



PSPA

Platform for Simulation of Particle Accelerators

On behalf of the PSPA Team:

M. Biagini, INFN-LNF and LAL

7th LER Workshop, CERN, Jan. 17th 2018

PSPA @ LAL



- **PSPA** (**P**latfom for **S**imulation of **P**article **A**ccelerators) is a user-friendly interactive web-based platform built at IN2P3-LAL for the design of accelerators
- PSPA aims at containing the tools to make a **start-to-end simulation** of any accelerator, and run interactively **most commonly used simulations codes** available worldwide:
 - suitable for testing different physical models
 - avoid time-consuming and error prone process of translating data formats between codes
 - useful for students training in accelerator design

PSPA Team

➤ Several accelerator physicists, engineers and computer science experts joined forces to develop the PSPA platform at LAL:

➤ **PSPA Team (FTE):**

- *Francois Touze, SI, LAL* (60%)
- *Antoine Perus, SI, LAL* (20%)
- *Marc Nicolas, SI, LAL* (40%)
- *Marica Biagini, SI/DEPACC, Bourse D'Alembert and INFN-LNF* (80%)

➤ **PSPA consultants:**

- *Hayg Guler, DEPACC, LAL*
- *Christian Helft, LAL, retired*
- *Guy Le Meur, LAL, retired*
- *Alessandro Variola, Accelerator Division, INFN-LNF*

Blue: expertise in accelerator design

PSPA objectives

- Final goal is to get a flexible tool that allows to easily link different models, corresponding to different physics problems, in a unique description, allowing for complex systems analysis and start-to-end simulations in intricate environments
- PSPA is not just a data conversion tool but a real design and simulation instrument

PSPA features

- Possibility to load an input file (*user's file format from the available codes*) or **interactively design a lattice** just by adding components like bricks
- Tests of the accelerator model by varying parameters and performing several commands with different codes
- Each simulation code is available in the form of a “*black box*”, the user is free to choose among the proposed codes, which are usually run standalone, the more suitable to his needs **▀ code benchmarking**

Available codes at present

TRANSPORT
**A COMPUTER PROGRAM FOR DESIGNING
CHARGED PARTICLE BEAM TRANSPORT SYSTEMS**

http://lss.fnal.gov/cgi-bin/find_paper.pl?nal-091

PARMELA **LAL version**
(Phase and Radial Motion in Electron Linear Accelerators)

http://laacg.lanl.gov/laacg/services/serv_codes.phtml



<http://madx.web.cern.ch/madx/>

elegant

ASTRA

A Space Charge Tracking Algorithm

<http://www.desy.de/~mpyflo/>

[http://www.aps.anl.gov/Accelerator Systems Division/Accelerator Operations Physics/software.shtml](http://www.aps.anl.gov/Accelerator%20Systems%20Division/Accelerator%20Operations%20Physics/software.shtml)

B BETA

<http://irfu.cea.fr/Sacm/logiciels/index6.php>

PSPA web interface

http://pspa-dev.lal.in2p3.fr/?_=/pspa

guest Logout



Version 0.5.0 - 2017-12-15 08:48 +0000

ABOUT PSPA LOAD FROM SERVER SAVE TO SERVER LOAD FROM USER SAVE TO USER **RESET** 

Accelerator elements

CATHODE BEAM DRIFT
BEND LINAC CAVITY SOLENO
RF CAVITY MARK MPOLE
QUAD SEXT TR WAVE

Sectors list

 sector1

Build Simulate

sector1 

Drag elements here....

Machine content

sector1

Console

Two toggles: **BUILD** and **SIMULATE**

PSPA windows: BUILD

Build a lattice from scratch or from file, modify lattice and parameters

1 **2** **3** **4**

ABOUT PSPA LOAD FROM SERVER SAVE TO SERVER LOAD FROM USER SAVE TO USER RESET

1. Load file from server (previously saved)
2. Save to server (after execution)
3. Load from user computer (local)
4. Save to user computer (local)

