PSPA: A WEB PLATFORM FOR SIMULATION OF PARTICLE ACCELERATORS *

M.E. Biagini[†1], L. Garnier, H. Guler, C. Helft, G. Le Meur, M. Nicolas, A. Pérus, F. Touze, IN2P3, LAL, Univ. Paris-Sud, CNRS/IN2P3, Université Paris-Saclay, Orsay, France
A. Variola, INFN-Laboratori Nazionali Frascati, Frascati, Italy
[1]also at INFN-Laboratori Nazionali Frascati, Frascati, Italy

Abstract

PSPA [1] (Platform for Simulation of Particle Accelerators) is an original web-based interactive simulation platform for designing and modelling particle accelerators created at Laboratoire de l'Accélérateur Linéaire, Orsay. It aims at eventually containing all the tools to perform a start-to-end simulation of an accelerator, and making it possible to run interactively, and in a cooperative way, several open source simulations codes available worldwide. At the moment, the focus is on electron/positron accelerators. PSPA will foster the work of accelerator designers by factoring once and for all the tedious, 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. Moreover, as a truly innovative feature, it will provide a convenient means for testing different physical models of a given part of a machine. It has to be noted that PSPA is not just a data conversion tool but a real design and simulation instrument. The status of the project is described in this paper, and examples of its application to the ThomX Compton backscattering source [2] at LAL are presented.

MOTIVATIONS AND OBJECTIVES

In the last decades, particle accelerators have become not only necessary for fundamental particle and nuclear physics studies, but also instruments used for medical, technological and educational applications, with their number enormously increased in the last years. Sophisticated simulations of a large number of components, as close as possible to the real ones, and detailed studies of the physical phenomena occurring to the beams are needed, in order to be able to build and operate an accelerator with the expected performances. Moreover, depending on the required beam characteristics, the same physical phenomenon can be described by different theoretical models, using different algorithms. For a deep knowledge of all the processes involved, it is also very important to link the simulations built for accelerator design optimization, done by adjusting different parameters, with the more realistic, but also more time-consuming, particle tracking so-called PIC (Particle In Cell) codes.

Nowadays, accelerator physicists worldwide have available quite a large number of simulation codes specific for each application. However, they are scattered in several laboratories, there may be several versions of the same code depending on the application, their data formats are usually not compatible, there are different computer environments, there is a general lack of documentation and, not less important, a lack of dissemination. This makes the accelerator design slower and sometimes painful, often requiring a transfer and format translation of data from one simulation code to another. It is also common to find, for specific physical problems, code developed in stand-alone mode by different users, without documentation and difficult to be made available for a general use.

In this framework, it is extremely important to have a set of coherent tools to be able not only to carefully model the accelerator in all its components, but also to compare the real device behaviour with different physical models.

The final goal of PSPA is to get a flexible tool that allows to easily link different models, corresponding to different physics problems, in a unique description allowing complex systems analysis and start-to-end simulations in intricate environments, with two main objectives:

- Optimize the work of an accelerator designer by leveraging the 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.
- Provide a convenient way to test different physical models of a given part of a machine (a truly innovative feature).

WEB PLATFORM

In order to design a lattice and perform beam dynamics simulations, the PSPA web platform allows for several codes to be run. Each code is available in the form of a "black box", easy to run as many times as needed, and at the same time easy to configure. The user is free to choose among the proposed codes for any given part of a machine, instead of traditionally run stand-alone, isolated and incompatible programs. It is therefore very straightforward to compare results from different codes.

It is possible to interactively design an accelerator just by connecting its individual components like bricks, as in a construction set kit, freely choosing and varying their parameters, or load a user file describing the lattice in one of the available codes. Once described the machine layout, it is possible to define separate "regions" where the suitable simulation code can be selected and run, and then concatenate the results to get a start-to-end description of the beam dynamics.

