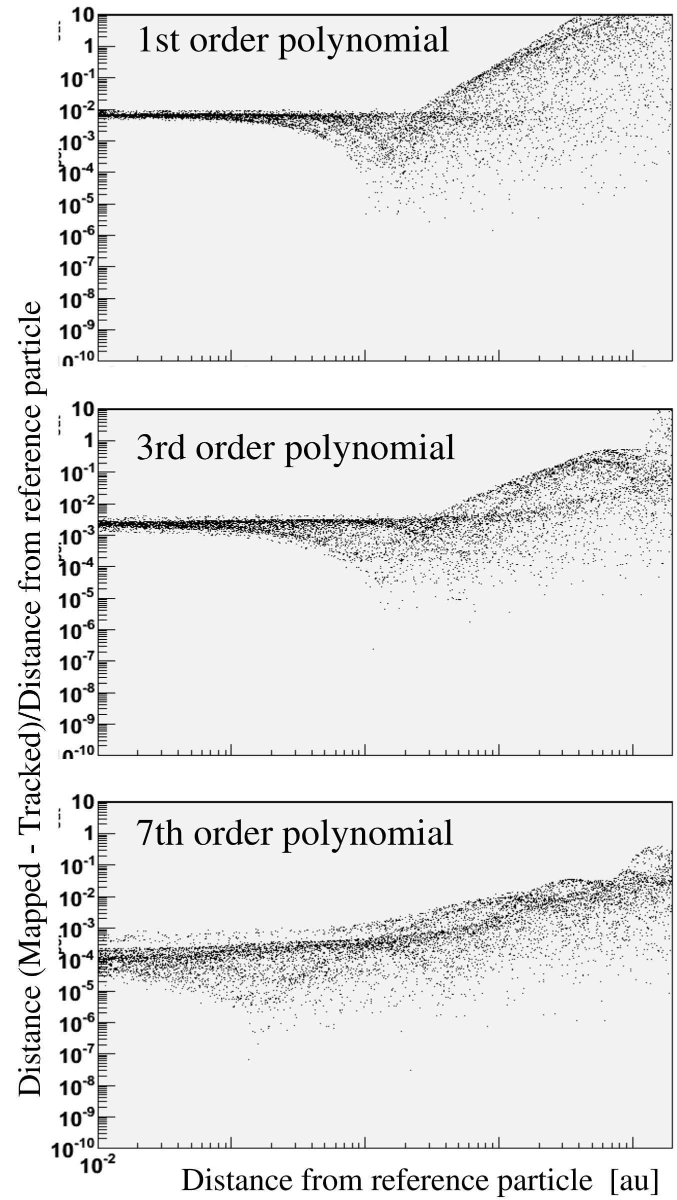
Linear Least Squares - Numerical Diff for 1st term and Chi2 Cut



- Convergence as a function of chi2 limit
  - Seems to converge reasonably well, better w/o numerical derivative
  - Prefer not to force the 1st order differential
    - Probably because error in numerical derivative

