# Linac Beam Dynamics Simulations with PyORBIT


***Andrei Shishlo***

***SNS Accelerator Physics Group***

***ORNL, USA***


***ICAP12, Rostock, Germany***

***August 20, 2012***



OAK RIDGE National Laboratory

# Outline

1. Why another tracking code for linac?
2. What is PyORBIT
3. PyORBIT Overview and the Linac Part
4. Models for RF Gap and Space Charge
5. Benchmark with Parmila
6. Benchmark with XAL Online Model
7. Plans for the Future
8. Conclusions

# Why Another Linac Code?

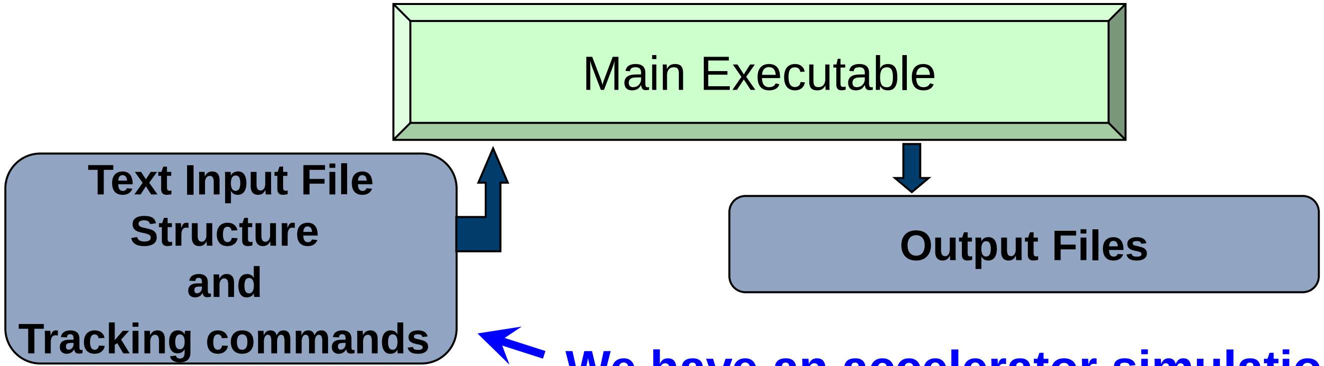
- **There are a lot of PIC linac codes available: Parmila, Impact, Track, TraceWin etc.**
- **They all have one fundamental drawback – SNS project does not have control over any of these codes.**
- **"To have control" – ability**
  - **to take a look at the source code**
  - **to make small reversible changes for computational experiments**
  - **to add new physics if we need it.**
  - **to add new diagnostics**
- **It is not universal – some people do not want to have this type of control**

**We started a linac code inside PyORBIT.**

# PyORBIT at a Glance

- **PyORBIT is descendant of ORBIT code**
- **ORBIT is a ring and transport line code**
- **PyORBIT & ORBIT have the two language structure: driving scripting shell and C++ underneath**
- **ORBIT's Super Code shell was replaced by Python**
- **Recent flavor of PyORBIT was started in 2006**
- **PyORBIT real applications until today:**
  - **laser stripping (ORNL, USA)**
  - **beam dynamics in lattices with highly non-linear elements (Tech-X Corp., Boulder, USA)**
- **Open source: means everybody can do anything, and it is open for future collaborators**

