

Stern-Gerlach Elegant Simulation

Katherine Woodruff

kwoodruf@nmsu.edu

Sebastian Szustkowski

sszustkowski@gmail.com

This report/tutorial is the result of a course on electrostatic storage rings at the January 2017 U.S. Particle Accelerator School (USPAS) hosted by the University of California, Davis. The goal of this lattice investigation is to determine how much deflection caused by the Stern-Gerlach effect we would be able to induce using quadrupoles and a solenoid in an electrostatic storage ring. We would like to acknowledge the instructors, Dr. Richard Talman at Cornell University and Dr. Jayakar C.T. Thangaraj at Fermilab, for all of their expertise and instruction.

1 How to run elegant

If elegant is fully installed, only two files are needed to run a simulation: a lattice file (.lte) and a command file (.ele). The lattice file is the description of the elements in the beamline, and the command file describes the actions of the simulation. To run the simulation, execute:

```
1 elegant <mycommandfile>.ele
```

in a bash session.

1.1 Lattice file

The configuration of each lattice element is defined in the lattice file. The provided “SG_example.lte” example file describes the quadrupole layout from section 11.4.1 in the class text before a solenoid was added. Here we briefly describe the elements in the example lattice file.

Quadrupoles:

```
5 % 8 sto qslices      ! number of quad slices
6
7 dB1: quad, l="0.004 qslices /", k1="-362.0 0.004 /"
8 dB2: quad, l="0.004 qslices /", k1="225.0 0.004 /"
9 dB3: quad, l="0.004 qslices /", k1="-59.0 0.004 /"
10
11 B1: LINE=(8*dB1)
12 B2: LINE=(8*dB2)
13 B3: LINE=(8*dB3)
```

Quadrupoles are defined by their “geometric strength”, k_1 , and length, l . In the example we define a small segment of the quadrupole and repeat it. The variable “qslices” is the number of segments we slice the quads into. The segmentation allows us to changes in the beam parameters on a finer scale. The elements dB1-dB3 are the quad segments, and the elements B1-B3 are the full quads.

Drift spaces:

```
17 lseg: drift, l=0.0005
18
19 L0: LINE=(16*lseg) ! L0 drift l=0.008
20 L1: LINE=(26*lseg) ! L1 drift l=0.013
21 L2: LINE=(160*lseg) ! L2 drift l=0.08
```

Drift spaces are defined by their length, l . In the example we slice the drifts into 5 mm segments ($lseg$). The three lengths of drift that we use here are 0.8 cm, 13 cm, and 8 cm.

Beamline:

```
26 SG: LINE=(q,START,L0,B1,B2,L1,B3,L2,B3,L1,B2,B1,L0,END)
27 USE,SG
```

The lattice is defined as a LINE element with each sub-element listed in order.

1.2 Command file

The command file is used to run the simulation. In the example, it is “SG_example.ele”. The total length of our beamline is 15 cm.

```
1 &run_setup
2     lattice="SG_example.lte",
3     use_beamline="SG"
4
5     default_order = 3
6     sigma="%s.s",
7     final="%s.fin",
8     output="%s.out",
9     magnets="%s.mag"
10    centroid="%s.cen"
11    p_central= 1.732
12 &end
```

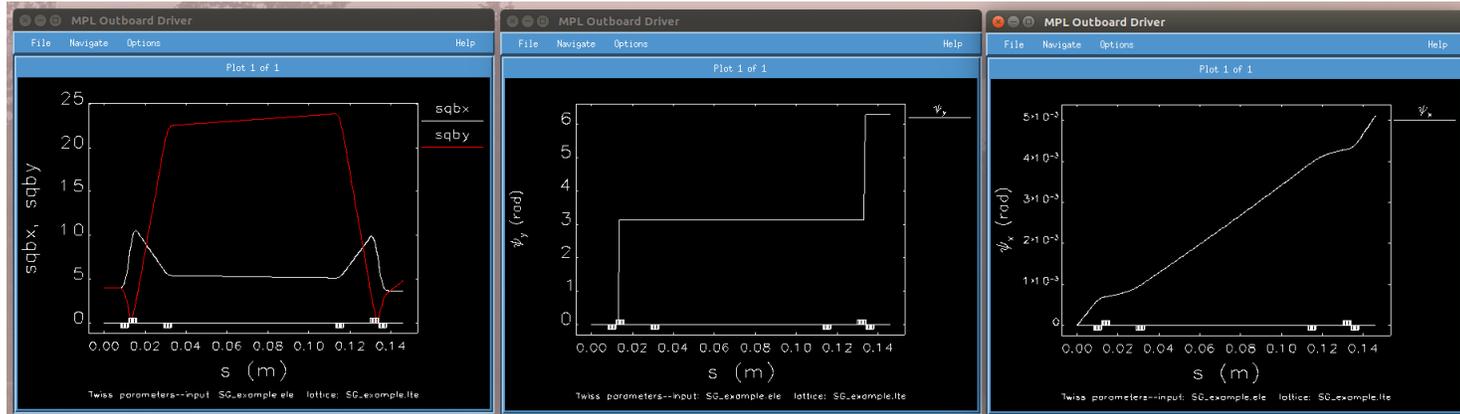
The “run_setup” section defines which lattice file to use, what the beamline is called, the central momentum, and some of the output files to create. Other sections can be added to define more parameters.