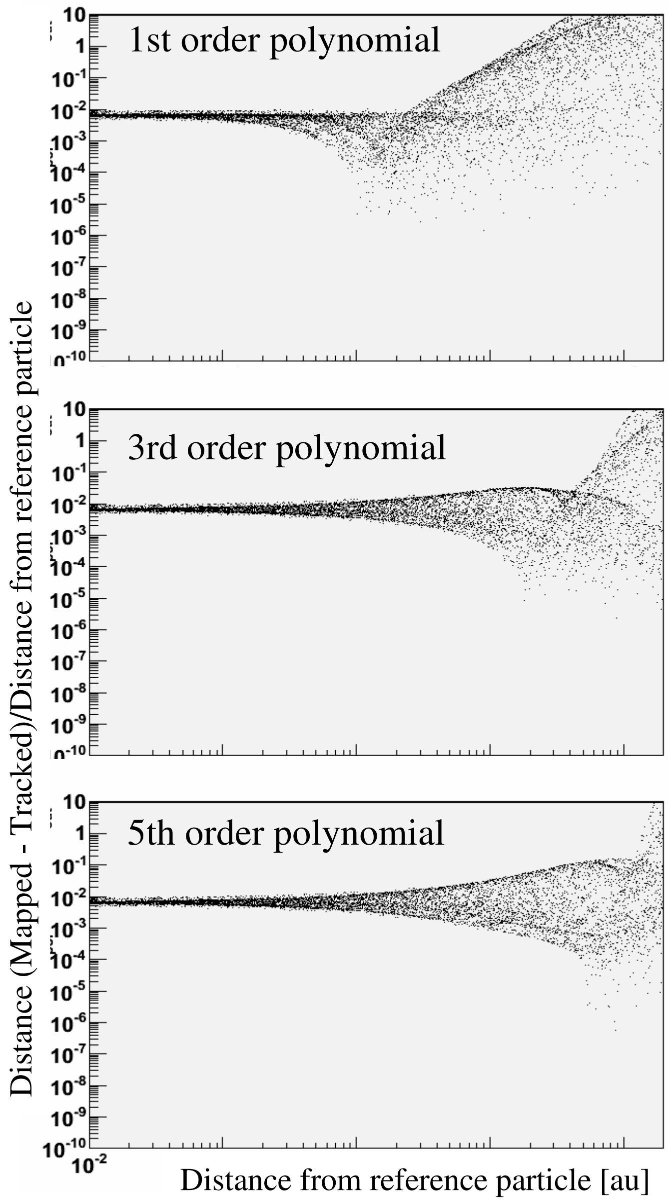
Longitudinal phase space

- Consider a “real” application - longitudinal phase space
  - Fire shells of particles at various amplitude
  - Look at (mapping - tracking) as a function of distance from the reference trajectory
  - In general, higher order polynomials => better fit
  - At some point adding extra terms doesn't really help
  - Using the trick of applying fit only in a region where the polynomial matches tracking results