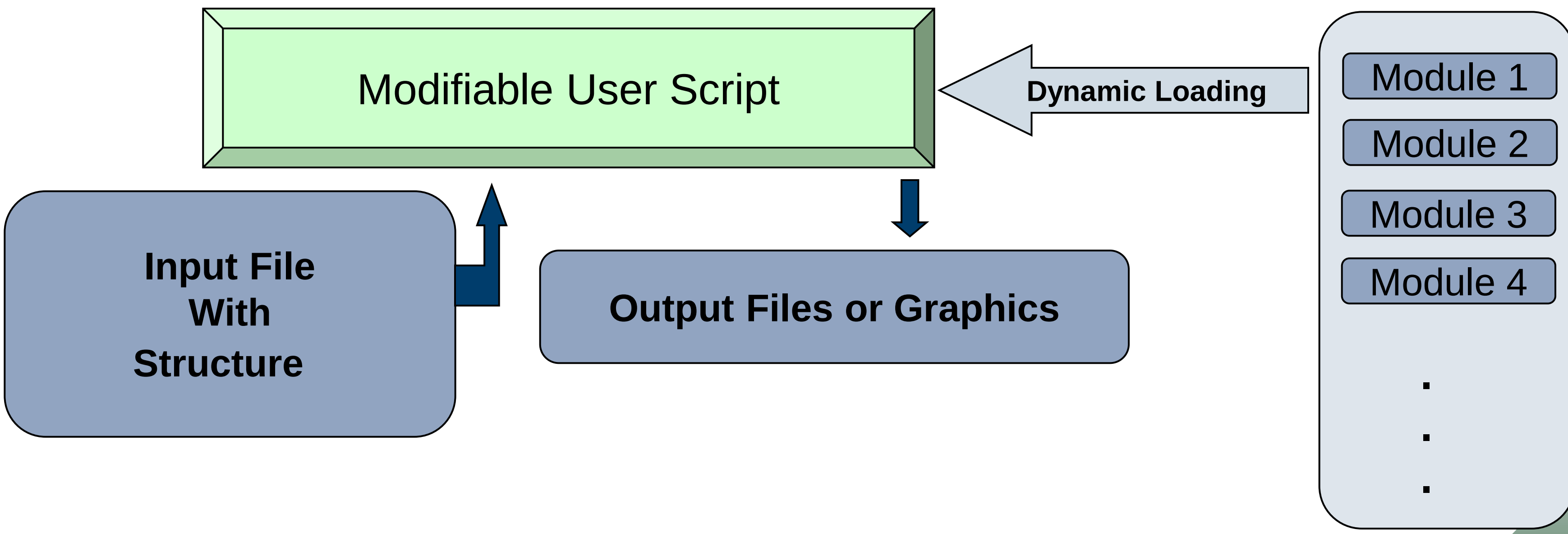
# PyORBIT : Scripting Shell Approach




**Old style:**
**MAD-like structure**
**Old languages**

**We have an accelerator simulation language**

**UAL, ORBIT, PyORBIT etc. : an existing programming language is extended**



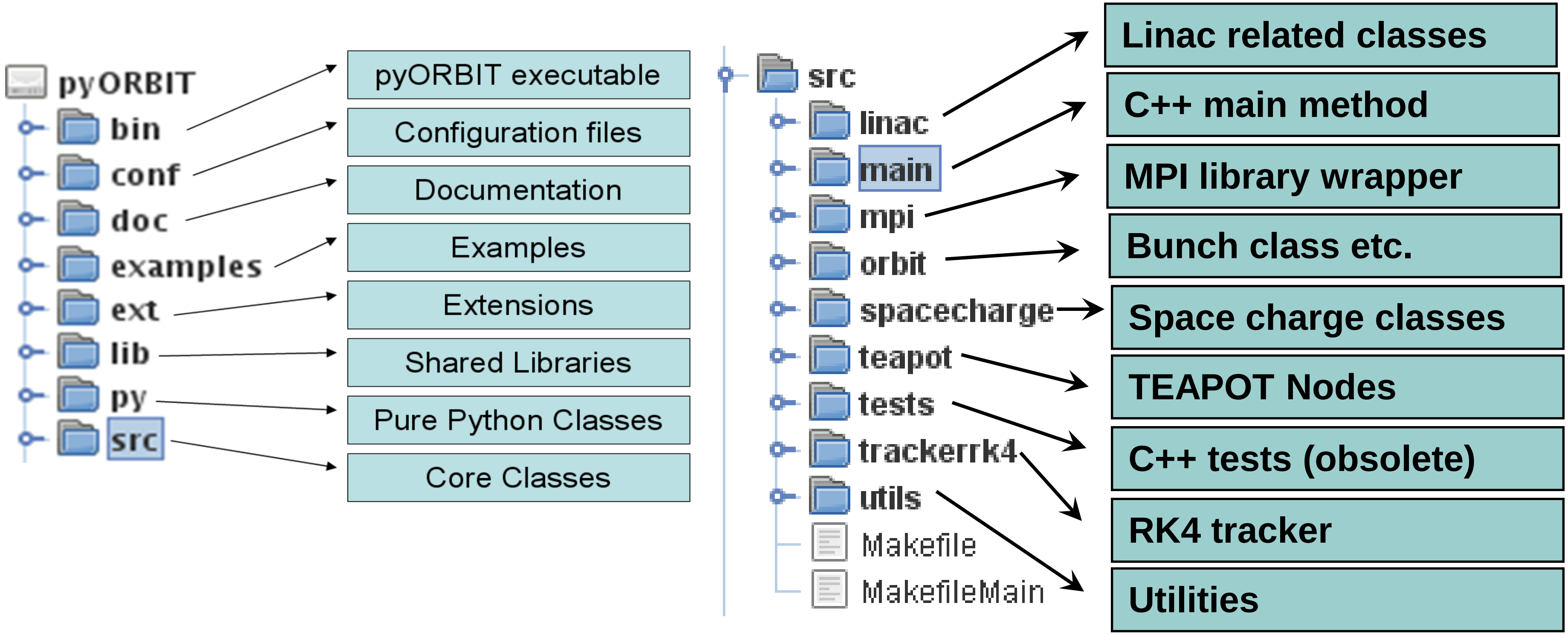

**Unfortunately, we have to use two language approach: Python + C++**

# Problems with Scripting Shell Approach

- **Much more complicated compared with one language approach**
- **It needs more skilled developers**
- **Steeper learning curve**
- **Debugging is more difficult (two levels)**
- **Documenting is more difficult (two levels)**

**But! The flexibility, power, and speedup of the development process overcome the problems.**

# PyORBIT Structure




- **The “scr” directory includes C++ classes and wrappers**
- **The dir “Extensions” includes mutually unrelated projects. After compilation the shared libraries are placed into the “lib” dir.**
- **The “py” directory includes pure Python classes and functions.**

# Additions to PyORBIT for Linac Model

- **New implementation of the abstract accelerator lattice package. The “sequences” and RF Cavities were added.**
- **RF Gap Model**
- **Two new 3D Space Charge classes**

C++ and wrappers

- **The linac structure parser for linac lattices**
- **The linac lattice factory**