1.3 Example script

To run the example, execute the “SG-example” bash script.

```
1 ./SG_example
```

The script runs the simulation and plots the β -function and phase advance along the beamline. Three plots should pop up:



SDDS Toolkit was used to access, process, and plot the data that is saved from the elegant simulation. In the elegant command (.ele) file, there is a section to define which twiss parameters to output and where. In this example, the output file containing the twiss parameters is “SG_example.twi”.

```
28 &twiss_output
29     matched=0
30     concat_order = 1,      ! for speed
31     alpha_x= 0
```

```

32     alpha_y= 0
33     beta_x=  16.0
34     beta_y=  16.0
35     filename="%s.twi"
36 &end

```

1.3.1 Plotting simulation variables

We can read the .twi and plot the values that we are interested in using sddsplot in the bash script:

```

13 exec sddsplot -column=s,psiy SG_example_1.twi -legend -graph=line ,vary -unsuppress=y -
      column=s,Profile SG_example.mag -overlay=xmode=normal ,yfactor=0.04 &
14 exec sddsplot -column=s,psix SG_example_1.twi -legend -graph=line ,vary -unsuppress=y -
      column=s,Profile SG_example.mag -overlay=xmode=normal ,yfactor=0.04 &

```

We also want to plot the square root of the β -functions, which isn't directly saved by elegant. We use sddsprocess to create and save new variables from the existing ones.

```

7  exec sddsprocess SG_example.twi SG_example_1.twi "-define=column,sqbx,betax sqrt" \
8  "-define=column,sqby,betay sqrt" \

```

Note that SDDS uses “reverse polish notation” like elegant. These new variables are written back into the .twi output file and can be accessed by sddsplot:

```

12 exec sddsplot -column=s,sqb? SG_example_1.twi -legend -graph=line ,vary -unsuppress=y -
      column=s,Profile SG_example.mag -overlay=xmode=normal ,yfactor=0.04 &

```

2 Stern-Gerlach elegant simulation

We start with an existing six-quadrupole beamline design. This original design assumes zero-length quadrupoles and no solenoid focusing. Our goals for this investigation are:

1. Replicate the quadrupole beamline design using actual quadrupole lengths.
2. Introduce a solenoid that will flip the spins of transversely polarized electrons.
3. Retune the optics to account for the solenoid focusing.

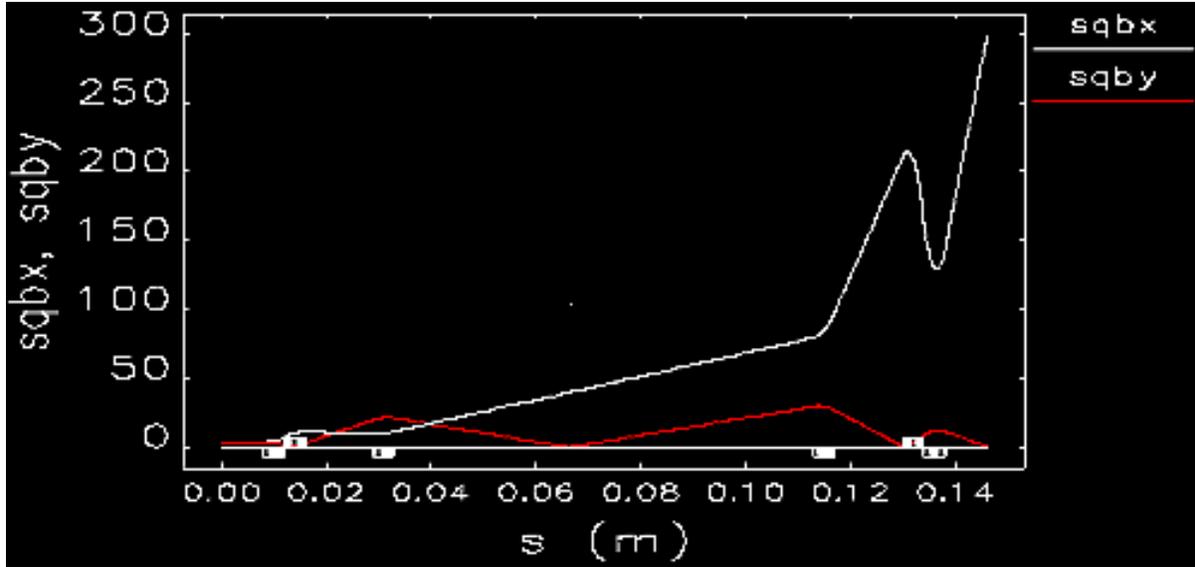
2.1 Replacing the zero-length quadrupoles

The six zero-length quadrupoles in the existing design are placed to “magnify the Stern-Gerlach displacement without excessively increasing the transverse beam dimensions.” Our goal is to place actual-length quadrupoles into the design and tune the strength and positions to keep similar S-G displacement magnification and the transverse beam dimensions.

The existing design parameters are:

quad label	center pos. cm	quad length mm	inv. focal length m^{-1}
B1	1.0	4	-362.73
B2	1.4	4	195.648
B3	2.4	4	-90.9039
C1	24.4	4	-90.9039
C2	25.4	4	195.648
C3	25.8	4	-362.73

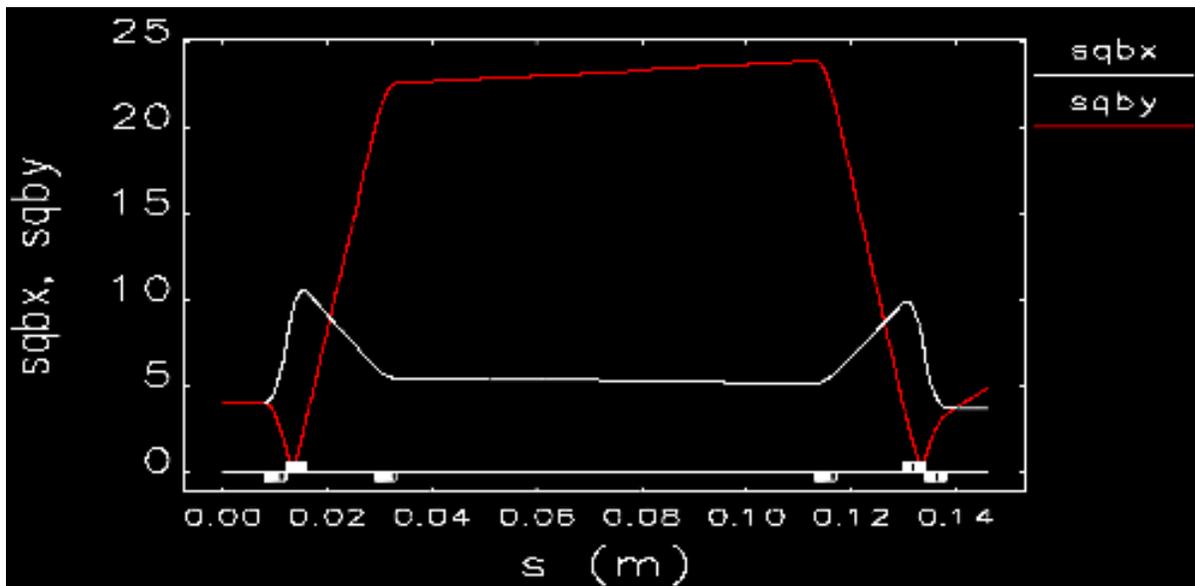
We first enter the design parameters exactly as given and plot the square root of the x and y β -functions. While β_y looks reasonable, β_x blows up after the long drift in the center of the beamline.



