

A HeadTail Analysis Tool

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



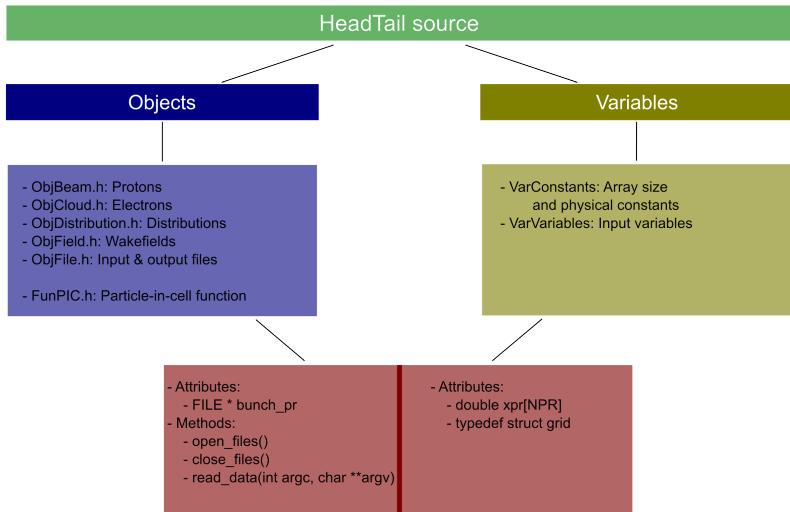
September 1, 2010

Outline

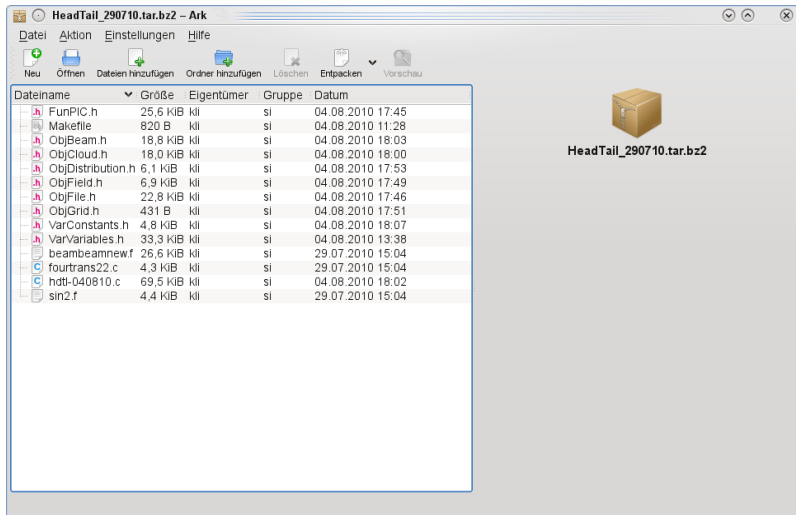
- 1 HeadTail development
- 2 HeadTail LSF submission tools
- 3 A PYthon HEadtail ANALysis tool
 - Output samples - PSB broadband impedance studies
 - Output samples - SPS e-cloud studies



Proposed file structure for future development



Proposed file structure for future development



The screenshot shows a file manager window titled "HeadTail_290710.tar.bz2 - Ark". The window displays a list of files and folders extracted from the archive. The files are organized into a directory structure. The main content area shows a list of files with columns for "Dateiname", "Größe", "Eigentümer", "Gruppe", and "Datum".

Dateiname	Größe	Eigentümer	Gruppe	Datum
FunPIC.h	25,6 KiB	kli	si	04.08.2010 17:45
Makefile	820 B	kli	si	04.08.2010 11:28
ObjBeam.h	18,8 KiB	kli	si	04.08.2010 18:03
ObjCloud.h	18,0 KiB	kli	si	04.08.2010 18:00
ObjDistribution.h	6,1 KiB	kli	si	04.08.2010 17:53
ObjField.h	6,9 KiB	kli	si	04.08.2010 17:49
ObjFile.h	22,8 KiB	kli	si	04.08.2010 17:46
ObjGrid.h	431 B	kli	si	04.08.2010 17:51
VarConstants.h	4,8 KiB	kli	si	04.08.2010 18:07
VarVariables.h	33,3 KiB	kli	si	04.08.2010 13:38
beambeamnew.f	26,6 KiB	kli	si	29.07.2010 15:04
fourtrans22.c	4,3 KiB	kli	si	29.07.2010 15:04
hdti-040810.c	69,5 KiB	kli	si	04.08.2010 18:02
sin2.f	4,4 KiB	kli	si	29.07.2010 15:04