# XML SNS Linac Structure Example

```xml
<?xml version = '1.0' encoding = 'UTF-8'?>
<sns>
    <MEBT bpmFrequency="8.05E8" length="3.633" name="MEBT" rfFrequency="4.025E8">
        <accElement length="0.061" name="MEBT_Mag:QH01" pos="0.128" type="QUAD">
            <parameters effLength="0.061" field="34.636"/>
        </accElement>
        <accElement length="0.0" name="MEBT_Mag:DCV01" pos="0.128" type="DCV">
            <parameters effLength="0.061"/>
        </accElement>
        <accElement length="0.0" name="MEBT_Mag:DCH01" pos="0.128" type="DCH">
            <parameters effLength="0.061"/>
        </accElement>
        <accElement length="0.0" name="MEBT_Diag:BPM01" pos="0.128" type="MARKER">
            <parameters/>
        </accElement>
        <accElement length="0.061" name="MEBT_Mag:QV02" pos="0.273" type="QUAD">
            <parameters effLength="0.061" field="-36.813"/>
        </accElement>
        <accElement length="0.061" name="MEBT_Mag:QH03" pos="0.418" type="QUAD">
            <parameters effLength="0.061" field="28.3266"/>
        </accElement>
        <accElement length="0.0" name="MEBT_RF:Bnch01:Rg01" pos="0.528" type="RFGAP">
            <parameters E0TL="0.075" amp="1.3" firstPhase="90.0" gapLength="0.015" gapOffset="0.0" modePhase="0.0" parentCvaity="MEBT_RF:Bnch01"/>
        </accElement>
        <accElement length="0.061" name="MEBT_Mag:QV04" pos="0.638" type="QUAD">
            <parameters effLength="0.061" field="-16.12"/>
        </accElement>
        <accElement length="0.0" name="MEBT_Mag:DCH04" pos="0.638" type="DCH">
            <parameters effLength="0.061"/>
        </accElement>
```

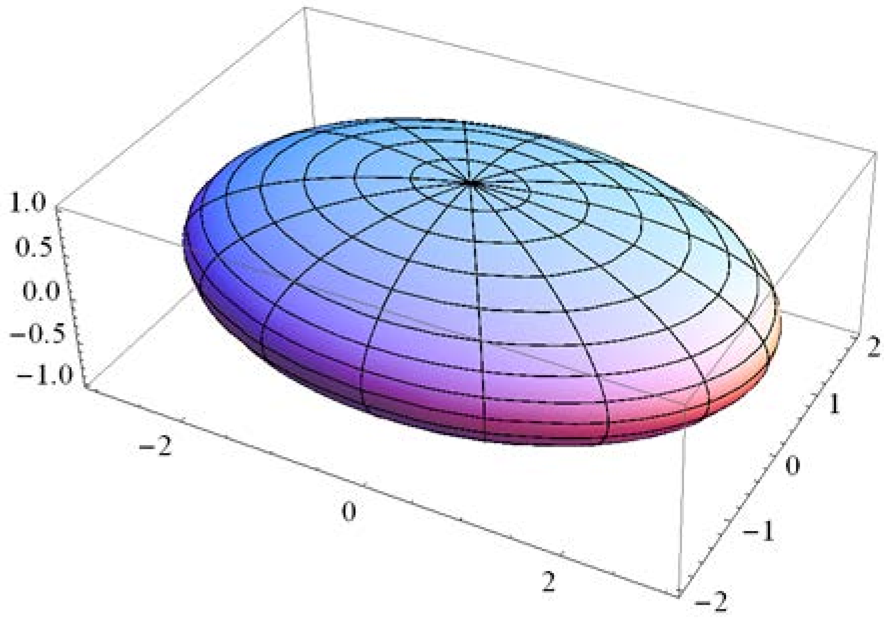
# RF Gap Model

$$W_{out} = W_{in} + E_0 TL \cdot \cos(\varphi_{RF} + \varphi)$$

$$r'_{out} = \frac{(\gamma\beta)_{in}}{(\gamma\beta)_{out}} \cdot r' - \frac{q \cdot E_0 TL}{2 \cdot m \cdot c^2 \cdot \gamma^3 \cdot \beta^3} \cdot \frac{2\pi}{\lambda} \sin(\varphi_{RF} + \varphi) \cdot r$$

**Simplified model: acceleration and transverse focusing**

# Uniformly Charged Ellipse Solver




$$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1$$

$$\phi(\mathbf{x}) \equiv \int_{V_0} \frac{\rho}{|\mathbf{x}-\mathbf{x}'|}\, dV(\mathbf{x}')$$

