OBLIG

In

FYS4565: Physics and Applications of accelerators and beams

By FURKAN KAYA

1.

a)

The relation between normalised emittance and geometric emittance is given by:

$$\epsilon_N = \beta\,\gamma\,\epsilon \quad (1)$$

Conserved under acceleration. In (1), $\beta\,\gamma$ = normalised velocity and the Lorentz factor. They are not related to the Twiss parameters. Normally emittance is only constant in beams without acceleration. Increasing the momentum of the beam reduces the emittance. The normalised emittance dos not change due to acceleration or energy.

b)

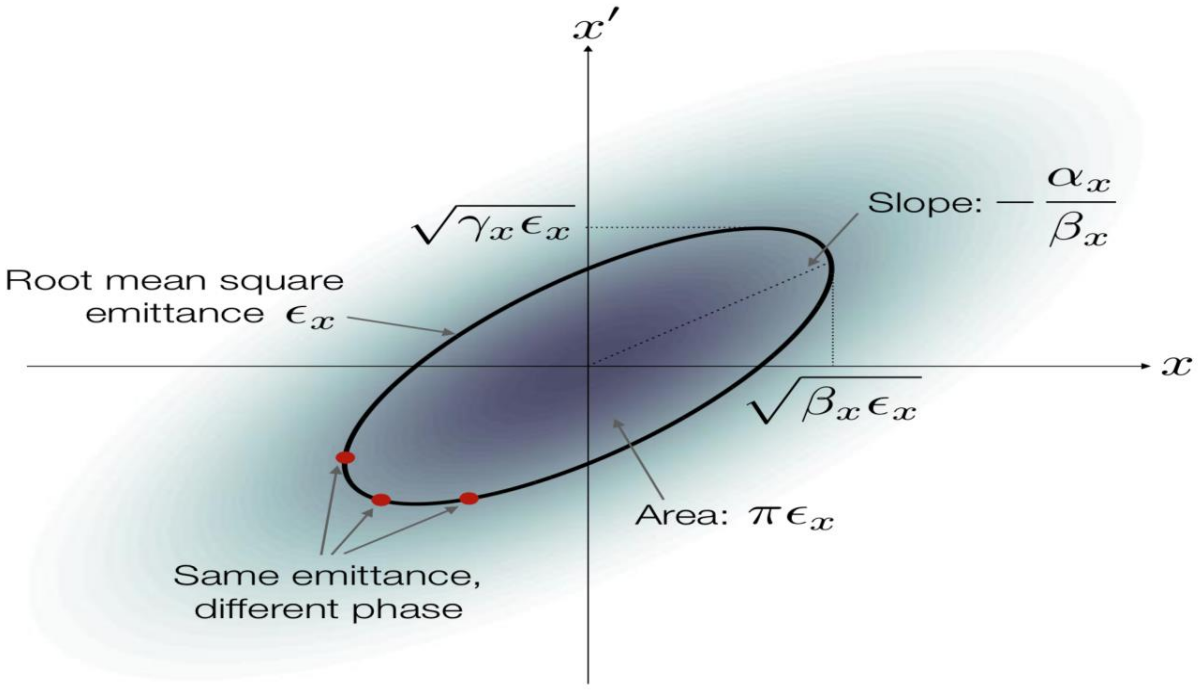

Figure 1: Schematic of the phase space. Taken from Dr. Adli

We have the Twiss parameters defined as:

$$\alpha = -\frac{1}{2}\,\beta'$$

$$\gamma = \frac{1 + \alpha^2}{\beta}$$

And an elliptic phase room, as seen on figure 1, with ϵ given as the single-particle emittance. From that we get an equation for ϵ:

$$\epsilon = \gamma\, x^2 + 2\,\alpha x x' + \beta\, x^{2\prime} \quad (2)$$

We determine x' through $\frac{dx'}{dx} = 0$ giving us:

$$x' = \sqrt{\epsilon\,\gamma} \quad (3)$$

$$x = \pm\alpha \sqrt{\frac{\epsilon}{\gamma}} \quad (4)$$

The Twiss parameters are needed to describe the propagation of a single particle. The parameters are given as solutions of the betatron equation and they propagate along the lattice, s.

c)

I believe this has partially been done in equation (2) above, but we could always rewrite it with the N subfixes.

$$\epsilon = \gamma x_N^2 + 2\alpha x_N x_N' + \beta x_N^{2\prime} \quad (5)$$

[Det viktige er at den geometriske emittansen er gitt av faserommet, N er ikke den samme. Så ta bort N]

2.

a)

