



NJOC activities summary

B. Viola

JET



This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

POSITION TITLE: KG1V CODE RESPONSIBLE OFFICER



Responsibilities:

- maintenance and development of the CHAIN-1 analysis code (KG1V) that analyses data from the multi-channel far infrared interferometer diagnostic (KG1) at JET, and for data validation of the KG1V data (fringe jump corrections).
- support the Integrated Modelling Group's modelling activities using plasma-edge and fully integrated plasma simulation codes, by utilizing and contributing to the development of the JINTRAC suite of transport codes



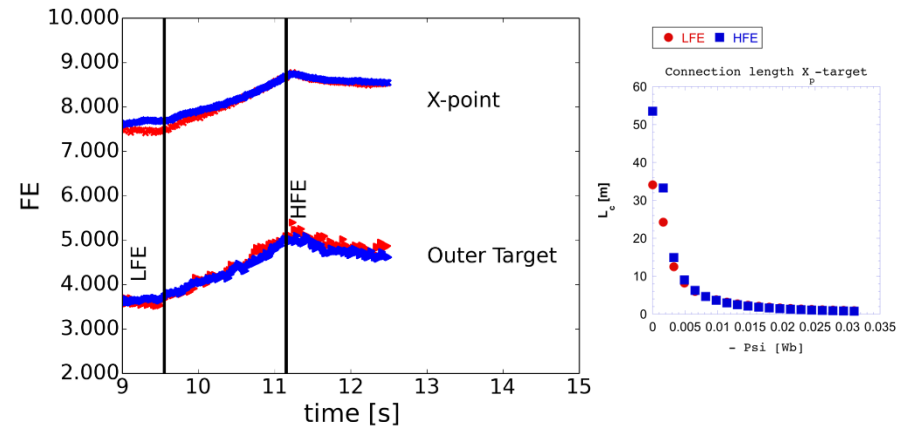
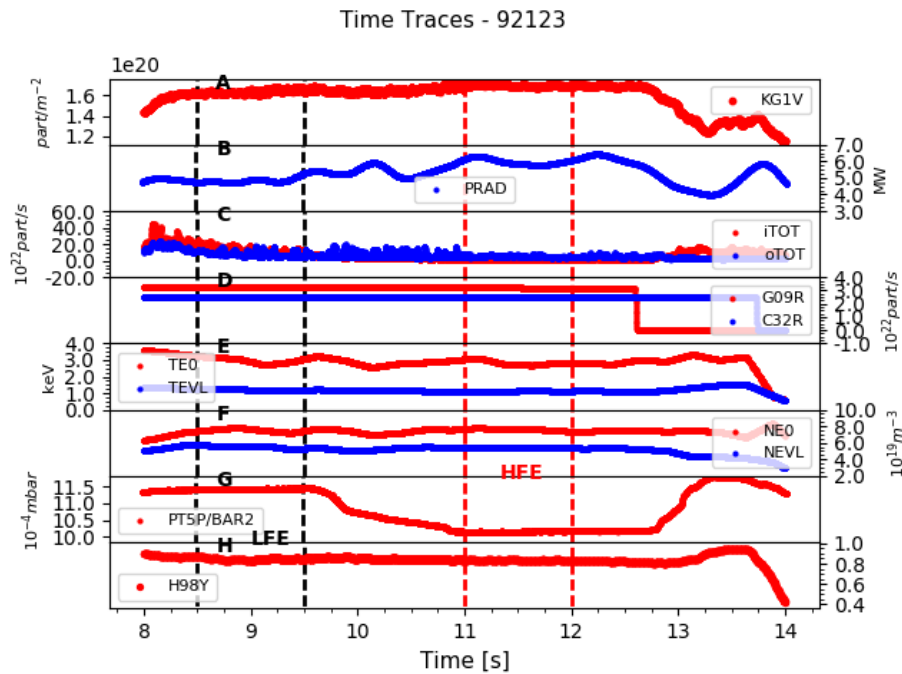
- KG1 **data** RO
- KG1 main software engineer
- Modelling activities in support of JET
- Software development activities in support of modellers
- EPROC_py library RO
- DVCM secretary
- DCO during JET experimental campaign

1 - Modelling activities

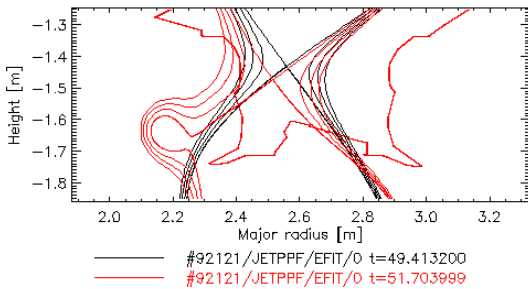


- Poloidal Flux expansion experiment + modelling
- M18-20 related modelling activities
- Software development for Modelling group

Result overview 1/3



Since $L_{\perp} \propto \tau_{dw}^{1/2} \propto L_c^{1/2}$ being L_c the total magnetic connection length and τ_{dw} the particle dwell time, we can pose $q_0 \Delta \Psi L_c^{1/2} |_{HFE} \sim q_0 \Delta \Psi L_c^{1/2} |_{REF}$. So we can expect an increase of divertor radiation equal to $f_r = (\Delta \Psi L_c^{1/2})_{HFE} / (\Delta \Psi L_c^{1/2})_{REF} = 1.22$

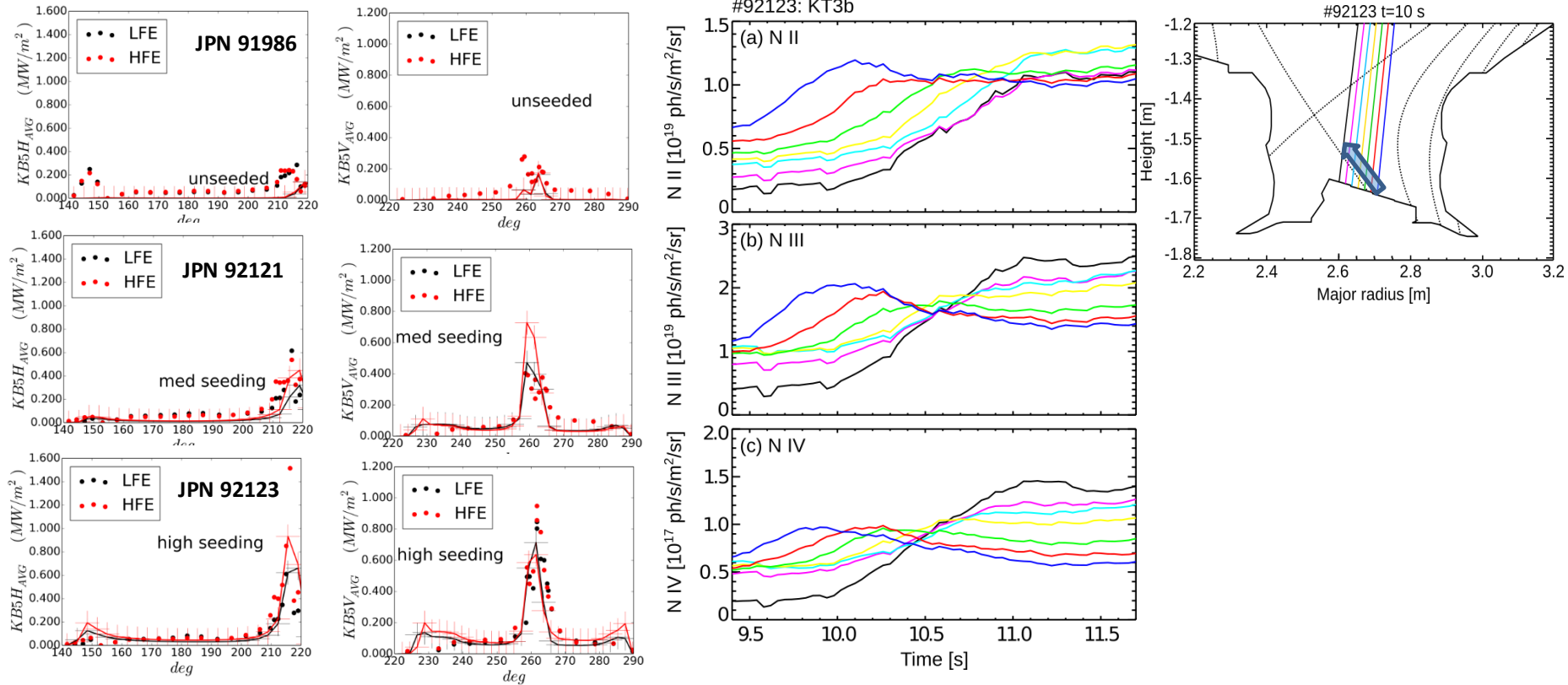


JPN	D2 (x10 ²² ele/s)	N2 (x10 ²² ele/s)	P _{NBI} (MW)	P _{RAD} (MW)	% Increase on Flux exp. X _p IN/OUT
92121	2.6	1.2	14.3	4.9	~22/~20
92123	2.5	3.3	14.4	6.7	~22/~20

X-point moved inwards!



Result overview 2/3

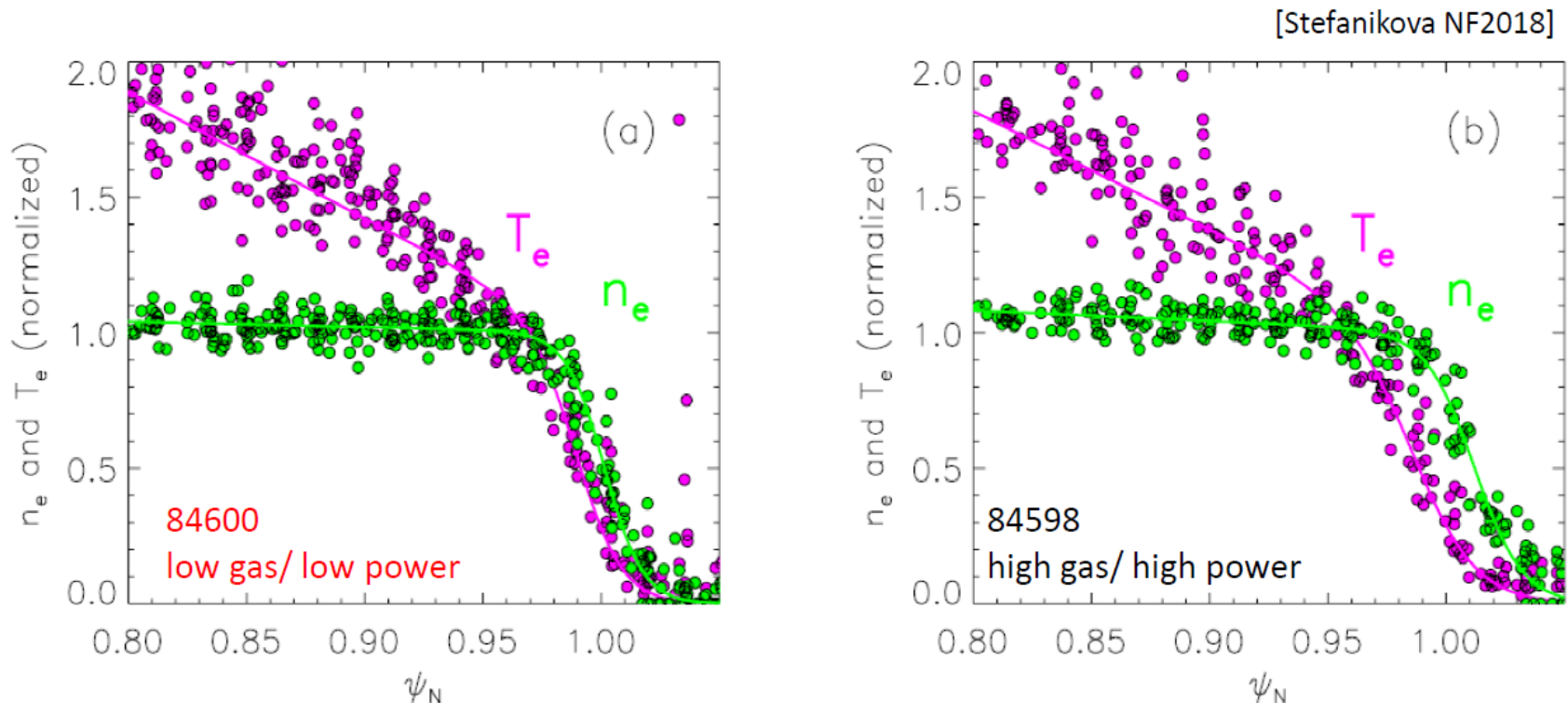


- Low charge nitrogen emission moves towards X-point as flux expansion increases
- N III and N IV moves closer to X-point in comparison to N II emission

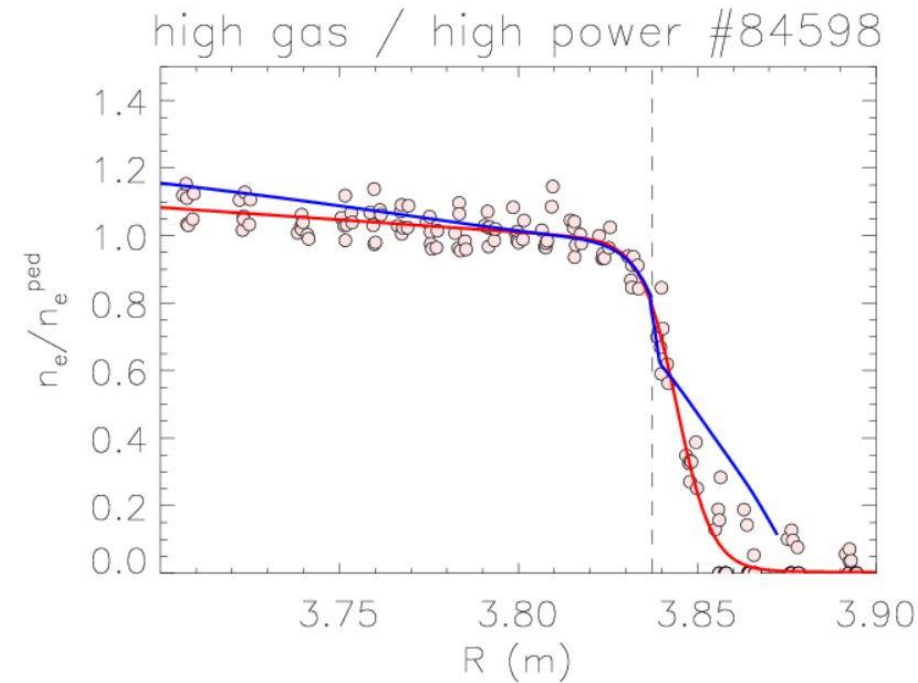
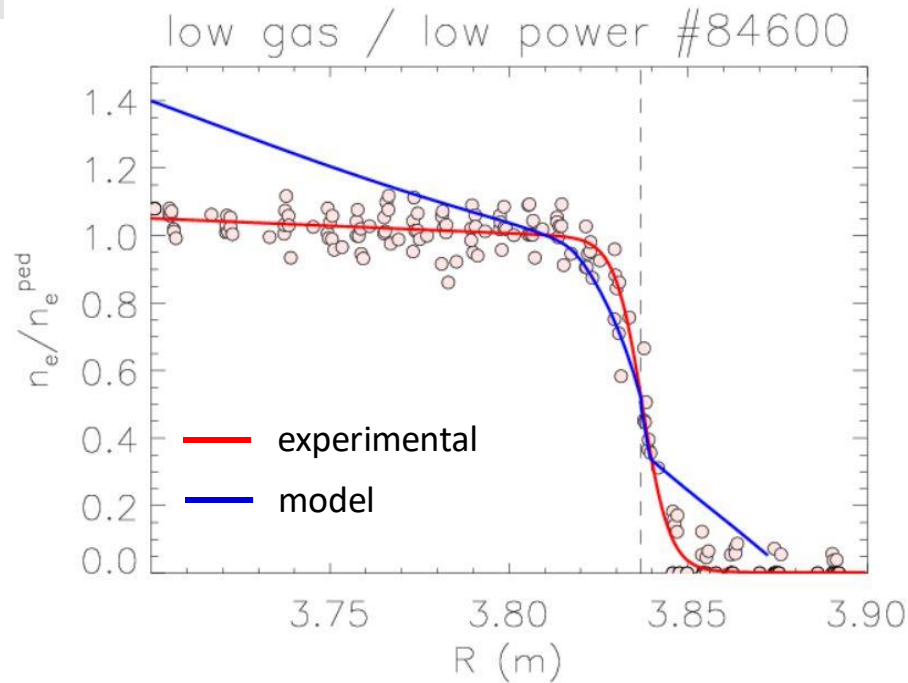
M18-20 - Dependence of pedestal structure on fuelling at constant beta



