

# Beam and Spin Optics Simulation Tutorials, Using Zgoubi Exercises and Solutions 2 - RHIC Run 22 Spin Transparency Experiment

F. Méot

September 2022

Collider Accelerator Department  
**Brookhaven National Laboratory**

**U.S. Department of Energy**

USDOE Office of Science (SC), Nuclear Physics (NP) (SC-26)

Notice: This technical note has been authored by employees of Brookhaven Science Associates, LLC under Contract No. DE-SC0012704 with the U.S. Department of Energy. The publisher by accepting the technical note for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this technical note, or allow others to do so, for United States Government purposes.

## **DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

F. Méot, BNL C-AD

# Beam and Spin Optics Simulation Tutorials, Using Zgoubi

## Exercises and Solutions

### 2 - RHIC Run 22 Spin Transparency Experiment

September 2022

*A tutorial series held at BNL C-AD,  
following RHIC Run 22*

Tech. Note C-A/AP/662



# Contents

Acknowledgements .....	1
<b>1 Spin Transparency Experiment at RHIC .....</b>	<b>3</b>
1.1 Introduction .....	3
1.2 These Simulations .....	4
Snake Field Maps .....	4
<b>2 Simulation Exercises .....</b>	<b>7</b>
2.1 Setting Up Snake Current Ramps .....	7
2.2 Setting Up RHIC Yellow Optics Model .....	10
2.3 Spin Transparency Simulations .....	12
2.3.1 Vertical to radial polarization .....	12
2.3.2 Vertical polarization flip .....	14
<b>3 Solutions of Simulation Exercises .....</b>	<b>15</b>
Snake Current Ramps .....	15
Setting Up Regular Yellow optics .....	20
Spin Transparency Simulations .....	24
Vertical to radial polarization .....	24
Vertical polarization flip .....	31
References .....	32



## **Acknowledgements**

I thank my colleagues from C-AD and EIC departments who attended this tutorial, for their feedback on the exercises and solutions.





# Tutorial Part 1

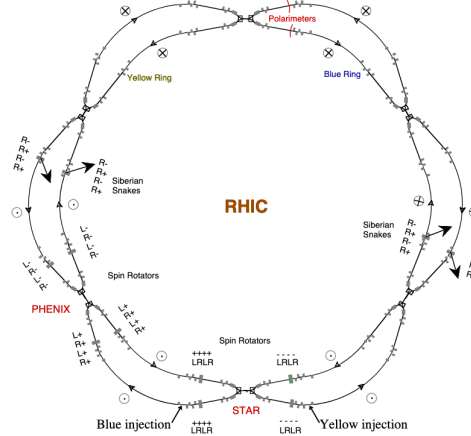
## Spin Transparency Experiment at RHIC

### 1.1 Introduction

High electron and ion polarizations are paramount design requirements for the future Electron Ion Collider (EIC) at BNL. A “spin transparency” mode of operation, inherited from the “figure-8” ring design [1], allows preservation and control of electron and ion spin polarization in a collider or storage ring [2, 3]. It makes the ring lattice “invisible” to the spin and allows for polarization control by small quasi-static magnetic fields with practically no effect on beam orbital parameters. Beam polarization can then be fully controlled by small adjustments of snake axis orientation (a few degrees) and snake strengths (a few percent); polarization orientation at IPs can be controlled without spin rotators, spins can be adiabatically flipped without the need of ac magnets.

The transparent spin mode has been demonstrated in simulations [2] for the purpose of being tested experimentally during RHIC Run 22. This experiment was aimed at providing data to validate the principles. Yellow RHIC ring (Fig. 1.1) was configured in the transparent spin mode by aligning the axes of its two helical snakes (located respectively in the 3 o’clock and 9 o’clock sextants) parallel to the longitudinal axis. Experiment goals and expected results, *ad hoc* procedures and hardware requirements, are addressed in Refs. [2, 3].

**Fig. 1.1** Locations of the helical snakes in RHIC Blue and Yellow rings, at 9 o'clock and 3 o'clock. Beam goes clockwise in Blue, counter-clockwise in Yellow. Zgoubi frame is (O;X/long.,Y/radial,Z/vertic.). In Blue: X tangent to orbit points as velocity (CW), Y points outward, Z points up (curvature is negative/center at negative Y, dispersion is  $> 0$ ); in Yellow: X tangent to orbit points as velocity (CCW), Y points outward, Z points down (curvature is negative/center at negative Y, dispersion is  $> 0$ ).



## 1.2 These Simulations

Most importantly:

they require having Zgoubi Users' Guide at hand.

it can be found here:

<https://sourceforge.net/p/zgoubi/code/HEAD/tree/trunk/guide/Zgoubi.pdf>

Navigating the INDEX is a convenient way to find out what the commands, keywords, and other numerical data found in the input data file do or represent.

Spin transparency mode of operation at RHIC had been demonstrated in simulations, prior to proposing Run 22 APEX. The goal of this tutorial is to go over these simulations in detail. They can be seen as an opportunity to review the computational methods and tools.

All the exercises and necessary files in these simulations are organized in folders, downloadable at

[https://sourceforge.net/p/zgoubi/code/HEAD/tree/trunk/exemples/tutorials/BNLSpinTransparency\\_2022](https://sourceforge.net/p/zgoubi/code/HEAD/tree/trunk/exemples/tutorials/BNLSpinTransparency_2022)

Note: from here on, "[pathTo]/zgoubi-code/" will represent the absolute address of the sourceforge package on your computer.

Found there as well is a copy of the present document:

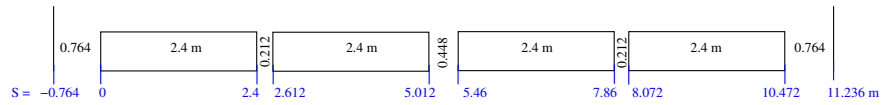
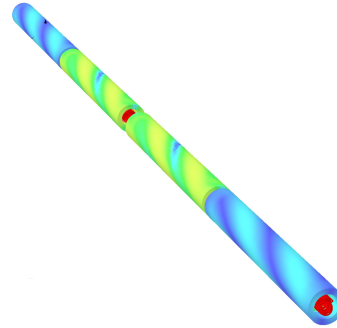
[pathTo]/zgoubi-code/exemples/tutorials/BNLSpinTransparency\_2022/tutorialAssignments/tutorialAssignments.pdf

## Snake Field Maps

RHIC snakes are comprised of four identical  $360^\circ$ -twist helices (Fig. 1.2), with the following parameters [4] (Figs. 1.1, 1.3)

- helices are right-handed,
- a helix is 2.4 m long,

**Fig. 1.2** RHIC snake model in OPERA, a 4-module structure, all four modules are right-handed helices, magnetic field sign alternates from one to the next. The two inner modules (2 and 3) are fed in series ( $I_{in}$  current), the two outer modules (1 and 4) are fed in series as well ( $I_{out}$  current). In red, visible in the center and at the right end: helical coils of an inner and an outer module.



**Fig. 1.3** A sketch of RHIC snake and rotator layouts, each rectangle represents a 2.4 m long 360°-twist helix. Field maps from the OPERA model extend an extra 0.764 m at both ends, to encompass field fall-offs.

- they are spaced 0.212/0.448/0.212 m,
- the central 0.448 m drift space includes a BPM,

hence an overall snake length of 10.472 m. Regarding the geometrical acceptance at the location of a snake: magnet bore is 10 cm.

OPERA field maps of the modules are used in these simulations. In this representation a full snake is a sequence of 4 field maps: field maps 1 and 4 have been computed at low field (100 A current), 2 and 3 have been computed at high field (322 A) [4]. Field signs to be assigned to module field maps, going counter-clockwise around RHIC Yellow, are as follows:

sign to be applied		field map file name
3 o'clock snake	9 o'clock snake	
(R+R-R+R-)	(R-R+R-R+)	
+	-	model3a2a - x - 4_4_y - 4_4_z - 180_180 - integral.table
-	+	model3a2a322a - x - 4_4_y - 4_4_z - 180_180 - integral.table
+	-	model3a2a322a - x - 4_4_y - 4_4_z - 180_180 - integral.table
-	+	model3a2a - x - 4_4_y - 4_4_z - 180_180 - integral.table

These OPERA maps are archived/available on C-AD computers at

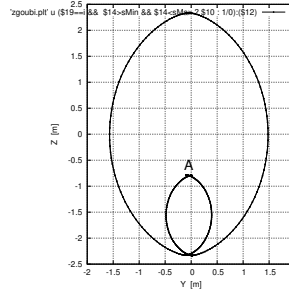
`/rap/lattice_tools/zgoubi/RHICZgoubiModel/snakeFieldMaps/161216_secondSet_inclSingleHelix`

or in sourceforge at

`[pathTo]/zgoubi - code/exemples/RHIC/snakesWithFieldMaps/fieldMaps/`

Binary versions of these files are also available therein, same name preceded with a “b\_” prefix; binary files offer the advantage of much faster reading.

