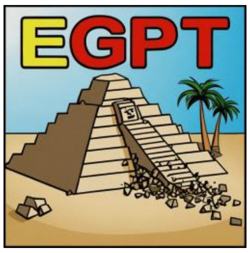
ASTeC-AP-EGPT-01.1 v0.1

22/08/2024

joe.crone@stfc.ac.uk

EGPT Dipoles



Joe Crone
Accelerator Physics Group
Accelerator Science and Technology Centre
STFC Daresbury Laboratory
Warrington WA4 4AD, United Kingdom

Keywords EGPT, Jitter Analysis, Error Analysis, Dipoles, Documentation, RUEDI

Summary

EGPT (Errorable General Particle Tracer) is a python-based package which acts as a wrapper for the General Particle Tracer code to simplify jitter and error analysis studies. Misalignment and erroring of dipoles within GPT requires special handling due to co-ordinate system changes and dependent parameters within the element. This report outlines a method of doing so whilst maintaining the physics consistency of the element.

Contents

Introduction

A convention for implementing dipoles in the GPT input is specified. The convention means that EGPT is not a fully generalised code for GPT jitter and error analysis however, it is unavoidable for proper handling of the dipole error analysis. An alternative is not possible because otherwise misalignment would be poorly handled and dipole parameters could be double errored or errored consistently. For ease of use and better standardisation the dipole convention in EGPT is based on the implementation of dipoles within the GPT manual.

Dipole Specification Convention

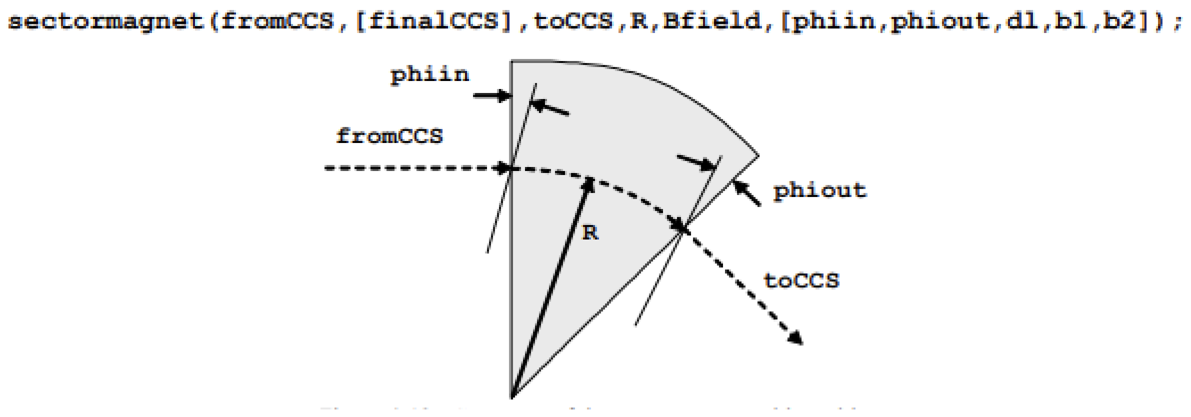
Explanation of the required convention/format for dipoles to be placed in the GPT input file for proper erroring

Need to also discuss how positional output should be setup

Future work will involve developing a code to convert any given dipole setup into a format specified by the convention here.

Dipole Misalignment Problem

Dipole misalignments in GPT are complicated by the form of the sector magnet command which takes a fromCCS and toCCS coordinate system. Typically, in the GPT input file this will be from the initial coordinate system to a bent co-ordinate system, as shown in Fig. 1. The bent co-ordinate system is specified using a ccs() command.



```
sectormagnet(fromCCS,[finalCCS],toCCS,R,Bfield,[phiin,phiout,dl,b1,b2]);
```

Figure 1. Top: sectormagnet() command from the GPT manual. Bottom: Schematic of a dipole implemented in the GPT code.

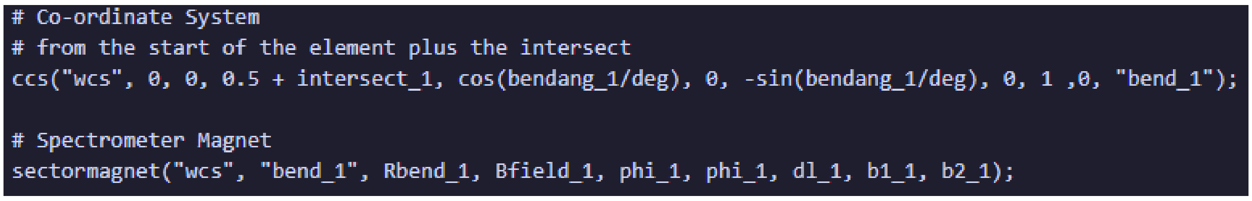
This does not allow the co-ordinate system in which the dipole magnet is placed to be misaligned, as in a standard element (see ASTeC-AP-EGPT-01 [cite]). Instead, the misaligned entrance and exit co-ordinate systems must be supplied. Additionally, these two co-ordinate systems must be consistent with each other in their misalignment specification or GPT assumes the beam is lost.

To modify a GPT sectormagnet() the misaligned entrance and exit co-ordinate systems must be generated using ccs() commands and the beam must be changed from the initial system to the initial errored co-ordinate system then from the errored bent system to the original bent co-ordinate system. Transferal between co-ordinate systems uses the ccsflip() command, shown in Fig.2, which transfers the beam between two co-ordinate systems at a specified location. Here we specify the start of the dipole as the location for the initial flip and the exit of the dipole as the position for the bent co-ordinate system flip. Two ccs() and ccsflip() elements must be generated and the existing sectormagnet() command must be modified.

For example, consider a single dipole placed after a short drift. The beam is initially in the world co-ordinate system wcs and must be transferred to the misaligned system wcs_err at the start of the dipole using a ccsflip(). The beam then traverses the sectormagnet() with co-ordinate systems wcs_err to bend_err_1, the misaligned and errored systems. The beam is then transferred from the misaligned bent co-ordinate system bend_err_1 to the original bent co-ordinate system using a ccsflip() command. A single misaligned dipole transformation is shown schematically in Fig. 2.

FIG 2

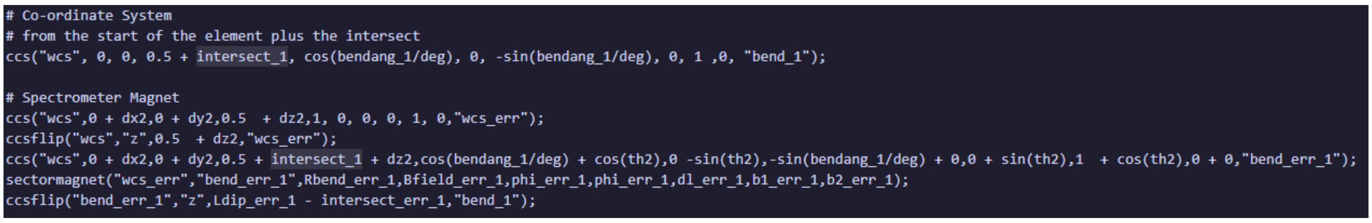
The code examples in Fig. 3 and 4 demonstrate the change in the initial dipole setup (sectormagnet() and bent ccs()) shown in Fig. X to the misaligned dipole setup in Fig.Y for a single dipole case as explained above.



```
# Co-ordinate System
# from the start of the element plus the intersect
ccs("wcs", 0, 0, 0.5 + intersect_1, cos(bendang_1/deg), 0, -sin(bendang_1/deg), 0, 1 ,0, "bend_1");

# Spectrometer Magnet
sectormagnet("wcs", "bend_1", Rbend_1, Bfield_1, phi_1, phi_1, dl_1, b1_1, b2_1);
```

Figure 2. The initial single dipole’s sectormagnet() and bent co-ordinate system ccs() bend_1 generation in the GPT input file.



```
# Co-ordinate System
# from the start of the element plus the intersect
ccs("wcs", 0, 0, 0.5 + intersect_1, cos(bendang_1/deg), 0, -sin(bendang_1/deg), 0, 1 ,0, "bend_1");

# Spectrometer Magnet
ccs("wcs",0 + dx2,0 + dy2,0.5  + dz2,1, 0, 0, 0, 1, 0,"wcs_err");
ccsflip("wcs","z",0.5  + dz2,"wcs_err");
ccs("wcs",0 + dx2,0 + dy2,0.5 + intersect_1 + dz2,cos(bendang_1/deg) + cos(th2),0 -sin(th2),-sin(bendang_1/deg) + 0,0 + sin(th2),1  + cos(th2),0 + 0,"bend_err_1");
sectormagnet("wcs_err","bend_err_1",Rbend_err_1,Bfield_err_1,phi_err_1,phi_err_1,dl_err_1,b1_err_1,b2_err_1);
ccsflip("bend_err_1","z",Ldip_err_1 - intersect_err_1,"bend_1");
```

Figure 3. The misaligned single dipole's sectormagnet() command, with the initial bent co-ordinate system bend_1, the initial errored co-ordinate system wcs_err and the errored bent co-ordinate system bend_err_1 as well as the two ccsflip() commands used to change to the misaligned co-ordinate systems at the entrance and exit of the dipole.

The misaligned dipole co-ordinate systems have been implemented such that the longitudinal misalignment dz is consistent throughout the co-ordinate system changes. The longitudinal misalignment is only handled in the initial errored coordinate system and the ccsflip() element, it is handled automatically elsewhere. The roll error (rotation about the longitudinal axis) is also handled similarly, only appearing in the initial errored co-ordinate system, with the further co-ordinate systems unmodified. Other rotations (pitch and yaw) are not yet supported for any elements.

The modifications shown in Fig. 3 are therefore generalised to any sectormagnet() element and can automatically handle any applied misalignment. These are automatically generated within the EGPT errored lattice. This requires that the initial dipole is specified with the correct convention shown in this report.

Dipole Independent Parameter Problem

Dipole parameter erroring in GPT

The sectormagnet() dipole command contains parameters as shown in Fig. 1, including the bending radius R, the magnetic flux density Bfield, the entrance and exit pole face angles of the dipole phiin and phiout, the fringe field offset dl and the Enge function coefficients b1 and b2. These parameters can be dependent upon one another or another variable. For example, the bending radius, magnetic flux density and entrance and exit angles phiin and phiout are all dependent upon the bending angle of the dipole for a rectangular dipole. Therefore, erroring all parameters independently would likely result in double or inconsistent erroring of the dipole.

This is solved by having a series of

Dipole Analysis & Co-ordinate System Ordering

How position analysis is performed when there are multiple screen commands in different co-ordinate systems

References

There are no sources in the current document.