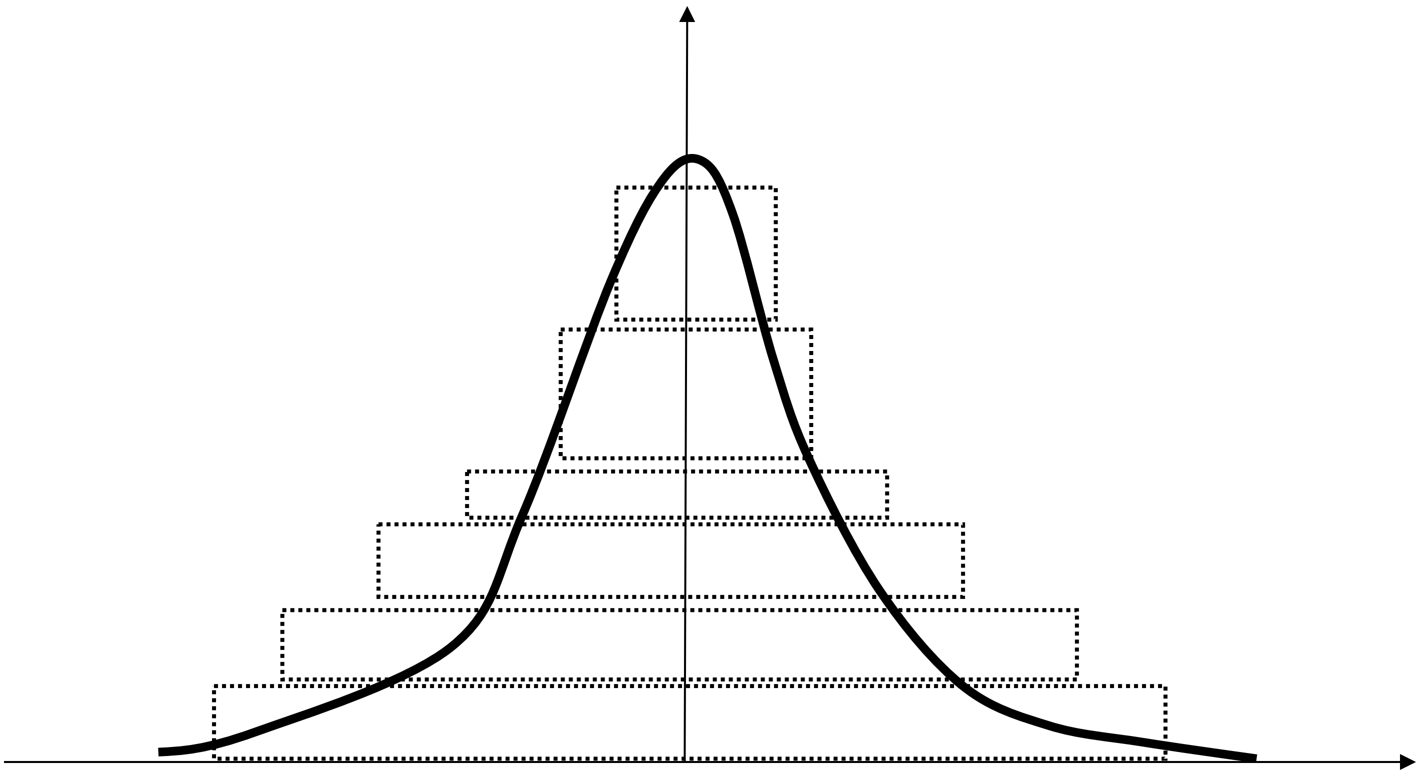
$$\phi(x,y,z)=\pi\, abc\, \rho \int_0^\infty \left[1-\frac{x^2}{a^2+s}-\frac{y^2}{b^2+s}-\frac{z^2}{c^2+s}\right]\frac{ds}{\sqrt{\varphi(s)}}$$ inside

$$\phi(x,y,z)=\pi\, abc\, \rho \int_\lambda^\infty \left[1-\frac{x^2}{a^2+s}-\frac{y^2}{b^2+s}-\frac{z^2}{c^2+s}\right]\frac{ds}{\sqrt{\varphi(s)}}$$ outside

$$\varphi(s) \equiv (a^2+s)(b^2+s)(c^2+s) \qquad f(s) \equiv \frac{x^2}{a^2+s}+\frac{y^2}{b^2+s}+\frac{z^2}{c^2+s}-1$$

where $\lambda$ is the greatest root of the equation $f(s)=0$

# Set of Ellipsoids



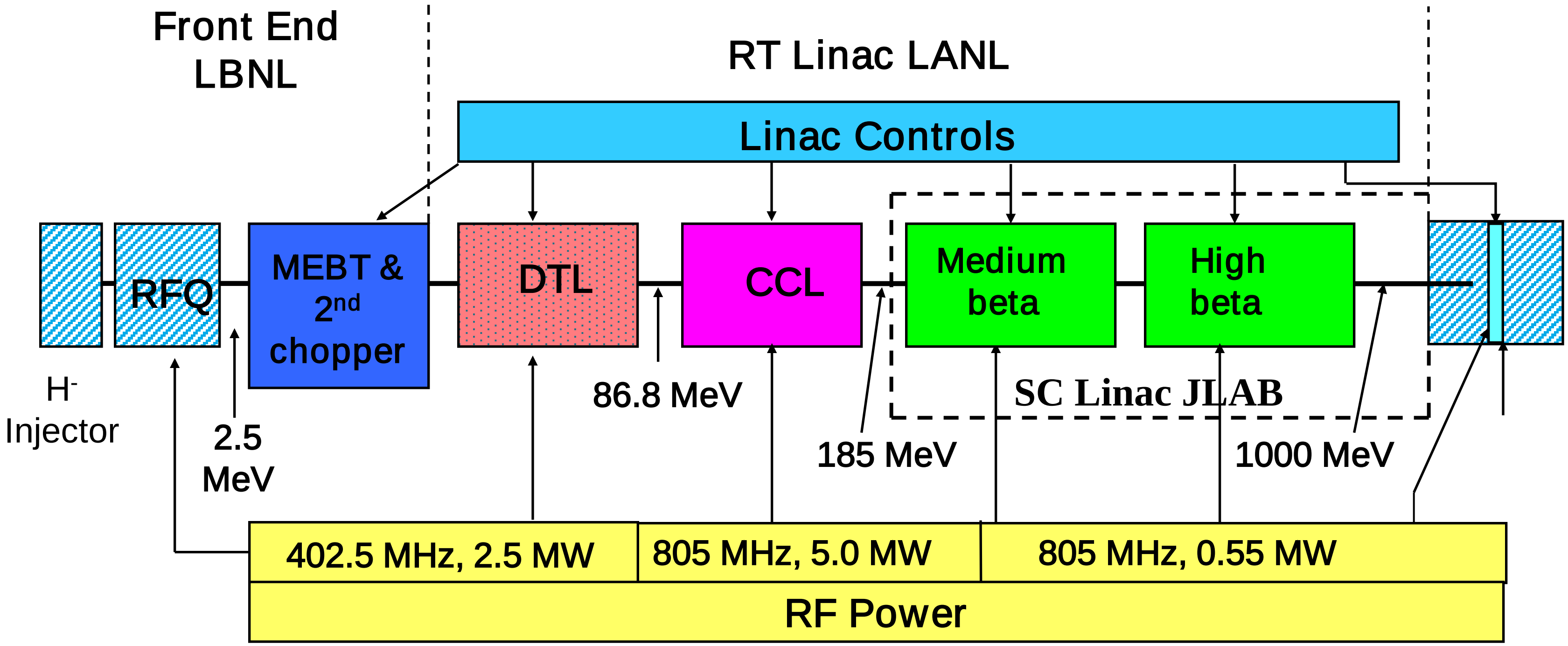
pyORBIT Space Charge Solver can use arbitrary number of ellipsoids