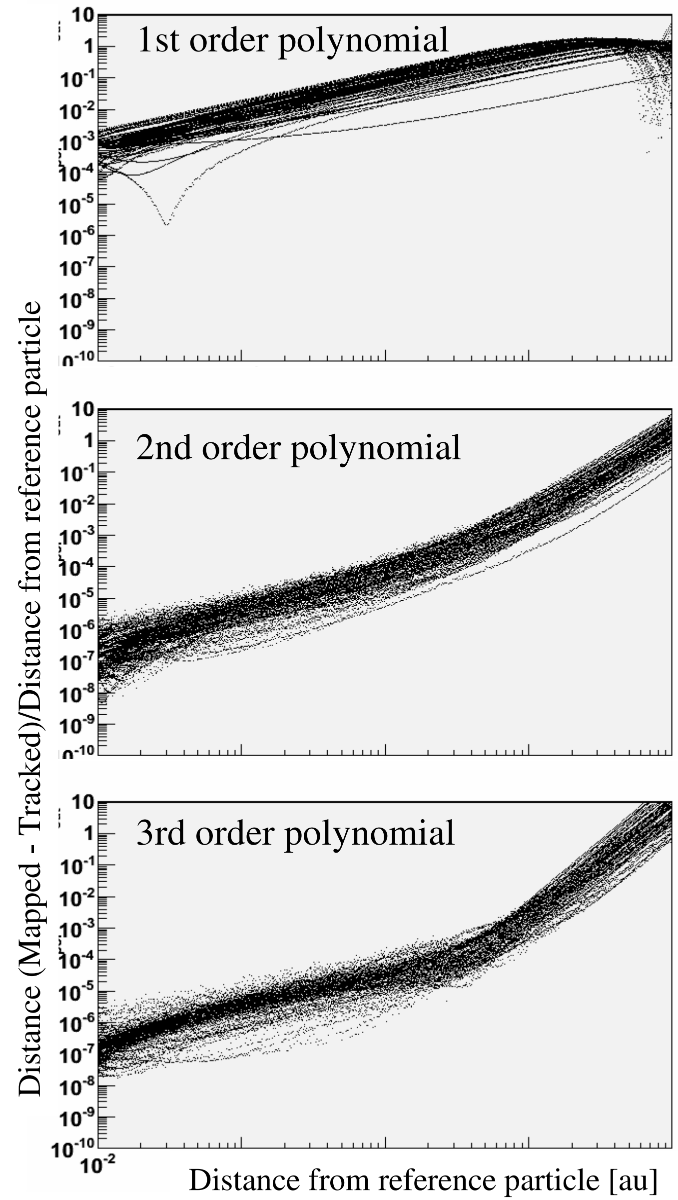
Constrained polynomial

- Try a slightly different algorithm
  - As above, but instead of fitting to an nth order polynomial, I:
    - Fit to a 1st order polynomial
    - Fit to a second order polynomial forcing first order terms to be as above
    - Repeat up to nth order
  - Fit looks a bit better...
  - Equivalent to "forcing differential" in 1D example
    - But can include higher polynomial terms