- Low gas/low power case \rightarrow low relative shift
- High gas/high power case \rightarrow high relative shift



Comparison model/experiment (normalized profiles)



- Qualitatively:
 - Reasonable agreement in the range $R=3.80\text{m} / R=3.84\text{m}$
 - Reasonable agreement in $n_e^{\text{sep}}/n_e^{\text{ped}}$
 - Density slightly overestimated in the SOL
 - Density overestimated at $R < 3.80\text{m}$ in the SOL
- Qualitatively, I would say that the comparison is rather good

2 - Software Engineering support to Modellers



- EPROC
- Python tools for modellers
- **Motivation**
 - Access to EDGE2D outputs using Python
 - Python tools for EDGE2D
- [git@git.ccf.ac.uk:bviola/EPROC_Freja.git](https://git.ccf.ac.uk/bviola/EPROC_Freja.git)
- https://users.euro-fusion.org/tfwiki/index.php/EPROC_Python_module



- Main outputs: tran-file, print-file, eirene.output, etc
 - EPROC (2007) is a user-friendly interface tool to read signals from 2D codes like EDGE2D and SOLPS
 - IDL
 - MATLAB
 - PYTHON
 - IDLBRIDGE
 - EPROC module



- Contains basic function available on EPROC: `read_ring`, `read_row`, `read_data`, `read_time`, `volint`
- Up to May 2017 available on JACS + Python/2.7.3:

```
In [1]: import eproc as ep
In [2]: ep.
ep.data          ep.getnames      ep.ring          ep.rowread      ep.timestepread
ep.dataread     ep.names         ep.ringread     ep.time         ep.volint
ep.geom         ep.pythonfilefilter ep.row          ep.timestep
```



- **Make a consolidate suite of tools for EDGE2D**
 - **Access to EDGE2D outputs using Python**
 - **Python tools for EDGE2D**
 - **Python tools for EIRENE**
- **Use effectively Python**
- **Make it easier for new comers**
- **Get same numbers!**



```
Terminal
File Edit View Search Terminal Help
PLATFORM x86_64
Welcome to FREIA
Loading python/3.5.1
Python 3.5.1
python/2.7.5(17):ERROR:150: Module 'python/2.7.5' conflicts with the currently load
python/2.7.5(17):ERROR:102: Tcl command execution failed: conflict python

setprocversion EPROC
.... finished .bashrc
python/2.7.5(17):ERROR:150: Module 'python/2.7.5' conflicts with the currently load
python/2.7.5(17):ERROR:102: Tcl command execution failed: conflict python

bviola@freia016 10:34:43 ~ \
$ bruviogui

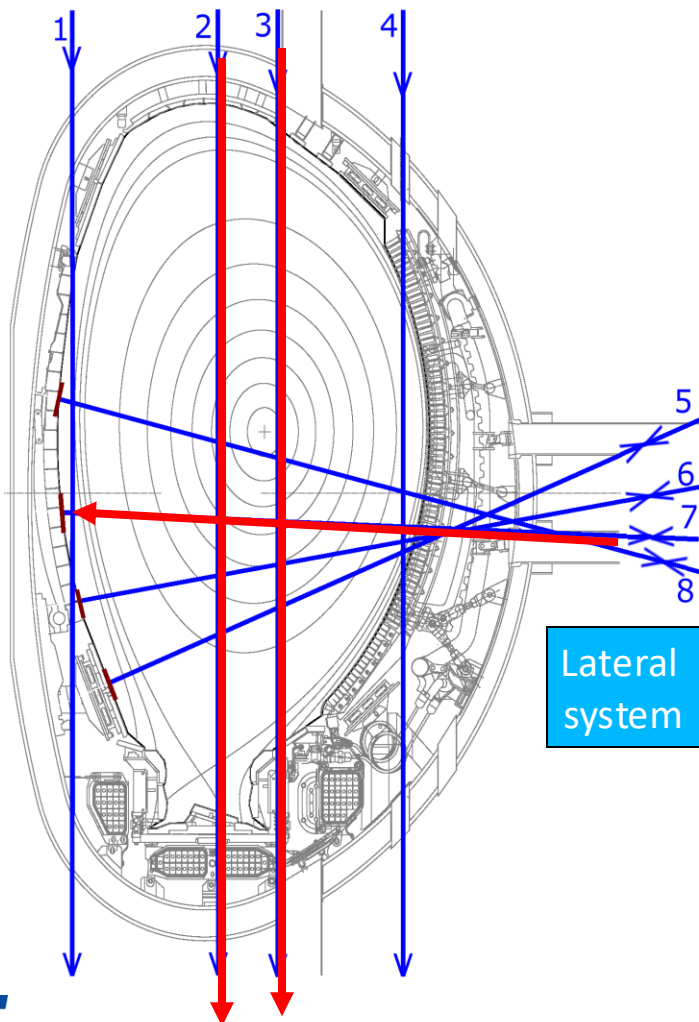
bviola@freia016 10:34:55 /u/bviola/work/Python/bruvio_tool (new_develop) \
$ python GO_bruvio_tools.py
INFO      Running bruvio tool.
DEBUG    [GO_bruvio_tools.py:70] start
DEBUG    [GO_bruvio_tools.py:79] USR in env
DEBUG    [GO_bruvio_tools.py:87] this is your username bviola
DEBUG    [GO_bruvio_tools.py:89] this is your homefold /u/bviola
DEBUG    [GO_bruvio_tools.py:90] this is edge2d fold /work/bviola/Python/EDGE2D
DEBUG    [GO_bruvio_tools.py:97] we are in /work/bviola/Python/bruvio_tool
█
```

<https://github.com/bruvio/multitool>

3 - KG1 Far Infrared Interferometer



Channels distribution



- Line Integrated Density on 8 channels
- Three data acquisition systems
 - KG1V (C40 VME) -1995(HW failing !)
 - KG1RT (PowerPC) -2002
 - KG1C (FPGA) -2012
- Two Far infrared (FIR) lasers with wavelength
 - Deuterated Cyanide(DCN) : 195 μm
 - Methanol(MET) : 118 μm
- Data availability
 - KG1V - used for current KG1V PPF production
 - KG1RT : 8 channels - only JPF
 - KG1C : 2 sets of 4 vertical channels – only JPF

Plasma Density Validation (PDV) usage *
LID2, LID3, LID7 (marked with red)

*Used for plasma density control and additional heating safety interlocks

Introduction - Rationale for change



During 2016-2017 shutdown a major refurbishment of the laser system and software infrastructure has been done that improved substantially the reliability of the system as a whole

New code was required to implement the following points:

- to incorporate the new data from two DAQ systems: KG1RT and KG1C
- the new JPF data more reliable than the original KG1V and much faster time resolution
- same or better performance in chain1 (execution time)
- implement an improved automatic fringe jumps correction
- add a time dependent flag for each LID
- move development from IDL/Fortran to Python

History

- *development was started by A.Boboc and L.Kogan in 2015 and it was on hold for more than 1 year due to lack of personnel*
- *code was tested during campaign C37 producing private PPF (DDA:KG1R, uid:chain1)*

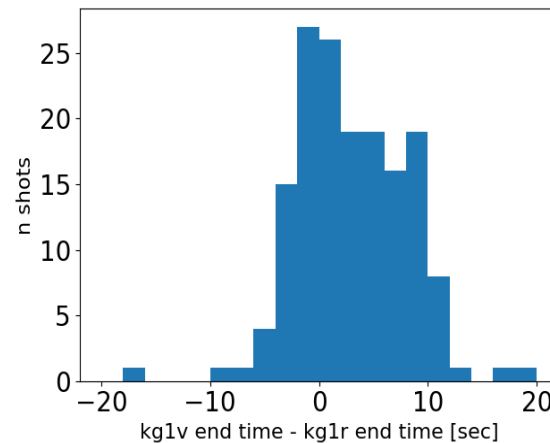
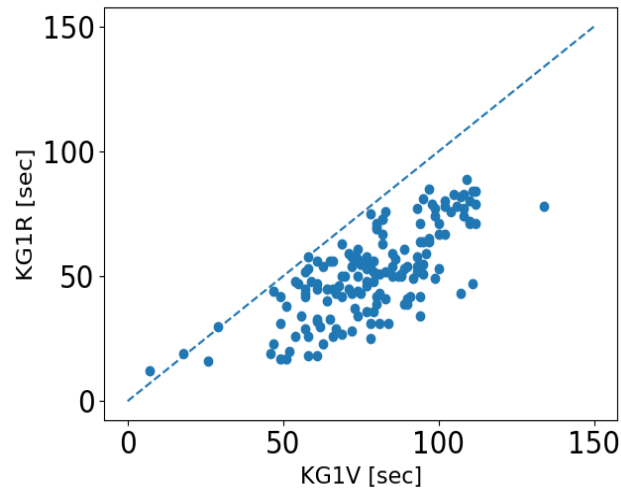
More details see the DVCM presentation “New Interferometry code for Chain 1: KG1R” L. Kogan and A. Boboc presented at DVCM on 7th December 2016

[https://users.euro-fusion.org/pages/codes-data/dvcm/meetings/2016/07-Dec-2016/DVCM_2016-12-](https://users.euro-fusion.org/pages/codes-data/dvcm/meetings/2016/07-Dec-2016/DVCM_2016-12-07.pdf)

Code Tests - C37-C37b (91619-92504)



Code performance in terms of execution time have been analysed for ~90shots using both KG1RT and KG1C.



The shots analysed include a significant number of with high plasma current and high NBI power



- L. Kogan handed over the code to B. Viola in May 2017
- Code was already tested during campaigns C37-C37b as Priority 1 producing private PPFs(DDA KG1R, uid: chain1)
- Upgraded code is now in Chain1 as priority 1
- We have reprocessed from pulse 90761

- **A few modules left to be implemented**
 - Manual validation
 - Dependent chain1 codes, in particular not maintained legacy codes
 - Scientific data compatibility



Completed (to be tested during campaign)

- Polarimetry data (KG4) for fringe corrections
- Time dependent status flag

To be developed

- Automatic validation using HRTS *
- Others: ELMs detection (to be improved), pellet counting, disruption detection

** This code to run as a reprocessing step in chain2.*



Graphical User Interface (GUI) that contains (almost) all the tools needed by KG1 RO.

It allows:

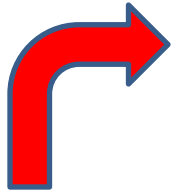
- Run manual validation code (i.e. CORMAT on JACs)
- Run PPF production codes (KG1L/H, KG1R_py, **KG1V (to be added?)**)
- Generate input file for KG1R_py code
- Plot data
- Check database of validated pulses (and more)
- Check JPN status flags
- Contact chain1 manager



- A. Boboc is on leave
- KG1V lateral channels **not working**
- Experimental campaign started



Data Production (previous) workflow



Kg1 data

kg1v_idl code



jetppf/KG1V/LIDx ppf

kg1_py code



chain1/KG1R/LIDx ppf



This is something you already know!

What you also should know is that for the past two years another code is running in Chain1 as Priority 1

Ignite!

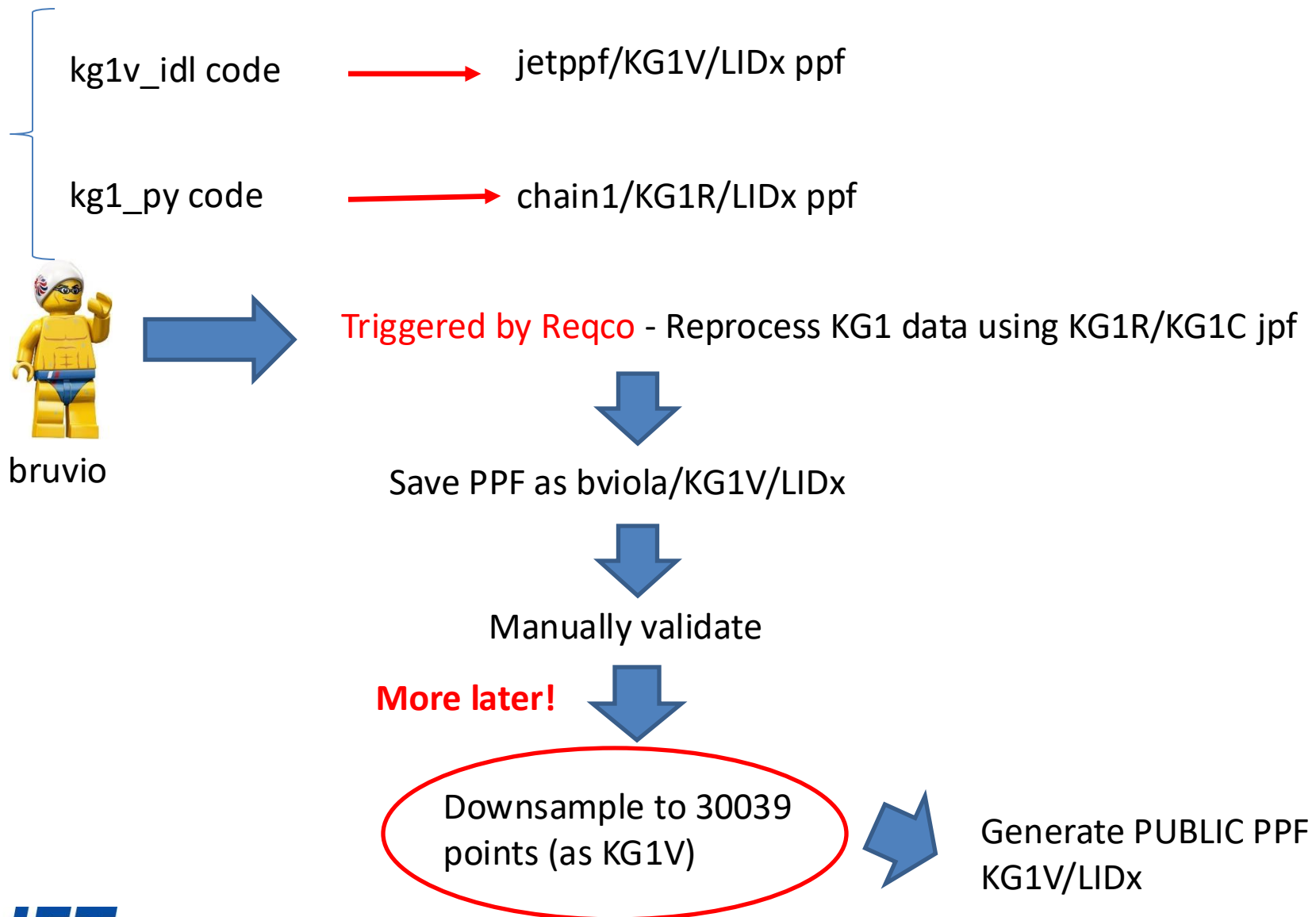


EIC

PPF/KG1R userid: chain1 , time res: 1ms across entire pulse !