# 3D FFT Poisson Solver

- **Open boundary**
- **Usual method doubling the size of the region is used**
- **FFTW**
- **Grid is not distributed in the case of  a parallel calculations**
- **Green function scalability is used**

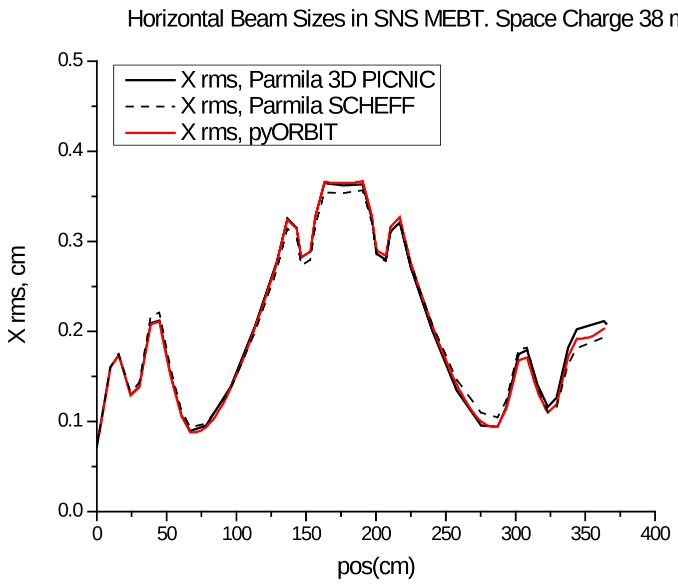
$$\phi_0(\vec{r}) = \int \frac{\rho(\vec{r} - \vec{r}\,') \cdot d\vec{r}\,'}{\left|\vec{r} - \vec{r}\,'\right|}$$

