

# LhARA linear optics documentation

A.N. Other

## 1 Introduction

Introduction to the documentation!

## 2 Coordinate systems

### 2.1 Laboratory coordinate system

- 5 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 points vertically upwards, and the  $x$  axis completes a right-handed orthogonal coordinate system.
- 10 Unit vectors along the  $x$ ,  $y$  and  $z$  axes are  $\mathbf{i}$ ,  $\mathbf{j}$  and  $\mathbf{k}$  respectively. The position of the reference particle, its momentum and energy are described as functions of the distance it has travelled from the origin of coordinates to its current position. The distance travelled is defined to be  $s$ , making the position,  $\mathbf{r}_r$ , momentum,  $\mathbf{p}_r$ , and energy,  $E_r$ , of the reference particle at position  $s$ :

$$\begin{aligned} \mathbf{r}_r &= \mathbf{r}_r(s); \\ \mathbf{p}_r &= \mathbf{p}_r(s); \text{ and} \\ E_r &= E_r(s). \end{aligned} \tag{1}$$

The time,  $t_r$ , at which the reference particle is at  $s$  is also a function of  $s$ :

$$t_r = t(s) = \frac{c}{s} \frac{p_r}{E_r}; \tag{2}$$

- 15 where  $p_r = |\mathbf{p}_r|$  and  $c$  is the speed of light.

### 2.2 Reference particle local coordinate system

A coordinate system defined relative to the position of the reference particle, the “reference particle local coordinate” (RPLC) system, may be defined using the direction in which the particle is travelling. The position of the particle defines the origin of the RPLC system, see figure 1.

- 20 The tangent to the reference particle trajectory at  $s$  defines the  $z_{\text{RPLC}}$  axis with unit vector  $\mathbf{k}_{\text{RPLC}}$ . In the laboratory frame, the presence of local electric or magnetic fields may cause the reference particle’s trajectory to change. In the neighbourhood of the particle, the curved trajectory may be described in terms of an arc of a circle. The  $x_{\text{RPLC}}$  axis (with unit vector  $\mathbf{i}_{\text{RPLC}}$ ) is then taken to be in the direction pointing towards the centre of the circle. The third coordinate axis,  $y_{\text{RPLC}}$ , is defined to complete the right-handed orthogonal coordinate system; the unit vector along the  $y_{\text{RPLC}}$  axis being given by  $\mathbf{j}_{\text{RPLC}} = \mathbf{k}_{\text{RPLC}} \times \mathbf{i}_{\text{RPLC}}$ .

The trajectory of the reference particle will be a straight line as it traverses a drift space or when the particle’s energy is increased (or decreased) through an electric field applied parallel to its direction of motion. In such cases the RPLC axes are taken to coincide with either the laboratory coordinate system or the RPLC system defined at the exit of the beam-line element that preceded the drift space or accelerating element.

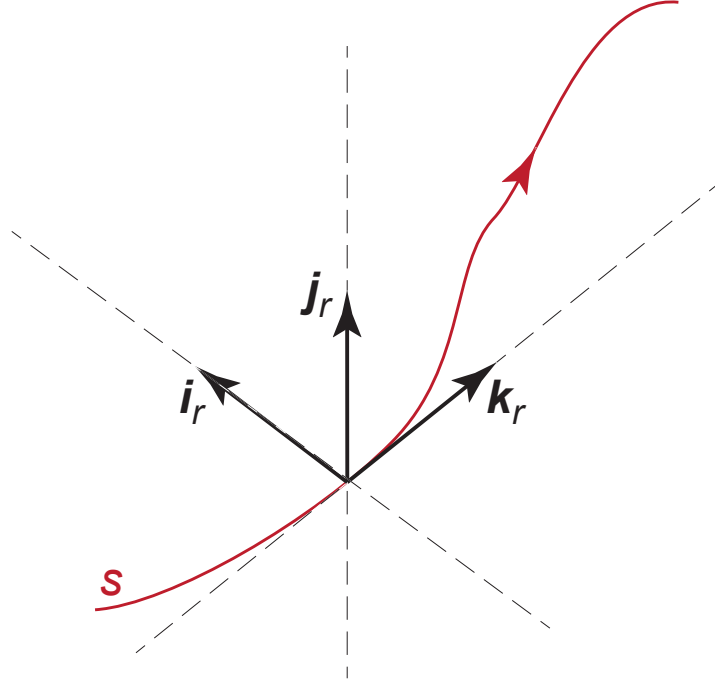


Figure 1: Reference particle local coordinate system. The trajectory of the reference particle is shown as the red line. The distance the reference particle has travelled, measured from the origin of coordinates in the laboratory frame, is labelled  $s$ . The origin of the RPLCs is coincident with the position of the reference particle. The directions of unit vectors along each of three righthanded, orthogonal coordinate axes are shown as black arrows labelled  $\mathbf{i}_r$ ,  $\mathbf{j}_r$ , and  $\mathbf{k}_r$ .