Accelerator elements

CATHODE BEAM DRIFT

BEND LINAC CAVITY SOLENO

RF CAVITY MARK MPOLE

QUAD SEXT TR WAVE

Sectors list

- s00
- ka1
- ma
- ma

Console

restore Lattice:: hexa done

Build: hexa Simulate

ka2

DRIFT BEND DRIFT QUAD DRIFT QUAD DRIFT BEND

dr6 kick2 dr6 qp2 dr6 qp1 dr8 sept2

Delete Delete Delete Delete Delete Delete Delete Delete

Machine content

s00 ka1 ma -ma kb -kb ma -ma ka2

bend

Label: kick2

Angle (rad): -0.011

Length (m): 0

Radius (m): -18.1822

Momentum (MeV/c): 0

Aperture (cm): 1e+06

Pole face rotations (rad) - Entrance: 0 Exit: -0.011

Fringe field integral: 0 Half gap (m): 0

Cancel OK

PSPA windows: SIMULATE

Simulate different Actions with different Codes

The screenshot shows the PSPA SIMULATE window. At the top, there are buttons for 'ABOUT PSPA', 'LOAD FROM SERVER', 'SAVE TO SERVER', 'LOAD FROM USER', 'SAVE TO USER', and 'RESET'. Below these are 'Accelerator elements' (CATHODE, BEAM, DRIFT, BEND, LINAC CAVITY, SOLENO, RF CAVITY, MARK, MPOLE, QUAD, SEXT, TR WAVE) and a 'Sectors list' (linac, TL, ka1, ma). The main 'Computing blocks' section is annotated with a red circle around the 'FROM' and 'TO' dropdowns. A yellow box at the top right lists: 1. RUN, 2. Output from code, 3. User plots, 4. MADX plots. A blue circle highlights the 'Action' buttons (tracking, twiss), and a black circle highlights the 'Software' buttons (generator, astra, madx, beta). A 'run number' field shows '0'. The bottom of the image contains three columns of text: 'Pick lattice section (block)', 'Select Action: Twiss, Matching, Tracking, Emittance', and 'Select Code: Generator, ASTRA, Transport, Beta, MADX, Elegant'.

1. RUN
2. Output from code
3. User plots
4. MADX plots

Pick lattice section (block)

Select Action:
Twiss
Matching
Tracking
Emittance

Select Code:
Generator
ASTRA
Transport
Beta
MADX
Elegant

PSPA actions: Matching

- **Matching:** optical functions, global parameters (tunes, chromaticity), transfer maps

The image displays the 'ACTION command : matching' dialog box in the PSPA software, which is used to configure matching actions. The dialog is divided into several sections:

- matching commands:** Includes checkboxes for 'initial values', 'constrain optics parameters', 'global constraints', and 'constraints on Transfer map'. Red arrows point from these options to their respective sub-dialogs.
- optimization setup:** Includes fields for 'mode' (minimize), 'method' (none), 'target' (none), 'tolerance' (-0.01), 'evaluations' (500), 'passes' (none), and 'restarts' (none). It also has an 'equation' field set to 'none'.

Four sub-dialogs are shown, each linked by a red arrow from the main dialog:

- twiss at Entrance:** A dialog for setting initial values for beta_x, alpha_x, dx, beta_y, alpha_y, and dpx. All fields are currently set to 'none'.
- constrain optic parameters:** A dialog for setting simple constraints and attributes to be varied for parameters qp1 through qp41. It includes a 'range' field (none), a 'beta_x' dropdown (none), and a 'weight' field (none). The 'attributes to be varied' section has a table with columns for parameter name, step, lower, and upper bounds, all currently set to 'none'.
- constrain global parameters:** A dialog for setting global constraints and attributes to be varied. It includes a 'tune/chromaticity' dropdown (Q1), a 'weight' field (none), and an empty 'attributes to be varied' section.
- Constraints on Transfer matrix:** A dialog for setting components of the transfer matrix. It includes a 'range' field (#e), an 'ij' field (none), a 'value' field (none), and a 'weight' field (none).