On the right side of the window, there is a large icon of a cardboard box representing the archive, with the text "HeadTail_290710.tar.bz2" below it.

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 "ObjFIC.h"  
#include "ObjBeam.h"  
#include "ObjCloud.h"  
#include "ObjDistribution.h"  
#include "ObjField.h"  
#include "ObjFile.h"  
#include "ObjGeometry.h"  
#include "ObjVariables.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)

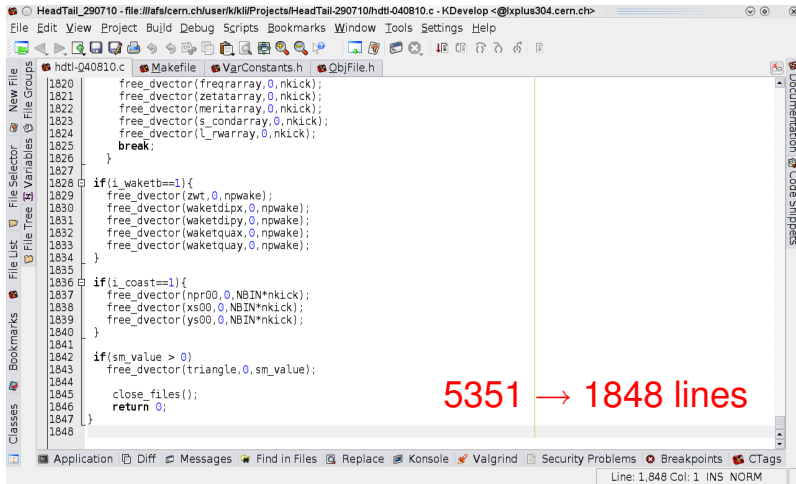


Possible advantages

- 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



The screenshot shows a code editor window titled "HeadTail_290710 - file:///afs/cern.ch/user/k/klil/Projects/HeadTail-290710/hdtl-040810.c - KDevelop <@xplus304.cern.ch>". The code is in C and includes several conditional blocks. A red arrow points from the text "5351 -> 1848 lines" to line 1848 in the code. The code is as follows:

```
1820     free_dvector(freqrarray,0,nkick);
1821     free_dvector(zetatarray,0,nkick);
1822     free_dvector(meritarray,0,nkick);
1823     free_dvector(s_condarray,0,nkick);
1824     free_dvector(l_rwarray,0,nkick);
1825     break;
1826 }
1827
1828 if(i_waketb==1){
1829     free_dvector(zwt,0,npwake);
1830     free_dvector(waketdpx,0,npwake);
1831     free_dvector(waketdpy,0,npwake);
1832     free_dvector(waketquax,0,npwake);
1833     free_dvector(waketquay,0,npwake);
1834 }
1835
1836 if(i_coast==1){
1837     free_dvector(npr00,0,NBIN*nkick);
1838     free_dvector(xs00,0,NBIN*nkick);
1839     free_dvector(ys00,0,NBIN*nkick);
1840 }
1841
1842 if(sm_value > 0)
1843     free_dvector(triangle,0,sm_value);
1844
1845     close_files();
1846     return 0;
1847 }
1848
```

