LhARA linear optics documentation

A.N. Other

1 Introduction

Introduction to the documentation!

2 Coordinate systems

2.1 Laboratory coordinate system

The origin of the LhARA coordinate system, the "laboratory coordinate system" or "laboratory reference frame", is at the position of the laser focus at the position of the laser-target interaction. The z axis is horizontal and points along the nominal capture axis, pointing in the downstream direction, i.e. away from the target. The y axis is vertical, pointing vertically upwards, and the x axis completes a right-handed coordinate system.

In the following, phase-space coordinates and vector and scalar quantities referred to the laboratory coordinate system will be written without a suffix. Unit vectors along the x, y and z axes are $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ respectively.

2.2 Reference particle local coordinate system

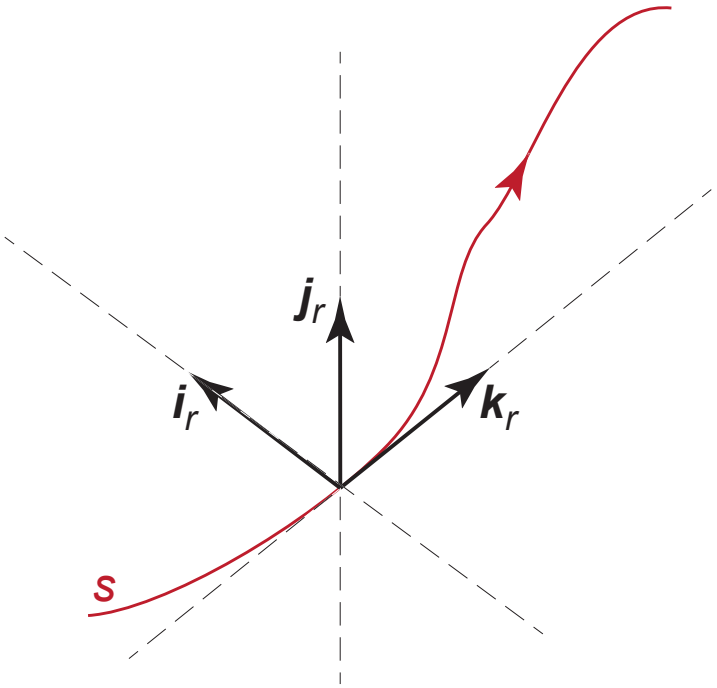

Figure 1: Reference particle local coordinate system

References

Introduction

This document summarises the steps needed to set-up and run LhARA_Beamline linear optics simulation of the LhARA beamine. A summary of the tasks that LhARA_Beamline software suite performs will be documented in due course. LhARA_Beamline has been developed in python; python 3 is assumed.

Getting the code

LhARA_Beamline is maintained using the GitHub version-control system. The latest release can be downloaded from the . . . should we moce to the LhARA repository ... its git, but, bespoke to CCAP/LhARA.

Dependencies and required packages

LhARA_Beamline requires the following packages:

- Python modules: `scipy`, `matplotlib`, `pandas`, and `iminuit`;
- CERN programme library: `pyroot` (which may be installed using the standard `root` installers, see the documentation at https://root.cern/install/).

It may be convenient to run LhARA_Beamline in a "virtual environment". To set this up, after updating your python installation to python 3.9.2, and installing root, execute the following commands:

1. `python3 -m venv --system-site-packages venv`
   - This creates the director `venv` that contains files related to the virtual environment.
2. `source venv/bin/activate`
3. `python -m pip install pandas scipy matplotlib iminuit`

To exit from the virtual environment, execute the command `deactivate`.
The command `source venv/bin/activate` places you back into the virtual environment.

Unpacking the code, directories, and running the tests

After downloading the package from GitHub, or cloning the repositiry, you will find a "`README.md`" file which provides some orientation and instructions to run the code. In particular, a `bash` script "`startup.bash`" is provided which:

- Sets the "`LhARAOpticsPATH`" environment variable so that the files that hold constants etc. required by the code can be located; and
- Adds "`01-Code`" (see below) to the PYTHONPATH. The scripts in "02-Tests" (see below) may then be run with the command "python 02-Tests/< filename >.py".

Below the top directory, the directory structure in which the code is presented is:

`01-Code`: contains the python implementation as a series of modules. Each module contains a single class or a related set of methods.

`02-Tests`: contains self-contained test scripts that run the various methods and simulation packages defined in the code directory.

`11-Parameters`: contains the parameter set used in `02-Tests/RunSimulation.py` to generate muon decays in the production straight.

The instruction in the `README.md` file should be followed to set up and run the code.

Running the code

The file in 02-Tests/RunSimulation.py - will run the code.

The file **RunSimulation.py** contains:

- The definition of csv input file to control the running of the Simulation; and
- The call to the Simulation class with; the number of events to generate; the central energy to generate; and the filenames.