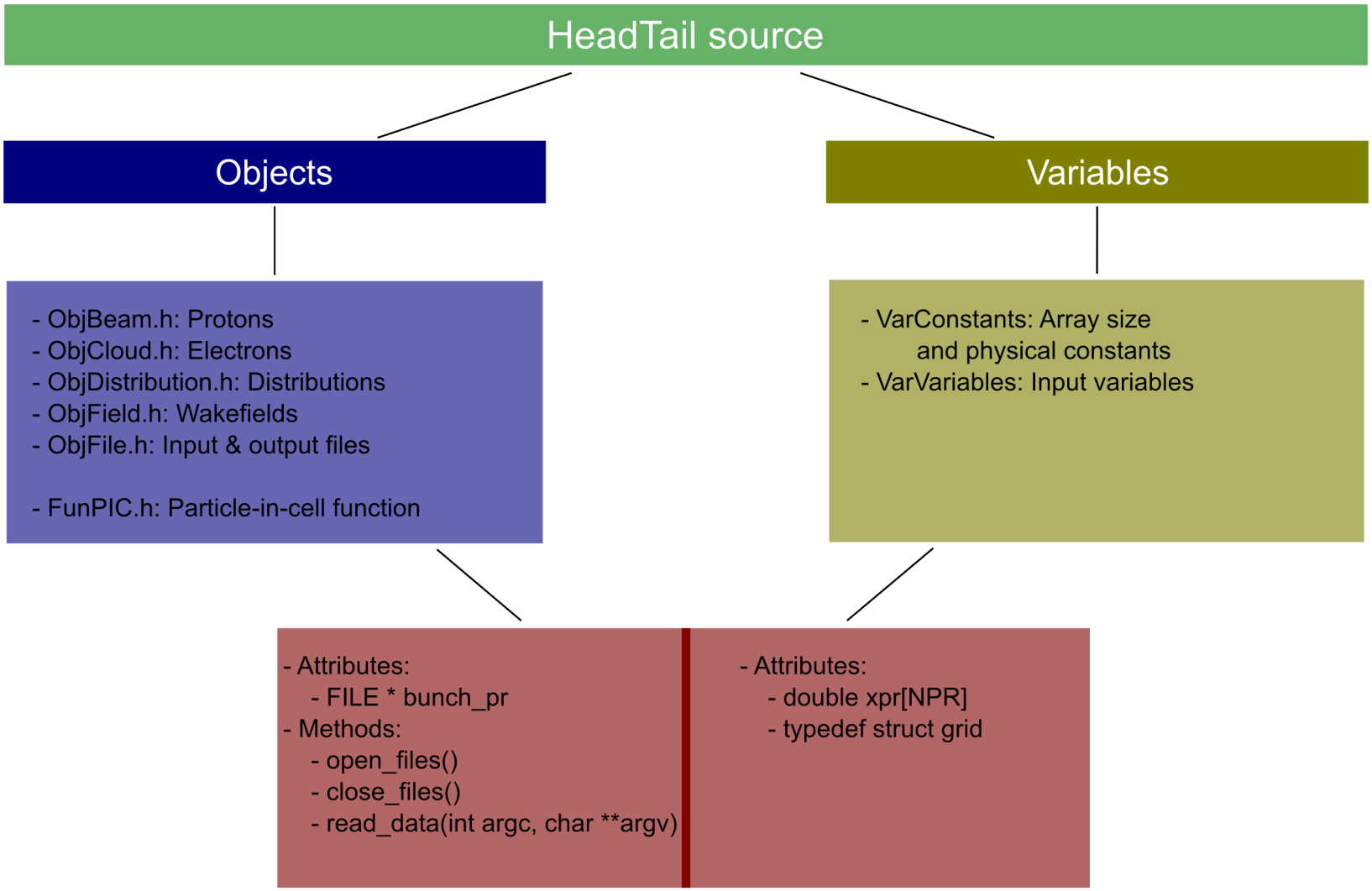
A HeadTail Analysis Tool

Kevin Li, H. Bartosik, G. Rumolo, B. Salvant...

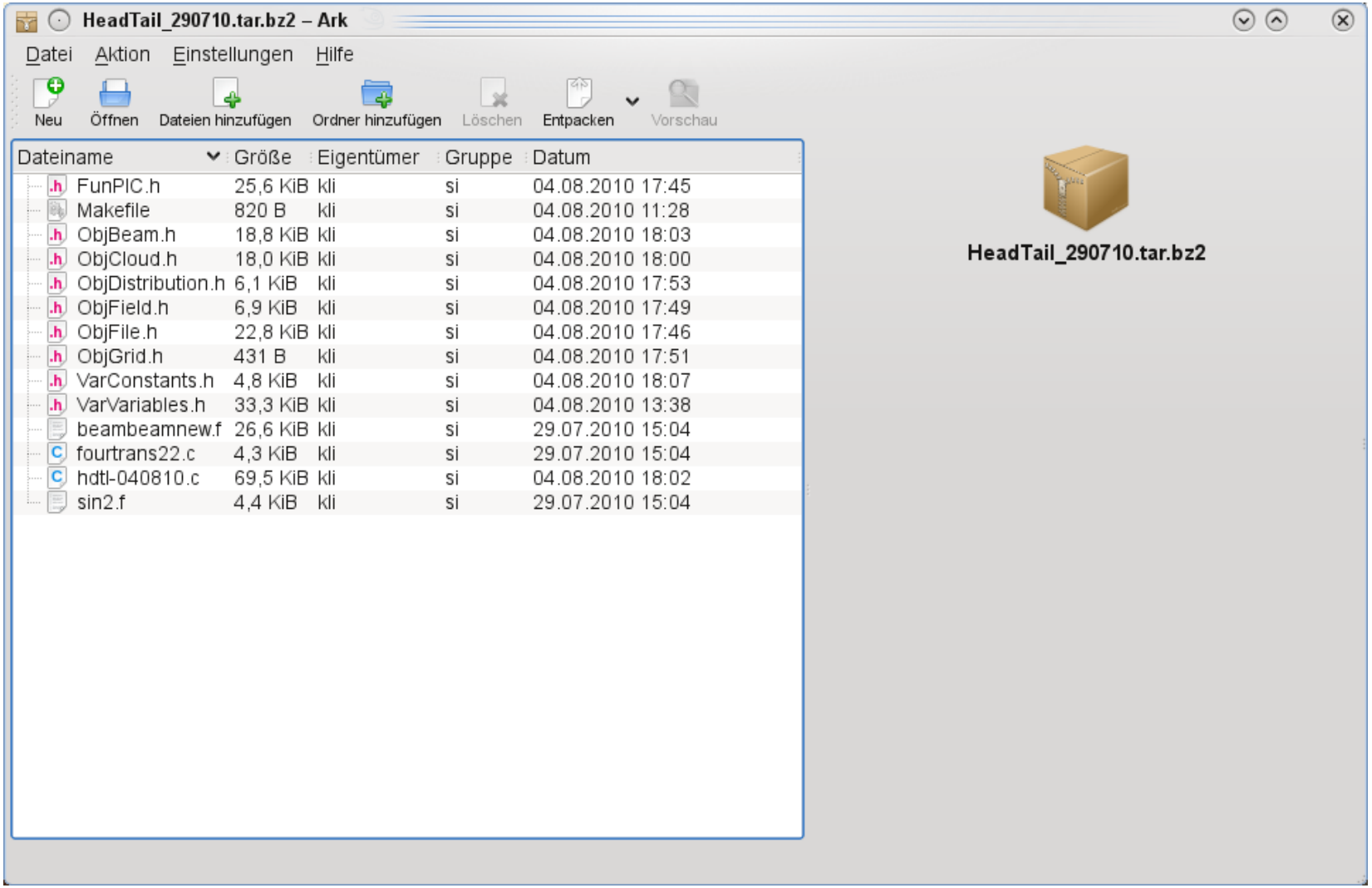
September 1, 2010

Outline

Proposed file structure for future development



Proposed file structure for future development



Include nrutils.h

Avoid redundant definition of standard routines

- ivector
- dvector
- nrerror
- frees etc.

Include nrutils.h

```
/*************************************************************************
    INCLUDES
*************************************************************************/
#include  <libio.h>
#include  <math.h>
#include  <stdio.h>
#include  <stdlib.h>
#include  <string.h>
#include  <time.h>

#include "fourtrans22.c"
#include "nrutil/nrutil.h"

#include <gsl/gsl_sf.h>
#include <gsl/gsl_sf_erf.h>
// #include <nag_stdlib.h>
// #include <nags.h>

#include "FunPIC.h"
#include "ObjBeam.h"
#include "ObjCloud.h"
#include "ObjDistribution.h"
#include "ObjField.h"
#include "ObjFile.h"
#include "VarConstants.h"
#include "VarVariables.h"


/*************************************************************************
    FUNCTIONS
```

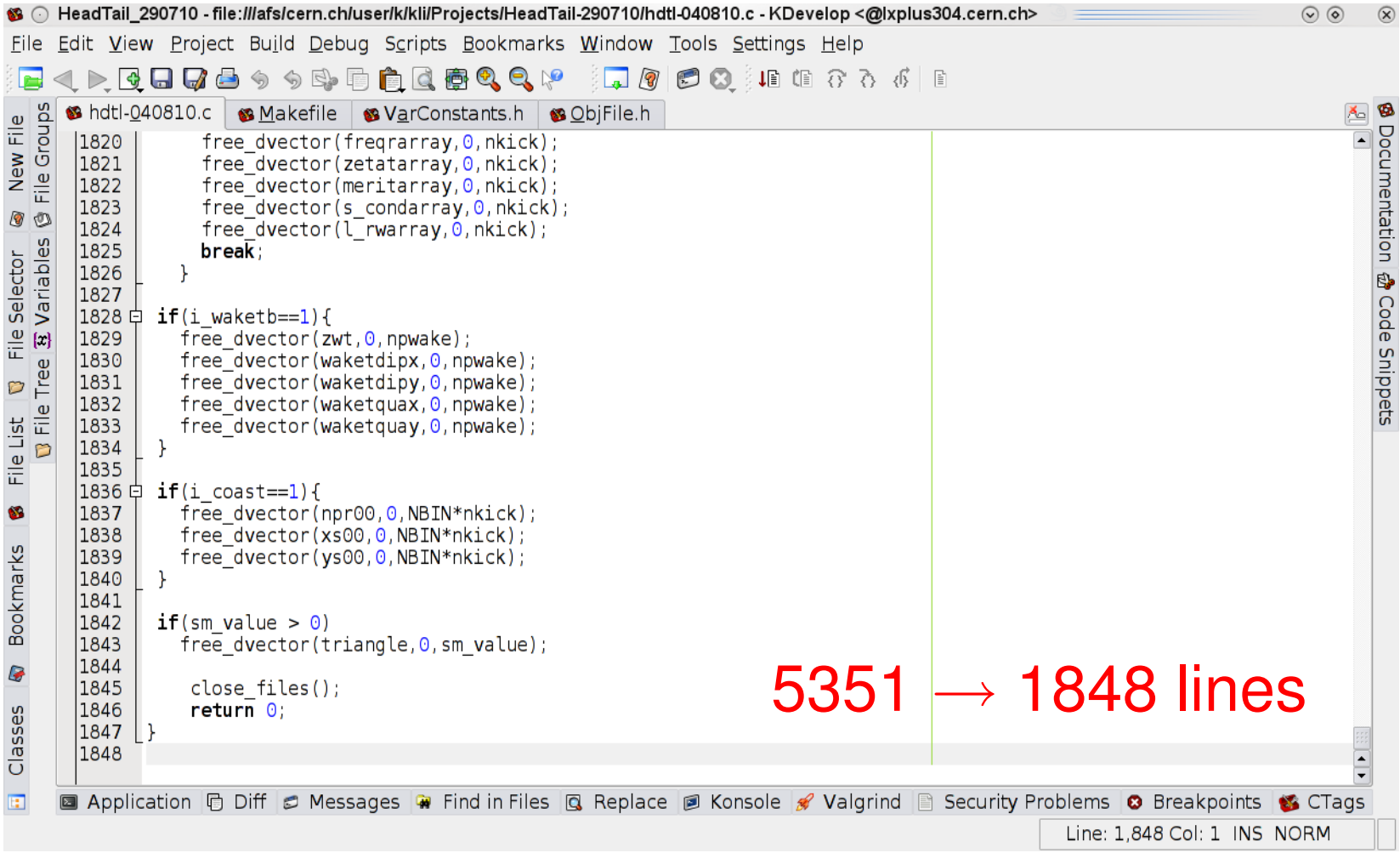
Possible advantages

- Simplicity
  - Reduced file sizes
  - Fast access to subroutines
  - Avoid redundancy
- Structure
  - Modularity
  - Fast and easily extendible
- Drawback:
  Must know the location of subroutines
  (forces a logical organisation & reduces the effectivity of ctrl+s)