This PPF contains the KG1 data coming from KG1R/KG1C only

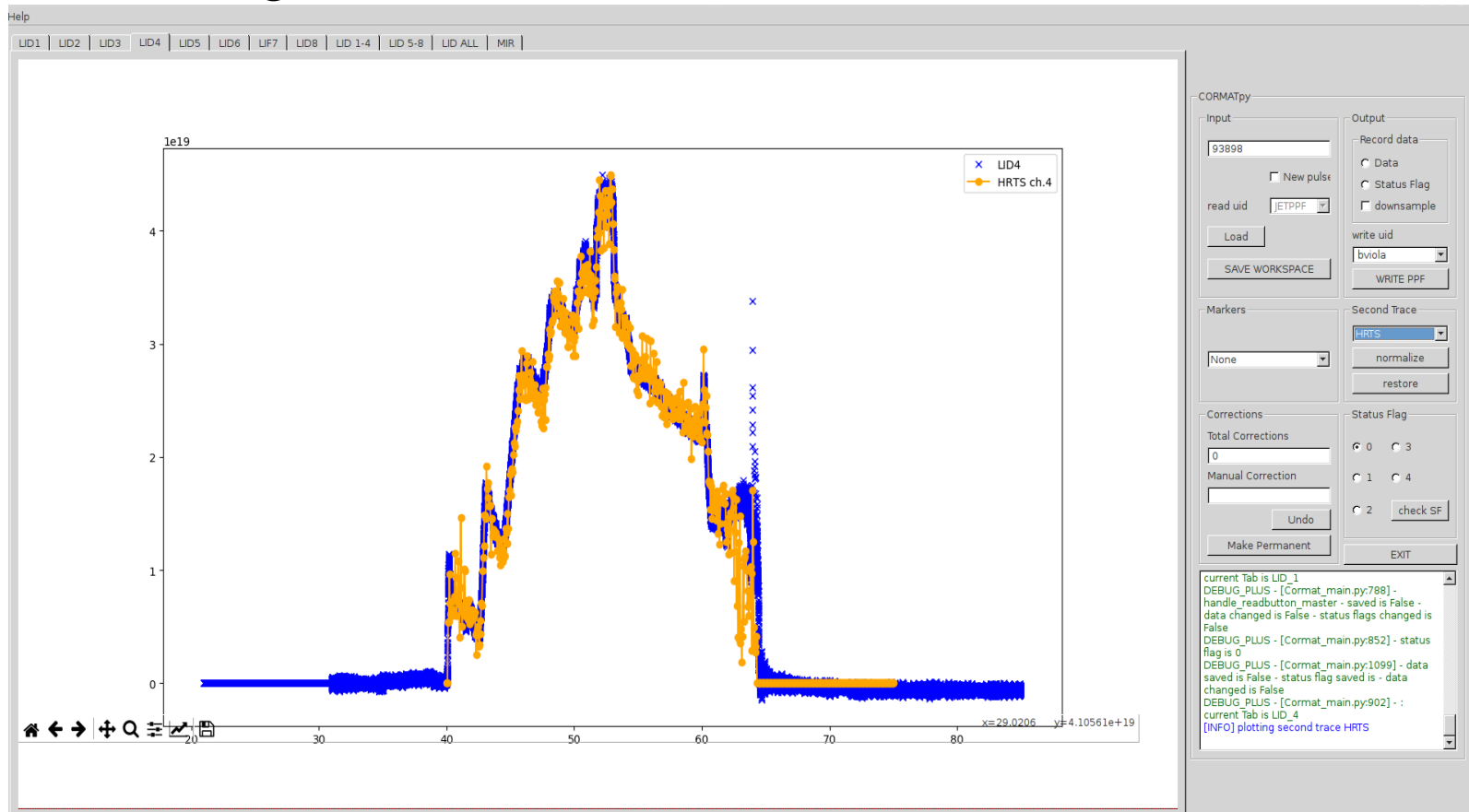
Data Production (new) workflow



KG1 codes Update



- KG1_py code (use to produce PPFs under dda=KG1V) is working



KG1 (re)processing options



[kg1_signals] (the ppf) data is now produced using KG1R and data **Option running on Chain1**

1: best
2: best
3: best
4: best
5: kglr
6: kglr
7: kglr
8: kglr

best: The code will use KG1C for vertical channels (if available), taking LD<DCN as the main signal, and LD<MET if not. Both DCN & MET signals & LDCOR signals will be used to help make corrections. If KG1C signals are unavailable then KG1R will be used. For the lateral channels KG1R is always used.

kglr : Use KG1R signals for all channels.

kg1c_ldraw : Use combined KG1C ldraw DCN & MET for vertical channels.

kg1c_ld : Use combined KG1C ld DCN & MET for vertical channels.

kg1c_ldcor : Use combined KG1C ldcor DCN & MET for vertical channels.

kg1c_ldraw_dcn : Use KG1C ldraw dcn for vertical channels.

kg1c_ld_dcn : Use KG1C ld dcn for vertical channels.

kg1c_ldcor_dcn : Use KG1C ld dcn for vertical channels.

kg1c_ldraw_met : Use KG1C ldraw met for vertical channels.

kg1c_ld_met : Use KG1C ld met for vertical channels.

kg1c_ldcor_met : Use KG1C ld met for vertical channels.

Configuration file generator - LID chooser



KG1 chain 1 file generator

KG1 chain 1 generator v.2.0

LID1 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> LD DCN <input type="radio"/> LDCOR DCN <input type="radio"/> LD MET <input type="radio"/> LDCOR MET <input type="radio"/> LDRAW DCN	LID2 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> LD DCN <input type="radio"/> LDCOR DCN <input type="radio"/> LD MET <input type="radio"/> LDCOR MET <input type="radio"/> LDRAW DCN	LID3 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> LD DCN <input type="radio"/> LDCOR DCN <input type="radio"/> LD MET <input type="radio"/> LDCOR MET <input type="radio"/> LDRAW DCN	LID4 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> LD DCN <input type="radio"/> LDCOR DCN <input type="radio"/> LD MET <input type="radio"/> LDCOR MET <input type="radio"/> LDRAW DCN
LID5 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> KG1C LD <input type="radio"/> KG1C LDCOR <input type="radio"/> KG1C LDRAW	LID6 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> KG1C LD <input type="radio"/> KG1C LDCOR <input type="radio"/> KG1C LDRAW	LID7 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> KG1C LD <input type="radio"/> KG1C LDCOR <input type="radio"/> KG1C LDRAW	LID8 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> KG1C LD <input type="radio"/> KG1C LDCOR <input type="radio"/> KG1C LDRAW

Pulse

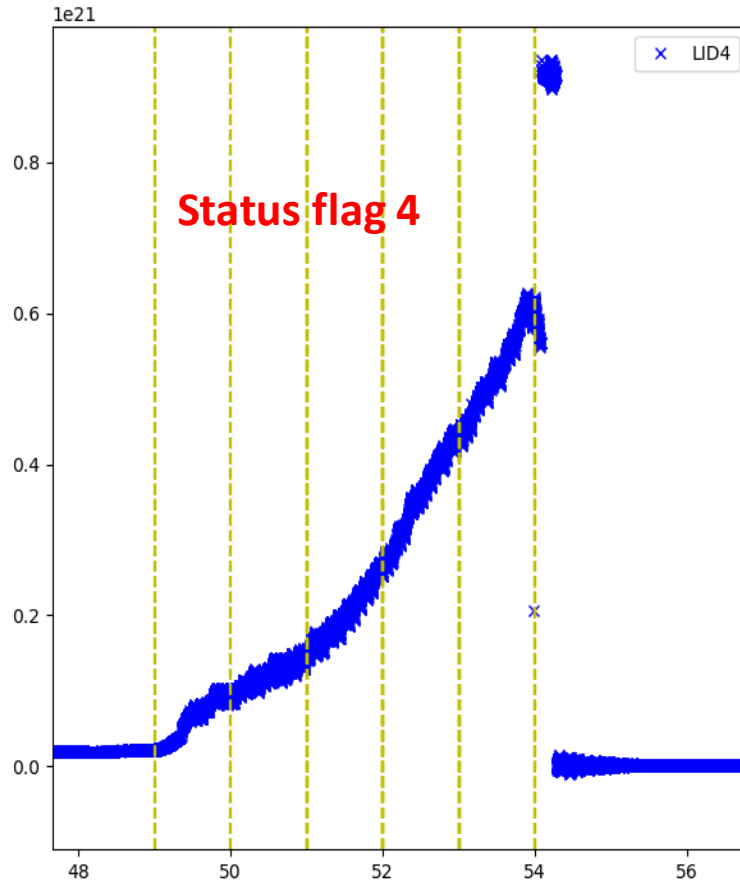
KG1V IDL KG1R_py

- It allows to modify configuration input file (kg1_const.ini) used by both KG1V_IDL and KG1R_py code

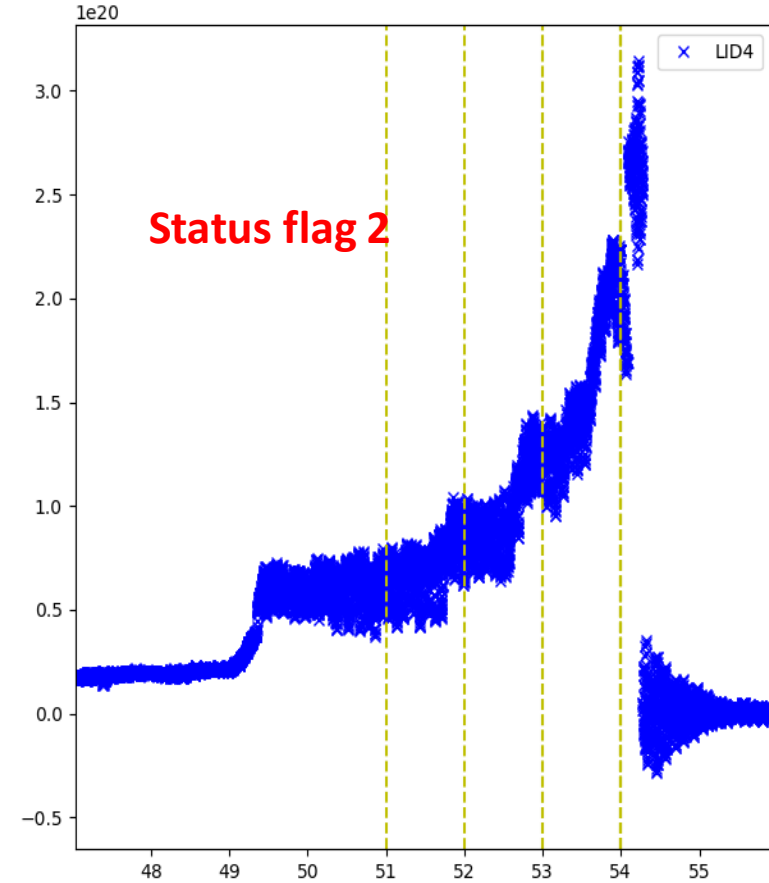
Example: JPN 94896/LID4



“best” option



“second colour only” option



Cause

Main DCN laser beams are refracted by plasma
Second colour methanol laser is 10 time smaller refraction

p.s. all M18-16 are like this!



- Bottleneck – code needs changes as KG1V (old system) parameters are hardcoded
- Not using the “full power” of KG1R and KG1C systems: need to reduce number of samples to run KG1L/H, i.e. down sample) **New KG1L/H code since 1st of November 2019**
- Old Fortran code can run only if data comes from KG1V or KG1R systems, does not have implemented to use data from KG1C

(old Fortran code = Pandora box !)

New KG1L/H code – main points



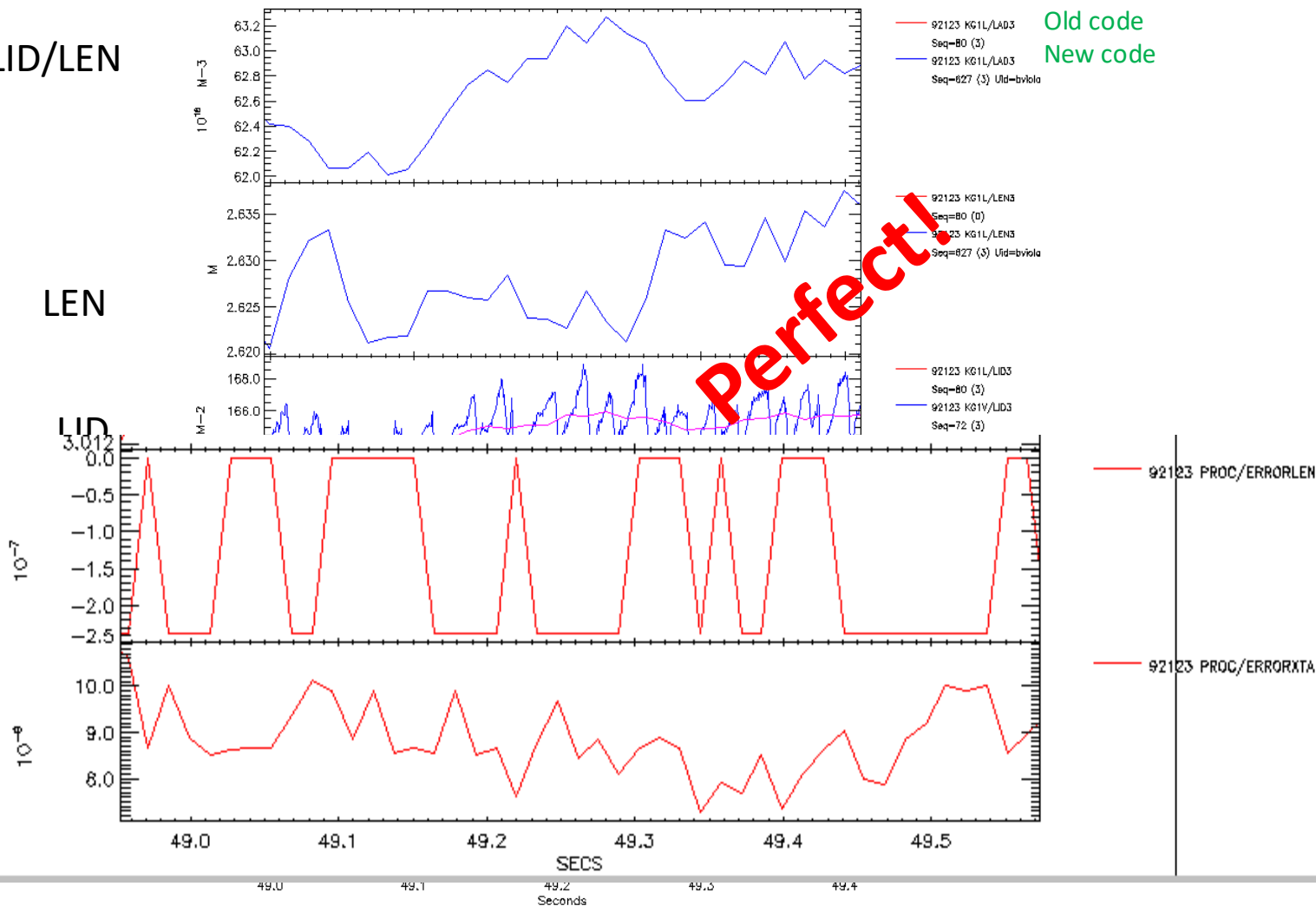
- Uses a FLUSH wrapper that works in Python3!
- Can use various filtering algorithm for slower time res of EFIT
- Multithreading
- OOP structure
- Fully documented
- Advanced debugging
- Fully integrated with ReqCo database
- Requires KG1V validation (as before)

Thanks to Derek S. for precious advice!

