Chapter I.6

PyHEADTAIL tutorials

Benoit Salvant
CERN, Geneva, Switzerland

The macro-particle tracking code PyHEADTAIL simulates the dynamics of charged particle beams in the presence of collective effects: direct space charge, beam coupling impedance and electron clouds. For the JUAS course PyHEADTAIL has been used in tutorials in small groups to allow students to simulate longitudinal beam dynamics without collective effects, as a direct illustration of the Longitudinal Beam Dynamics course.

I.6.1 Goals of the PyHEADTAIL tutorials

There were multiple objectives for the PyHEADTAIL tutorials in the JUAS course:

- Simulate several case-studies related to the course and give an alternative angle to longitudinal beam dynamics from a practical point of view,
- Allow students to play with beam and machine parameters so that they gain operational experience of the impact of these parameters on beam dynamics,
- Show students concrete examples of common issues in critical phases for longitudinal beam dynamics in circular accelerators (e.g. injection errors from one machine to the next in the accelerator chain, transition crossing, bunch splitting),
- Give experience of a state of the art macro-particle tracking code,
- Give the possibility and the recipe to download all tools so that students can use it in the future should they need to.

I.6.2 Installation of the packages

There are three options to run the PyHEADTAIL examples of the course: install locally if the operating system permits it, install a virtual machine to recover the common installation procedure, and - since 2023 - use the installation on SWAN (swan.cern.ch [1]).

I.6.2.1 Local installation

The macroparticle code PyHEADTAIL was developed at CERN to simulate beam dynamics in particle accelerators with collective effects [2].

This chapter should be cited as: PyHEADTAIL simulation code, B. Salvant, DOI: 10.23730/CYRSP-2024-003.313, in: Proceedings of the Joint Universities Accelerator School (JUAS): Courses and exercises, E. Métral (ed.),
CERN Yellow Reports: School Proceedings, CERN-2024-003, DOI: 10.23730/CYRSP-2024-003, p. 313.

PyHEADTAIL is currently compatible with Python v3.6 or later. For using PyHEADTAIL without modifying the source code, it is recommended to install the latest version via PyPI through the command "pip install PyHEADTAIL".

From a Linux operating system, one would need to install the Anaconda package at https://www.anaconda.com/distribution/#download-section, as well as gcc (command: "sudo apt-get install gcc") and gfortran (command: "sudo apt-get install gfortran"). The Anaconda distribution includes many useful packages such as numpy (array manipulation), scipy (scientific functions such as Fourier transforms) and matplolib (data plotting).

This local installation is the most direct, the most versatile and would use the full power of the local machine, but one can expect incompatibilities of the notebooks examples with the Python environment of the student, especially with plotting libraries.

I.6.2.2 Use of a virtual machine

A virtual machine was chosen for common installation in the computer room to allow students to be on a unique controlled environment, so that time is not wasted in technical incompatibilities during the tutorial.

The virtual machine installation can also be done from any operating system, and is also possible on the laptops of the students. The main drawback is the size required on disk by the virtual machine (of the order of 10 Go to install all the required software).

The installation procedure is:

- Install the software to create a virtual environment, (https://www.virtualbox.org/wiki/Downloads),
- Create a new virtual machine with Ubuntu 64-bit and start it,
- Download (https://www.ubuntu.com/download/desktop) and select as start-up disk for the virtual machine to install it,
- Download and install Anaconda for Linux (https://www.anaconda.com/download),
- Install gcc and fortran from the shell ("sudo apt-get install gcc" and "sudo apt-get install gfortran").

I.6.2.3 Use SWAN

Since the ESI computer room was dismantled in 2022, the virtual box solution was not ideal for the JUAS tutorial as one could not assume that the laptops of students could have 10 to 20 Go of free space.

With the help of David Amorim (CERN), the JUAS Jupyter notebooks were modified to be run into the SWAN environment so that PyHEADTAIL could be run from an internet browser with adequate internet connection. This however requires that all students have a CERN computer account, and the main drawback is that students lose the possibility of running on SWAN if they lose their connection to CERN after JUAS.

In fact, SWAN (Service for Web-based ANalysis) is a CERN service that allows users to perform interactive data analysis in the cloud. It is built upon the widely-used Jupyter notebooks, which allows users to write - and run - their data analyses using only a web browser. Moreover, by connecting to

SWAN, users have immediate access to the CERN storage, software and computing resources they need to do their analyses [1].