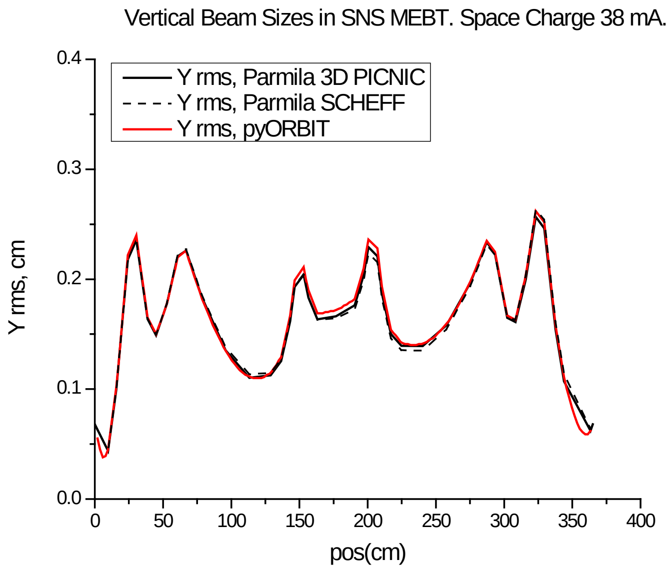
# The Benchmark Case – SNS linac

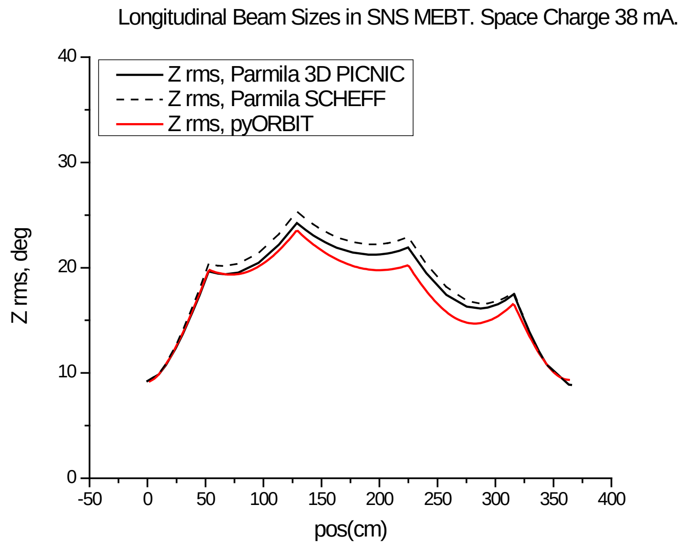


# PyORBIT – Parmila Benchmark - MEBT

SNS MEBT 38 mA. PARMILA vs. pyORBIT



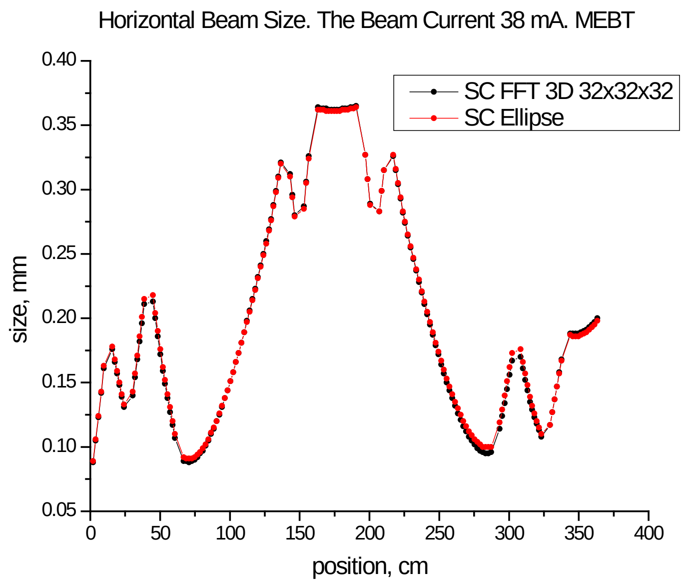



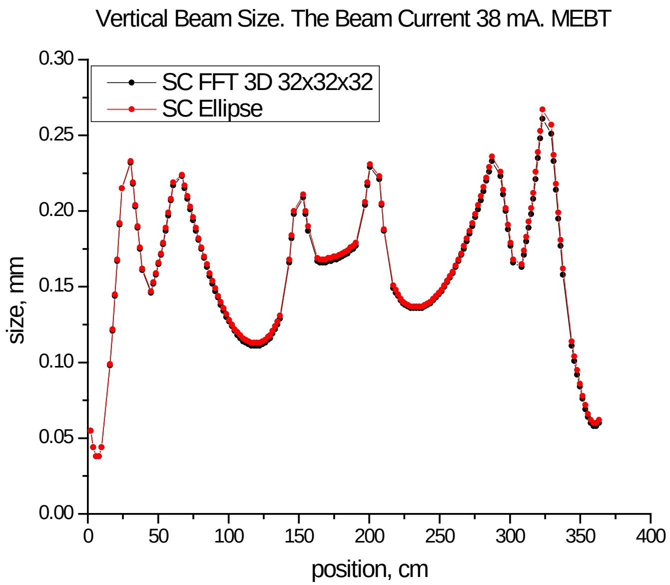



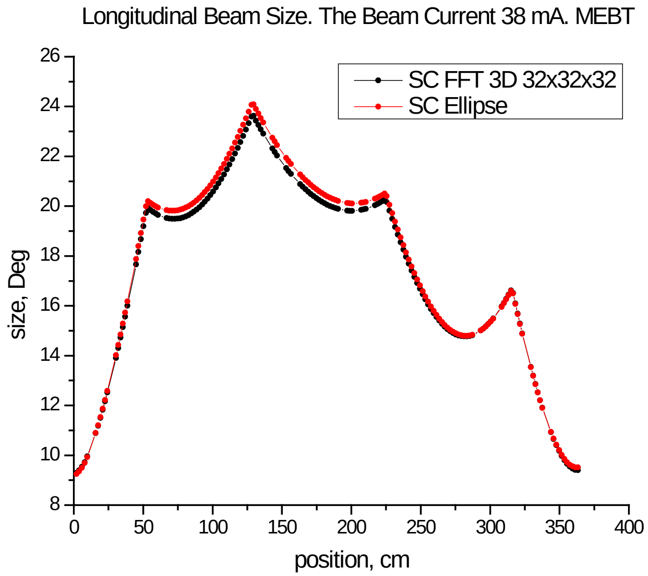

20,000 macro-particles
Water Bag 3D
Bunchers RF is on, $90^0$ phases

Timing:
Parmila SHEFF – about 4 sec
PARMILA 3D PICNIC – 8 sec
pyORBIT 1 Ellipsoid – 1.6 sec