Possible advantages

- Simplicity
  - Reduced file sizes
  - Fast access to subroutines
  - Avoid redundancy
- Structure
  - Modularity
  - Fast and easily extendible
- Drawback:
  Must know the location of subroutines
  (forces a logical organisation & reduces the effectivity of ctrl+s)

Possible advantages



Inputfile argument

Note! Slight modification: input file is now passed as argument
$\longrightarrow$ makes execution and scripting slightly more compact

```
/*************************************************************************
 *** Subroutine     : read_data                                          ***
 *** Effect         : Define main parameters by reading data from specified ***
 ***                   configuration-file                                 ***
 *** Parameters     : none                                               ***
 *** Gbl var   used :                                                    ***
 *** Gbl var effect :                                                    ***
 *** Constants used : none                                               ***
 *** Subrout.  used : none                                               ***
 *************************************************************************/

void read_data (int argc, char **argv) {
/* @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   5. Feb 10 - Kevin: changed towards C++ (with strcpy remnant)
   to enable direct filename input from terminal; simply enter entire
   input filename (with extension)
*/
    char  cfg_filename[300], dummy_string[64], beta_filename[300], wake_filename[300];
    long   k_cou, condition;
    double accumx, accumy, dummy1, dummy2, dummy3, dummy4, dummy5;
    FILE  *cfg_file_ptr, *beta_file, *wake_file;

    /* kev august-10: changed to pass filename as argument */
    /*printf ("\n\n    Please specify name (without extension and 300 characters maximum)") ;
    printf ("\n    of the desired configuration file : ") ;
    scanf ("%s",filename) */;

    strcpy (filename, argv[1]);
    printf ("Using input file: %s\n\n", filename);
    strcpy (cfg_filename, filename);
    //   strcat (cfg_filename,".cfg");

    //strcpy (filename,"ecsps");
    //strcpy (filename,"ecj");
    //strcpy (filename,"ecjlastali");
    //strcpy (filename,"ecEPAC");

    cfg_file_ptr = fopen(cfg_filename, "r");
    if (cfg_file_ptr == NULL) {
        printf ("\n\n    Configuration-file does not exist.\n");
        exit (2);
    }
```

SUB-HDTL/SUB-ECLD

Fast and simple script that makes use of LSF job arrays
$\longrightarrow$ (quasi-)parallel exection of jobs.

Required user input

- filename
- scan variable
- scan range

SUB-HDTL/SUB-ECLD

```
#!/afs/cern.ch/eng/sl/lintrack/Python-2.5/bin/python2.5
#!/usr/bin/python
#=============================================================================
#=============================================================================
# 5. Feb 10 - Li: script to run HEADTAIL in batch mode
# some time - Kevin: included the offset
#=============================================================================
#=============================================================================
import re,shutil,stat,sys,os
from numpy import *


class Input():
    def __init__(self):
        self.vname  = []
        self.varray = []


# =============================================================================
# ================== Begin user interface                    ==================
# =============================================================================
filename = 'PSB-bbw-C0c16-Z2-TEST.cfg';
# vname='HorChr'; varray = arange(-1,5.2,0.2);
# vname='NumPar'; varray = arange(0,1020,20); # from 0 to 1e15
vname='NumPar'; varray = arange(0,1010,10); # in units 1e10
offset=0; # to continue simulations from a previous run

# =============================================================================
# ================== END user interface                      ==================
# =============================================================================
# print filename
# print vname
# print varray
# sys.exit()
```

--:-- **sub-hdtl-v05.py**   (Python)--L26--Top----------

SUB-HDTL/SUB-ECLD usage

```
Datei  Bearbeiten  Ansicht  Verlauf  Lesezeichen  Einstellungen  Hilfe
-rw-r--r-- 1 kli si 5148 Aug  5 15:10 PSB-bbw-C0c16-Z2-TEST.cfg
[lxplus310] /afs/cern.ch/user/k/kli/HDTL/PSB-Test > ../sub-hdtl-v05.py

********************************************************************************
********************************************************************************
 This is sub-hdtl version 5 to submit HEADTAIL job arrays via LSF.
 To submit, change arguments in sub-hdtl-v05.py

 The output files will have the format:
 "hdtl.'Scan-parameter-name'.'lsf_jobindex'.cfg_'headtail_extension'.dat"
 Currently implemented Scan-parameter-names are: 'HorChr' 'VerChr' 'NumPar' [in units 1e10]

 ... oh! And you need to launch the program from the directory where your
 inputfile is located in!

 Good Luck!
********************************************************************************
********************************************************************************


********** Reading inputfile:  PSB-bbw-C0c16-Z2-TEST.cfg

********** Scanning parameter: NumPar
********** from 0 to 1000 in steps of 10


WARNING: Path already exists! Overwrite? [y or n]
y

@@@@@@@@@@ Wahaha!

********** Made directory: /afs/cern.ch/user/k/kli/scratch0/PSB-bbw-C0c16-Z2-TEST-NumPar/

********** Made directory on pcbe13664: /backup/CERN/PSB-bbw-C0c16-Z2-TEST-NumPar/

********** Lauch of simulation array succeeded!
Job <87915853> is submitted to queue <1nw>.
[lxplus310] /afs/cern.ch/user/k/kli/HDTL/PSB-Test >
lxplus :
```

SUB-HDTL/SUB-ECLD usage

Datei Bearbeiten Ansicht Verlauf Lesezeichen Einstellungen Hilfe

```
87899580  kli     RUN   1nw         lxplus240   lxbst2228   *array[69] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsu1229   *array[70] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsq0529   *array[71] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsu1314   *array[72] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbst2225   *array[73] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbra3903   *array[74] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsq1311   *array[75] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsu1305   *array[76] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsp0944   *array[77] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsq1518   *array[78] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsp0801   *array[79] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsq1219   *array[80] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbst2110   *array[81] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxb8304     *array[82] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsq1435   *array[83] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbra4107   *array[84] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsp0714   *array[85] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsp0738   *array[86] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsp0703   *array[87] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsq1403   *array[88] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsq0543   *array[89] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbst2125   *array[90] Aug  5 13:29
87899705  kli     RUN   1nw         lxplus240   lxbsq1127   *_array[1] Aug  5 13:29
87899705  kli     RUN   1nw         lxplus240   lxbsq1329   *_array[2] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsu0847   *array[91] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsp0715   *array[92] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsq1441   *array[93] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsq1228   *array[94] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsq1433   *array[95] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsp0712   *array[96] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsu1220   *array[97] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsu1409   *array[98] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsq0829   *array[99] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsq1439   *rray[100] Aug  5 13:29
87899580  kli     RUN   1nw         lxplus240   lxbsq1230   *rray[101] Aug  5 13:29
87899705  kli     RUN   1nw         lxplus240   lxbsu1401   *_array[3] Aug  5 13:29
87899705  kli     RUN   1nw         lxplus240   lxbsu1148   *_array[4] Aug  5 13:29
87899705  kli     RUN   1nw         lxplus240   lxbsu1125   *_array[5] Aug  5 13:29
```

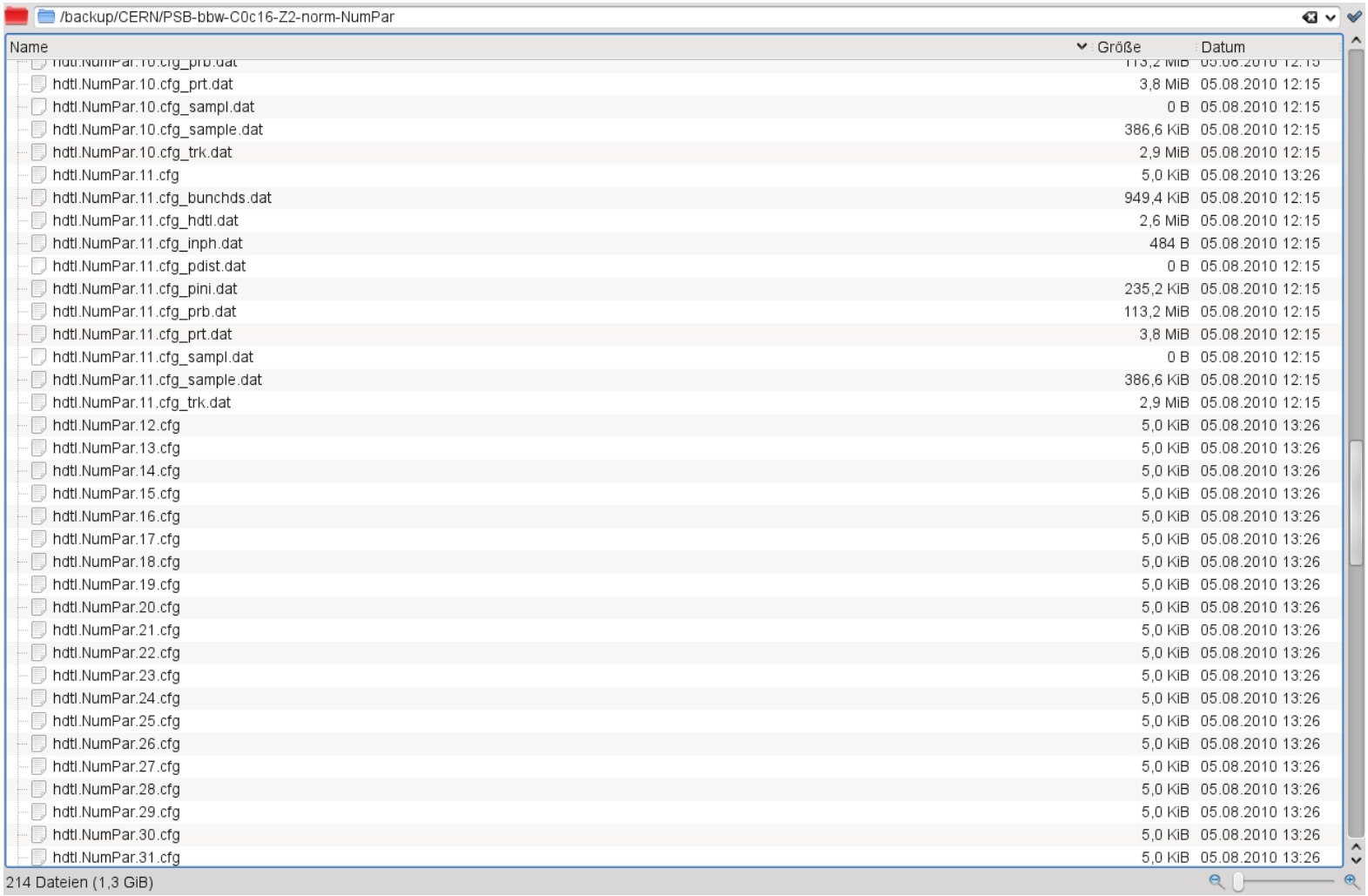
lxplus :

SUB-HDTL/SUB-ECLD output

Output directories and files

- lxplus directory:
  ∼/scratch0/"filename(-.cfg)(+scan variable name)"
- local directory:
  /backup/CERN/"filename(-.cfg)(+scan variable name)"
- filename format:
  hdtl.(+scan variable name).(+job array index+).cfg(+headtail extension)