I.6.3 PyHEADTAIL examples

Tutorial 1: Getting started with a beam in the center of the bucket

The first tutorial guides the student through the code set up (installation of PyHEADTAIL on SWAN, importation of libraries and PyHEADTAIL functions, set up of plotting and animation functions), as well as the definition of relevant machine parameters (circumference, kinetic energy, optics functions, transition energy, RF voltage, harmonic number, magnetic field acceleration rate, bending radius) and simulation parameters (number of macro-particles, initial bunch length, simulated number of turn).

The notebook then goes on to generate the distribution by matching it to the bucket, to start the tracking of the macro-particles, and to display the evolution of the longitudinal distribution along the simulation of a stationary situation at injection energy.

The tutorial is based on the CERN PS parameters and proposes the student to realise the meaning of the synchrotron period and synchrotron tune by seeing how many machine turns are needed to perform a revolution in the synchrotron phase space (as shown on Fig. I.6.1). The student can also observe the direction of rotation of the macroparticles in phase space, and see how phase space and the direction of rotation changes when the classical units are used (Φ;ΔE p or z;$\Delta p/p$).

Finally, the student is asked to change the energy to PS extraction energy and to observe the changes that this creates in the longitudinal phase space (height of the separatrix as shown in Fig. I.6.2, direction and speed of rotation of macroparticles).

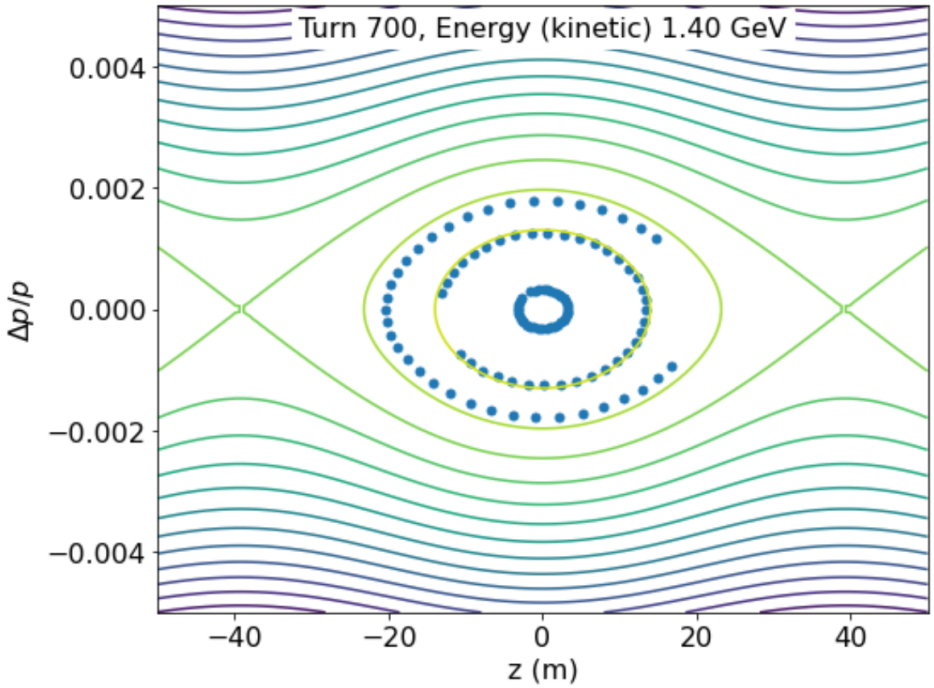

Fig. I.6.1: Trajectories of 3 macroparticles simulated by PyHEADTAIL for an example case of PS at injection energy over 700 machine turns in the longitudinal phase space (in units of distance z and momentum $\Delta p/p$) represented by blue dots. These trajectories are overlayed with contours obtained analytically. The students can observe the separatrix, the closed trajectories inside the bucket and the open-ended trajectories outside the bucket. They can also observe that the particle that undergoes small amplitude oscillations in the longitudinal phase space completes one full turn in longitudinal phase space faster than particles closer to the separatrix, which illustrates the non-linearity of the restoring force. From this plot, the students can also estimate that it takes about 700 machine turns for small amplitude particles to complete one synchrotron period, and that the synchrotron tune is therefore $Q_s \approx 1/700 \approx 1.4 \times 10^{-3}$.

