Thin Element Comparison Between MAD-X and SixTrack

AUGUST 25, 2018

AUTHOR:
Joel Andersson

CERN
BE-ABP-HSS

SUPERVISOR:
Tobias Persson

Acknowledgements

I would like to thank the CERN openlab administration for the arrangement and planning of my stay at CERN, without whom I would not only have been unable to attend interesting lectures or visit Zurich, but also arrive here to work in the first place. I would like to thank all my fellow openlab summer students for making this summer one not only a vocational one but also one of camaraderie. Specifically I would like to thank my flatmates Niklas, Leonardo and Marco for distracting me from work every once in a while, and keeping my feet on the ground.

I am also thankful to my parents for supporting my career decisions, even if they involve me occasionally spending extended periods of time away from them. Particle physics, welding or professional wrestling, it would be all the same to them as long as my interest is earnest.

I would like to thank the SixTrack development team for being accomodating and friendly, something I appreciated immensely as a rookie developer.

Lastly, I have to thank the key person in this endeavour, my supervisor Tobias. There are many different styles of project supervision, some more authoritative, others less so, and I can confidently say that he struck a good balance. I am thankful that Tobias gave me the task, the means and left the execution to me, all the while providing indispensable help on demand. In the end, he trusted me enough to give me the reins of the project, the result of which I hope vindicates his decision. Thank you for the trust Tobias.

Project Specification

Simulation codes are very important to design and study particle accelerators. This project aims to compare the results in test cases between different simulation codes such as: MAD-X, PTC and SixTrack. The student will be involved in understanding the differences between the codes and find possible bugs. Most differences are due to differences in modeling but there will also be differences due to numerical issues. It will be important to separate these cases from real issues with the implementation of the physics.

Abstract

In this report thin, single elements were compared between MAD-X and SixTrack. A testing framework for efficient comparisons between the two tracking codes was developed. A few differences between the tracking codes were found then documented and two bugs, one in the Trombone element and one in the Solenoid element, were fixed. The results from the comparison indicated that SixTrack and MAD-X give virtually identical outputs for most elements up to numerical noise, and that the few differences that exist are either due to tractable bugs or to features not yet implemented in SixTrack.

Contents

List of Figures

List of Tables

Listings

1. Background

Before getting into the report, it would be prudent to first make a brief account of what is meant by the term 'beam tracking code'. A beam tracking code is in this context a simulation software for tracking the state of individual or bundles of particles as they move in an accelerator.

Finally, before accounting for the beam tracking codes used in the comparisons, it is worth mentioning that there are other beam tracking codes available and that these may be built upon other models.

1.1 Physical model

For the beam tracking codes covered in this report, an accelerator is described as a sequence of beam elements. Such a sequence of elements is referred to as a lattice. The reference system can be viewed as in Figure 1.1.

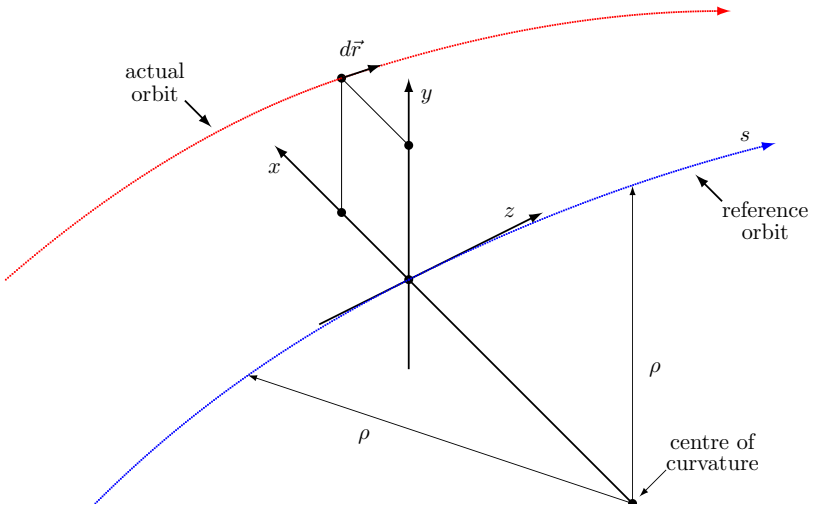


Figure 1.1: The local reference system used in the tracking codes [1].

The reference orbit is a series of straight line segments and circular arcs that together make up the accelerator path, under the assumption that all elements are perfectly aligned. The actual orbit of a particle in the accelerator need not (and for various reasons often does not) align with the reference orbit. Any given accelerator in this model has a reference orbit. Due to misalignment errors of the elements, fringe fields etc. the closed orbit of the particle might deviate from the reference orbit.

As can be seen in Figure 1.1, the position of the particle at any point along the reference orbit can be described by its deviation from it, hence the particle position can be described in a

curvilinear reference system in the coordinate system (x, y, s), where x denotes the horizontal displacement, y the vertical displacement and s its radial location in the accelerator.

For a full six-dimensional representation of the particle more quantities are needed. The total energy of the particle, E, the momentum component in x and y, and the latency in distance traveled serve to complete the set of physical quantities. Keeping track of these quantities for particles as they traverse an accelerator is called 6D tracking. Below in Table 1.1, the full nomenclature of the physical quantities used in this report is given.

Table 1.1: Nomenclature of the variables used

Variable	Description
c	Speed of light in vacuum
m	Mass of the accelerated particle
p	Momentum of the accelerated particle
p_x	Momentum in the x-direction of the accelerated particle
p_y	Momentum in the y-direction of the accelerated particle
E	Total energy of the accelerated particle
β	Relativistic beta factor $\beta \approx pc/E$ for high-energy physics
x	Horizontal deviation from the reference orbit
y	Vertical deviation from the reference orbit
x'	Horizontal angular deviation from the reference orbit
y'	Vertical angular deviation from the reference orbit
s	Coordinate tracking the projection on the reference orbit

1.2 MAD-X

MAD-X is a general-purpose tool for charged-particle optics design and studies of beam lines in accelerators [2]. The canonical variables used in MAD-X are shown in Table 1.2 below.

Table 1.2: Canonical tracking variables in MAD-X

Variable	Description
`X`	Horizontal deviation [m]
`PX`	Horizontal momentum normalized, $\mathrm{PX} = p_x/p_0$, [1]
`Y`	Vertical deviation [m]
`PY`	Vertical momentum normalized, $\mathrm{PY} = p_y/p_0$, [1]
`T`	Negative scaled time difference, $\mathrm{T} = -c\Delta t$, [m]
`PT`	Normalized energy difference, $\mathrm{PT} = \Delta E/p_0 c$. [1]

The input-to-output pipeline for MAD-X is straightforward and involves a single input `.madx` file resulting in a number of output files depending on predefined settings in the input.

1.3 SixTrack

SixTrack is a single particle 6D symplectic tracking code optimized for long term tracking in high energy rings. It is mainly used for the LHC for dynamic aperture studies, tune optimization, and collimation studies. [3], Table 1.3 shows what is referred to as the SixTrack tracking variables in this report.

Table 1.3: Tracking variables in SixTrack

Variable	Description
X	Horizontal deviation [mm]
XP	Horizontal angular deviation, $\mathrm{XP} = x'$, [mrad]
Y	Vertical deviation [mm]
YP	Vertical angular deviation, $\mathrm{YP} = y'$, [mrad]
Z	Negative scaled time difference, $\mathrm{Z} = -c\beta_0\Delta t$, [m]
PS	Normalized energy difference, $\mathrm{PS} = \Delta E/p_0 c\beta_0$. [1]

As an addendum, it should be noted SixTrack outputs the simulation result in terms of the deviation in energy not as PS, but as dE/E defined as $\Delta E/E_0$. Unlike MAD-X, the input-to-output pipeline is complex and takes as input a variable number of files ranging from a single one up to seven [3]. To mitigate this, an export option exists in MAD-X which allows lattices defined in a .madx file to be exported as several files usable as input to SixTrack [4]. We will refer to this functionality and its source code as mad_6t throughout the report.

1.4 Transformation between SixTrack and MAD-X

In order for any comparison between the output of SixTrack and MAD-X to take place, transformations between the two are necessary. For the X and Y variables the transform is only a factor of 1000, whereas the other transformations are shown below in equations 1.1 to 1.4.

$$\mathrm{PX} = 0.001 * (1 + \Delta P/p_0) * \mathrm{XP} \tag{1.1}$$

$$\mathrm{PY} = 0.001 * (1 + \Delta P/p_0) * \mathrm{YP} \tag{1.2}$$

$$\mathrm{T} = \frac{1}{\beta_0} * \mathrm{Z} \tag{1.3}$$

$$\mathrm{PT} = \beta_0 * \mathrm{PS} \tag{1.4}$$

2. A testing framework

2.1 An overview

In order to achieve fast comparisons between SixTrack and MAD-X a framework was developed. The concept as well as skeleton code for such a framework had already been proposed and produced by Tobias Persson, and this was then expanded and extended to meet the needs of the project. The basic idea behind the framework was to supply a means of defining accelerator lattices and testing parameters that were independent of the tracking codes. The comparison between SixTrack and MAD-X was naturally split into three modules: `compareSix2Mad.py`, `latticeConstructor.py` and `elementTest.py`.

A high-level description of how these modules interact is shown in figure 2.2, and a more in-depth explanation of the process as well as a complete example is found in the subsequent sections.

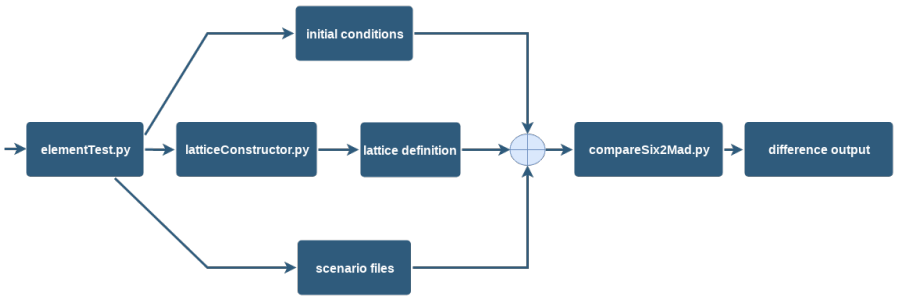


Figure 2.1: A high-level flowchart of the comparison process.

2.2 compareSix2Mad.py

This module contains the bulk of the necessary code to construct simulation scenarios independent of tracking code (MAD-X/SixTrack). Specifically, it provides the following functionality:

- transformation between SixTrack and MAD-X variables
- setup of testing scenario given input initial conditions
- execution of tracking code-independent scenario in MAD-X and SixTrack
- computation of difference between a SixTrack and MAD-X tracking output file

The transformations between the formats are based on the definitions from the MAD-X [2] and SixTrack [1] documentation.

A flowchart for the process in `compareSix2Mad.py` can be seen below in Figure 2.2.

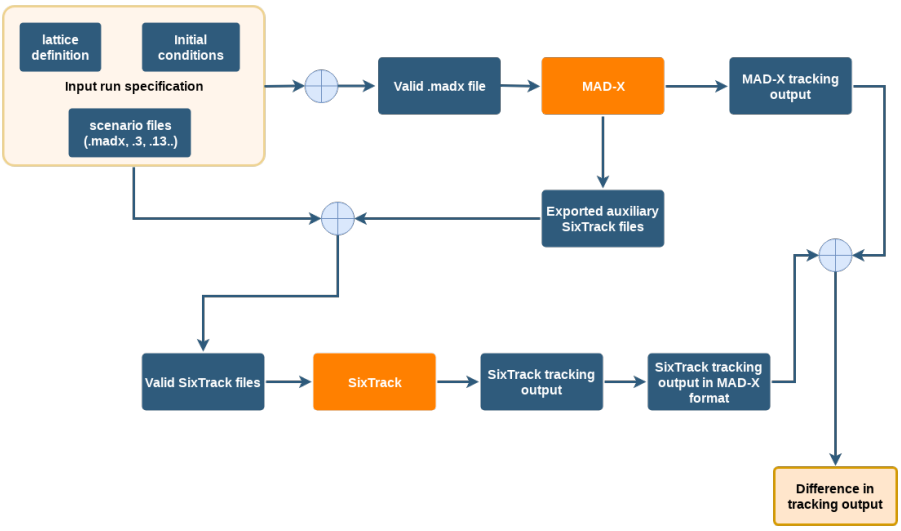


Figure 2.2: A flowchart of the comparison process within `compareSix2Mad.py`.

The functionality which allows for the construction of tracking code independent scenarios is elaborate string formatting inside 'scenario' files. A scenario file in this framework is defined as any file providing a template for creating a simulation run. An example of a scenario file necessary for producing a run for MAD-X can be seen below in Listing 2.1

```
circum=%(circum)s;

beam, particle=proton, energy = %(E0_GeV)s;
%(s_defs)s

seq: sequence, refer=entry, l=circum;
%(s_elems)s
endsequence;

use, sequence=seq;

twiss;
TRACK, ONEPASS=true, FILE="mad_output";
START, X=%(x_m)s, PX=%(PX)s, Y=%(y_m)s, PY=%(PY)s, T=%(time_mad)s, PT=%(pt_mad)s;
OBSERVE, PLACE="%(mad_track_element)s";
RUN, TURNS=%(nbr_turns)s;

sixtrack, CAVALL;

stop;
```

Listing 2.1: A .madx file used as a scenario

By defining all variables enclosed in `%(..)s` and then performing Python I/O + string formatting a valid MAD-X run can be created, and by making sure these variables match the corresponding ones in the SixTrack scenario files, runs identical across the two tracking codes can be constructed.

2.3 latticeConstructor.py

This module was introduced to provide an easy way to create string representations of the accelerator lattices without having to manually produce correctly formatted strings for the individual tests. Another reason for the addition of this functionality is that it further abstracts the testing

process; as the user is exempt from using the MAD-X language syntax, the tracking code independence of the code is enforced.

2.4 elementTest.py

As part of analyzing the difference between SixTrack and MAD-X, numerous test cases had to be covered. The module `elementTest.py` was introduced as the script containing all these different scenarios hard-coded as well as functionality for the plotting of the results.

2.5 A complete example

In this section a complete example, `elementTest.py` to output difference, is showcased. Starting off, the following code in Listing 2.2 defines a simulation setup as well as computes the difference between the results in MAD-X and SixTrack for it.

```
nbr_turns = 10
E0_GeV = 5
mad_init_coords = (0.001, 0.002, 0.003, 0.004, 0.005, 0.0)

l = Lattice(20)
l.addMultipoleDef(name='qf', order=2, KN=0.11755705)
l.addMultipoleDef(name='qd', order=2, KN=-0.11755705)
l.addRFCavityDef(name='cav', VOLT=100, LAG=0.0, L=0.0, HARMON=100, FREQ=0)
l.addElement(name='qd', pos=10)
l.addElement(name='cav', pos=10.001)
l.addElement(name='qf', pos=19.9999)
l_s = l.getLatticeDefinition()

diff = compareSix2Mad.compare('element_test', nbr_turns, E0_GeV,
    mad_init_coords, printToFile=0, lattice=l_s, norm='', silent=1, all_files=0,
    trk_element=('cav', 'cav'))
```

Listing 2.2: A definition of a comparison run in `elementTest.py`

This code takes the scenario 'element_test' and inserts the lattice above as well as the initial condition into the corresponding MAD-X file. Using the `.madx` file from Listing 2.1 would yield the following formatted one seen in Listing 2.3.

```
circum=20;

beam, particle=proton, energy = 5;

qf: MULTIPOLE, TILT=0, KNL={0,0.11755705}, KSL={0,0};
qd: MULTIPOLE, TILT=0, KNL={0,-0.11755705}, KSL={0,0};
cav: RFCAVITY, VOLT=100, LAG=0.0, L=0.0, HARMON=100;

seq: sequence, refer=entry, l=20;
qd: qd, at = 10;
cav: cav, at = 10.001;
qf: qf, at = 19.9999;
endsequence;

use, sequence=seq;

twiss;
TRACK, ONEPASS=true, FILE="mad_output";
START, X=0.001, PX=0.002, Y=0.003, PY=0.004, T=0.005, PT=0.0;
OBSERVE, PLACE="cav";
RUN, TURNS=10;

sixtrack, CAVALL;

```

```
stop;
```

Listing 2.3: A scenario `.madx` file correctly formatted

Unlike the `.madx` file in Listing 2.1, this file is valid MAD-X input and will output the result of a simulation in the file `mad_output.txt` among other outputs. The specific line `sixtrack, CAVALL;` in the above listing exports the lattice to SixTrack input format via the `mad_6t` [4] module mentioned in Section 1.3. The main exported file, `fort.2`, containing the lattice geometry can be seen below in Listing 2.4.

```
SINGLE ELEMENTS----------------------------------------------------------------
drift_0              0    0.000000000e+00    0.000000000e+00    1.000000000e+01
    ↪ 0.000000000e+00    0.000000000e+00    0.000000000e+00
qd                   2    1.175570500e-01    0.000000000e+00    0.000000000e+00
    ↪ 0.000000000e+00    0.000000000e+00    0.000000000e+00
drift_1              0    0.000000000e+00    0.000000000e+00    1.000000000e-03
    ↪ 0.000000000e+00    0.000000000e+00    0.000000000e+00
cav                 12    1.000000000e+02    1.000000000e+02    1.800000000e+02
    ↪ 0.000000000e+00    0.000000000e+00    0.000000000e+00
drift_2              0    0.000000000e+00    0.000000000e+00    9.998900000e+00
    ↪ 0.000000000e+00    0.000000000e+00    0.000000000e+00
qf                   2   -1.175570500e-01    0.000000000e+00    0.000000000e+00
    ↪ 0.000000000e+00    0.000000000e+00    0.000000000e+00
drift_3              0    0.000000000e+00    0.000000000e+00    1.000000000e-04
    ↪ 0.000000000e+00    0.000000000e+00    0.000000000e+00
NEXT
BLOCK DEFINITIONS--------------------------------------------------------------
1  1
BLOC1               drift_0
BLOC2               drift_1
BLOC3               drift_2
BLOC4               drift_3
NEXT
STRUCTURE INPUT----------------------------------------------------------------
BLOC1               qd                  BLOC2
cav                 BLOC3               qf
BLOC4
NEXT
```

Listing 2.4: A `fort.2` file produced from an export script in MAD-X

After having run the `.madx` file from Listing 2.1 and subsequently having produced a `fort.2` file as seen in Listing 2.4, a `fort.3` file is necessary in order to run a SixTrack simulation. These are dealt with in the same manner as the `.madx` file, that is, as a scenario file. Such a scenario file before formatting can be seen below in Listing 2.5.

```
GEOMETRYFILE Long FODO-cell for SixTrack tests
PRINT INPUT
COMMENT
Long FODO-cell for SixTrack test cases
NEXT
TRACKING PARAMETERS
%(nbr_turns)s 0 1 0.0 0.0 0 1
0 0 2 1 1
0 0 1 1 1 2000 1 1 1
NEXT
INITIAL COORDINATES
2 0.0 0.0 0.0 0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
%(E0_MeV)s
%(E0_MeV)s
%(E0_MeV)s
NEXT
LINEAR OPTICS CALCULATION
ELEMENT 0 1
NEXT
FLUC
0 1 4 0
NEXT
%(fc3_aux_s)s
%(trom_s)s
DUMP
%(six_track_element)s 1 660 2 six_output.txt
NEXT
ENDE
```

Listing 2.5: An unformatted `fort.3` scenario file.

This `fort.3` file necessitates the use of a `fort.13`, which also needs to be a scenario file and can be seen unformatted below in Listing 2.6.

```
%(x_mm)s
%(xp)s
%(y_mm)s
%(yp)s
%(sigma_z)s
%(delta_p)s
0.0
0.0
0.0
0.0
0.0
0.0
%(E0_MeV)s
%(E0_MeV_mod)s
%(E0_MeV)s
```

Listing 2.6: An unformatted `fort.13` scenario file.

These files are, after having run MAD-X on the `.madx` file, then formatted by `compareSix2mad.py` and the result can be seen in listings 2.7 and 2.8.

```
GEOMETRYFILE Long FODO-cell for SixTrack tests
PRINT INPUT
COMMENT
Long FODO-cell for SixTrack test cases
NEXT
TRACKING PARAMETERS
10 0 1 0.0 0.0 0 1
0 0 2 1 1
0 0 1 1 1 2000 1 1 1
NEXT
INITIAL COORDINATES
2 0.0 0.0 0.0 0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
5000
5000
5000
NEXT
LINEAR OPTICS CALCULATION
ELEMENT 0 1
NEXT
FLUC
0 1 4 0
NEXT
SYNC
          100   0.004038    100.000 0.      20.000000   938.272081   1
      1.          1.
NEXT
BEAM
   0.0000e+00    5.23428e+06    5.23428e+06   1.0000e+00   1.0000e-03   1   0
NEXT
DUMP
cav 1 660 2 six_output.txt
NEXT
ENDE
```

Listing 2.7: A formatted `fort.3` file.

```
1.0
2.0
3.0
4.0
4.911175572306078
0.0
0.0
0.0
0.0
0.0
0.0
0.0
5000
5000.0
5000
```

Listing 2.8: A formatted `fort.13` file.

Now that all necessary `fort.*` input files have been supplied, the SixTrack is run which supplies a `six_output.txt`. At this point the output of two identical simulation setups have been produced in MAD-X and SixTrack, and can be seen in Listings 2.9 and 2.10 below.

```
@ NAME             %19s  "TRACK.OBS0002.P0001"
@ TYPE             %08s  "TRACKOBS"
@ TITLE            %08s  "no-title"
@ ORIGIN           %16s  "5.04.01 Linux 64"
@ DATE             %08s  "11/08/18"
@ TIME             %08s  "22.14.16"
*       NUMBER       TURN                X                PX                Y                PY                T                PT
        ↪                                S                 E
          %d           %d              %le               %le               %le               %le               %le               %le
        ↪                              %le               %le
          1            1     0.02100466877     0.004468721562     0.04299934499    -0.001055000174     0.004898179122    -0.003065911824
        ↪                0                 5
          1            2     0.03302747241    0.0006127914671     0.06005875774    -0.004304949036     0.00244788788    -0.004602502929
        ↪                0                 5
          1            3    -0.0009445532783    -0.004104073467    -0.006587821805    -0.001554444758    -0.001142667101    -0.003884692777
        ↪                0                 5
          1            4    -0.03361065583    -0.003100538003    -0.06398839234     0.003359361747    -0.004186861377    -0.001261322632
        ↪                0                 5
          1            5    -0.01959328542     0.002196268646    -0.03244478746     0.00360546393    -0.005319621393     0.002066103559
        ↪                0                 5
          1            6     0.02151962553     0.004452670029     0.04365086788    -0.001111094336    -0.004007161142     0.004577465985
        ↪                0                 5
          1            7     0.03312037806    0.0006067191508     0.05966653283    -0.004294333073    -0.0008868617727     0.005134628119
        ↪                0                 5
          1            8    -0.0006044059001    -0.004066852311    -0.005948048307    -0.001602599202     0.002630317305     0.003483765028
        ↪                0                 5
          1            9    -0.03347830493    -0.003167542782    -0.06355518122     0.003292789461     0.004972154843    0.0003719108387
        ↪                0                 5
```

```
18          1         10   -0.02025299581    0.002108827573   -0.03373340648    0.003656515051    0.005034275285   -0.002778510522
        ↪              0                 5
```

Listing 2.9: Simulation output from MAD-X

```
1  # DUMP format #2, bez=cav          , number of particles=          2, dump period=          1, first turn=          1, last turn=
       ↪      -1, HIGH=F, FRONT=F
2  # ID turn s[m] x[mm] xp[mrad] y[mm] yp[mrad] z[mm] dE/E[1] ktrack
3         1         1      10.00100   2.100466877E+01   4.482714510E+00   4.299934499E+01   -1.058303705E+00   4.811163531E+00   -3.011446251E-03
       ↪ 2
4         2         1      10.00100   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00
       ↪ 2
5         1         2      10.00100   3.302747241E+01   6.156766096E-01   6.005875774E+01   -4.325217581E+00   2.404401433E+00   -4.520739992E-03
       ↪ 2
6         2         2      10.00100   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00
       ↪ 2
7         1         3      10.00100   -9.445532783E-01   -4.120370474E+00   -6.587821805E+00   -1.560617356E+00   -1.122367748E+00   -3.815681657E-03
       ↪ 2
8         2         3      10.00100   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00
       ↪ 2
9         1         4      10.00100   -3.361065583E+01   -3.104524723E+00   -6.398839234E+01   3.363681266E+00   -4.112482261E+00   -1.238915384E-03
       ↪ 2
10        2         4      10.00100   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00
       ↪ 2
11        1         5      10.00100   -1.959328542E+01   2.191658725E+00   -3.244478746E+01   3.597896137E+00   -5.225118928E+00   2.029399461E-03
       ↪ 2
12        2         5      10.00100   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00
       ↪ 2
13        1         6      10.00100   2.151962553E+01   4.432017386E+00   4.365086788E+01   -1.105940791E+00   -3.935974388E+00   4.496147824E-03
       ↪ 2
14        2         6      10.00100   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00
       ↪ 2
15        1         7      10.00100   3.312037806E+01   6.035643108E-01   5.966653283E+01   -4.272003247E+00   -8.711067827E-01   5.043412041E-03
       ↪ 2
16        2         7      10.00100   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00
       ↪ 2
17        1         8      10.00100   -6.044059008E-01   -4.052479978E+00   -5.948048308E+00   -1.596935586E+00   2.583590011E+00   3.421876349E-03
       ↪ 2
18        2         8      10.00100   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00
       ↪ 2
19        1         9      10.00100   -3.347830493E+01   -3.166343894E+00   -6.355518122E+01   3.291543168E+00   4.883825077E+00   3.653038957E-04
       ↪ 2
20        2         9      10.00100   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00
       ↪ 2
21        1        10      10.00100   -2.025299581E+01   2.114810168E+00   -3.373340648E+01   3.666888327E+00   4.944841964E+00   -2.729150592E-03
       ↪ 2
22        2        10      10.00100   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00   0.000000000E+00
       ↪ 2
```

Listing 2.10: Simulation output from SixTrack

After having done all of this, `compareSix2Mad.py` transforms the SixTrack output to MAD-X standard, and then calculates the difference between the two simulations. The result can be seen below in Listing 2.11.

```
1  0,0.0,-2.295316714473472e-13,0.0,5.816232911959318e-13,4.58947116421804e-13,3.8140974753519963e-13
2  1,0.0,-9.552037980031791e-15,-6.938893903907228e-18,6.31500060577217e-14,1.0180641052404127e-12,-7.387293901595093e-13
3  2,-1.0842021724855044e-19,-7.496347292912375e-14,8.673617379884035e-19,3.181734389845481e-13,2.79138886484509e-12,-2.5826285554586548e-12
4  3,6.938893903907228e-18,-4.253455226921332e-13,1.3877787807814457e-17,-1.3499357881530116e-13,2.9227592568403793e-12,-4.260648691495339e-12
5  4,-3.469446951953614e-18,2.503401132225558e-13,6.938893903907228e-18,-7.751993491567077e-13,-1.5914179696263630e-13,-4.849235596404711e-12
6  5,-3.469446951953614e-18,7.416983693886436e-13,0.0,-2.6263279745419865e-14,-5.113943990497915 6e-12,-1.7498823570716304e-12
7  6,-6.938893903907228e-18,-1.0564259970285983e-13,0.0,-3.851884089467461e-13,-8.023271500304241e-12,2.8692317619039 01e-12
8  7,-7.000000622550684e-13,6.265873236932507e-13,-9.999995623233282e-13,3.5093564339228056e-13,-8.293320891139544e-12,8.11052353749675e-12
9  8,0.0,9.428655772802941e-14,0.0,-2.9543078053362315e-13,-4.411359298683859e-12,1.0669684591142065e-11
10 9,0.0,-9.099496330045032e-13,0.0,-7.315472012880964e-13,3.2667904839578377e-12,8.764053285176265e-12
```

Listing 2.11: Simulation difference output in raw `.csv` format

The format of the output is raw `.csv`. The first column is the 0-indexed row number and the remaining columns are the difference expressed in MAD-X tracking variables between MAD-X and SixTrack simulation outputs per turn. These columns appear in the same order as covered in Table 1.2.

3. A comparison

3.1 Testing methodology

The presupposition behind the testing is that a particle for a fix lattice and set of initial conditions ought to behave identically in SixTrack and MAD-X. This is not necessarily always the case and anomalies to this can arise from either bugs or different assumptions in modelling. Since there exists an infinitude of feasible testing situations but only a limited amount of time, the governing principle for the choice of test cases was to cover as much ground as possible while keeping time spent per test case at a minimum. The product of this principle was the following testing process:

1. Define initial conditions.
2. Create a reference lattice.
3. Insert the element to be tested with preordained parameters at the start of the lattice,
4. Run simulation for one turn with tracking directly after particle passes the testing element.
5. Possibly reiterate from 3. with a variation of element parameters.
6. Investigate difference between MAD-X and SixTrack.

While this testing process is simple, it provides efficient means of testing the individual elements in SixTrack and MAD-X. Since the particle is tracked at the very beginning of the lattice for a setup that is equivalent across SixTrack and MAD-X, any difference between the outputs must then be due to how the element being investigated is implemented.

3.2 Testing Results

All testing was performed on SixTrack version 5.0.2 (commit: `1e3f343fb2`) and MAD-X version 5.04.01 (commit: `5faaf3e092`) and using the initial conditions:

$$(\mathtt{X}, \mathtt{PX}, \mathtt{Y}, \mathtt{PY}, \mathtt{T}, \mathtt{PT}) = (0.001, 0.002, 0.003, 0.004, 0.005, 0.01)$$

with a reference energy of 5 GeV and a reference lattice of the form seen below in Listing 3.1.

```
qf: MULTIPOLE, TILT=0, KNL={0,0.11755705}, KSL={0,0};
qd: MULTIPOLE, TILT=0, KNL={0,-0.11755705}, KSL={0,0};
cav: RFCAVITY, VOLT=100, LAG=0.0, L=0.0, HARMON=100;

seq: sequence, refer=entry, l=20;
qd: qd, at = 10;
cav: cav, at = 10.001;
qf: qf, at = 19.9999;
endsequence;
```

Listing 3.1: Reference lattice used for the individual element testing.

Results for each element are based on the analysis of graphs depicting the difference in the output from SixTrack and MAD-X. The results of these are summarized in matrices where green denotes equal to numerical precision, blue a difference due to a feature not yet implemented and red a difference for any other reason. Every cell in these matrices corresponds to a test case with a matching pair of graphs, found in the Appendix A of the report.

3.2.1 Drift

For the investigation of the drift, the only variable parameter is the length of the drift which was varied between 0 and 10 meters in steps of 0.1. The result from the investigation can be seen below in Figure 3.1.

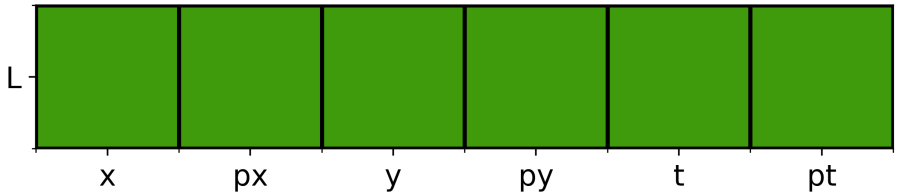


Figure 3.1: Qualitative analysis of the Drift.

3.2.2 RF Cavity

For the investigation of the RF Cavity element `VOLT`, `LAG` and `HARM`onic number were independently varied. The parameters set for the test cases are seen below where `k` goes from 0 to 99

1. `VOLT`: `VOLT=k+1, LAG=0.25, L=0.0, HARMON=100`
2. `LAG`: `VOLT=100, LAG=0.01*k, L=0.0, HARMON=100`
3. `HARM`: `VOLT=100, LAG=0.25, L=0.0, HARMON=k`

The qualitative analysis can be seen below in Figure 3.2.

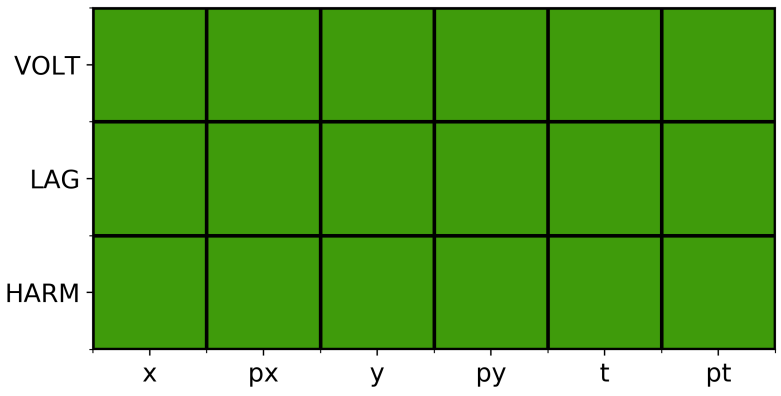


Figure 3.2: Qualitative analysis of the RF Cavity.

3.2.3 Kicker

For the investigation of the kicker the horizontal kick `HKICK`, the vertical kick `VKICK` and the tilt `TILT` were independently varied. The parameters set for the test cases are seen below where `k` goes from 0 to 99.

1. `HKICK`: `HKICK=0.0003*(k+1), VKICK=0.0, L=0.0, TILT=0.0`
2. `VKICK`: `HKICK=0.0, VKICK=0.0003*(k+1), L=0.0, TILT=0.0`
3. `TILT`: `HKICK=0.003, VKICK=0.0, L=0.0, TILT=0.02*pi*k`

The qualitative analysis is shown below in Figure 3.3.

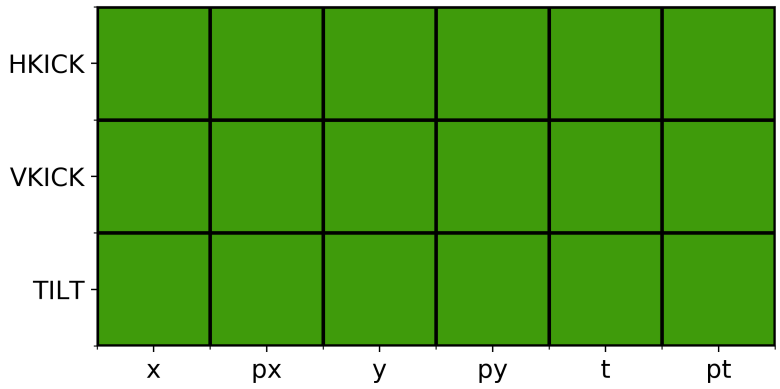


Figure 3.3: Qualitative analysis of the Kicker.

3.2.4 Solenoid

For the investigation of the solenoid the solenoid strength `KS`, the integrated strength `KSI` were varied independently and at the same time (where `KSI` was set to be equal to `KS`). The parameters set for the test cases are seen below where `k` goes from 0 to 99.

1. `KS`: `KS=0.1*k, KSI=0.000001, L=0.0`
2. `KSI`: `KS = 0.000001, KSI = k*0.01, L=0.0`
3. `KSI=KS`: `KS = k*0.005, KSI = k*0.005, L=0.0`

The qualitative analysis can be seen below in figure 3.4

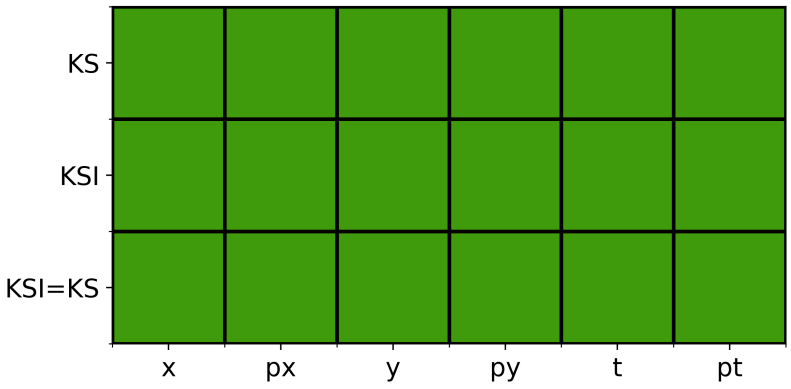


Figure 3.4: Qualitative analysis of the Solenoid.

3.2.5 Trombone

For the investigation of the Trombone element, a kick was varied in each coordinate independently of the other. The transfer matrix was set to the identity matrix and for the kick in each dimension, every other kick dimension was set to zero. The kick was set to `0.0001*k` for every dimension tested where `k` went from 0 to 99. The result of the analysis can be found in Figure 3.5.

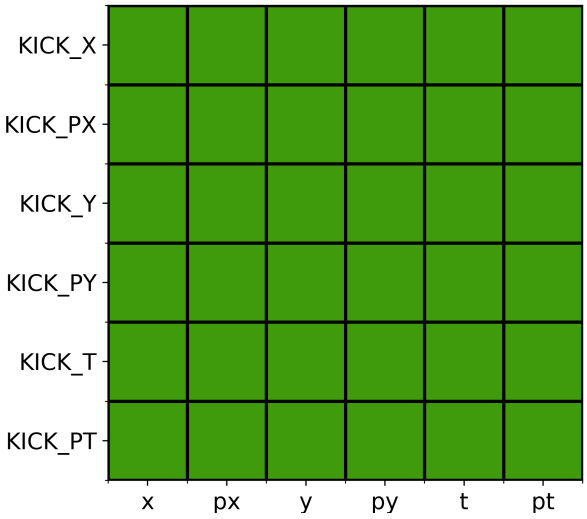


Figure 3.5: Qualitative analysis of the Trombone Kick.

3.2.6 Dipole

For the investigation of the Dipole, the normal dipole strength `KN`, the skew dipole strength `KS` and the `TILT` for a normal and a skewed dipole were varied. The parameters set for the test cases are seen below where `k` goes from 0 to 99.

1. `KN`: `KN=0.0008*(k+1), KS=0.0, TILT=0.0, THICK=false`
2. `KS`: `KN=0.0, KS=0.0008*(k+1), TILT=0.0, THICK=false`
3. `TILTKN`: `KN=0.008, KS=0.0, TILT=0.02*pi*k, THICK=false`
4. `TILTKS`: `KN=0.0, KS=0.008, TILT=0.02*pi*k, THICK=false`

The qualitative analysis can be seen below in Figure 3.6.

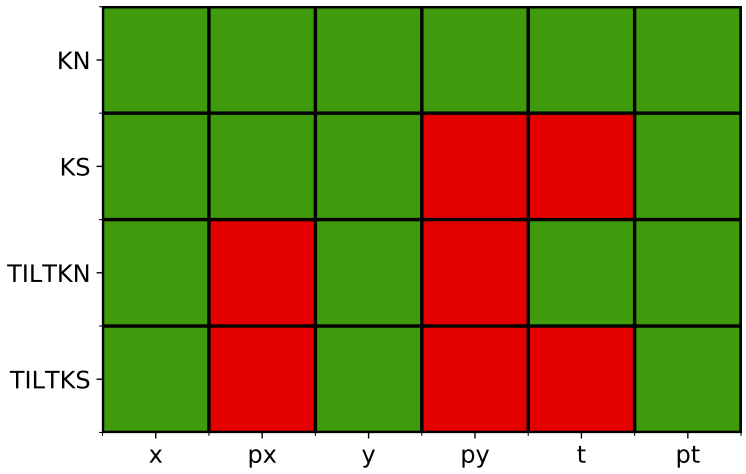


Figure 3.6: Qualitative analysis of the Dipole.

3.2.7 Quadrupole

For the investigation of the Quadrupole, the normal quadrupole strength `KN`, the skew quadrupole strength `KS` and the `TILT` for a normal and a skewed Quadrupole were varied. The parameters set for the test cases are seen below where `k` goes from 0 to 99.

1. `KN`: `KN=0.0008*(k+1), KS=0.0, TILT=0.0, THICK=false`
2. `KS`: `KN=0.0, KS=0.0008*(k+1), TILT=0.0, THICK=false`
3. `TILTKN`: `KN=0.008, KS=0.0, TILT=0.02*pi*k, THICK=false`
4. `TILTKS`: `KN=0.0, KS=0.008, TILT=0.02*pi*k, THICK=false`

The qualitative analysis can be seen below in Figure 3.7.

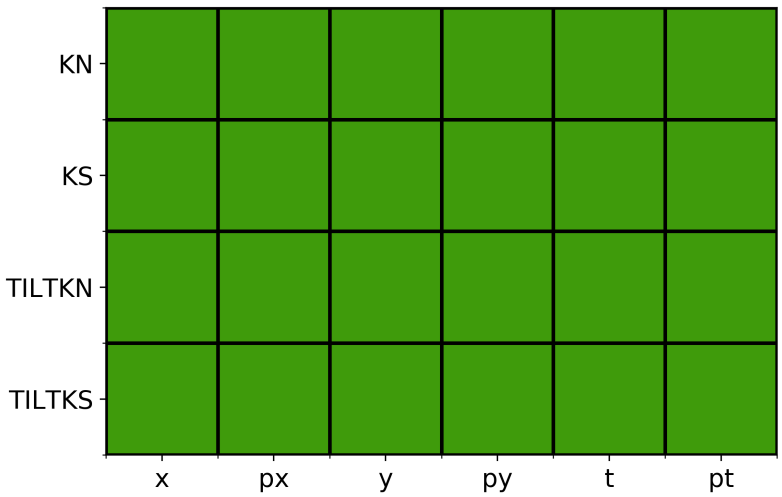


Figure 3.7: Qualitative analysis of the Quadrupole.

3.2.8 Sextupole

For the investigation of the Sextupole, the normal sextupole strength `KN`, the skew sextupole strength `KS` and the `TILT` for a normal and a skewed Sextupole were varied. The parameters set for the test cases are seen below where `k` goes from 0 to 99.

1. `KN`: `KN=0.0008*(k+1), KS=0.0, TILT=0.0, THICK=false`
2. `KS`: `KN=0.0, KS=0.0008*(k+1), TILT=0.0, THICK=false`
3. `TILTKN`: `KN=0.008, KS=0.0, TILT=0.02*pi*k, THICK=false`
4. `TILTKS`: `KN=0.0, KS=0.008, TILT=0.02*pi*k, THICK=false`

The qualitative analysis can be seen below in Figure 3.8.

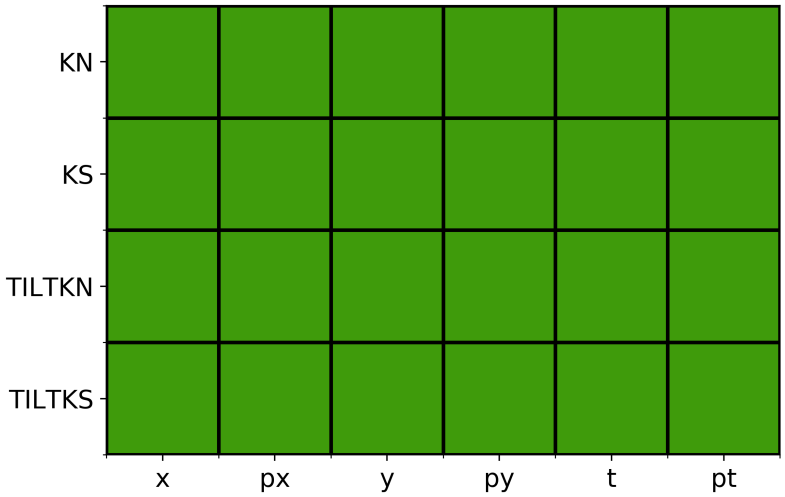


Figure 3.8: Qualitative analysis of the Sextupole.

3.2.9 Octupole

For the investigation of the Octupole, the normal octupole strength `KN`, the skew octupole strength `KS` and the `TILT` for a normal and a skewed Octupole were varied. The parameters set for the test cases are seen below where `k` goes from 0 to 99.

1. `KN`: `KN=0.0008*(k+1), KS=0.0, TILT=0.0, THICK=false`
2. `KS`: `KN=0.0, KS=0.0008*(k+1), TILT=0.0, THICK=false`
3. `TILTKN`: `KN=0.008, KS=0.0, TILT=0.02*pi*k, THICK=false`
4. `TILTKS`: `KN=0.0, KS=0.008, TILT=0.02*pi*k, THICK=false`

The qualitative analysis can be seen below in Figure 3.9.

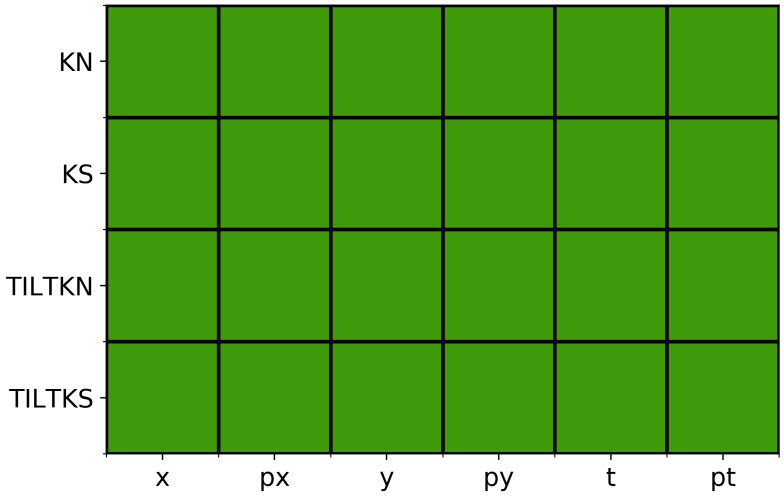


Figure 3.9: Qualitative analysis of the Octupole.

3.2.10 Decapole

For the investigation of the Decapole, the normal decapole strength `KN`, the skew decapole strength `KS` and the `TILT` for a normal and a skewed Decapole were varied. The parameters set for the test cases are seen below where `k` goes from 0 to 99.

1. `KN`: `KN=0.0008*(k+1), KS=0.0, TILT=0.0, THICK=false`
2. `KS`: `KN=0.0, KS=0.0008*(k+1), TILT=0.0, THICK=false`
3. `TILTKN`: `KN=0.008, KS=0.0, TILT=0.02*pi*k, THICK=false`
4. `TILTKS`: `KN=0.0, KS=0.008, TILT=0.02*pi*k, THICK=false`

The qualitative analysis can be seen below in Figure 3.10.

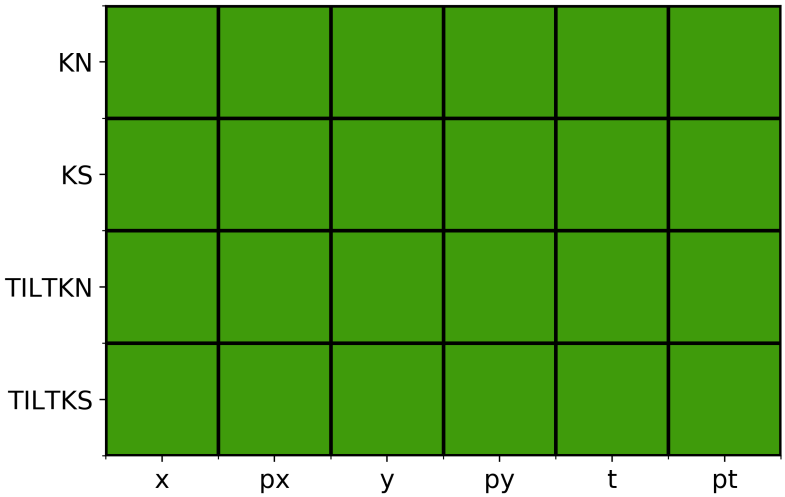


Figure 3.10: Qualitative analysis of the Decapole.

3.2.11 RF Dipole

For the investigation of the RF Dipole the normal dipole strength `KN`, the skew dipole strength `KS`, the frequency `FREQ`, the normal phase `PN`, the `TILT` for a normal and skewed RF Dipole as well as the voltage `VOLT` were independently varied. The parameters set for the test cases are seen below where `k` goes from 0 to 99.

1. `KN`: `KN=0.0002*(k+1), KS=0.0, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.0, PN=0.0, PS=0.0`
2. `KS`: `KN=0.0, KS=0.0002*(k+1), LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.0, PN=0.0, PS=0.0`
3. `FREQ`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.0, FREQ=2*(k+1), TILT=0.0, PN=0.0, PS=0.0`
4. `PN`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.0, PN=0.01*k, PS=0.0`
5. `TILTKN`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.02*pi*k, PN=0.0, PS=0.0`
6. `TILTKS`: `KN=0.0, KS=0.02, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.02*pi*k, PN=0.0, PS=0.0`
7. `VOLT`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.02*k, FREQ=200, TILT=0.0, PN=0.0, PS=0.0`

The qualitative analysis can be seen below in Figure 3.11.

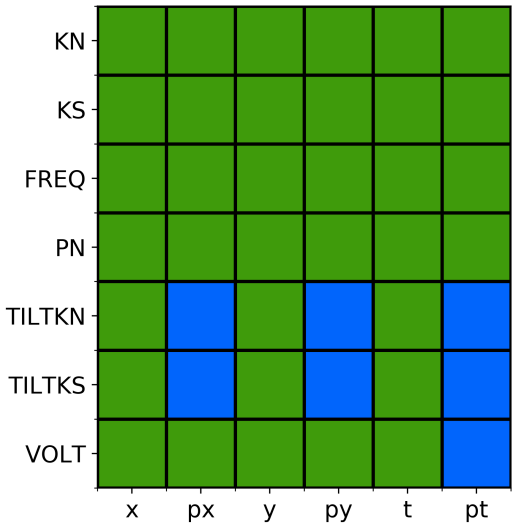


Figure 3.11: Qualitative analysis of the RF Dipole.

3.2.12 RF Quadrupole

For the investigation of the RF Quadrupole the normal quadrupole strength `KN`, the skew quadrupole strength `KS`, the frequency `FREQ`, the normal phase `PN`, the `TILT` for a normal and skewed RF Quadrupole as well as the voltage `VOLT` were independently varied. The parameters set for the test cases are seen below where `k` goes from 0 to 99.

1. `KN`: `KN=0.0002*(k+1), KS=0.0, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.0, PN=0.0, PS=0.0`
2. `KS`: `KN=0.0, KS=0.0002*(k+1), LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.0, PN=0.0, PS=0.0`
3. `FREQ`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.0, FREQ=2*(k+1), TILT=0,0, PN=0.0, PS=0.0`
4. `PN`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.0, PN=0.01*k, PS=0.0`
5. `TILTKN`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.02*pi*k, PN=0.0, PS=0.0`
6. `TILTKS`: `KN=0.0, KS=0.02, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.02*pi*k, PN=0.0, PS=0.0`
7. `VOLT`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.02*k, FREQ=200, TILT=0.0, PN=0.0, PS=0.0`

The qualitative analysis can be seen below in Figure 3.12.

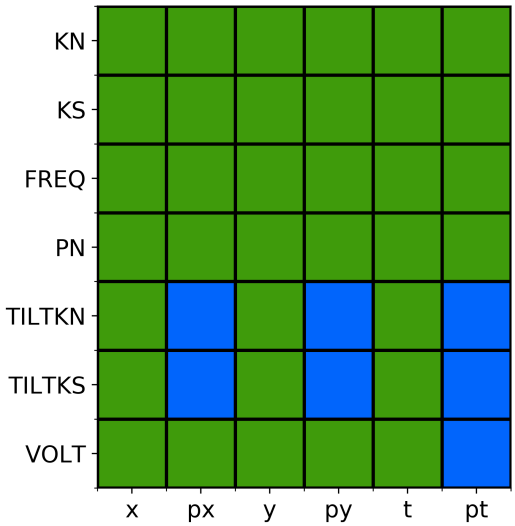


Figure 3.12: Qualitative analysis of the RF Quadrupole.

3.2.13 RF Sextupole

For the investigation of the RF Sextupole the normal sextupole strength `KN`, the skew sextupole strength `KS`, the frequency `FREQ`, the normal phase, the `TILT` for a normal and skewed RF Sextupole as well as the voltage `VOLT` were independently varied. The parameters set for the test cases are seen below where `k` goes from 0 to 99.

1. `KN`: `KN=0.0002*(k+1), KS=0.0, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.0, PN=0.0, PS=0.0`
2. `KS`: `KN=0.0, KS=0.0002*(k+1), LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.0, PN=0.0, PS=0.0`
3. `FREQ`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.0, FREQ=2*(k+1), TILT=0.0, PN=0.0, PS=0.0`
4. `PN`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.0, PN=0.01*k, PS=0.0`
5. `TILTKN`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.02*pi*k, PN=0.0, PS=0.0`
6. `TILTKS`: `KN=0.0, KS=0.02, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.02*pi*k, PN=0.0, PS=0.0`
7. `VOLT`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.02*k, FREQ=200, TILT=0.0, PN=0.0, PS=0.0`

The qualitative analysis can be seen below in Figure 3.13.

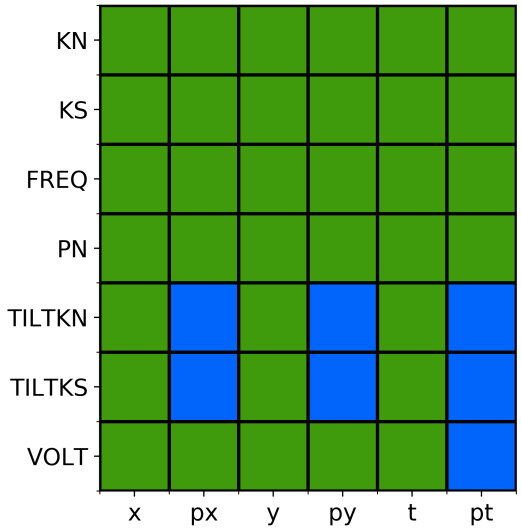


Figure 3.13: Qualitative analysis of the RF Sextupole.

3.2.14 RF Octupole

For the investigation of the RF Octupole the normal octupole strength `KN`, the skew octupole strength `KS`, the frequency `FREQ`, the normal phase `PN`, the `TILT` for a normal and skewed RF Octupole as well as the voltage `VOLT` were independently varied. The parameters set for the test cases are seen below where `k` goes from 0 to 99.

1. `KN`: `KN=0.0002*(k+1), KS=0.0, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.0, PN=0.0, PS=0.0`
2. `KS`: `KN=0.0, KS=0.0002*(k+1), LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.0, PN=0.0, PS=0.0`
3. `FREQ`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.0, FREQ=2*(k+1), TILT=0.0, PN=0.0, PS=0.0`
4. `PN`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.0, PN=0.01*k, PS=0.0`
5. `TILTKN`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.02*pi*k, PN=0.0, PS=0.0`
6. `TILTKS`: `KN=0.0, KS=0.02, LAG=0.25, VOLT=0.0, FREQ=200, TILT=0.02*pi*k, PN=0.0, PS=0.0`
7. `VOLT`: `KN=0.02, KS=0.0, LAG=0.25, VOLT=0.02*k, FREQ=200, TILT=0.0, PN=0.0, PS=0.0`

The qualitative analysis can be seen below in Figure 3.14.

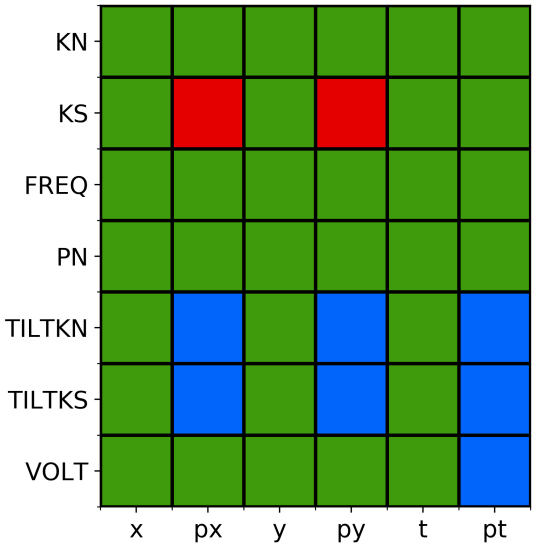


Figure 3.14: Qualitative analysis of the RF Octupole.

3.3 Analysis of differences

3.3.1 Dipole

Of all the regular multipole elements, the only one that displays any anomalies is the Dipole element. Specifically, the results suggests that the `TILT` and `KS` parameters leads to different results between SixTrack and MAD-X, and very noticeably different at that. One of the errors is properly accounted for in Figure 3.15 below.

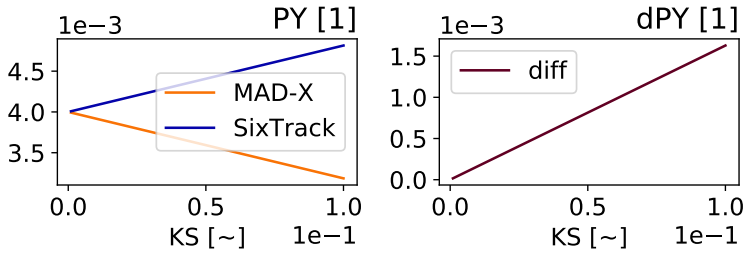


Figure 3.15: Quantitative analysis of `PY` when `KS` is being varied independently for the Dipole element.

As can be seen in Figure 3.15 (and numerically verified) `KS` in SixTrack and `KS` in MAD-X are off by a minus sign. This can be fixed by switching the sign in the MAD-X to SixTrack conversion in `mad_6t` or by internally making them consistent. A difference of a different character can be seen in Figure 3.16.

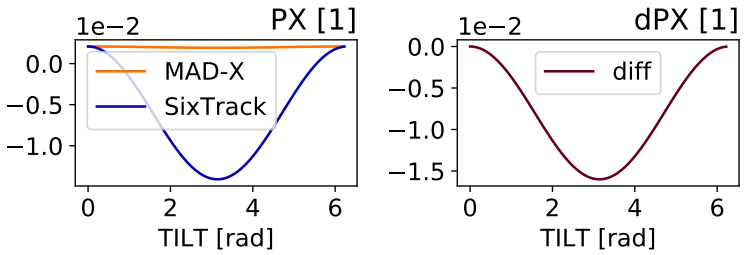


Figure 3.16: Quantitative analysis of PX when TILT is being varied independently with non-zero KN for the Dipole element.

Studying the raw data displayed in Figure 3.16 reveals that the MAD-X curve assumes a value in the range of 10^{-5} for a TILT of π and has the same shape as the SixTrack curve, suggesting that it could be a scaling issue.

3.3.2 RF Multipoles and TILT/VOLT

The qualitative analysis of the RF Multipoles reiterates an already known fact: TILT and VOLT are not implemented in SixTrack. For example, naively exporting a tilted normal RF Quadrupole via mad_6t for example yields results as can be seen in Figure 3.17.

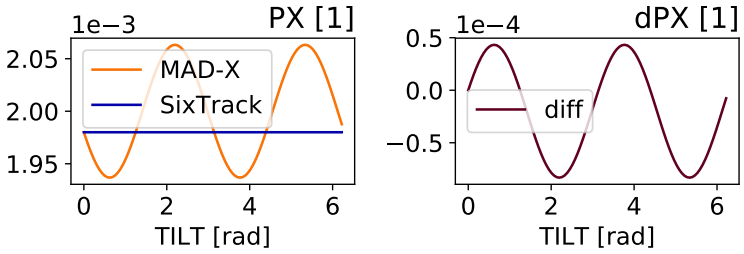


Figure 3.17: Quantitative analysis of PX when TILT is being varied independently with non-zero KN for the RF Quadrupole element.

This kind of deviation warrants no further inquiries until SixTrack supports the functionality investigated.

3.3.3 RF Octupole

The only RF Multipole that exhibits an unknown difference is the RF Octupole for variations in the skew octupole strength KS. This is displayed for PX below in Figure 3.18

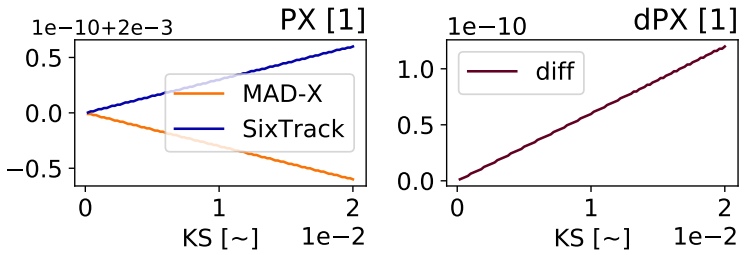


Figure 3.18: Quantitative analysis of `PX` when `KS` is being varied independently for the RF Octupole element.

This error appears similar to that of the Dipole when varying its skew dipole strength, and the author would guess that the cause is a sign error somewhere in the implementation of either MAD-X or SixTrack, or that `mad_6t` is off by a minus sign.

3.4 Bugfixes

This section describes two bugged scenarios that were found in SixTrack/MAD-X during the project and was subsequently fixed.

3.4.1 Solenoid

The Solenoid element had two notable differences which can be seen in Figure 3.19 and 3.20.

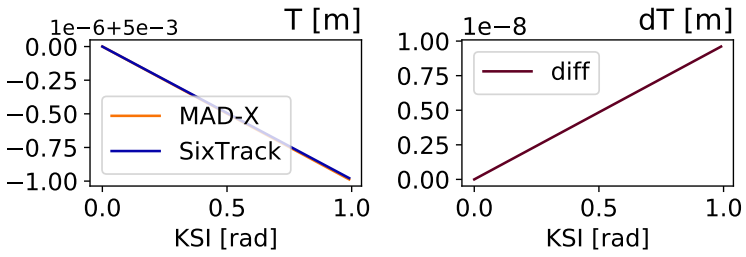


Figure 3.19: Quantitative analysis of `T` when `KSI` is being varied independently for the Solenoid element.

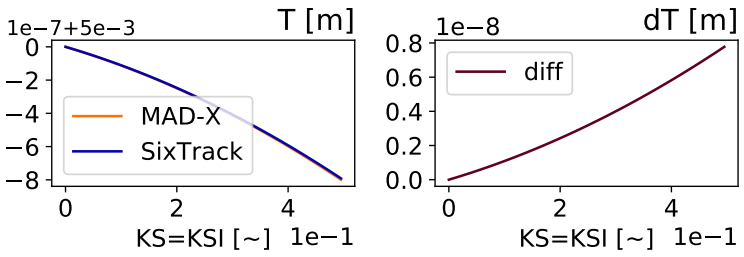


Figure 3.20: Quantitative analysis of `T` when `KS` and `KSI` are both varied for the RF Octupole element.

After some investigation the error was identified and is displayed below in Listing 3.2

```
onedp    =  (one+dpsv(j))/mtc(j)
fppsig   = ( one + ((e0f/e0) **2)*temptr(6) ) / onedp

temptr(1)=xv(1,j)
temptr(2)=yv(1,j)
temptr(3)=xv(2,j)
temptr(4)=yv(2,j)

temptr(6)=(ejv(j)-e0)/(e0f*(e0f/e0))
```

Listing 3.2: Snipped of bugged code in `kickvso1.f90`.

The error was that an internal variable in the Solenoid code, `fppsig`, used the value of `temptr(6)` before it was updated. Since `temptr(6)` stores `PS`, this effectively meant that the Solenoid used last turn's energy for the computations, and on the first turn it always used the reference energy. This was corrected by moving the update of `fppsig` to just after `temptr(6)`, and now the error cases in Figure 3.19 and 3.20 instead yield Figure 3.21 and 3.22 respectively.

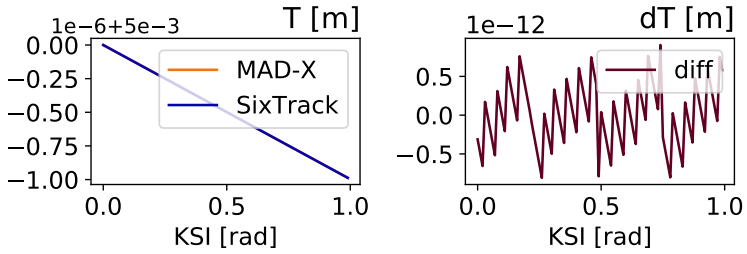


Figure 3.21: Quantitative analysis of `T` when `KSI` is being varied independently for the Solenoid element.

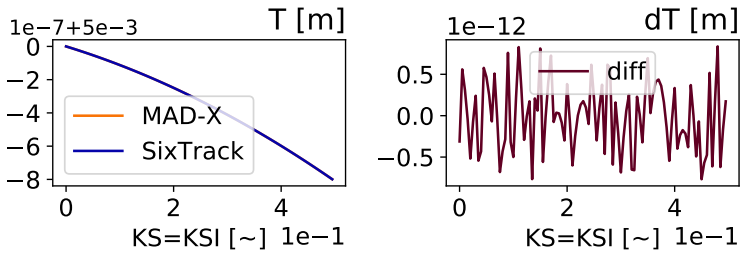


Figure 3.22: Quantitative analysis of `T` when `KS` and `KSI` are both varied for the RF Octupole element.

3.4.2 Trombone

As could be seen in Figure 3.5, the Trombone suffered from several notable differences. A number of graphs portraying these can be seen below in Figure 3.23, 3.24 and 3.25.

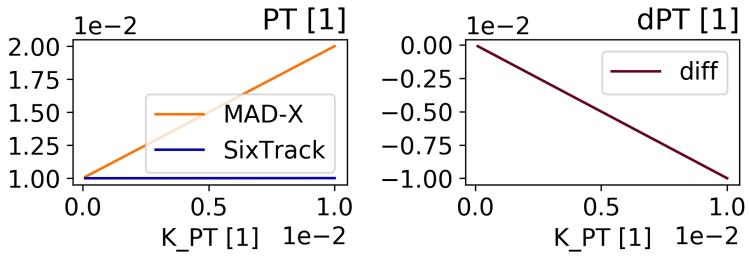


Figure 3.23: Quantitative analysis of PT when varying a kick in PT for the Trombone element.

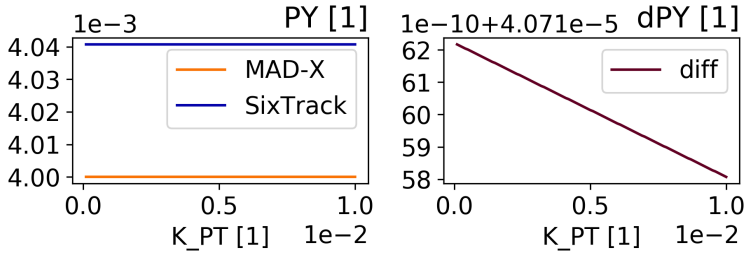


Figure 3.24: Quantitative analysis of PY when varying a kick in PT for the Trombone element.

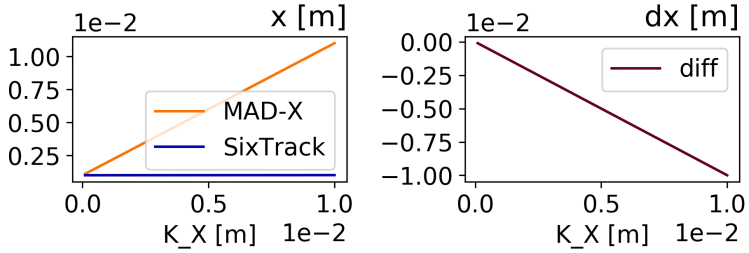


Figure 3.25: Quantitative analysis of X when varying a kick in X for the Trombone element.

As the element is only supposed to perform an affine transform of the state of the particle, the difference could not have arisen from different physical assumptions. After investigating the source code of SixTrack and MAD-X two bugs were found. The first bug was due to flipped indices in the SixTrack source file `track_thin.f90`, as can be seen below in Listing 3.3

```
!Out-commented code below was bugged
!yv(j,1) = yv(j,1)*mtc(j)/(one+dpsv(j))
!yv(j,2) = yv(j,2)*mtc(j)/(one+dpsv(j))
yv(1,j) = yv(1,j)*mtc(j)/(one+dpsv(j))
yv(2,j) = yv(2,j)*mtc(j)/(one+dpsv(j))
```

Listing 3.3: Trombone bugfix in `track_thin.f90`

This eliminated the asymmetry between PX and PY seen in Figure 3.5, but there were still errors present. After some investigation it was deduced that the bug was present in the `mad_6t` module and consisted of incorrect scaling. The fix is seen below in Listing 3.4.

```
if ((i+1)%dim==0){
  //value=value/beta
  value=value*beta;
}
if (i%dim==0){
  //value=value*beta
  value=value/beta;
}
if (i>(dim+24) && i <(31+dim)){
  //value=value*beta
  value = value/beta;
}
if (i>(dim+30) && i < (37+dim)){
  //value=value/beta
  value = value*beta;
}
// The entire if-clause below was added in the bugfix
if (i<(dim+1)){
  value = value * 1000;
}
```

Listing 3.4: Trombone bugfix in `mad_6track.c`

Further testing suggests that these two bugfixes have corrected the Trombone element since no anomalies can be found anymore. Specifically, the previously deviating test cases in Figure 3.23, 3.24 and 3.25 now yield the output in Figure 3.26, 3.27 and 3.28 respectively.

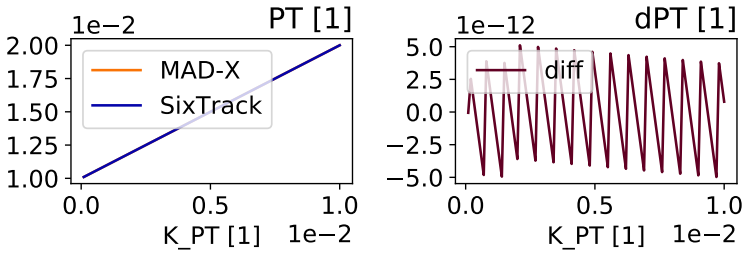


Figure 3.26: Quantitative analysis of `PT` when varying a kick in `PT` for the Trombone element.

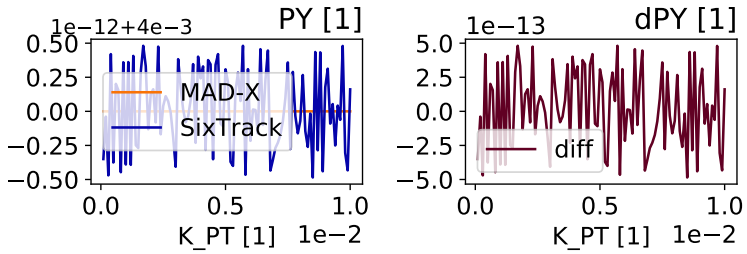


Figure 3.27: Quantitative analysis of `PY` when varying a kick in `PT` for the Trombone element.

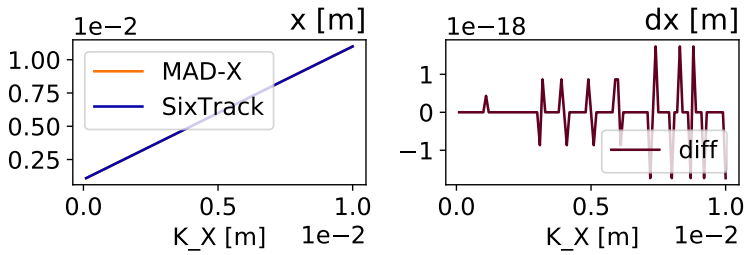


Figure 3.28: Quantitative analysis of `X` when varying a kick in `X` for the Trombone element.

4. Conclusions

4.1 Testing framework

A testing framework has been developed that supports robust comparison between MAD-X and SixTrack tracking. The scripts can easily be tweaked to support additional features, such as conversions between different twiss parameters and the possibility to modify the `TRACK` and `INIT` switches. These are minor modifications that can easily be added by following the set-up standard in the framework. The reason these and other features have not been implemented is due to time constraints and priorities arising from them.

In the current state, arbitrary lattices consisting of thin elements can be created inside the framework (`latticeConstructor.py`) and subsequently used for comparison. Alternatively, predefined lattices in `.madx` files can be directly run through `compareSix2Mad.py` to achieve comparisons. The `elementTest.py` script should only be viewed as an example of what is possible to achieve in the framework and contains the code for producing all tests in this report.

As a final comment on the testing framework as is I would like to point out some challenges in constructing it, as well as further developing it. Since SixTrack necessitates multiple input files and the MAD-X-to-SixTrack conversion outputs auxiliary files for this purpose, the script inherently involves a considerable amount of I/O. Not all `fort.*` are used as input on every run and which ones are used often differ in-between scenarios. Given the large number of variations in inputs and their interdependencies, not all possible combinations are supported by the current framework, and because of the considerable I/O involved in producing these input files nor would it necessarily be a good idea to include the most exotic cases in the `compare(...)` function. The most sound course of action for the further development of the testing framework is, in the author's opinion, to create separate functions for the more exotic use-cases building upon the already present functions, rather than striving towards one universal function for all comparison.

4.2 Testing results

Summing up the testing results, this report overall validates the presupposition that single, thin elements common to MAD-X and SixTrack perform the same up to numerical noise. Having said that, as could be seen in Section 3.2, there are nevertheless a few differences. These differences are for the most part relating to exotic use cases rarely used. In spite of this, it is still imperative that these discrepancies are fixed in order to expand what is the ground truth between the two tracking codes.

4.3 Future work

There is considerable future work to be done on the comparison between MAD-X and SixTrack. Expanding the testing framework has already been touched upon in Section 4.1 and would be one possible future endeavour, e.g. better support for reading `twiss` files. Having said this, the current testing framework should be sufficient for comparing thick elements (support for defining

such elements is easily added to `latticeConstructor.py`) and running comparisons on 'simple' `twiss` files.

A future project that would considerably contribute toward the work already presented in this report is to do more advanced comparisons. As this work consisted of solely investigating single, thin elements for a single pass, there are numerous natural extensions to this. A natural extension would be to investigate single, thick elements, and another would be to include the tracking code PTC [5] as a point of reference.

Bibliography

[1] M. Fjellstrom R. De Maria, M. Fitterer and A. Patapenka. *SixTrack Physics Manual (Draft)*. CERN BE/ABP, August 20 2018. Available at: http://sixtrack.web.cern.ch/SixTrack/docs/physics_manual.pdf. vii, 1, 4

[2] Ghislain Roy Laurent Deniau, Hans Grote and Frank Schmidt. *The MAD-X Program (Methodical Accelerator Design): User's Reference Manual*. CERN BE/ABP, 5.04.01 edition, July 2018. Available at: http://madx.web.cern.ch/madx/releases/last-rel/madxguide.pdf. 2, 4

[3] F. Schmidt et al. *SixTrack: User's Reference Manual*. CERN BE/ABP, 5.0 edition, March 2018. Available at: http://sixtrack.web.cern.ch/SixTrack/docs/user_dev_manual.pdf. 2, 3

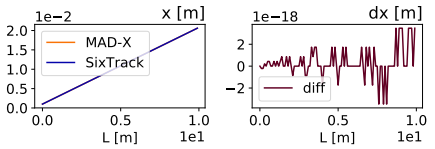
[4] H Grote. A MAD-Sixtrack Interface. Technical Report SL-Note-97-02-AP, CERN, Geneva, Jan 1997. 3, 7

[5] Etienne Forest, Frank Schmidt, and Eric McIntosh. Introduction to the Polymorphic Tracking Code. Technical Report KEK Report 2002-3, CERN-SL-2002-044-AP, KEK, CERN, July 2002. http://ccdb4fs.kek.jp/cgi-bin/img/allpdf?200302020, http://cern.ch/madx/doc/sl-2002-044.pdf. 30

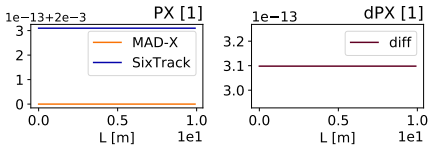
A. Graphical output

In this appendix all the graphical output of the different testing scenarios is contained for reference. For the specifics of each testing scenario, see Section 3.2.

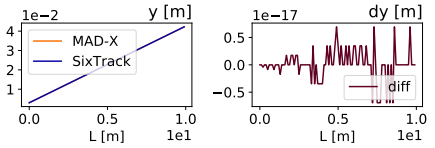
A.1 Drift



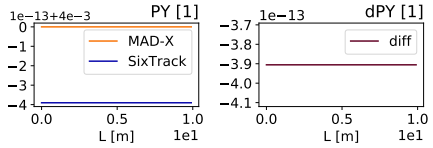
(a) X vs. drift length



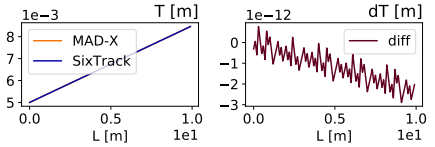
(b) PX vs. drift length



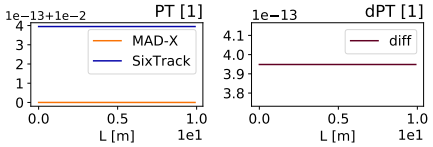
(c) Y vs. drift length



(d) PY vs. drift length



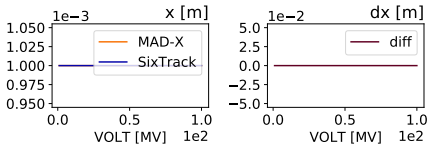
(e) T vs. drift length



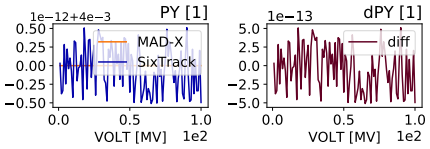
(f) PT vs. drift length

Figure A.1: Comparison output for the Drift element when varying L.

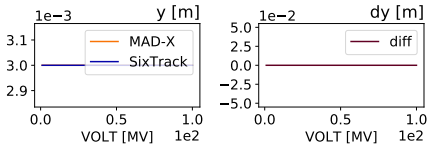
A.2 RF Cavity



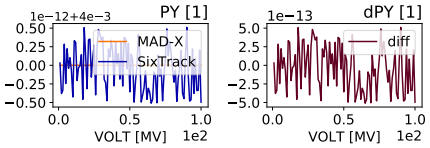
(a) X vs. voltage



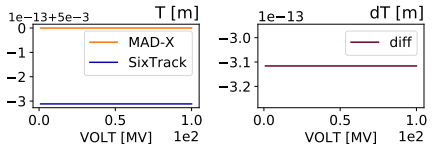
(b) PX vs. voltage



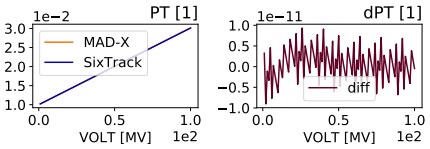
(c) Y vs. voltage



(d) PY vs. voltage

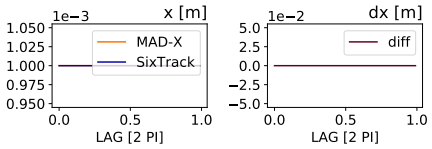


(e) T vs. voltage

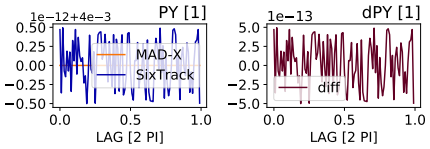


(f) PT vs. voltage

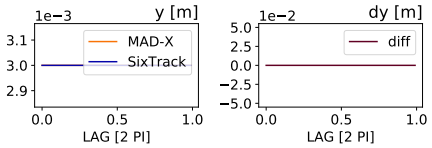
Figure A.2: Comparison output for the RF Cavity element when varying VOLT.



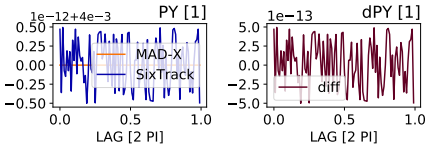
(a) X vs. lag



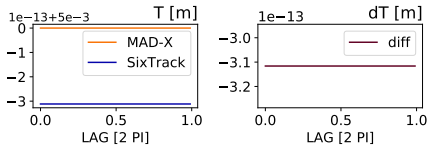
(b) PX vs. lag



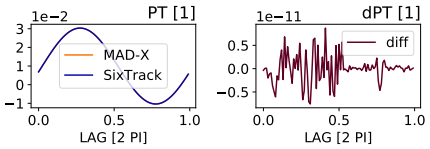
(c) Y vs. lag



(d) PY vs. lag

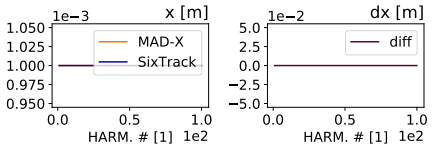


(e) T vs. lag

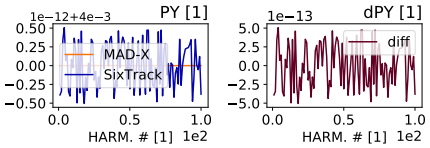


(f) PT vs. lag

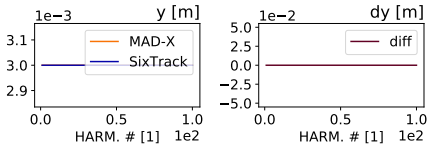
Figure A.3: Comparison output for the RF Cavity element when varying LAG.



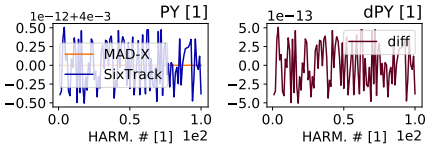
(a) X vs. harmonic number



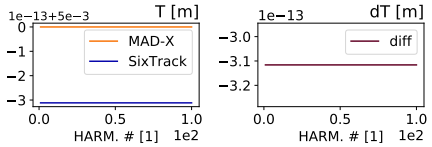
(b) PX vs. harmonic number



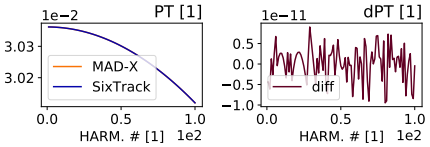
(c) Y vs. harmonic number



(d) PY vs. harmonic number



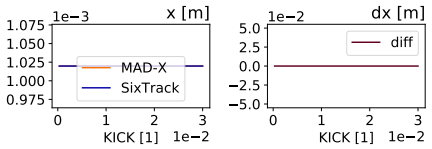
(e) T vs. harmonic number



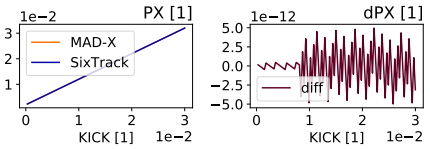
(f) PT vs. harmonic number

Figure A.4: Comparison output for the RF Cavity element when varying HARM.

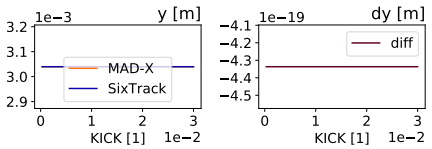
A.3 Kicker



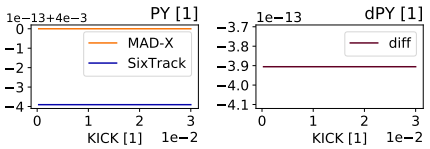
(a) X vs. horizontal kick



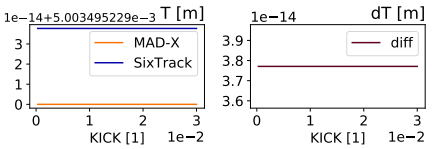
(b) PX vs. horizontal kick



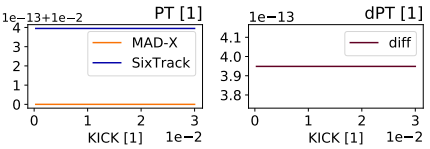
(c) Y vs. horizontal kick



(d) PY vs. horizontal kick

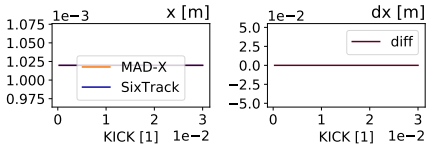


(e) T vs. horizontal kick

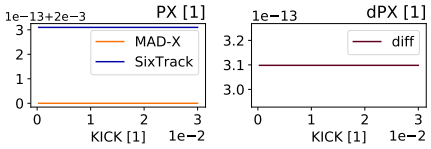


(f) PT vs. horizontal kick

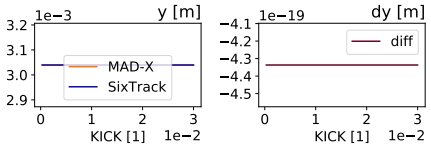
Figure A.5: Comparison output for the Kicker element when varying HKICK.



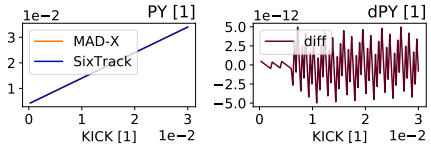
(a) X vs. vertical kick



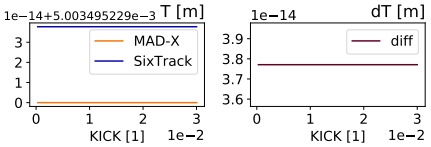
(b) PX vs. vertical kick



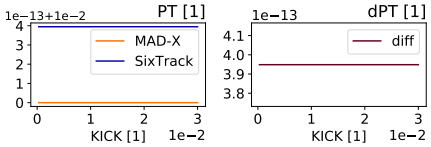
(c) Y vs. vertical kick



(d) PY vs. vertical kick

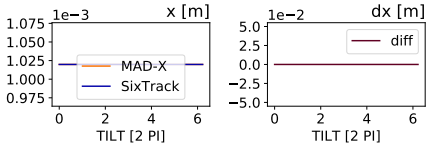


(e) T vs. vertical kick

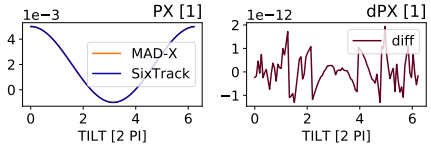


(f) PT vs. vertical kick

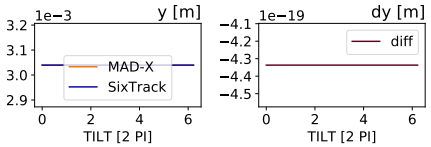
Figure A.6: Comparison output for the Kicker element when varying VKICK.



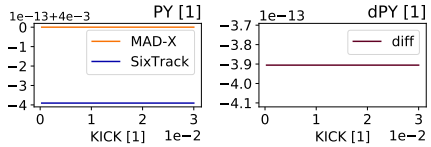
(a) X vs. tilt



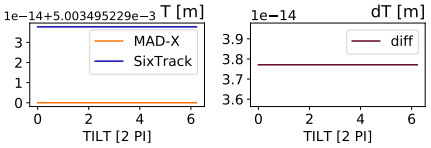
(b) PX vs. tilt



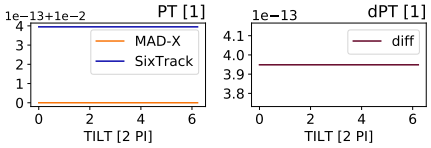
(c) Y vs. tilt



(d) PY vs. tilt



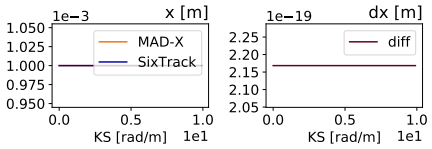
(e) T vs. tilt



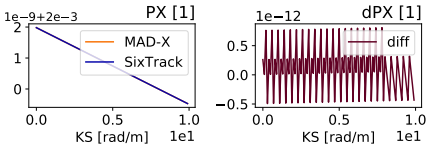
(f) PT vs. tilt

Figure A.7: Comparison output for the Kicker element when varying TILT for horizontal kick.

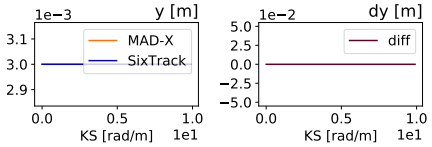
A.4 Solenoid Before Bugfix



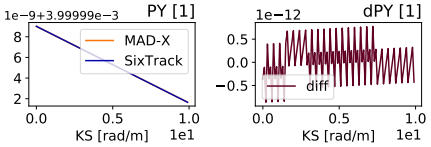
(a) X vs. KS



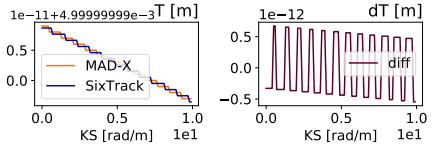
(b) PX vs. KS



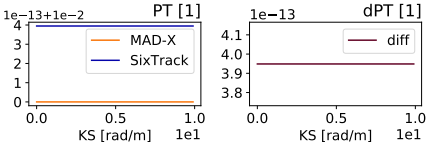
(c) Y vs. KS



(d) PY vs. KS

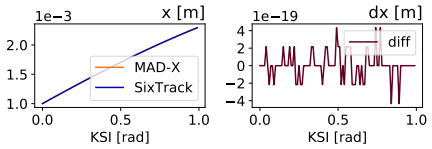


(e) T vs. KSI

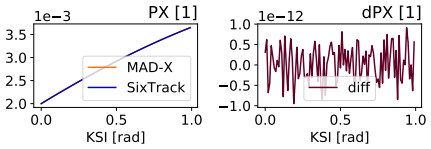


(f) PT vs. KSI

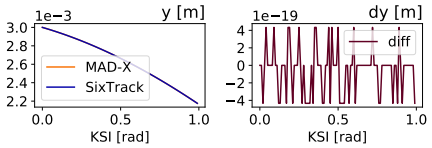
Figure A.8: Comparison output for the Solenoid element when varying KS.



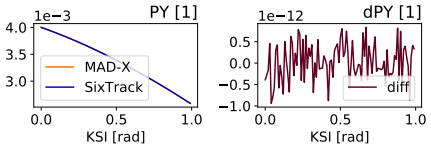
(a) X vs. KSI



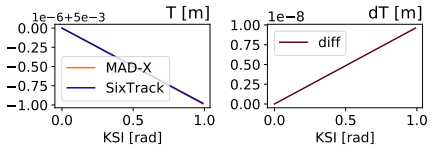
(b) PX vs. KSI



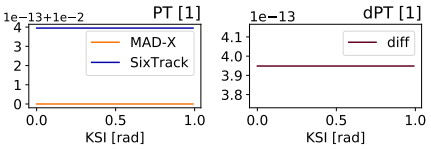
(c) Y vs. KSI



(d) PY vs. KSI

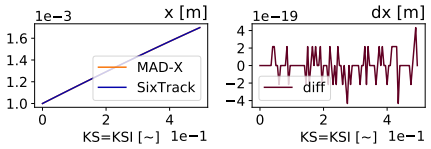


(e) T vs. KSI

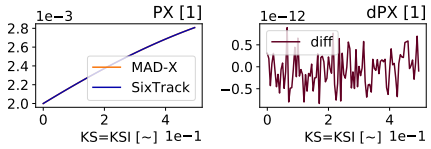


(f) PT vs. KSI

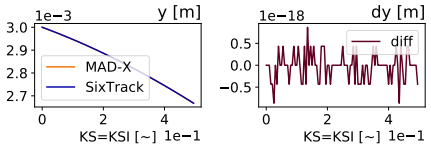
Figure A.9: Comparison output for the Solenoid element when varying KSI.



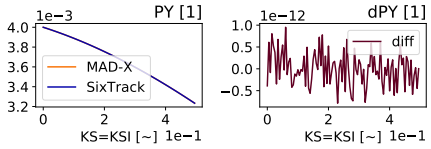
(a) X vs. KS and KSI



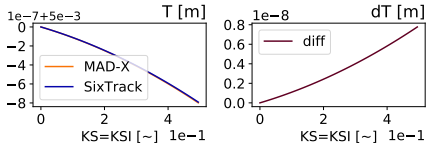
(b) PX vs. KS and KSI



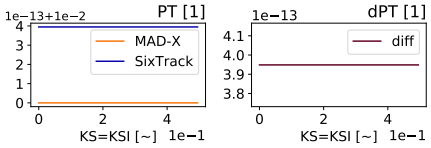
(c) Y vs. KS and KSI



(d) PY vs. KS and KSI



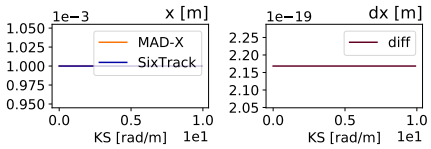
(e) T vs. KS and KSI



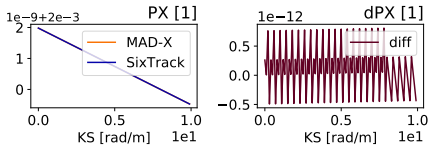
(f) PT vs. KS and KSI

Figure A.10: Comparison output for the Solenoid element when varying KS and KSI.

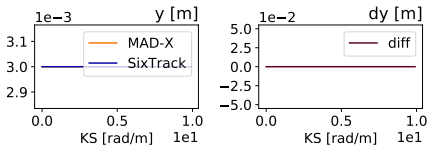
A.5 Solenoid After Bugfix



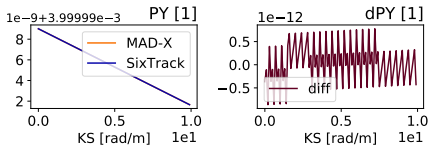
(a) X vs. KS



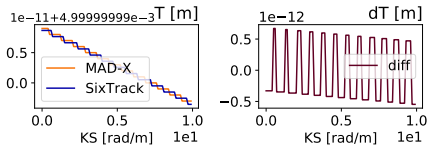
(b) PX vs. KS



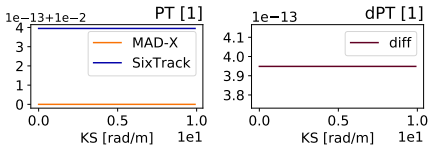
(c) Y vs. KS



(d) PY vs. KS

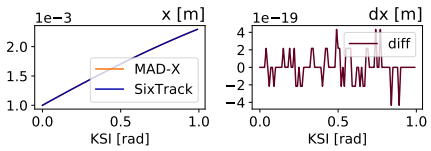


(e) T vs. KSI

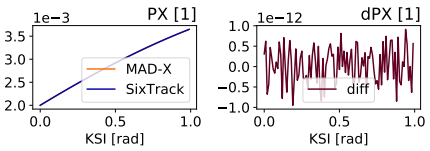


(f) PT vs. KSI

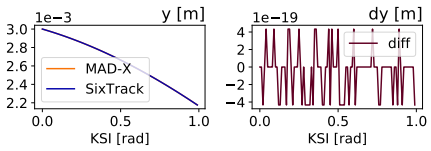
Figure A.11: Comparison output for the Solenoid element when varying KS.



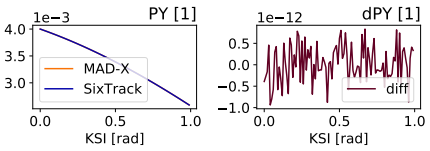
(a) X vs. KSI



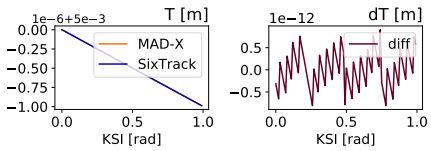
(b) PX vs. KSI



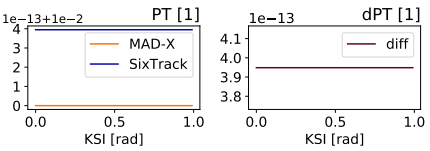
(c) Y vs. KSI



(d) PY vs. KSI

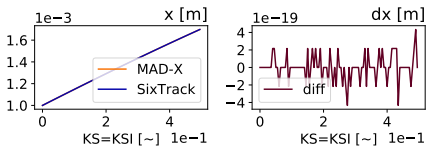


(e) T vs. KSI

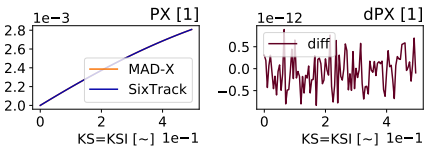


(f) PT vs. KSI

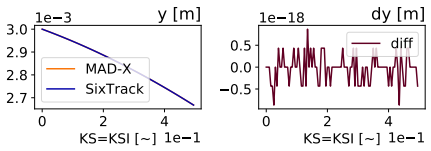
Figure A.12: Comparison output for the Solenoid element when varying KSI.



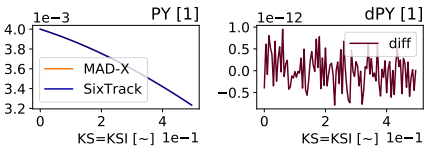
(a) X vs. KS and KSI



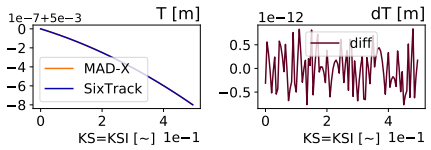
(b) PX vs. KS and KSI



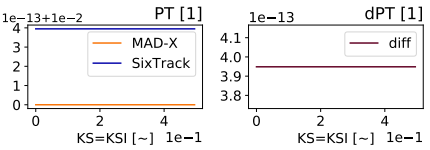
(c) Y vs. KS and KSI



(d) PY vs. KS and KSI



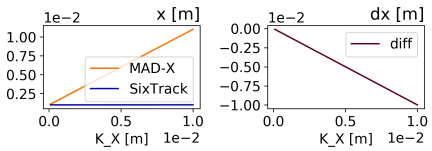
(e) T vs. KS and KSI



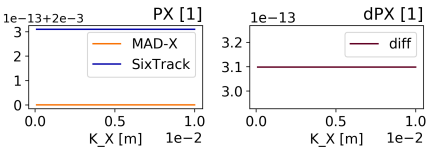
(f) PT vs. KS and KSI

Figure A.13: Comparison output for the Solenoid element when varying KS and KSI.

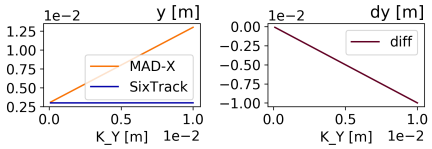
A.6 Trombone Before Bugfix



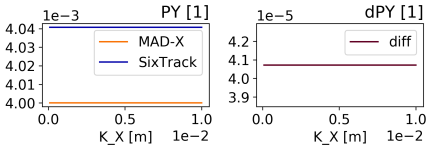
(a) X vs. kick in X



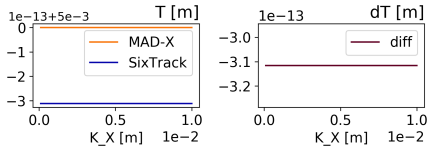
(b) PX vs. kick in X



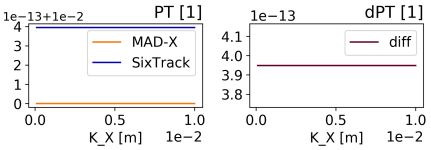
(c) Y vs. kick in X



(d) PY vs. kick in X

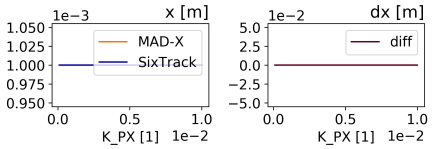


(e) T vs. kick in X

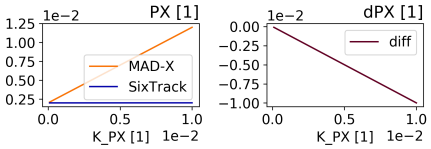


(f) PT vs. kick in X

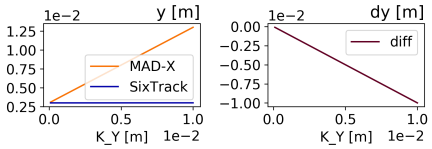
Figure A.14: Comparison output for the broken Trombone element when varying a kick in X.



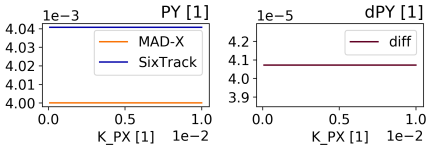
(a) X vs. kick in PX



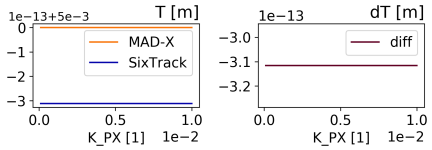
(b) PX vs. kick in PX



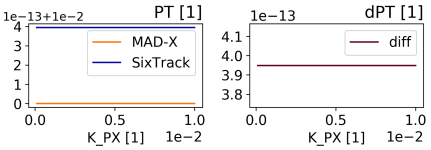
(c) Y vs. kick in PX



(d) PY vs. kick in PX

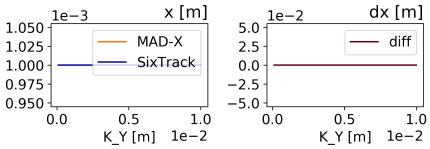


(e) T vs. kick in PX

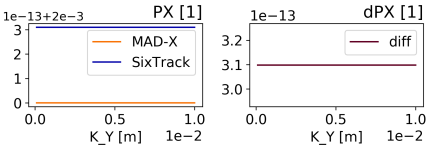


(f) PT vs. kick in PX

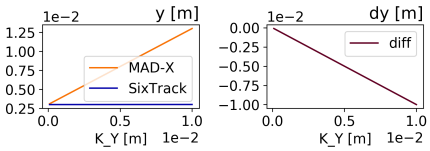
Figure A.15: Comparison output for the broken Trombone element when varying a kick in PX.



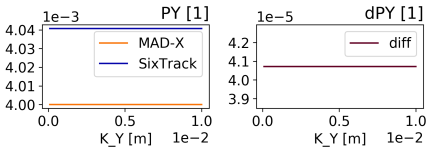
(a) X vs. kick in Y



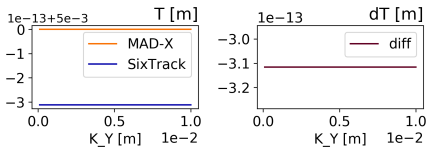
(b) PX vs. kick in Y



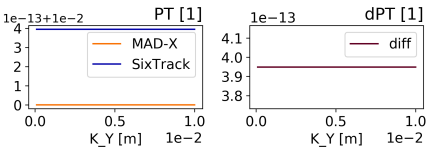
(c) Y vs. kick in Y



(d) PY vs. kick in Y

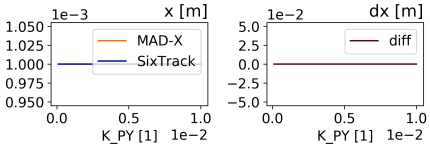


(e) T vs. kick in Y

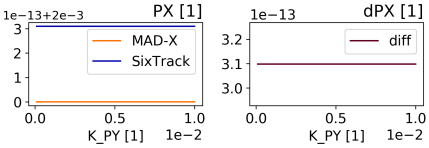


(f) PT vs. kick in Y

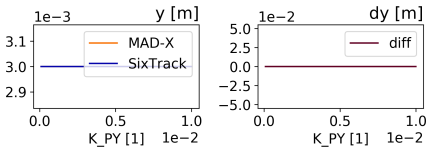
Figure A.16: Comparison output for the broken Trombone element when varying a kick in Y.



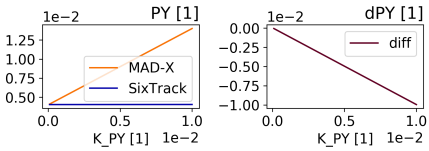
(a) X vs. kick in PY



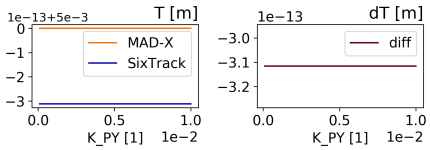
(b) PX vs. kick in PY



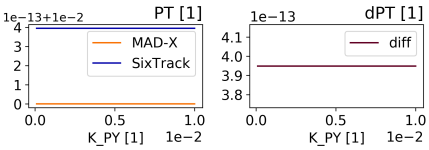
(c) Y vs. kick in PY



(d) PY vs. kick in PY

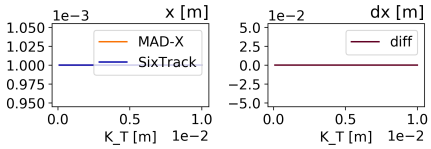


(e) T vs. kick in PY

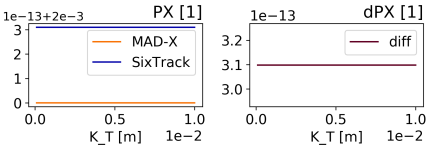


(f) PT vs. kick in PY

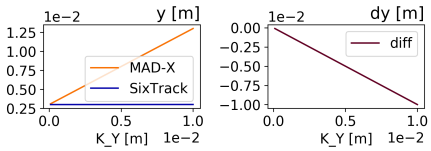
Figure A.17: Comparison output for the broken Trombone element when varying a kick in PY.



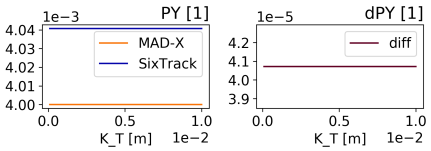
(a) X vs. kick in T



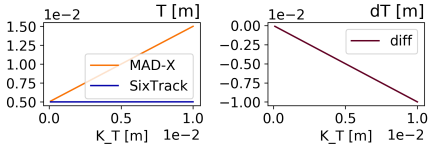
(b) PX vs. kick in T



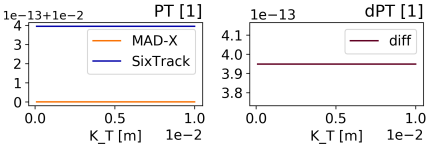
(c) Y vs. kick in T



(d) PY vs. kick in T

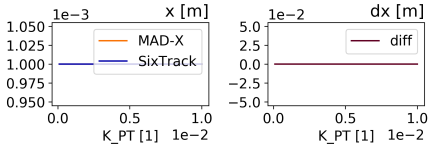


(e) T vs. kick in T

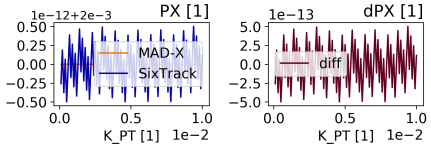


(f) PT vs. kick in T

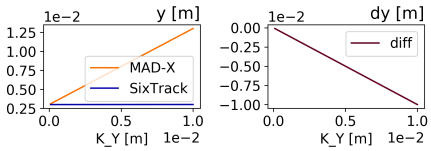
Figure A.18: Comparison output for the broken Trombone element when varying a kick in T.



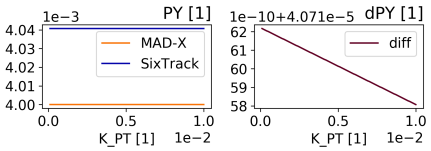
(a) X vs. kick in PT



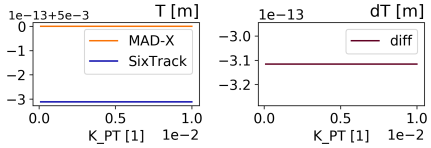
(b) PX vs. kick in PT



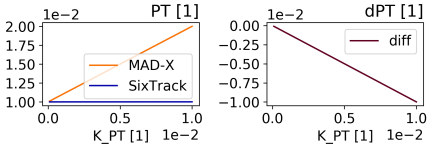
(c) Y vs. kick in PT



(d) PY vs. kick in PT



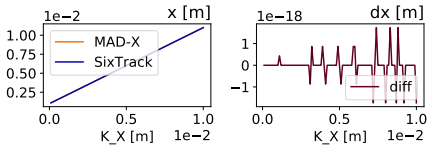
(e) T vs. kick in PT



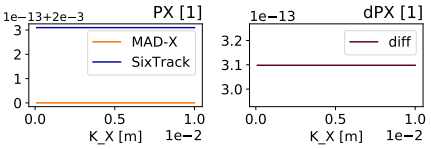
(f) PT vs. kick in PT

Figure A.19: Comparison output for the broken Trombone element when varying a kick in PT.

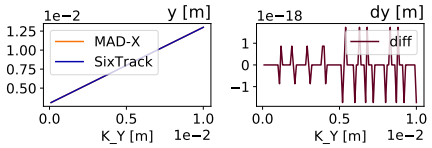
A.7 Trombone After Bugfix



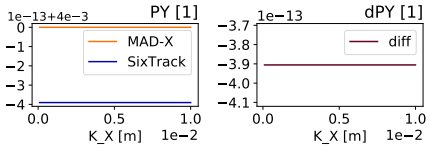
(a) X vs. kick in X



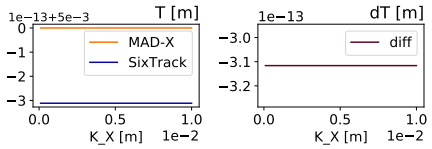
(b) PX vs. kick in X



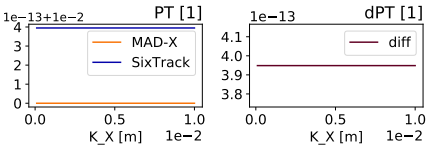
(c) Y vs. kick in X



(d) PY vs. kick in X

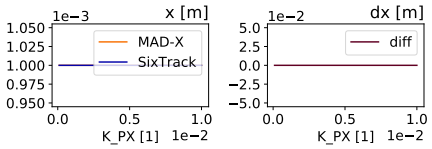


(e) T vs. kick in X

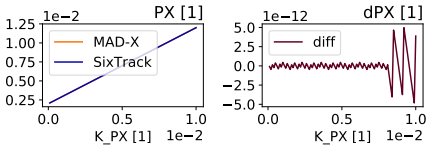


(f) PT vs. kick in X

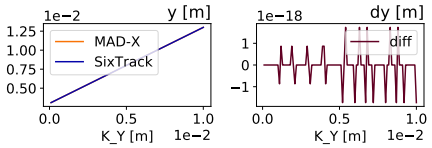
Figure A.20: Comparison output for the fixed Trombone element when varying a kick in X.



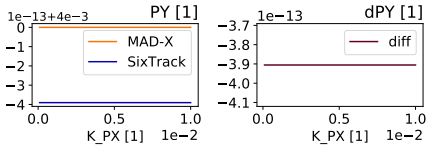
(a) X vs. kick in PX



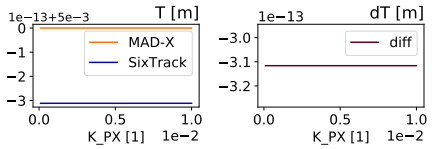
(b) PX vs. kick in PX



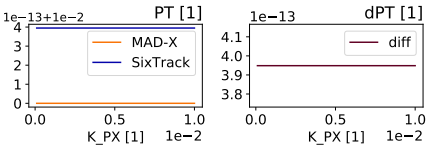
(c) Y vs. kick in PX



(d) PY vs. kick in PX

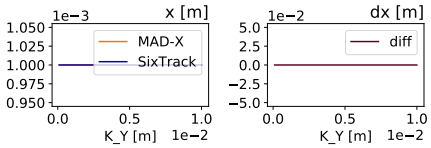


(e) T vs. kick in PX

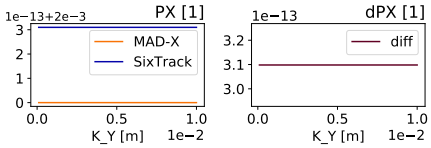


(f) PT vs. kick in PX

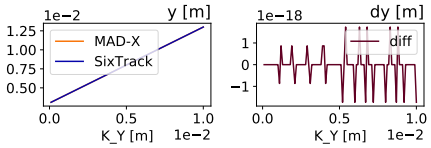
Figure A.21: Comparison output for the fixed Trombone element when varying a kick in PX.



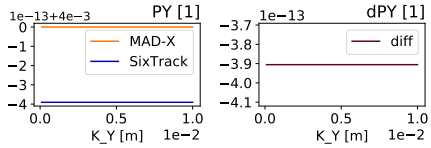
(a) X vs. kick in Y



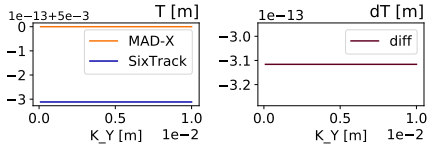
(b) PX vs. kick in Y



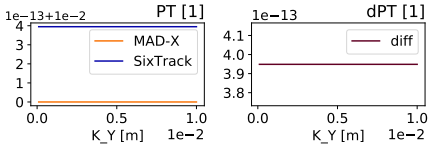
(c) Y vs. kick in Y



(d) PY vs. kick in Y

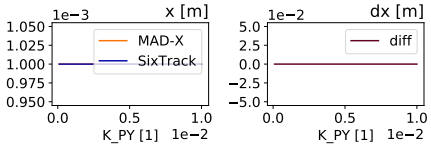


(e) T vs. kick in Y

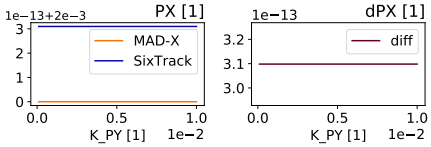


(f) PT vs. kick in Y

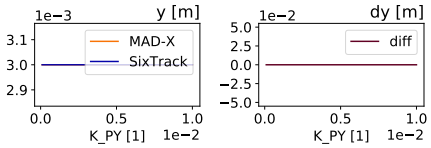
Figure A.22: Comparison output for the fixed Trombone element when varying a kick in Y.



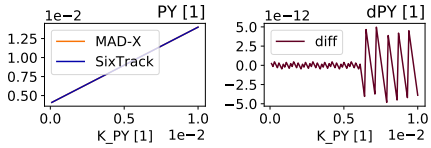
(a) X vs. kick in PY



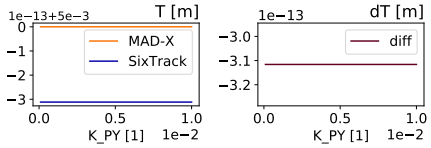
(b) PX vs. kick in PY



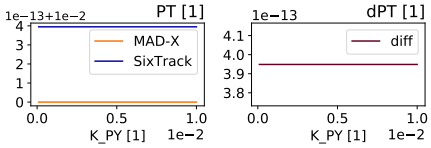
(c) Y vs. kick in PY



(d) PY vs. kick in PY

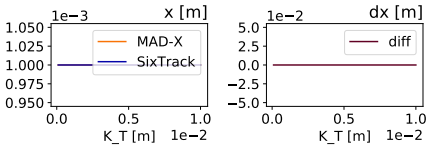


(e) T vs. kick in PY

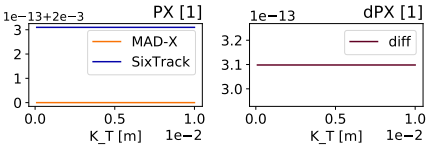


(f) PT vs. kick in PY

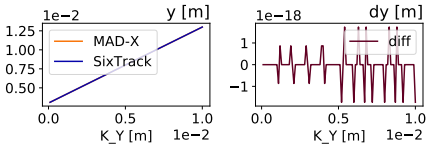
Figure A.23: Comparison output for the fixed Trombone element when varying a kick in PY.



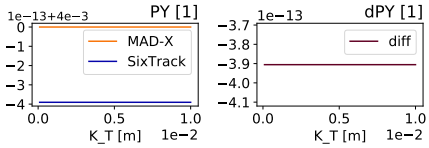
(a) X vs. kick in T



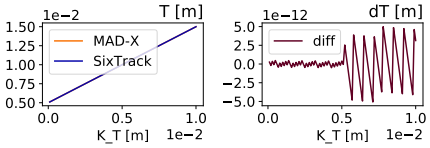
(b) PX vs. kick in T



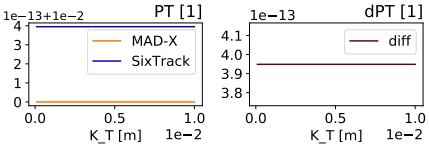
(c) Y vs. kick in T



(d) PY vs. kick in T

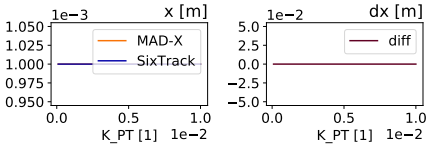


(e) T vs. kick in T

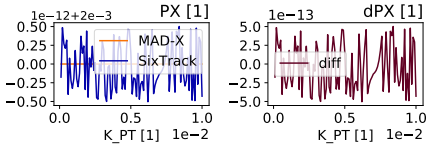


(f) PT vs. kick in T

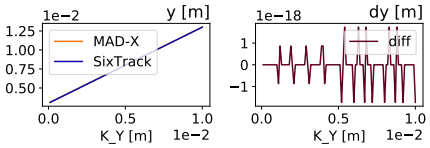
Figure A.24: Comparison output for the fixed Trombone element when varying a kick in T.



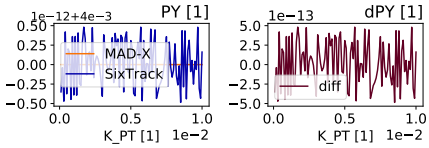
(a) X vs. kick in PT



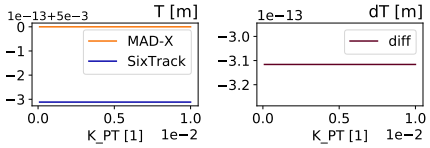
(b) PX vs. kick in PT



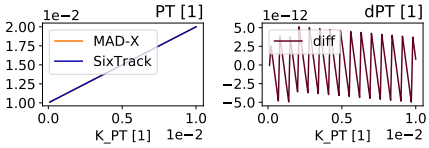
(c) Y vs. kick in PT



(d) PY vs. kick in PT



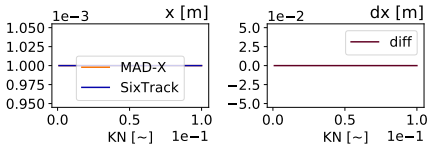
(e) T vs. kick in PT



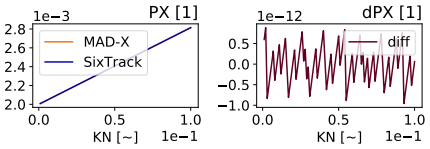
(f) PT vs. kick in PT

Figure A.25: Comparison output for the fixed Trombone element when varying a kick in PT.

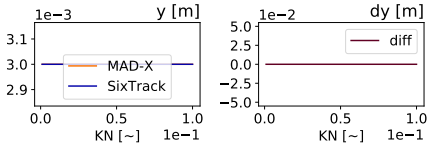
A.8 Dipole



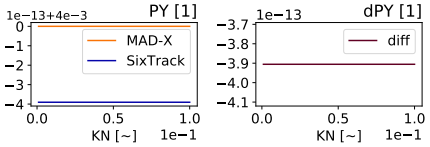
(a) X vs. normal strength



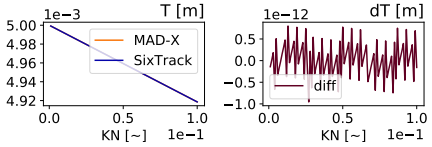
(b) PX vs. normal strength



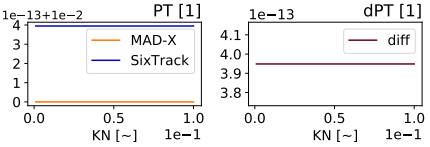
(c) Y vs. normal strength



(d) PY vs. normal strength

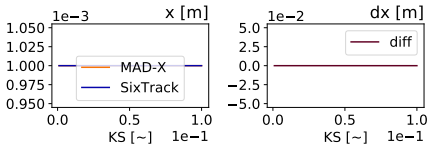


(e) T vs. normal strength

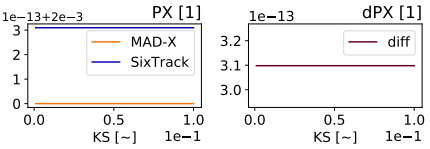


(f) PT vs. normal strength

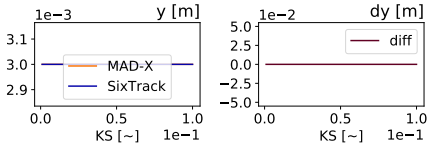
Figure A.26: Comparison output for the Dipole element when varying KN.



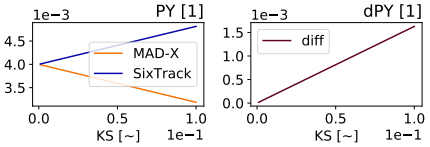
(a) X vs. skewed strength



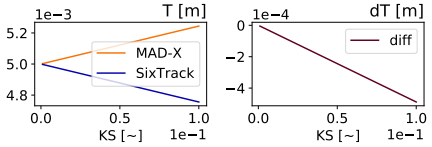
(b) PX vs. skewed strength



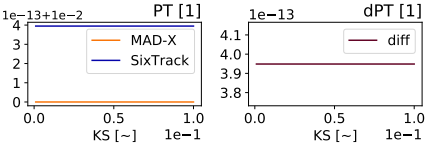
(c) Y vs. skewed strength



(d) PY vs. skewed strength

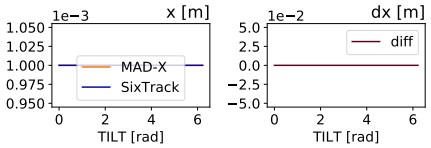


(e) T vs. skewed strength

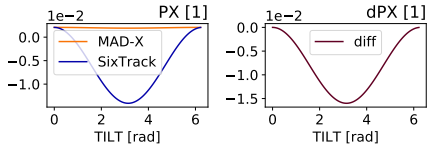


(f) PT vs. skewed strength

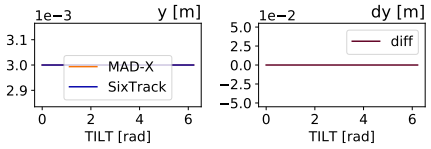
Figure A.27: Comparison output for the Dipole element when varying KS.



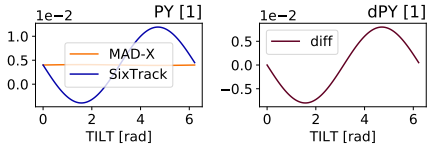
(a) X vs. TILT for normal Dipole



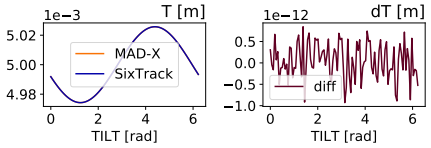
(b) PX vs. TILT for normal Dipole



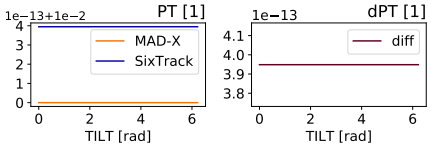
(c) Y vs. TILT for normal Dipole



(d) PY vs. TILT for normal Dipole

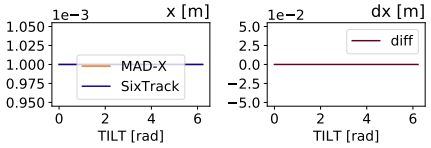


(e) T vs. TILT for normal Dipole

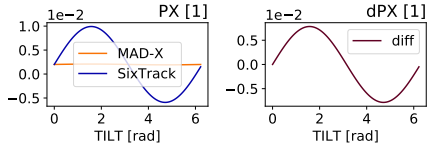


(f) PT vs. TILT for normal Dipole

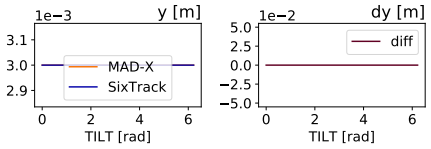
Figure A.28: Comparison output for the Dipole element when varying TILT for normal Dipole.



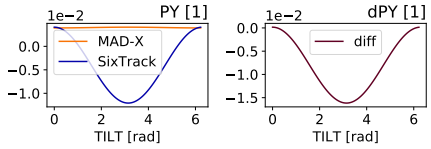
(a) X vs. TILT for skewed Dipole



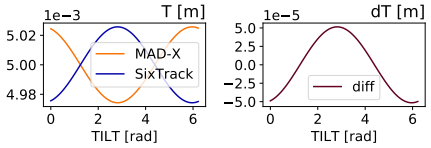
(b) PX vs. TILT for skewed Dipole



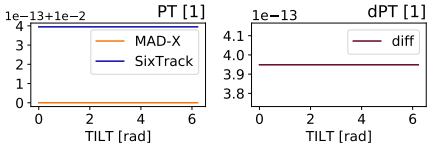
(c) Y vs. TILT for skewed Dipole



(d) PY vs. TILT for skewed Dipole



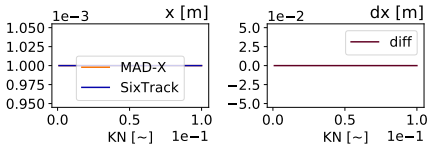
(e) T vs. TILT for skewed Dipole



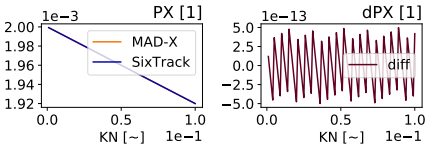
(f) PT vs. TILT for skewed Dipole

Figure A.29: Comparison output for the Dipole element when varying TILT for skewed Dipole.

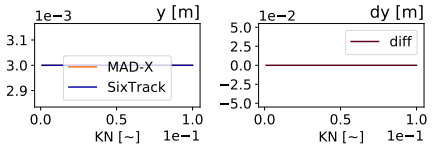
A.9 Quadrupole



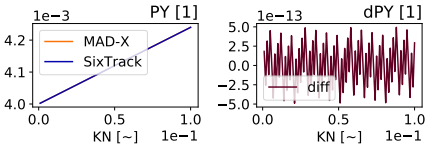
(a) X vs. normal strength



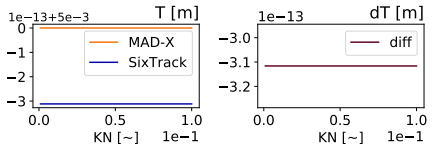
(b) PX vs. normal strength



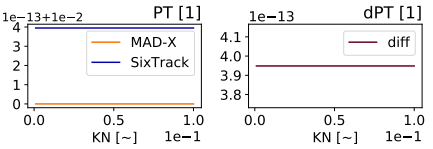
(c) Y vs. normal strength



(d) PY vs. normal strength

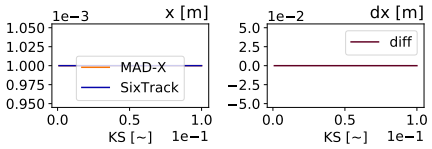


(e) T vs. normal strength

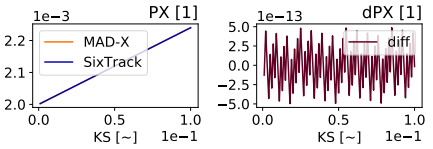


(f) PT vs. normal strength

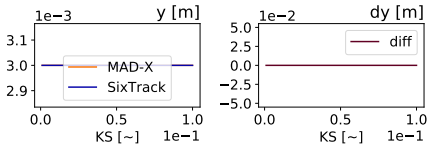
Figure A.30: Comparison output for the Quadrupole element when varying KN.



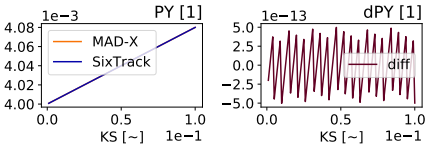
(a) X vs. skewed strength



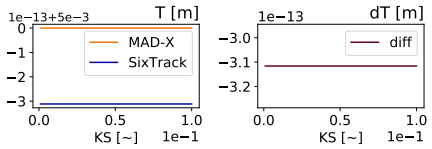
(b) PX vs. skewed strength



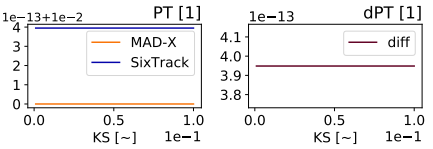
(c) Y vs. skewed strength



(d) PY vs. skewed strength

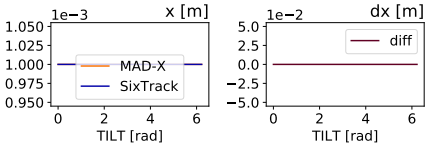


(e) T vs. skewed strength

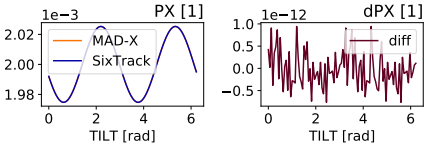


(f) PT vs. skewed strength

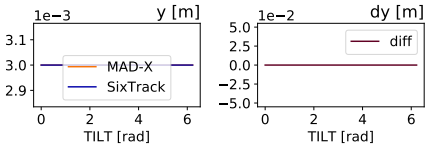
Figure A.31: Comparison output for the Quadrupole element when varying KS.



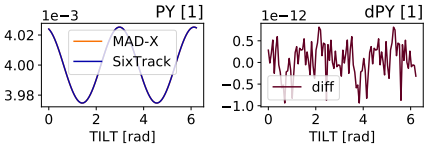
(a) X vs. TILT for normal v



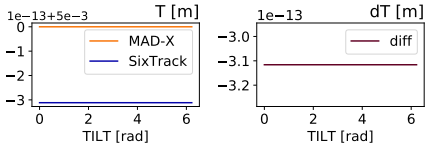
(b) PX vs. TILT for normal Quadrupole



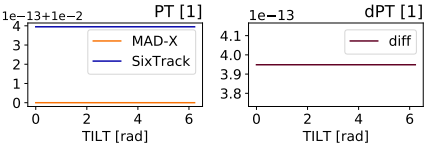
(c) Y vs. TILT for normal Quadrupole



(d) PY vs. TILT for normal Quadrupole

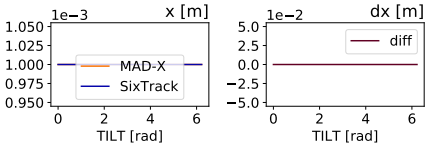


(e) T vs. TILT for normal Quadrupole

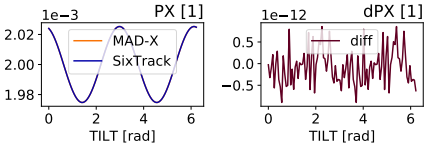


(f) PT vs. TILT for normal Quadrupole

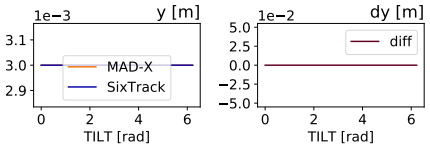
Figure A.32: Comparison output for the Quadrupole element when varying TILT for normal Dipole.



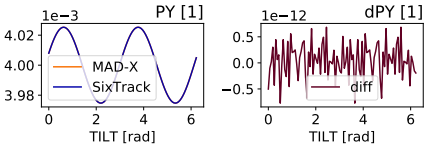
(a) X vs. TILT for skewed Quadrupole



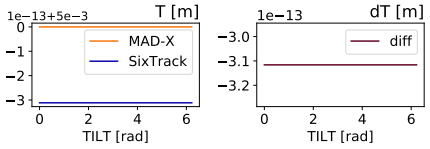
(b) PX vs. TILT for skewed Quadrupole



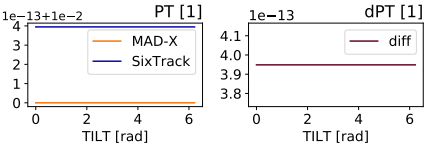
(c) Y vs. TILT for skewed Quadrupole



(d) PY vs. TILT for skewed Quadrupole



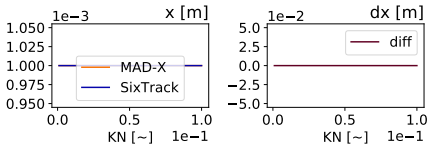
(e) T vs. TILT for skewed Quadrupole



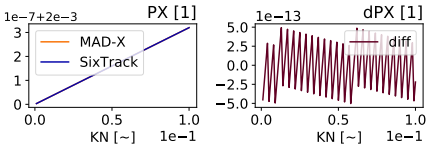
(f) PT vs. TILT for skewed Quadrupole

Figure A.33: Comparison output for the Quadrupole element when varying TILT for skewed Quadrupole.

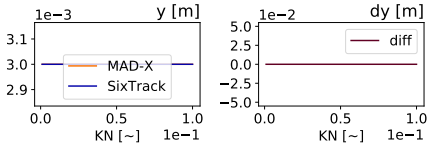
A.10 Sextupole



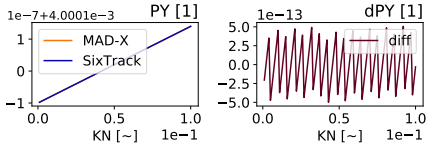
(a) X vs. normal strength



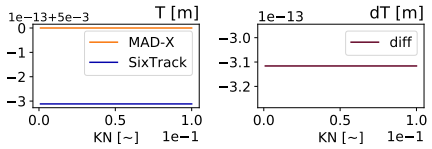
(b) PX vs. normal strength



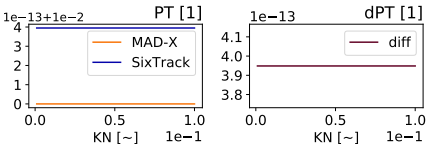
(c) Y vs. normal strength



(d) PY vs. normal strength

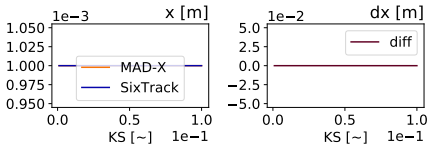


(e) T vs. normal strength

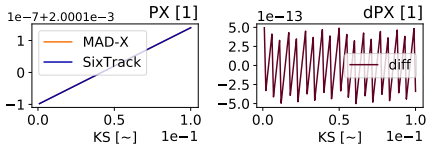


(f) PT vs. normal strength

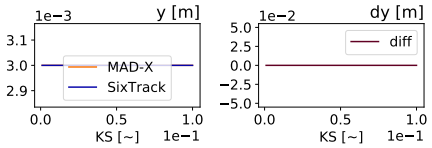
Figure A.34: Comparison output for the Sextupole element when varying KN.



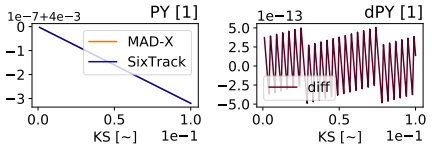
(a) X vs. skewed strength



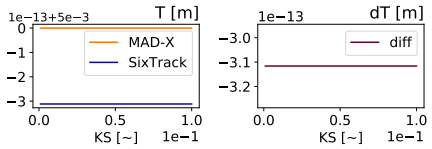
(b) PX vs. skewed strength



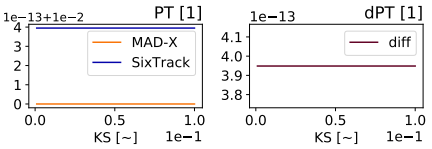
(c) Y vs. skewed strength



(d) PY vs. skewed strength

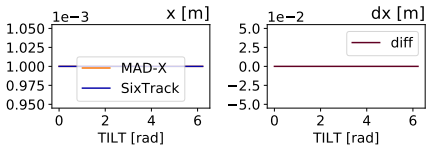


(e) T vs. skewed strength

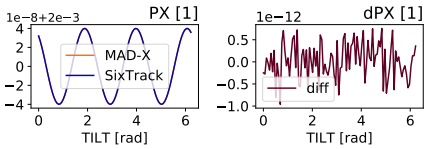


(f) PT vs. skewed strength

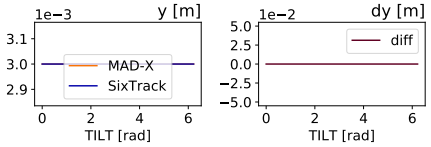
Figure A.35: Comparison output for the Sextupole element when varying KS.



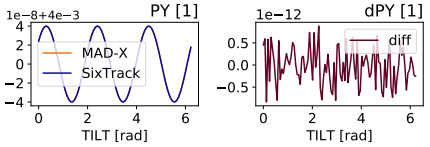
(a) X vs. TILT for normal Sextupole



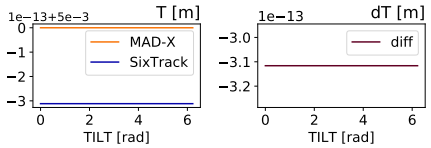
(b) PX vs. TILT for normal Sextupole



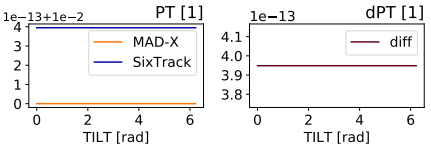
(c) Y vs. TILT for normal Sextupole



(d) PY vs. TILT for normal Sextupole

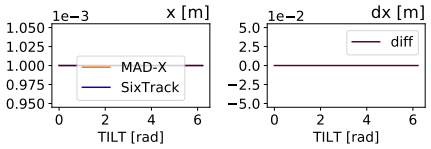


(e) T vs. TILT for normal Sextupole

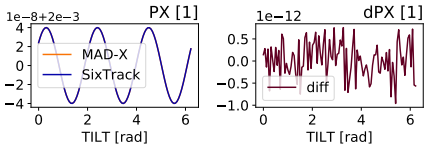


(f) PT vs. TILT for normal Sextupole

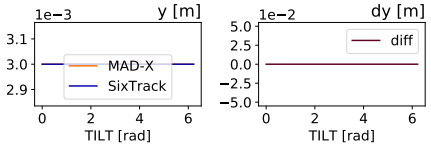
Figure A.36: Comparison output for the Sextupole element when varying TILT for normal Dipole.



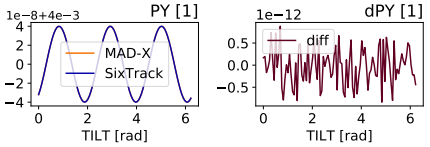
(a) X vs. TILT for skewed Sextupole



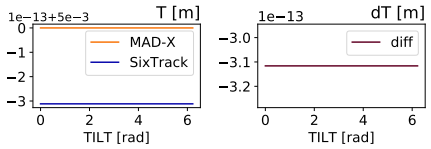
(b) PX vs. TILT for skewed Sextupole



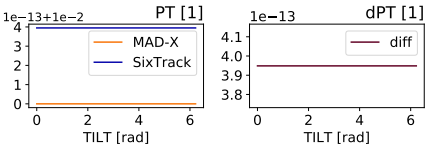
(c) Y vs. TILT for skewed Sextupole



(d) PY vs. TILT for skewed Sextupole



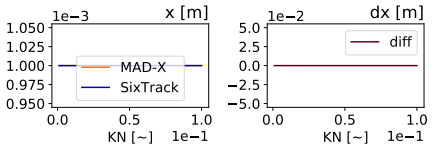
(e) T vs. TILT for skewed Sextupole



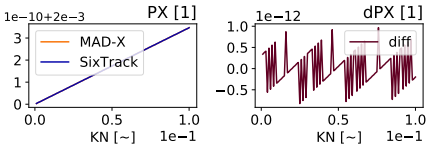
(f) PT vs. TILT for skewed Sextupole

Figure A.37: Comparison output for the Sextupole element when varying TILT for skewed Sextupole.

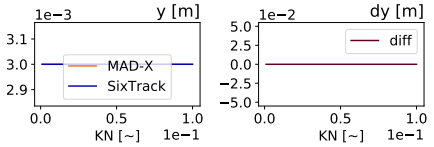
A.11 Octupole



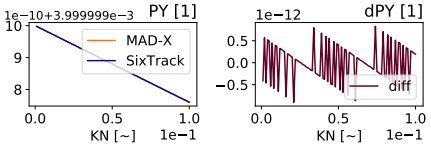
(a) X vs. normal strength



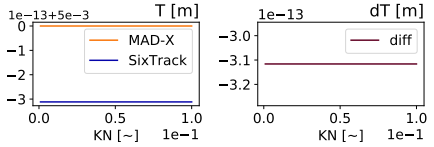
(b) PX vs. normal strength



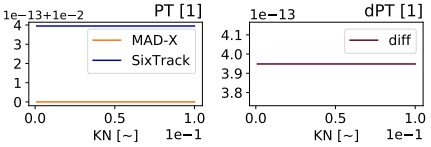
(c) Y vs. normal strength



(d) PY vs. normal strength

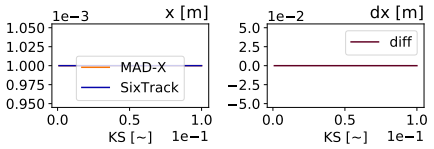


(e) T vs. normal strength

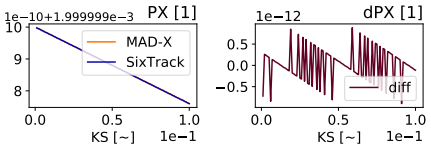


(f) PT vs. normal strength

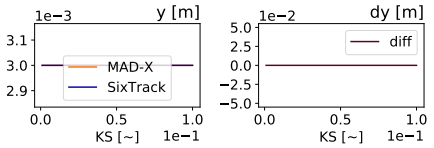
Figure A.38: Comparison output for the Octupole element when varying KN.



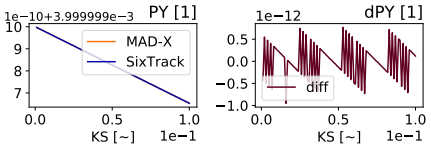
(a) X vs. skewed strength



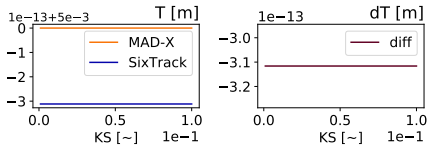
(b) PX vs. skewed strength



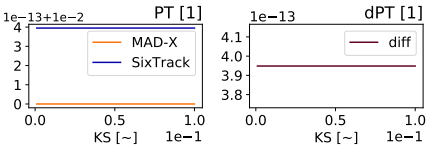
(c) Y vs. skewed strength



(d) PY vs. skewed strength

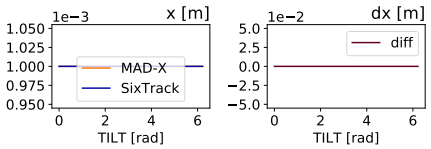


(e) T vs. skewed strength

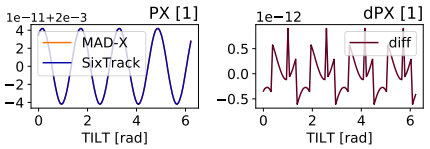


(f) PT vs. skewed strength

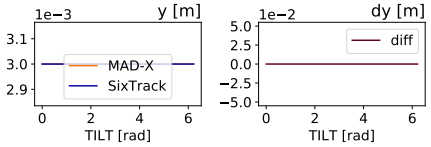
Figure A.39: Comparison output for the Octupole element when varying KS.



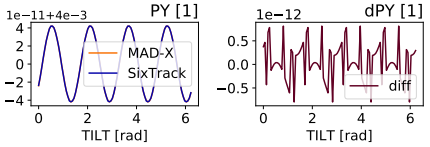
(a) X vs. TILT for normal Octupole



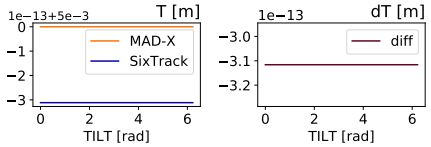
(b) PX vs. TILT for normal Octupole



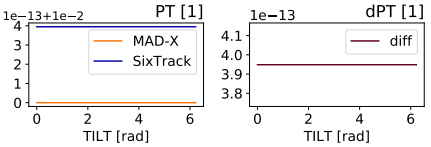
(c) Y vs. TILT for normal Octupole



(d) PY vs. TILT for normal Octupole

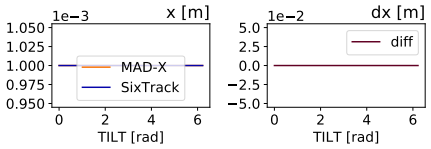


(e) T vs. TILT for normal Octupole

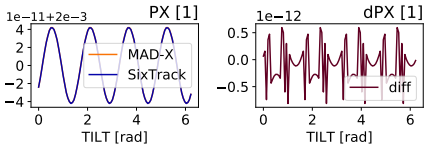


(f) PT vs. TILT for normal Octupole

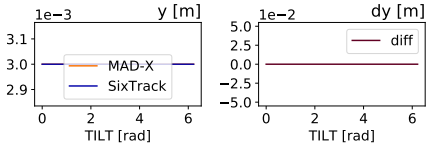
Figure A.40: Comparison output for the Octupole element when varying TILT for normal Dipole.



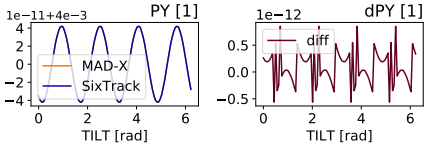
(a) X vs. TILT for skewed Octupole



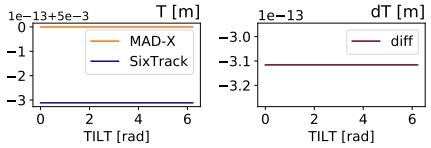
(b) PX vs. TILT for skewed Octupole



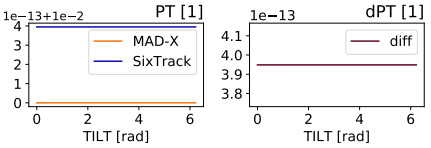
(c) Y vs. TILT for skewed Octupole



(d) PY vs. TILT for skewed Octupole



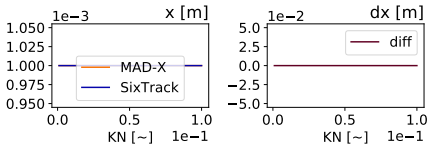
(e) T vs. TILT for skewed Octupole



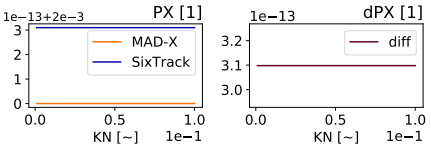
(f) PT vs. TILT for skewed Octupole

Figure A.41: Comparison output for the Octupole element when varying TILT for skewed Octupole.

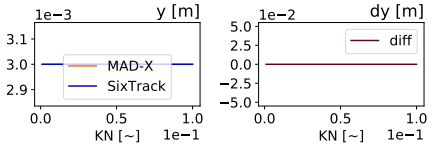
A.12 Decapole



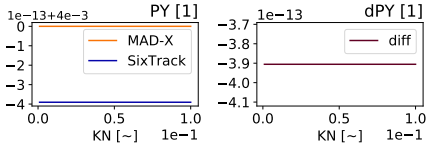
(a) X vs. normal strength



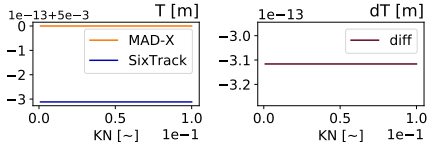
(b) PX vs. normal strength



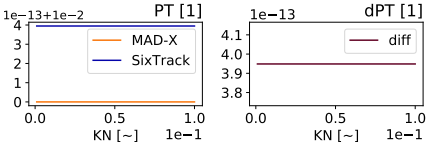
(c) Y vs. normal strength



(d) PY vs. normal strength

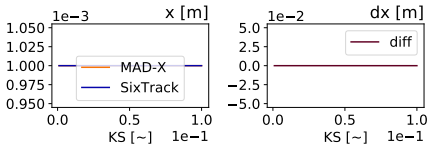


(e) T vs. normal strength

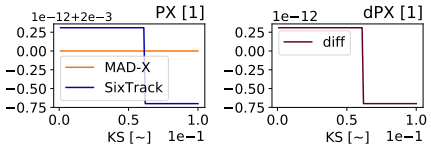


(f) PT vs. normal strength

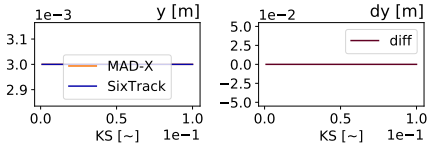
Figure A.42: Comparison output for the Decapole element when varying KN.



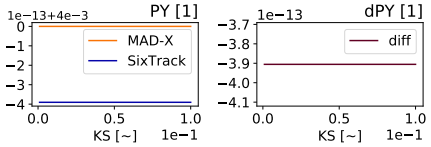
(a) X vs. skewed strength



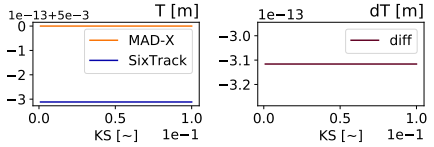
(b) PX vs. skewed strength



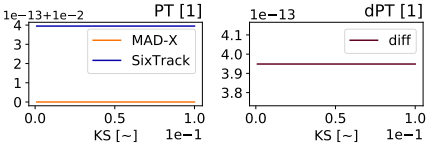
(c) Y vs. skewed strength



(d) PY vs. skewed strength

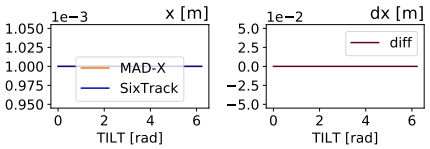


(e) T vs. skewed strength

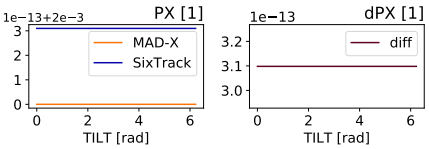


(f) PT vs. skewed strength

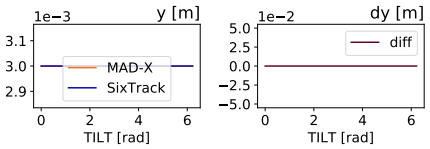
Figure A.43: Comparison output for the Decapole element when varying KS.



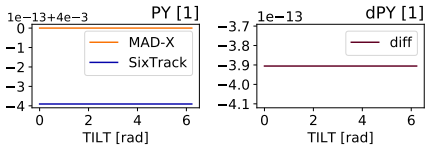
(a) X vs. TILT for normal Decapole



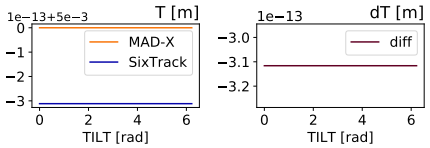
(b) PX vs. TILT for normal Decapole



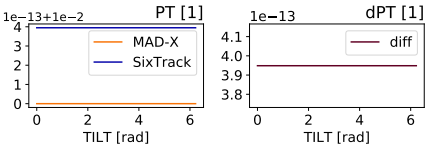
(c) Y vs. TILT for normal Decapole



(d) PY vs. TILT for normal Decapole

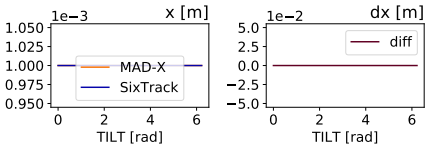


(e) T vs. TILT for normal Decapole

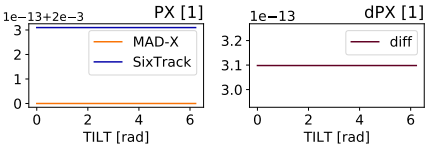


(f) PT vs. TILT for normal Decapole

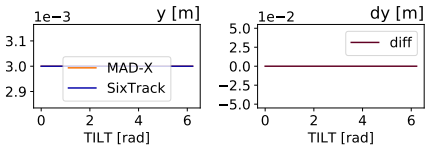
Figure A.44: Comparison output for the Decapole element when varying TILT for normal Dipole.



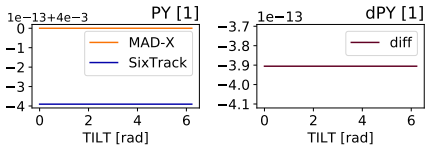
(a) X vs. TILT for skewed Decapole



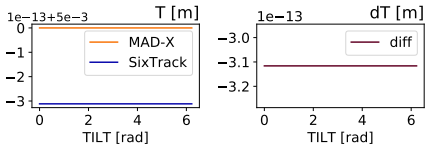
(b) PX vs. TILT for skewed Decapole



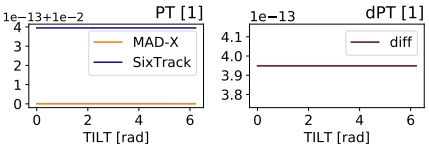
(c) Y vs. TILT for skewed Decapole



(d) PY vs. TILT for skewed Decapole



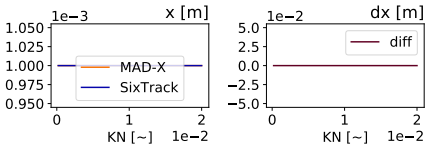
(e) T vs. TILT for skewed Decapole



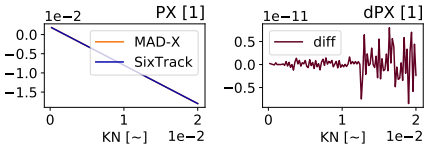
(f) PT vs. TILT for skewed Decapole

Figure A.45: Comparison output for the Decapole element when varying TILT for skewed Decapole.

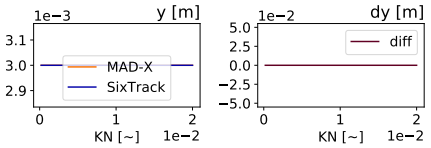
A.13 RF Dipole



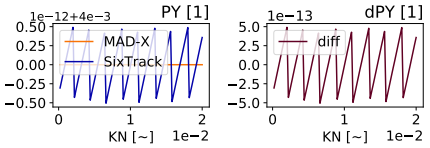
(a) X vs. normal strength



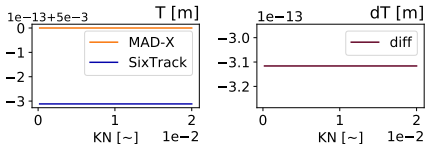
(b) PX vs. normal strength



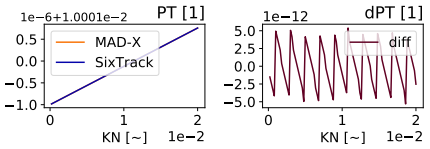
(c) Y vs. normal strength



(d) PY vs. normal strength

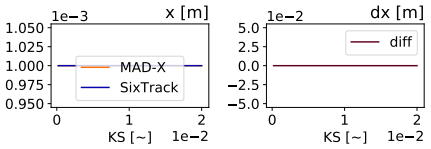


(e) T vs. normal strength

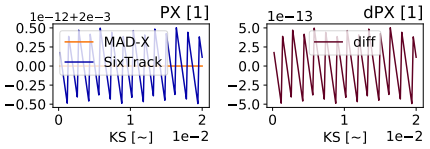


(f) PT vs. normal strength

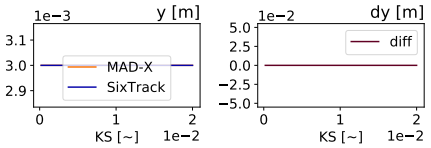
Figure A.46: Comparison output for the RF Dipole element when varying KN.



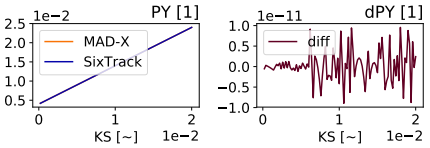
(a) X vs. skew strength



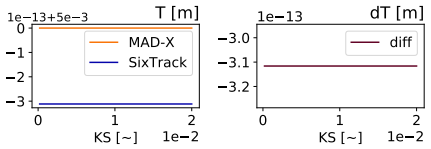
(b) PX vs. skew strength



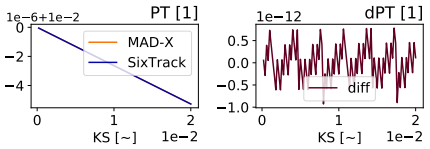
(c) Y vs. skew strength



(d) PY vs. skew strength

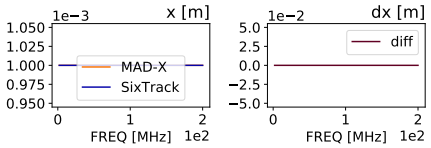


(e) T vs. skew strength

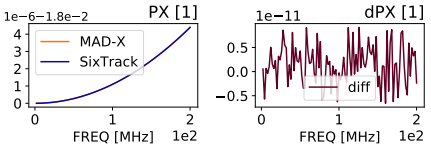


(f) PT vs. skew strength

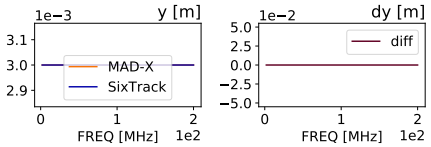
Figure A.47: Comparison output for the RF Dipole element when varying KS.



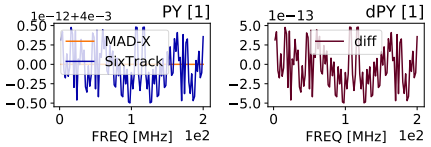
(a) X vs. frequency



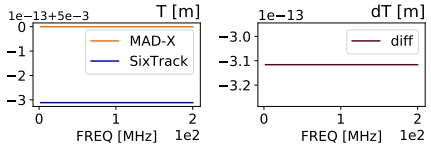
(b) PX vs. frequency



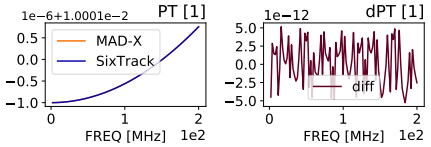
(c) Y vs. frequency



(d) PY vs. frequency

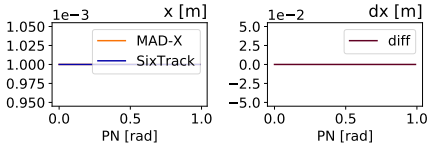


(e) T vs. frequency

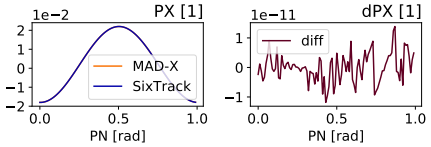


(f) PT vs. frequency

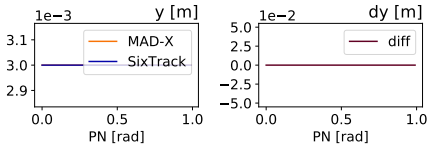
Figure A.48: Comparison output for the RF Dipole element when varying FREQ.



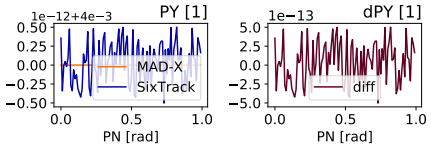
(a) X vs. normal phase



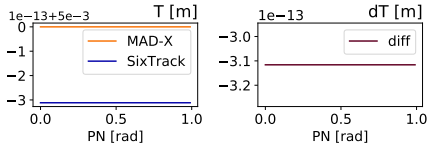
(b) PX vs. normal phase



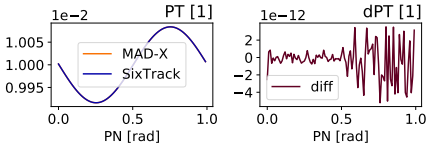
(c) Y vs. normal phase



(d) PY vs. normal phase

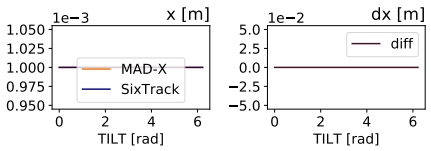


(e) T vs. normal phase

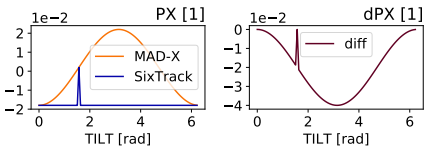


(f) PT vs. normal phase

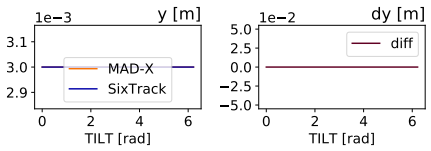
Figure A.49: Comparison output for the RF Dipole element when varying PN.



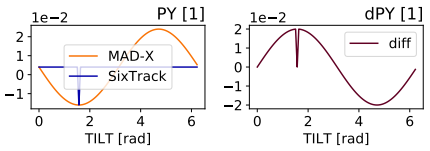
(a) X vs. tilt for normal RF Dipole



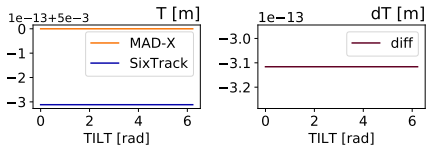
(b) PX vs. tilt for normal RF Dipole



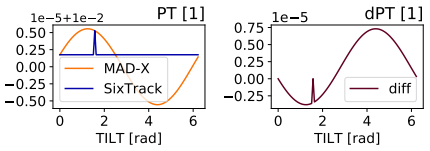
(c) Y vs. tilt for normal RF Dipole



(d) PY vs. tilt for normal RF Dipole

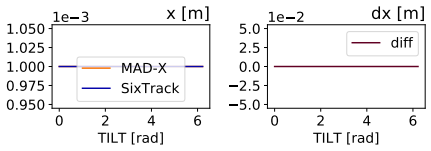


(e) T vs. tilt for normal RF Dipole

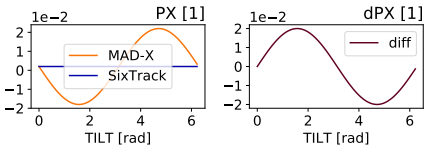


(f) PT vs. tilt for normal RF Dipole

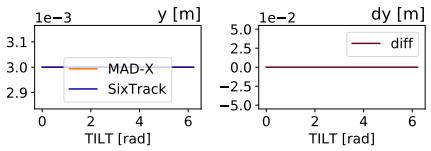
Figure A.50: Comparison output for the RF Dipole element when varying TILT for normal RF Dipole.



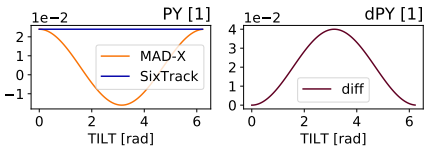
(a) X vs. tilt for skewed RF Dipole



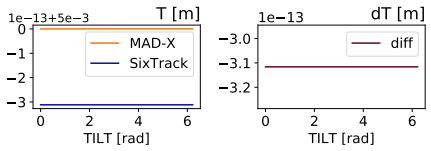
(b) PX vs. tilt for skewed RF Dipole



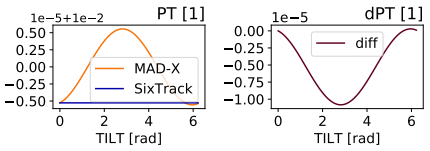
(c) Y vs. tilt for skewed RF Dipole



(d) PY vs. tilt for skewed RF Dipole

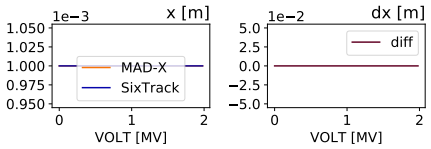


(e) T vs. tilt for skewed RF Dipole

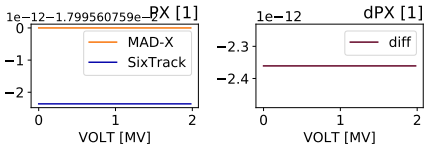


(f) PT vs. tilt for skewed RF Dipole

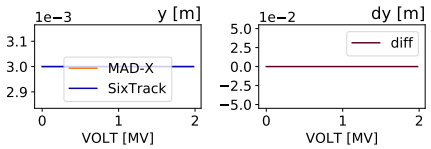
Figure A.51: Comparison output for the RF Dipole element when varying TILT for skewed RF Dipole.



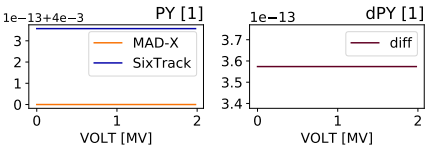
(a) `X` vs. `VOLT`



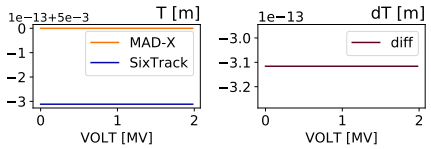
(b) `PX` vs. `VOLT`



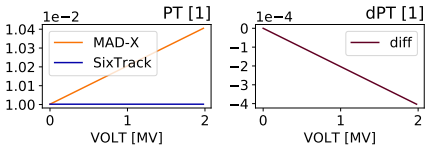
(c) `Y` vs. `VOLT`



(d) `PY` vs. `VOLT`



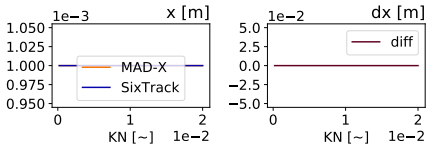
(e) `T` vs. `VOLT`



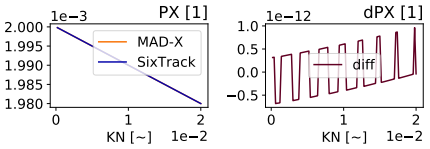
(f) `PT` vs. `VOLT`

Figure A.52: Comparison output for the RF Dipole element when varying `VOLT`.

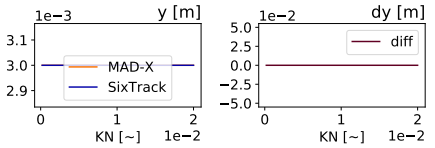
A.14 RF Quadrupole



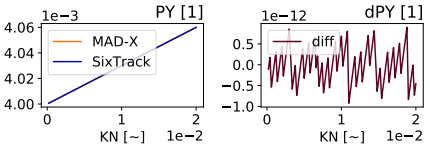
(a) X vs. normal strength



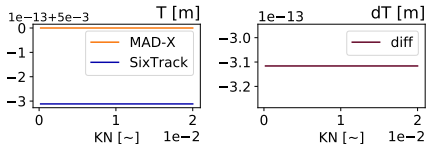
(b) PX vs. normal strength



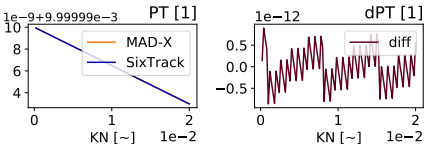
(c) Y vs. normal strength



(d) PY vs. normal strength

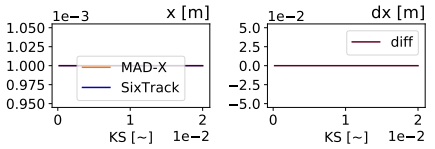


(e) T vs. normal strength

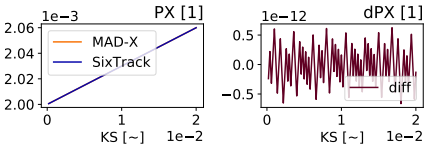


(f) PT vs. normal strength

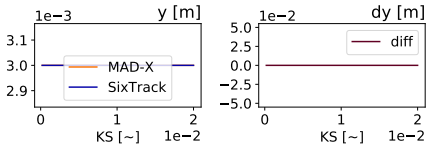
Figure A.53: Comparison output for the RF Quadrupole element when varying KN.



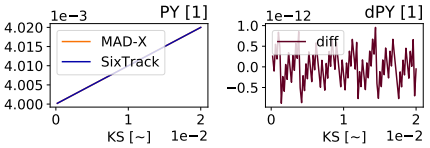
(a) X vs. skew strength



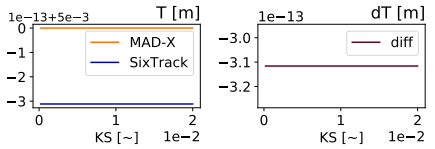
(b) PX vs. skew strength



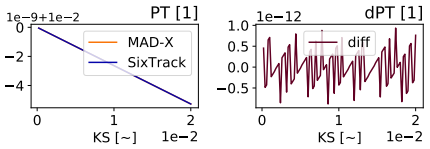
(c) Y vs. skew strength



(d) PY vs. skew strength

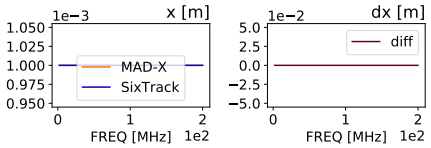


(e) T vs. skew strength

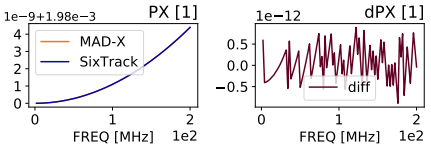


(f) PT vs. skew strength

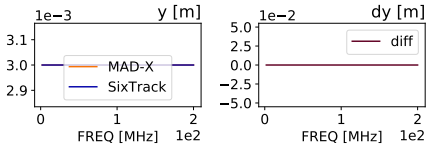
Figure A.54: Comparison output for the RF Quadrupole element when varying KS.



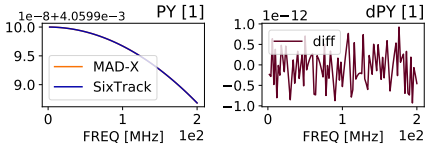
(a) X vs. frequency



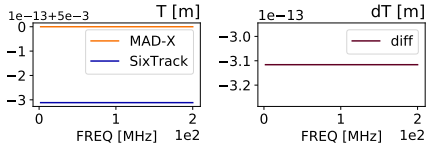
(b) PX vs. frequency



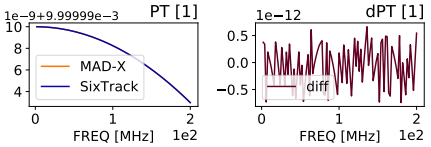
(c) Y vs. frequency



(d) PY vs. frequency

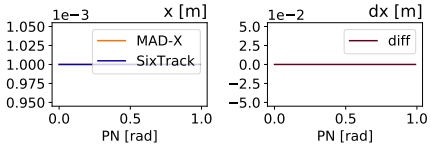


(e) T vs. frequency

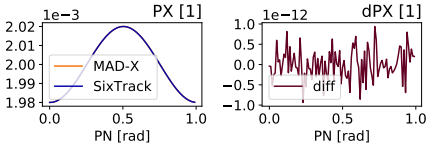


(f) PT vs. frequency

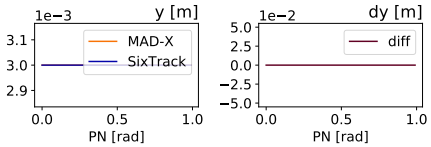
Figure A.55: Comparison output for the RF Quadrupole element when varying FREQ.



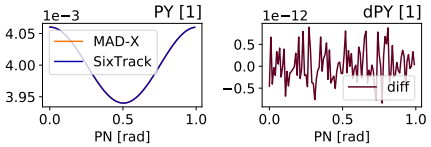
(a) X vs. normal phase



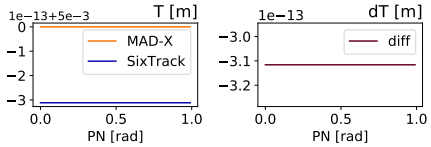
(b) PX vs. normal phase



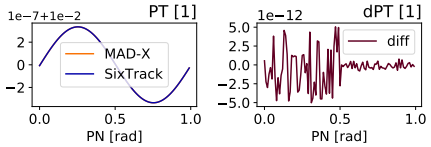
(c) Y vs. normal phase



(d) PY vs. normal phase

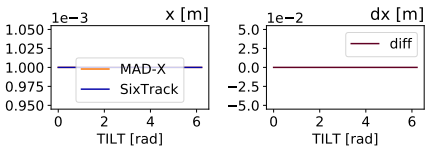


(e) T vs. normal phase

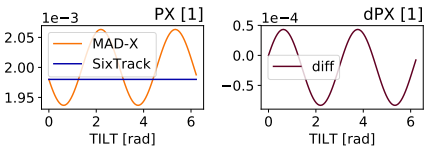


(f) PT vs. normal phase

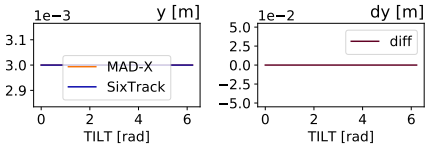
Figure A.56: Comparison output for the RF Quadrupole element when varying PN.



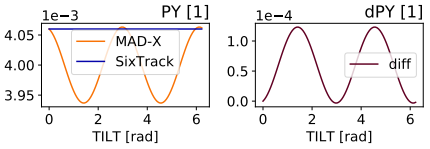
(a) X vs. tilt for normal RF Quadrupole



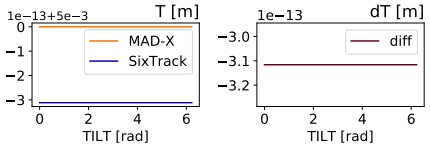
(b) PX vs. tilt for normal RF Quadrupole



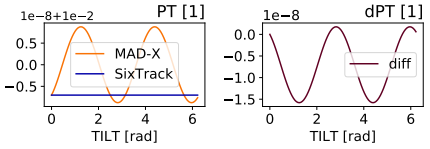
(c) Y vs. tilt for normal RF Quadrupole



(d) PY vs. tilt for normal RF Quadrupole

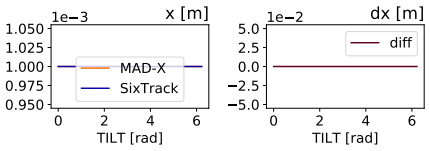


(e) T vs. tilt for normal RF Quadrupole

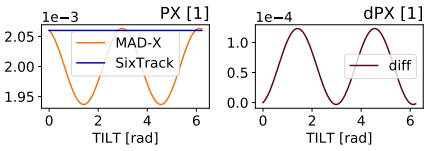


(f) PT vs. tilt for normal RF Quadrupole

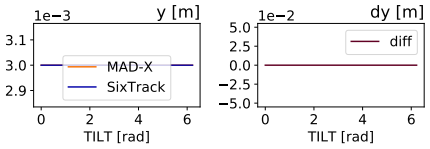
Figure A.57: Comparison output for the RF Quadrupole element when varying TILT for normal RF Quadrupole.



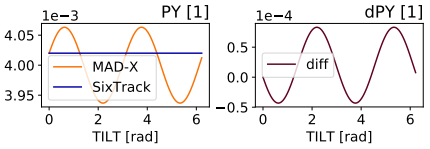
(a) X vs. tilt for skewed RF Quadrupole



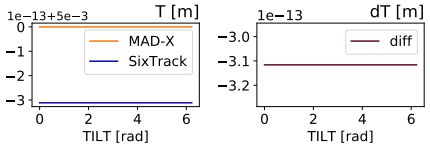
(b) PX vs. tilt for skewed RF Quadrupole



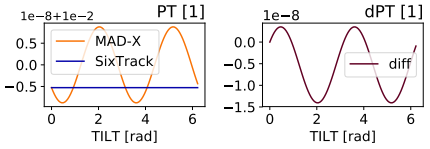
(c) Y vs. tilt for skewed RF Quadrupole



(d) PY vs. tilt for skewed RF Quadrupole

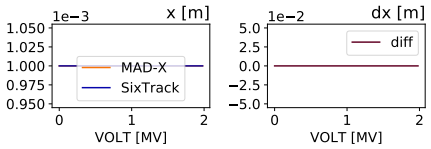


(e) T vs. tilt for skewed RF Quadrupole

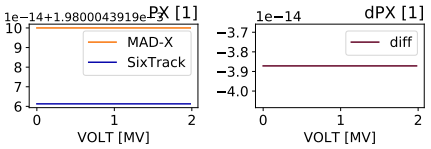


(f) PT vs. tilt for skewed RF Quadrupole

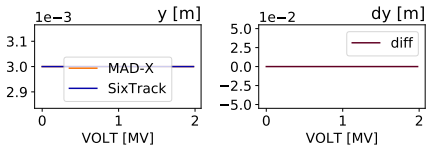
Figure A.58: Comparison output for the RF Quadrupole element when varying TILT for skewed RF Quadrupole.



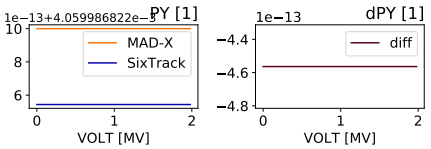
(a) X vs. VOLT



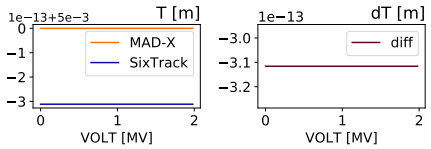
(b) PX vs. VOLT



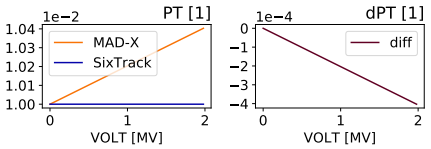
(c) Y vs. VOLT



(d) PY vs. VOLT



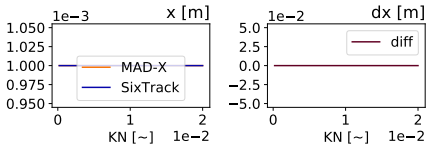
(e) T vs. VOLT



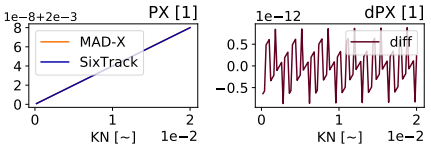
(f) PT vs. VOLT

Figure A.59: Comparison output for the RF Quadrupole element when varying VOLT.

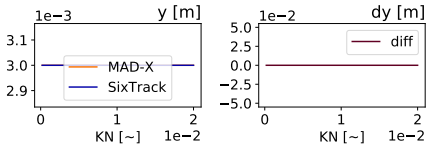
A.15 RF Sextupole



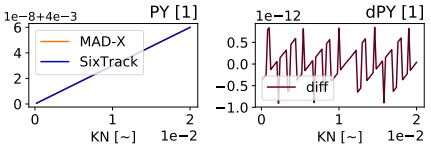
(a) X vs. normal strength



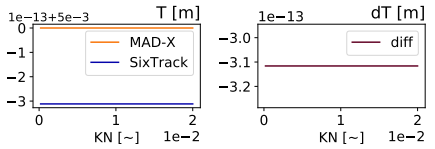
(b) PX vs. normal strength



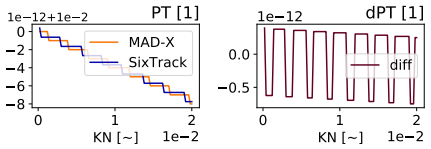
(c) Y vs. normal strength



(d) PY vs. normal strength

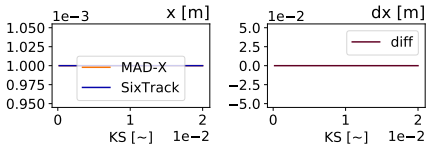


(e) T vs. normal strength

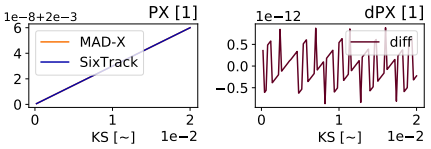


(f) PT vs. normal strength

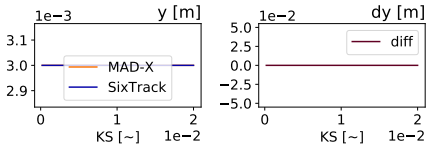
Figure A.60: Comparison output for the RF Sextupole element when varying KN.



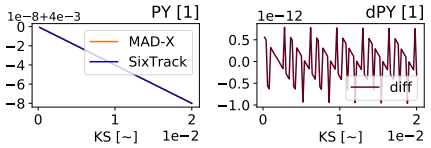
(a) X vs. skew strength



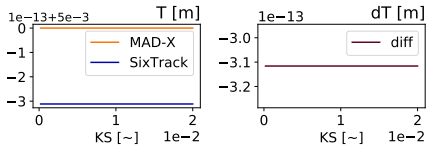
(b) PX vs. skew strength



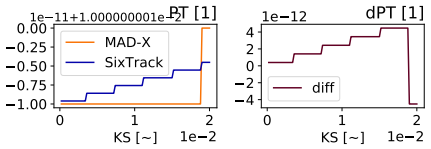
(c) Y vs. skew strength



(d) PY vs. skew strength

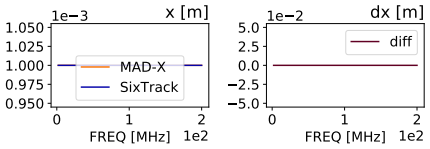


(e) T vs. skew strength

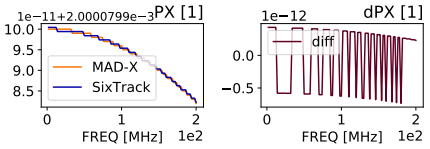


(f) PT vs. skew strength

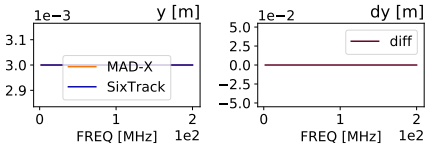
Figure A.61: Comparison output for the RF Sextupole element when varying KS.



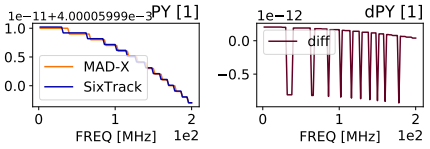
(a) X vs. frequency



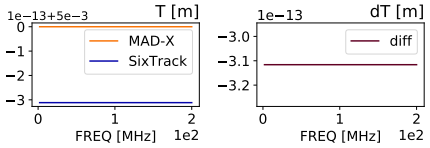
(b) PX vs. frequency



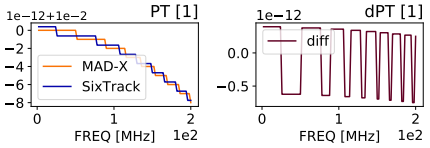
(c) Y vs. frequency



(d) PY vs. frequency

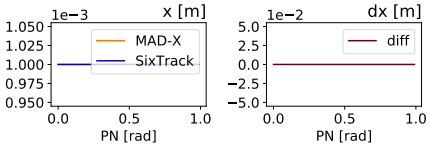


(e) T vs. frequency

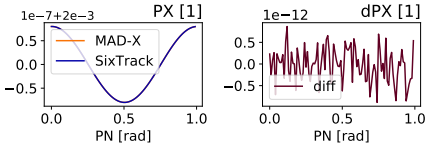


(f) PT vs. frequency

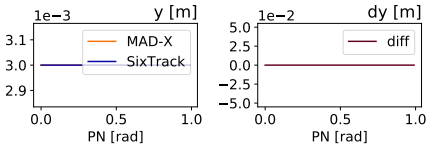
Figure A.62: Comparison output for the RF Sextupole element when varying FREQ.



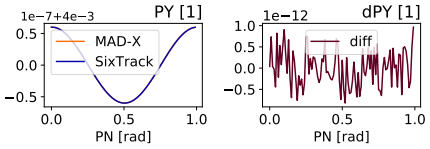
(a) X vs. normal phase



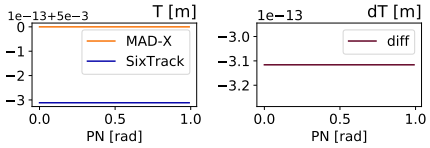
(b) PX vs. normal phase



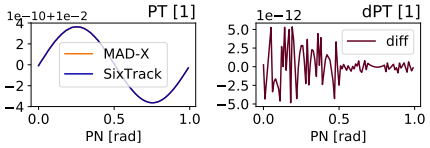
(c) Y vs. normal phase



(d) PY vs. normal phase

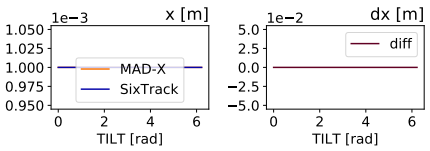


(e) T vs. normal phase

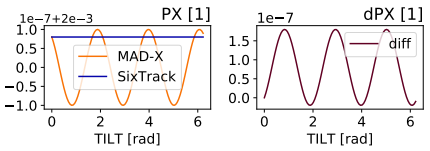


(f) PT vs. normal phase

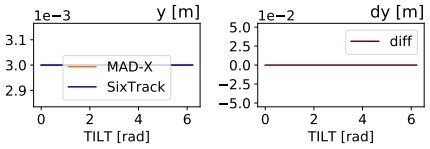
Figure A.63: Comparison output for the RF Sextupole element when varying PN.



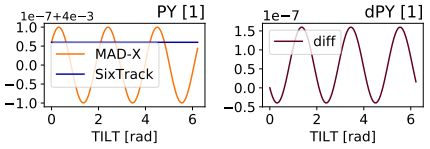
(a) X vs. tilt for normal RF Sextupole



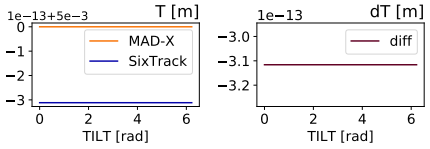
(b) PX vs. tilt for normal RF Sextupole



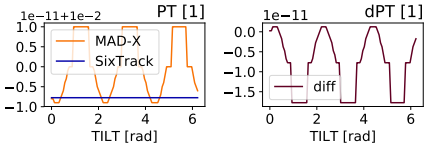
(c) Y vs. tilt for normal RF Sextupole



(d) PY vs. tilt for normal RF Sextupole

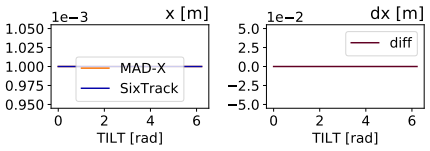


(e) T vs. tilt for normal RF Sextupole

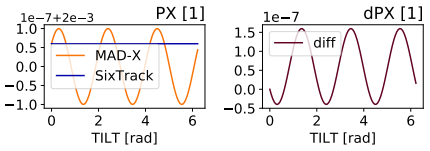


(f) PT vs. tilt for normal RF Sextupole

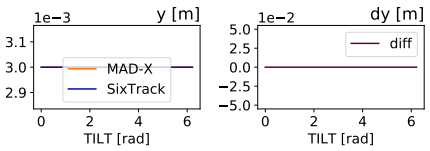
Figure A.64: Comparison output for the RF Sextupole element when varying TILT for normal RF Sextupole.



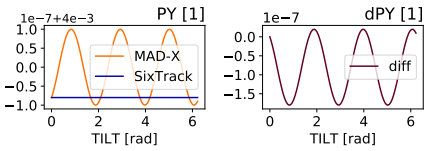
(a) X vs. tilt for skewed RF Sextupole



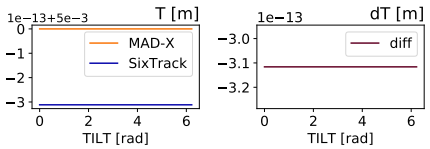
(b) PX vs. tilt for skewed RF Sextupole



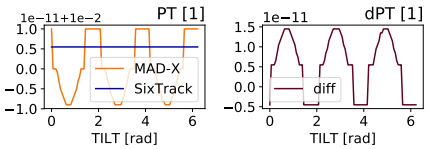
(c) Y vs. tilt for skewed RF Sextupole



(d) PY vs. tilt for skewed RF Sextupole

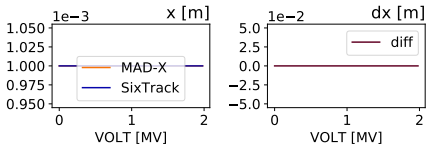


(e) T vs. tilt for skewed RF Sextupole

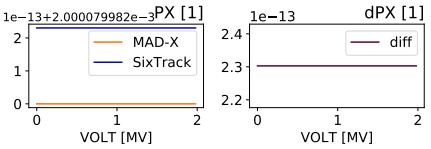


(f) PT vs. tilt for skewed RF Sextupole

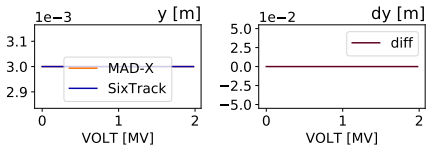
Figure A.65: Comparison output for the RF Sextupole element when varying TILT for skewed RF Sextupole.



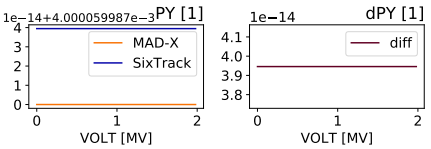
(a) X vs. VOLT



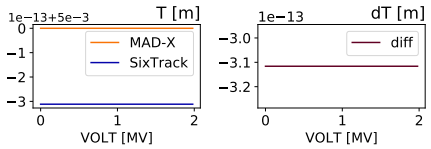
(b) PX vs. VOLT



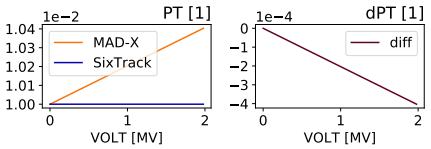
(c) Y vs. VOLT



(d) PY vs. VOLT



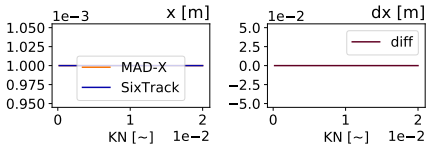
(e) T vs. VOLT



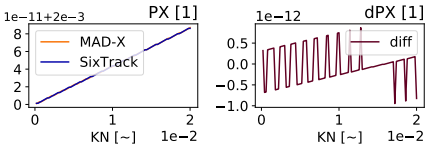
(f) PT vs. VOLT

Figure A.66: Comparison output for the RF Sextupole element when varying VOLT.

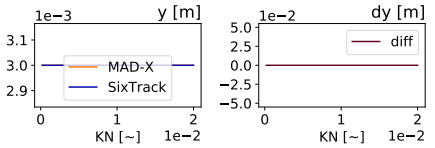
A.16 RF Octupole



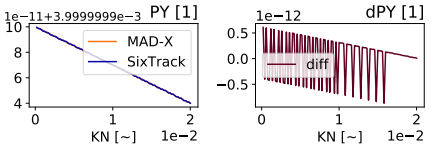
(a) X vs. normal strength



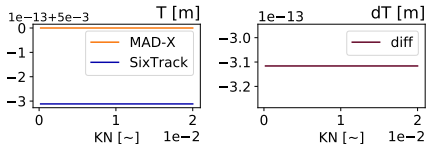
(b) PX vs. normal strength



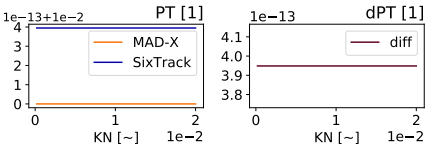
(c) Y vs. normal strength



(d) PY vs. normal strength

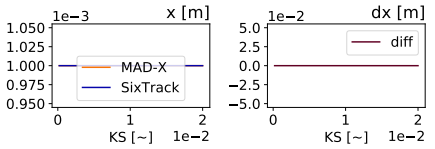


(e) T vs. normal strength

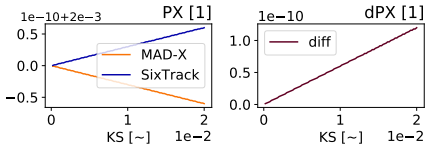


(f) PT vs. normal strength

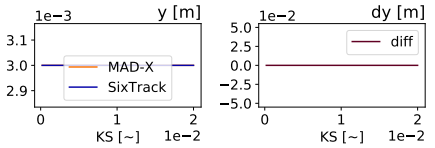
Figure A.67: Comparison output for the RF Octupole element when varying KN.



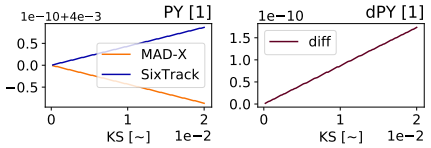
(a) X vs. skew strength



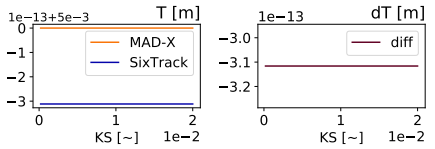
(b) PX vs. skew strength



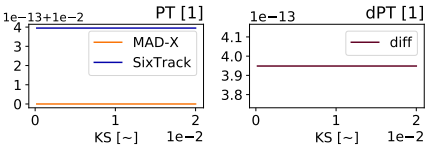
(c) Y vs. skew strength



(d) PY vs. skew strength

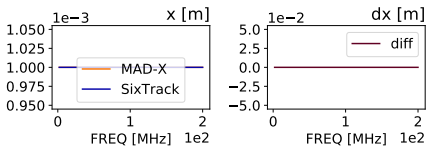


(e) T vs. skew strength

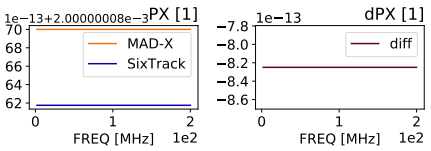


(f) PT vs. skew strength

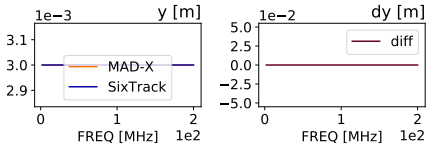
Figure A.68: Comparison output for the RF Octupole element when varying KS.



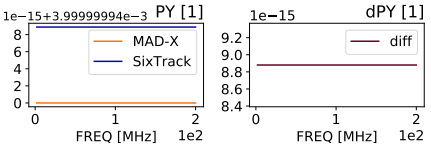
(a) X vs. frequency



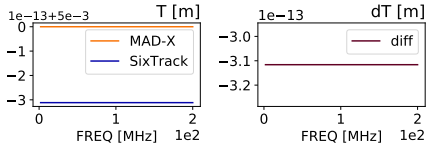
(b) PX vs. frequency



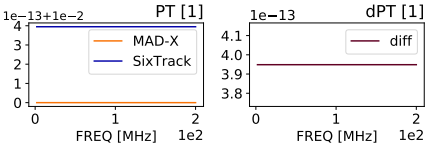
(c) Y vs. frequency



(d) PY vs. frequency

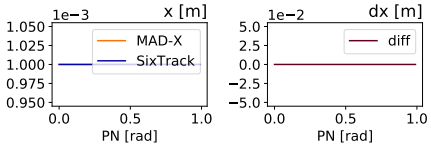


(e) T vs. frequency

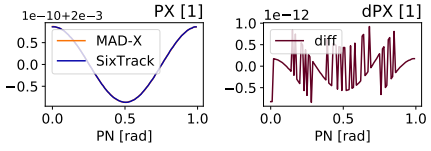


(f) PT vs. frequency

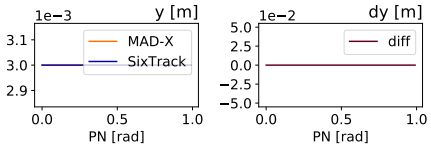
Figure A.69: Comparison output for the RF Octupole element when varying FREQ.



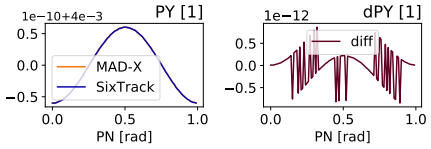
(a) X vs. normal phase



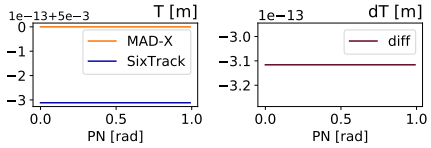
(b) PX vs. normal phase



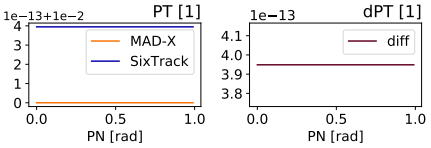
(c) Y vs. normal phase



(d) PY vs. normal phase

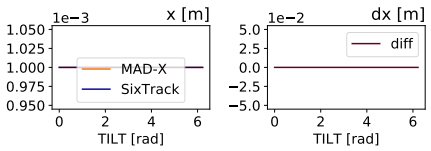


(e) T vs. normal phase

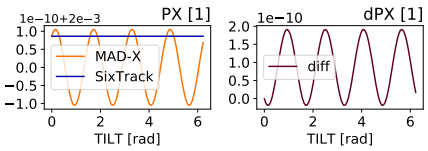


(f) PT vs. normal phase

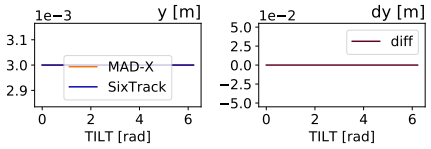
Figure A.70: Comparison output for the RF Octupole element when varying PN.



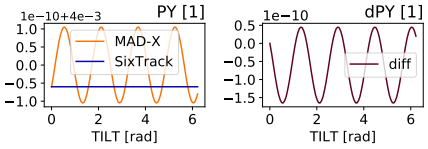
(a) X vs. tilt for normal RF Octupole



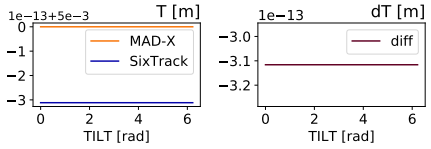
(b) PX vs. tilt for normal RF Octupole



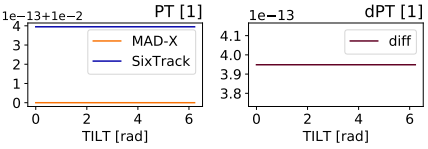
(c) Y vs. tilt for normal RF Octupole



(d) PY vs. tilt for normal RF Octupole

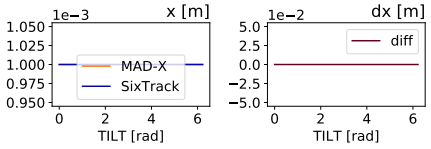


(e) T vs. tilt for normal RF Octupole

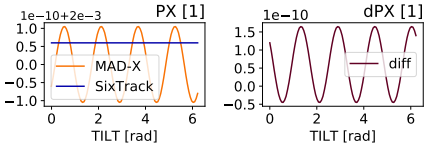


(f) PT vs. tilt for normal RF Octupole

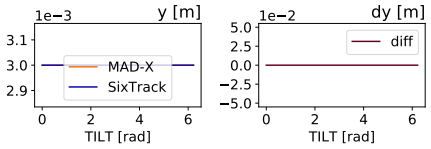
Figure A.71: Comparison output for the RF Octupole element when varying TILT for normal RF Octupole.



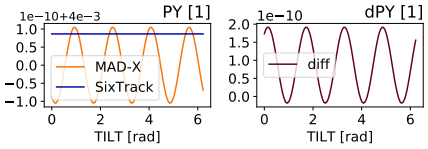
(a) X vs. tilt for skewed RF Octupole



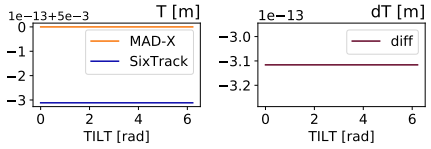
(b) PX vs. tilt for skewed RF Octupole



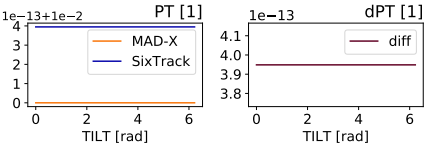
(c) Y vs. tilt for skewed RF Octupole



(d) PY vs. tilt for skewed RF Octupole

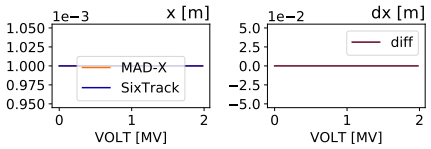


(e) T vs. tilt for skewed RF Octupole

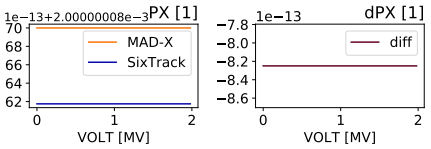


(f) PT vs. tilt for skewed RF Octupole

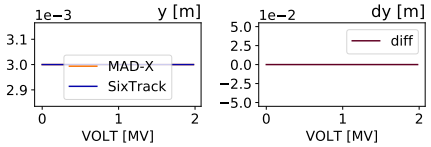
Figure A.72: Comparison output for the RF Octupole element when varying TILT for skewed RF Octupole.



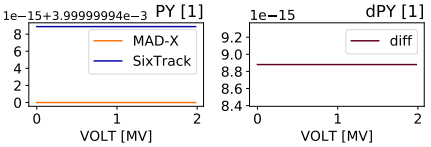
(a) `X` vs. `VOLT`



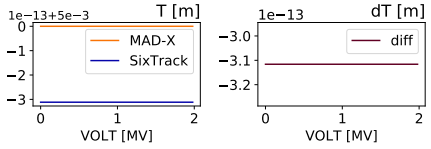
(b) `PX` vs. `VOLT`



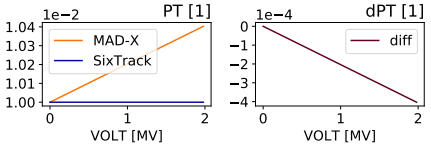
(c) `Y` vs. `VOLT`



(d) `PY` vs. `VOLT`



(e) `T` vs. `VOLT`



(f) `PT` vs. `VOLT`

Figure A.73: Comparison output for the RF Octupole element when varying `VOLT`.