

# DATA MANAGEMENT AND ANALYSIS FOR BEAM DYNAMICS SIMULATION

D. Zyuzin, Forschungszentrum Juelich, Germany

S. Andrianov, N. Kulabukhova, Saint Petersburg State University, Russia

## Abstract

Computer simulation of modern accelerator system is based on a number of specialized computer codes. Complexity of concerned processes and interpretation of simulation results requires comfortable and effective tools to design accelerator structure and beam characteristics, carry out computer experiments, process and visualize data. This paper proposes a prototype system with web-interface which allows the full research cycle: from lattice generation to data visualization. This approach represents a valuable tool for beam physicist providing methods to benchmark simulation engines as well as providing additional instruments for understanding physical processes in accelerator. The corresponding tools were used in application to the spin-orbit motion problems in electrostatic accelerators.

## INTRODUCTION

In the recent years the development of the industry accelerator facilities for various purposes has led to the need to revise certain provisions of the design process of new systems and maintenance of existing systems. Operation of such expensive facilities is not possible without the corresponding information support, as well as during the design phase and operational phase. To increase the efficiency of processes in present were developed various technologies, among which, first of all, we need to mention the EPICS project [1], which is an integrated set of software and hardware for design and automated management and control systems for large scientific projects. However, the use of EPICS is possible only if there is an interaction between the relevant mathematical models and an integrated information processing environment. In particular, for this purpose can effectively be used concept of LEGO objects [2]. In this paper we consider development concept of special software that provides this kind of communication.

## EDM SEARCH PROBLEM

To search the electric dipole moment was proposed to use polarized protons at the so-called "magic" momentum of 0.7 GeV/c in an electric storage ring. For studying beam dynamics in electrostatic rings different computational methods can be used. There should be a lot of calculations for different lattices, different deflectors, initial distributions and so on. Many results are not structured, hard to analyze and a lot of post calculation work has to be done. For example, to calculate lattice with several thousands particles there should be many scripts for each datafile. Lattice generation, initial beam distribution,

code generating, tracking and data analysis should be done with the same scripts to be sure in the results.

To solve this problem we need a tool to automate this process and to be able to make virtual experiments without long preparations.

## SIMULATION PROGRAMS

To study beam dynamics there are a lot of different program codes. There are well-known codes like MAD-X [3], OptiM [4], COSY Infinity [5]. Each program has its own advantages, disadvantages, and it is very hard to start writing code without learning and reading manuals. In our simulations we use COSY Infinity, because it can calculate orbit and spin motion in electrostatic accelerators for billion turns. Unfortunately, it's not a very simple program and it needs to be evaluated before using.

## DATA ANALYSIS

To analyse data received from simulation program one needs a lot of work. Usually it's just a set of raw data (like coordinates in reference coordinate system), and it is impossible to directly compare different simulation programs. Many important things, like finding tunes (betatron oscillation frequencies or spin tunes), chromaticities, stability regions and so on have to be done manually and if you have a lot of lattices it's could be complicated and long procedure. Data analysis can be fully automated, if you have usual things to do for each experiment.

## WORKING CYCLE

To automate all described things an automated system should be able to do the following: for a given particle physicist creates a lattice, system analyzes it in simple linear model and calculates basic accelerators values, like TWISS functions, calculates tunes, chromaticities and so on. After that, if the lattice is good enough, one can start to evaluate lattice in details, for example, track particles through the channel. Experimenter can choose initial distribution (e.g. shaped like a Gaussian function or uniformly distributed in the given interval), initial polarization, choose number of turns, select simplification method. After the tracking has been done, data could be analyzed (trajectories could be plotted, tunes could be found using Fourier analysis). Using this data, lattice can be changed in the desirable way and the whole process could be repeated. Also the system can change some lattice parameters automatically and find what changes in the result. This can be used to optimize channel and to find minimum

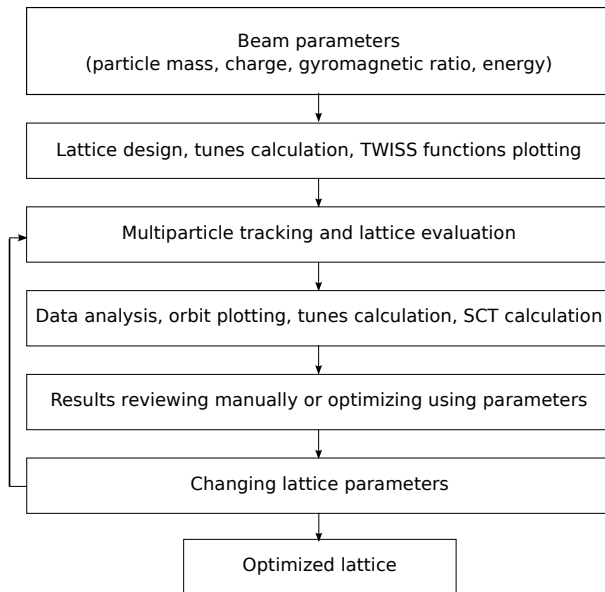


Figure 1: Program's working cycle.