Tutorial 2: Simulating a phase mismatch

In this second tutorial, the student is asked to simulate a phase mismatch by changing abruptly the longitudinal position coherently by the same amount for all macroparticles (longitudinal kick, see Fig. I.6.3). This offsets the whole beam in the longitudinal phase space and generates filamentation as particles closer to the separatrix travel slower in phase space compared to particles closer to the center of the bucket (the non-linear term is getting larger for large amplitude oscillations), as shown in Fig. I.6.4.

The student can also see that particles that are moved outside of the bucket by the longitudinal kick are travelling around the buckets, but are not lost in a stationary situation.

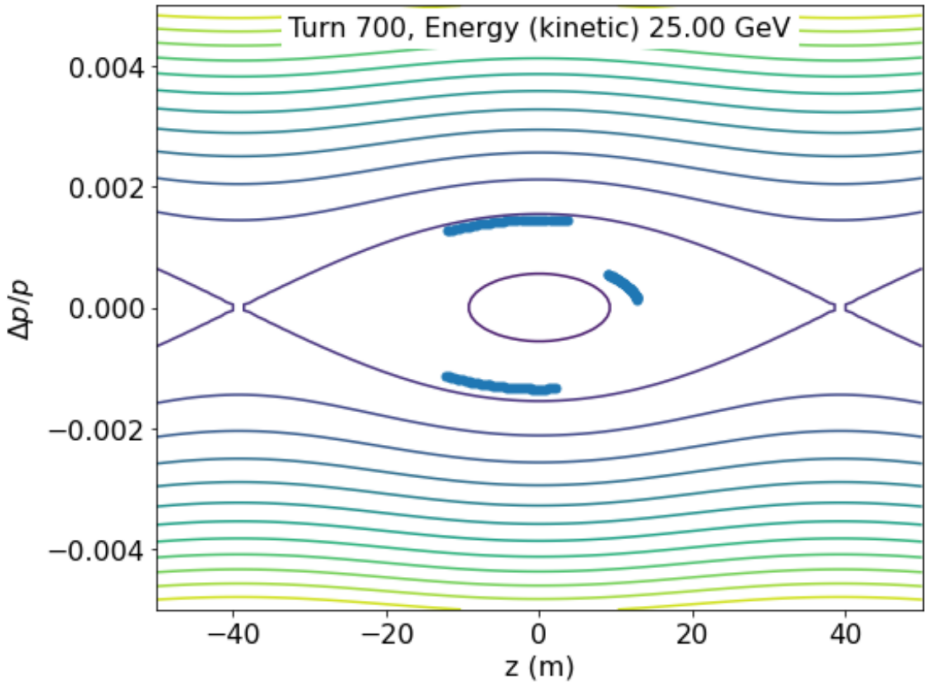

Fig. I.6.2: Position of 3 macroparticles in the longitudinal phase space (in units of distance z and momentum $\Delta p/p$) for an example case of PS at extraction energy represented by blue dots. Comparing with the situation at injection in the previous figure, the students can observe that the height of the separatrix is lower and that the synchrotron tune is also lower.

Phase errors or phase jitters therefore lead to filamentation and eventually emittance growth after several hundreds of thousands of turns. Students can also observe that the longitudinal motion is slow compared to the machine revolution timescale.