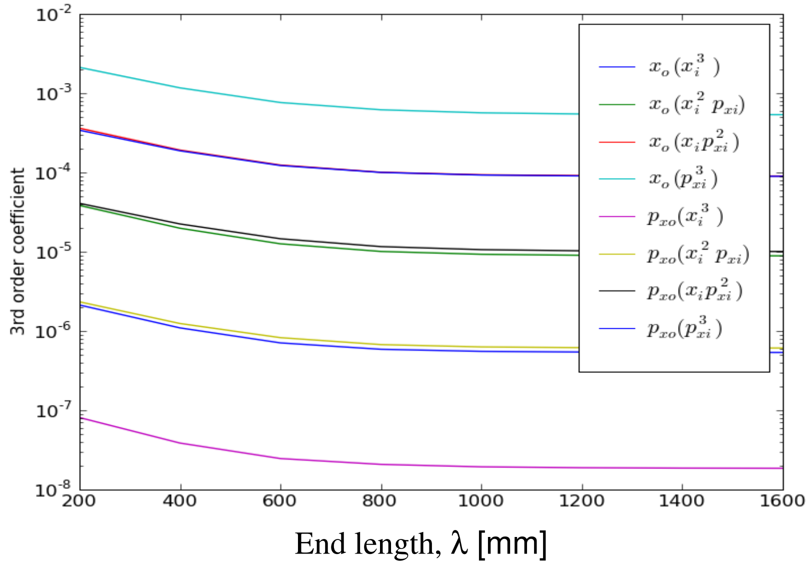


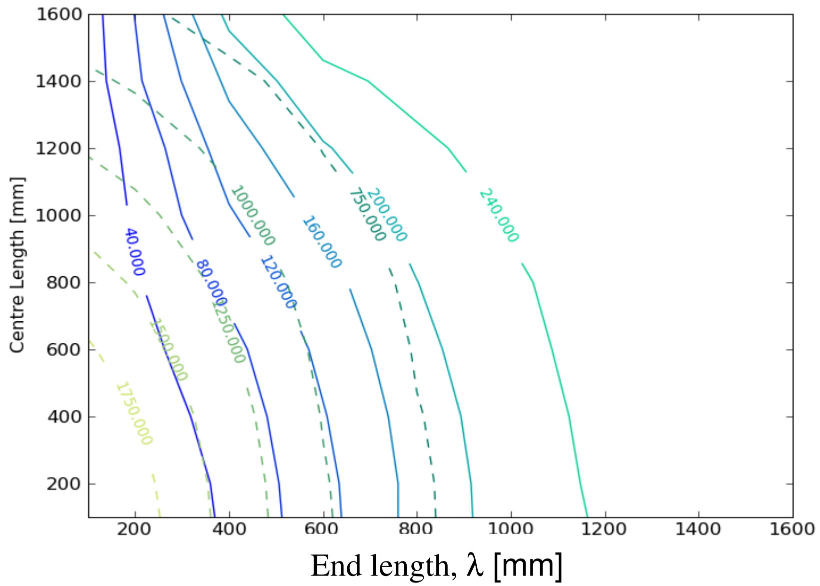
Transverse phase space

- Can extend to 4D (x,y,px,py)
  - Main contribution from 3rd order polynomial terms
  - Fits with beam optics
  - In theory expect 3rd order spherical aberrations
  - Slight improvement from 4th order terms
  - No real contribution beyond 4th order
    - Presumably algorithm starts running out of steam beyond 4th order
- No improvement from constraining at lower order



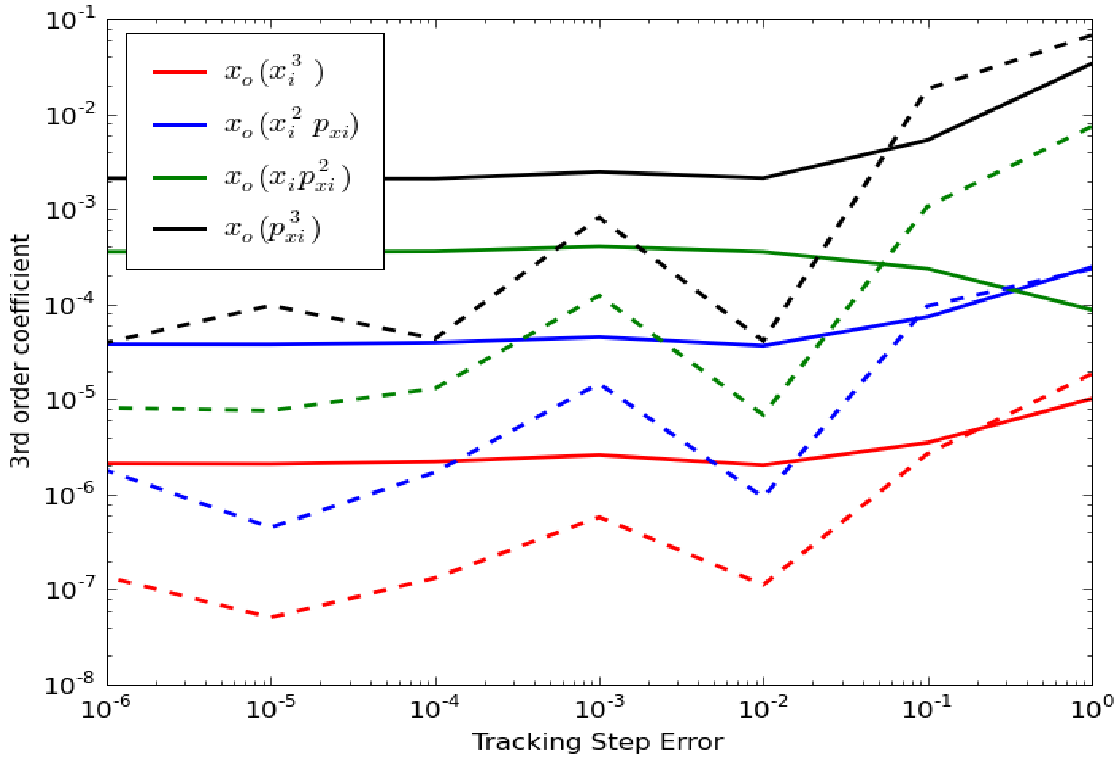
Non-Linear Terms vs End Field





- These non-linear terms are quite dependent on length of solenoid fringe field
- For very short fringe fields 3rd order terms become large
  - d^2B_z/dz^2 becomes large
  - e.g. consider tanh model for $B_z(r=0)$
  - $B_z = \tanh[(z-z_0)/\lambda] + \tanh[(z-z_0)/\lambda]$