PSPA actions: Twiss

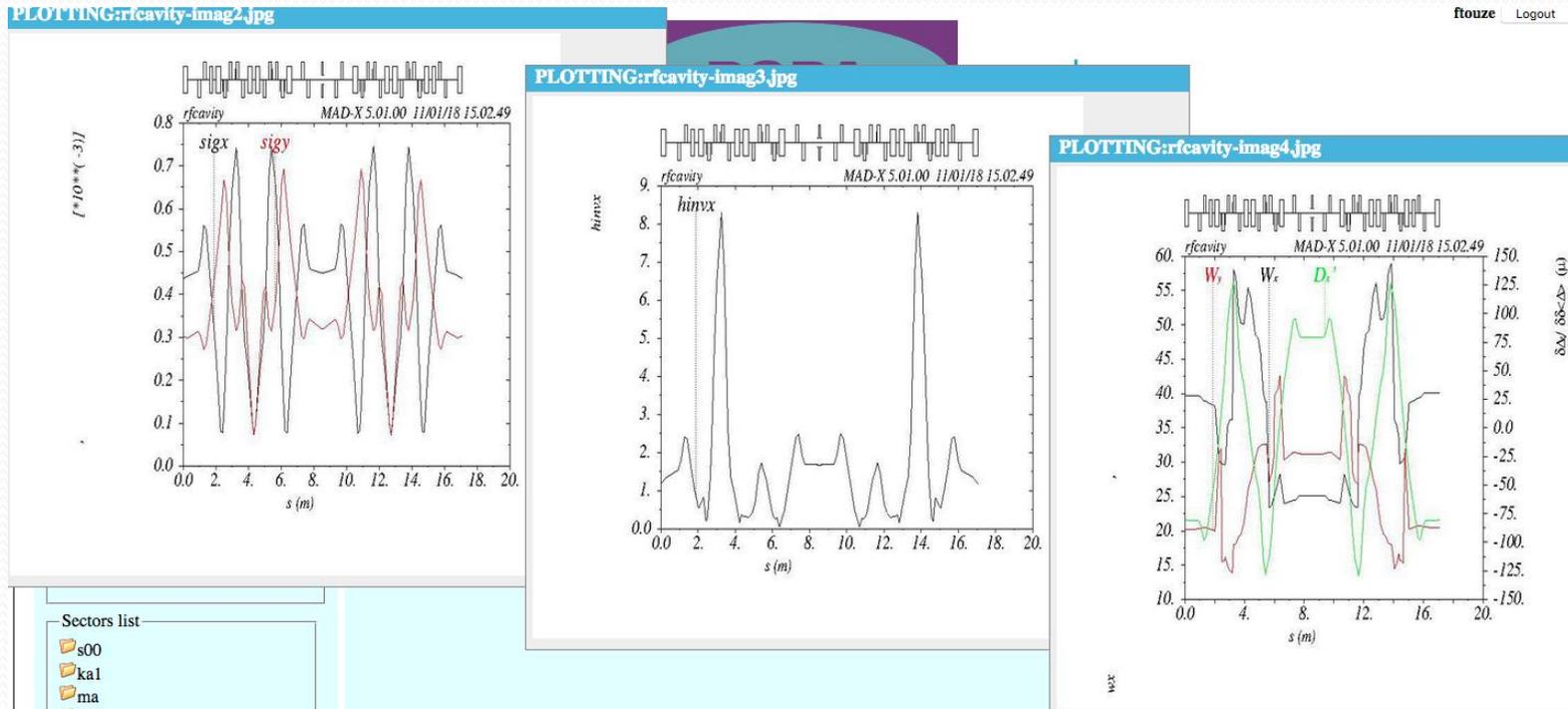
- **Twiss**: optical functions calculation. Plots available: user, MADX

The screenshot displays the PSPA software interface with several key components:

- Top Panel:** Includes buttons for 'ABOUT PSPA', 'LOAD FROM SERVER', 'SAVE TO SERVER', 'LOAD FROM USER', 'SAVE TO USER', and 'RESET'. Below these are 'Accelerator elements' (CATHODE, BEAM, DRIFT) and 'Build: hexa' / 'Simulate' options.
- ACTION command : twiss:** A dialog box with 'twiss commands' (radio buttons for 'load parameters', 'initial values', and 'periodic solution', with 'periodic solution' selected) and 'output functions' (checkboxes for 'chromatic functions' and 'beam parameters').
- twiss parameters:** A dialog box for 'initial values' with input fields for 'beta_x', 'alpha_x', 'dx', 'beta_y', 'alpha_y', and 'dpx', all currently set to 'none'.
- graphical analysis:** A window with a 'runId' field (value: 1) and a 'plots' section with radio buttons for 'envelopes', 'beam parameters', 'phase space after element', 'histogram after element', and 'optical functions' (selected). It also has dropdown menus for 'x', 'sigma-x', 'b00', 'courant_snyder', 'x', 'xp', 'b00', 'dE/E', and 'betx,bety'.
- twiss functions: hexa (1):** A plot showing two oscillating curves (red and blue) over a distance 's [m]' from 0.0 to 16.0. The y-axis ranges from 0 to 12.
- PLOTTING:hexa-imag1.jpg:** A plot showing multiple oscillating curves (black, red, green) over a distance 's (m)' from 0.0 to 20.0. The left y-axis is labeled 'I B' and the right y-axis is labeled 'D (m)'. The plot includes text: 'hexa', 'MAD-X 5.01.00 11/01/18 14.58.46', and labels for β_x , β_y , and D_x .