Line: 1,848 Col: 1 INS NORM



Inputfile argument

Note! Slight modification: input file is now passed as argument
 → 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, "ecspcs");
    //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\n");
        exit (2);
    }
}

```



SUB-HDTL/SUB-ECLD

Fast and simple script that makes use of LSF job arrays
→ (quasi-)parallel execution 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-CDc16-22-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 pcbel3664: /backup/CERN/PSB-bbw-C0c16-Z2-TEST-NumPar/

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



SUB-HDTL/SUB-ECLD usage

Datei	Bearbeiten	Ansicht	Verlauf	Lesezeichen	Einstellungen	Hilfe					
87899580	k.li	RUN	lnw	lxplu240	lxbst2228	*array[69]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsu1229	*array[70]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsq0529	*array[71]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsu1314	*array[72]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbst2225	*array[73]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbra3903	*array[74]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsq1311	*array[75]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsu1305	*array[76]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsp0944	*array[77]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsq1518	*array[78]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsp0801	*array[79]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsq1219	*array[80]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbst2110	*array[81]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbs304	*array[82]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsq1435	*array[83]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbra4107	*array[84]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsp0714	*array[85]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsp0738	*array[86]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsp0703	*array[87]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsq1403	*array[88]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsq0543	*array[89]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbst2125	*array[90]	Aug	5	13:29		