**Fig. 1.4** (Y,Z) projection of the helical orbit in a snake, at  $G\gamma = 45.5$ . Closed orbit entrance and exit points are at A. Note that the motion is a *non-circular* helix



### Orbit and Spin Perturbation by Snakes

At injection energy ( $G\gamma = 45.5$ ) the helical closed orbit along the inner two helices of a snake is centered on the snake axis, Fig. 1.4. This imposes presenting the closed orbit at a non-zero vertical excursion at the entrance of the snake (point “A” in Fig. 1.4), and the orbit will exit the snake at a similar vertical excursion (details in [4]). Snakes also introduce a small horizontal orbit defect. The subsequent orbit bump is closed locally, over  $\approx 90$  meters around the snake, this local orbit compensation in turn impacts on the spin closed orbit. These effects are dealt with as part of RHIC optics by accounting for detailed closed orbit geometry and spin motion throughout the 10.4 m snake extent, which requires using their field maps, this is the object of simulations in Sec. 2.

Regarding RHIC optics, the starting point for these spin transparency simulations is Run 22 MADX model, which has been translated to zgoubi. Preliminary checks of RHIC optics modeling with field maps are the object of simulations in Sec. 2.2.

Spin transparency simulations proper are worked out in Sec. 2.3. In a first simulation, Sec. 2.3.1, polarization is moved from vertical to radial. In a second simulation, Sec. 2.3.2, vertical polarization is flipped.

## Tutorial Part 2

# Simulation Exercises

### 2.1 Setting Up Snake Current Ramps

This first exercise aims at determining the snake current ramps required to move the polarization from vertical to radial in a first experiment, and for polarization flip in a second experiment.

Simulation files concerned in this Section can be found at,  
regarding vertical to radial orientation:

[pathTo]/zgoubi-code/exemples/RHIC/spinTransparency/snakeSettings/adjustRadialPolar/  
and, regarding polarization flipping:

[pathTo]/zgoubi-code/exemples/RHIC/spinTransparency/snakeSettings/reverseVerticalPolar/  
In both folders, the two jobs to run are

[pathTo]/zgoubi-code/zgoubi/zgoubi -in snake3OCCW\_scanMuPhi.dat  
and

[pathTo]/zgoubi-code/zgoubi/zgoubi -in snake9OCCW\_scanMuPhi.dat

Note that regular currents in normal RHIC polarized proton operation, which yield snake axes at  $\pm 45^\circ$  from longitudinal and spin rotation  $180^\circ$ , are

$$I_{\text{out}} = 100 \text{ A}, \quad I_{\text{in}} = 322 \text{ A},$$

for both snakes. These currents undergo substantial changes for spin transparency operation.

#### • Question 2.1-1

First read the next question (Question 2.1-2), then devote some thinking to understanding the content of snake3OCCW\_scanMuPhi.dat and snake9OCCW\_scanMuPhi.dat data files, including the various keywords and their numerical arguments: OBJECT, PARTICULAR, SPNTRK, OPTIONS, SPNPRT[MATRIX,PRINT], beam line sequence, FIT procedure, SYSTEM. Consult the Users' Guide for that.

In particular, explain

- the role of REBELOTE do-loop in this simulation, what parameter it changes from an iteration to the next,

- the role of the OPTIONS[.plt 2] command, specific to this orbit bump design simulation;

- what causes these jobs to end up with a graph of the orbits.

• **Question 2.1-2**

Solution: page 16

The first test of the spin transparency experiment, vertical to radial polarization, requires determining the snake currents needed to achieve the 11 different  $\mu, \phi$  snake angle couples given in Tab. 2.1.

The folder concerned is

[pathTo]/zgoubi-code/exemples/RHIC/spinTransparency/snakeSettings/adjustRadialPolar/

**Table 2.1** Required 3 o'clock and 9 o'clock snake angle sequence for moving polarization from vertical (at step 0) to radial (at step 10).

Step #	3 o'clock		9 o'clock	
	$\mu$	$\phi$	$\mu$	$\phi$
0	180	0	180	170
1	177	0	180	170
2	174	0	180	170
3	171	0	180	170
4	168	0	180	170
5	165	0	180	170
6	165	0	177	172
7	165	0	174	174
8	165	0	171	176
9	165	0	168	178
10	165	0	165	180

**Table 2.2** Required 3 o'clock and 9 o'clock snake angle sequence for vertical polarization flip.

Step #	3 o'clock		9 o'clock	
	$\mu$	$\phi$	$\mu$	$\phi$
0	180	0	180	-10
1	177.5	0	180	-10
2	175	0	180	-10
3	175	0	180	-5
4	175	0	180	-2
5	175	0	180	0
6	175	0	180	2
7	175	0	180	5
8	175	0	180	10
9	177.5	0	180	10
10	180	0	180	10

The two jobs to run, reminder:

[pathTo]/zgoubi-code/zgoubi/zgoubi -in snake3OCCW\_scanMuPhi.dat

and

[pathTo]/zgoubi-code/zgoubi/zgoubi -in snake9OCCW\_scanMuPhi.dat

are respectively set up for these 9 o'clock and 3 o'clock snake current scans.

Run these files and check the following:

- FIT (or FIT2 - figure out the nature of both algorithms) finds the  $I_{out}, I_{in}$  snake currents (FIT variables) for a given  $(\mu, \phi)$  pair (FIT constraints);
- REBELOTE is used to repeat over the 11 different  $(\mu, \phi)$  values.
- gnuplot scripts under SYSTEM provide various graphs, including the 11 orbits across each snake - this allows verifying that they remain about centered throughout the scan (tight centering is unnecessary as far as spin motion is concerned); magnetic field along these orbits, etc.

Complete Tabs. 2.1 with, for each snake, its corresponding  $I_{out}, I_{in}$  columns.

- **Question 2.1-3** Solution: page 18

The second test of the spin transparency experiment, vertical spin flip, requires determining the snake currents needed to achieve the 11 different  $\mu, \phi$  snake angle couples given in Tab. 2.2.

The folder concerned is

[pathTo]/zgoubi-code/exemples/RHIC/spinTransparency/snakeSettings/reverseVerticalPolar/

Here as well, the two jobs to run, reminder:

[pathTo]/zgoubi-code/zgoubi/zgoubi -in snake3OCCW\_scanMuPhi.dat

and

[pathTo]/zgoubi-code/zgoubi/zgoubi -in snake9OCCW\_scanMuPhi.dat

are respectively set up for these 9 o'clock and 3 o'clock snake current scans.

Repeat the procedure of Question 2.1-2 for this polarization flip case.

Complete Tab. 2.2 with, for each snake, the corresponding  $I_{out}$ ,  $I_{in}$  columns.

## 2.2 Setting Up RHIC Yellow Optics Model

This preliminary simulation makes sure RHIC Run 22 Yellow optics, counterclockwise (CCW), is all set, ready to go!

The folder concerned is

```
[pathTo]/zgoubi-code/exemples/tutorials/
BNLSpinTransparency_2022/RHIC-run22_ppLattice/
```

This simulation of RHIC Yellow ring involves a main file,  
pp22-yellow-CCW.INC.dat,

this file includes the complete Yellow optical sequence, CCW. It is a good idea to make sure that this file is correctly set up and yields the expected optical functions, as most of all other simulation files resort to INCLUDEs grabbing from that pp22-yellow-CCW.INC.dat file.

That file pp22-yellow-CCW.INC.dat also resorts itself to INCLUDEs, concerning namely the three ancillary files

- SCALING\_snks100A-322A.inc, which sets the power supplies of the optical elements, including 100A/322A snake currents,
- snake3OCCW\_Segment.inc,
- snake9OCCW\_Segment.inc.

The optical sequences in the latter two files are ~ 90 meter long segments of RHIC lattice encompassing, respectively, the 3 o'clock snake (snake 2 in the following, coming first when going around Yellow counter-clockwise, see Fig. 1.1) and the 9 o'clock snake (snake 1, coming second when going around Yellow counter-clockwise). These ~ 90 meter segments include each 4 upstream and 4 downstream H and V kickers which allow designing a local closed orbit bump. They are stand-alone files, and allow the design of these orbit bumps by means of a fitting procedure (FIT command, at the bottom of the file).