PSPA actions: Twiss

- Twiss: calculation of:
 - ✓ beam sizes
 - ✓ H invariant
 - ✓ chromatic W functions, second order dispersion



PSPA actions: Tracking

- **Tracking:** in Linac, gun. Defines beam parameters, output file, integration steps and range, actions (space charge, aperture)

ACTION command : tracking

newRun

FileName
Qbunch [nC] Xoff [mm] Yoff [mm]
Track_All Auto_Phase

output

ZSTART [m] ZSTOP [m]
Zemit Zphase

namelists

SPCharge aperture error
 scan modules

Ok Cancel

namelist APERTURE

parameters

#	File_Ap	Ap_Z1 [m]	Ap_Z2 [m]	Ap_R [mm]
1	RAD	0.000	0.020	40.
2	RAD	0.020	0.041	15.
3	RAD	0.041	0.066	42.
4	RAD	0.066	0.087	15.
5	RAD	0.087	0.112	42.
6	RAD	0.112	0.1615	15.
7	RAD	0.1615	0.2835	17.5
8	RAD	0.2835	0.3655	19.
9	RAD	0.3655	0.6865	30.
10	RAD	0.6865	0.8525	50.
11	RAD	0.8525	6.0	17.5

Cancel OK

namelist SPCHARGE

parameters (for cylindrical grid algorithm)