KG1L_py output: Fortran algorithm



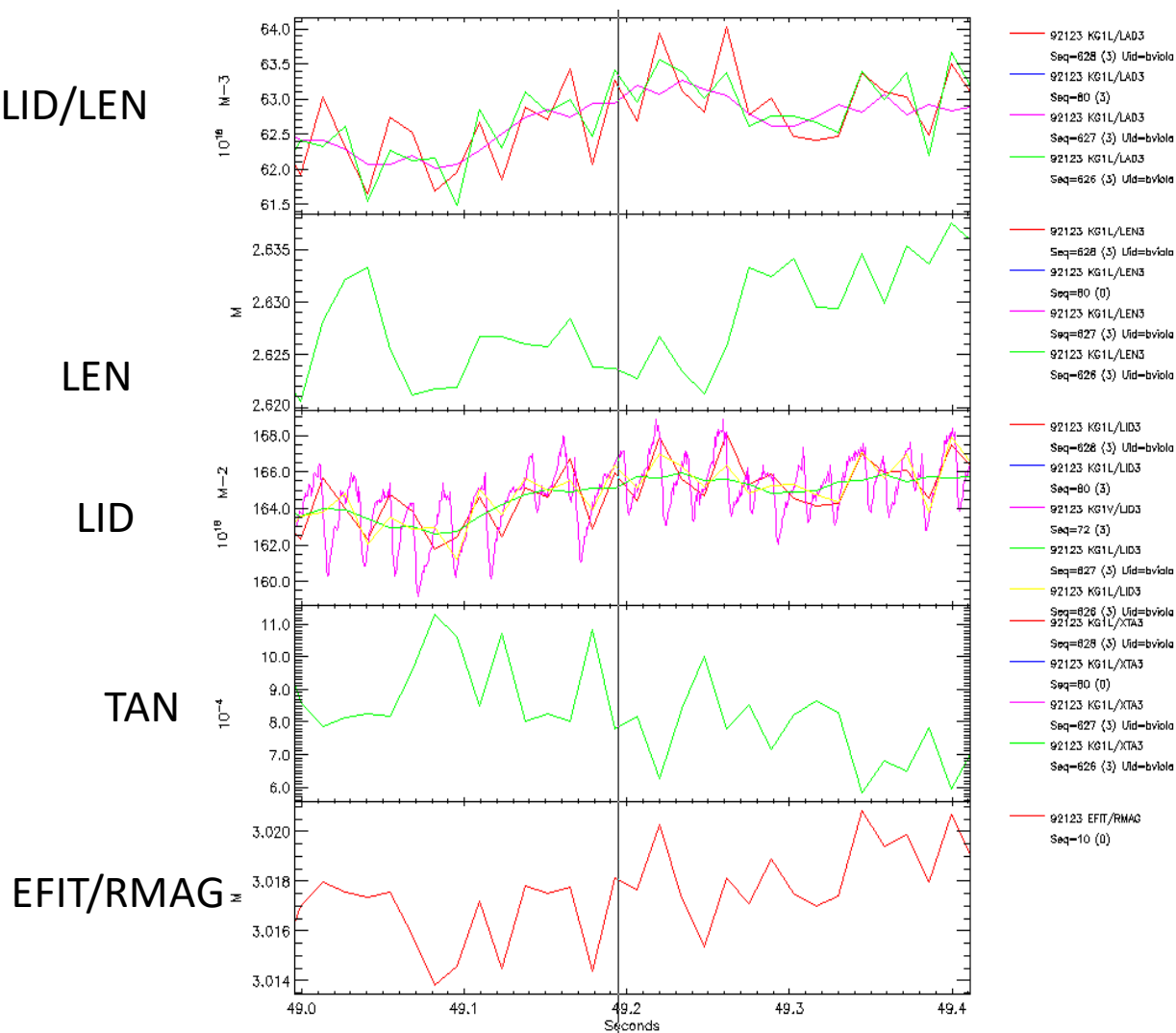
$$\text{LAD} = \text{LID}/\text{LEN}$$



KG1L_py output: NEW FILETERING ALGORITHMS



LAD = LID/LEN

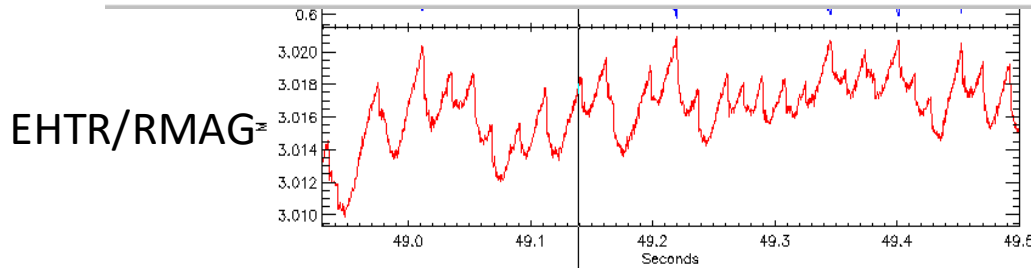
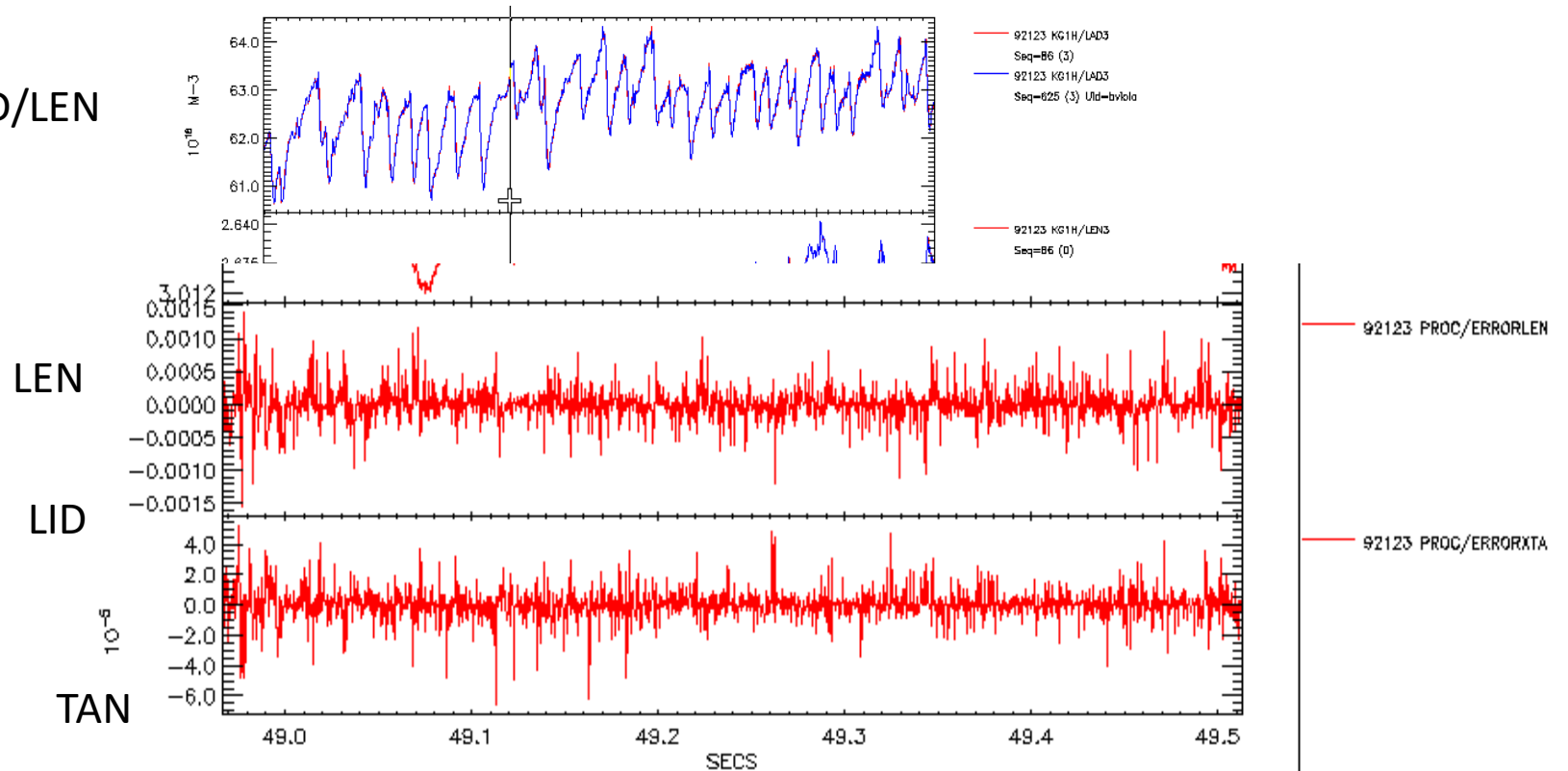


Using new filtering options speeds up code by ~20s! (40% speed increment)

KG1H_py output



LAD = LID/LEN



Abs Err = max 3mm
Len = 2.7m
Rel Error 0.1%

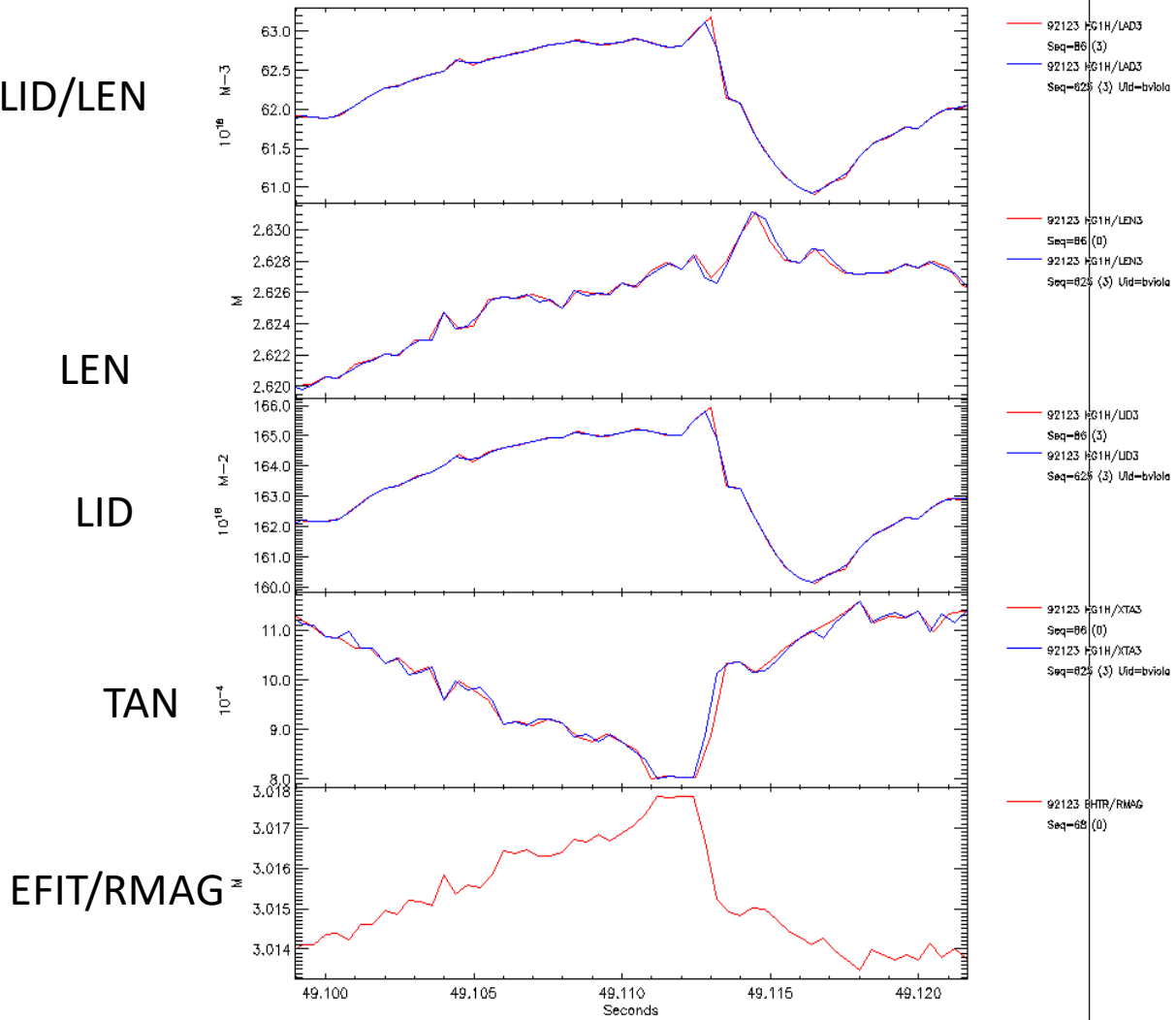
0.5s



KG1H_py output - zoom



LAD = LID/LEN



Old Fortran code filters the data (twice!), so there are less points, this is why they look different.