By running the script: runMADX, we received the following output:

```
>> runMADX

 ++++++++++++++++++++++++++++++++++++++++++++
 +     MAD-X 5.04.02  (64 bit, Windows)     +
 + Support: mad@cern.ch, http://cern.ch/mad +
 + Release   date: 2018.10.03               +
```

```
 + Execution date: 2019.04.18 12:49:34      +
 ++++++++++++++++++++++++++++++++++++++++++++
OPTION, ECHO = FALSE, TWISS_PRINT = FALSE, INFO = FALSE;
initial->betx      =        3.414200001 ;
initial->bety      =       0.5858000001 ;
initial->alfx      =       -2.41418082 ;
initial->alfy      =       0.4142191801 ;
emit_x             =        5.11e-009 ;
emit_y             =        5.11e-009 ;
 GXPLOT-X11  1.50 initialized
 plot number =           1
 plot number =           2
enter TRACK module
one pass is on
exit TRACK module

Number of warnings: 0
 ++++++++++++++++++++++++++++++++++++++++++++
 +         MAD-X finished normally         +
 ++++++++++++++++++++++++++++++++++++++++++++
Done.
>> mit_x             =        5.11e-009 ;
emit_y             =        5.11e-009 ;
 GXPLOT-X11  1.50 initialized
```

```
 plot number =           1
 plot number =           2
enter TRACK module
one pass is on
exit TRACK module
 Number of warnings: 0
 ++++++++++++++++++++++++++++++++++++++++++++
 +         MAD-X finished normally          +
 ++++++++++++++++++++++++++++++++++++++++++++
Done
```

The entire output was included in order to show that the program worked as expected.

b)

The script getInitialBeam() was loaded and the 3 first and 3 last values are written down below to show that the script worked:

```
  -0.0432   -0.0344   -0.0563   -0.0975   -0.0048    0.0000
   0.1692    0.1117   -0.0669   -0.0153   -0.0095    0.0000
   0.0752    0.0090   -0.0201   -0.1156    0.0001    0.0000

   0.1266    0.0428   -0.0827    0.1489   -0.0110    0.0000
  -0.1534   -0.1427    0.0449   -0.1205    0.0091    0.0000
   0.2217    0.1484   -0.0438    0.3110   -0.0206    0.0000
   0.0821    0.1014    0.0248    0.0306    0.0109    0.0000
```

c)

First of all, we are supposed to find the rms normalized emittance from the matrix given in the scripts. The code that was was:

```
function [ beam ] = getInitialBeam()

    % declare tracking file
    trackFile = 'particles.one';

    % extract beam 6D phase space (after last BPM)
    headerLinesIn = 54;
    track_data = importdata(trackFile, ' ',
headerLinesIn);
    %beam = track_data.data(:,3:8);%
    %beam = track_data.data(:,3);
    beam = track_data.data(:,4);

    end
```

This way both x (position) and xp (moment) was found. Values obtained were:

x = 1.3226e-4;

xp = 1.0197e-4;

The script written for this task was:

```
betx = 3.414200001;
alpx = -2.41418082;
gamma = ((1 + (alpx).^2)./(betx));
gamma1 = 1.9569e+03;
bet1 = 0.999999;

x = 1.3226e-4; %rms-value for position
xp = 1.0197e-4; %rms-value for momentum

emittance = (gamma*x.^2 + 2*alpx*x*xp + betx*xp.^2);

normal_emi = bet1*gamma1*emittance
```

And the output was:

>> Acc_c

normal_emi = 1.0503e-05

This gives us the same answer as we were expected to get.

d)

The Twiss-parameters in runMADX are as following:

initial->betx = 3.414200001 ;

initial->bety = 0.5858000001 ;

initial->alfx = -2.41418082 ;

initial->alfy = 0.4142191801 ;

The script for the code was written as:

```
x = [-3.37434965000000e-05;9.79233786200000e-
05;0.00013714984460000;….];

xp = [-5.73404622300000e-
06;0.00011706594750000;9.66465145800000e-05;….];

plot(x,xp);

xlabel('x');
ylabel('xp');
title('Phasespace');
```

And that led to this plot of the phase space:

For the plot, look at the next page due to space constraint.