Nrad Cell_var Nlong_in
N_min min_grid [m]
Max_scale
Lmirror

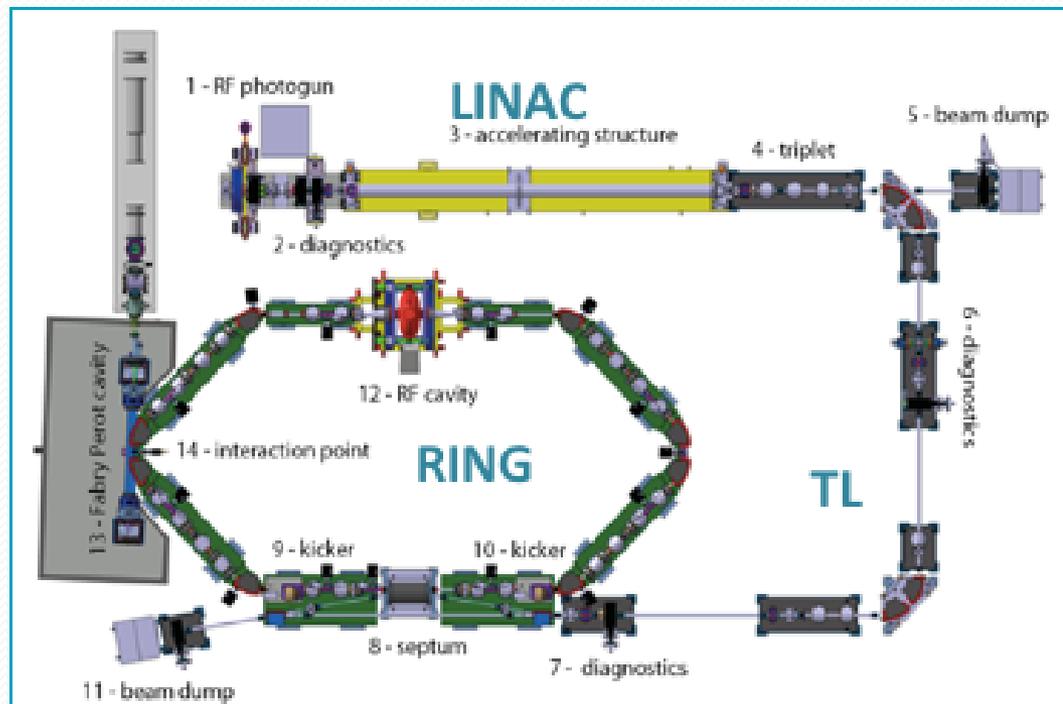
Cancel OK

PSPA actions

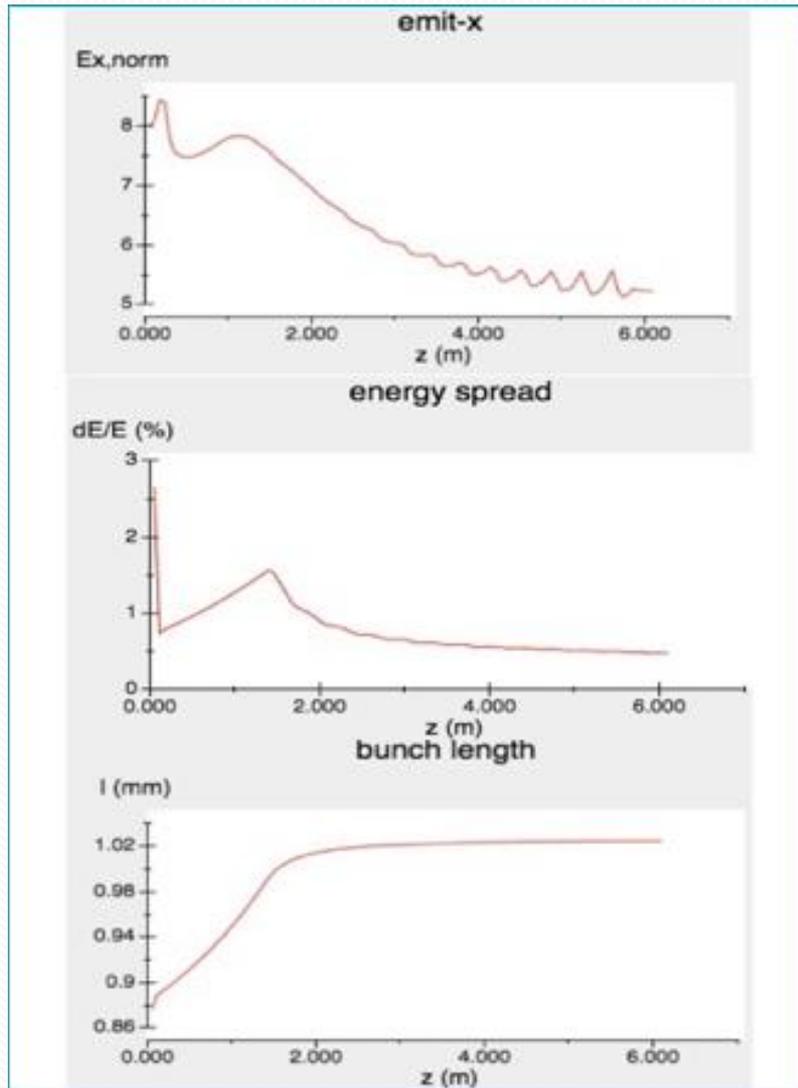
- Available EMIT (calculation of emittance, damping times etc...) within MADX
- In progress implementation of:
 - tunes and optical functions dependence on momentum
 - tunes dependence on particle amplitude

Application to ThomX

- The **ThomX** project revealed itself as a real-world use case of choice for PSPA: **Gun** + **Linac** + **Transfer Line** + **Ring**
- ASTRA code was used to simulate beam dynamics from the RF gun to the end of the Linac, BETA code for the TL dynamics, and MADX code for the ring **👉 PSPA makes it easier and faster!**