To get the β -functions to look like the original design, we tune the focal lengths of the quadrupoles and the drift distance between quads B2 and B3 (also C1 and C2 to retain symmetry). The tuned parameters are shown below. The bold items in the table are tuned values.

quad label	center pos. cm	quad length mm	inv. focal length m^{-1}
B1	1.0	4	-362.0
B2	1.4	4	225.0
B3	3.1	4	-59.0
C1	11.5	4	-59.0
C2	13.2	4	225.0
C3	13.6	4	-362.0

The resulting β -functions look more symmetric and not excessively large.



2.2 Adding the solenoid

In the original design of the beamline, the focusing of the solenoid was assumed to be small. Without changing the strength or positions of the quads, we add an 8 cm long solenoid in the central drift space. The magnetic field of the solenoid was calculated to exactly flip the spin of the electrons,

$$\begin{aligned}
 B_\pi &= \gamma\pi \frac{1}{e/m_e} \frac{v}{L_s} \\
 &= 2\pi \frac{1}{1.759820 \cdot 10^{11}} \frac{0.863 \cdot 2.99 \cdot 10^8}{0.08} \\
 &= 0.1158 \text{ T} .
 \end{aligned}$$

To add the solenoid to the elegant simulation, we just add a SOLE object to the lattice file and place in the lattice sequence:

```

24 dS0: sole , l=0.01, B=0.115870513
25 S0: LINE=(8*dS0)

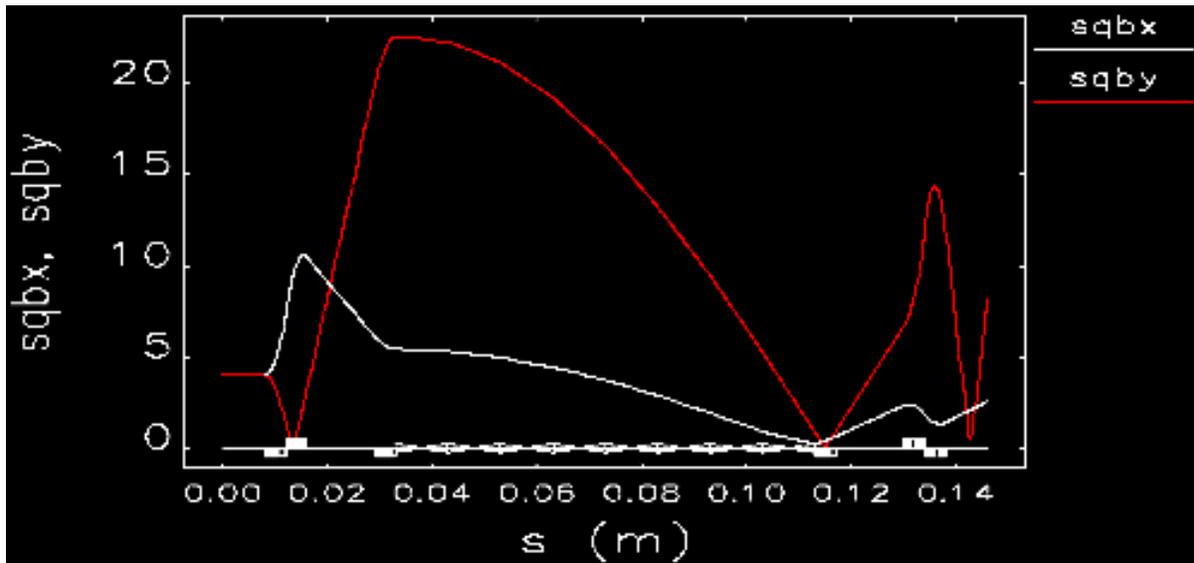
```

```

30 SG: LINE=(q,START,L0,B1,B2,L1,B3,S0,B3,L1,B2,B1,L0,END)

```

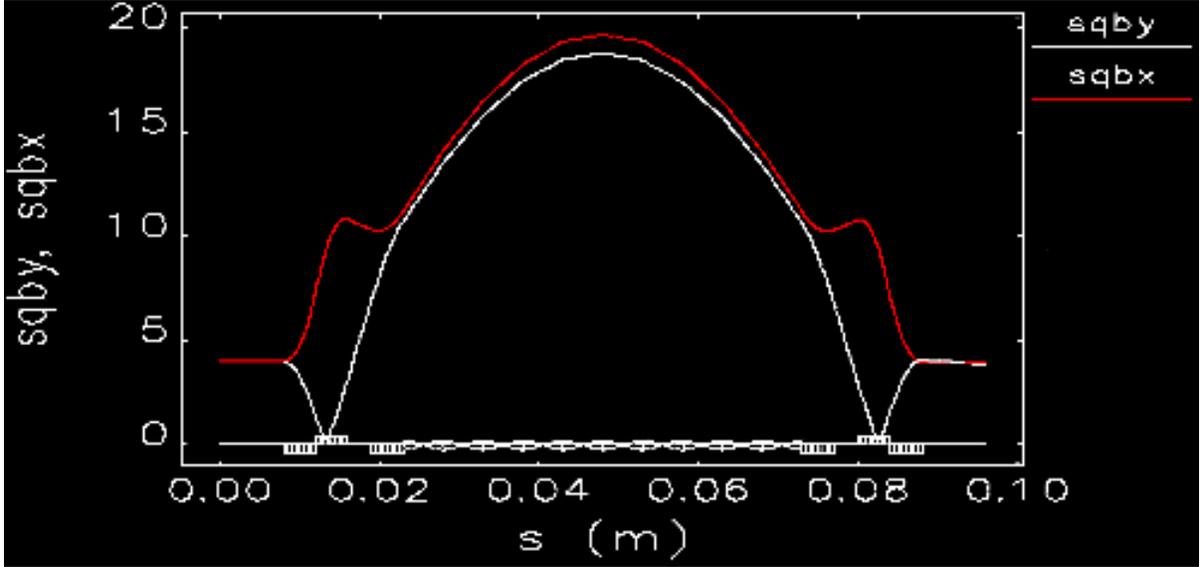
The focusing from the solenoid destroyed the β -functions.