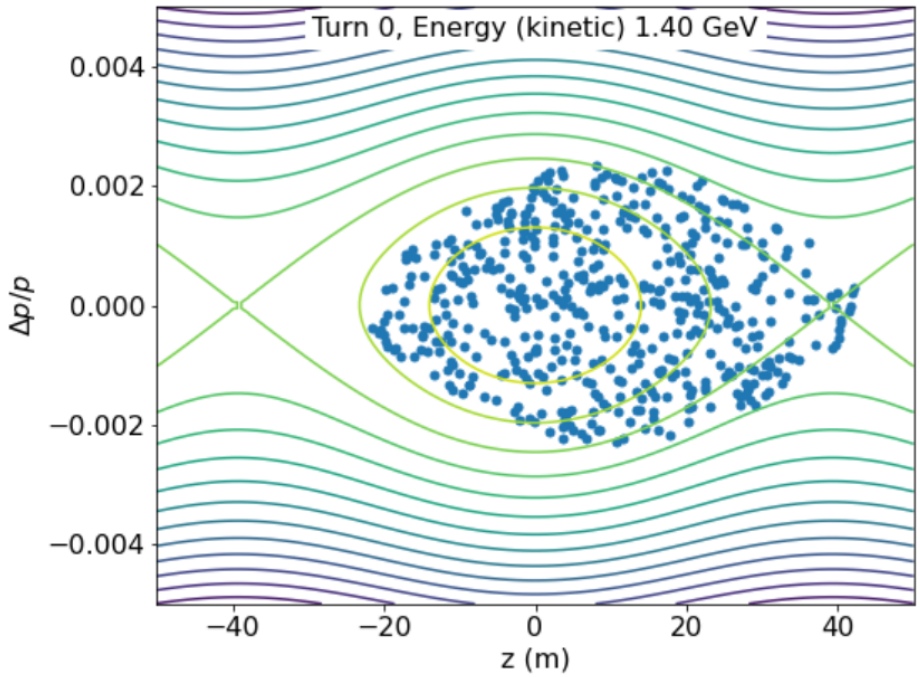

Fig. I.6.3: Position of 500 macroparticles for an example case of PS at injection energy after a phase mismatch was introduced in the longitudinal phase space (in units of distance z and momentum $\Delta p/p$) represented by blue dots. One can see that several particles now lie outside of the bucket and in the neighbouring bucket.

Tutorial 3: Simulating a voltage mismatch

In the third tutorial, the student is asked to simulate a voltage mismatch by changing abruptly the RF voltage. This changes the height of the bucket in the longitudinal phase space and generates filamentation and eventually emittance growth, as in the case of phase mismatch.

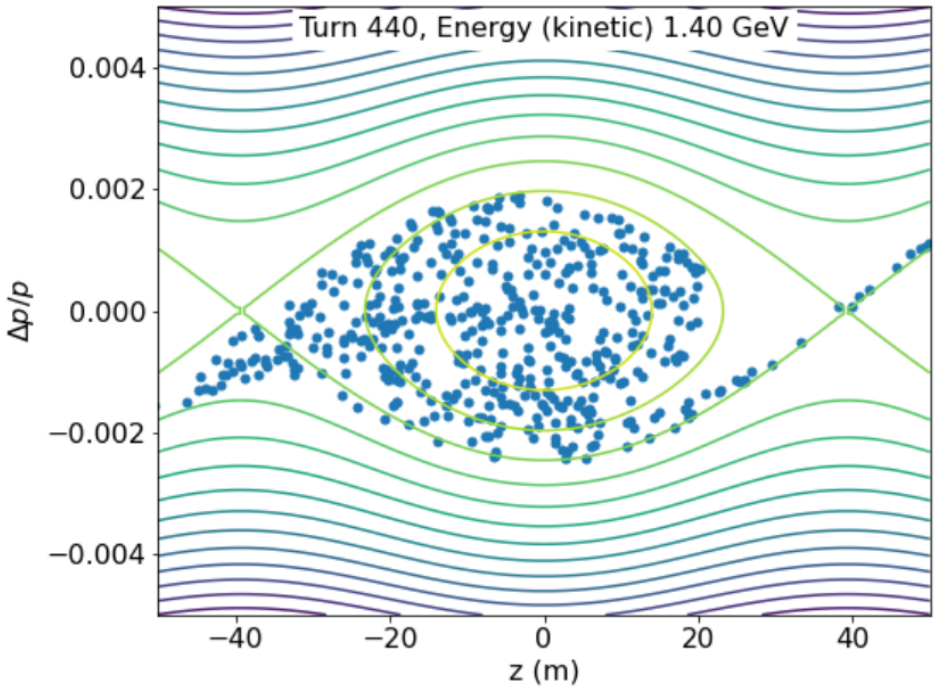

Fig. I.6.4: Position of 500 macroparticles represented by blue dots in the longitudinal phase space (in units of distance z and momentum $\Delta p/p$) for an example case of PS at injection energy 440 turns after a phase mismatch was introduced. The student can observe how the phase mismatch can lead to capture particles in the wrong bucket, and how particles outside the buckets are uncaptured by the RF system and wander around the buckets without necessarily being lost.

The student can also observe that an abrupt voltage increase by a large amount triggers bunch rotation and can be used to decrease bunch length in view of extraction if phased correctly (as it is done for LHC beams in the PS and SPS).

Tutorial 4: Simulating acceleration

In the fourth tutorial, the student tries to accelerate the bunch from injection energy.

First he should realise that there is a minimum RF voltage required to obtain a non-void stable bucket area during acceleration.

Then, he will observe the major changes that occur to the bucket during acceleration (stable phase is no longer 0 or π, bucket shape is deformed and stable area is reduced, as seen in Fig. I.6.5).