SUB-HDTL/SUB-ECLD output



Pyheana

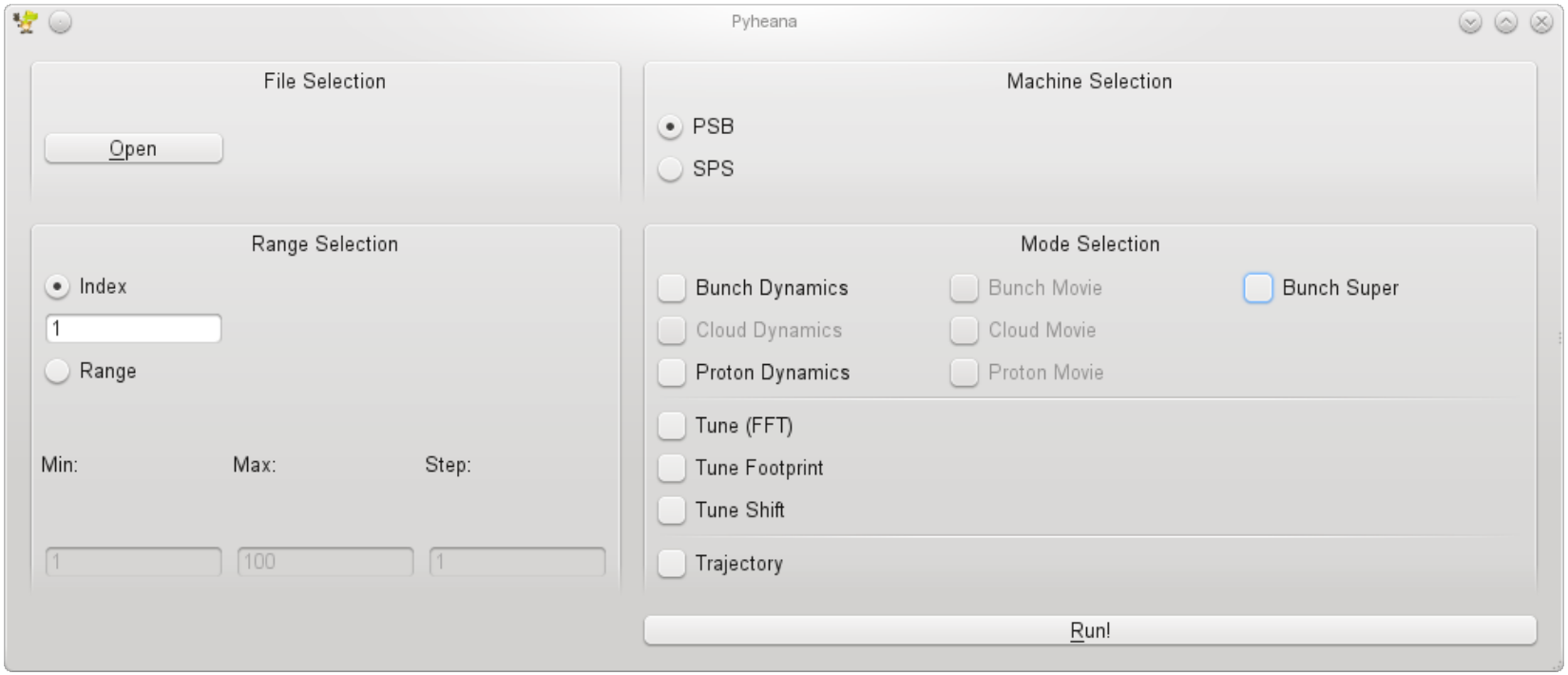
A PYthon HEadtail ANAlysis tool
or...

Pyheana

A Pyheana!

Pyheana - usage: interactive

User panel



Purpose of the analysis

- Obtain an approximate value to model the transverse broadband impedance of the PSB Ring

Broadband impedance model

Broadband impedance is characterised by its:

- resonance frequency $\longrightarrow$ calculate from pipe geometry
- shunt impedance $\longrightarrow$ simulate and extrapolate from measurements

Impedance model:

- Impedance:

$$Z_1^{\perp}(\omega) = \frac{\omega_r}{\omega} \frac{Z_t}{1 + iQ\left(\frac{\omega_r}{\omega} - \frac{\omega}{\omega_r}\right)}$$

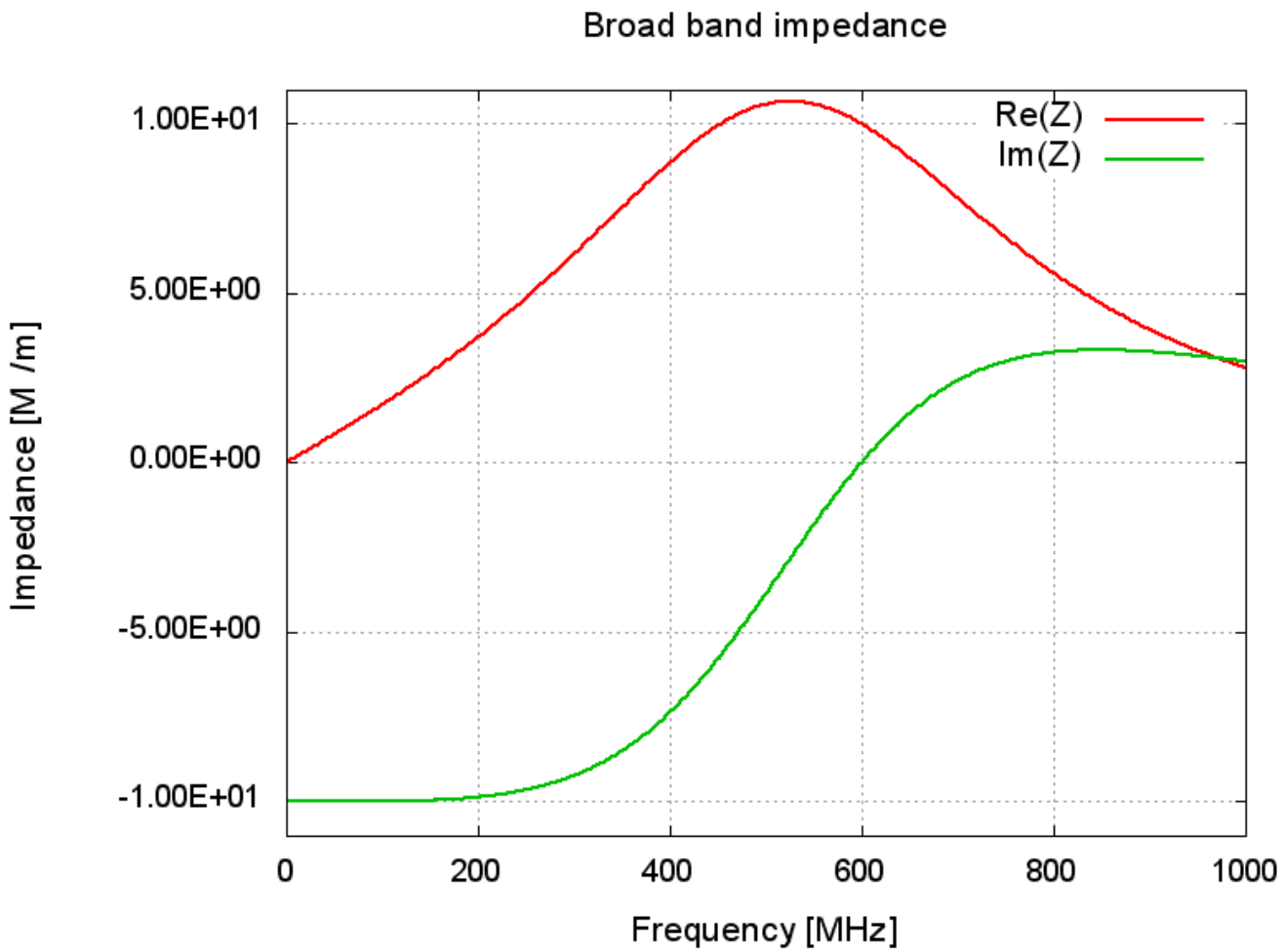
- Resonance frequency $\leftrightarrow$ cut-off frequency:

$$\omega_r = c \cdot \frac{x_{mn}}{b} \approx \frac{c}{b}, \quad \text{cirlular}$$

$$\omega_r = c\pi \cdot \sqrt{\frac{m^2}{a^2} + \frac{n^2}{b^2}}, \quad \text{rectangular}$$

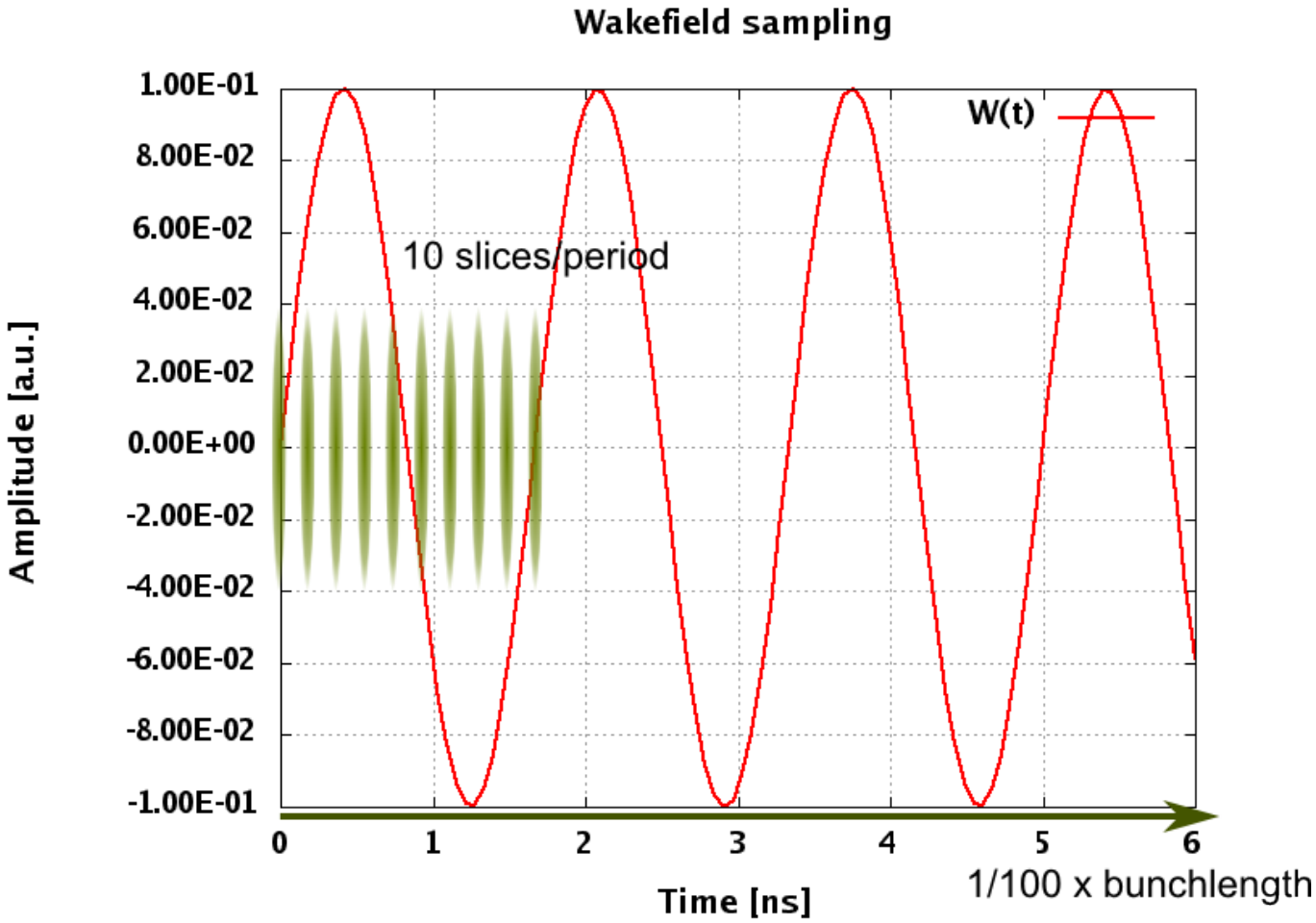
Broadband impedance model

Broadband impedance at 600 kHz:



Frequency dependent sampling

Wakefields should be sufficiently sampled by the bunch:



Frequency dependent sampling

- Machine parameters:
  - Beam pipe: $b = 8$ cm
    (cut-off frequency: $\omega_r \approx 3.75$ GHz $\rightarrow f \approx 600$ MHz)
  - Bunch length: $T_b = 600$ ns
- Sampling:
  - Wakefield sampling:
    $$\frac{T_{\text{wake}}}{T_{\text{slice}}} = \frac{2\pi b N_{\text{slice}}}{cT_b} = \frac{N_{\text{slice}}}{fT_b} \approx 10 \Rightarrow N_{\text{slice}} = 10 \times fT_b = 3\text{'}600$$
  - Slice sampling:
    $$\frac{N_{\text{total}}}{N_{\text{slice}}} \approx 5\text{'}000 \Rightarrow N_{\text{total}} = 5\text{'}000 \times N_{\text{slice}} = 18\text{'}000\text{'}000!$$

$\rightarrow$ 1 week per run! Highly inefficient!