One solution that we try is to try to match β_x and β_y at the point right before the solenoid, so that they would be focused the same amount and we could keep the quads symmetric. We match the β -functions by adjusting the strength of the quads and the position of the third quad (B3). We then adjust the magnetic field of the solenoid so that the magnitude of β at would be the same at the beginning and end of the solenoid. The new parameters are shown here (the solenoid magnetic field is given in place of focal length):

quad label	center pos. cm	quad length mm	inv. focal length m^{-1}
B1	1.0	4	-362.0
B2	1.4	4	205.0
B3	2.1	4	-75.5
SOL	4.8	50.0	(0.2280 T)
C1	7.5	4	-75.5
C2	8.2	4	205.0
C3	8.6	4	-362.0

The new parameters give us symmetric and reasonably-sized β -functions.



The transverse spin precession caused by the solenoid is

$$\begin{aligned}
 \theta_s &= \frac{1}{2} \frac{e}{\gamma m_e} B \frac{L_s}{v} \\
 &= \frac{1}{2} 1.759820 \cdot 10^{11} \cdot 0.2280 \frac{0.08}{0.863 \cdot 2.99 \cdot 10^8} \\
 &= 3.8 \text{ rad } (1.2\pi)
 \end{aligned}$$

2.2.1 Compare solenoid transfer matrices

To make sure that we understand how elegant was handling the solenoid, we calculate the elements of the transfer matrix for a solenoid segment. The transfer matrix for a solenoid is:

$$R_{sole} = \begin{pmatrix} C^2 & \frac{SC}{K} & SC & \frac{S^2}{K} \\ -KSC & C^2 & -KS^2 & SC \\ -SC & -\frac{S^2}{K} & C^2 & \frac{SC}{K} \\ KS^2 & -SC & -KSC & C^2 \end{pmatrix},$$

where S is $\sin(KL)$, C is $\cos(KL)$, and $K = -\frac{B_0}{2B\rho}$. We use $B_0 = 0.2280$ T for the magnetic field of the solenoid, $L = 5$ mm for the length of the solenoid segment (the total solenoid length is 5 cm), and $B\rho = 2.918 \cdot 10^{-3}$ Tm for our beam. We get

$$K = -39.068$$

$$C = 0.9810$$

$$S = 0.1941$$

which makes the transfer matrix

$$R_{sole} = \begin{pmatrix} 0.9624 & 0.0049 & -0.1904 & -0.0010 \\ -7.4390 & 0.9624 & 1.4719 & -0.1904 \\ 0.1904 & 0.0010 & 0.9624 & 0.0049 \\ -1.4719 & 0.1904 & -7.4390 & 0.9624 \end{pmatrix}.$$

The “matrix_output” section in the command (.ele) file contains the settings for what matrix information to save.

```

20 &matrix_output
21     printout = "%s.mpr"
22     printout_order = 1
23     individual_matrices = 1
24     SDDS_output = "%s.mat"
25     full_matrix_only = 0
26 &end

```

Setting “individual_matrices” to 1 on line 23 tells elegant to save the transfer matrix for every single lattice element instead of the full matrix only. The matrices are saved to “SG_example.mpr” which is a text file that can be opened and read without any special processing.

The solenoid transfer matrix given by elegant is

$$R_{elegant} = \begin{pmatrix} 0.9632 & 0.0049 & -0.1883 & -0.0010 \\ -7.2738 & 0.9632 & 1.4223 & -0.1883 \\ 0.1883 & 0.0010 & 0.9632 & 0.0049 \\ -1.4223 & 0.1883 & -7.2738 & 0.9632 \end{pmatrix},$$

which closely agrees with R_{sole} — each element in $R_{elegant}$ is within 4% of the corresponding element in R_{sole} .

We also investigated changing the strength of the solenoid magnetic field to get specific spin rotations. This is detailed in appendix A.