In more details, three panes allow to:

* Work supported by: Bourse D'Alembert, Univ. Paris-Saclay, France
† email address: biagini@lal.in2p3.fr

1. build a lattice from scratch, by dragging most common accelerator elements and defining their properties, or by loading a file;
2. manage the lattice, for example by duplicating and/or adding/removing elements;
3. define and select each region of interest, pick a command, select a code and run. Several regions can be concatenated and beam dynamics can be simulated with different codes.

The built lattice is graphically represented by a sequence of elements as icons and the user can easily change the element's properties if needed.

An "Output" window shows the code execution log. Simulation results are saved on user's area, and plots are available for a number of variables.

At present PSPA is able to (Figure 1 shows an example of the RUN window):

- Download/upload files;
- Build a machine lattice: from scratch or from file;
- Modify the lattice;
- Pick a lattice section ("block");
- Select a command (Twiss, Match, Track,…);
- Pick a simulation code from a list;
- RUN the code on the lattice chosen;
- Look at output and plots, custom-made or from the code;
- Save, modify, re-run if needed.

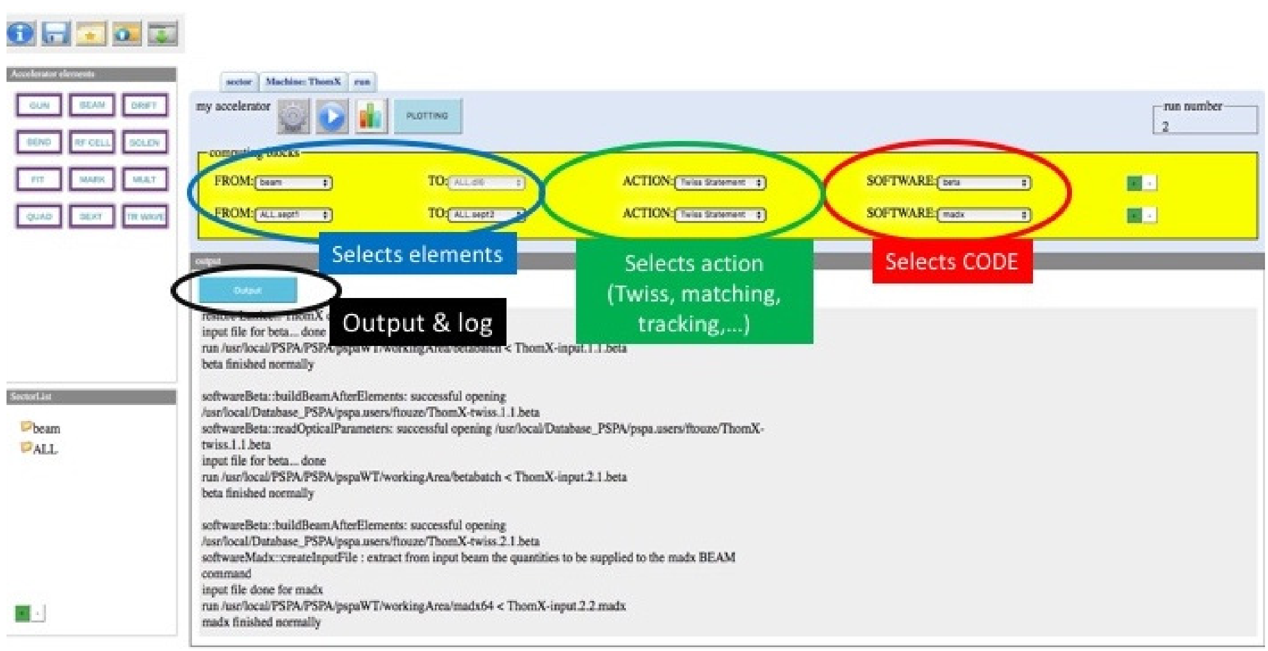

Figure 1: Example of a PSPA Run window.