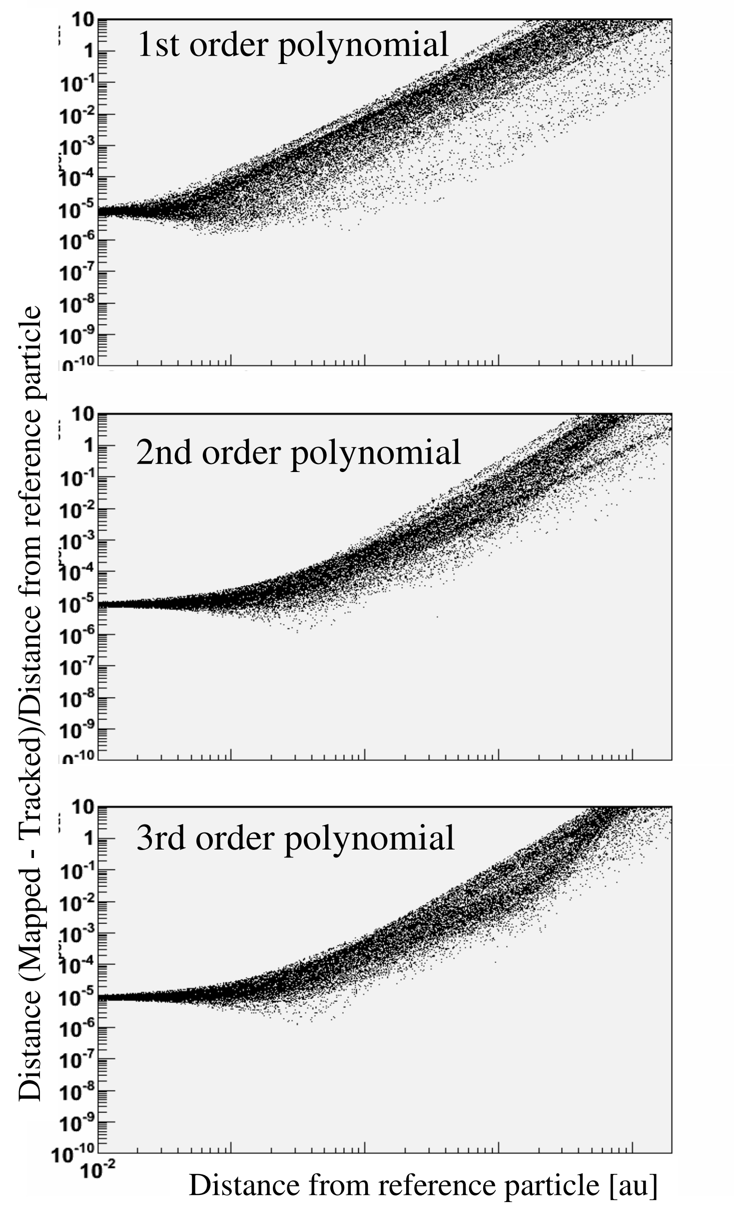
Stability vs tracking - x vs px



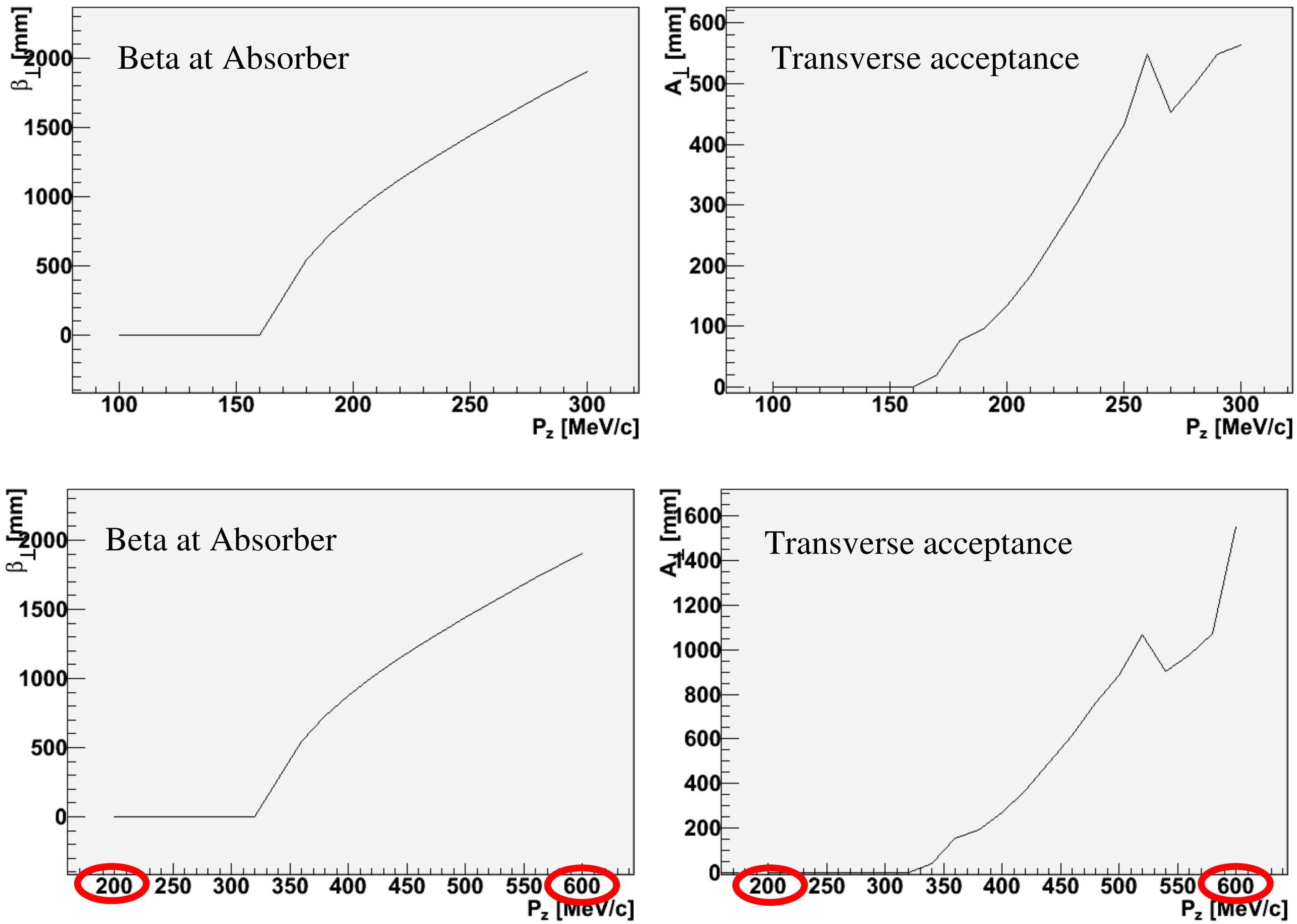
- Is this an effect from tracking accuracy?
  - Estimate algorithmic stability by looking at coefficient variance after calculation with several sets of particles
  - Reasonably stable so long as tracking is ok
    - error on polynomial coefficient ~ 1%
  - (Can do better with better set of particles - this is a Gaussian beam)

6D Mapping

- Can extend to 6D (t,E,x,y,px,py)
  - In 6D need to constrain at lower order
  - Non-linear terms at 2nd order
  - 3rd order contribution doesn't make much difference
    - Algorithm running out of steam even at 3rd order...



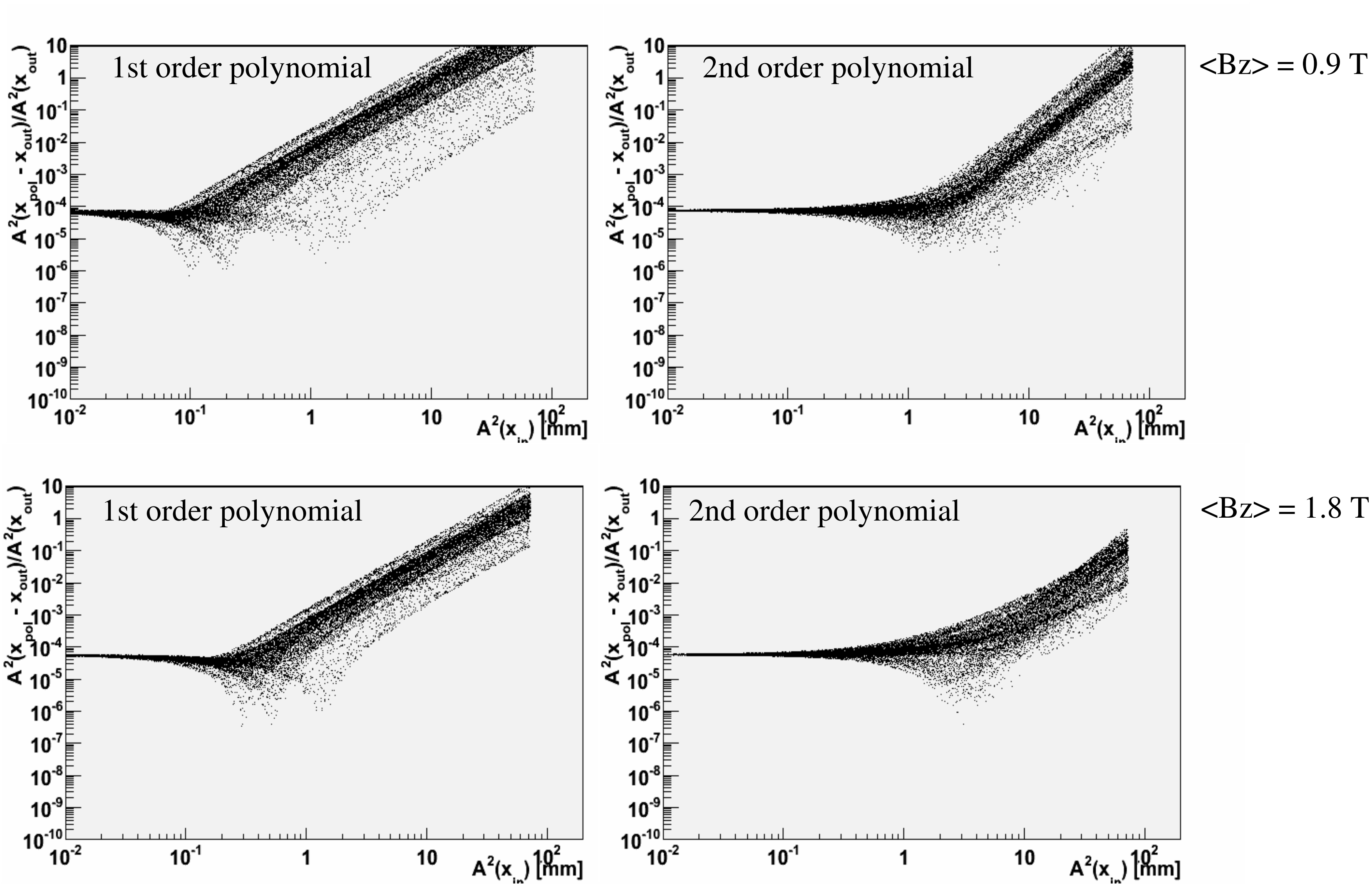
E.g. Effect of Momentum



<Bz> = 0.9 T

<Bz> = 1.8 T

E.g. Effect of Momentum



Conclusion

- Focus on algorithms here
  - Interesting algorithms developed
  - Enable study of non-linear terms in a tracking code
- Try to use them to develop some lattices...