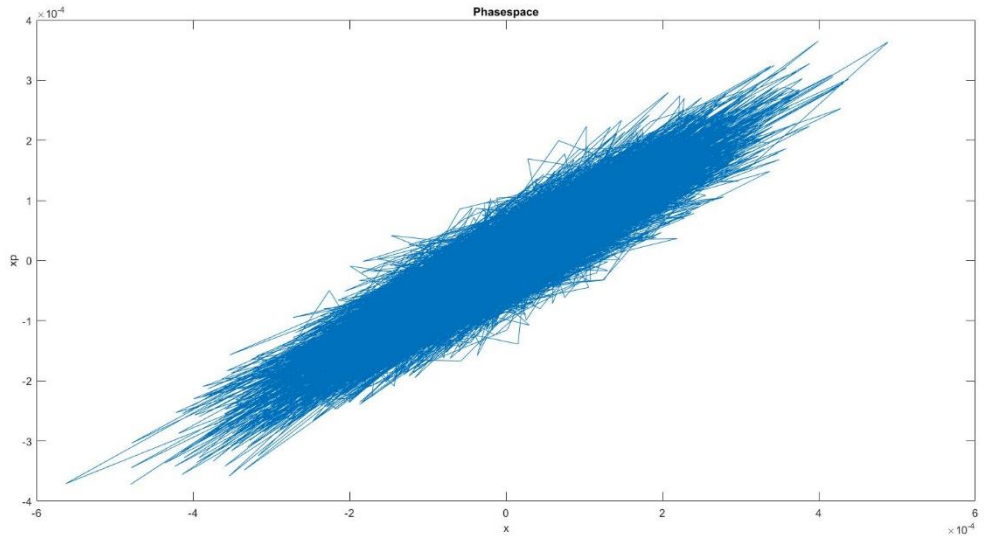

Figure 2: Schematic of the phase space for x and xp

Accompanying figure 2 is:

>> max(x)

ans = 4.8866e-04

>> max(xp)

ans = 3.6498e-04

The slope is found by using:

>> b = polyfit(x,xp,1)

b =0.7141

From these results we use the relations given in figure 1 to find the Twiss parameters.

$$Slope = -\frac{\alpha_x}{\beta_x} => \beta_x = -\frac{\alpha_x}{slope} => \beta_x = -\left(\frac{\alpha_x}{0.7141}\right)$$

$$\gamma = \frac{1+\alpha^2}{\beta} => \gamma = -\frac{1+\alpha_x^2}{\frac{\alpha_x}{0.7141}}$$

We also have use for the following relations:

$$(xp)_{maks} = \sqrt{\epsilon\gamma} => 3.6498 * 10^{-4} = \sqrt{\epsilon\gamma}$$

$$x_{maks}\ 4.8866 * 10^{-4} = \sqrt{\beta\epsilon}$$

$$4.8866 * 10^{-4} = \sqrt{\frac{\alpha_x}{0.7141}\,\epsilon}$$

First equation we look at is for xp(maks). After some calculations we get:

$$\frac{-1.865 * 10^{-7}\alpha_x}{1 + \alpha_x^2} = \epsilon$$

Then we go back to the equation for x(maks).

$$4.8866 * 10^{-4} = \sqrt{-\left(\frac{1.865 * 10^{-7}}{1 + \alpha_x^2}\right)\left(\frac{\alpha_x}{0.7141}\right)}$$

After further calculations we get:

$$\frac{2.388 * 0^{-7}}{2.612 * 10^{-7}} = \frac{\alpha_x^2}{1 + \alpha_x^2} => 0.914 = \frac{\alpha_x^2}{1 + \alpha_x^2}$$

$$0.914 + 0.914\alpha_x^2 - \alpha_x^2 = 0$$

That gives us the Twiss parameters for alpha_x and beta_x below:

$$\alpha_x^2 = \frac{0.914}{0.086} = 10.63$$

$$\alpha_x = \pm 3.26 = -3.26$$

$$\beta_x = \frac{-3.26}{-0.7141} = 4.57$$

The Twiss-parameters are slightly skewed from the ones obtained with runMADX because we know from the previous assignment that the emittance was off by an order. But despite that, they are quite close. We could use time to find both beta_y and alpha_y, but the method is the same. Only difference is changing the values used. Due to timeconstraint, this was not done.

e)

The quantitative values for the function getBPMreadings() is:

[1.0e-05 *(-0.0969 0.1216 0.1468 0.3275 0.0969 -0.1216 -0.1468 -0.3275]

BPM stands for beam position monitoring systems. The beam trajectory may be deflected from the pipe center due to a variety ofreasons. That is why having a beam position monitoring system ishighly needed in accelerators. The beam position monitoring system (BPM) is the system that determines the position of the beam.

We were supposed to make a plot, so this follows in figure 3 below as well.