New code is more accurate, for those who use KG1H data for studying fast dynamics

0.01s

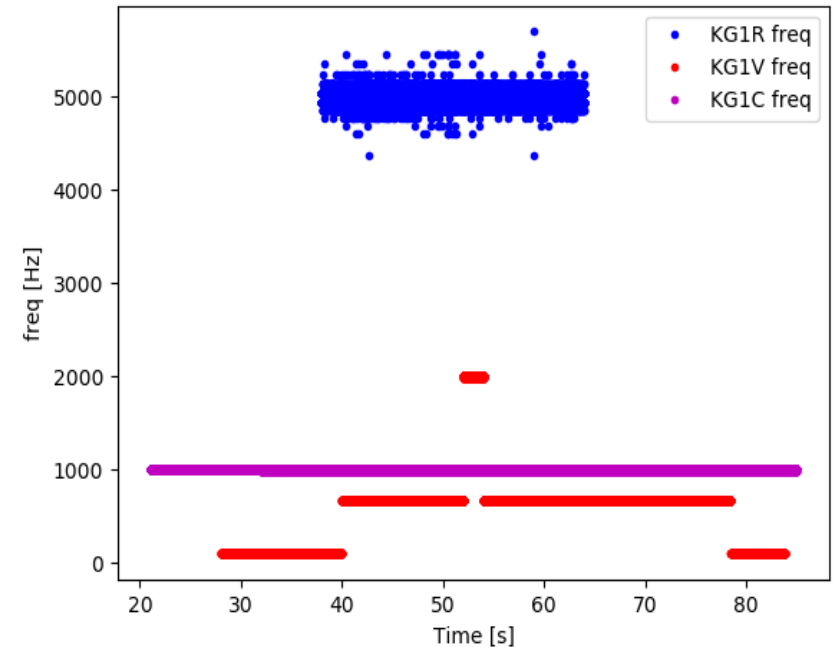


Down sampling of KG1 data

Reasons for down sampling



- Previous KG1 code (old Cormat running in MATLAB, KG1L/H Fortran, and others???) have KG1V data points (30039) hardcoded
- We still don't know what is the impact on the chain of using a different number of data points
- To keep it simple we decided with Chain1 RO to maintain this until we finalize tests
- Tests on their way to remove this constraint



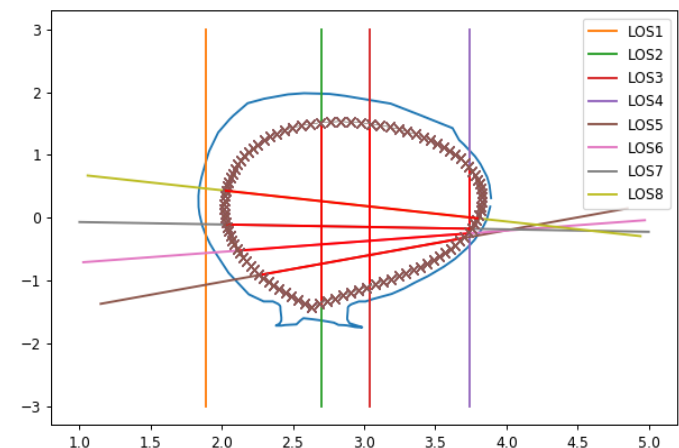
KG1V – hard coded 30039 points- 5 time windows with three time resolutions
KG1C- fixed time window 1ms time resolution
KG1R – variable time window, fixed time res range 0.2ms

Repetita iuvant ---> LENx - discussion



- LENx ppf are produced by KG1L/H codes
- After KG1V is validated, i.e. after Reqco requests
- KG1L/H uses flush (code runtime using standard EFIT ~70s)
- KG1LH, it appears, is under my only responsibility (i.e. Alex does not want it!)
- People want LENx (possibly LADx) intershot – pronto, schnell, rapide, subito!
- Need of a new way to do this calculation

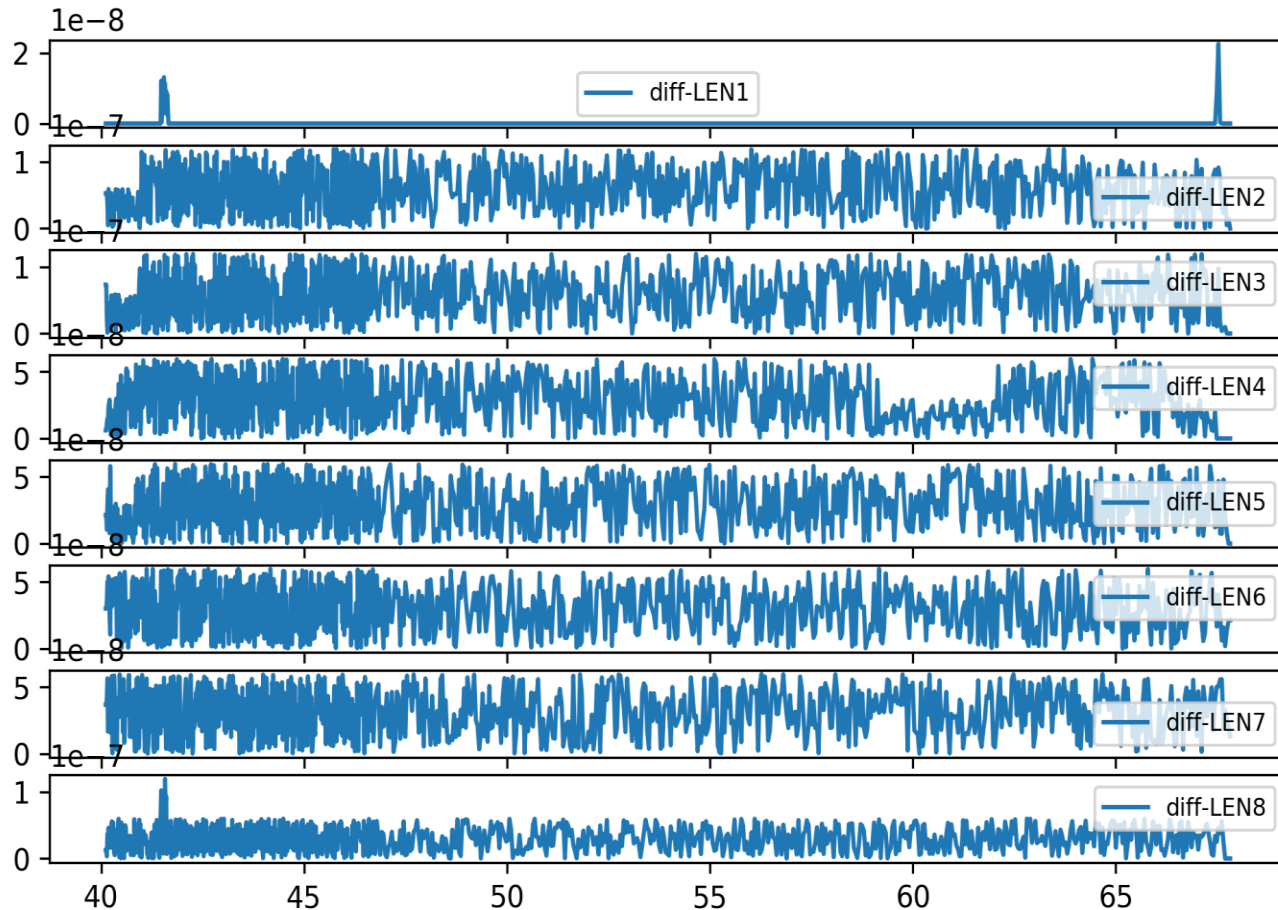
Suggested by E. Solano = use
(RBND,ZBND) from EFIT





1. Compute LENx in chain1 inside KG1V_py code, using EFIT as source
 2. Compute LENx in chain1 inside EFIT, using EFIT as source
 3. Compute LENx in chain1 inside KG1V_py code using XLOC as source
-
1. Compute LENx using KG1LH codes (implementing new algorithm) – i.e. leaving things as they are

1 - Compute LEN in chain1 inside KG1V_py code, using EFIT as source



jetppf',seq=0)

37]

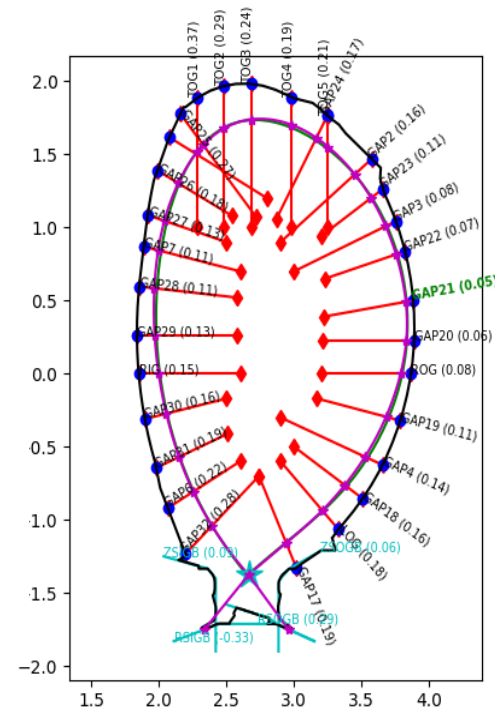
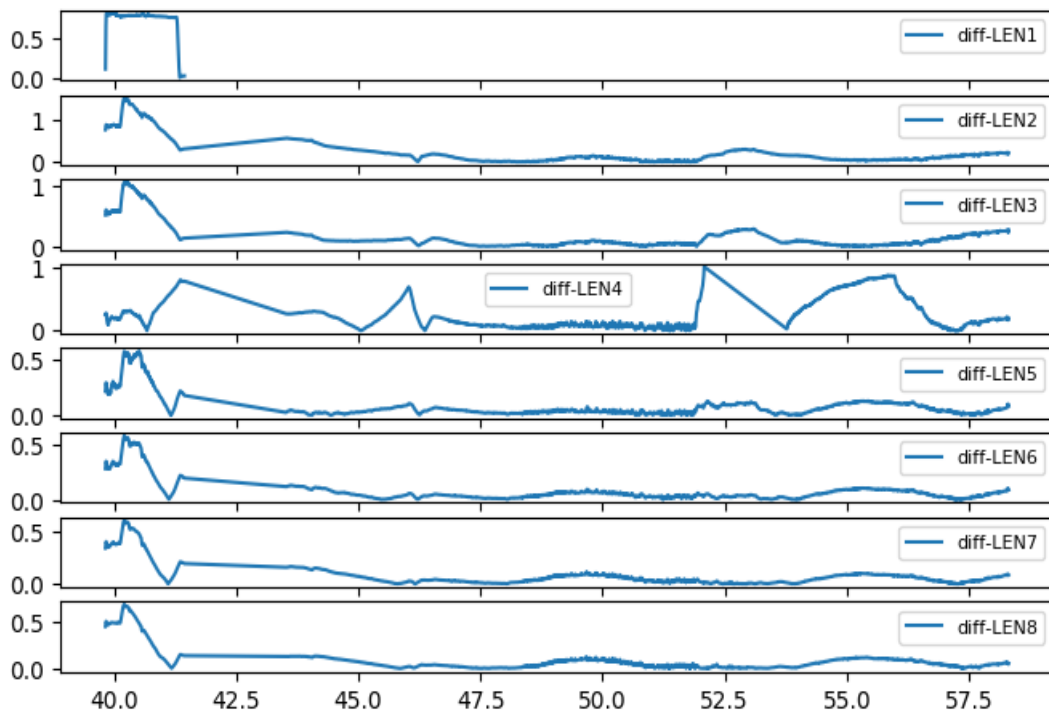
62]

2 - Compute LEN in chain1 inside KG1V_py code using XLOC as source



- Calculation slower (XLOC sampling rate is higher 0.01s)
- XLOC data is JPF so in theory available at the moment

O any



3- Compute LEN in chain1 inside EFIT, using EFIT as source



- Pro:
 - EFIT always available
 - Data would be always ready intershot
 - LEN calculation available for all members of EFIT family (EFTP, EHTR...)
 - Data provenance: each EFIT sequence produces and stores its associated lengths. **Especially important for EFTP and kinetic reconstruction.**
 - LENx available immediately for any EFIT, including kinetic reconstructions. Otherwise, after each EFIT run, producing a new sequence, a new validation of KG1L would need to be requested, specifying the EFIT sequence to be used as input. This is awkward and error-prone, and the most recent KG1L sequence may not always be the best one.

Con:

- EFIT RO needs to modify code – **(check with EFIT RO timeline for implementation)**
- KG1 line of sight data needs to be read by EFIT (this is already necessary if polarimetry is used for a kinetic reconstruction). EFIT RO needs to be made aware of recalibrations/changes.
- Estimated 10s delay in production of JETPPF/EFIT **(worst case scenario)**



KG1 GUI Tools for data generation, display and validation



Graphical User Interface (GUI) that contains (almost) all the tools needed by KG1 RO.

It allows:

- Run manual validation code (i.e. CORMAT on JACs)
- Run PPF production codes (KG1L/H, KG1R_py, **KG1V (to be added?)**)
- Generate input file for KG1R_py code
- Plot data
- Check database of validated pulses (and more)
- Check JPN status flags
- Contact chain1 manager



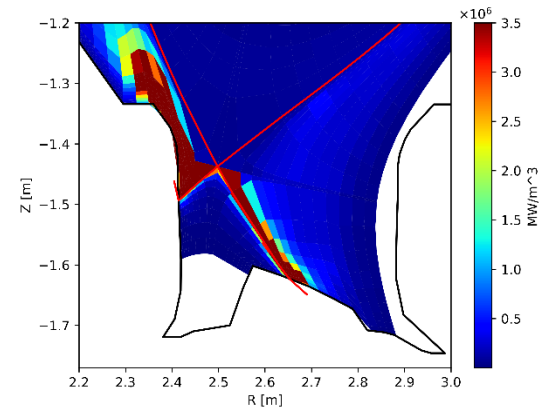
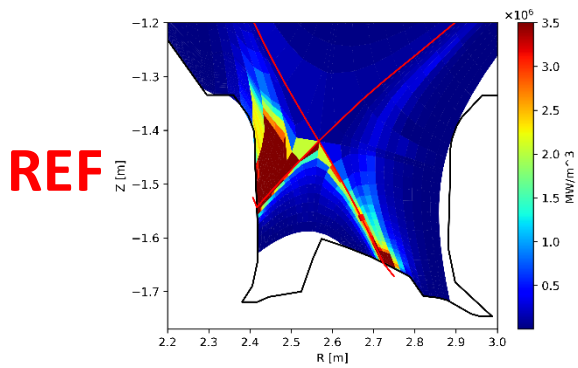
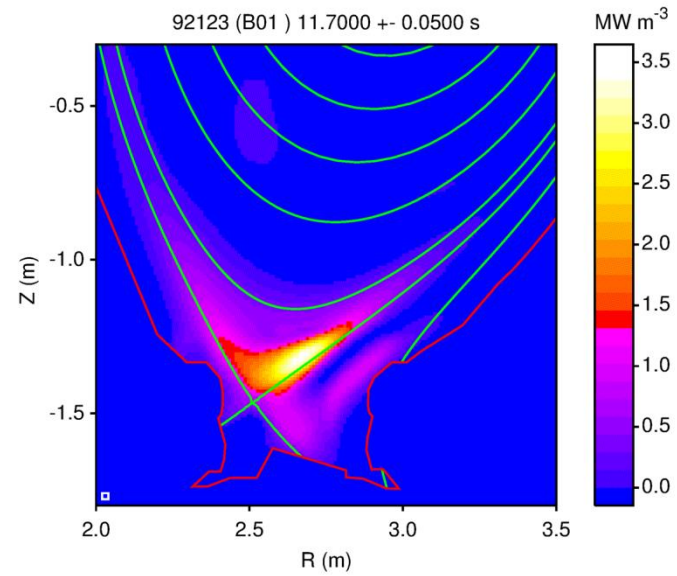
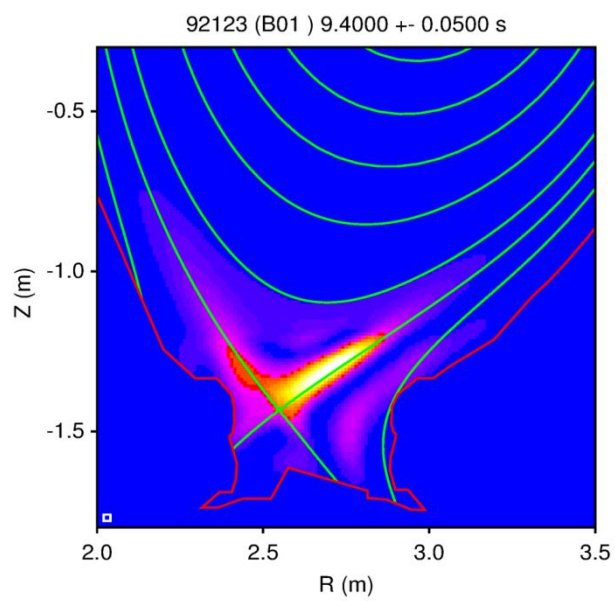
- Includes the ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures
- Design Principles
 - Perform operations as code - Infrastructure as code
 - Annotate documentation - Automate the creation of annotated documentation after every build
 - Make frequent, small, reversible changes - So that in case of any failure, you can reverse it
 - Refine operations procedures frequently - And ensure that team members are familiar with it
 - Anticipate failure
 - Learn from all operational failures