87899705	k.li	RUN	lnw	lxplu240	lxbsq1127	*_array[1]	Aug	5	13:29		
87899705	k.li	RUN	lnw	lxplu240	lxbsq1329	*_array[2]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsu0847	*array[91]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsp0715	*array[92]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsq1441	*array[93]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsq1228	*array[94]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsq1433	*array[95]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsp0712	*array[96]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsu1220	*array[97]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsu1409	*array[98]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsq0829	*array[99]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsq1439	*rray[100]	Aug	5	13:29		
87899580	k.li	RUN	lnw	lxplu240	lxbsq1230	*rray[101]	Aug	5	13:29		
87899705	k.li	RUN	lnw	lxplu240	lxbsu1401	*_array[3]	Aug	5	13:29		
87899705	k.li	RUN	lnw	lxplu240	lxbsu1148	*_array[4]	Aug	5	13:29		
87899705	k.li	RUN	lnw	lxplu240	lxbsu1125	*_array[5]	Aug	5	13:29		

lxplus:



SUB-HDTL/SUB-ECLD output

Output directories and files

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



SUB-HDTL/SUB-ECLD output

Name	Größe	Datum
hdtl NumPar.10.cfg_pro.dat	113,2 KiB	05.08.2010 12:10
hdtl NumPar.10.cfg_prt.dat	3,8 MiB	05.08.2010 12:15
hdtl NumPar.10.cfg_sampl.dat	0 B	05.08.2010 12:15
hdtl NumPar.10.cfg_sample.dat	386,6 KiB	05.08.2010 12:15
hdtl NumPar.10.cfg_trk.dat	2,9 MiB	05.08.2010 12:15
hdtl NumPar.11.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.11.cfg_bunchds.dat	949,4 KiB	05.08.2010 12:15
hdtl NumPar.11.cfg_hdtl.dat	2,6 MiB	05.08.2010 12:15
hdtl NumPar.11.cfg_inph.dat	484 B	05.08.2010 12:15
hdtl NumPar.11.cfg_pdst.dat	0 B	05.08.2010 12:15
hdtl NumPar.11.cfg_pini.dat	235,2 KiB	05.08.2010 12:15
hdtl NumPar.11.cfg_prb.dat	113,2 MiB	05.08.2010 12:15
hdtl NumPar.11.cfg_prt.dat	3,8 MiB	05.08.2010 12:15
hdtl NumPar.11.cfg_sampl.dat	0 B	05.08.2010 12:15
hdtl NumPar.11.cfg_sample.dat	386,6 KiB	05.08.2010 12:15
hdtl NumPar.11.cfg_trk.dat	2,9 MiB	05.08.2010 12:15