### • Question 2.2-1

This question addresses the setting up of the local  $\approx 90$  m extent closed orbit bumps at the snakes.

Run snake3OCCW\_Segment.inc. Get from that

- (i) a graph of the projection of the helical motion in each snake;
- (ii) a graph of the closed orbit in the snake region.

Repeat for snake9OCCW\_Segment.inc,

### • Question 2.2-2

The file pp22-yellow-CCW.INC.dat is set for regular RHIC optics, which means regular snake currents,  $I_{out} = 100$  A,  $I_{in} = 322$  A.

1. Describe the (O;X,Y,Z) frame in this this Yellow-CCW model: the X axis points in the direction of the motion by definition; clarify orientation of Y axis (inward or outward) and Z axis (up or down).



2. Make sure about that (snake currents are set by TOSCA power supplies under SCALING command).
3. Produce the closed orbit and optical functions around the ring. In other words:  
`[pathTo]/zgoubi-code/zgoubi/zgoubi -in pp22-yellow-CCW.INC.dat`  
 In passing: that creates as always a zgoubi.res listing. Copy it to temp. Run temp, check any difference between temp and its new zgoubi.res (use 'diff -u temp zgoubi.res'). Explain.
4. Figure out what is going on, what zgoubi does when going through pp22-yellow-CCW.INC.dat.
5. Right before TWISS, following the sequence in pp22-yellow-CCW.INC.dat (bottom of the file), a FIT command has been commented. What would it do? In view of the previous observations regarding closed orbits, should it be un-commented?

• **Question 2.2-3**

While we are in that folder,

1. and dealing with paraxial optics, find the stable spin precession axis  $\mathbf{n}_0$  at IP6. In other words:  
`[pathTo]/zgoubi-code/zgoubi/zgoubi -in pp22-yellow-CCW_spinN0.INC.dat.`  
 Figure out what zgoubi does when going through pp22-yellow-CCW\_spinN0.INC.dat.
2. Propagate  $\mathbf{n}_0(IP6)$  around the ring, give a graph of the three components  $\mathbf{n}_{0,x,y,z}(s)$ .

## 2.3 Spin Transparency Simulations

### 2.3.1 Vertical to radial polarization

In order to manipulate a slimmer (and clearer) main input file, arrange it under the form of a series of INCLUDEs, namely:

pp22-yellow-CCW\_spinN0.INC.dat

In that file, 5 RHIC segments are subject to INCLUDE commands, namely

- Clock 6 to 3 o'clock snake
- 3 o'clock snake segment proper,
- 3 o'clock snake to 9 o'clock snake,
- 9 o'clock snake segment proper,
- 9 o'clock snake to Clock6.

The snake segments remain unchanged, namely, snake3OCCW\_Segment.inc and snake9OCCW\_Segment.inc as used in the previous simulations. The other 3 RHIC lattice segments above are grabbed from the previously used optical sequence file pp22-yellow-CCW.INC.dat. Note that the 3 o'clock snake to 9 o'clock snake segment can be split at POL3 and the 6 segments permuted circularly, in order to start the sequence at pC polarimeter (this will be necessary in upcoming questions).

#### • Question 2.3.1-1

In SCALING.inc, set the snake currents to the values needed at step 0 of the vertical to radial polarization adjustment experiment, namely,

- snake 2 (3 o'clock):  $I_{out} = 164.16$  A,  $I_{in} = 220.455$ ;
- snake 1 (9 o'clock):  $I_{out} = 184.08$  A,  $I_{in} = 199.07$ .

A consequence of this change of snake currents is that the orbit kickers need be adjusted, for proper centering of the helix in the snake, and proper local closed orbit bump. To achieve this, run the snake segment jobs to find (a FIT does it) the proper orbit kicker settings (as was done in Sec. 2.1 in the case of the regular 100A/322A snake currents).

Update SCALING.inc with these new kicker values.

Note that FIT produces zgoubi.FIT.out.dat. Explain the goal in doing so.

#### • Question 2.3.1-2

It does not hurt to check RHIC closed orbit once snake currents have been updated (in SCALING.inc) and local orbit kickers have been updated (in SCALING.inc as well).

Do this check of RHIC closed orbit, by running the job pp22-yellow-CCW\_spinN0.INC.dat.

#### • Question 2.3.1-3

Find the spin closed orbit with these snake currents, running pp22-yellow-CCW\_spinN0.INC.dat.

Explain what pp22-yellow-CCW\_spinN0.INC.dat does, in particular, explain the arguments in OBJET, SPNTRK, and SPNPRT[MATRIX].

Explain in detail the arguments of the FIT procedure in pp22-yellow-CCW\_spinN0.INC.dat; this FIT command is commented as can be seen. Uncommented and run the job again. What does this add, what is the interest?

- **Question 2.3.1-4**

Get the 1-turn spin matrix with origin at the pC polarimeter.

- **Question 2.3.1-5**

Using REBELOTE command, set up a scan of the 1-turn spin matrix over the 11 snake angle steps of Tab. 2.1.

Produce a graph of the spin tune and spin precession vector components so obtained (with step number as horizontal axis).

- **Question 2.3.1-6**

Produce a fast simulation of the experiment, as follows: launch a single particle on the closed orbit with its initial spin aligned on  $\mathbf{n}_0$  at step 0; track over 5000 turns, while snake currents follow steps 0 to 10, Tab. 3.1. Use SCALING to update current values during tracking. Take the origin of the optical sequence at pC polarimeter, as this is where the polarization measurements are done during the APEX.

Produce a graph of the spin tune and spin vector components along the snake current ramp. A discrete Fourier transform of spin motion over a few tens of turns can be used to compute the spin motion spectrum along the current ramp. The expected results include that (i) the particle remain on the closed orbit, (ii) 5000 turns is slow enough that the particle's spin vector follow the stable spin precession direction.

- **Question 2.3.1-7**

Replace the snake field maps, which induce orbital motion perturbation, by SPINR which has no effect on orbital motion. Leave H and V local closed orbit bump corrector settings unchanged.

2.3.1-7.a - repeat the scan of question 2.3.1-5. Compare the results.

2.3.1-7.b - repeat the ramp of question 2.3.1-6. Compare the results.

- **Question 2.3.1-8**

It can be observed from the results of questions 2.3.1-5, 6, that although prescribed snake angles follow from theory, the spin which should come out constant, does not:

- find an explanation to this seemingly discrepancy with theory;
- devise a fitting method which results in accurately constant spin tune, by relaxing on snake angles.

### 2.3.2 Vertical polarization flip

The folder concerned in this simulation is

[pathTo]/zgoubi-code/exemples/tutorials/

BNLSpinTransparency\_2022/spinTransparency\_2\_reverseVerticalPolarization/

The jobs concerned are

pp22-yellow-CCW\_spinN0\_reverseVPol\_pCPol.INC.dat (Question 2.3.2-1)

sweep\_origin-pCPol\_0-10.INC.dat (Question 2.3.2-2)

Snake settings are to be taken from the answer to Question 2.1-4 (Tab. 3.2). Note that, because the snake currents do not change much over step 0-step 10, closed orbit bump settings obtained for step 0 in Sec. 2.1-2 can still be used here.

Otherwise, the simulation method is the same as in the Sec. 2.3.1, which can be referred to.

- **Question 2.3.2-1**

Produce a scan of the spin closed orbit vector components, and spin tune, for the 11 snake current steps. Plot the results.

- **Question 2.3.2-2**

Produce a fast simulation of the experiment, as follows: launch a single particle on the closed orbit with its initial spin aligned on  $\mathbf{n}_0$  at step 0; track over 5000 turns, while snake currents follow steps 0 to 10 values, Tab. 3.2. Use SCALING to update current values during tracking. Take the origin of the optical sequence at pC polarimeter, as this is where the polarization measurements are done during the APEX.

Produce a graph of the spin tune and spin vector components along the ramp. A discrete Fourier transform of spin motion over a few tens of turns can be used to compute the spin motion spectrum along the current ramp. The expected results include that (i) the particle remain on the closed orbit, (ii) 5000 turns is slow enough that the particle's spin vector follow the stable spin precession direction.

## Tutorial Part 3

# Solutions of Simulation Exercises

### Setting Up Snake Current Ramps

#### Question 2.1 series, answers

- **2.1-1:**

OPTIONS[.plt,IL=2] forces IL=2 flag at all optical elements, so causing, for all elements, logging of stepwise particle data in zgoubi.plt. The content of zgoubi.plt is detailed in the Users' Guide [5, Sec. 8.3]. A 0 or 1 flag in column 50 of zgoubi.plt allows distinguishing from one another, if needed, particle data along the optical sequence (coordinates, spin vector, etc.) produced during the first pass through the latter (before reaching FIT) or during the final pass following FIT (thus, with matched variable values).