- KG1V data RO
- KG1 main software engineer
- Modelling activities in support of JET
- Software development activities in support of modellers
- EPROC_py library RO
- DVCM secretary
- DCO during JET experimental campaign



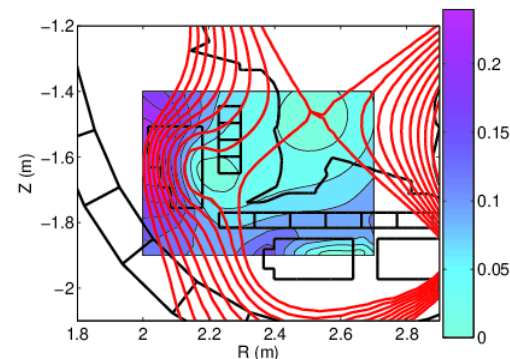
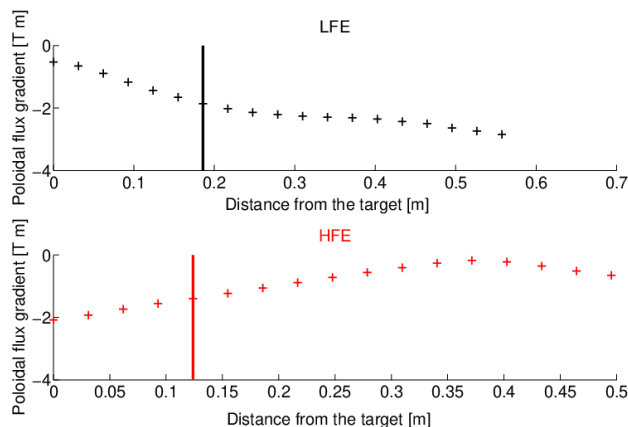
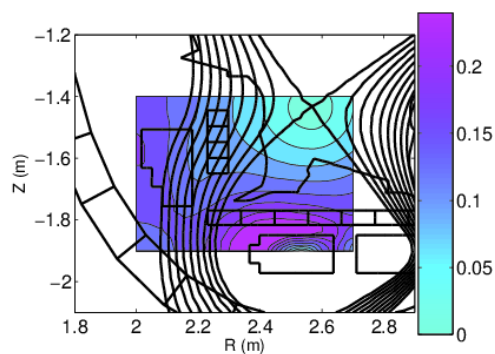
Result overview 3/3



Conclusions



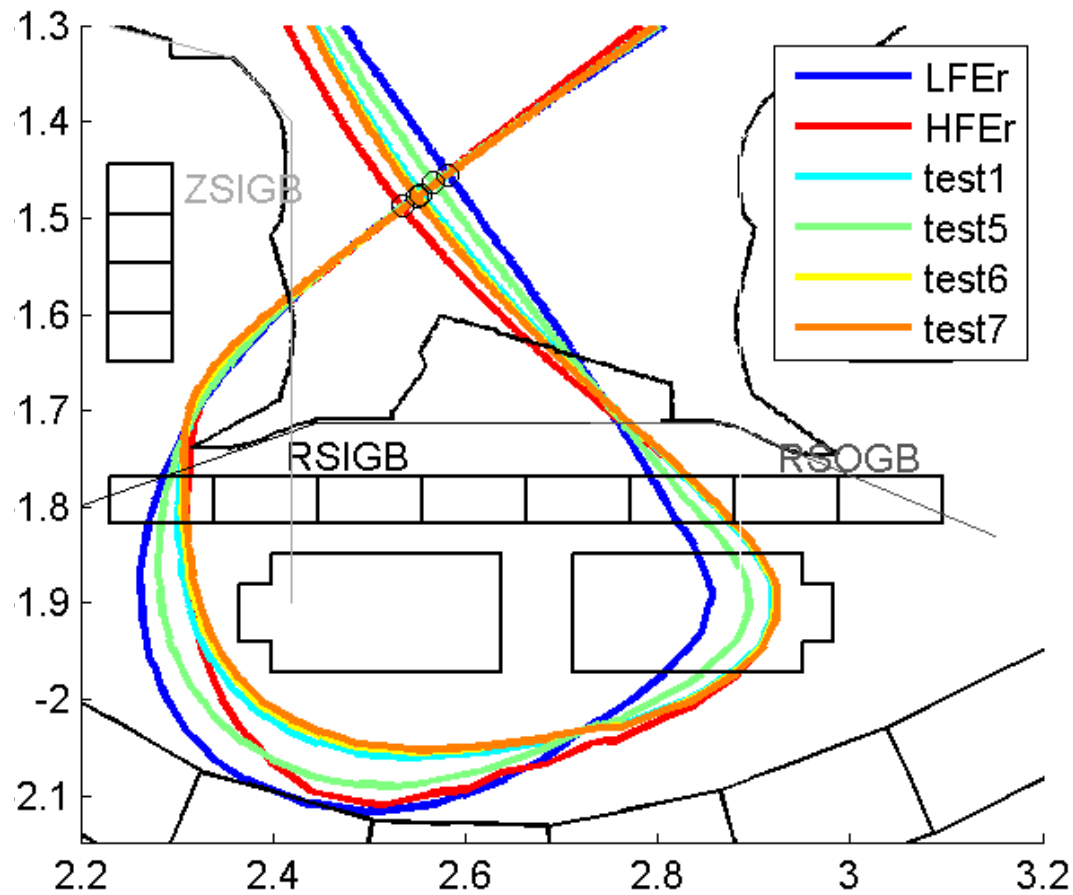
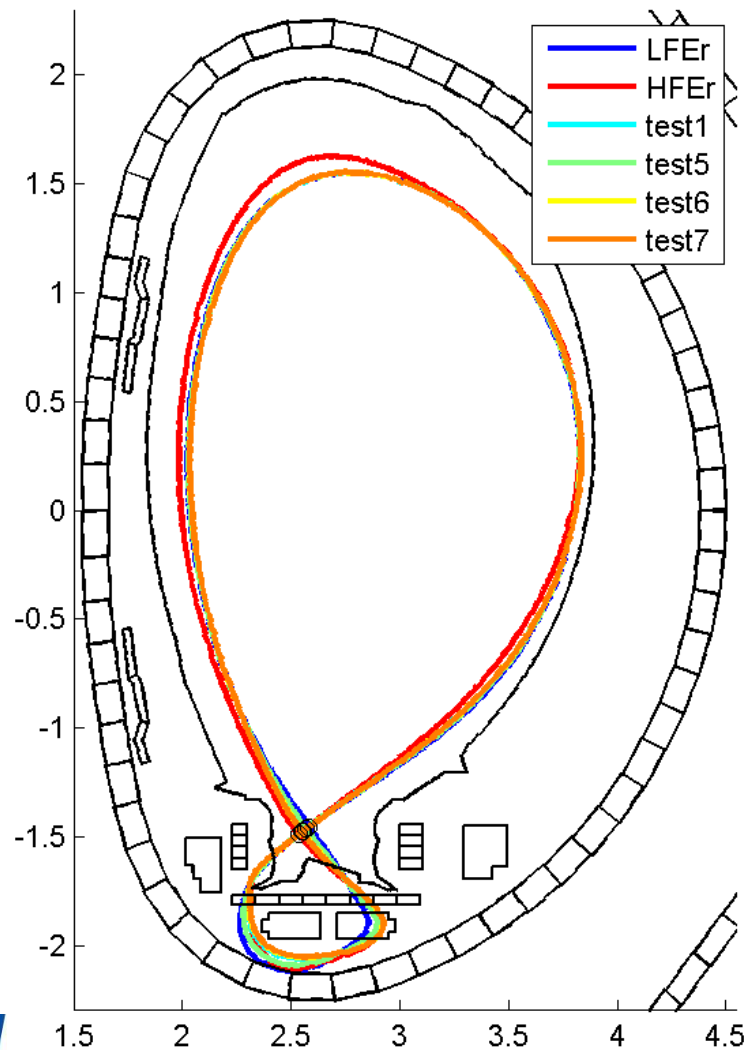
- Experiment have shown that Flux expansion has effect on divertor radiation and that is possible to obtain with current divertor coils setup a flaring flux at JET (see picture below)
- Flux expansion have effect on the movement of the radiation (nitrogen) front toward the X-point
- EDGE2D modelling confirms experimental results
- *Improve constraint on X-point position (future?)*



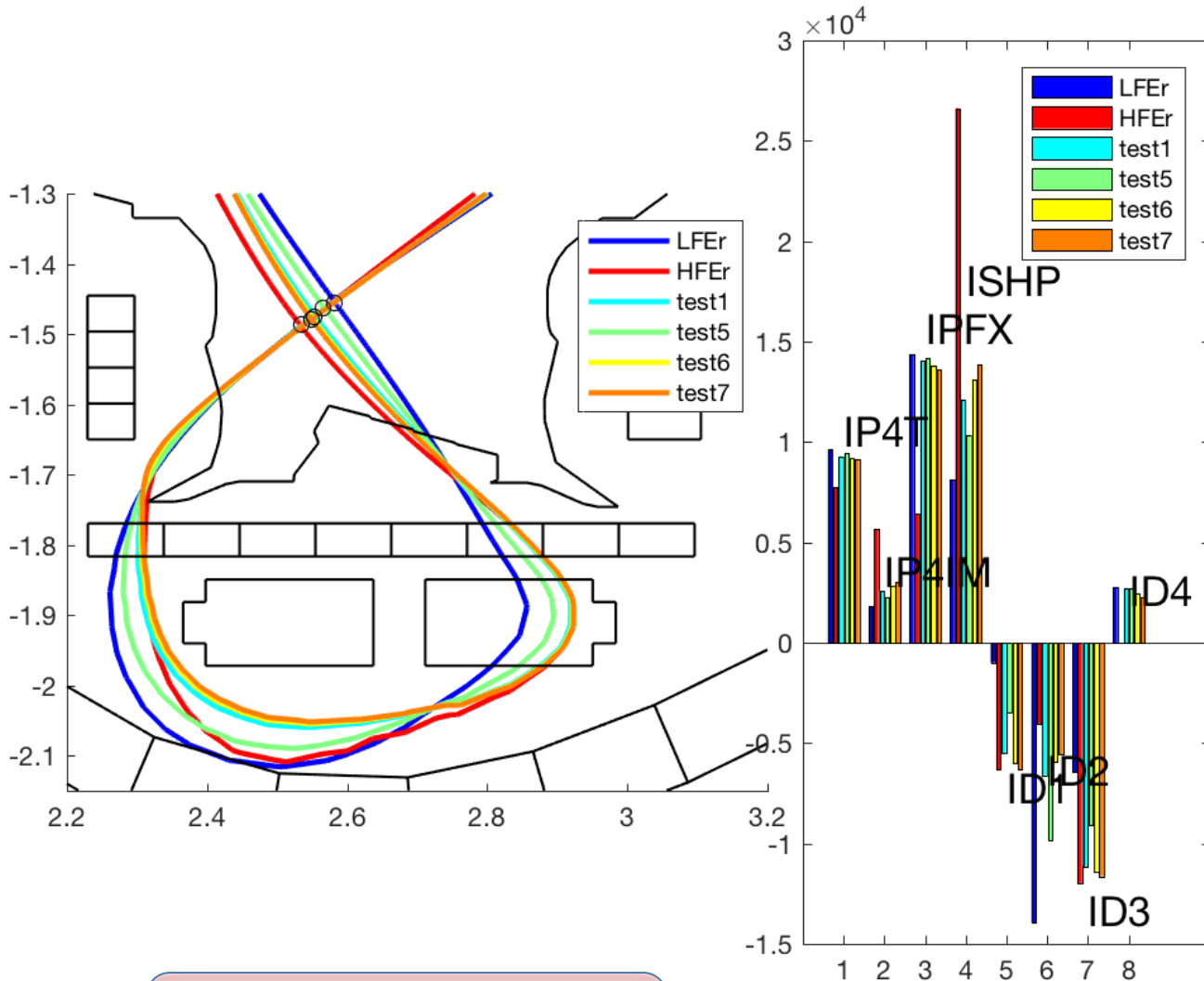
Shapes comparison 1/2



Here we compare all the shapes: with the new minimisation we keep frozen TOG, RIG ROG
As shown in pag 5 the trade-off is a 10% reduction in max FE achievable (test 7 vs 92121@51.7s)



Shapes comparison 2/2



Name	Min	Max
IPRIM	-4.000e+001	4.000e+001
IP4T	0.000e+000	2.860e+001
IP4IM	-1.400e+001	1.400e+001
IPFX	0.000e+000	3.290e+001
ISHP	0.000e+000	3.800e+001
ID1	-1.900e+001	0.000e+000
ID2	-3.700e+001	0.000e+000
ID3	-3.700e+001	0.000e+000
ID4	0.000e+000	1.800e+001

ΔI_i is now lower

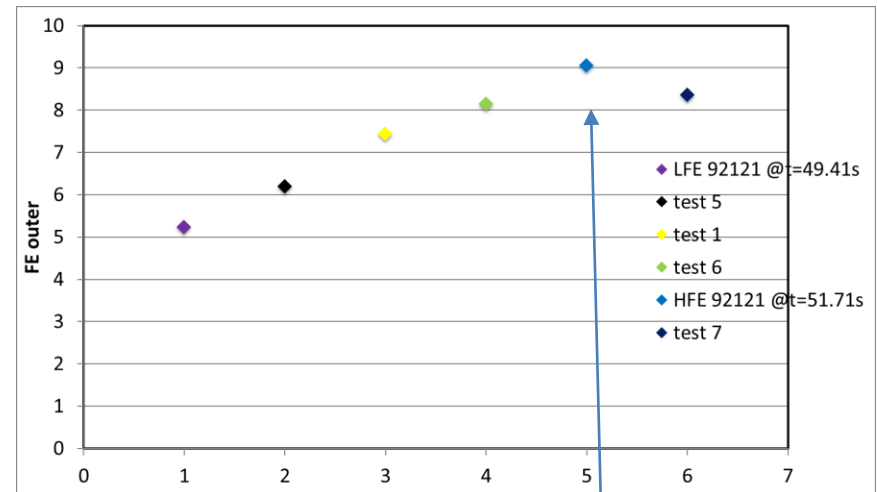
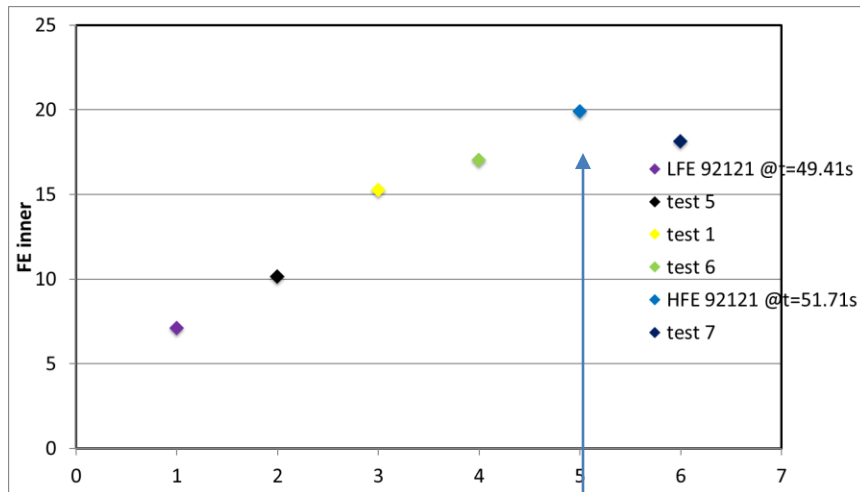
I_{D4} not 0!