hdtl NumPar.12.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.13.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.14.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.15.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.16.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.17.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.18.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.19.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.20.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.21.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.22.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.23.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.24.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.25.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.26.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.27.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.28.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.29.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.30.cfg	5,0 KiB	05.08.2010 13:26
hdtl NumPar.31.cfg	5,0 KiB	05.08.2010 13:26

214 Dateien (1,3 GiB)



Pyheana

A PYthon HEadtail ANALysis tool
or...



Pyheana

A Pyheana!



Pyheana - usage: interactive

User panel

The screenshot shows the Pyheana user panel interface, which is a window titled "Pyheana". The interface is divided into several sections:

- File Selection:** Contains an "Open" button.
- Range Selection:** Includes a radio button for "Index" (selected) and a text input field containing "1". Below it is a radio button for "Range". At the bottom, there are three input fields labeled "Min:", "Max:", and "Step:", with values "1", "100", and "1" respectively.
- Machine Selection:** Includes two radio buttons: "PSB" (selected) and "SPS".
- Mode Selection:** Includes a grid of checkboxes for various simulation modes:
 - Bunch Dynamics
 - Cloud Dynamics
 - Proton Dynamics
 - Tune (FFT)
 - Tune Footprint
 - Tune Shift
 - Trajectory
 - Bunch Movie
 - Cloud Movie
 - Proton Movie
 - Bunch Super (checked)
- Run:** A large "Run" button at the bottom center.

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 \rightarrow calculate from pipe geometry
- shunt impedance \rightarrow 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:

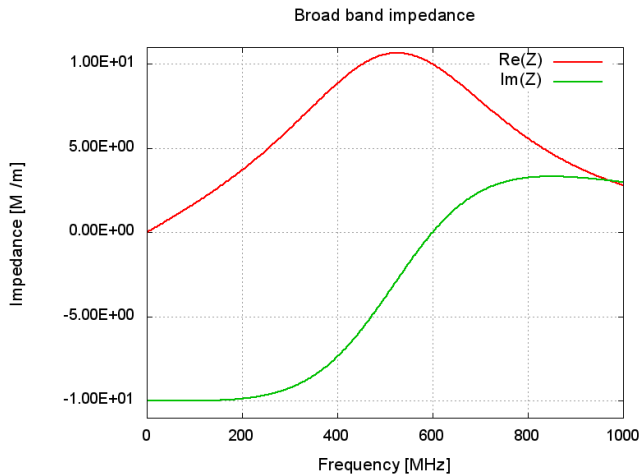