Frequency dependent sampling

- Machine parameters:
  - Beam pipe: $b = 8$ cm
    (cut-off frequency: $\omega_r \approx 3.75$ GHz $\rightarrow f \approx 600$ MHz)
  - Bunch length: $T_b = 600$ ns
- Sampling:
  - Wakefield sampling:
    $$\frac{T_{\text{wake}}}{T_{\text{slice}}} = \frac{2\pi b N_{\text{slice}}}{c T_b} = \frac{N_{\text{slice}}}{f T_b} \approx 10 \Rightarrow N_{\text{slice}} = 10 \times f T_b = 3'600$$
  - Slice sampling:
    $$\frac{N_{\text{total}}}{N_{\text{slice}}} \approx 5'000 \Rightarrow N_{\text{total}} = 5'000 \times N_{\text{slice}} = 18'000'000!$$

$\rightarrow$ 1 week per run! Highly inefficient!

PS Booster paramaters - general settings

Machine parameters

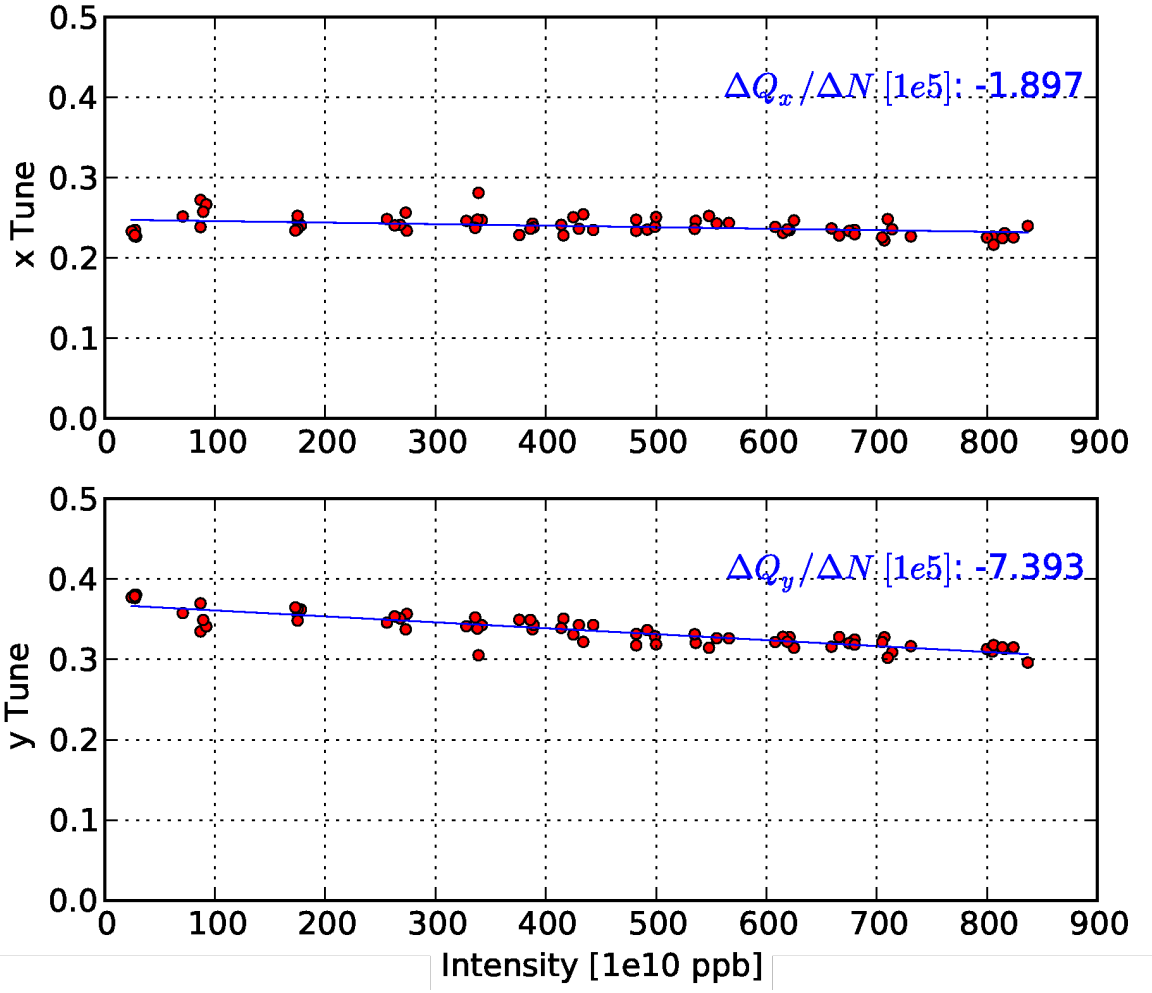
Circumference [m]	157
Energy range [MeV]	160
	1400
N Cycles	850
Injection cycle	250
Extraction cycle	850

Simulation parameters

β_x [m]	5.9
β_y [m]	5.7
Q_S	0.002106
Q_x	4.223
Q_y	4.550
α	0.060
f_Z [MHz]	600
N Slices	3600
N Macro	18e6

Measurement & simulation results

PSB tuneshift measurements at 160 MeV

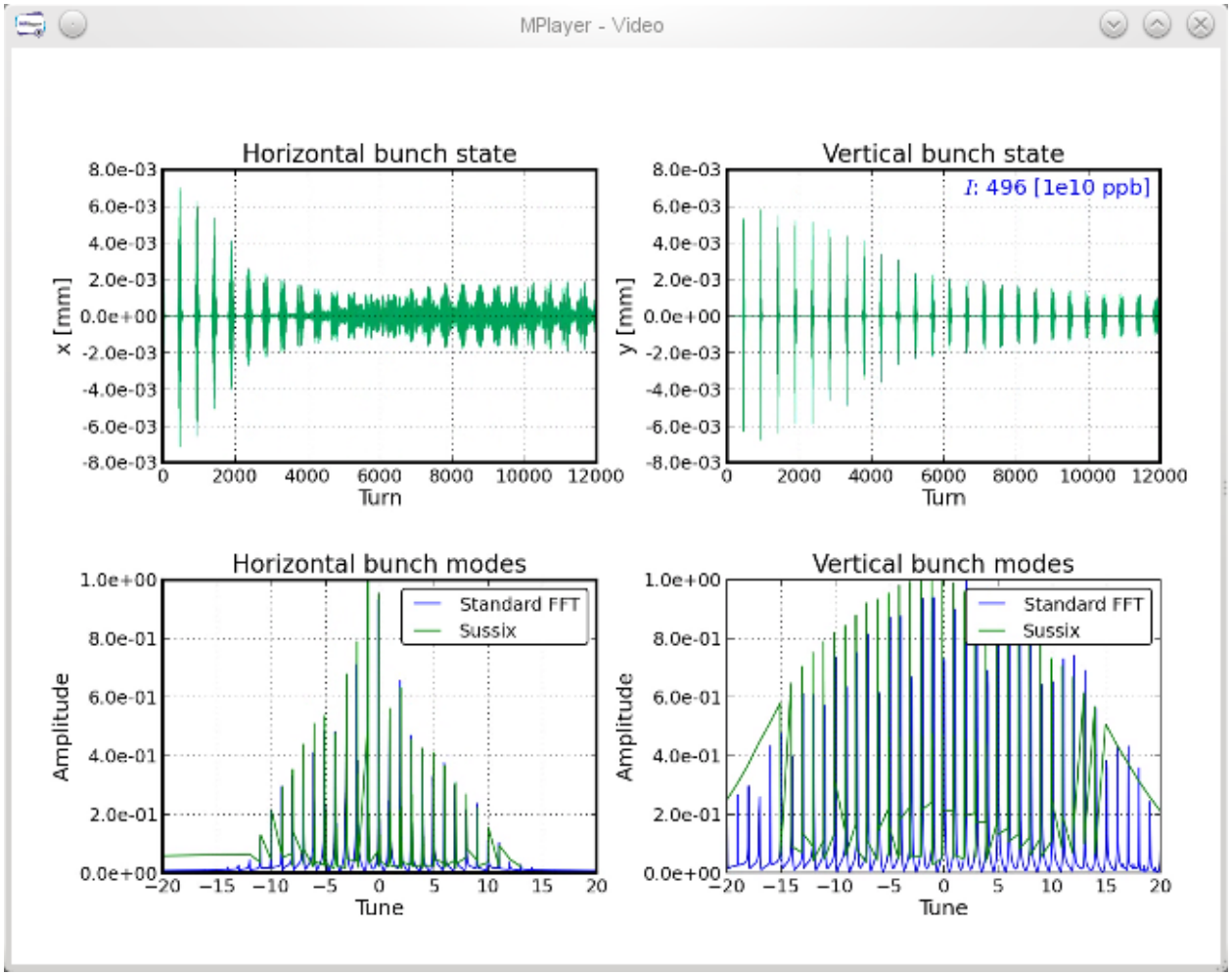


	Plot	D.Q.*
Q_x [1e-5]	-1.90	–
Q_y [1e-5]	-7.39	-7.2

∗: Diego Quatraro et al.: *Coherent tune shift and instabilities measurements at the CERN Proton Synchrotron Booster*

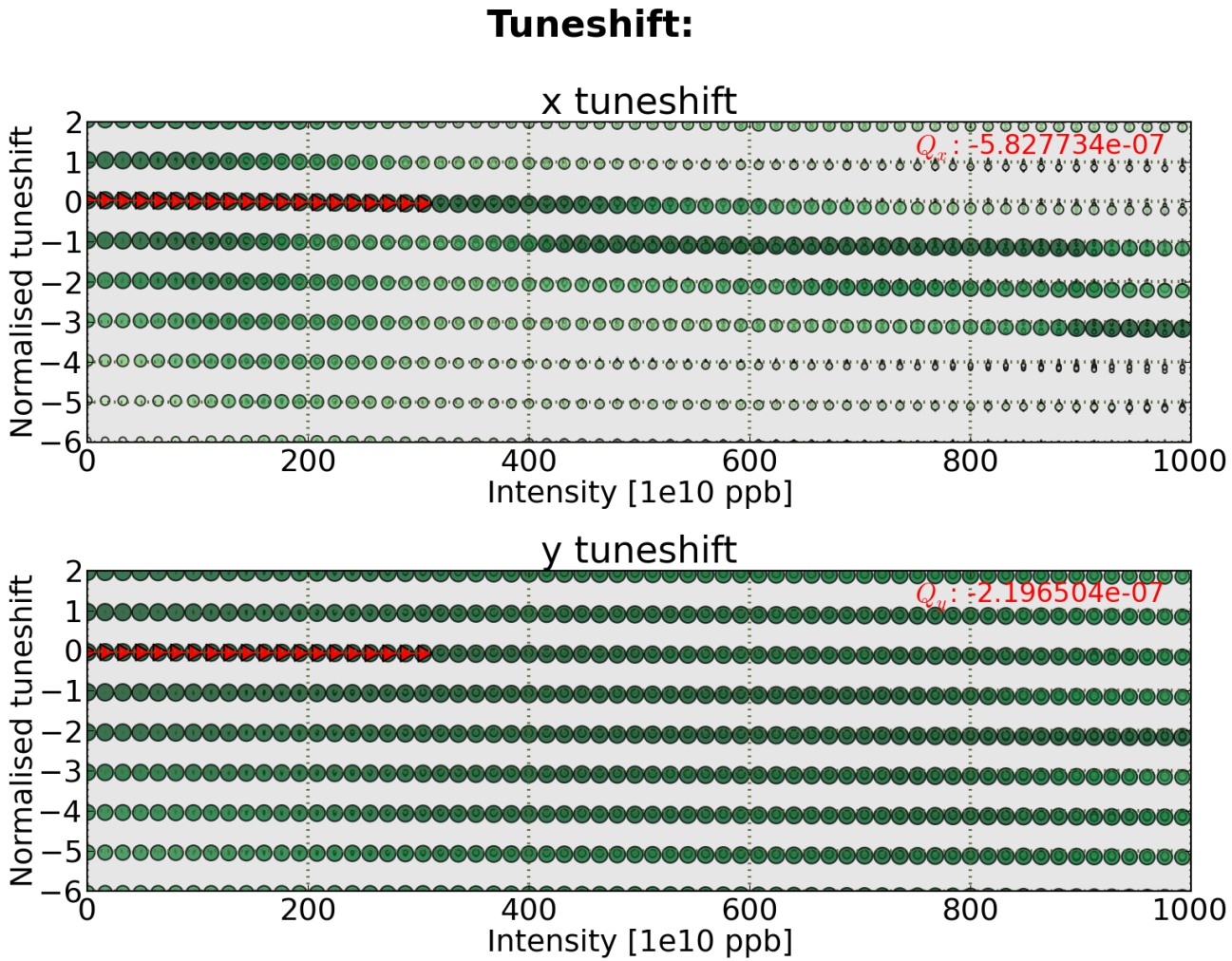
Measurement & simulation results

PSP bunch modes assuming Z=0.5 MΩ/m:



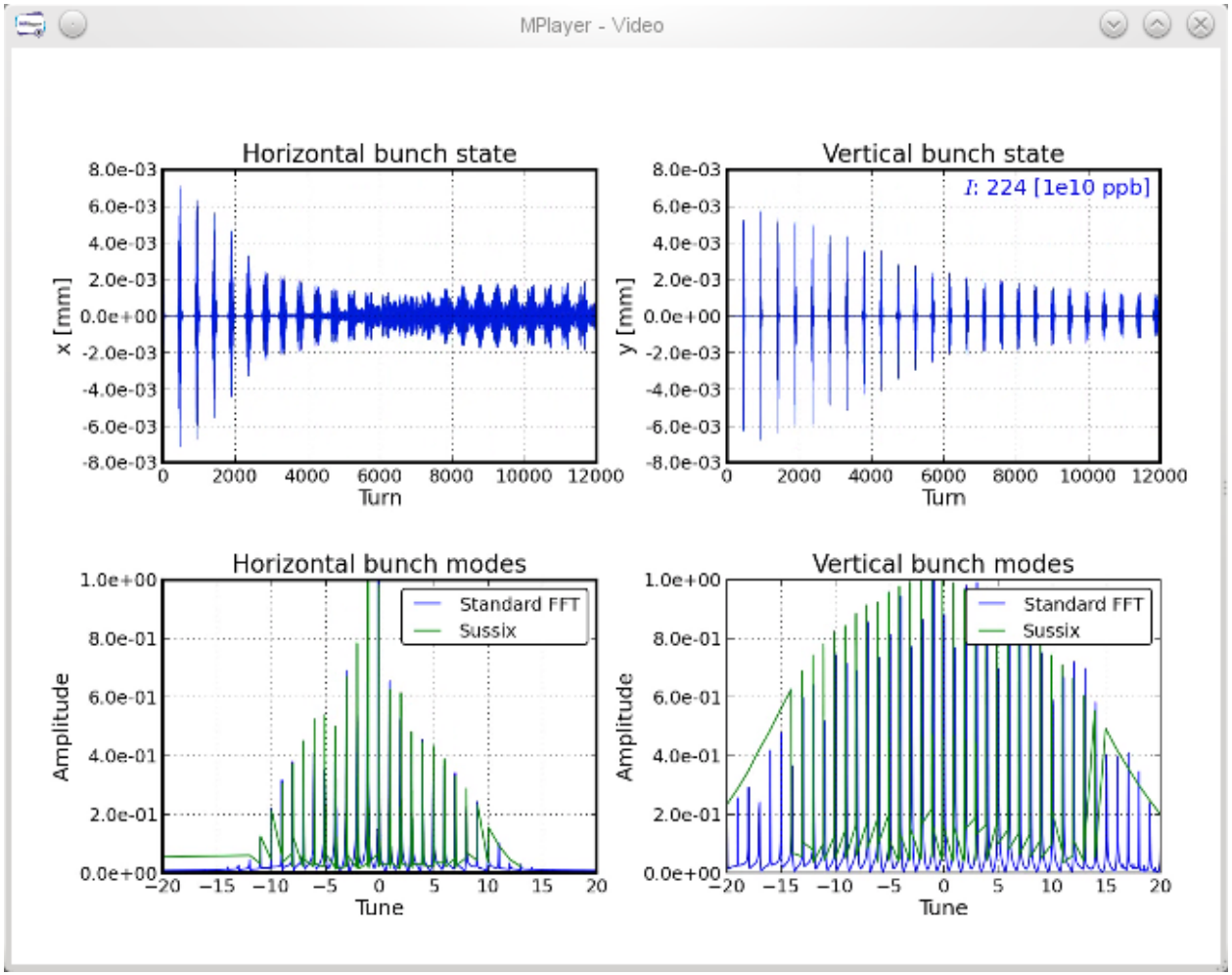
Measurement & simulation results

PSB tuneshift diagram for Z=0.5 MΩ/m



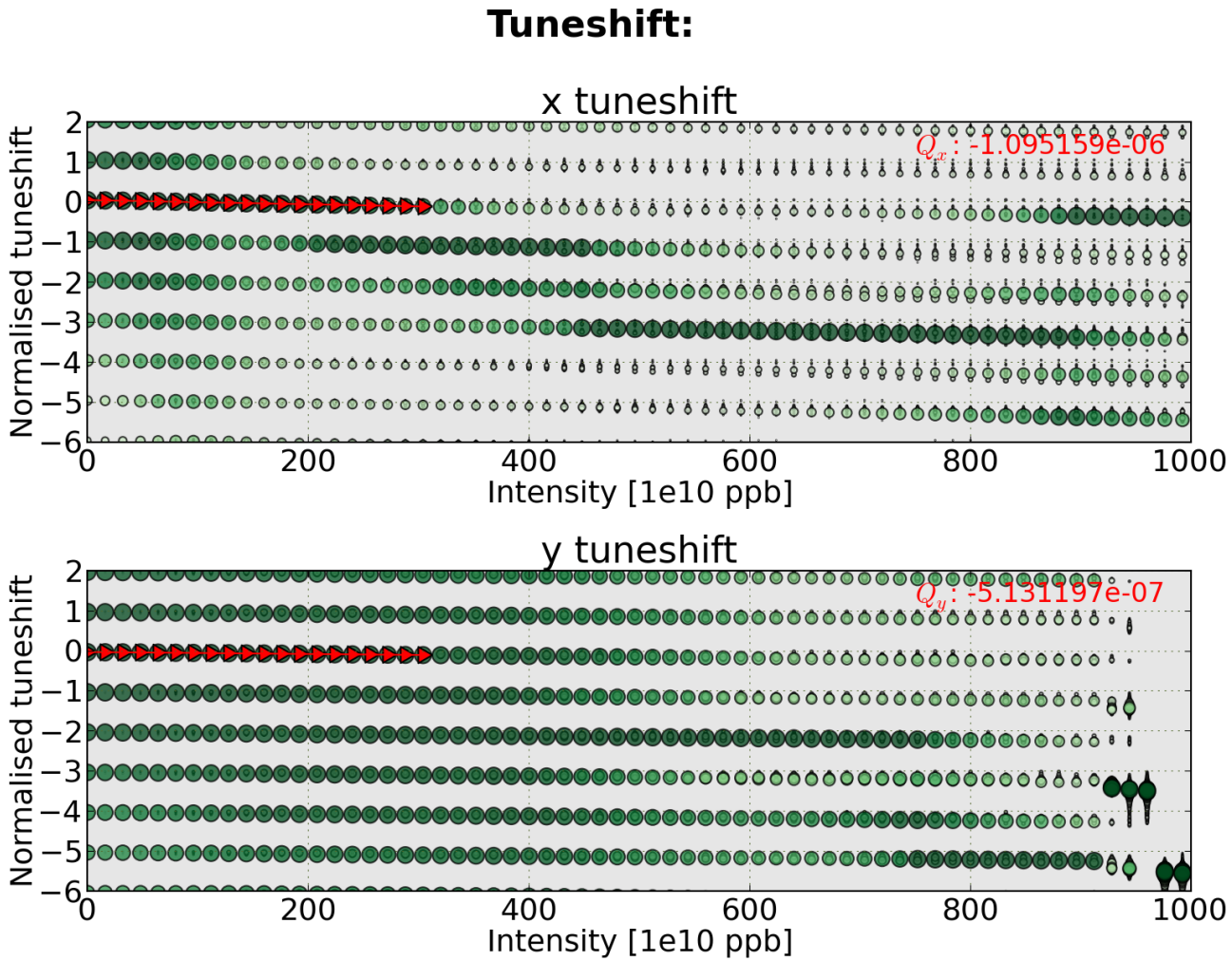
Measurement & simulation results

PSP bunch modes assuming Z=1 MΩ/m:



Measurement & simulation results

PSB tuneshift diagram for Z=1 MΩ/m



Outlook

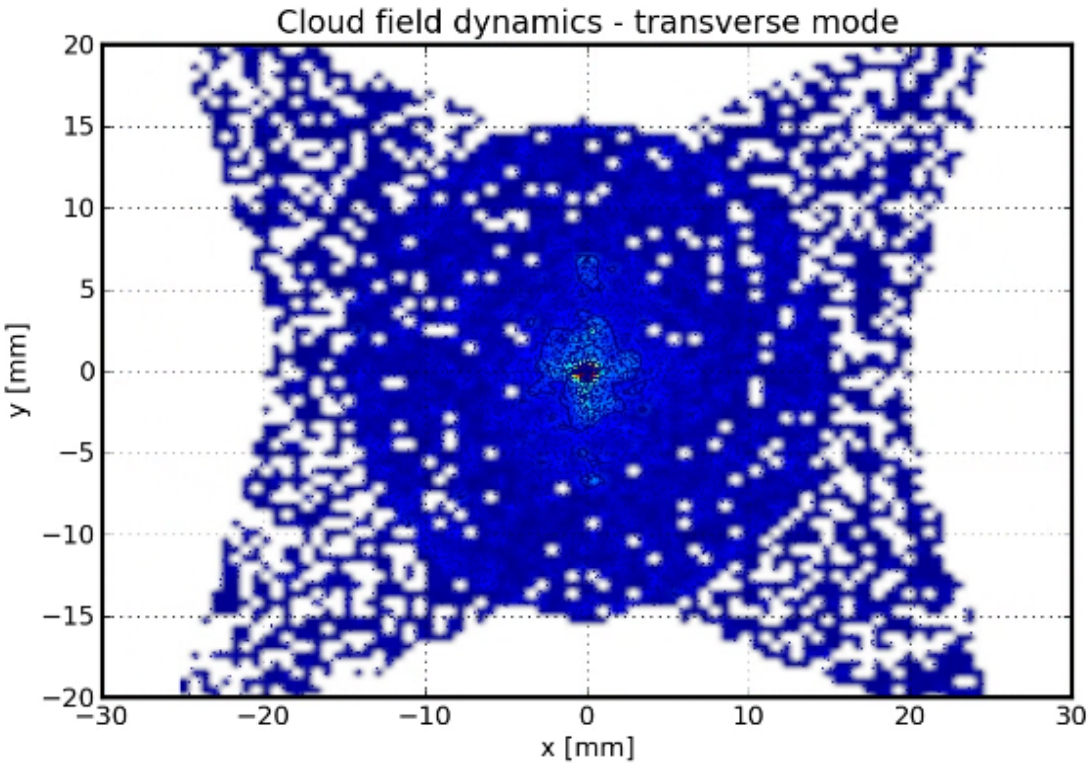
- Perform an impedance scan using rectangular beam pipe geometry (new version HeadTail)
  $\longrightarrow Z_t$
- Decompose bunch mode spectrum
- Perform an impedance frequency and value scan to identify narrow band impedances responsible for head-tail instabilities in the PSB
  $\longrightarrow Z_t\ , \omega_r$

