

REVIEW OF PROTON LINAC BEAM DYNAMIC SIMULATION CODE

X. Feng, J. Peng, Institute of High Energy Physics (IHEP CSNS), Dongguan, China

Abstract

CSNS-II project design a linac accelerate 40 mA H-beam from 3.8 MeV to 300 MeV, which should not only overcome the spacecharge effect at low energy, but also have high efficiency at high energy. Therefore, lots of simulation studies should be done on a variety of codes. Each of them has its own characteristics. For example, MAD can easily match quadrupole fast while it couldn't do multiparticle calculation. This paper will introduce some common and efficient code which used to design linac and study beam dynamic performance.

INTRODUCTION

CSNS-II will improve current 100 kW target power to 500 kW, which demand linac part provides beam with higher energy and stronger current compared with present beam. We plan to update beam current to 30 mA, and design a superconductive linac (SC linac) to achieve 300 MeV output energy after current normal temperate accelerator. Lots of work had been done, we compared few different SC linac designs and will decide the final plan soon. To simulate such a high-power beam, several codes would be prepared to calculate beam dynamic behavior during linac design. TraceWin and PyORBIT are the most famous and comprehensive function linac beam dynamic codes in the world, and are the major simulation codes in many laborites surround the world. This paper will make an introduction of these two codes, compared their characteristics, and see some other codes which are specializing in a particular area.

HIGH POWER PROTON LINAC

For the last decades, high power proton linac (HPPL) has been an important subject for all accelerator researchers. Not only Spallation neutron source, like CSNS in Fig. 1, but also accelerator driven system (ADS) need high power proton linac as injector. Both are hot research topics and both have many questions to be studied. Simulation is the most important research method at present, which means researchers need to choose the best tool for their work, or develop their own code.



Figure 1: CSNS project aerial image.

TraceWin Code

The TraceWin code is development by The French Alternative Energies and Atomic Energy Commission (CEA) has been one of the most famous linac beam dynamic simulation codes. The code has been refined for 20 years and is still evolving. As show in Fig. 2 [1], the simple navigating interface and good interaction could lead the whole system to a better user experience, detailed manual make it easy to learn and operate this code. Built-in examples show users the basic use of the software, like adjust input beam, track RFQ and DTL cavities, and apply chopper and solenoid in linac lattice. This is a measure to make the TraceWin code more novice friendly. Now TraceWin code can track both simple envelope model and a few common multiparticle models with 2D or 3D space-charge through whole linac lattice from source to target. Users can set the beam to electron, proton, H- beam and many kinds of heavy ion. Transverse and longitudinal matching calculation makes users easily find best beam and elements parameters of their lattice. Another advantage of the TraceWin code is that it has a wealth of error analysis tools. You can apply both statistical error and dynamic error in a study. In addition, powerful GUI and various help tools helps this code show the results clearly.

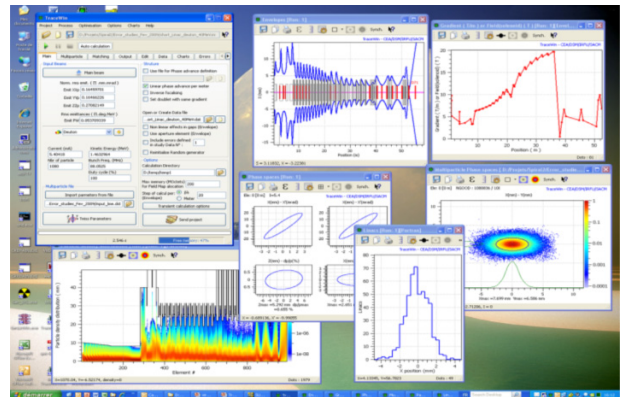


Figure 2: TraceWin software example.

TraceWin is a powerful, easy and clear, comprehensive simulate code. After more than ten years of commercial operation, TraceWin has been built into an efficient, easy-to-use, mature commodity, now it still keeps the product updated and upgraded.

PyORBIT Code

Based on origin ORBIT code, ORNL developed a new simulation code PyORBIT. As descendant of ORBIT code, PyORBIT has the two language structure: Python as driving scripting shell and C++ as underlying code. The different with TraceWin is that PyORBIT is open source, which makes this code more flexible and has more possibilities, and everyone can download the source code from github [2]. The basic structure of PyORBIT is show in

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2021). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

Fig. 3, beginners can start with examples folder to learn how to operate this code. The two most important folders are src and py, where C++ classes and wrappers and Python classes are stored. Origin ORBIT code can only track transport line and ring, RF gap model has been added into PyORBIT which make it possible to track beam through DTL, CCL and other RF cavities. There are currently 5 kinds of space-charge model available, including uniform ellipsoid, 2-D, 2.5-D, and 3D space-charge. Users can also add their own model if necessary. Although Python includes many drawing tools, visualization software is not yet available in PyORBIT. And linac simulate hasn't match function yet. Users can extract beam data easily from calculation. Due to Python shell, add some new function into PyORBIT code would be convenient. As an increasingly mature software, PyORBIT will attract more and more researchers to use it for simulation work.

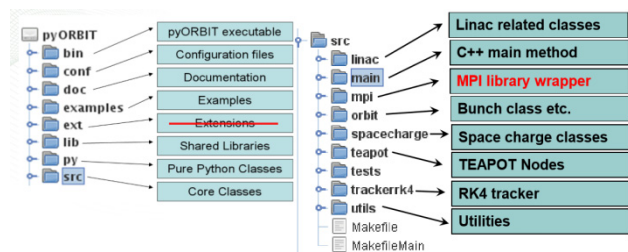


Figure 3: PyORBIT code structure.

Other Codes

There are many other codes can used in dynamic simulation, but most of them only good at one or few situations. For example, MAD has been used to calculate and match transport line for decades, it can track envelope and do match calculation based on envelope. And Parmila can PARMILA uses a Particle-In-Cell approach to computing the beam space charge [3]. In a real study, more than one code will be used, and We need to figure out what works best for the design.

CONCLUSION

In general, TraceWin is more friendly for users, provide efficient calculate power and simple result presentation ability. And PyORBIT is more flexible, maybe getting started is more difficult, but have more possibilities. Other code is unique in terms of features and functionality, Table 1 shows comparison of above codes. They are all very useful simulate codes and we can choose according to personal preference.

Table 1: Comparison of Several Types of Code

Code	Envelope	Multiparticle	Match
TraceWin	√	√	√
PyORBIT	√	√	×
MAD	√	×	√
Parmila	×	√	√

ACKNOWLEDGEMENT

The author thanks Mr. Peng and Mr. Wang for their guidance. At the same time thanks Institute of High Energy Physics, and University of Chinese Academy of Sciences.

REFERENCES

- [1] D. Uriot and N. Pichoff, "Status of TraceWin Code", in *Proc. 6th Int. Particle Accelerator Conf. (IPAC'15)*, Richmond, VA, USA, May 2015, pp. 92-94.
doi:10.18429/JACoW-IPAC2015-MOPWA008
- [2] PyORBIT-Collaboration, <https://github.com/pyorbit-collaboration/py-orbit>
- [3] R. Ryne, "Beam Dynamics Simulations Using a Parallel Version of PARMILA", in *Proc. 18th Int. Linear Accelerator Conf. (LINAC'96)*, Geneva, Switzerland, Aug. 1996, paper MOP72, pp. 234-236.