ARCHITECTURE OF PSPA

PSPA's architecture is based upon a data bus sharing the lattice description among simulation codes using an XML common format with the Accelerator Markup Language (AML) [4]. The Universal Accelerator Parser (UAP) [5], initially developed to handle AML and MAD formats, is used to exchange lattice information in other formats.

This allows PSPA to read a variety of different file formats and to convert them in order to use otherwise incompatible simulation codes.

The PSPA engine works according to the client-server model using state of the art web technologies (HTML5, CSS3 and JavaScript); it is developed with the Web toolkit Wt [3], a C++ library which offers interface design patterns tailored to the web.

SIMULATION CODES

An effort has been put on supporting the most used open source simulation codes for beam dynamics of electron/positron beams on the platform: Parmela@LAL (LAL private version) [6] and ASTRA [7] for the simulation of injectors and Linacs, TRANSPORT [8] for trajectory tracking, BETA [9], Elegant [10] and MADX [11] for optics design, matching and tracking in Linacs, Transfer Lines and Rings. Once the platform is fully operative, the plan is to add other codes as requested by users. Implementation of machine errors procedure is also under development. A web site [1] has been setup which will soon provide tutorials, instructions and demos for users.

APPLICATION TO THOMX

The ThomX project, a 50 MeV Compton backscattering facility, revealed itself as a real-world use case of choice for PSPA. Figure 2 shows a layout of the com-

ISBN 978-3-95450-182-3

plex, which includes a photo-injector, a Linac with one LIL-type accelerating cavity, a short Transfer Line (TL) to guide the electron beam to the small ring. PSPA was used for a start-to-end simulation of the whole complex. The ASTRA code was used to optimize beam dynamics from the RF gun to the end of the Linac, Elegant and BETA for the TL dynamics, and MADX for the ring. Some of the simulation results are shown in the following plots (PSPA made).

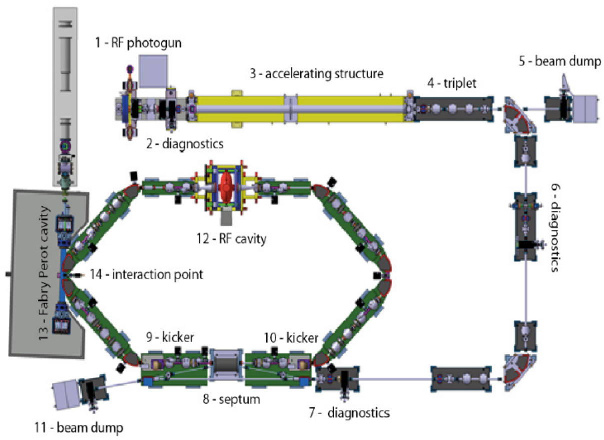

Figure 2: ThomX complex layout.

Figure 3 shows the evolution of horizontal normalised emittance, energy spread and bunch length along the ThomX Linac for one of the configurations studied, as computed by ASTRA inside PSPA.

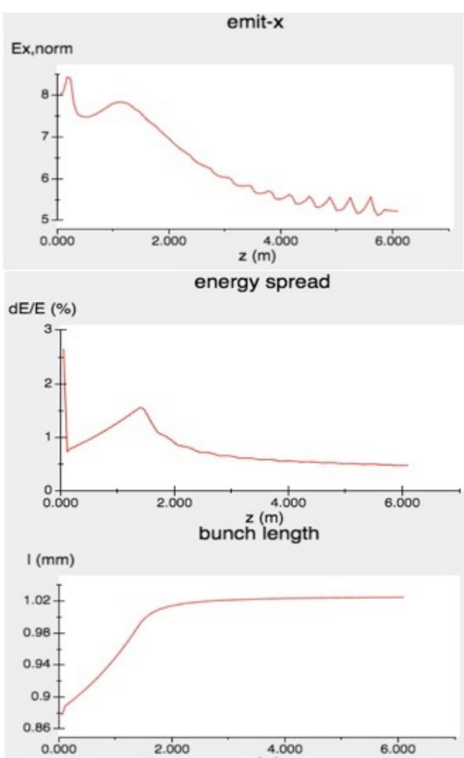

Figure 3: Evolution of horizontal normalised emittance (top), energy spread (middle) and bunch length (bottom) in ThomX Linac as computed by ASTRA code in PSPA.