Note the following:

1/ this final pass through the input data sequence occurs once FIT has been completed, and uses the updated variable values; the final pass can be skipped if explicitly requested by specifying 'noFinal' under FIT[2];

2/ zgoubi.FIT.out is a log of constraint values and final values of the FIT variables;

3/ an updated version of the input data file, with FIT instruction commented, is logged in zgoubi.FIT.out.dat once FIT is completed.

The gnuplot scripts under the SYSTEM command produce graphs; looking into these scripts shows that plotted data are read from zgoubi.FITVALS.out or from zgoubi.plt.

Navigating the INDEX in Zgoubi Users' Guide will allow completing this question.

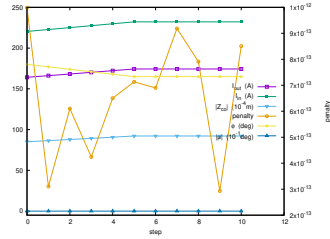
- **2.1-2:**

The first test of the spin transparency experiment, vertical to radial polarization, requires the snake currents given in Tab. 3.1. Graphs of the various parameters including snake currents, produced by the gnuplot script `gnuplot_FITVALS.gnu` under the `SYSTEM` command, are given in Figs. 3.1, 3.2.

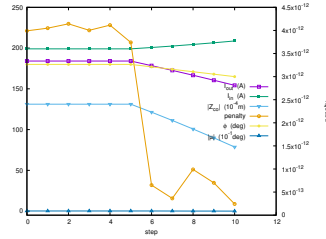
**Table 3.1** Snake angles and corresponding currents, for moving polarization from vertical (at step 0) to radial (at step 11)

Step #	3 o'clock snake, CCW				9 o'clock snake, CCW			
	$\mu$	$\phi$	$I_{out}$	$I_{in}$	$\mu$	$\phi$	$I_{out}$	$I_{in}$
0	180	0	164.16759	220.45335	180	170	184.08966	199.07334
1	177	0	166.17474	222.79739	180	170	184.08967	199.07334
2	174	0	168.19734	225.14344	180	170	184.08966	199.07336
3	171	0	170.23614	227.49256	180	170	184.08966	199.07335
4	168	0	172.29233	229.84602	180	170	184.08967	199.07334
5	165	0	174.36709	232.20533	180	170	184.08966	199.07334
6	165	0	174.36707	232.20531	177	172	178.43276	200.72111
7	165	0	174.36709	232.20531	174	174	172.63753	202.52287
8	165	0	174.36712	232.20540	171	176	166.69559	204.46703
9	165	0	174.36710	232.20535	168	178	160.59563	206.54226
10	165	0	174.36709	232.20534	165	180	154.32192	208.73700

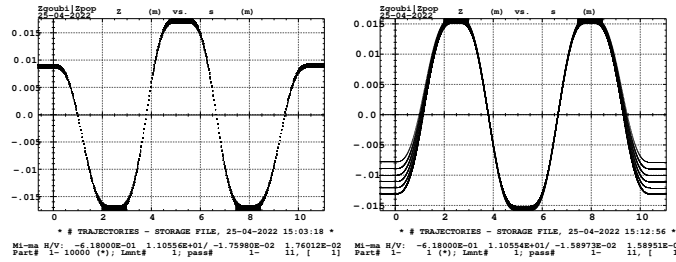
During the scan, the orbits are logged in `zgoubi.plt` (this is an effect of `IL=2` under TOSCA keywords, or `OPTIONS[.plt,IL=2]`). They are plotted reading from `zgoubi.plt`, Fig. 3.3.



**Fig. 3.1** Vertical to radial polarization adjustment, case of 3 o'clock snake, step 0 to 10: coil currents, snake angles, closed orbit coordinate at snake entrance, and (right vertical axis) FIT penalty to control the convergence of the fitting



**Fig. 3.2** Vertical to radial polarization adjustment, case of 9 o'clock snake, step 0 to 10: coil currents, snake angles, closed orbit coordinate at snake entrance, and (right vertical axis) FIT penalty to control the convergence of the fitting



**Fig. 3.3** Case of vertical to radial polarization adjustment: vertical helical orbits through 3 o'clock (left) and 9 o'clock (right) snakes, for the 11 different snake current steps. A graph obtained using zplot, stepwise particle data read from zgoubi.plt: menu 7; 1/1 to open zgoubi.plt; 2/[6,4] to choose Z versus s; 7 to plot. An alternate possibility is to use the gnuplot script gnuplot\_Zplt\_sYZ.gnu provided in the sourceforge folder

- **2.1-3:**

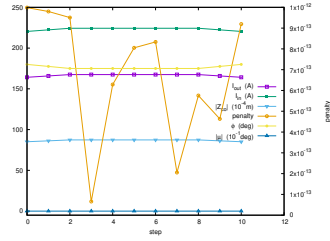
The second test of the spin transparency experiment, vertical spin flip, requires the snake currents given in Tab. 3.2. Graphs of various parameters including snake currents, produced by the gnuplot script `gnuplot_FITVALS.gnu` under the SYSTEM command, are given in Figs. 3.4, 3.5.

**Table 3.2** Snake angles and corresponding currents, for vertical polarization flip

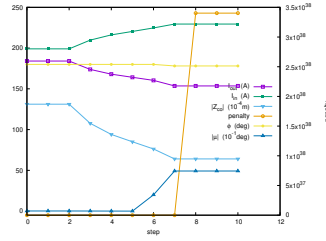
Step #	3 o'clock snake, CCW				9 o'clock snake, CCW			
	$\mu$	$\phi$	$I_{out}$	$I_{in}$	$\mu$	$\phi$	$I_{out}$	$I_{in}$
0	180	0	164.16759	220.45335	180	-10	184.08967	199.07333
1	177.5	0	165.83915	222.40659	180	-10	184.08967	199.07334
2	175	0	167.52137	224.36109	180	-10	184.08966	199.07335
3	175	0	167.52137	224.36111	180	-5	173.87827	209.78901
4	175	0	167.52137	224.36111	180	-2	167.99538	216.19137
5	175	0	167.52134	224.36106	180	0	164.16805	220.45281
6	175	0	167.52141	224.36113	180	2	160.41133	224.71226
7	175	0	167.52137	224.36111	180	5	154.90344	231.10352
8	175	0	167.52135	224.36106	180	10	146.03570	241.81008
9	177.5	0	165.83916	222.40660	180	10	146.03569	241.81009
10	180	0	164.16758	220.45335	180	10	146.03571	241.81005

During the scan, the orbits are logged in `zgoubi.plt` (this is an effect of `IL=2` under TOSCA keywords, or `OPTIONS[.plt,IL=2]`). They are plotted reading from `zgoubi.plt`, Fig. 3.6.

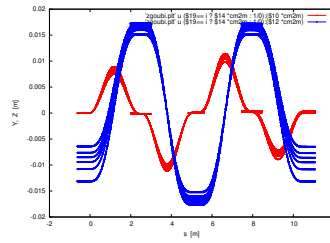
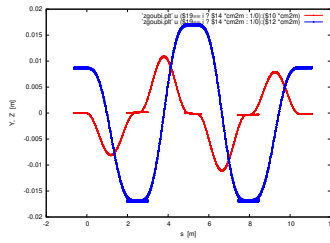




**Fig. 3.4** Vertical polarization flip, case of 3 o'clock snake, step 0 to 10: coil currents, snake angles, closed orbit coordinate at snake entrance, and (right vertical axis) fit penalty



**Fig. 3.5** Vertical polarization flip, case of 9 o'clock snake, step 0 to 10: coil currents, snake angles, closed orbit coordinate at snake entrance, and (right vertical axis) FIT penalty



**Fig. 3.6** Case of vertical polarization flip: : horizontal and vertical helical orbits through 3 o'clock (left) and 9 o'clock (right) snakes, for the 11 different snake current steps.

## Setting Up RHIC Yellow Optics Model

### Question 2.2 series, answers

#### • 2.2-1:

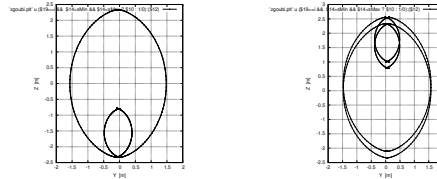
Looking up the field map regions in zgoubi.res (lookup TOSCA for instance), for the respective snake3OCCW\_Segment.inc and snake9OCCW\_Segment.inc jobs, allows finding the azimuthal extent of the field map region, namely,

3 o'clock snake:  $37.001996 < s < 48.840015$  m;

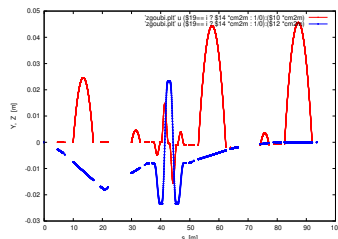
9 o'clock snake:  $37.041440 < s < 48.879459$  m.