3 Determine Stern-Gerlach deflection

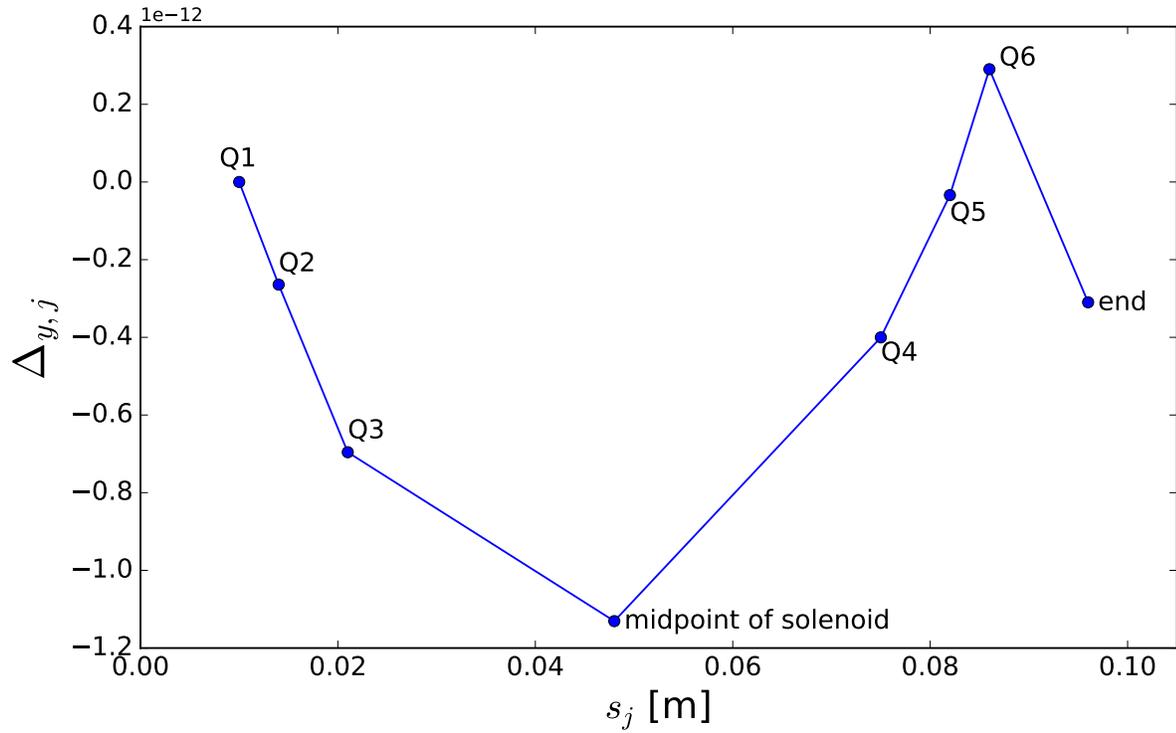
The equation for the Stern-Gerlach displacement at a point j due to the displacement at a point i is

$$\Delta_{y,j} = q_{y,i} (1.93 \cdot 10^{-13} \text{m}) \sqrt{\beta_{y,j} \beta_{y,i}} \sin(\psi_{y,j} - \psi_{y,i}).$$

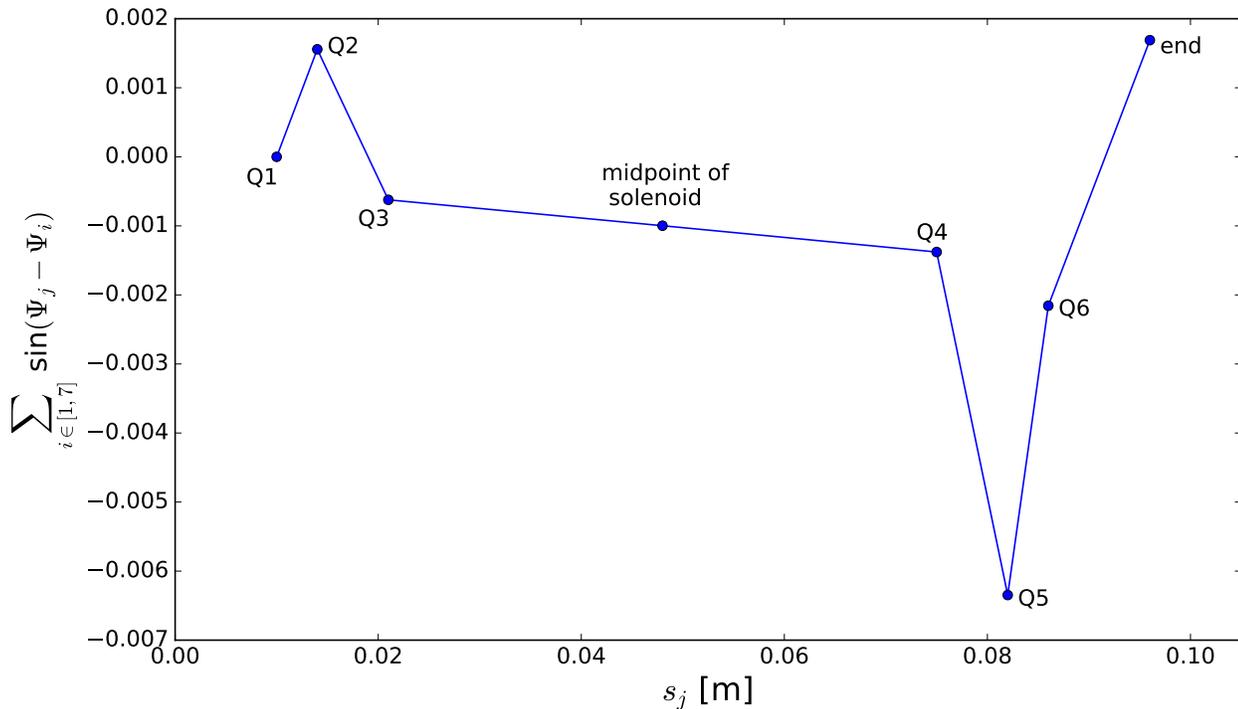
The values of β_y and ψ_y can be output for each of the longitudinal positions in the elegant simulation, so we can calculate $\Delta_{y,j}$ at any j . We evaluate the displacement at each quad and at the midpoint and end of the beamline

$$\Delta_y = \begin{pmatrix} 0.0 & -1.3695 & -5.1148 & -7.6812 & -2.3291 & 0.8013 & 2.7370 & 0.1530 \\ 0.0 & 0.0 & 1.5093 & 3.4493 & 1.8367 & 0.2551 & -0.4538 & -0.8952 \\ 0.0 & 0.0 & 0.0 & -1.6267 & -1.5810 & -0.6762 & -0.4867 & 1.1693 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -0.5556 & -1.0677 & 0.4852 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.7754 & 0.5523 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -3.0699 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{pmatrix}.$$