Purpose of the analysis

- Quantification of coherent and incoherent tuneshift & tunespread for different electron cloud models
- Distribution and dynamics of tunefootprint to gain insight into slow losses in the SPS

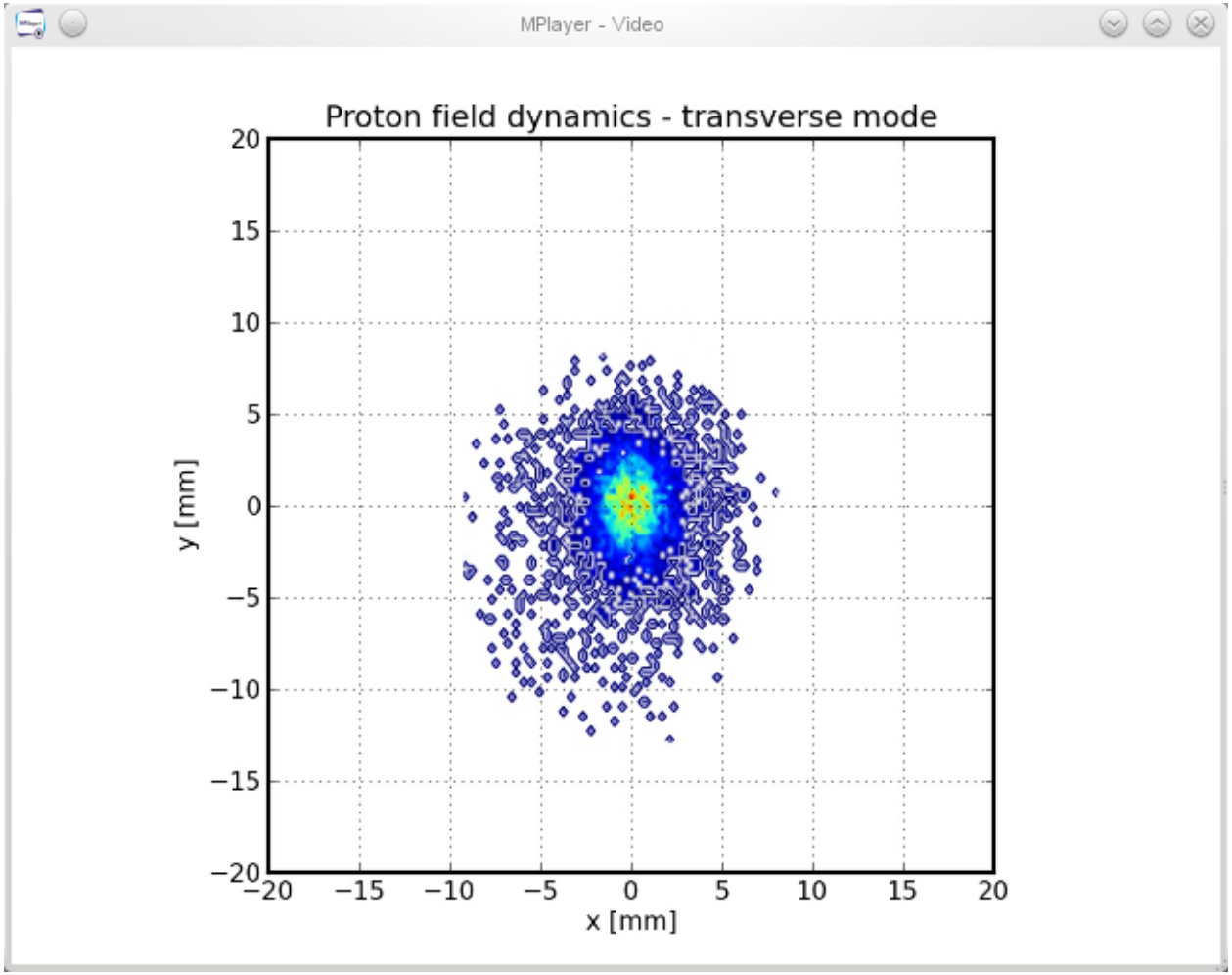
SPS e-cloud density: 100e10 m^{-3}; standard

Electron cloud dynamics:



SPS e-cloud density: 100e10 m^{-3}; standard

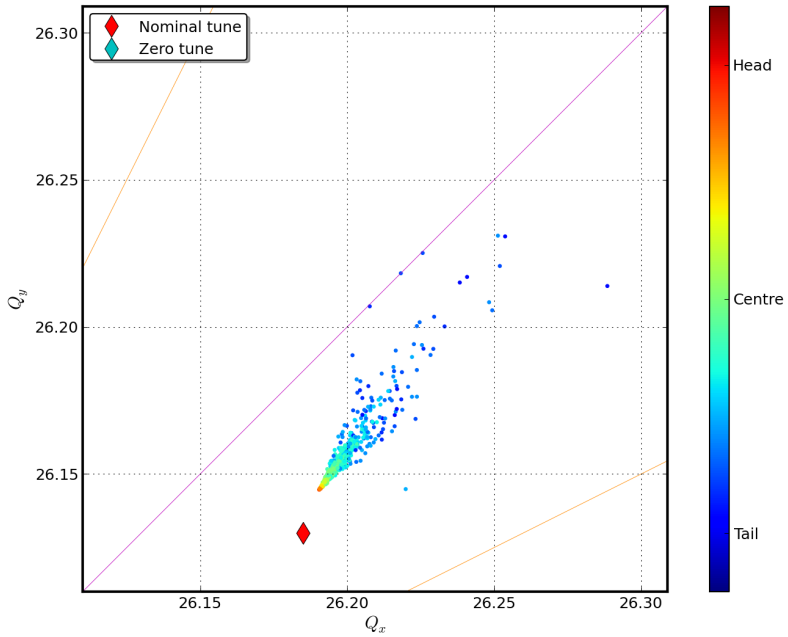
Proton dynamics:



SPS e-cloud density: 100e10 m^{-3}; standard

Tune footprint

Electron cloud 2D dynamics - Electrons: 1.00e+12/m^3, Protons: 4.40e+09

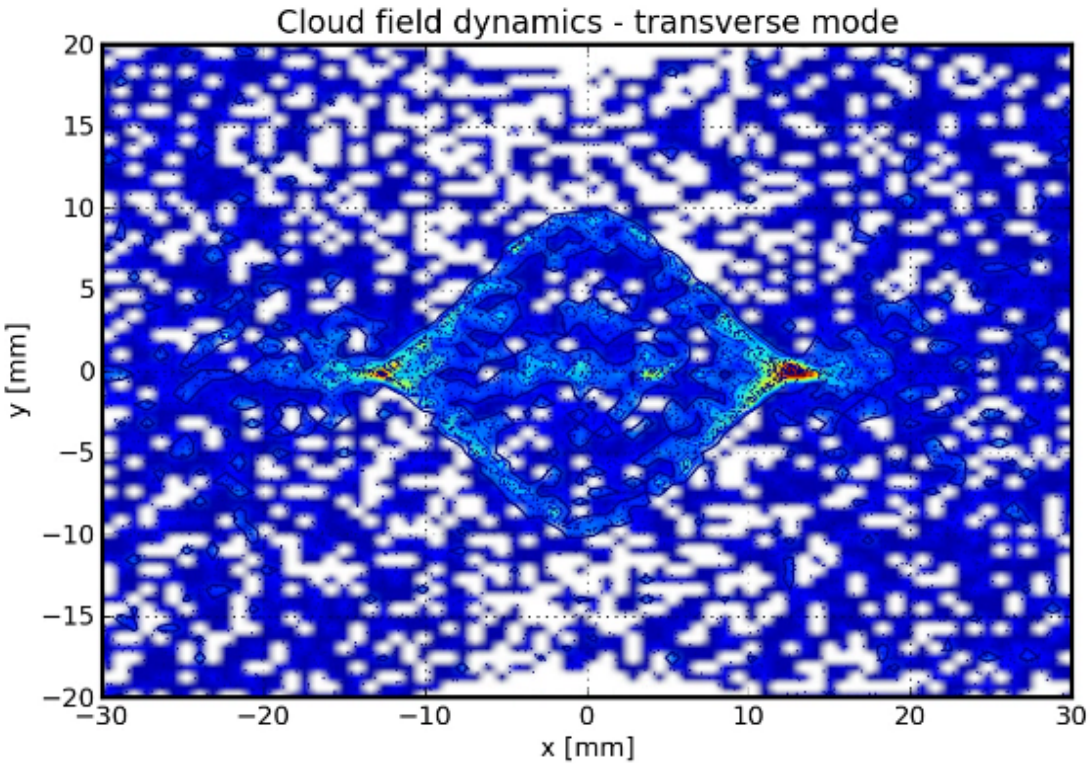


Approximate values at $N_p = 10^{11}$ ppb:

	Coherent	Spread
ΔQ_x [1e-2]	0.5	6.5
ΔQ_y [1e-2]	1.5	10.5

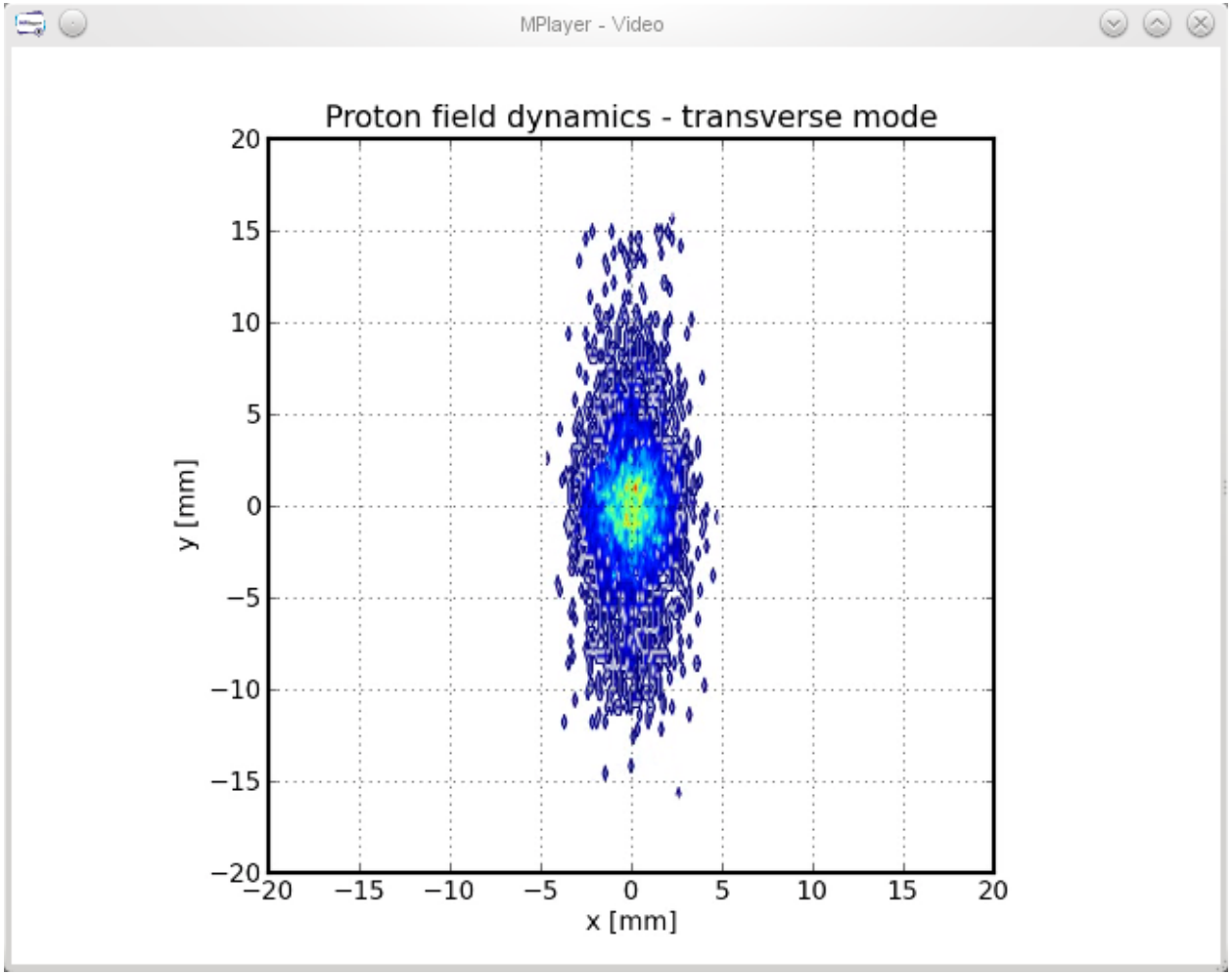
SPS e-cloud density: 100e10 m^{-3}; dipole