These ranges are accounted for in gnuplot script gnuplot\_Zplt\_YZ\_modified.gnu which produces the graph and saves it in a .eps file. The expected results are shown in Fig. 3.7.

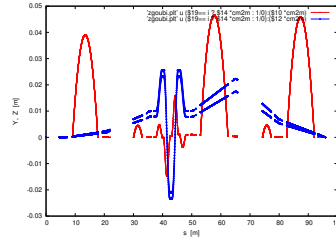
**Fig. 3.7** (Y,Z) projections of the helical orbits in 3 o'clock and 9 o'clock snakes. In the case of the 9 o'clock snake, the initial trajectory was slightly off-center, it shows in addition to the final one following from FIT



Y(s) and Z(s) particle coordinates (before and after FIT on the same graph) are plotted by gnuplot\_Zplt\_sYZ.gnu script, Figs. 3.8, 3.9.



**Fig. 3.8** Horizontal (red) and vertical (blue) local closed orbit bump at 3 o'clock snake. Particle coordinates are in the magnet frame, a Cartesian frame in the case of MULTIPOL, which results in curved Y(s) trajectories along the kicker dipoles



**Fig. 3.9** Horizontal (red) and vertical (blue) local orbit bump at 9 o'clock snake. The initial trajectory (before FIT) and final one (closed orbit, following from FIT) both show on this graph. Using the flag in column 50 of zgoubi.plt would allow to eliminate one or the other

#### • 2.2-2:

Zgoubi (O;X,Y,Z) direct triad is oriented as follows:

- X points in the direction of the motion (CCW),

- Y points outward (as a consequence, the dispersion function  $\eta(s)$  is positive, Fig. 3.10),
- Z points down (correlatively, the magnetic field in bends is positive, curvature is negative).

The complete optical sequence, all INCLUDEs deployed (if any), is logged in the execution listing zgoubi.res, always. Note that the latter can be run as is, namely,

```
cp zgoubi.res blahblah ; [pathTo]/zgoubi-code/zgoubi/zgoubi -in blahblah
```

does work and produces a similar zgoubi.res to blahblah - try it?

The reason is that zgoubi.res starts with a copy of the complete input data file, down to the END command, which ends the execution - whatever follows 'END' is ignored.

Zgoubi execution strictly follows that sequence, the execution won't move to the next keyword until execution of the previous one is completed. This applies as well to FIT, however FIT is an implicit loop on the sequence, only when it is converged, or stopped for boundary reasons, will the execution move beyond FIT keyword, *i.e.* here, FAISCEAU, SPNPRT[MATRIX], etc.

About field modeling: during stepwise integration of motion, whereas TOSCA is used to produce local field  $B(X, Y, Z)$  at particle location by interpolation from a field map [5, p. 135], BEND and MULTIPOL provide  $B(X, Y, Z)$  from built-in analytical field models [5, pp. 74, 122].

The TWISS command (bottom of the main input data file) is an implicit do-loop, it causes a computation of the periodic optical functions, and logs these data in zgoubi.TWISS.out (in a similar format to MADX twiss command output). The header part of zgoubi.TWISS.out is as follows (an excerpt):

```
@ LENGTH      %le  3833.849630
@ ALFA        %le  0.1914832878E-02
@ ORBIT5      %le  -0
@ GAMMATR     %le  22.85254444
@ Q1          %le  0.6910416027 [fractional]
@ Q2          %le  0.7083241286 [fractional]
@ DQ1         %le  9.043350134
@ DQ2         %le  1.501993566
@ DXMAX       %le  1.89765150E+00      @ DXMIN %le  -2.90662088E-01
@ DYMAX       %le  1.96194743E-01      @ DYMIN %le  -1.98209723E-01
@ XCOMAX      %le  1.12487732E-01      @ XCOMIN %le  -2.02526362E-03
@ YCOMAX      %le  0.00000000E+00      @ YCOMIN %le  0.00000000E+00
@ BETXMAX     %le  1.47936731E+02      @ BETXMIN %le  2.4655235E+00
@ BETYMAX     %le  1.63219155E+02      @ BETYMIN %le  3.23642550E+00
@ XCORMS      %le  1.66593610E-04
@ YCORMS      %le  0. not computed
@ DXRMS       %le  6.78335062E-01
@ DYRMS       %le  5.54089006E-02
@ DELTAP      %le  0.00000000E+00
```

A FIT may precede a TWISS command, allowing to accurately find the on-momentum and chromatic closed orbits (otherwise, (i) OBJET[KOBJ=5] is assumed to provide the on-momentum particle 1 with closed orbit coordinates, and (ii) chromatic closed orbit coordinates are deduced from the periodic dispersion which TWISS derives in its first pass through the sequence [5, p. 198]). In the present case the closed orbit is small thus particle 1 coordinates are taken null, with marginal effect on first order transport coefficients computed from particle trajectories.

Additional results of the TWISS command can be found in zgoubi.res listing (bottom of the file). In particular it can be checked there that, at injection rigidity, snakes slightly change the tunes and chromaticities; the 1-turn particle transport mapping, namely,

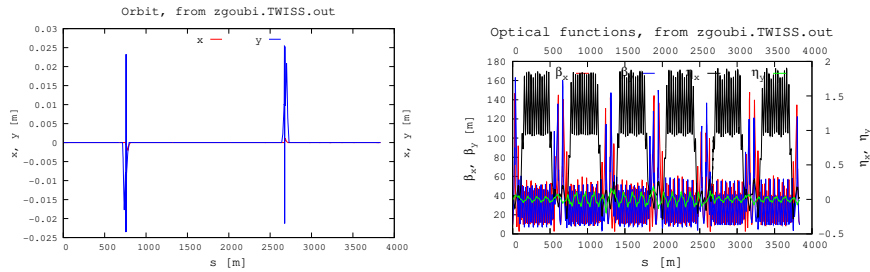
```

TRANSFER MATRIX ORDRE 1 (MKSA units)
-0.194162      -9.31691      -2.586613E-02  -0.137013      0.00000      6.823047E-02
 9.658110E-02  -0.499288      -2.703870E-05  -3.964016E-02  0.00000      9.277359E-04
-4.171478E-02   0.200229      -6.923506E-02  -9.17961      0.00000      1.133383E-02
-5.384989E-03  -1.286964E-02   0.105113      -0.478062      0.00000      4.315142E-03
-7.108407E-03  2.726700E-02  -1.717272E-03  -3.051211E-02  1.00000      7.34140
 0.00000      0.00000      0.00000      0.00000      0.00000      1.00000

```

and its analysis show that snakes also introduce coupling, coupling strength  $C=0.01216169$ .

Once TWISS is done, the execution moves to the next command, `SYSTEM[gnuplot < ./gnuplot_TWISS.gnu]` here, which causes a plot of orbits and optical functions as read from `zgoubi.TWISS.out` (Fig. 3.10).



**Fig. 3.10** Closed orbits and optical functions around RHIC Yellow, counter-clockwise, case of regular snake settings for polarized proton physics runs, *i.e.*  $I_{\text{out}} = \pm 100$  A,  $I_{\text{in}} = \mp 322$  A in both snakes

### • 2.2-3:

Two different ways to obtain the stable spin precession at the start of the optical sequence are,

(i) using `SPNPRT[MATRIX]`, in this case 3 particles need be defined in `OBJET`, all three identical orbit-wise: launched on the closed orbit, on-momentum, but with their respective initial spins aligned on the each of the Serret-Frénet frame axis. Zgoubi input data are as follows:

```

ääZOBJETAääZ
79.366778931425273 * 1d3
2
3 1
0. 0. 0. 0. 0. 1. ääZoaääZ
0. 0. 0. 0. 0. 1. ääZoaääZ
0. 0. 0. 0. 0. 1. ääZoaääZ
1 1 1
ääZPARTICULAääZ
PROTON
ääZSPNTRKääZ
4
1. 0. 0.
0. 1. 0.
0. 0. 1.

```

(ii) using a single particle launched on the closed orbit and `FIT`, to find the periodic spin vector of that particle.