$$\omega_r = c \cdot \frac{x_{mn}}{b} \approx \frac{c}{b}, \quad \text{circular}$$

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



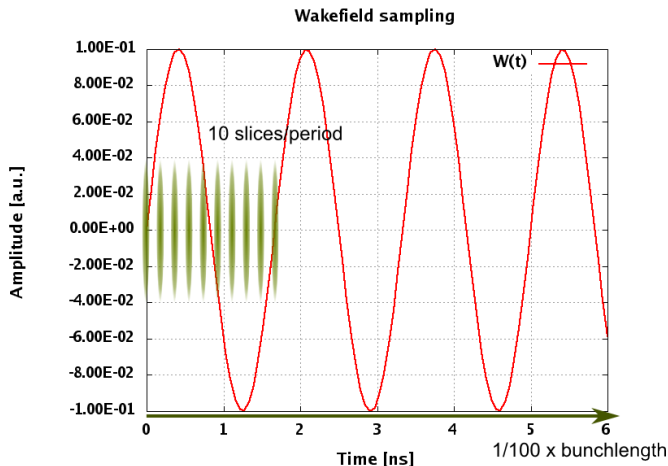
Broadband impedance model

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:

$$\frac{T_{\text{wake}}}{T_{\text{slice}}} = \frac{2\pi b N_{\text{slice}}}{c T_b} = \frac{N_{\text{slice}}}{f T_b} \approx 10 \Rightarrow N_{\text{slice}} = 10 \times f T_b = 3'600$$

- 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!



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 parameters - general settings

Machine parameters

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

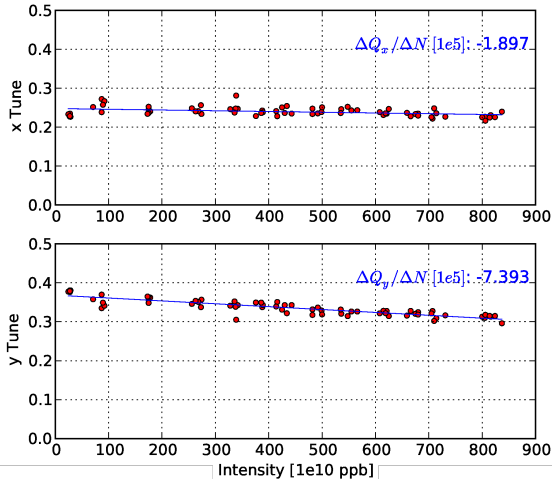
Simulation parameters

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

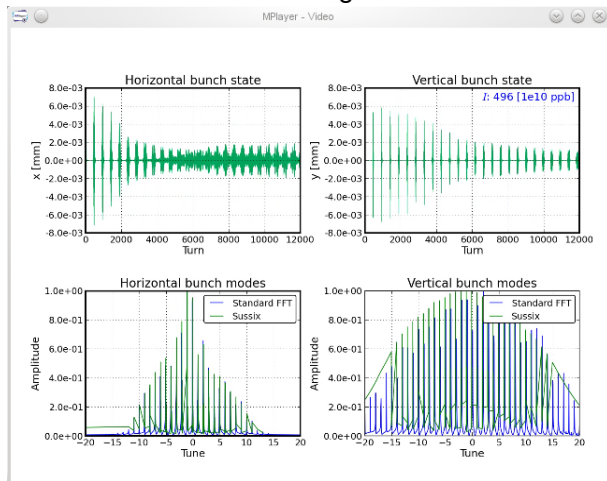


	Plot	D.Q.*
$Q_x [1e-5]$	-1.90	—
$Q_y [1e-5]$	-7.39	-7.2

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

Measurement & simulation results

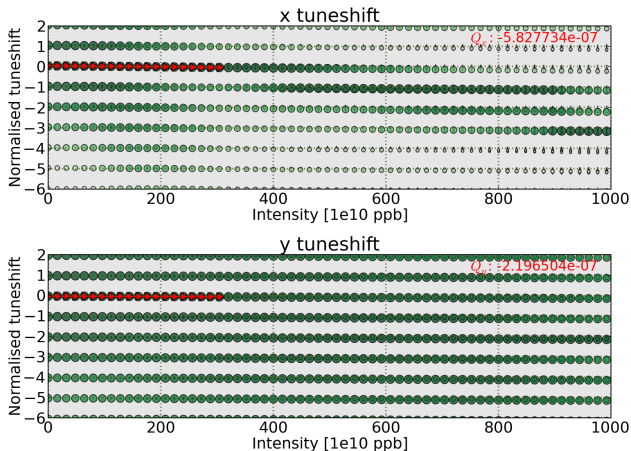
PSP bunch modes assuming $Z=0.5 \text{ M}\Omega/\text{m}$:



Measurement & simulation results

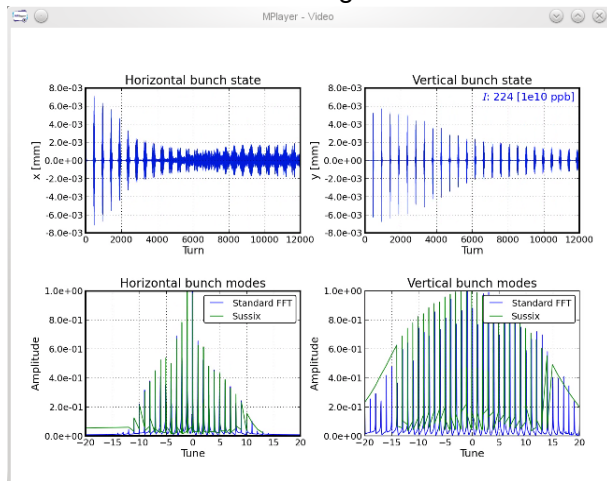
PSB tuneshift diagram for $Z=0.5 \text{ M}\Omega/\text{m}$

Tuneshift:



Measurement & simulation results

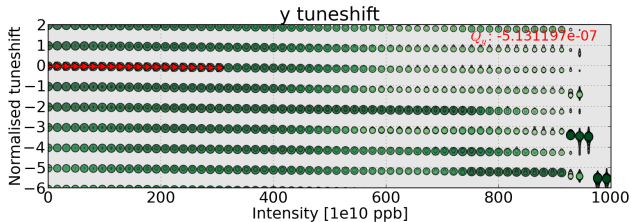
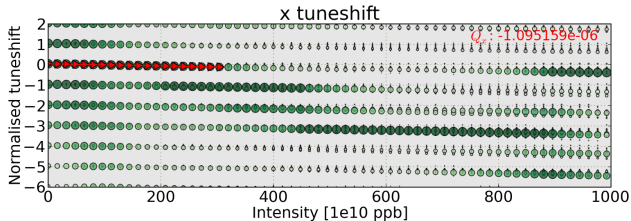
PSP bunch modes assuming $Z=1 \text{ M}\Omega/\text{m}$:



Measurement & simulation results

PSB tuneshift diagram for $Z=1 \text{ M}\Omega/\text{m}$

Tuneshift:



Outlook

- Perform an impedance scan using rectangular beam pipe geometry (new version HeadTail)
→ 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
→ Z_t, ω_r



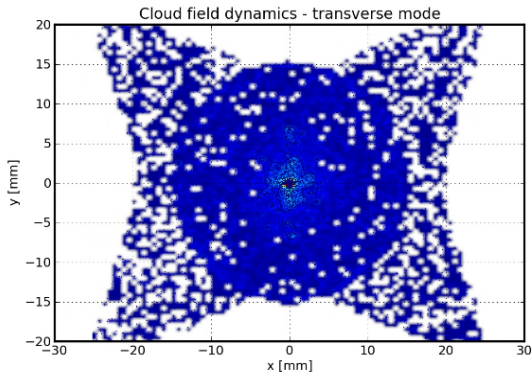
Purpose of the analysis

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



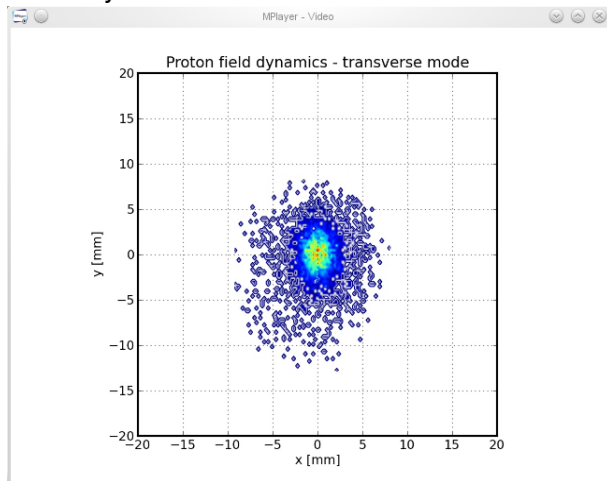
SPS e-cloud density: $100e10 \text{ m}^{-3}$; standard

Electron cloud dynamics:



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

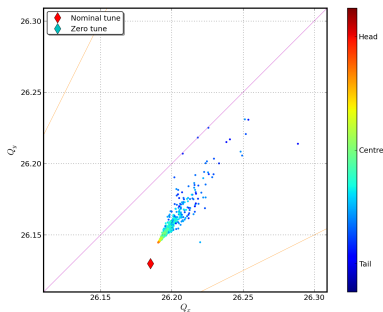
Proton dynamics:



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

Tune footprint

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

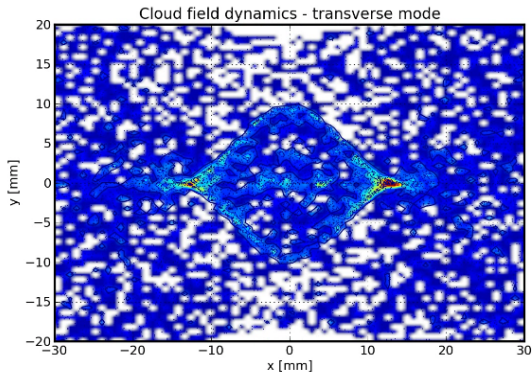


Approximate values at
 $N_p = 10^{11}$ ppb:

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

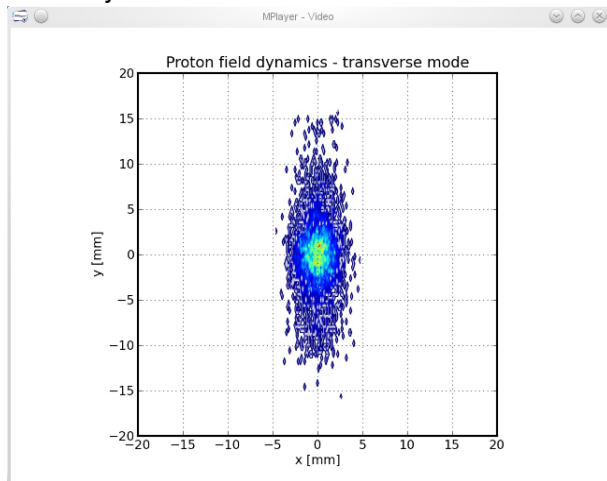
SPS e-cloud density: $100e10 \text{ m}^{-3}$; dipole

Electron cloud dynamics:



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

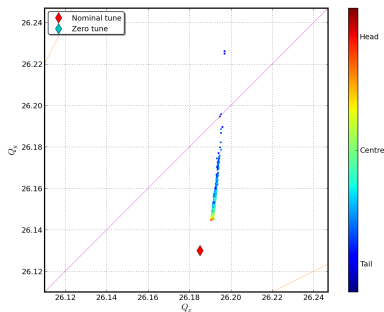
Proton dynamics:



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

Tune footprint

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



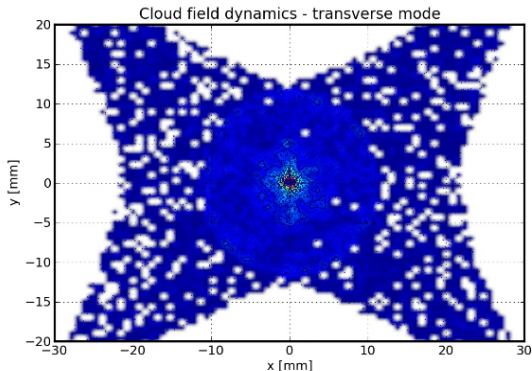
Approximate values at
 $N_p = 10^{11}$ ppb:

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



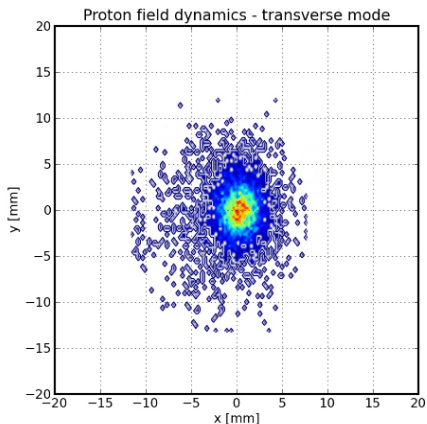
SPS e-cloud density: $100e10 \text{ m}^{-3}$; open boundaries

Electron cloud dynamics:



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

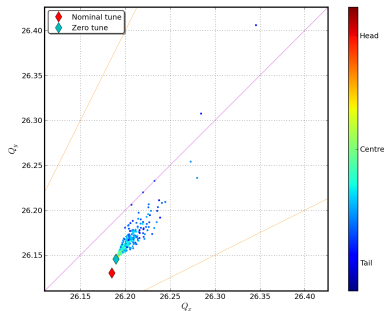
Proton dynamics:



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

Tune footprint

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

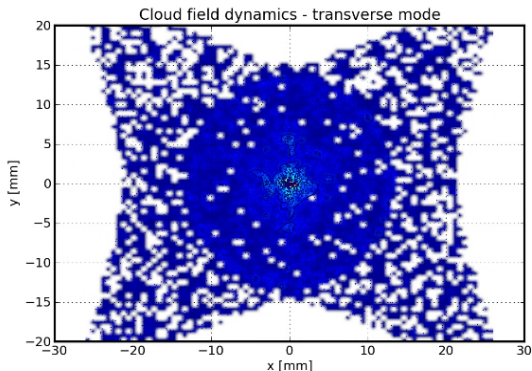


Approximate values at
 $N_p = 10^{11}$ ppb:

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

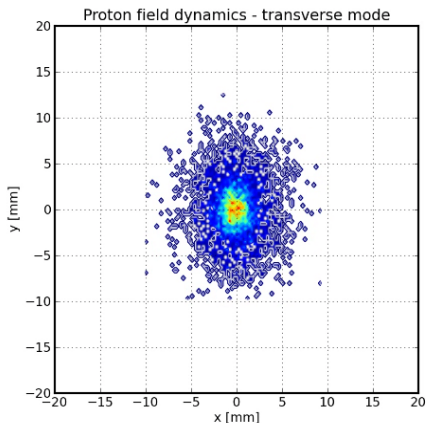
SPS e-cloud density: $100e10 \text{ m}^{-3}$; with synchrotron motion

Electron cloud dynamics:



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

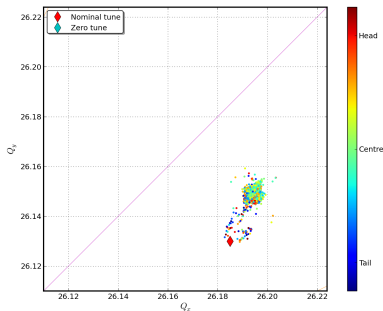
Proton dynamics:



SPS e-cloud density: $100e10 \text{ m}^{-3}$; 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:

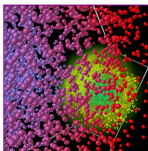
	Coherent	Spread
ΔQ_x [1e-2]	–	1.5
ΔQ_y [1e-2]	–	2.5

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

The screenshot shows the HeadTail software interface. On the left is a project tree for 'hdl NumPar. 1 cfp. prb h5'. The main area contains four data tables, each representing a different variable: ID, z, x, and y. Each table has 17 rows corresponding to steps from Step#0 to Step#17.

ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0.0																	
1	60.0																	
2	170.0																	
3	180.0																	
4	240.0																	
5	200.0																	
6	360.0																	
7	420.0																	
8	480.0																	
9	440.0																	
10	600.0																	
11	660.0																	