## 30 2.3 Transforming to and from reference particle local coordinates to laboratory co-ordinates

In RPLC, the trajectory of the reference particle,  $\mathbf{R}_r$ , is:

$$\mathbf{R}_r(s) = \mathbf{0}. \quad (3)$$

The position of a test particle in the RPLC frame ( $\mathbf{R}_t$ ) is described with reference to the position of the reference particle. In the laboratory frame, the position of the test particle is:

$$\mathbf{r}_t(s) = \mathbf{r}_r(s) + \delta\mathbf{r}(s); \quad (4)$$

where:

$$\delta\mathbf{r}(s) = \underline{\underline{R}}(s)\mathbf{R}_t(s); \text{ and} \quad (5)$$

$\underline{\underline{R}}(s)$  is a rotation matrix that takes the RPLCs at  $s$  to the laboratory frame coordinates.

In the laboratory frame, the unit vectors  $\mathbf{i}_{\text{RPLC}}$ ,  $\mathbf{j}_{\text{RPLC}}$  and  $\mathbf{k}_{\text{RPLC}}$  are given by:

$$\begin{aligned} \mathbf{i}_{\text{RPLC}} &= \begin{pmatrix} i_{rx} \\ i_{ry} \\ i_{rz} \end{pmatrix}; \\ \mathbf{j}_{\text{RPLC}} &= \begin{pmatrix} j_{rx} \\ j_{ry} \\ j_{rz} \end{pmatrix}; \text{ and} \\ \mathbf{k}_{\text{RPLC}} &= \begin{pmatrix} k_{rx} \\ k_{ry} \\ k_{rz} \end{pmatrix}. \end{aligned} \quad (6)$$

The rotation matrix,  $\underline{\underline{R}}$ , may now be written:

$$\underline{\underline{R}}(s) = \begin{bmatrix} i_{\text{RPLC}x} & j_{\text{RPLC}x} & k_{\text{RPLC}x} \\ i_{\text{RPLC}y} & j_{\text{RPLC}y} & k_{\text{RPLC}y} \\ i_{\text{RPLC}z} & j_{\text{RPLC}z} & k_{\text{RPLC}z} \end{bmatrix}. \quad (7)$$

## 3 Phase space and trace space

35 Para 1:

- Quantum effects required to describe, e.g., development of polarisation in electron storage rings;
- Description of beam dynamics largely done using classical, Hamiltonian, mechanics;
- In classical mechanics, the equations of motion are solved to give the evolution of the position, momentum, and energy of a particle as a function of time;
- 40 • Check Goldstein, time as parameter.

Para 2:

- Relativistic mechanics exploits four-vector position ( $\underline{R} = (\mathbf{r}, ct)$ ) and four-vector momentum ( $\underline{P} = (\mathbf{p}, E)$ );
- Hamiltonian mechanics uses time as a paramter, i.e.  $\underline{R} = \underline{R}(t)$  and  $\underline{P} = \underline{P}(t)$ ;
- 45 • Time, or a quantity directly related to time such as the path length,  $s$ , is therefore a paramter. In addition,  $E$  and  $\mathbf{p}$  are related by the invariant mass of particle.

Para 3:

- Phase space is the position of the particle in coordinate and momentum space;
- Six phase-space coordinates are required and are usually taken to be  $\mathbf{r}$  and  $\mathbf{p}$ ;
- This section defines the 6-dimensional spaces used to describe particle phase space in the linear optics code.

### 3.1 Phase space

The 6D phase-space vector is defined in terms of the three-vector position and three vector momentum as:

$$\begin{bmatrix} \mathbf{r} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \\ \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \end{bmatrix} \quad (8)$$

The trajectory of the particle is evaluated as a function of time,  $t$  or  $s$ .

### 3.2 Trace space

Trace space is defined to simplify the calculation of the trajectory of particles through the accelerator lattice and is derived from the phase space expressed in the RPLC frame. Consider a particle with position  $(x_{\text{RPLC}}, y_{\text{RPLC}}, z_{\text{RPLC}})$  and momentum  $p_{\text{RPLC}}$  with components  $(p_{x \text{ RPLC}}, p_{y \text{ RPLC}}, p_{z \text{ RPLC}})$ . Taking the momentum of the reference particle to be  $p_{0\text{RPLC}}$ , the trace-space coordinates are given by:

$$\phi = \begin{pmatrix} x_{\text{RPLC}} \\ x'_{\text{RPLC}} \\ y_{\text{RPLC}} \\ y'_{\text{RPLC}} \\ t \\ \frac{\Delta p_{\text{RPLC}}}{p_{\text{RPLC}}} \end{pmatrix}; \quad (9)$$

55 where:

$$x'_{\text{RPLC}} = \frac{\partial x}{\partial s} = \frac{p_{x \text{ RPLC}}}{p_{\text{RPLC}}}; \quad (10)$$

$$y'_{\text{RPLC}} = \frac{\partial y}{\partial s} = \frac{p_{y \text{ RPLC}}}{p_{\text{RPLC}}}; \text{ and} \quad (11)$$

$$\Delta p_{\text{RPLC}} = p_{\text{RPLC}} - p_{0\text{RPLC}}. \quad (12)$$

## References

## Introduction

This document summarises the steps needed to set-up and run LhARA\_Beamline linear optics simulation of the LhARA beamline. 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 move 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 repository, 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.

95 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.