All currents within limits!

Experimental Strategy (quick summary 1/2)



- Perform/repeat previous pulse 92121 for better spectroscopy measurements
- Develop flux expansion discharges

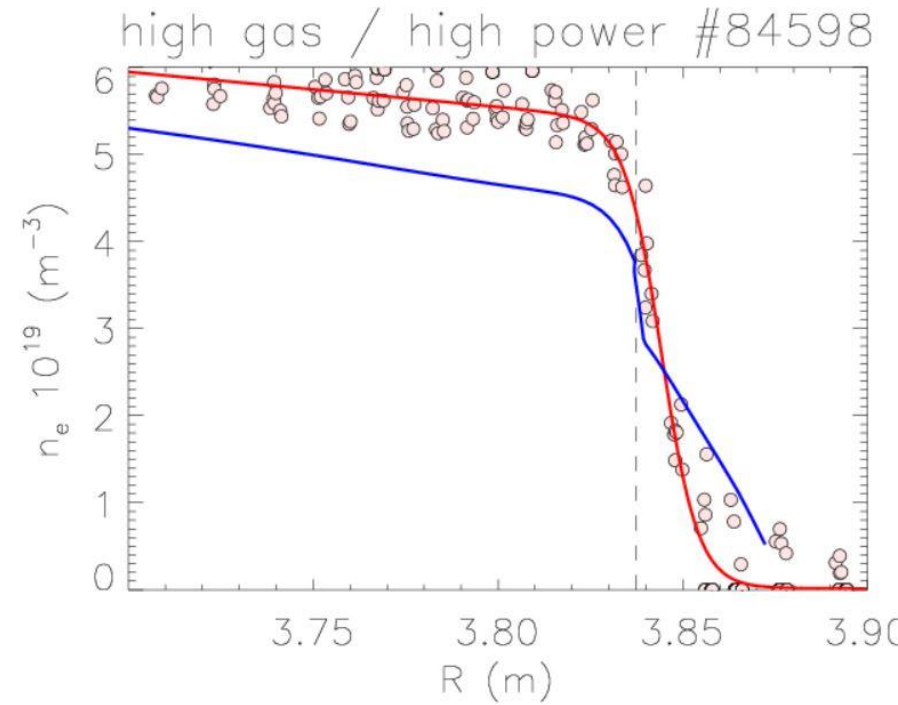
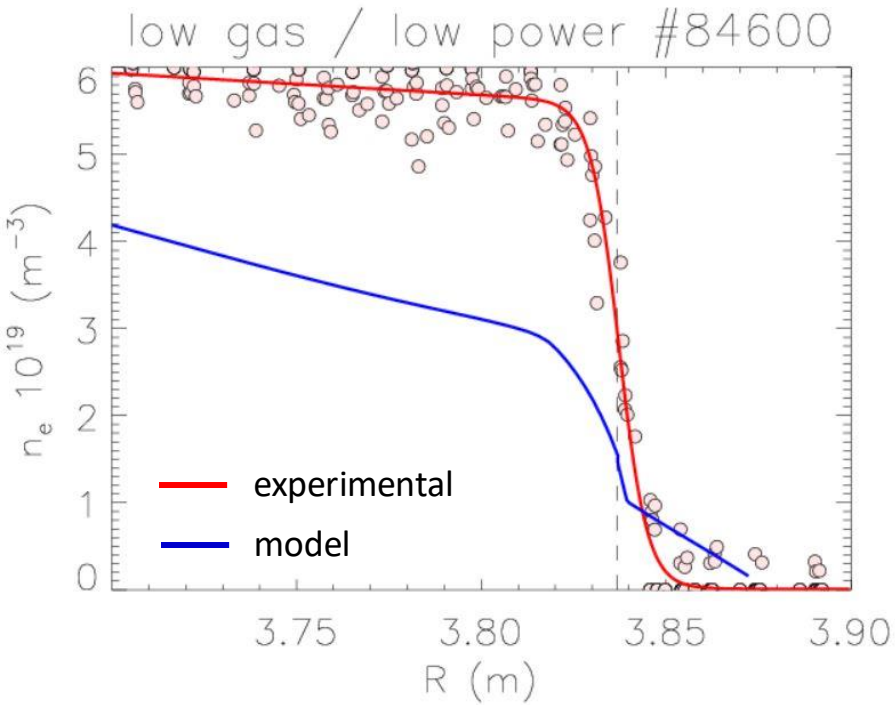


This data point is not requested during the scan in the Experiment (see following)



- Allow for repeat pulses to try different Flux Expansion configuration with power ($\sim 15\text{MW}$ or more if available)
- N scan (3 shots)
- Experimental proposal draft handed over to TFLs asks for $\sim 18\text{shots}$

Comparison model/experiment



- Quantitatively:
 - n_e^{ped} underestimated in the low gas case.
 - n_e^{ped} slightly underestimated in the high gas case.

What is EPROC?



- Eproc is library of IDL procedures.
- A reference manual for the Eproc library is available by typing the Unix command *eprdoc*.
- Manual of EPROC GUI can be found here:
https://users.euro-fusion.org/pages/data-cmg/eprocgui/manual/eproc_gui_manual.pdf

List of Routines

- EPROCADD
- EPROCBOUND_DEFINE
- EPROCCAT
- EPROC DATA
- EPROC DATARANGE
- EPROC DATASPEC
- EPROC DIFF
- EPROC DIVIDE
- EPROC ERRORSIG
- EPROC FILE_DEFINE
- EPROC FLT DATA STRUCT_DEFINE
- EPROC FLUX
- EPROC GET CODEIDS
- EPROC GET DATEIDS
- EPROC GET MACHIDS
- EPROC GET NAMES
- EPROC GET OWNERS
- EPROC GET SEQNUMS
- EPROC GET SHOTIDS
- EPROC GET SQLVALUES
- EPROC GRID
- EPROC GRID2DDATA
- EPROC GRID2DDATA STRUCT_DEFINE
- EPROC GRID2DPLOT
- EPROC GRID2DUI
- EPROC GRIDUI
- EPROC INTEGRATE
- EPROC LON DATA STRUCT_DEFINE
- EPROC MCDATA
- EPROC MCDATA STRUCT_DEFINE
- EPROC MULTIPLY
- EPROC NAMES
- EPROC PLOT
- EPROC POINT
- EPROC POLY
- EPROC POLYUI
- EPROC PS
- EPROC RANGE
- EPROC RING
- EPROC ROW
- EPROC STRUCT_DEFINE
- EPROC SUBTRACT
- EPROC SURF
- EPROC SURFINT
- EPROC SURFREC



- git@git.ccfе.ac.uk:bviola/EPROC_Freja.git
- https://users.euro-fusion.org/tfwiki/index.php/EPROC_Python_module

EDGE2D GUI



The screenshot displays the EDGE2D GUI interface. The main window, titled "edge2d tools", contains several sections:

- Analyze:** Includes input fields for "input_dict_84600.json" and "compare_dict_84600.json", with checkboxes for "edit JSON1", "compare pulse", and "edit JSON2". A "run" button is present.
- compare sims:** Features an "input dictionary" field containing "input_dict_84600.json" and a "var name" field containing "denel". A "run" button is at the bottom.
- search sim:** A search interface with a dropdown menu, an "add" button, and a "test" input field.
- Get sim names:** A panel with checkboxes for "Profile", "Time", "Flux", and "Geom", and a "run" button.
- targetfilename:** A field containing "denel".
- Navigation buttons:** A vertical stack of buttons including "profiles", "pump currents", "download print", "power balance", "contour plot", and "radiation contour".
- exit:** A button at the bottom center of the main window.

In the background, a file manager window shows a list of files:

Filename	Size	Type
MMfast_snowflake_23may2011_2attempt.eqdsk	179.2 kB	plain text document
MMfast_snowflake_23may2011_3attempt.eqdsk	676.5 kB	plain text document
MMfast_snowflake_23may2011_4attempt.eqdsk	676.5 kB	plain text document
MMfast_snowflake_23may2011_5attempt.eqdsk	2.6 MB	plain text document
MMfast_snowflake_23may2011_6attempt.eqdsk	1.5 MB	plain text document

The bottom of the screen shows a terminal window with the text "Terminal" and "Search for 'eqdsk'", and a taskbar with icons for "bruvio tools" and "edge2d tools". The system tray on the right indicates the date and time: "Tue 16 Apr, 10:54".

Class sim

```
# -----
```

```
class sim  
given a catalogues edge2d simulation it provides with tools to extract data
```

```
a simulation is defined by  
owner  
date  
sequence  
machine  
shot number
```

the following functions are available:

```
Read_print_file_edge2d()  
Write_print2file(simu_list,path,filename)  
Read_ionrecom  
Read_psol_pradpsol(ExtraInput)  
Read_time_data(var, interval)  
Read_pow_lfs_div(interval)  
Read_row_data(variable, region, Extrainput)  
Read_data(variable, ExtraInput)  
Read_edge2d_pedestal_fuel()  
Read_radiation_split(interval,ExtraInput)  
Read_qpeak_ot(ExtraInput)  
Read_prad_sol(ExtraInput)  
Read_imp_content(ExtraInput)  
Read_imp_content_core(ExtraInput)  
Read_n2(region)  
Read_e(ExtraInput)  
Read_x_point()  
Read_div_mol_atom_ratio(ExtraInput)  
Read_sirec_soun_lfs()  
Read_power_balance()  
Write_powerbalance2file(simu_list,path_filename)  
Read_omp(ExtraInput)  
Read_netflux(surface_in,surface_out)  
Eirene_netcur(surface_in,surface_out, ExtraInput)  
Write_eirene_cur2fil(simu_list, path, filename)  
Read_eirene_pump()  
Eirene_pumpcur(ExtraInput)  
Write_pump_cur2file(simu_list,path,filename)  
Read_e2d_ppf(dda,dtype, sequence, ExtraInput)  
Contour(ExtraInput, var, fname, lowerbound=None,upperbound=None)  
Execute_makeppf(simu_list, outputfile)
```




Applications Places System Tue 16 Apr, 10:36

MagSurf (SL) - V.0

File Edit View Help

JPN #

time [s]

configuration:

- FW
- GAP Strike Points
- XLOC (JPF)
- WALLS (JPF)
- EFIT (PPF)

dPsi[V/s] Core lines

SOL[cm] N lines

dPsi[V/s]

bruvio tools

Help

bruvio tools

by Bruno Viola (bruno.viola@uka.ac.uk)

xmessage Terminal bruvio tools MagSurf (SL) - V.0

EQDSK



Applications Places System Tue 16 Apr, 10:42

Terminal

```
File Edit View Settings Help
eqdsk tools
EQDSK
source
  g_p92121_t49.445_mod.e...
  eFit
  other
mapping tools
  write_input_matrix
  open input file
  run LCMSmap
  run LCMSXmap
  run SOLmap
  read
  7.4032
magnetic data
  get magnetic data LFE_81472 label IN
  write magnetic data LFEexp_JET_python label OUT
  normalize
  invert
  exit
Killed
bviola@bruvio:~/Python/bruvio_tool$ python G0_bruvio_tools.py
INFO Running bruvio tool.
DEBUG [G0_bruvio_tools.py:70] start
DEBUG [G0_bruvio_tools.py:79] USR in env
DEBUG [G0_bruvio_tools.py:87] this is your username bviola
DEBUG [G0_bruvio_tools.py:89] this is your homfold /u/bviola
DEBUG [G0_bruvio_tools.py:90] this is edge2d fold /work/bviola/Python/EDGE2D
DEBUG [G0_bruvio_tools.py:97] we are in /work/bviola/Python/bruvio_tool
INFO eqdsk tool
/work/bviola/JET/m15-20/exp/g_p92121_t49.445_mod.eqdsk
{'fixfree': False, 'efit': True}
```

Figure 1

The figure shows a contour plot of a magnetic field configuration. The x and y axes both range from -1.5 to 4.0. The plot features concentric contours representing magnetic flux surfaces, with values ranging from approximately 6.570 to 6.710. Two distinct data series are overlaid: a series of red dots forming an inner ring and a series of blue dots forming an outer ring. The contours are color-coded, with yellow and green representing higher flux values and blue representing lower values. A toolbar at the bottom of the window includes icons for home, back, forward, zoom, pan, and save.