PSPA simulations of Thomx linac with ASTRA code

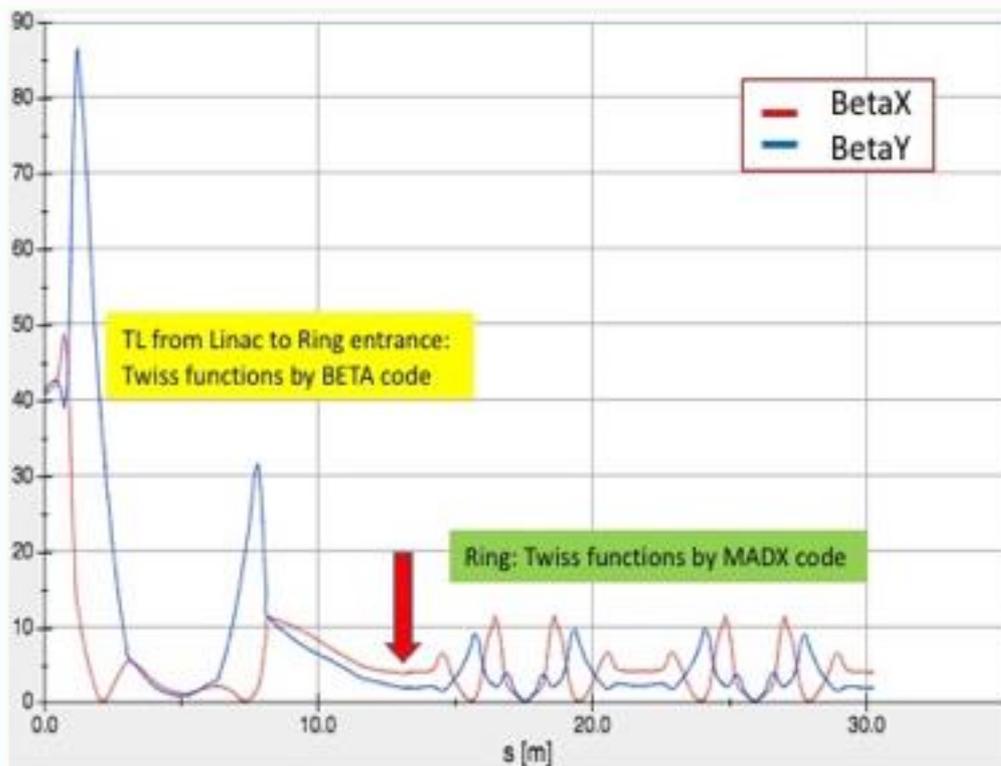


Evolution of horizontal normalised emittance (top), energy spread (middle) and bunch length (bottom) in ThomX Linac as computed by ASTRA code in PSPA

Since ASTRA computing time can be long, depending on the number of particles and space charge, it is possible to store the results from one run to be used later

PSPA simulations of ThomX Transfer Line and Ring with BETA and MADX

Example of codes concatenation



Optical functions from Linac end through TL and ThomX Ring (computed by Beta and MADX codes)
The red arrow shows the concatenation point of the two codes

Each time a simulation is performed the variables are updated, and the results can be saved for a single part or for the whole

PSPA WEBSITE

To be completed
with a “demo” (in
progress) and some
documentation

<https://groups.lal.in2p3.fr/pspa/>

Would like to
add a “forum”
for exchanges of
information,
updates and
debugging



PSPA: Design your Accelerator! Simulation Codes Documentation Who are we Contact us

PSPA: Design your Accelerator!

PSPA is a web-based platform to design accelerators and simulate beam physics

PSPA aims at optimizing the work of accelerator designers by factoring once and for all the time-consuming and error prone process of translating data formats between the various codes involved in the modelling of a machine, controlling the repeated execution of these models by easily varying some parameter and managing the associated data and finally provide a convenient mean for testing different physical models of a given part of a machine (a truly innovative feature).

With PSPA you will be able to:

- describe parts of, or a whole, accelerator using a construction kit and a collection of elementary components such as magnets, accelerating cavities, and so on ;
- import an existing machine (lattice) in several standard formats;
- assign a modeling program to each component or group of components, to be chosen from a range of standard codes available;
- change setting parameters;
- run a simulation of the machine;
- simulate several beam physics processes;
- look, plot, save results for further use;
- export results and files on user area;
- simulate the same device/lattice with different codes for modeling and comparison.

To be noted: PSPA is NOT a simple “translator” from code to code, but a real accelerator design tool



What's next

- Several improvements and new features are on the PSPA road map:
 - improved **user interface** and **user experience** (UI/UX)
 - capability to add/run **user-written codes**
 - plot of the machine geometric layout
 - use of Cloud/Grid computational resources
 - add errors in Astra and tracking in Elegant

Lesson learned

- For the success of PSPA it is very important to have a close interaction between:
 - software experts: build the interface in all its aspects (coding, graphical interface, user requests, website, server management, etc...)
 - machine experts: suggestions on what to implement, how to improve the physics processes, tests, inputs on other existing tools in the accelerator community
 - users: feedback on problems, reports on user-friendliness, suggestions for new features

PSPA ergonomics improvement

- To make PSPA a tool widely used by the accelerator community we need **an improvement of its interface**  make it more “**user friendly**” from the point of view of accelerator designers:
 - easy to use
 - intuitive
 - offering most used simulation and graphics tools



For the ergonomics of the platform 4 companies are being scrutinized, work will start in February by the chosen one

Conclusions

- PSPA aims at becoming a useful tool both for accelerator designers and accelerator physics beginners
- Its first stage is ready to be released
 - lattice design, tracking and matching
- The close interaction with accelerator designers is essential to develop in PSPA the needed physics applications: **user needs?**
- For PSPA dissemination we need a better and more intuitive user interface **👉 new ergonomics soon**