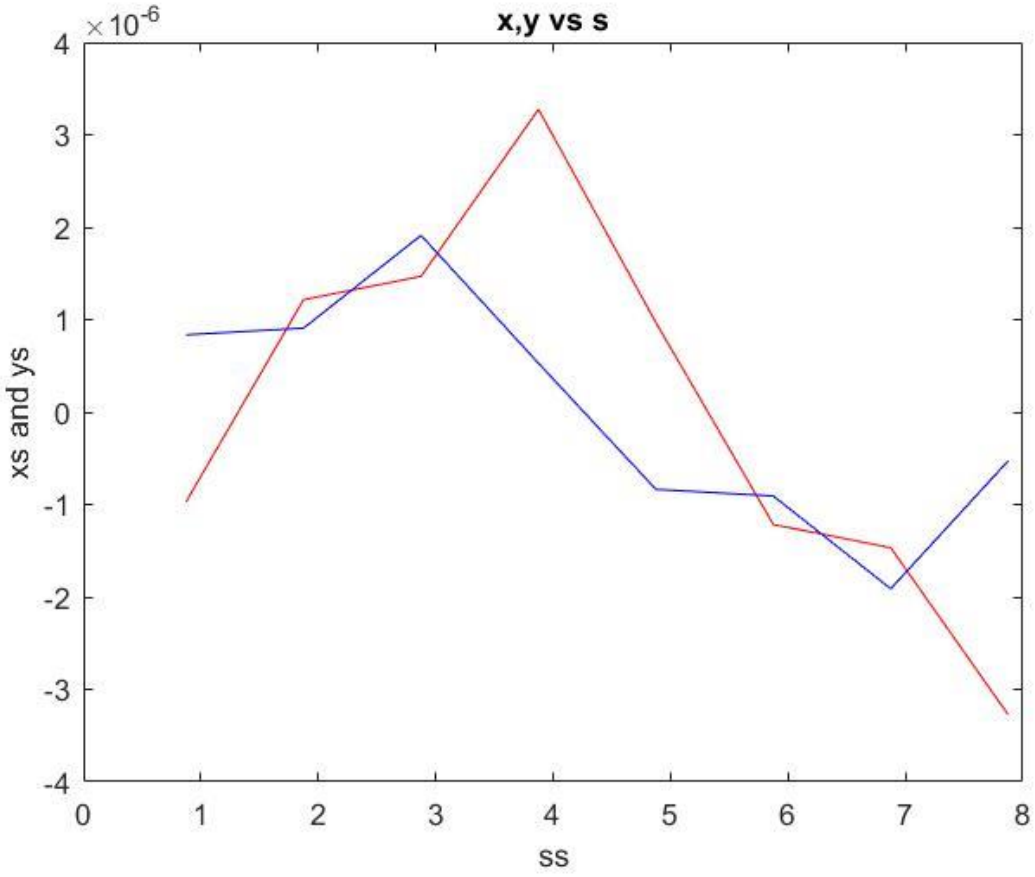

Figure 3: Schematic shows x,y vs s for BPMreadings

Since I am a bit uncertain of if I've actually understood the assignment properly I will add a figure 4 where only the getBPMreadings values in the table earlier in this sub-assignment.

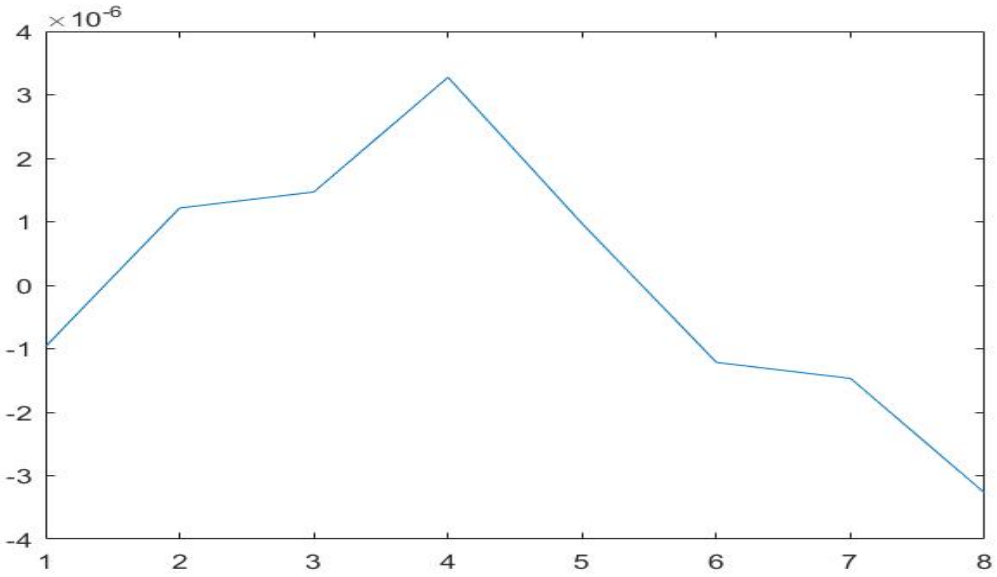

Figure 4: Schematic shows only the getBPMreading-values

f)