12	720.0																	
13	780.0																	
14	840.0																	
15	900.0																	
16	960.0																	
17	1020.0																	

z	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0.337547701																	
1	0.00118122064																	
2	-0.115951844																	
3	-0.0840249276																	
4	-0.195024371																	
5	0.0444883195																	
6	-0.014087214																	
7	-0.298155639																	
8	0.00104011																	
9	-0.0117089135																	
10	0.012665951																	
11	-0.0210560798																	
12	0.0961210923																	
13	-0.069628279																	
14	-0.173925193																	
15	0.119424197																	
16	-0.11139011																	
17	-0.0345952319																	

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1.77846388E-5																	
1	-9.27150416E-5																	
2	-0.0013213176																	
3	0.00167448872																	
4	2.90416282E-4																	
5	3.77872442E-4																	
6	0.00219873696																	
7	-4.60148045E-4																	
8	8.90264622E-4																	
9	3.3880678E-4																	
10	-1.67438701E-5																	
11	-7.85430995E-4																	
12	0.00134564935																	
13	9.55962153E-4																	
14	9.04152421E-4																	
15	-3.14842134E-4																	
16	-7.99357037E-4																	
17	0.00264529236																	

y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0.00122199141																	
1	0.0024018492																	
2	-0.0015460982																	
3	-0.003149170183																	
4	-0.00133261657																	
5	0.0033674874																	
6	0.00158923823																	
7	0.00186135396																	
8	-0.00181895971																	
9	-0.0021942629																	
10	0.0041664899																	
11	-0.00109822357																	
12	-7.6418166E-4																	
13	-0.00242613556																	
14	8.94055806E-4																	
15	-0.0024541074																	
16	-0.00220424419																	
17	0.00151201999																	

At the bottom, a status bar shows: F (43728) 64-bit floating-point, 5000 Number of attributes = 0. Log Info Metadata



End

Thank you!