or maximum values. For example, in the EDM project we need an electrostatic lattice with high spin coherence time (SCT) — time while the beam conserves initial horizontal polarization, and this value can be changed using different deflectors [6] or sextupoles [7]. So, in this problem, the system can optimize structure with free parameters, like field in the deflectors, or sextupoles strength, in order to maximize SCT. The whole working cycle is explained on Figure 1.

## SYSTEM PROTOTYPE

The system prototype is made as a web site which allows user access the system using only any modern web browser, on any platform. The system consists of the following components:

- core application;
- web front-end;
- engine servers (with COSY Infinity application);
- task distributor service;
- file distribution service;
- database to store the results;
- filestore server for long-time results storing;
- analysis server.

Figure 2 shows a diagram with all components.

Core application is written on Python and it connects all components into one whole thing. User interacts with the system through a web browser. Web application is written on Django to simplify web development, using default

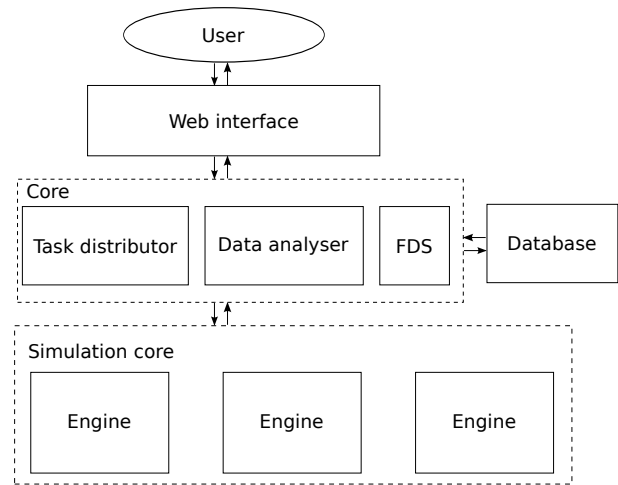


Figure 2: Program's diagram.

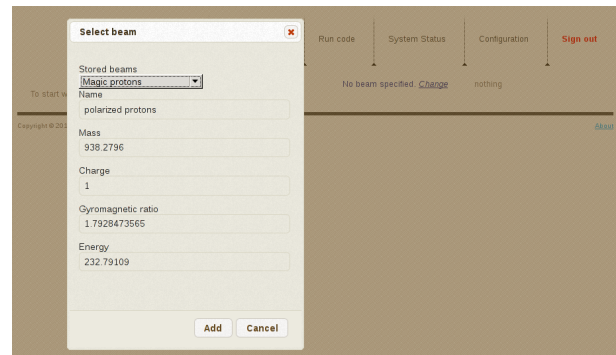


Figure 3: Screenshot with beam selection.

Django user account control and database model. After logging on to the web site user can start simulation process.

On this page user has to select beam (or define a new one) and then to define lattice using COSY Infinity syntax to describe particle physics elements. Beam selection page is shown on Figure 3. The simple lattice looks like

```
DL .1 ; {drift}
MQ .2 .1 .05 ; {quadrupole}
DL .1 ;
MQ .2 -.1 .05 ; {defocussing quadrupole}
DL .1 ;
```

After this step, the system shows plot with TWISS functions and gives some values from linear analysis. Then one can start to evaluate lattice in details.

For tracking particles through the system, one should define initial distribution. Also it is possible to specify free parameters in the lattice, like drift length, quadrupole or sextupole strength, RF-cavity frequency and so on and to change it to see how the lattice behaves.

To perform tracking one needs to have engine servers added. This is just a Unix server with Python interpreter and Fortran compiler installed. To add server one needs to know SSH-credentials and the system will install necessary

scripts and compile COSY Infinity from scratch using Fortran compiler provided. After compiling COSY Infinity for the first time, engine is in working state and can be used to execute COSY Infinity programs. Communication between the core and the workers is done via RPyC protocol over SSH secure channel.