We are to introduce quad misalignments of 1mm rms offset. This is done as shown in the code below:

>> getQuadMisalignments

ans =1.0000e-03

>> getBPMreadings

ans = -0.0010 -0.0045 -0.0040 -0.0076 0.0008 0.0122 0.0076 0.0084

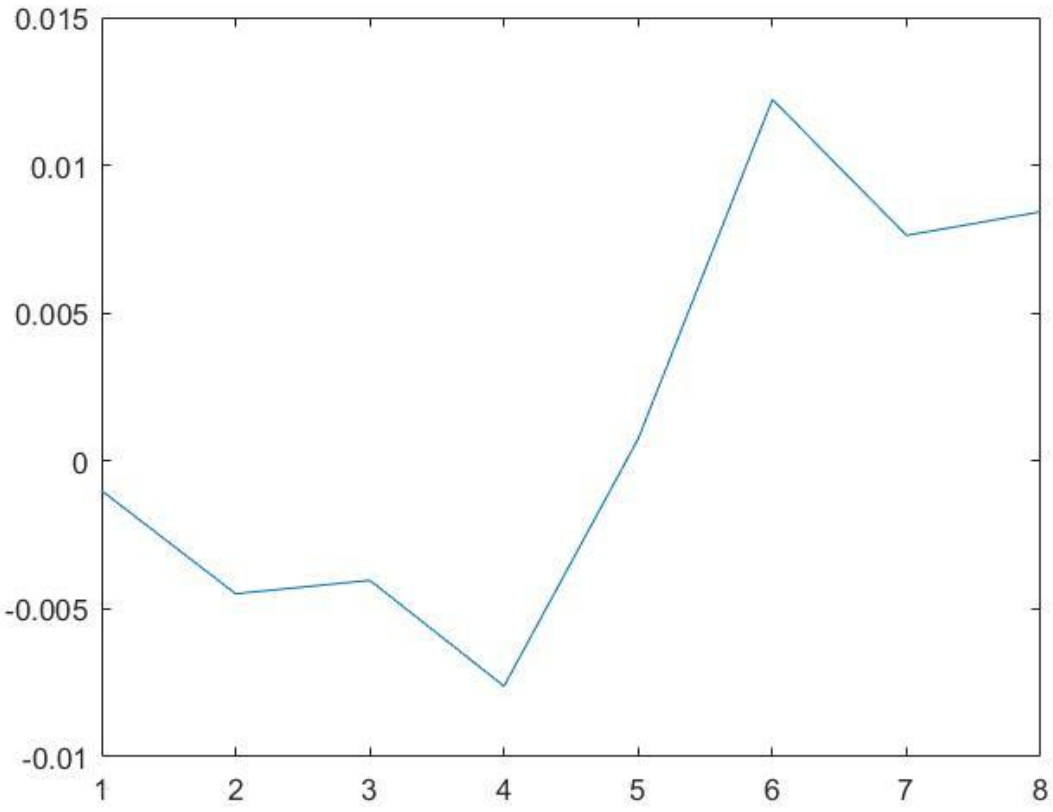

Figure 5: Schematic of only the quad misalignment-adjusted BPMreadings

As in the previous assignment, I am still uncertain of what to do, so I will add the plot of x and y vs separately.