Figure 4 shows one of the unique PSPA features: Twiss functions were computed in a single run with the BETA code for the TL section and with MADX for the ring, and results were then concatenated. This allows for an easy match of the TL values at the ring entrance and exploiting different code capabilities, without the hassle to translate data from one code to the other.

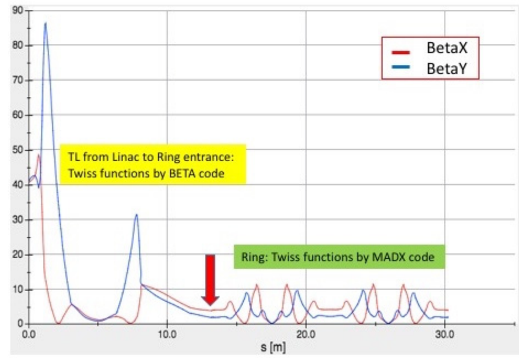

Figure 4: Optical functions from Linac end through TL and ThomX Ring (computed by Beta and MADX codes). The red arrow shows the concatenation point of the two codes.

Next steps in this study, in order to perform a start-to-end simulation, will be:

1. develop a special interface to be able to study the parameters tolerance to random machine errors;
2. translate the ASTRA results from the Linac into the Elegant code (already implemented in PSPA) to study the beam dynamics in the TL.

PERSPECTIVES AND CONCLUSIONS

PSPA aims at providing a useful tool both for accelerator designers and accelerator physics beginners. At the moment, it can be used for lattice design and matching, but the goal is to get as many as possible codes for up-level simulations (collective effects for example).

Several improvements are in PSPA road map:

- addition of the SAD [12] code to external application list (for simulation of lepton colliders beam dynamics and luminosity optimization);
- improved user interface;
- capability to run user-written code;
- plot of the machine geometric layout;
- off-line mode for time-consuming simulations;
- use of Virtual Data [13] massive computational resources;
- desktop based standalone application.

Accelerator designers are invited to use PSPA and help in implementing new features.

ACKNOWLEDGEMENT

PSPA designers warmly thank the authors of the open source simulation codes which have been installed in the platform [6,7,8,9,10,11].

REFERENCES

[1] https://groups.lal.in2p3.fr/pspa/.

[2] ThomX Technical Design Report, LAL/RT 14-21, in2p3-00971281, April 2014

[3] https://www.webtoolkit.eu/wt

[4] D. Sagan *et al.*, "The Accelerator Markup Language and Universal Accelerator Parser", Proc. EPAC 2006, Edinburgh, Scotland, June 2006.

[5] D. A. Bates *et al.*, "The Universal Accelerator Parser", Proc. ICAP 2006, Innsbruck, Austria, May 2006.

[6] B. Mouton *et al.*, PARMELA@LAL, http://laacg.lanl.gov/laacg/services/serv_codes.phtml

[7] K. Floetmann, ASTRA, http://desy.de/~mpyflo/.

[8] K. Brown et al., TRANSPORT, http://lss.fnal.gov/cgibin/find_paper.pl?nal-091

[9] Payot et al. BETA, http://irfu.cea.fr/Sacm/logiciels/index6.php

[10] M. Borland & al., Elegant, http://www.aps.anl.gov/Accelerator_Systems_Division/Accelerator Operations Physics/software.shtml

[11] MAD team, MADX, http://madx.web.cern.ch/madx

[12] K. Oide et al., SAD, http://acc-physics.kek.jp/SAD/

[13] Virtual Data, https://indico.cern.ch/event/595396/contributions/2532580/attachments/1451140/2238052/Orsay_Datacenter_Extension.pdf

ISBN 978-3-95450-182-3