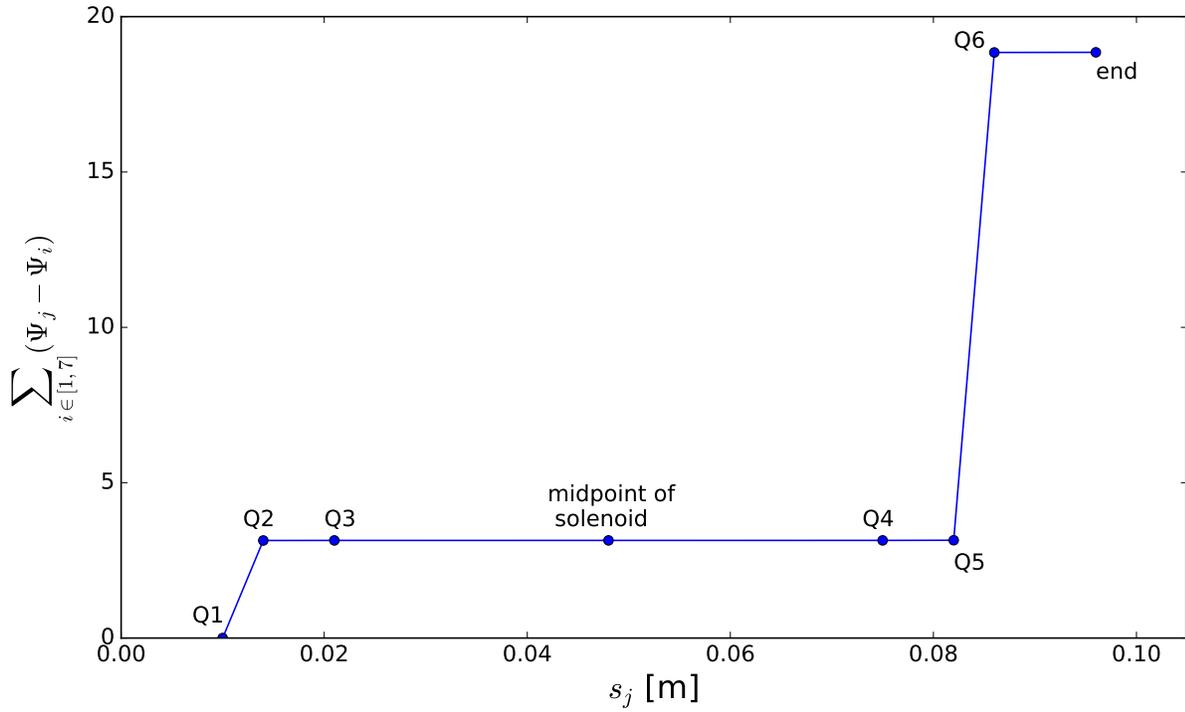
If we sum down the columns of the matrix, we get the total S-G displacement at each point j along the line.



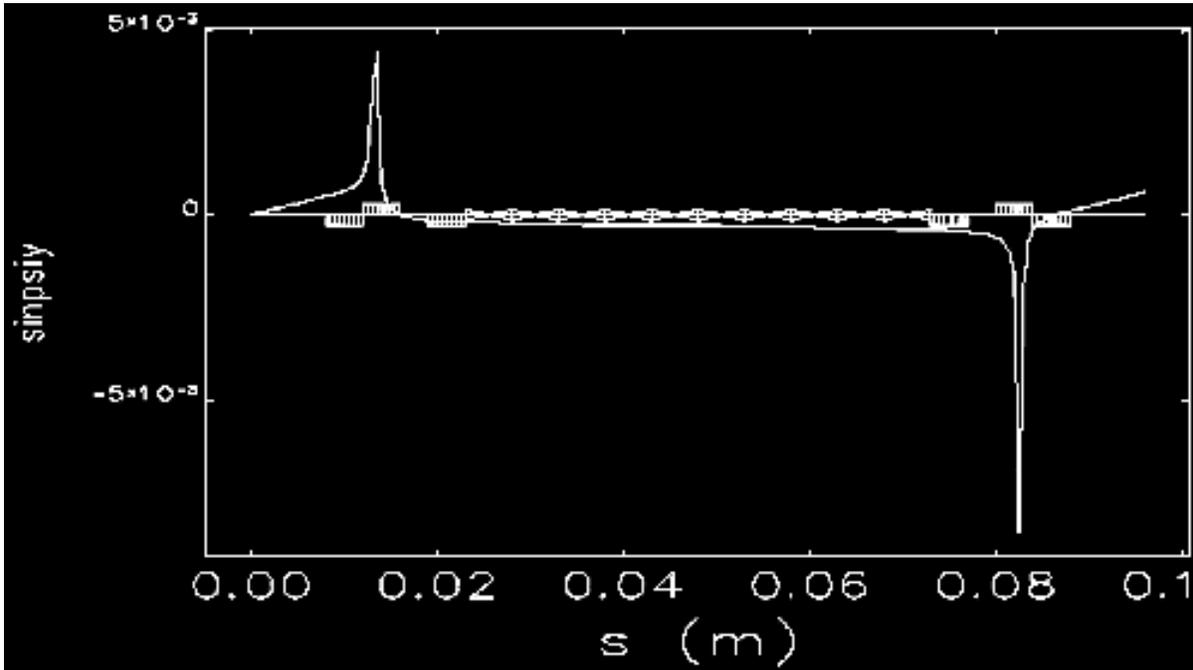
A few things are noticeable in this plot. First, there isn't much more than a pico-meter displacement at any point along the beamline. Second, most of the displacement due to the first three quads is canceled from solenoid and the last three quads. The fact that there is very little displacement is due to the fact that there is very little phase advance along the line. The values for the $\sin(\psi_{y,j} - \psi_{y,i})$ factor are very small.



The sine of the phase advance is very small even though the magnitude of the phase jumps by 2π radians at the two points in the beamline where β_y is at a minimum.