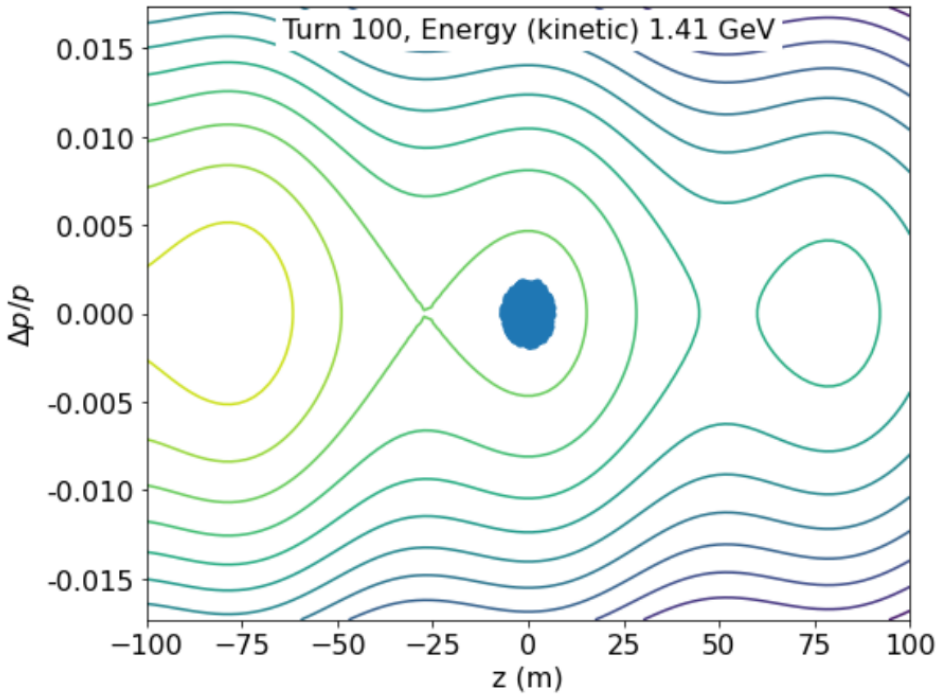

Fig. I.6.5: Position of 500 macroparticles represented by blue dots in the longitudinal phase space (in units of distance z and momentum $\Delta p/p$) for an example case of PS ramping up from injection energy. The student can observe how the shape of the bucket is distorted and acceptance is reduced.

The simulation is then left running over tens of thousands of turns to reach transition energy. The student can appreciate the changes to longitudinal dynamics just before transition (bucket height increases significantly leading to a decreased bunch length as shown in Fig. I.6.6, the bunch is almost frozen inside the bucket).