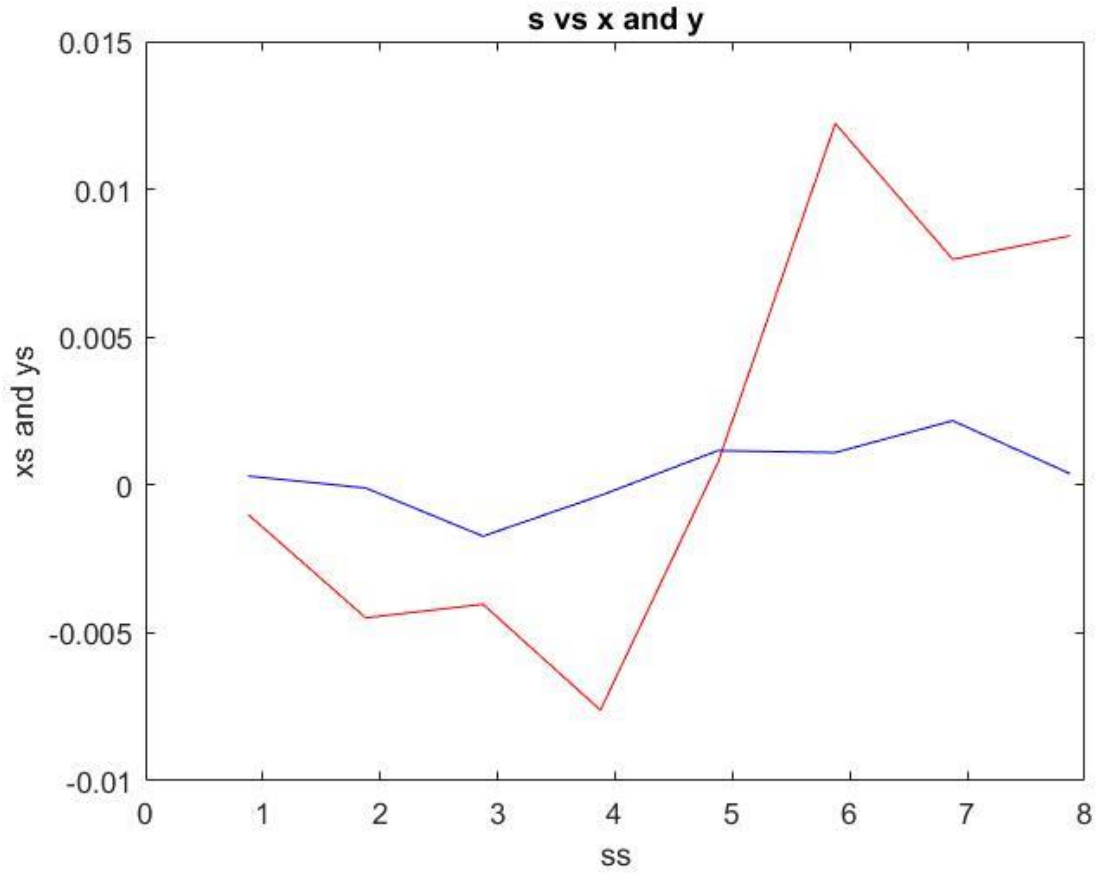

Figure 6: Schematic shows quad misalignment-adjusted BPMreadings

g)

The dispersion function D(s) defines the local sensitivity of the beam trajectory or orbit x(s) to a relative energy error $\Delta E/E$:

$$D(s) = \frac{x(s)}{\frac{\Delta E}{E}}$$

Based on the text of the assignment I will plot the initial beam with two different energies to see how the functions differs from each other. The coding and function is as following:

>> setEnergy(1)

>> getEnergy

ans =1

>> runMADX;

>> b = getBPMreadings()

b = [-0.000995048865850 -0.004471751330609 -0.004036051618496 -0.007635737789109 0.000702958104449 0.012077897185454 0.007624356642073 0.008586665716979]

>> setEnergy(1.01)

>> getEnergy

ans = 1.010000000000000

>> runMADX

>> c = getBPMreadings()

c =[-0.000995044031593 -0.004471757267710 -0.004036058891055 -0.007635754106943 0.000702953183055 0.012077902927884 0.007624363856951 0.008586682106083]

>> plot(b-c)

The plot is therefore:

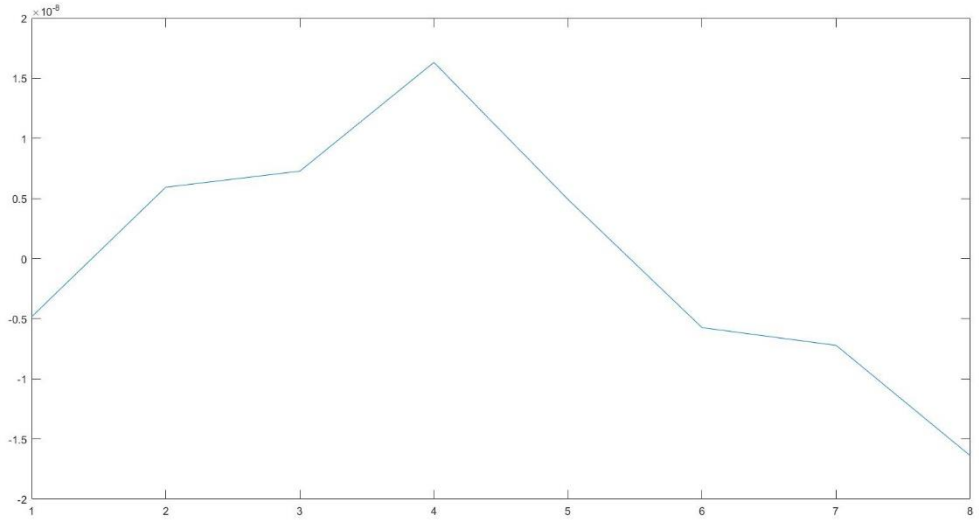

Figure 7: Schematic of the dispersion function

h) and i)

We are required to setEnergySpread() as 0. And then to plot the relative emittance growth ($\Delta\epsilon/\epsilon$) at the end of the of the lattice as a function of quadrupole misalignment. This is done below:

>> setEnergySpread(0)

>> getEnergySpread

ans = 0

By finding the emittance of both the final and initial beam, we subtracted the initial beam from the final beam to find an average value of the emittance growth. Which is basically the mean area of the phase space of the position and the moment.