# Space Charge: 3D FFT & Ellipsoid - MEBT

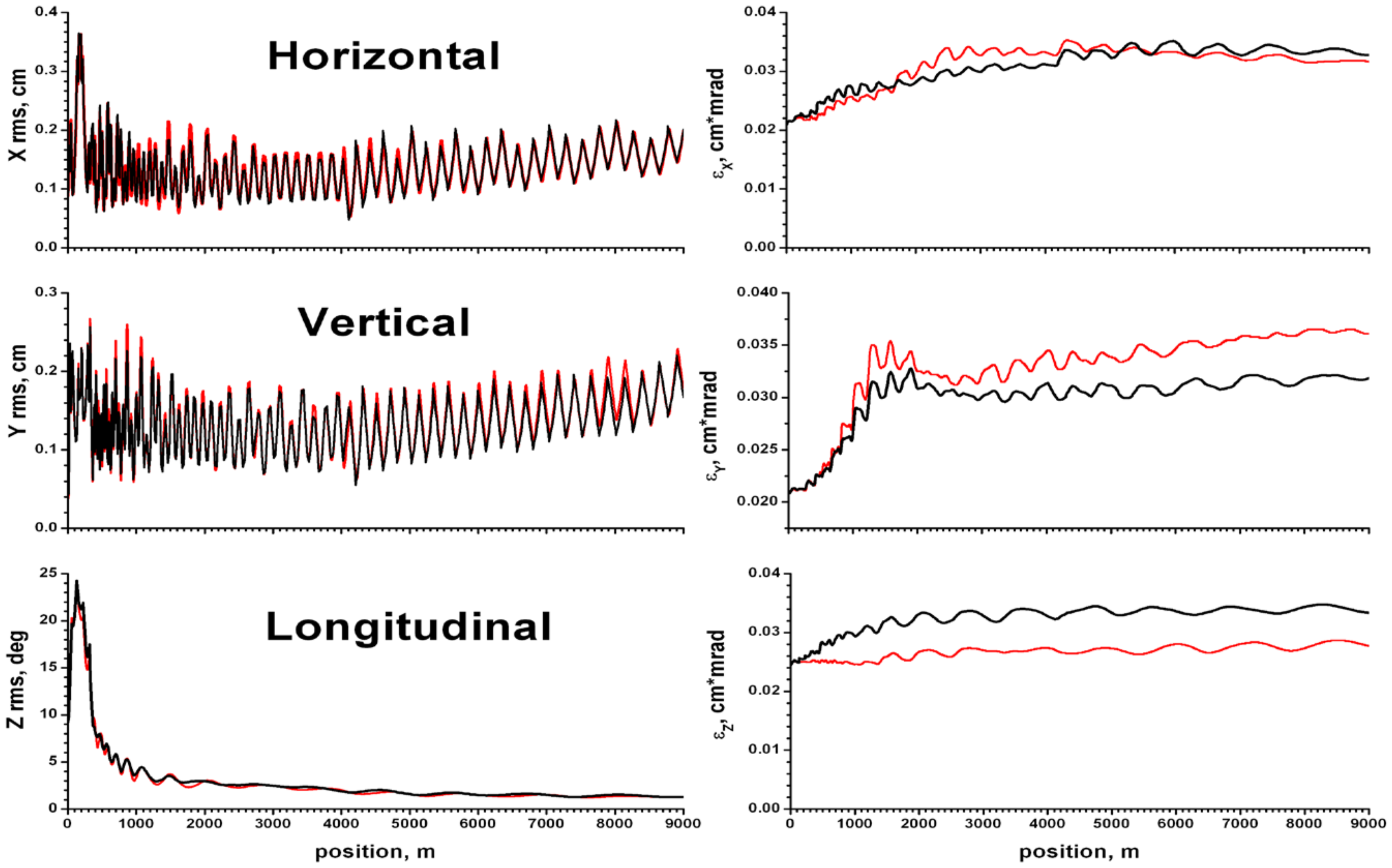


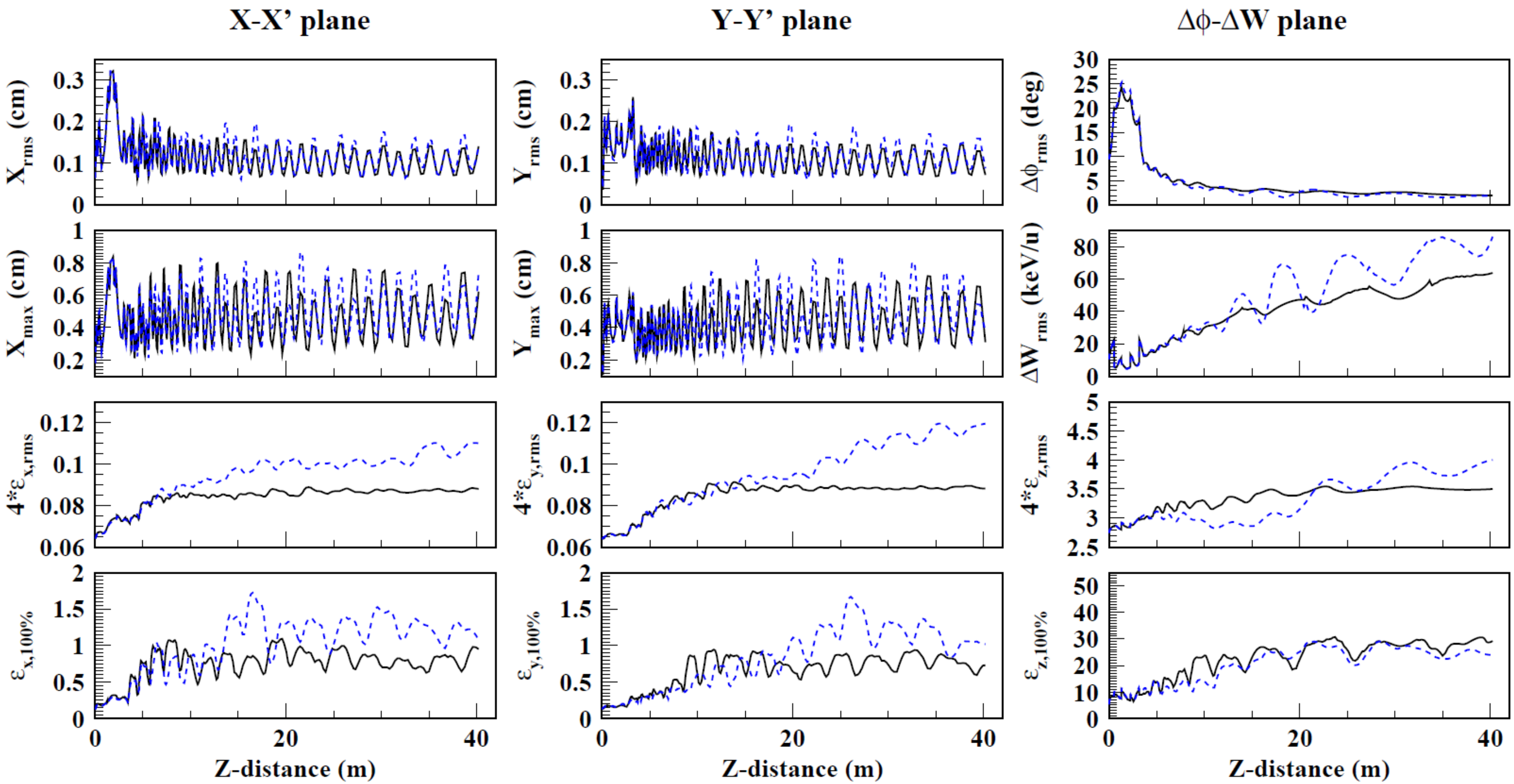

Water Bag 3D, 38 mA
2,000 macro-particles for Ellipse SC
20,000 macro-particles for 3D FFT
32 x 32 x 32 grids
1.3 sec for Ellipse SC
4.6 sec for FFT 3D