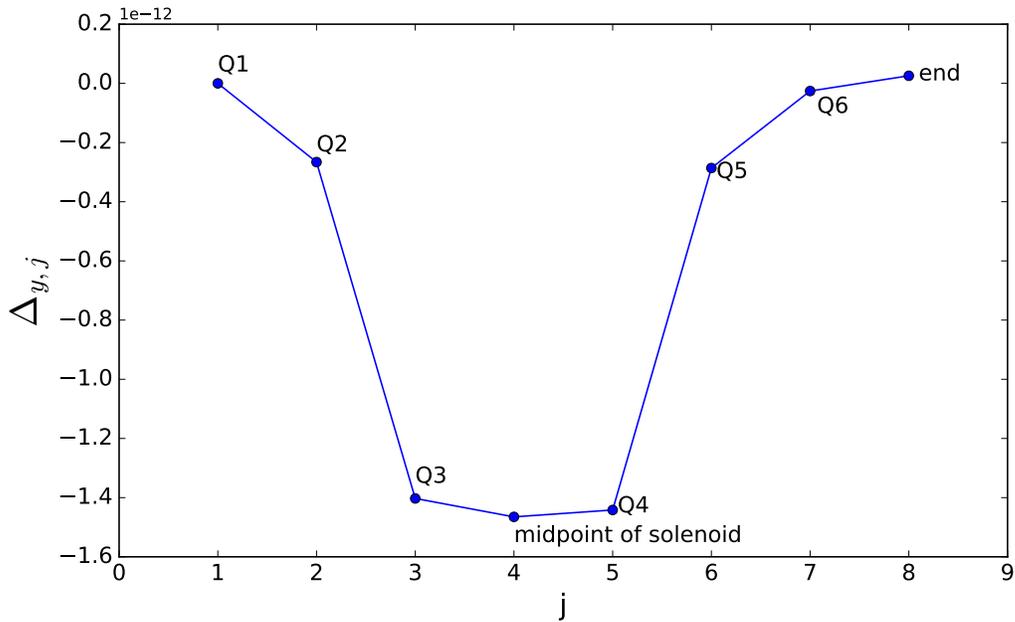


The length that the phase advance is large is very short, but there are two regions where a larger $\sin(\psi_y)$ exists.



The maximum magnitude of $\sin(\psi_y)$ in this plot is almost 10^{-2} , which is about one order of magnitude larger than the values we used to find the S-G displacement in the earlier plot.

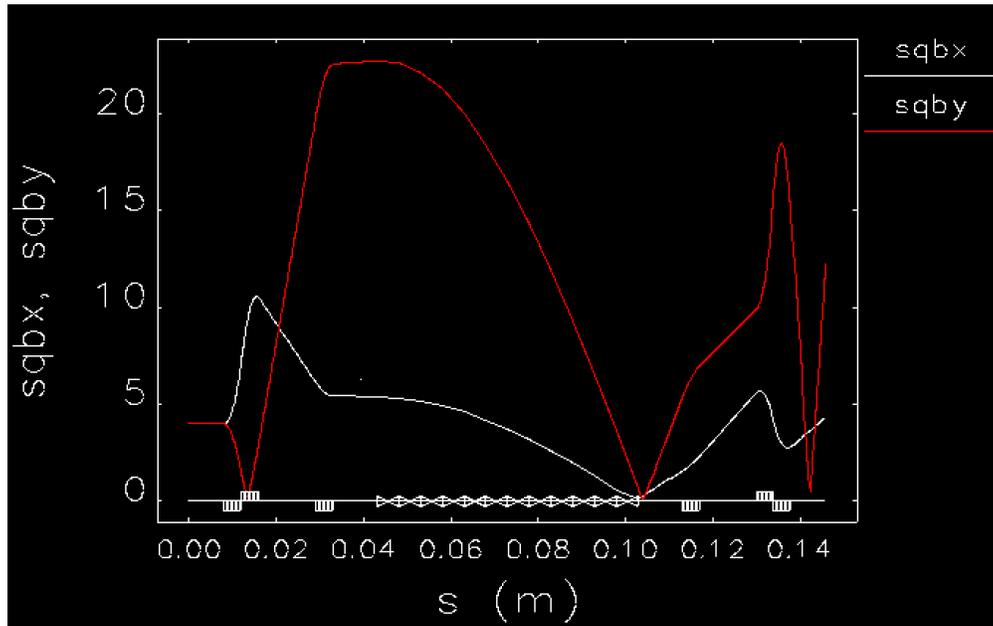
To verify that the change in spin precession due to the solenoid contributes to the total displacement, we calculate and plot $\Delta_{y,j}$ for the beamline without any solenoid from section 2.1 and see that without a solenoid, the last three quads do cancel the S-G displacement from the first three:



A Solenoid matching investigation

Adding a solenoid to the drift space in the center of the beam line has a large focusing effect on the β -functions in both x and y. We keep the solenoid the same length as the original design, 6 cm, and vary the magnetic field strength to compare the effect on β .

The original design was to rotate the spin by $\pi/2$ to maximize the Stern-Gerlach total deflection. We see in section 2.2 that the β -function isn't symmetric without changing the strength and position of the quads.

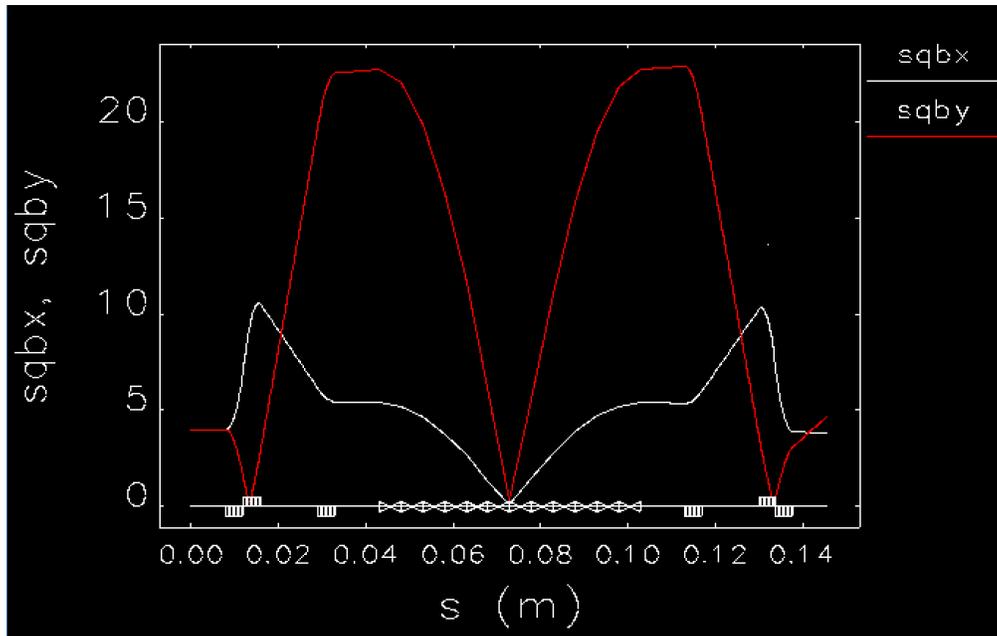


Next, we look at solenoid strengths that would rotate the spin by even-integer multiples of π . Using the solenoid strength equation from section 2.2, for $L_s = 6$ cm, we get