The plot is:

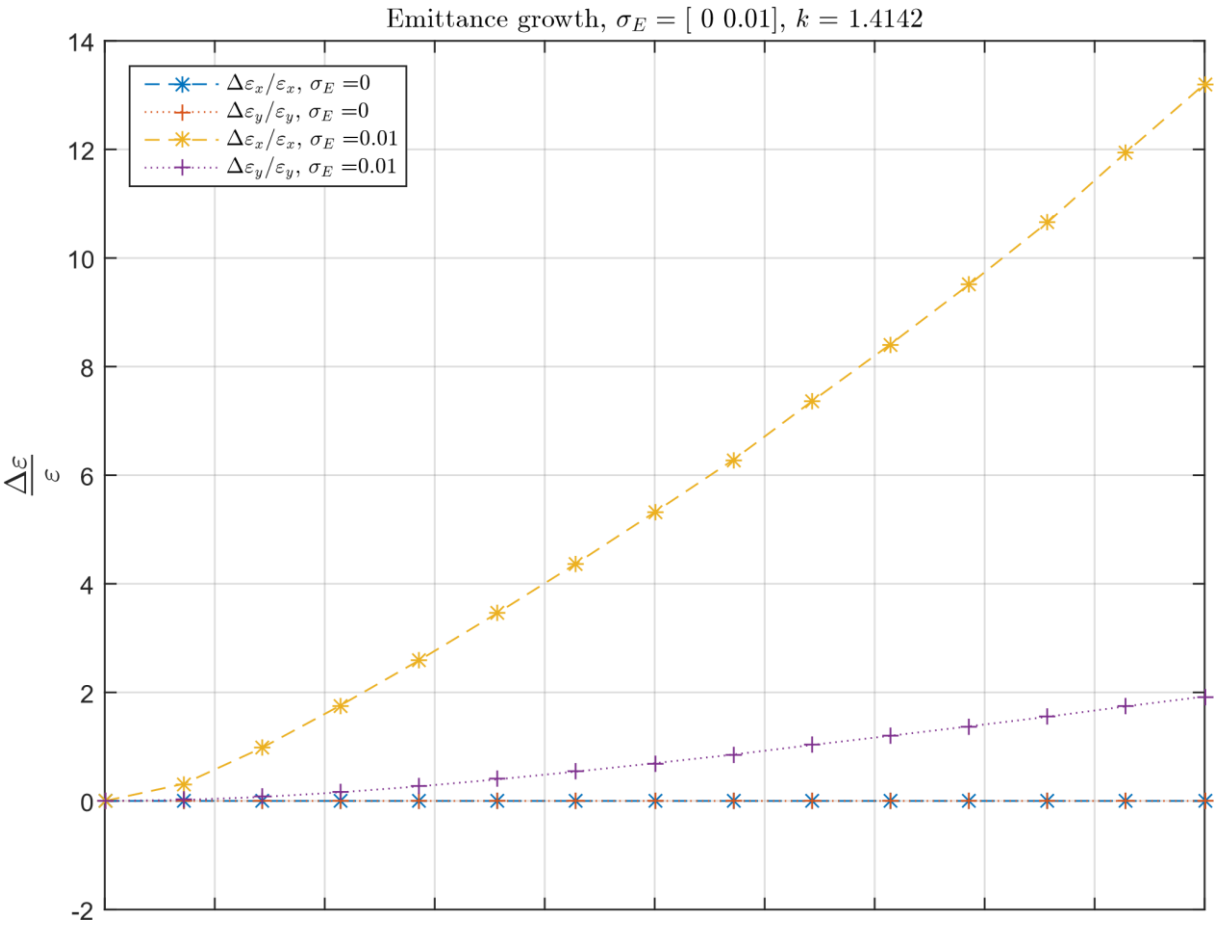

Figure 8: Schematic shows the emittance growth for setEnergySpread = 0 and 0.01. Quadrupole Misalignment on the x-axis and relative emittance growth on the y-axis.

It should be said that figure 8 was borrowed from Elisabeth Christensen, with her permission. My own figure was basically wrong and due to several reasons like tough workload and not participating in the classes made my plot inadequate.

j)

SetEnergySpread = 0 gives a lower emittance growth compared to setEnergySpread = 0.01, as seen on figure 8. The emittance basically represents the area of the phase space of the position and moment. That suggests that a wider energy spread leads to a larger area, which is natural since a higher energy for a particle gives a higher moment (speed) and also leads a larger coverage of distance (meaning position). From that one can conclude that a larger energyspread should lead to a higher emittance.