# PyORBIT–Parmila Benchmark – MEBT-DTL-CCL




MEBT-DTL-CCL = 90 meters, PyORBIT (red)    Parmila (black)

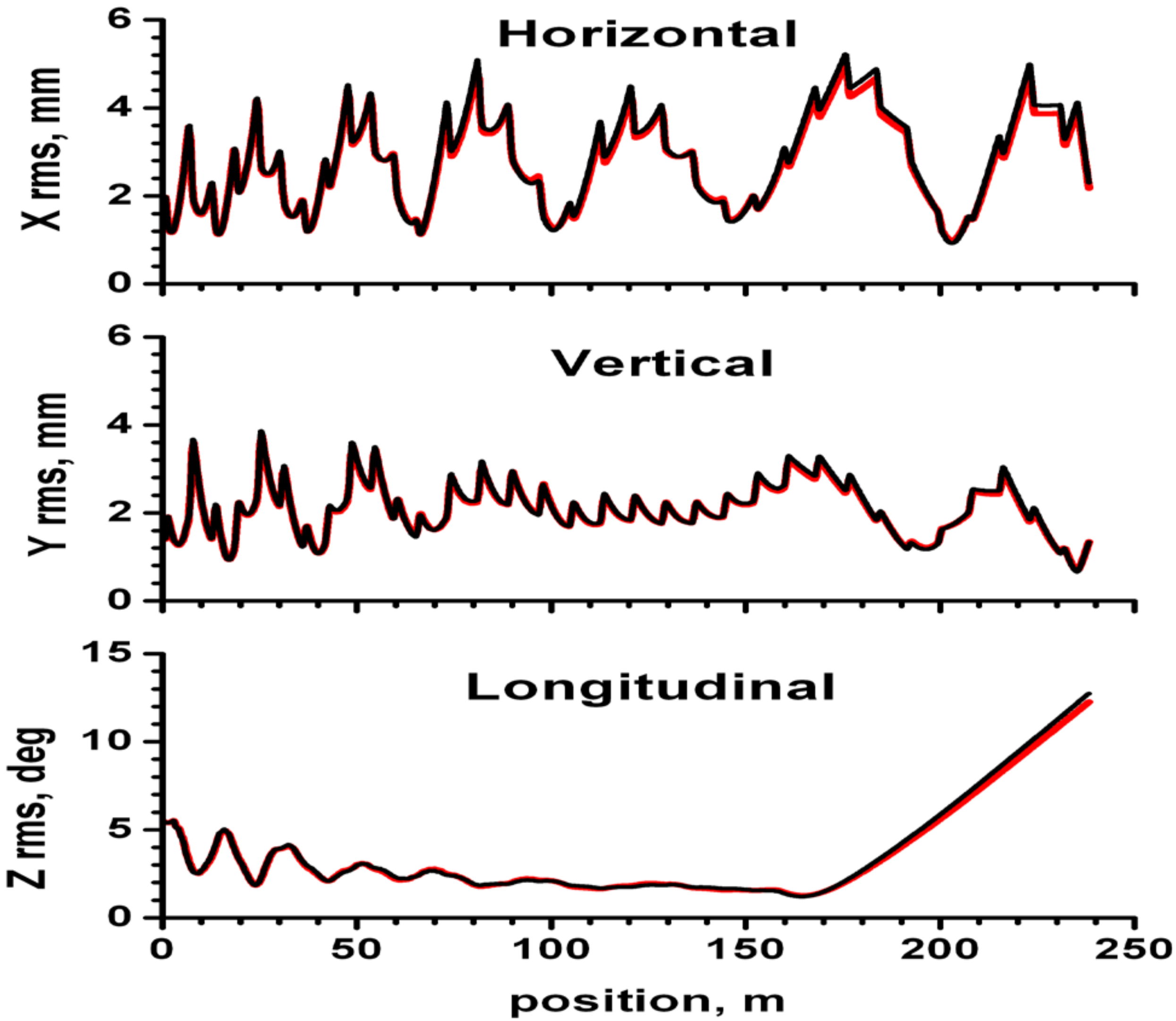
# Track-Parmila Benchmark – LINAC'06




**DTL only - 40 meters, Blue – Track code**

B. Mustapha, "First Track Simulation of the SNS Linac," LINAC'06, Knoxville, TN 2006, TUP076, p. 432 (2006); http://www.JACoW.org

# SCL: PyORBIT – XAL Online Model Benchmark




**XAL Online Model is an envelop tracking code**

# Future Development

- **Realistic RF Gap models**
- **More diagnostics**
- **Faster and parallel scalable 3D SC solvers**
- **Collimation**

# PyORBIT Developers

- **Xiyin Zhang (CSNS) : CSNS ring lattice testing**
- **Sarah Cousineau (ORNL) : collimator module rewriting from the original ORBIT**
- **Jeff Holmes (ORNL) : time dependent lattices**
- **Stephen Webb (TechX): non-linear optics**
- **Timofey Gorlov (ORNL) : laser stripping module further development**
- **Andrei Shishlo (ORNL) : linac development**

# Summary

- **The initial framework for linac simulations is ready**
- **PyORBIT linac part was successfully benchmarked against Parmila and XAL Online Model for the SNS linac**
- **More realistic models for RF are needed**

**http://code.google.com/p/py-orbit/**

# Thanks for your attention!