Experimenter needs to specify order of calculations, number of processors to be used and select the engines where calculations will be performed. To start tracking, core initially prepares COSY Infinity source file with all necessary routines, like lattice description, map generation, initial data generation, repetitive tracking and output results to the file. After source file is generated, task distribution service sends it to the engines. When engine finishes simulations, it sends a signal to the core and the core starts to analyze data.

Data analysis process consists of several steps: data fetching, filtering the results, calculating values and plotting. COSY Infinity saves data to file on the machine where it is installed, but the data could be analyzed on another machine. To distribute the data across all machines, file distribution service (FDS) is used. Each file in the system has a unique hash value which is stored in the FDS. Also FDS knows on which server the file is located. When the simulation program finishes the execution, the resulting data file added to the FDS and this could be fetched on any machine when needed.

Data analysis script runs on an arbitrary machine and it knows hash of the data file. Script fetches the data file, performs analysis and sends analyzed data to the core. The core saves the results to the database, which is used to store lattices, file hashes and analyzed data, so when the user wants to analyze lattice which is already in the database, the system will show the results without performing recalculation.

After all these steps, the web site represents the results to the user. On Figure 4 the page with the results is shown.

Also the user can download raw datafiles to do the analysis by hand. After reviewing the results, the experimenter can change free lattice parameters, analyze the system again, or this can be done using optimizer and some analyzed value can be maximized or minimized. The procedure could take a long time, but it can be done on different workers to parallelize calculations. All plots can be saved as raw datafiles or as a PostScript file.

All lattices are stored in the database forever and one can find the needed results, even if the calculation was done long time ago. This can simplify life of experimenter and there is no need to remember all parameters of calculations as they are all stored.

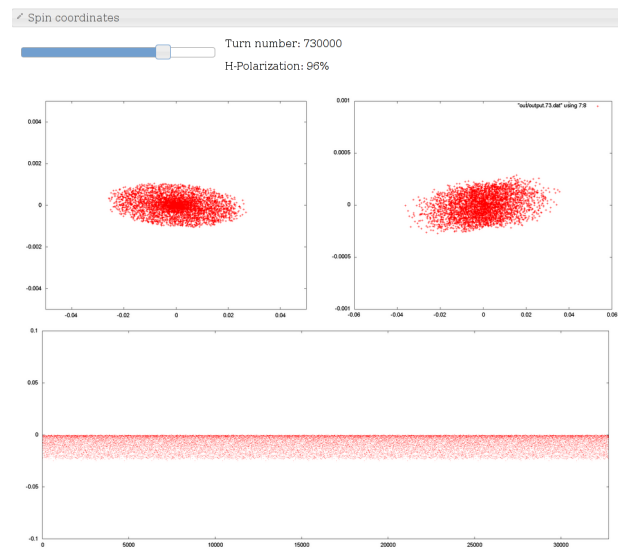


Figure 4: Analyzed data.

## CONCLUSION

The system proposed in this article can significantly simplify carrying out the virtual experiments. The system now is in early development stage and many improvements can be made, like plotting analyzed values depending on the free lattice parameters, code generation for different simulation programs can be added, also version to support multi-particle tracking on the MPI-systems can be developed.

## REFERENCES

- [1] EPICS Home Page, <http://www.aps.anl.gov/epics/>.
- [2] S. Andrianov, Lego-Technology Approach for Beam Line Design, Proc. of EPAC 2002, Paris, France, pp. 1607–1609.
- [3] MAD — Methodical Accelerator Design, <http://mad.web.cern.ch/mad/>.
- [4] OptiM — Computer code for linear and non-linear optics calculations, <http://www-bdnew.fnal.gov/pbar/organizationalchart/lebedev/OptiM/optim.htm>.
- [5] COSY Infinity, [http://www.bt.pa.msu.edu/index\\_cosy.htm](http://www.bt.pa.msu.edu/index_cosy.htm).
- [6] Yu. Senichev, R. Maier, D. Zyuzin, M. Berz, Alternating Spin Aberration Electrostatic Lattice for EDM Ring, Proc. of the 3rd International Particle Accelerator Conference IPAC'12, pp. 1332–1334, New Orleans, LA, USA, 2012.
- [7] Yu. Senichev, R. Maier, D. Zyuzin, N. Kulabukhova, Spin Tune Decoherence Effects in Electro- and Magnetostatic Structures, these proceedings.