Electron cloud dynamics:



SPS e-cloud density: 100e10 m^{-3}; dipole

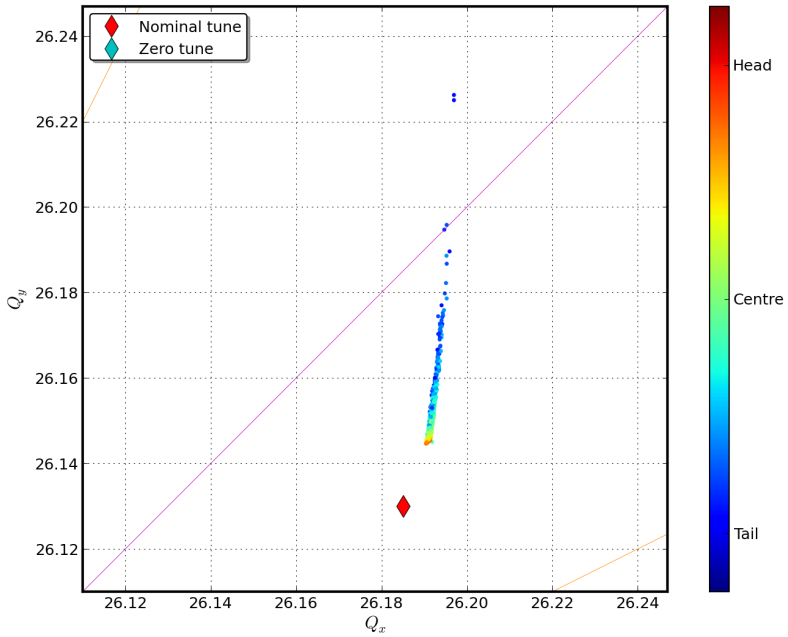
Proton dynamics:



SPS e-cloud density: 100e10 m^{-3}; dipole

Tune footprint

Electron cloud 2D dynamics - Electrons: 1.00e+12/m^3, Protons: 4.40e+09

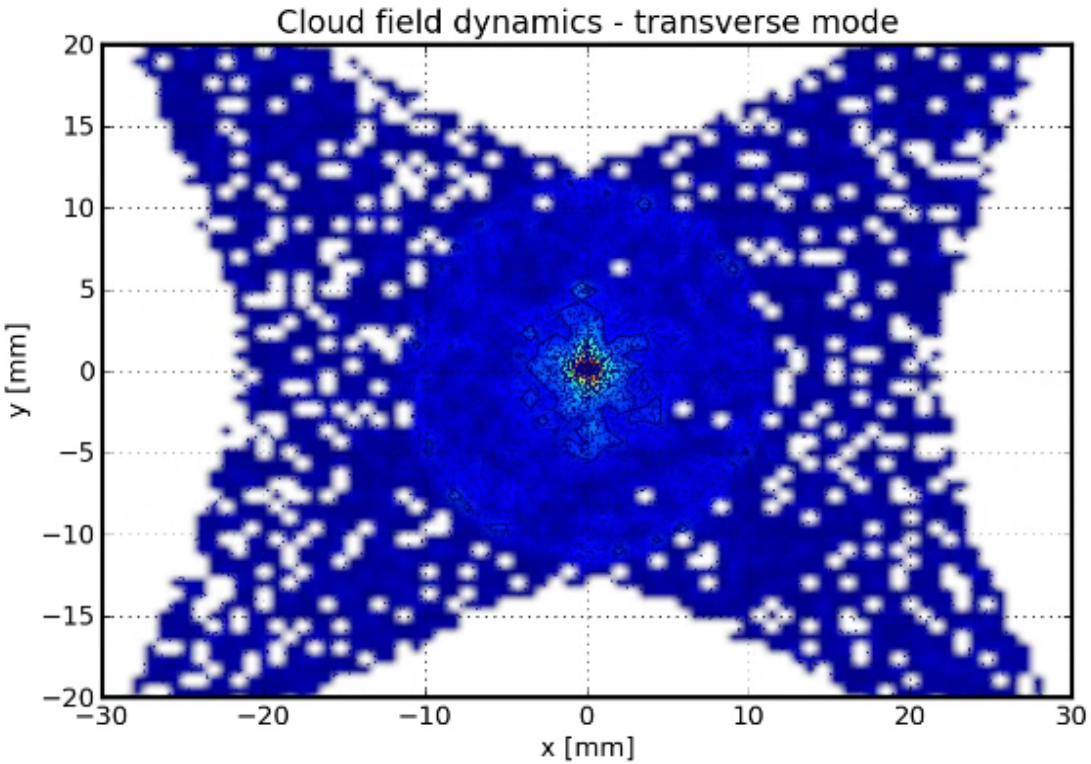


Approximate values at $N_p = 10^{11}$ ppb:

	Coherent	Spread
ΔQ_x [1e-2]	0.5	0.5
ΔQ_y [1e-2]	1.4	6.6

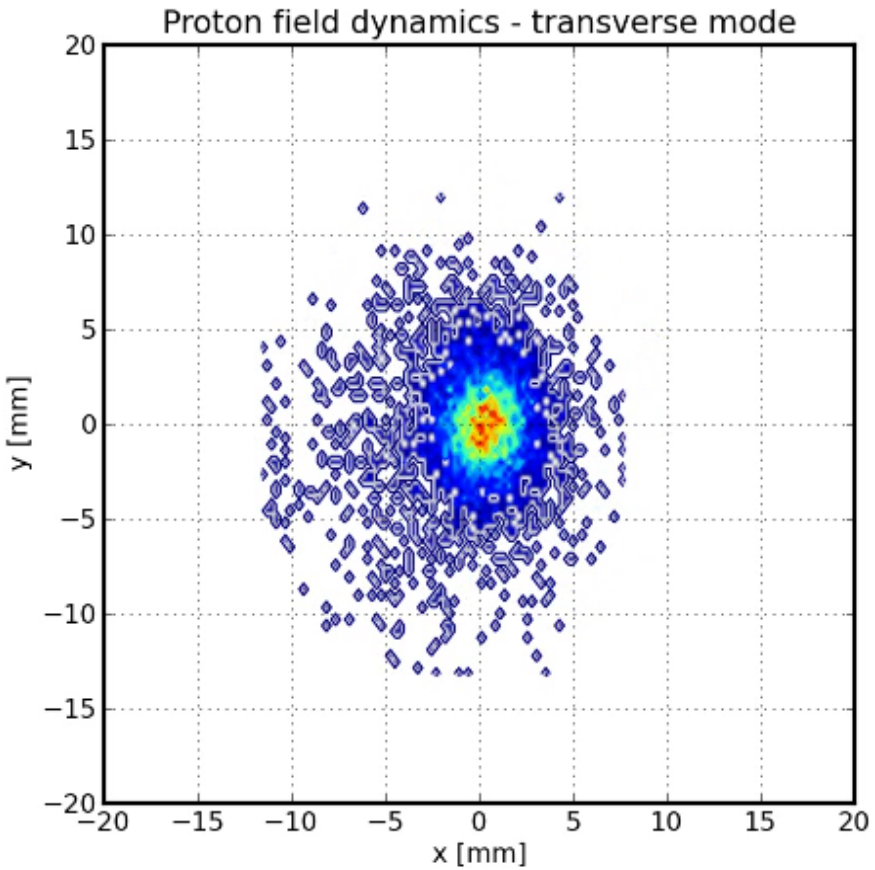
SPS e-cloud density: 100e10 m^{-3}; open boundaries

Electron cloud dynamics:



SPS e-cloud density: 100e10 m^{-3}; open boundaries

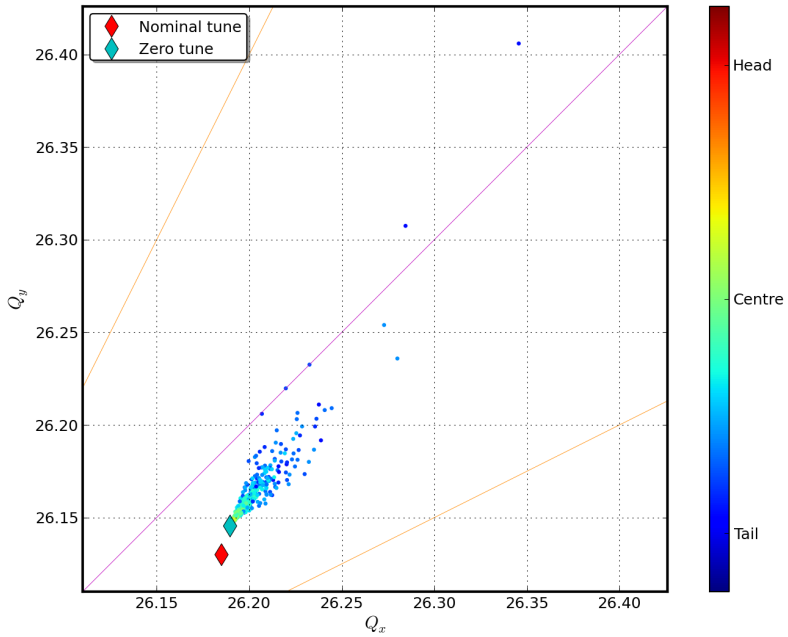
Proton dynamics:



SPS e-cloud density: 100e10 m^{-3}; open boundaries

Tune footprint

Electron cloud 2D dynamics - Electrons: 1.00e+12/m^3, Protons: 4.40e+09

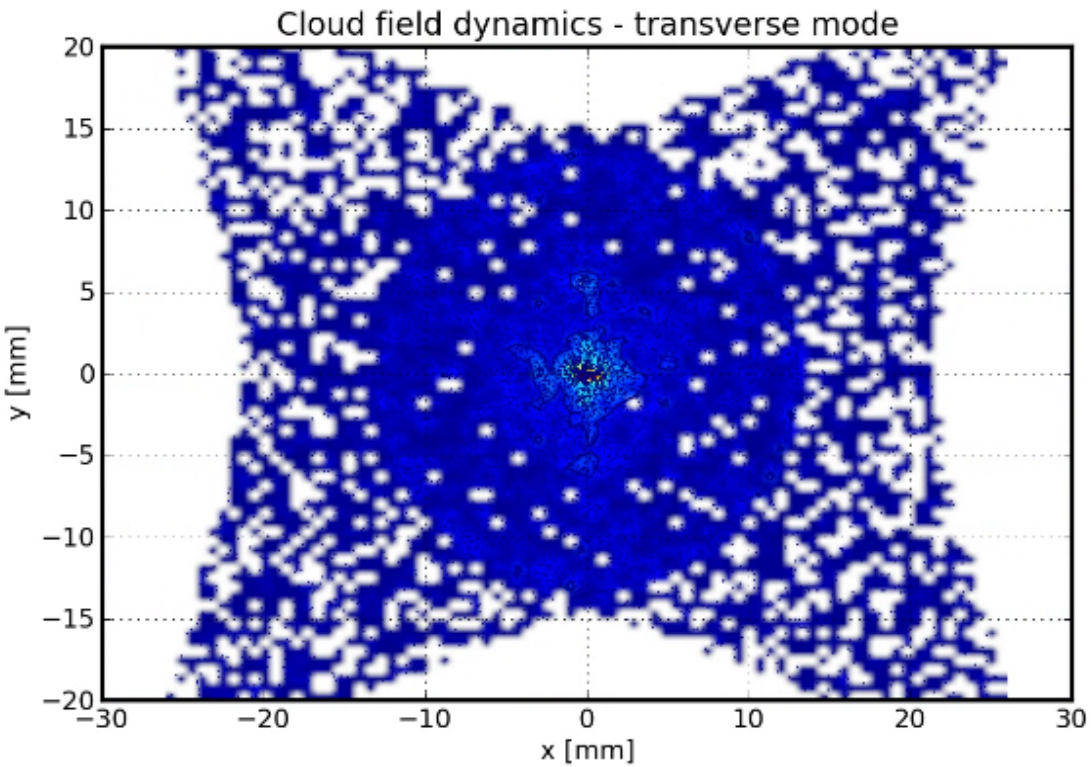


Approximate values at $N_p = 10^{11}$ ppb:

	Coherent	Spread
ΔQ_x [1e-2]	0.5	6.0
ΔQ_y [1e-2]	1.6	10.4

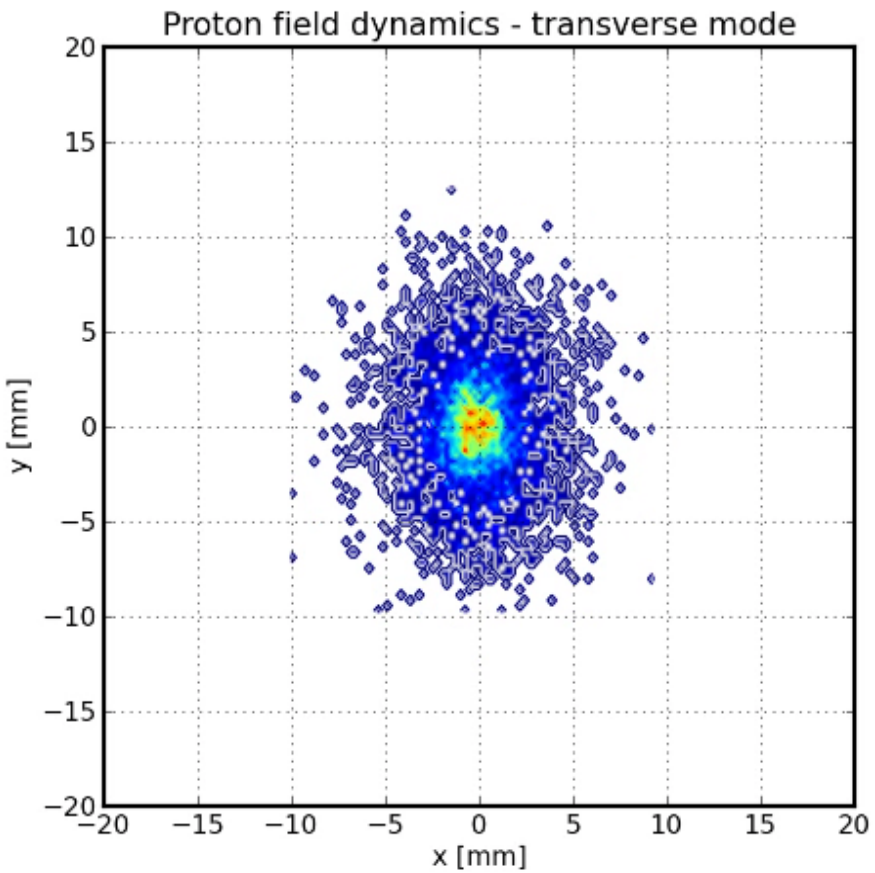
SPS e-cloud density: 100e10 m^{-3}; with synchrotron motion

Electron cloud dynamics:



SPS e-cloud density: 100e10 m^{-3}; with synchrotron motion

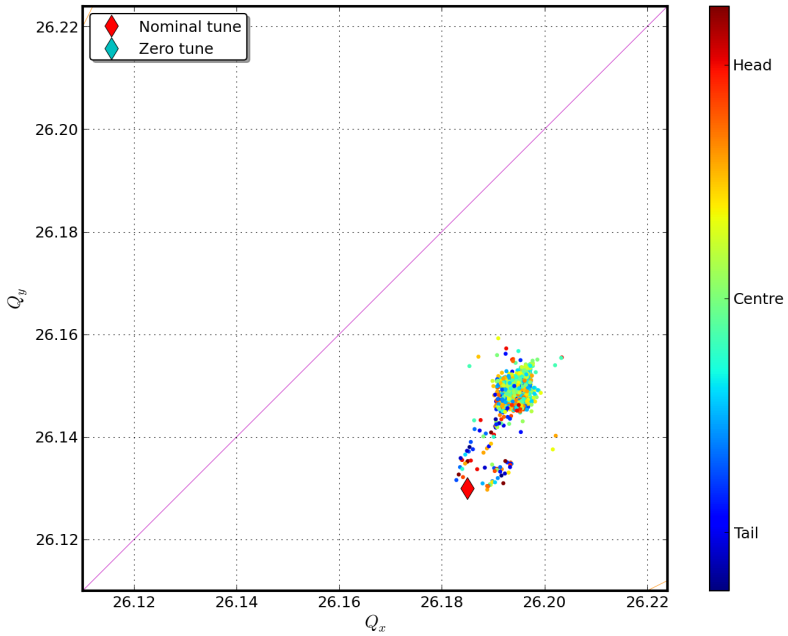
Proton dynamics:



SPS e-cloud density: 100e10 m^{-3}; with synchrotron motion

Tune footprint

Electron cloud 2D dynamics - Electrons: 1.00e+12/m^3, Protons: 4.40e+09



Approximate values at $N_p = 10^{11}$ ppb:

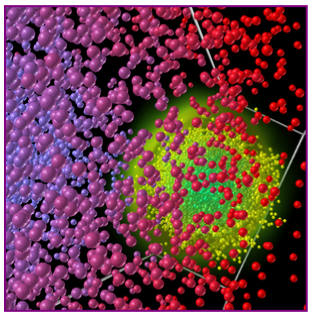
	Coherent	Spread
ΔQ_x [1e-2]	–	1.5
ΔQ_y [1e-2]	–	2.5

Outlook

- Outsource (parallelise) the Sussix analysis (could use some help here...)
- Store data in HDF5 data format using H5Part (PSI/LBNL)

H5Part: a Portable High Performance Parallel Data Interface to HDF5

Motivation



Beam-beam collision simulation.

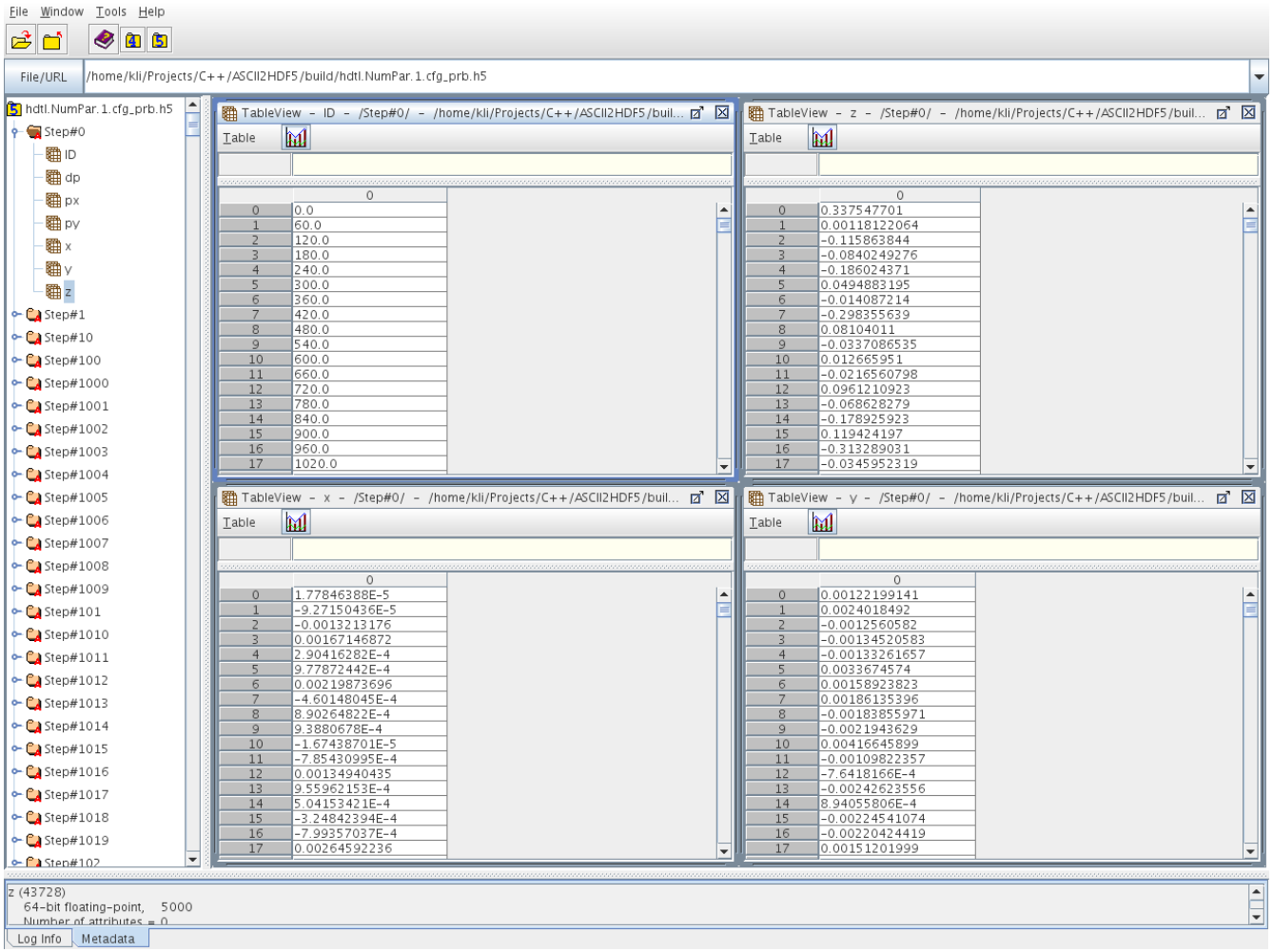
Particle based simulations of accelerator beam-lines, especially in six dimensional phase space, generate vast amounts of data. Even though a subset of statistical information regarding phase space or analysis needs to be preserved, reading and writing such enormous restart files on massively parallel supercomputing systems remains challenging.

H5Part is a very simple data storage schema and provides an API that simplifies the reading/writing of the data to the HDF5 file format. An important foundation for a stable visualization and data analysis environment is a stable and portable file storage format and its associated APIs. The presence of a "common file storage format," including associated APIs, will help foster a fundamental level of interoperability across the project's software infrastructure. It will also help ensure that key data analysis capabilities are present during the earliest phases of the software development effort.

H5Part is built on top of the HDF5 (Hierarchical Data Format). HDF5 offers a self-describing machine-independent binary file format that supports scalable parallel I/O performance for MPI codes on a variety of supercomputing systems, and works equally well on laptop computers. The API is available for C, C++, and Fortran codes. The H5Part file format and APIs enable disparate research groups with different simulation implementations to transparently share datasets and data analysis tools. For instance, the common file format will enable groups that depend on completely different simulation implementations to share data analysis tools.

H5Part file organization and API

Outlook


File Window Tools Help
File/URL
/home/kli/Projects/C++/ASCII2HDF5/build/hdtl.NumPar.1.cfg_prb.h5
hdtl.NumPar.1.cfg_prb.h5
Step#0
ID
dp
px
py
x
y
z
Step#1
Step#10
Step#100
Step#1000
Step#1001
Step#1002
Step#1003
Step#1004
Step#1005
Step#1006
Step#1007
Step#1008
Step#1009
Step#101
Step#1010
Step#1011
Step#1012
Step#1013
Step#1014
Step#1015
Step#1016
Step#1017
Step#1018
Step#1019
Step#102
TableView - ID - /Step#0/ - /home/kli/Projects/C++/ASCII2HDF5/buil...
TableView - z - /Step#0/ - /home/kli/Projects/C++/ASCII2HDF5/buil...
TableView - x - /Step#0/ - /home/kli/Projects/C++/ASCII2HDF5/buil...
TableView - y - /Step#0/ - /home/kli/Projects/C++/ASCII2HDF5/buil...
Table
z (43728)
64-bit floating-point, 5000
Log Info
Metadata

End

Thank you!