`OBJET` in `pp22-yellow-CCW_spinN0.INC.dat` has been designed to allow the computation of the spin matrix following (i) above: it includes a set of 3 particles with identical initial coordinates (on the on-momentum closed orbit, here), and

orthogonal spins as specified under SPNTRK. However, OBJET also includes a 4th particle, which would be used if the FIT procedure at the end of pp22-yellow-CCW\_spinN0.INC.dat is un-commented; both procedure can be run at once, that allows to verify that the spin precession axis computed by SPNRT[MATRIX] on the one hand, and the spin closed orbit  $\mathbf{n}_0$  computed by FIT do coincide.

The answer is found at the SPNRT[MATRIX] command which follows the optical sequence, just before 'END', namely at the bottom of zgoubi.res:

```
*****
3053 Keyword, label(s) : SPNRT      MATRIX                               IPASS= 1
                        -- 1 GROUPS OF MOMENTA FOLLOW --
                        -----
                        Momentum group #1 (D= 1.000000E+00; particles 1 to 3 ;
Spin components of the 3 particles, spin angles :
      INITIAL          FINAL
      SX      SY      SZ      |S|      SX      SY      SZ      |S|      GAMMA |Si,Sf| (Z,Sf_yz) (Z,Sf)
                        (deg.) (deg.) (deg.)
                        (Sf_yz : projection of Sf on YZ plane)
o 1 1.000000 0.000000 0.000000 1.000000 -0.990752 0.056078 -0.123557 1.000000 25.3786 -172.202 132.321 97.097 1
o 1 0.000000 1.000000 0.000000 1.000000 -0.053248 -0.998240 -0.026097 1.000000 25.3786 -176.600 91.497 91.495 2
o 1 0.000000 0.000000 1.000000 1.000000 -0.124803 -0.019276 0.991994 1.000000 25.3786 7.255 45.005 7.255 3

Spin transfer matrix, momentum group # 1 :
-0.990752 -5.324753E-02 -0.124803
5.607821E-02 -0.998240 -1.927646E-02
-0.123557 -2.609690E-02 0.991994

Determinant = 1.0000000000
Trace = -0.9969976995; spin precession acos((trace-1)/2) = 176.8601850189 deg
Precession axis : (-0.0623, -0.0114, 0.9980) -> angle to (X,Y) plane, to X axis : 86.3712, 93.5696 deg
Spin precession/2pi (or Qs, fractional) : 4.9128E-01
                        -----
Remaining 1 particles ;
Spin components of the 1 particles, spin angles :
      INITIAL          FINAL
      SX      SY      SZ      |S|      SX      SY      SZ      |S|      GAMMA |Si,Sf| (Z,Sf_yz) (Z,Sf)
                        (deg.) (deg.) (deg.)
                        (Sf_yz : projection of Sf on YZ plane)
o 1 -0.062259 -0.011371 0.997990 0.999994 -0.062264 -0.011378 0.997995 1.000000 25.3786 0.000 45.002 3.629 4
*****
```

What we see there:

- the first 3 particles have been used to compute the “Spin transfer matrix, momentum group # 1”. A sub-product of this matrix calculation is the stable spin precession direction:

$$\text{Precession axis : } (-0.0623, -0.0114, 0.9980)$$

- the 4th particle is just let go through the ring. However it has been launched with its spin aligned on the stable precession axis (SPNTRK command at the beginning of pp22-yellow-CCW\_spinN0.INC.dat), thus it ends up with the same spin coordinates after 1 turn around the ring, see the bottom in zgoubi.res excerpt lines above.

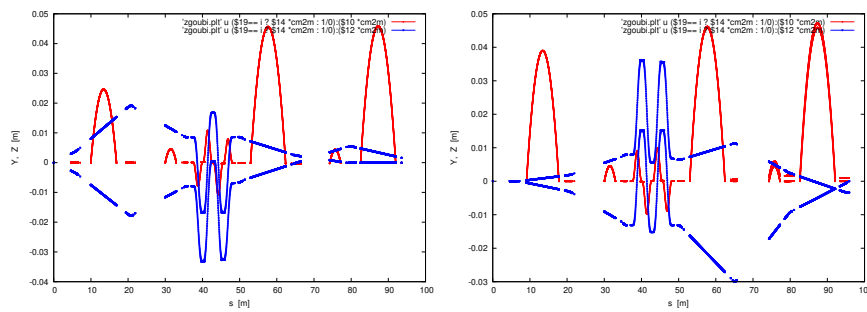
## Spin Transparency Simulations

### Vertical to radial polarization

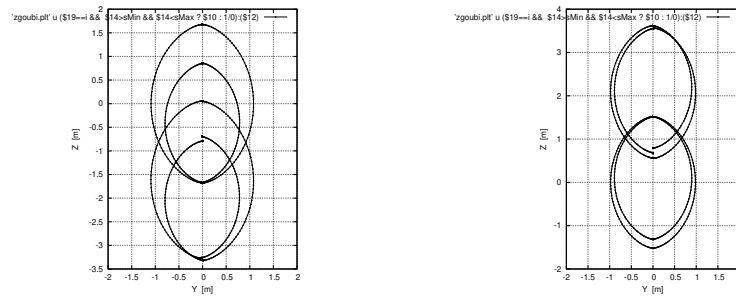
- Question 2.3.1 series, answers

- 2.3.1-1:

Snake currents are set to the values needed at step 0 of the vertical to radial polarization adjustment. The gnuplot scripts called by the SYSTEM command save the graphs in .eps files. The outcomes are shown in Figs. 3.11, 3.12.



**Fig. 3.11** Horizontal (red) and vertical (blue) local orbit bumps at 3 o'clock and 9 o'clock snakes. Orbits resulting from kicker values at start and end of FIT are plotted



**Fig. 3.12** (Y,Z) projections of the helical orbits in 3 o'clock and 9 o'clock snakes. Off-centered helices at start of FIT (*i.e.*, with orbit kickers set for the earlier 100A/322A snake currents) are displayed together with the centered helices resulting from kicker values adjusted for the new snake currents

The orbits inside the snakes are part of the data logged in zgoubi.plt. Looking up the field map regions in zgoubi.res (lookup “ TOSCA “, for instance), for the respective snake3OCCW\_Segment.inc and snake9OCCW\_Segment.inc jobs, allows finding the azimuthal extent of the field map region, namely,

$$3 \text{ o'clock snake: } 37.001996 < s < 48.840015 \text{ m,}$$

9 o'clock snake':  $37.041440 < s < 48.879459$  m.

These ranges are accounted for in gnuplot script `gnuplot_Zplt_YZ.gnu` which produces the graph and saves it in a .eps file. The expected results are shown in Fig. 3.11.

An excerpt of SCALING.inc files, with proper orbit kicker settings:

```

MULTIPOL Y09_TV5 ! Next 6 elements: 9 o'clock snake orbit kickers.
-1
1 1.7905968411E+05
MULTIPOL Y09_TV7
-1
1 3.6564632690E+05
MULTIPOL Y09_TH8
-1
1 6.4252488673E+03
MULTIPOL Y09_TV9
-1
1 5.2125941079E+04
MULTIPOL Y09_TH10
-1
1 6.7996064758E+03
MULTIPOL Y09_TV11
-1
1 5.0732776649E+05
1

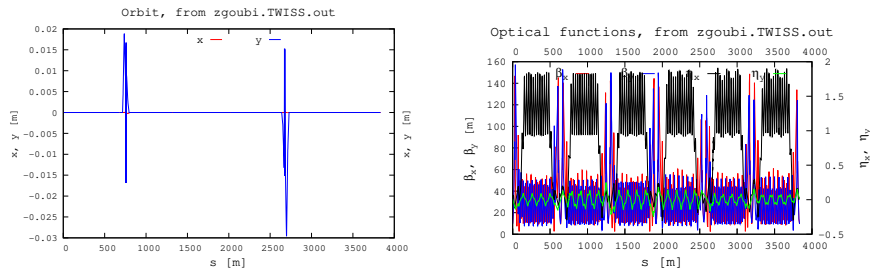
MULTIPOL YI3_TV4 ! Next 6 elements: 3 o'clock snake orbit kickers.
-1
1 -5.0364070518E+05
MULTIPOL YI3_TV6
-1
1 -2.4489762740E+05
MULTIPOL YI3_TV8
-1
1 -2.9093216368E+05
MULTIPOL YI3_TH9
-1
1 1.9964797598E+03
MULTIPOL YI3_TV10
-1
1 -1.3881781797E+05
MULTIPOL YI3_TH11
-1
1 4.1869407887E+03
1

```

The file `zgoubi.FIT.out.dat`: an input data file, produced by FIT once done. It is a replica of the original input data file, updated with variable values found by FIT.