3.

a)

We are to setKickers() according to 1:1 steering. And use the getResponseMatrix() to calculate the response matrix. The matrix before adjustment is:

```
>> getResponseMatrix()

Tracking reference orbit... Done

Calculating response from kickers... 1 2 3 4 5 6 7 8 Done.

ans =

   0.7500         0         0         0         0         0         0         0
   2.7453    0.7500         0         0         0         0         0         0
   1.4561    0.7547    0.7500         0         0         0         0         0
   1.0807    1.4561    2.7453    0.7500         0         0         0         0
  -0.2986    0.5043    1.4561    0.7547    0.7500         0         0         0
  -2.4103   -0.2986    1.0807    1.4561    2.7453    0.7500         0         0
  -1.5487   -0.5984   -0.2986    0.5043    1.4561    0.7547    0.7500         0
  -1.8279   -1.5487   -2.4103   -0.2986    1.0807    1.4561    2.7453    0.7500
```

The assignment requires us to plot the uncorrected BPM and the 1:1 corrected. This is done below in a script. The response matrix and number of Kickers given as 8 and BPM as 8 suggests that there is a 1:1 steering. So we plot it then:

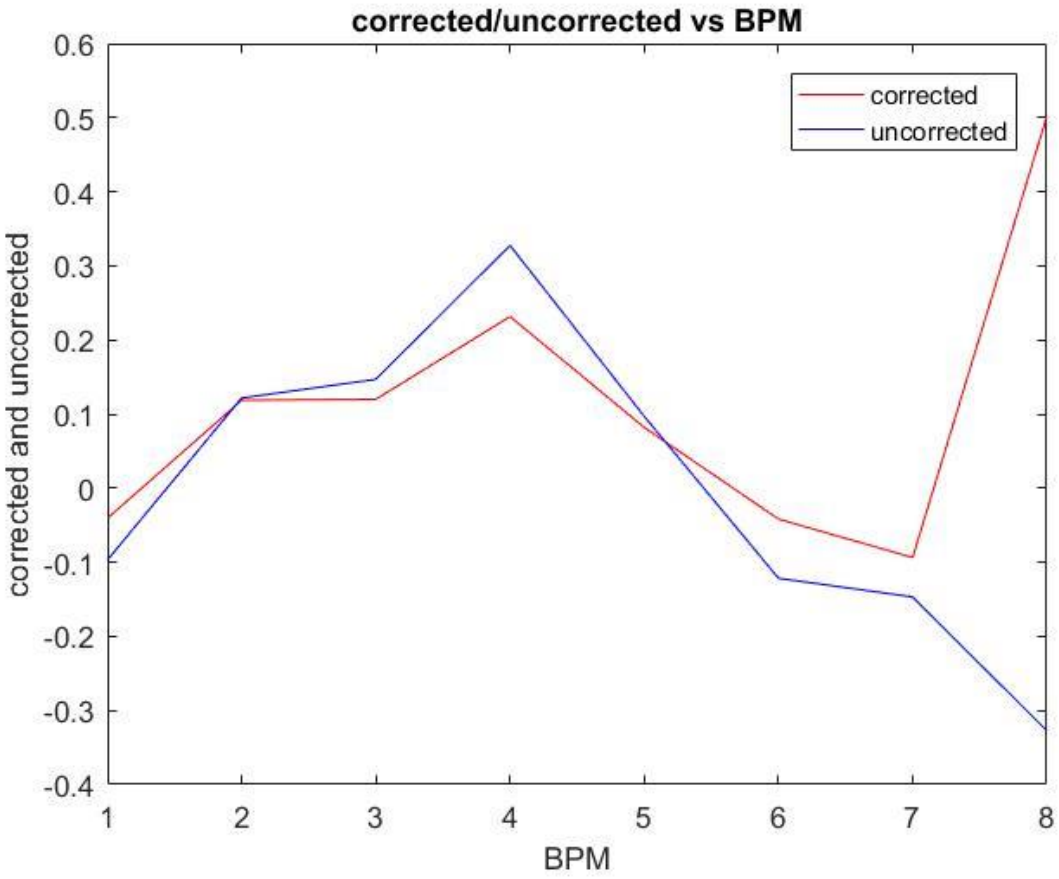

Figure 9: schematic shows the corrected and uncorrected BPM

b)

We are to plot the emittance growth vs quadrupole misalignment for both corrected and uncorrected orbits. [Could not get the correct answer]

c)

Assignment a) is done again with a BPM misalignment introduced. The result is the plot below: