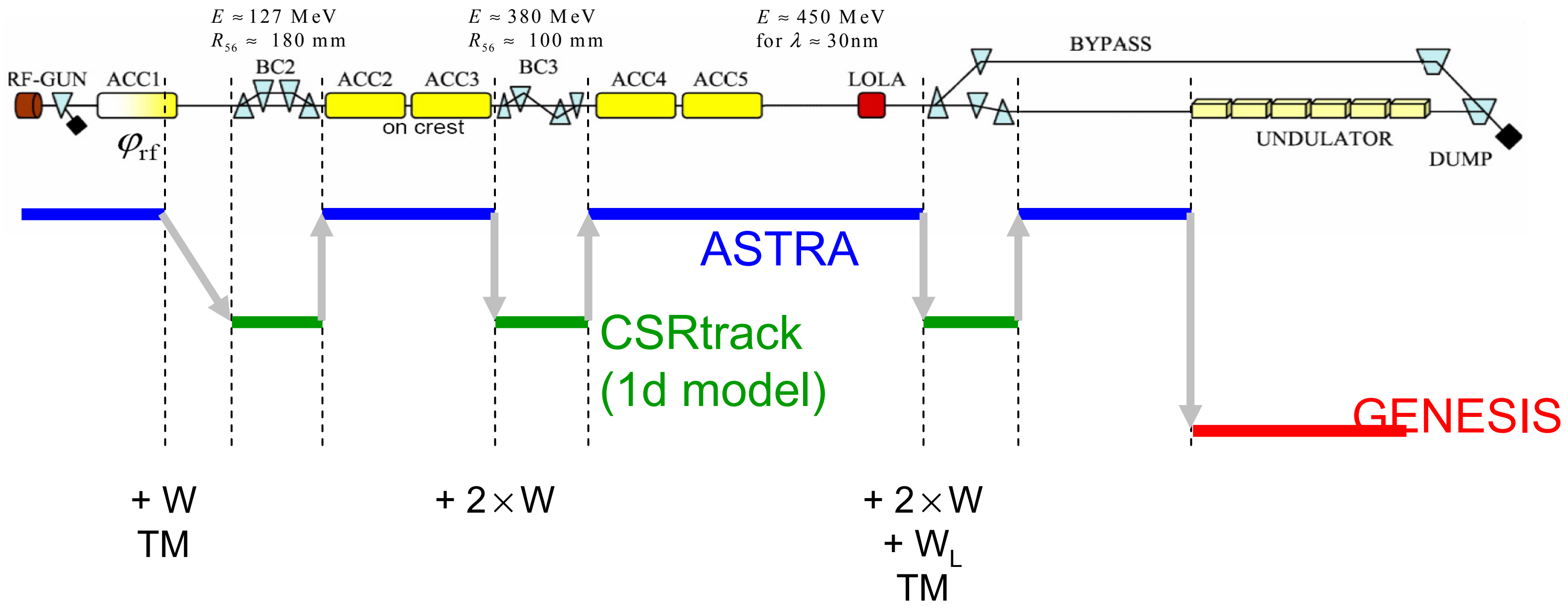
GlueTrack interpreter for S2E beam dynamic simulations

Igor Zagorodnov
BDGM, DESY

23.01.06

M.Dohlus

TTF2 – s2e



W = wake of one TTF module

W_L = wake of LOLA structure

TM = transverse matching to design optic

We want to glue all components, transformations and conversions in one script.

Scripting language?
Python (Guido van Rossum)

- Open Source (OSI Certified)
  - copyrighted but use not restricted
- Mature (13 years old)
- Supportive user community
  - plenty of good books, too
- Simple design, easy to learn
  - reads like “pseudo-code”
  - Suitable as first language
  - Suitable as last language :-)

Why Python?

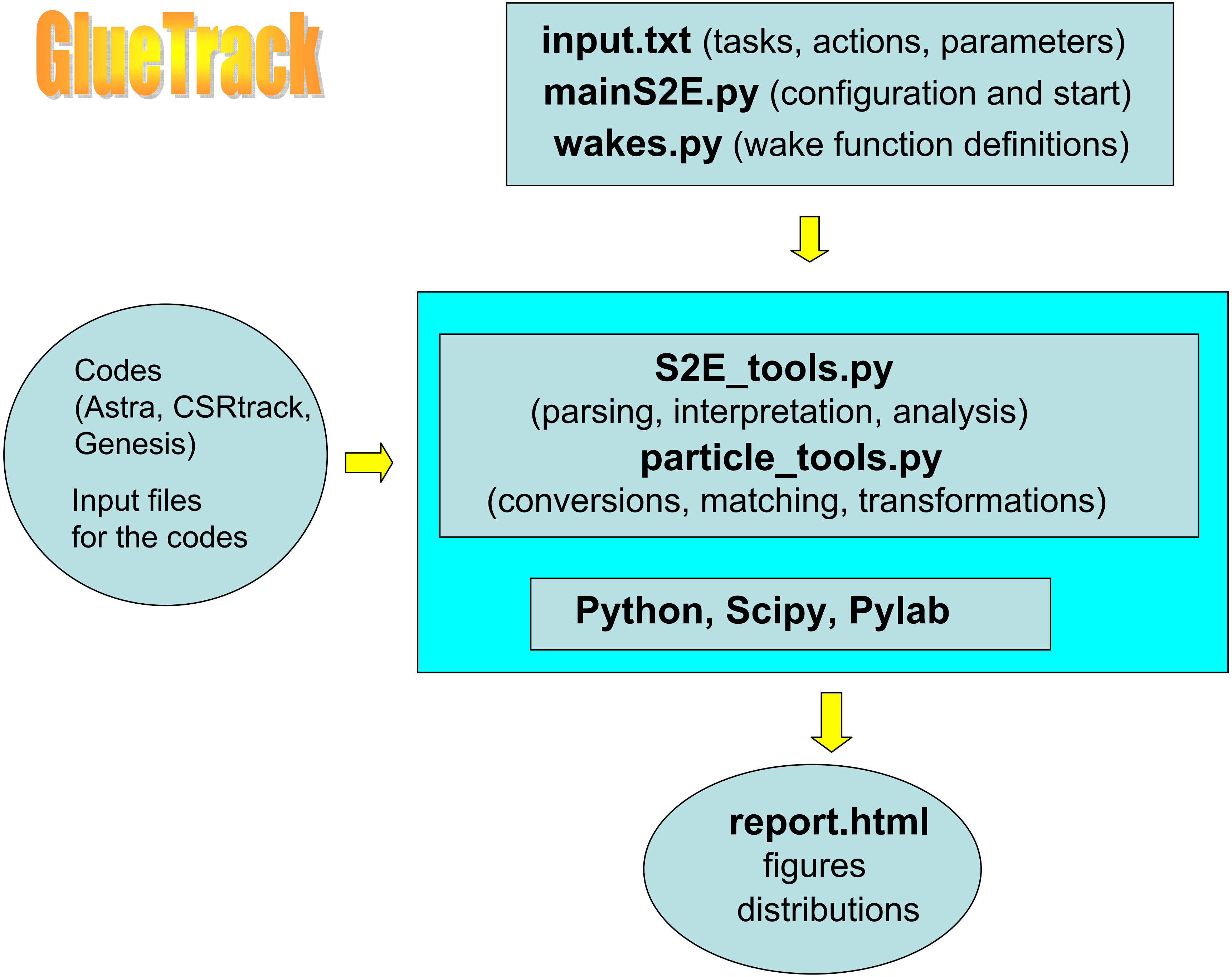
- Extremely portable
  - Unix/Linux, Windows, Mac, PalmOS, WindowsCE, RiscOS, VxWorks, QNX, OS/2, OS/390, AS/400, PlayStation, Sharp Zaurus, BeOS, VMS…
- Compiles to interpreted byte code
  - compilation is implicit and automatic
- Memory management automatic
- “Safe”: no core dumps due to your bugs
- Extensible (add new modules)
  - C/C++/Fortran/whatever
  - Java (through Jython)

Who is using Python?

- Google (various projects)
- NASA (several projects)
- NYSE (one of only three languages "on the floor")
- Industrial Light & Magic (everything)
- Yahoo! (Yahoo mail & groups)
- RealNetworks (function and load testing)
- RedHat (Linux installation tools)
- LLNL, Fermilab (steering scientific applications)
- Zope Corporation (content management)
- ObjectDomain (embedded Jython in UML tool)
- Alice project at CMU (accessible 3D graphics)

J.Amundson et al. *Synergia: An accelerator modeling tool with 3-D space charge*, Journal of Comp.Phys., 211 (2006)

GlueTrack



input.txt (tasks, actions, parameters)

mainS2E.py (configuration and start)

wakes.py (wake function definitions)

Codes
(Astra, CSRtrack,
Genesis)

Input files
for the codes

S2E_tools.py
(parsing, interpretation, analysis)

particle_tools.py
(conversions, matching, transformations)

Python, Scipy, Pylab

report.html
figures
distributions

input.txt (actions, tasks, parameters)

```
! input file for Start-to-End simulation of TTF2
#00 ASTRA -> CSRtrack
action=processor
        task=read
                sort_len=True
                in_file=ttf2.1359.001
                format=astra
        task=add_wake
                wake_delta=0
                m_interv=[-0.005, 0.005]
                sigmaI=0.0001
                wake=wake0
        task=center
                centerx=True
                centery=True
                centerxs=True
                centerys=True
                innterv=[-0.002, 0.002]
        task=matching
                interv=[-0.002, 0.002]
                xpar=[4.619, 20.174,1.0]
                ypar=[-0.012, 2.809,1.0]
        task=write
                out_file=bc1_in.fmt1
                format=fmt1
```

```
#01 BC1 (CSRtrack)
action=CSRtrack
        task=run
                work_dir=N1_bc1
                command=./src_11
#02 CSRTrack  -> ASTRA
action=processor
        task=read
                in_file=N1_bc1/out/bc1_out.fmt1
                format=fmt1
        task=center
                centerz=True
        task=define_ref_part
        task=drift
                length=-1
        task=write
                z0=24.854
                out_file=N2_bc_to_bc/bc1_out.ini
                format=astra
#03 BC1 to BC2 (ASTRA)
action=ASTRA
        task=run
                command=./astra, bc_to_bc100.in
                work_dir=N2_bc_to_bc
        task=run
                command= ./astra, bc_to_bc300.in
                work_dir=N2_bc_to_bc
```

input.txt (actions, tasks, parameters)

processor

read
write
add_wake
matching
drift
drift_to
define_ref_part
center
extract_slice_centr
mix

analysis

read
write
slice_analysis
m_current
m_emmitance
m_momentum
m_rms_momentum
m_hor_offset
m_hor_slope
m_Twissx
m_Twissy
m_topview

ASTRA, CSRTrack

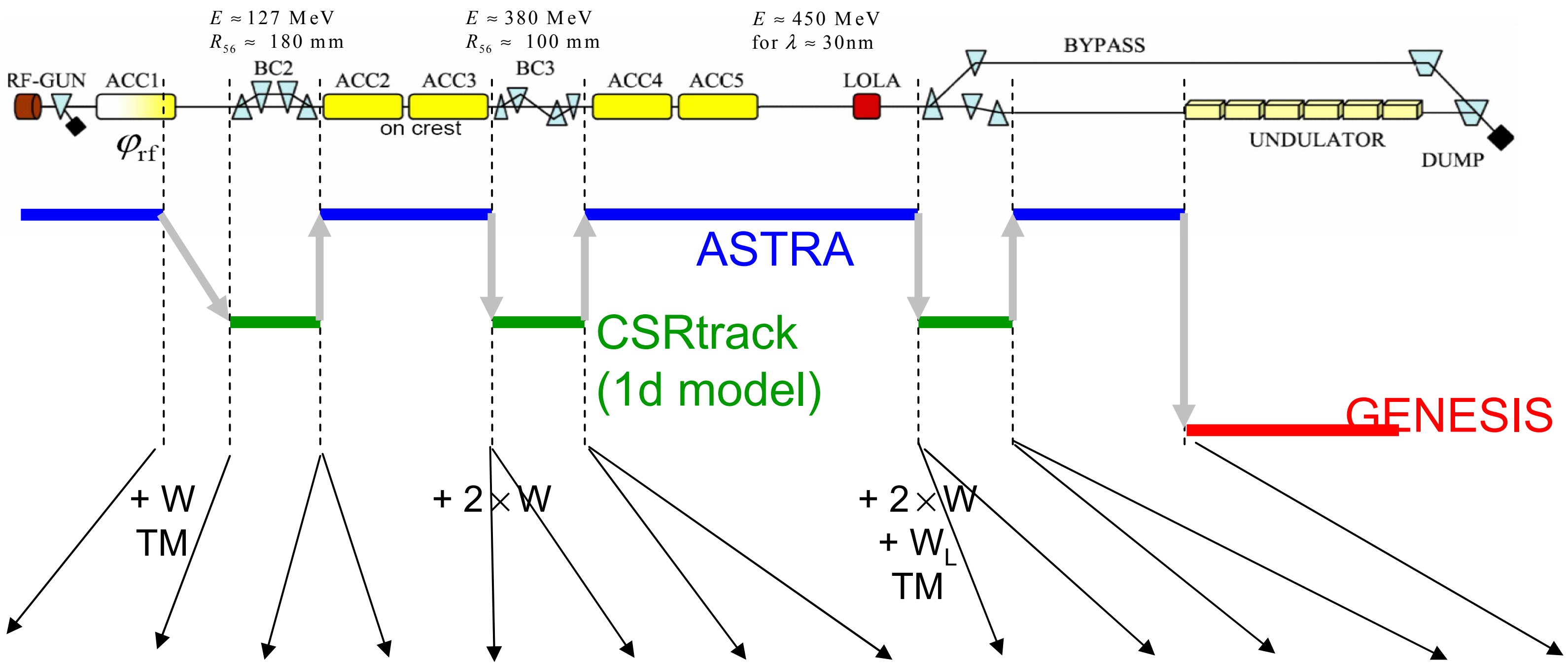
run
os

mainS2E.py

```
from S2E_tools import *

#configuration
runtype='restart'; #'restart','new'
fig_dir='./Figures/';
report_file=fig_dir+'report.html';
inputfile='input.txt'
outputfile='output.txt'


# main code
if runtype=='new':
   html=open(report_file,'w');
   html.write('<HTML><BODY BGCOLOR="white">\n')
   start,stop,indfig=0,1000,0
if runtype=='restart':
   html=open(report_file,'a+');
   start,stop,indfig=FindRestartParameters(html)

actions=ReadTask(inputfile);
indfig=RunTask(actions,fig_dir,html,start=start,stop=stop,indfig=indfig)
html.write('</BODY></HTML>\n'); html.close()
```

wakes.py

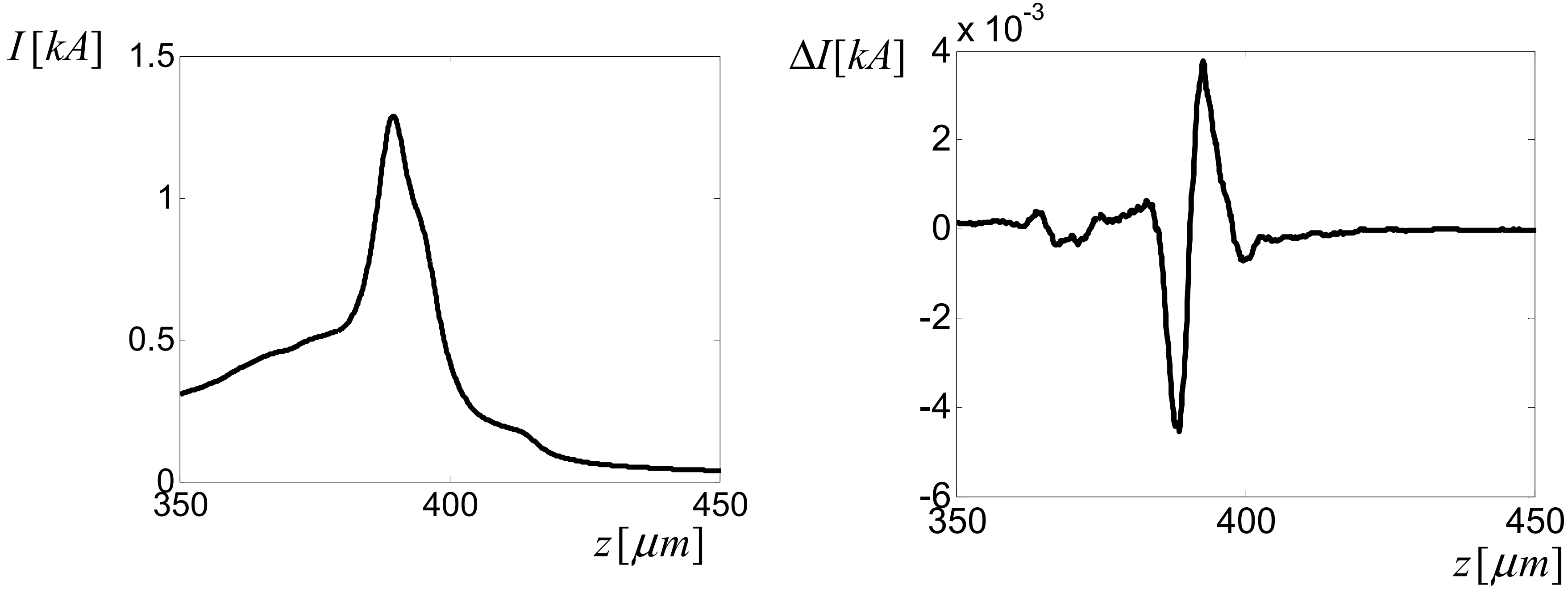
```
from scipy import *

def wakeTM(s): return -1e12*(344*exp(-sqrt(s/0.00174)))

def wakeLOLA(s): return -1e12*(257.6*exp(-sqrt(s/0.00396))+1.16*cos(1760*s**0.72)/(sqrt(s)+1600*s**1.23+1e-20))

def wake0(s): return wakeTM(s)
wake0_delta=0

def wake1(s): return 2*wakeTM(s)
wake1_delta=0

def wake2(s): return 2*wakeTM(s)+wakeLOLA(s)
wake2_delta=-0.036e12*20/22.44
```



ACC1 output	BC2 Input	BC2 output	ACC2 input	ACC3 output	BC3 Input	BC3 output	ACC4 input	LOLA output	dogleg input	dogleg output	ASTRA input	ASTRA output
0	0.4e-7	0.1e-3	0.2e-3	0.09	0.09	0.5	4	6	7	3	3	5

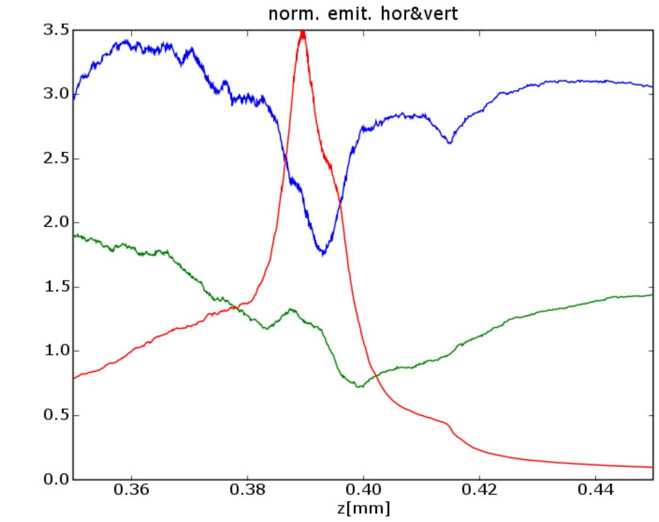
relative error[%]

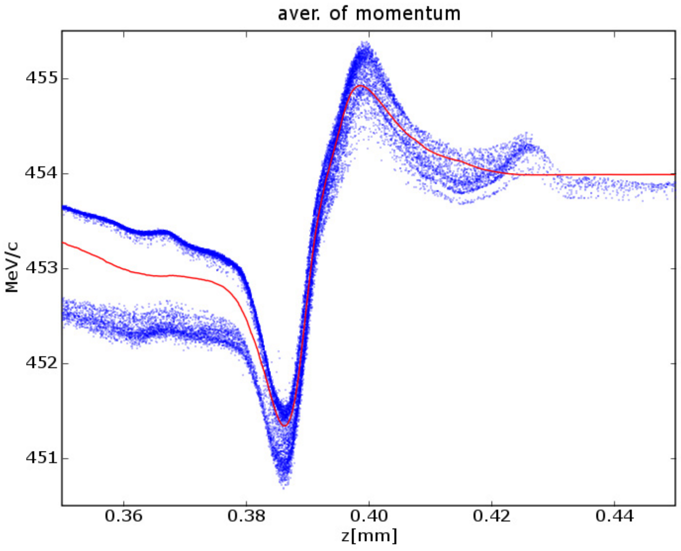
$$error = \max_j \frac{\left|a_{ij}^{MD} - a_{ij}^{GT}\right|}{\max_i \left|a_{ij}^{MD}\right|} 100\%$$



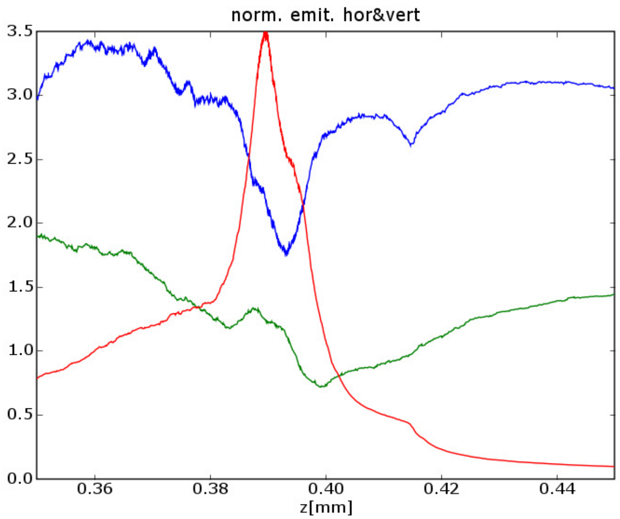
$$\Delta I = I^{MD} - I^{GT}$$

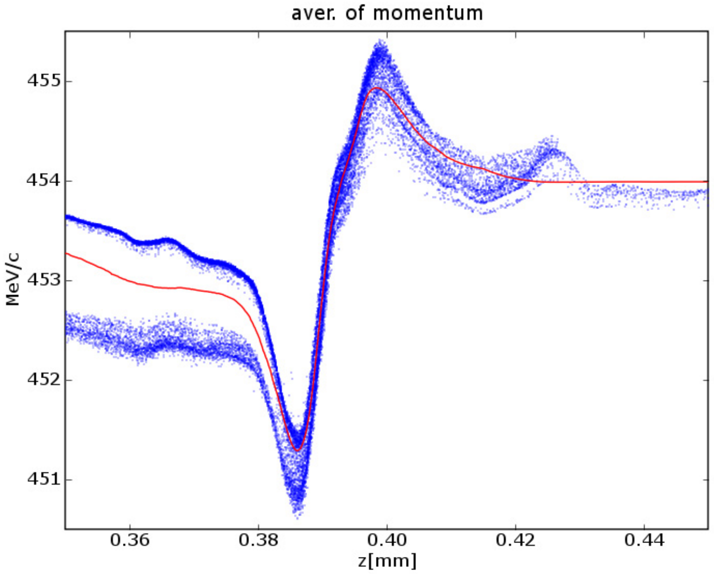
Martin Dohlus





GlueTrack





Genesis input file difference?

```
#
? VERSION = 1.0
? SIZE = 319
? COLUMNS ZPOS GAMMA0 DELGAM EMITX EMITY RXBEAM RYBEAM CURPEAK
```

$$error = \max_{j} \frac{\left| a_{ij}^{MD} - a_{ij}^{GT} \right|}{\max\limits_{i} \left| a_{ij}^{MD} \right|} 100\% = 1.5\%$$

Next

1. Parameter scans
2. XFEL simulations

Aknowledgements to Martin Dohlus
for guidance and MathCAD scripts