### • 2.3.1-2:

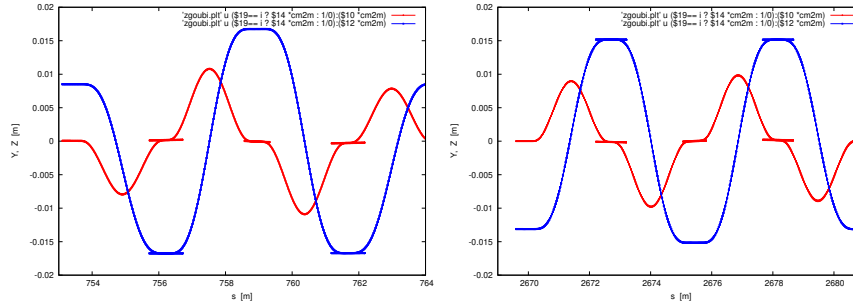
Running the job `pp22-yellow-CCW_spinN0.INC.dat` produces the orbits and optical functions displayed in Fig. 3.13.



**Fig. 3.13** Closed orbits and optical functions around RHIC Yellow, counter-clockwise. This accounts for step 0 of the snake settings for vertical to radial polarization adjustment; 3 o'clock snake:  $I_{out} = 164.16$  A,  $I_{in} = 220.455$ , 9 o'clock snake:  $I_{out} = 184.08$  A,  $I_{in} = 199.07$ .

Because the option `IL=2` under TOSCA keywords has been maintained (look up “TOSCA” in `snake3OCCW_Segment.inc` and `snake9OCCW_Segment.inc` files), particle data are still logged in `zgoubi.plt` when running `pp22-yellow-CCW.INC.dat`. Thus the local vertical orbit can be plotted, reading from `zgoubi.plt`. The script `gnuplot_Zplt_sYZ.gnu` is used for that; the orbital coordinate range for each snake is read from `zgoubi.res`, namely (look up “TOSCA” in `zgoubi.res`):

3 o'clock snake :  $753.10910 < s < 764.781084$  m;    9 o'clock snake :  $2669.60052 < s < 2681.27252$  m



**Fig. 3.14** Horizontal (red) and vertical (blue) helical orbits, centered, across 3 o'clock and 9 o'clock snake, at step 0 of the spin transparency experiment

### • 2.3.1-3:

The job pp22-yellow-CCW\_spinN0.INC.dat computes the 1-turn spin matrix. The command SPNPRT[MATRIX] performs that computation, from spin coordinates. For SPNPRT[MATRIX] to work, the transport of the 3 different spins: (1,0,0), (0,1,0) and (0,0,1), is required, this requires in turn OBJET to define 3 times the on-orbit particle, and SPNTRK to assign these different initial spin values to the three particles.

The results are found under SPNPRT, bottom of zgoubi.res execution listing:

```

Spin transfer matrix, momentum group # 1 :

    0.926849    0.348378    0.139942
   -0.344618    0.937354   -5.105082E-02
   -0.148960   -9.101234E-04    0.988843

Determinant =      1.0000000000
Trace =      2.8530455354;   spin precession acos((trace-1)/2) =      22.1008928920 deg
Precession axis : ( 0.0666,  0.3839, -0.9210) -> angle to (X,Y) plane, to X axis :   -67.0658,    80.1540 deg
Spin precession/2pi (or Qs, fractional) :    6.1391E-02

```

The FIT procedure works on particle number 4, as the first 3 particles can not be touched (they are used for 1-turn spin matrix computation by SPNPRT). FIT constrains the spin vector associated with particle 4 to be periodic: initial coordinates identical to initial ones. For this it varies the initial coordinates.

FIT is commented as it happens to be redundant with SPNPRT[MATRIX] computing the spin closed orbit from the spin matrix. Anyway, FIT can be run for cross-check with the latter, letting it go yields the following final FIT state (there are three possible ways to find the FIT status as copy-pasted below: it is saved, by the FIT command, in fitVals\_n0\_IP6; lookup "FIT" in zgoubi.res listing; it appears on the computer screen in real-time as FIT is running).



---

```

FIT variables and constraints in good order, FIT will proceed.
STATUS OF VARIABLES (Iteration # 0 / 999 max.)
LMNT VAR PARAM MINIMUM INITIAL FINAL MAXIMUM STEP NAME
3 1 40 -1.11 0.00 -6.66319420E-02 1.11 2.736E-05 SPNTRK
3 2 41 -1.11 0.00 -0.38393333 1.11 2.764E-05 SPNTRK
3 3 42 -1.11 1.00 0.92096090 1.11 2.319E-05 SPNTRK
STATUS OF CONSTRAINTS (Target penalty = 1.0000E-10)
TYPE I J LMNT# DESIRED WEIGHT REACHED KI2 NAME
10 4 1 3050 0.000000E+00 1.000E+00 1.704904E-06 Infinity OPTIONS
10 4 2 3050 0.000000E+00 1.000E+00 1.384296E-06 Infinity OPTIONS
10 4 3 3050 0.000000E+00 1.000E+00 6.675809E-06 Infinity OPTIONS
10 4 4 3050 1.000000E+00 1.000E-01 1.000000E+00 NaN OPTIONS
Fit reached penalty value 4.9389E-11

```

---

The periodic spin vector found by FIT appears to be (FINAL column in the spin status above)

$$\text{periodic } \mathbf{S} = (-0.0666319, -0.383933, 0.920960)$$

and does coincide with the spin precession axis as computed by SPNPRT[MATRIX], namely,

$$\mathbf{n}_0 = (0.0666, 0.3839, -0.9210)$$

• **2.3.1-4:**

The 1-turn spin mapping is computed by SPNPRT[MATRIX], taking the optical sequence as a whole, and assuming it periodic. As a consequence it is necessary to move “MARKER’ YO12\_POL3.INST”, which is the location of the pC polarimeters, to the end of the optical sequence.

This requires,

(i) splitting pp22-yellow-CCW.INC.dat[DRIFT\_250:YO9\_QS4] at the polarimeter, yielding two files,

pp22-yellow-CCW.INC.dat[DRIFT\_250:YO12\_POL3.INST] and

pp22-yellow-CCW.INC.dat[YO12\_POL3.INST:YO9\_QS4]

, (ii) circularly permuting the INCLUDE statements to bring pp22-yellow-CCW.INC.dat[YO12\_POL3.INST:YO9\_QS4] to be the first INCLUDE of the Yellow ring series, starting at pC polarimeter.

This yields the file

pp22-yellow-CCW\_spinN0\_origin-pCPol.INC.dat

available in the same folder, namely,

[pathTo]/zgoubi-code/exemples/tutorials/BNLSpinTransparency\_2022/spinTransparency\_1\_adjustRadialPolarisation

The results are found under SPNPRT, bottom of zgoubi.res execution listing:

```

Spin transfer matrix, momentum group # 1 :

0.928391      -0.347719      -0.131077
0.338901      0.936958      -8.517969E-02
0.152432      3.465804E-02      0.987706

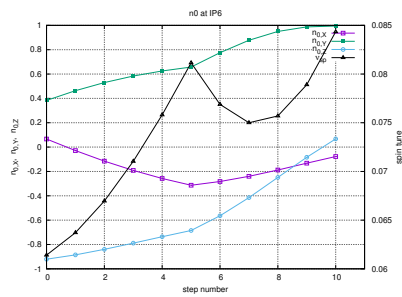
Determinant =      1.0000000000
Trace =      2.8530556616; spin precession acos((trace-1)/2) =      22.1001218426 deg
Precession axis : ( 0.1593, -0.3768, 0.9125) -> angle to (X,Y) plane, to X axis :      65.8544,      -67.0865 deg
Spin precession/2pi (or Qs, fractional) :      6.1389E-02

```

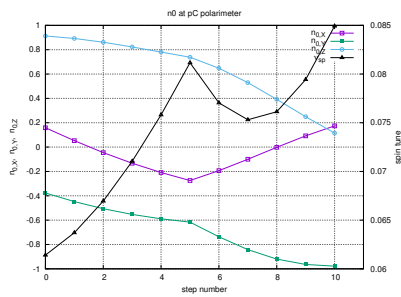
### • 2.3.1-5:

Make sure the END statement preceding REBELOTE at the bottom of pp22-yellow-CCW\_spinN0.INC.dat is commented, this will allow the execution to carry on and execute the REBELOTE command.

The data for a graph are extracted from the execution listing zgoubi.res, using linux “grep” command. This is done by the gnuplot script gnuplot\_ZRes\_spinPrecAxis.gnu, results are given in Figs. 3.15, 3.16..



**Fig. 3.15** Components of the stable spin precession direction at IP6, and spin tune, at the 11 different steps of the vertical to radial polarization adjustment scan



**Fig. 3.16** Components of the stable spin precession direction at pC-pol polarimeter in IR12, and spin tune, at the 11 different steps of the vertical to radial polarization adjustment scan

### • 2.3.1-6:

First the earlier input data file pp22-yellow-CCW\_spinN0.INC.dat is saved to sweep\_origin-pCPol\_0-10.INC.dat, as a starting point. This file can also be downloaded from sourceforge. A few things need to be changed in order to perform the ramp, namely,

- OBJET is set to launch a single particle:

```

ãĀZOBJETãĀZ
79.366778931425273 * 1d3
2
1 1
0. 0.
0. 0.
0. 1. ãĀZoãĀZ
1

```

- the initial spin component is set to  $n_0$  values (Tab. 3.1):

```

âĀZSPNTRKâĀZ
4
-5.93769E-02 -0.42047 0.90535

```

- the file name under FAISTORE: remove any  $\hat{a}Ij_b\hat{a}I$  prefix, to allow easy use of gnuplot. FAISTORE[zgoubi.fai] is adopted here: turn by turn particle data will be logged in zgoubi.fai. The TOSCA components under SCALING are

```

TOSCA snk1LowB ! snk1 = 9'o
11
184.08966 184.08967 184.08966 184.08966 184.08967 184.08966 178.43276 172.63753 166.69559 160.59563 154.32192
1 501 1001 1501 2001 2501 3001 3501 4001 4501 5001
TOSCA snk1HighB ! snk1 = 9'o
11
199.07334 199.07334 199.07336 199.07335 199.07334 199.07334 200.72111 202.52287 204.46703 206.54226 208.73700
1 501 1001 1501 2001 2501 3001 3501 4001 4501 5001
TOSCA snk2LowB ! snk2 = 3'o
11
164.16759 166.17474 168.19734 170.23614 172.29233 174.36709 174.36707 174.36709 174.36712 174.36710 174.36709
1 501 1001 1501 2001 2501 3001 3501 4001 4501 5001
TOSCA snk2HighB ! snk2 = 3'o
11
220.45335 222.79739 225.14344 227.49256 229.84602 232.20533 232.20531 232.20531 232.20540 232.20535 232.20534
1 501 1001 1501 2001 2501 3001 3501 4001 4501 5001

```

It is advised to create a modified version of the earlier, e.g. SCALING\_radialPolarization\_sweep.inc, with these changes included. The sequence is directly followed by a REBELOTE loop for multiturn tracking, and a request for particle (FAISCEAU) and spin coordinate (SPNPR) print out in the execution listing zgoubi.res:

```

âĀZREBELOTEâĀZ
5000 0.3 99
âĀZFAISCEAUâĀZ
âĀZSPNPRâĀZ
âĀZENDâĀZ

```

Note: the execution can be speeded up by setting  $IL=0$  under TOSCA keyword in the snake segment files (or using  $OPTIONS[.plt,IL=0]$ ): that avoids stepwise particle data storage (in zgoubi.plt) during execution, which would eat up CPU time (file I/Os are much time consuming).

A Fortran program, spinTuneFromFai\_iterate, is available in zgoubi toolbox to compute the turn-by-turn spin motion DFT. Its input file should be

```

1 5001 100 98 ! first turn, last turn, turn step, DFT sampling
0.0 0.499999 ! spin tune spectrum range
2500 ! nbin ! spectrum sampling over the spin tune spectrum range
zgoubi.fai ! input data file name

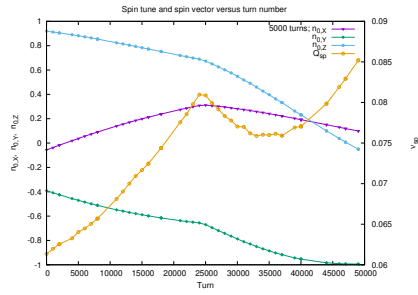
```

Outcomes of this job are logged in spinTuneFromFai\_iterate.Out, and can be plotted using the following gnuplot script, to produce the requested graph (displayed in Fig. 19):

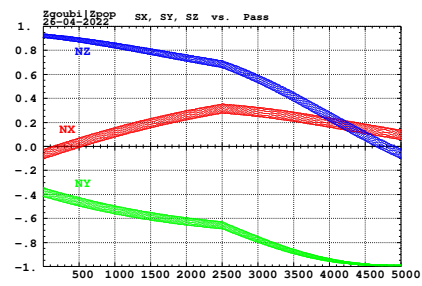
```

# gnuplot_spinTuneFromFai.gnu
set tit "Spin tune and spin vector versus turn number "
set xlabel "Turn"; set ylabel "n_{0,X}, n_{0,Y}, n_{0,Z}, Q_{sp}"
set xtics; set ytics nomirror; set key maxcol 1
set xrange [1:6000]; set yrange [-1.:1.]
plot \
"spinTuneFromFai_iterate.Out" u (10* $8):($2) w lp pt 11 tit "5000 turns; n_{0,X}" ,\
"spinTuneFromFai_iterate.Out" u (10* $8):($3) w lp pt 13 tit "n_{0,Y}" ,\
"spinTuneFromFai_iterate.Out" u (10* $8):($4) w lp pt 15 tit "n_{0,Z}" ,\
"spinTuneFromFai_iterate.Out" u (10* $8):($1) w lp pt 17 ps .8 tit "Q_{sp}"
pause 10

```



**Fig. 3.17** Evolution of spin  $\mathbf{n}_0$  vector coordinates at pC-polarimeter (left vertical axis) and spin tune (from a DFT of turn-by-turn spin motion - right axis) over 50,000 turns tracking.



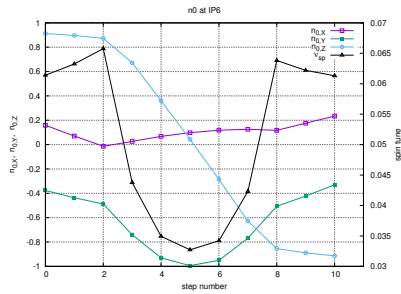
**Fig. 3.18** Turn-by-turn spin components at pCpolarimeter, from zgoubi.fai, using zpop

### Vertical polarization flip

- Question 2.3.2 series, answers

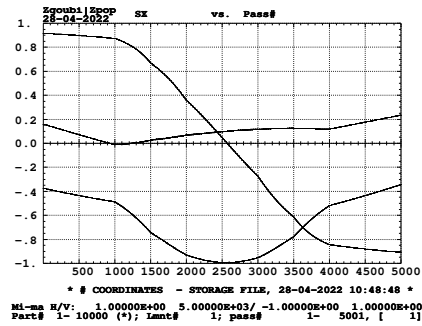
- 2.3.2-1:

The data for a graph are extracted from the execution listing zgoubi.res, using linux “grep” command. This is done by the gnuplot script gnuplot\_ZRes\_spinPrecAxis.gnu, results are given in Fig. 3.15.



**Fig. 3.19** Components of the stable spin precession direction at pC-pol polarimeter in IR12, and spin tune, at the 11 different steps of the vertical polarization flip scan. Solid lines are to guide the eye

- 2.3.2-2:



**Fig. 3.20** Components of the stable spin precession direction at pC-pol polarimeter in IR12, and spin tune, over a 5000-turn continuous ramp of snake currents from step 0 to step 10. Data plotted from zgoubi.fai, using zpop

## References

1. Y.S. Derbenev, et als.: Siberian Snakes, Figure-8 and Spin Transparency Techniques for High Precision Experiments with Polarized Hadron Beams in Colliders. *Symmetry* 2021, 13, 398. <https://doi.org/10.3390/sym13030398>
2. V. Morozov, et al.: Experimental verification of transparent spin mode in RHIC. <https://accelconf.web.cern.ch/ipac2019/papers/wepgw122.pdf>
3. H. Huang, et al.: SIMULATION OF TRANSPARENT SPIN EXPERIMENT IN RHIC. <https://accelconf.web.cern.ch/napac2019/papers/wepls11.pdf>
4. F. Méot, et al.: Re-visiting RHIC snakes: OPERA fields,  $n_0$  dance. C-A/AP/590; BNL-114379-2017-IR (Sept. 2017). <https://technotes.bnl.gov/PDF?publicationId=42159>
5. Zgoubi Users' Guide: <https://sourceforge.net/p/zgoubi/code/HEAD/tree/trunk/guide/Zgoubi.pdf>