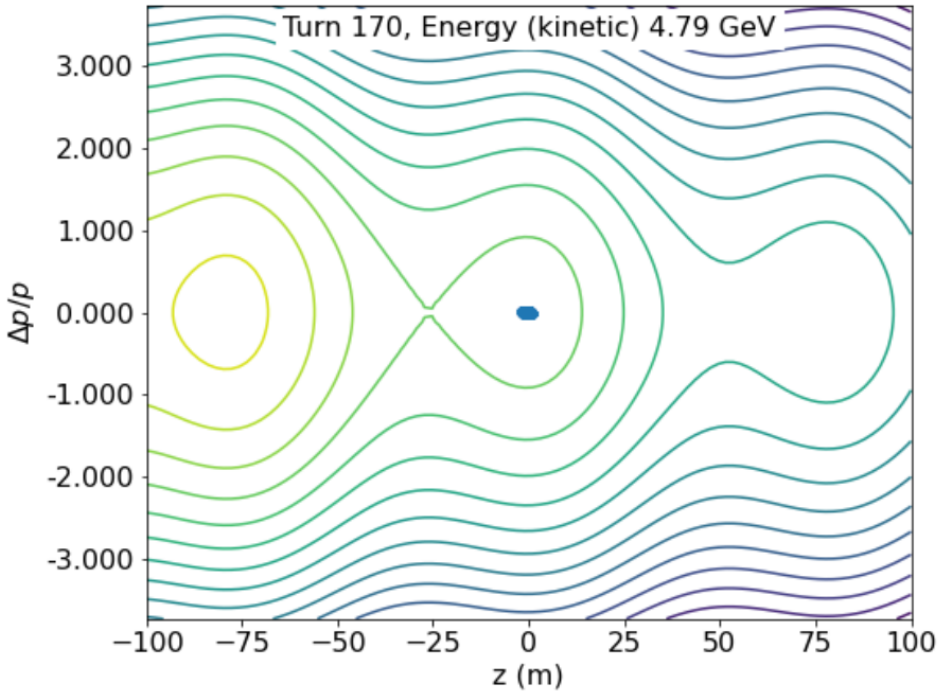

Fig. I.6.6: Position of 500 macroparticles represented by blue dots in the longitudinal phase space (in units of distance z and momentum $\Delta p/p$) for an example case of PS ramping up just before transition energy. The student can observe with the vertical scale change how the vertical size of the bucket blows up just before transition (30170 turns after ramp start), leading to a reduction in bunch length.

The simulation then goes beyond transition energy and the student realises that all the beam is lost if nothing is done as shown in Fig. I.6.7.

He also can observe the process of how particles are lost when they are outside of the bucket, as their energy quickly diverges from the energy of the synchronous particle, which means that they would be lost in high dispersion sections in the accelerator or in low aperture sections.

The student then relaunches the simulations and is given the opportunity to pause it just after transition, in order to switch from the unstable phase to the stable phase. He then sees that no particle is lost from the bucket. He can also see that there is an allowed range of time to change the phase around transition as longitudinal motion is slow and almost frozen around transition.

Tutorial 5: Simulating bunch splitting

In this last tutorial, the student can see the impact of using 2 RF systems simultaneously, and how ramping up the higher harmonic system and ramping down the fundamental RF voltage slowly allows the bunch to adapt to the shape of the new buckets. Ramping up the voltage too fast leads to losses and

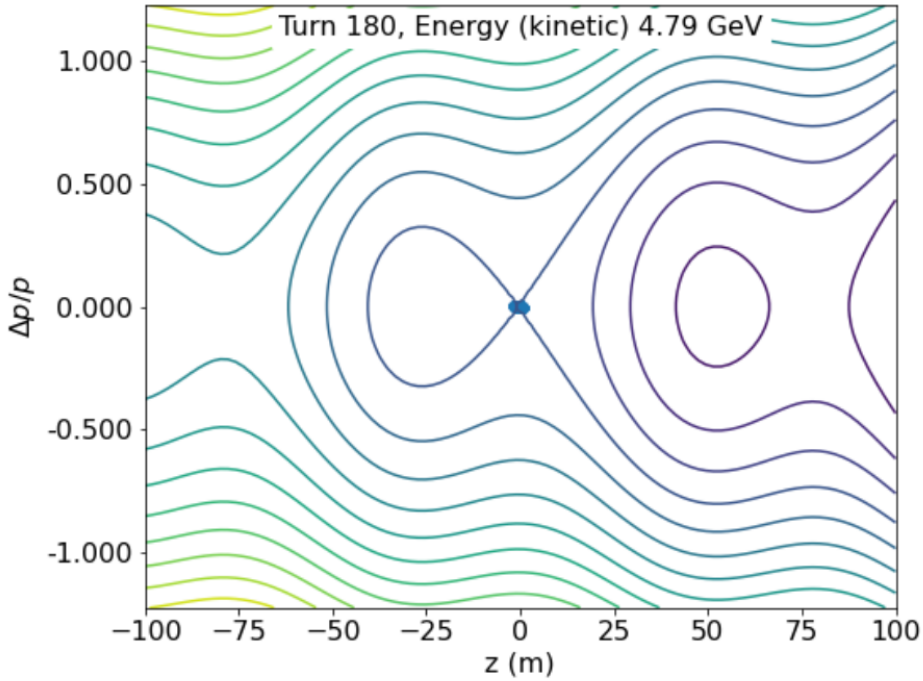

Fig. I.6.7: Position of 500 macroparticles represented by blue dots in the longitudinal phase space (in units of distance z and momentum $\Delta p/p$) for an example case of PS ramping up just after transition energy if no phase change is performed. The student can observe that all the particles are suddenly on the unstable phase when crossing transition, and that one needs to adapt the phase when crossing transition.

large emittance growth, even though it was one operational method used at CERN to produce doublet beams in the SPS.

References

[1] The Swan Service: Service for Web based ANalysis, last accessed 4 December 2023, https://swan.web.cern.ch/swan/.

[2] PyHEADTAIL github repository, https://github.com/PyCOMPLETE/PyHEADTAIL.