Taskbar: xmessage Terminal [figures] bruvio tools eqdsk tools Figure 1

Plot Exp data



plot time traces

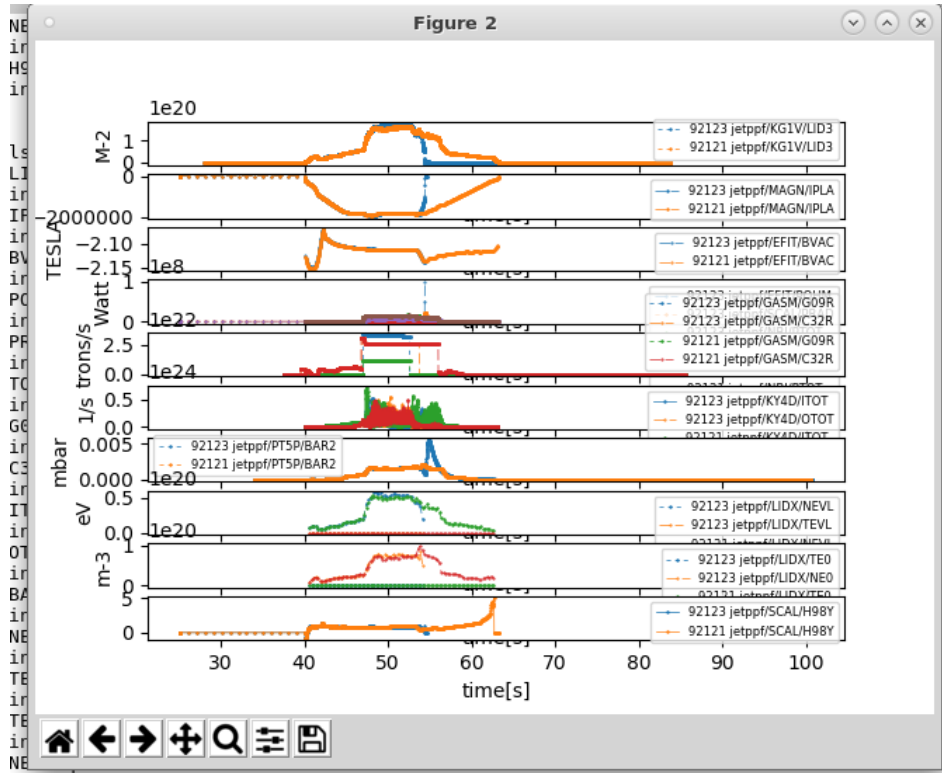
Pulse list

92123,92121

save Figure

select dataset edit_JSON

plot time traces



Select Standard set

Look in: /work/bviola/Python/kg1..._tools_gui/standard_set

Name	Size	Type
comparison_KG1_jetppf_chain1_bviola.json	1 KB	json File
comparison-KG1_chain1_bviola.json	2 KB	json File
KG1_vert_ch_new2.json	2 KB	json File
KG1_horz_ch_new2.json	1 KB	json File
horz_ch.json	606 bytes	json File
vert_ch.json	472 bytes	json File
vert_ch_new.json	2 KB	json File
KG1_input_g1r-lid.json	421 bytes	json File
PLASMA_main_parameters_new.json	2 KB	json File
horz_ch_new.json	1 KB	json File
comparison-KG1_jetppf_bviola.json	2 KB	json File
kg1r.json	957 bytes	json File
comparison_KG1_jetppf-kg1v_vs_chain1-kg1r.json	774 bytes	json File

File name: Open

Files of type: JSON Files (*.json) Cancel

Introduction



During 2016-2017 shutdown a major refurbishment of the laser system and software infrastructure has been done that improved substantially the reliability of the system as a whole

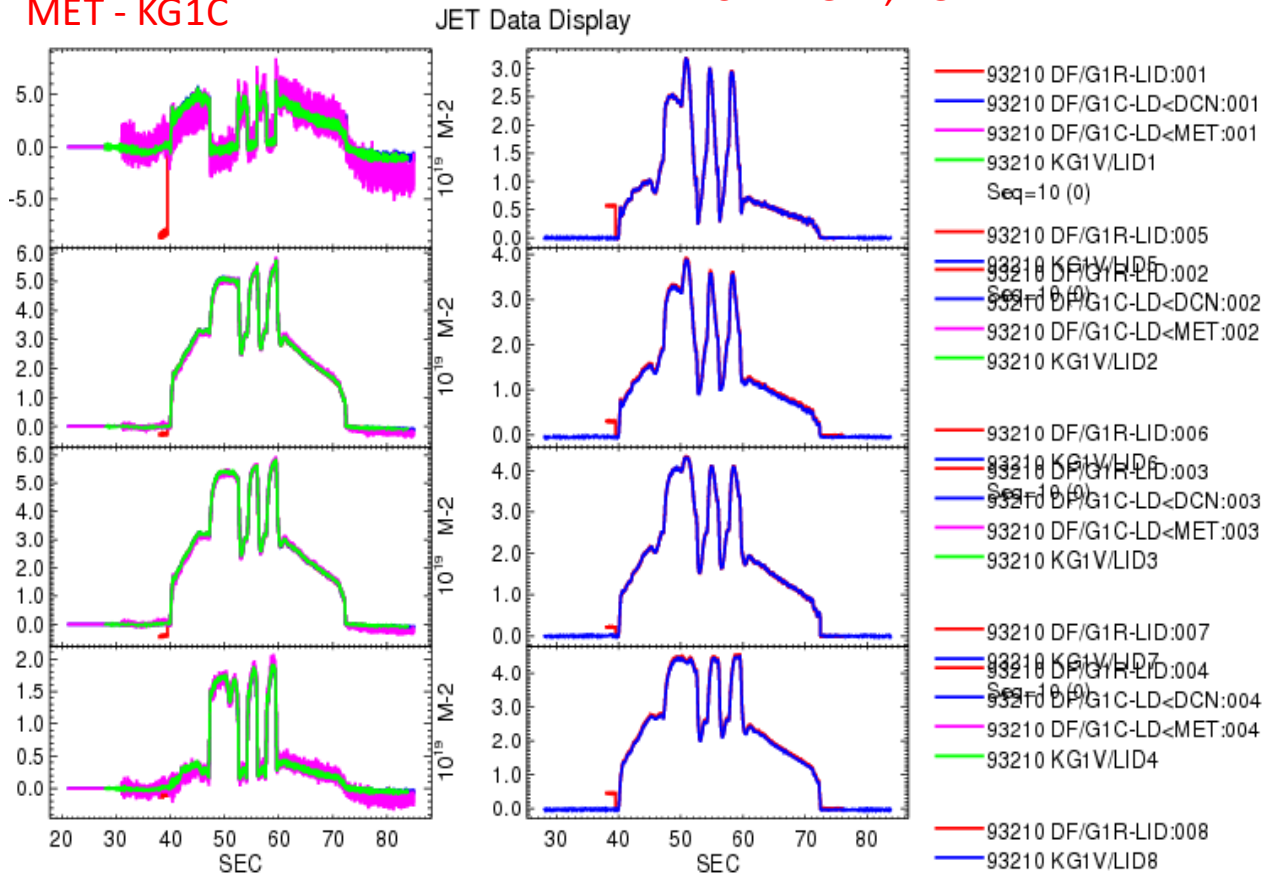
Vertical channels

DCN - KG1V, KG1C, KG1RT

MET - KG1C

Lateral channels

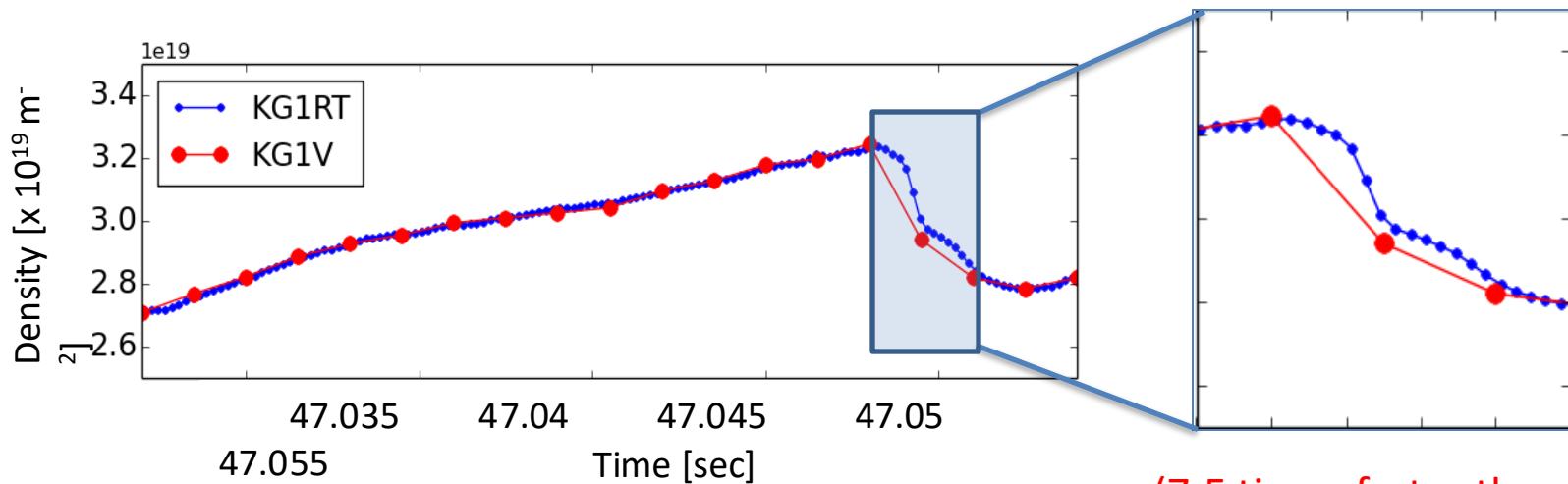
DCN - KG1V, KG1RT



Code Overview : KG1RT input



- **Default input** PowerPC (PPC architecture) VME
- **LIDs evaluated in software** using phase-sensitive detection algorithm (developed by R.Felton, A.Boboc and S.Dalley in 2016)
- Provides **8 x LIDs**
- **Time resolution 0.2 ms**, start time fixed, end time variable linked with plasma current signal

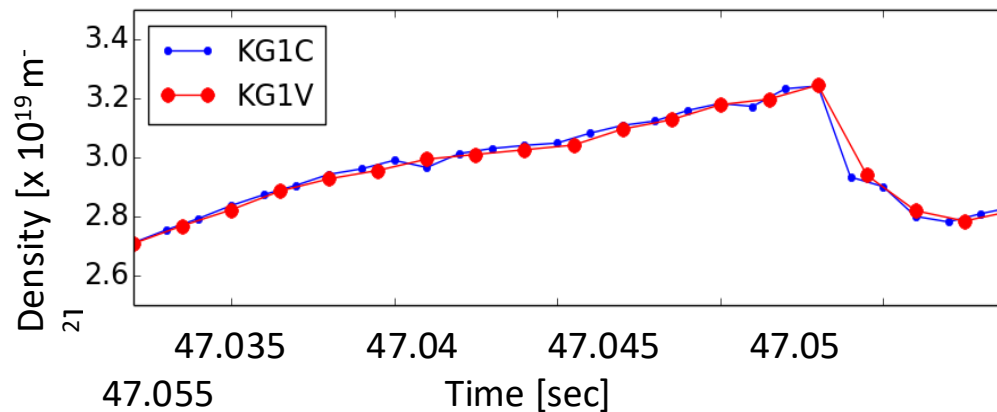


(7.5 times faster than KG1V!)

Code Overview : KG1C input



- **Backup option for vertical channels only ***
- **FPGA** based data acquisition + Linux Box for JPF/RT data production
- Analog zero-crossing with 25MHz clock of the phase is detected in hardware (much more precise than in software)
- **Built in fringe-jump algorithm** in real-time using two colours
- Provides a second colour measurement for LIDs
- 4x LIDs only (vertical)
- **Time resolution 1 ms, fixed start & end times**



Configuration file generator - LID chooser



KG1 chain 1 file generator

KG1 chain 1 generator v.2.0

LID1 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> LD DCN <input type="radio"/> LDCOR DCN <input type="radio"/> LD MET <input type="radio"/> LDCOR MET <input type="radio"/> LDRAW DCN	LID2 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> LD DCN <input type="radio"/> LDCOR DCN <input type="radio"/> LD MET <input type="radio"/> LDCOR MET <input type="radio"/> LDRAW DCN	LID3 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> LD DCN <input type="radio"/> LDCOR DCN <input type="radio"/> LD MET <input type="radio"/> LDCOR MET <input type="radio"/> LDRAW DCN	LID4 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> LD DCN <input type="radio"/> LDCOR DCN <input type="radio"/> LD MET <input type="radio"/> LDCOR MET <input type="radio"/> LDRAW DCN
LID5 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> KG1C LD <input type="radio"/> KG1C LDCOR <input type="radio"/> KG1C LDRAW	LID6 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> KG1C LD <input type="radio"/> KG1C LDCOR <input type="radio"/> KG1C LDRAW	LID7 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> KG1C LD <input type="radio"/> KG1C LDCOR <input type="radio"/> KG1C LDRAW	LID8 <input checked="" type="radio"/> KG1V <input type="radio"/> KG1RT <input type="radio"/> KG1C Best <input type="radio"/> KG1C LD <input type="radio"/> KG1C LDCOR <input type="radio"/> KG1C LDRAW

Pulse

KG1V IDL KG1R_py

- Originally created by A. Boboc and improved by L. Kogan to generate input file for KG1_py code
- It allows to modify configuration input file (kg1_const.ini) used by both KG1V_IDL and KG1R_py code

Process ReqCo requests



New ReqCo requires ROs to write new scripts to validate the date in ReqCo database

Process

lid3

lid4

lidall

SCAN DATABASE

run

SCAN OLD Requests

97137

run

SINGLE PULSE

97137

mark pulse as

DONE

IMPOSSIBLE

CLOSED

DOWNLOAD DATABASE

run

Thanks to D. Grist for the help!

Run KG1L/H codes

A screenshot of a graphical user interface window titled "run KG1L and KG1H codes". The window has a light gray background and standard window controls (minimize, maximize, close) in the top right corner. On the left side, there are two text input fields: the first is labeled "JPN" and contains the text "93212"; the second is labeled "userid" and contains the text "bviola". Below the "userid" field is a button labeled "go!". On the right side, there are two radio button options: "KG1L" (which is selected) and "KG1H". Below these is a "debug level" section with two radio buttons labeled "1" (selected) and "2". At the bottom right, there is a checked checkbox labeled "print to screen".

- KG1L/H PPFs are now produced on requests (not by chain1 manager)
- Agreed between Chain1 manager and KG1 RO
- **Batch procedure to be implemented**



```
bviola@heimdall025 14:15:22 /u/bviola/work/Python/kg1_tools/kg1_tools_gui (master) \  
$ python G0_kg1_tools.py  
INFO Running KG1 tool.  
INFO Logbook folder will be  
INFO /work/bviola/Python/kg1_tools/kg1_tools_gui/kg1_tools_logbook  
INFO last reprocessing done on 2018-07-20 - 669 days ago (668)  
INFO last reprocessing done on 2018-09-14 - 613 days ago (612)  
INFO last reprocessing done on 2018-12-07 - 529 days ago (528)  
INFO last reprocessing done on 2019-04-01 - 414 days ago (413)  
INFO last reprocessing done on 2019-06-26 - 328 days ago (327)  
INFO last reprocessing done on 2019-08-06 - 287 days ago (286)  
INFO last reprocessing done on 2019-09-17 - 245 days ago (244)  
INFO last reprocessing done on 2019-10-11 - 221 days ago (220)  
INFO last reprocessing done on 2019-10-29 - 203 days ago (202)  
INFO last reprocessing done on 2019-11-08 - 193 days ago (192)  
INFO last reprocessing done on 2019-12-03 - 168 days ago (167)  
INFO last reprocessing done on 2020-01-14 - 126 days ago (125)  
INFO last reprocessing done on 2020-01-27 - 113 days ago (112)  
INFO last reprocessing done on 2020-02-11 - 98 days ago (97)  
INFO last reprocessing done on 2020-02-25 - 84 days ago (83)  
INFO last reprocessing done on 2020-03-02 - 78 days ago (77)  
INFO last reprocessing done on 2020-03-13 - 67 days ago (66)  
INFO last reprocessing done on 2020-03-17 - 63 days ago (62)  
INFO last reprocessing done on 2020-03-25 - 55 days ago (54)  
INFO last reprocessing done on 2020-05-14 - 5 days ago (4)  
INFO  
INFO INIT DONE  
□
```



run KG1L and KG1H codes

JPN

read_uid

userid

KG1L
 KG1H
 both

debug level:
 1 2

print to screen

setup KG1_py code

JPN

read uid

write uid

options

plot figure
 save figure
 interactive

final data
 data corrections
 timings
 elms
 pellets

kg4_data
 kg1_data
 statistics
 comp_self
 comp_kg4
 kg4_cmdata

kg1_correct_one_color
 kg1_correct_two_color_lk
 pellet_data
 mag_data
 elms_data
 kg1_correct_large_jumps

debug level

0
 1
 2
 3
 4

decimate KG1R to KG1V Test mode Force



RO KG1 local database

Search PPF in JET database

pulse 1

pulse 2

pulselist

 kg1v kg1l kg1h kg1r

Search for validated pulses

Update Database

Search pulses to reprocess

Status Flag checker

JPN

userid