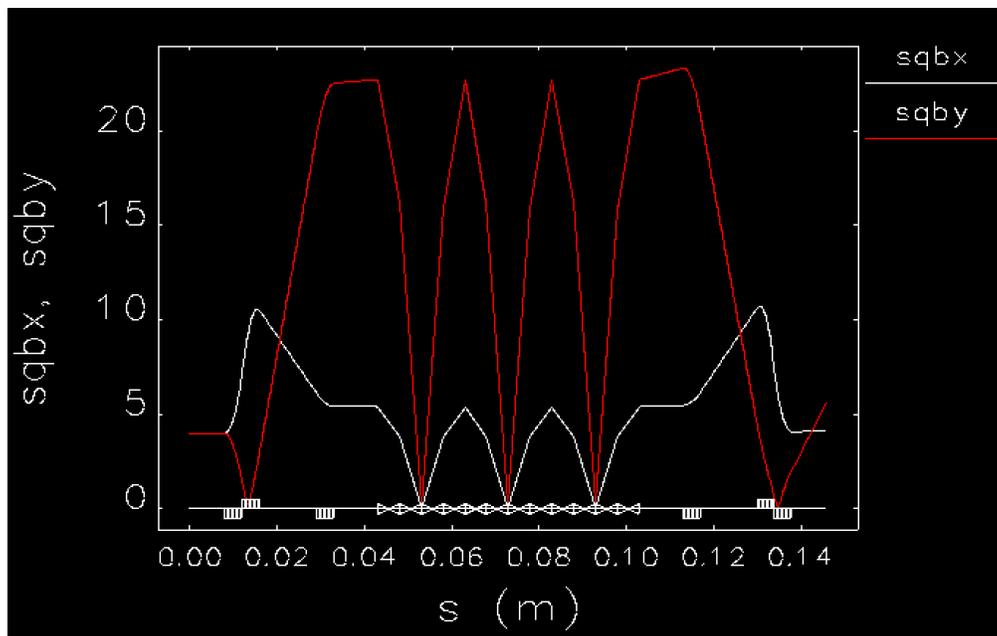
$$B_{2\pi} = 0.3090 \text{ T}$$

$$B_{6\pi} = 0.9270 \text{ T}.$$

$\theta = 2\pi$ and $L_s = 6$ cm ($B = 0.3090$ T):



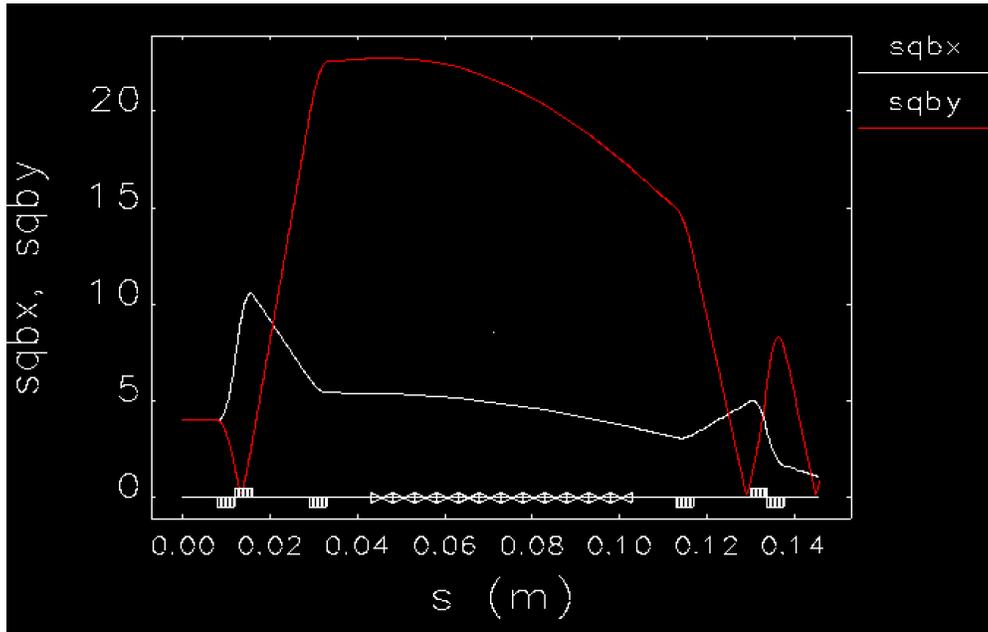
$\theta = 6\pi$ and $L_s = 6$ cm ($B = 0.9270$ T):



From the simulation, you can see β oscillate with the number of spin rotations. If we want to avoid β going to zero in the solenoid, we need the spin rotation to be between zero and π . For $\theta = \pi/2$, we want a solenoid magnetic field of

$$B_{\pi/2} = 0.0772 \text{ T}.$$

$\theta = \pi/2$ and $L_s = 6$ cm ($B = 0.0772$ T):



With $\theta < \pi$, β doesn't pass through zero, but it's no longer symmetric. The quads can be adjusted to make the β -functions symmetric as in section 2.2.

The solenoid-off transfer matrix through the central drift section is quite close to an identity matrix (because the drift length is short compared to the beta functions). For special values of the solenoid magnetic field there are values for which a cyclotron orbit completes exactly an integer number of turns. For these field values the solenoid transfer matrix is approximately equal to an identity matrix also. So the drift can be replaced by the solenoid and the discrepancies between their transfer matrices compensated by minor retuning of the other lattice parameters.