Update on Single-beam Tuning of the CLIC 3 TeV Traditional Lattice

Dr Ryan Bodenstein – Presenter (remote)
University of Oxford – John Adams Institute

Dr Edu Marin Lacoma
CERN

CLIC Workshop, March 2017
CERN, Geneva, Switzerland

Before I begin…

- It should be noted that I did not work alone on this task.
  - In the past, I worked with Jochem Snuverink (now at PSI).
  - Currently, I work with Edu Marin Lacoma, recycling many of his codes.

- I apologize for giving this talk remotely. I am expecting my first child within days.

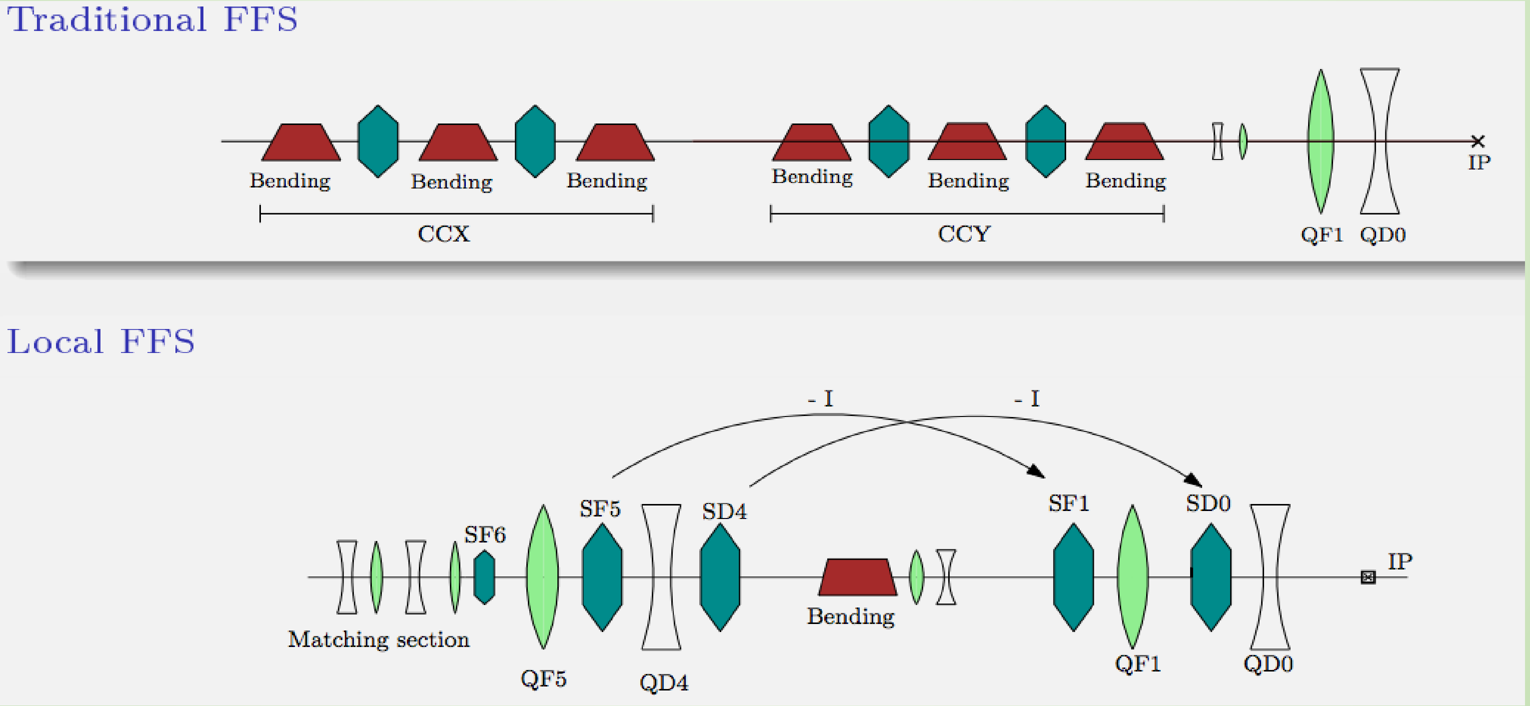
- Finally, please forgive my lack of a voice. I’ve had a cold all weekend, and am heavily-medicated. I realize I sound like Batman.

What will I be talking about?

- Bit of background on the beam delivery system (BDS) of CLIC
- Past attempts at tuning the traditional final focus system (FFS) lattice
- Current investigations of tuning parameters
- Planned work

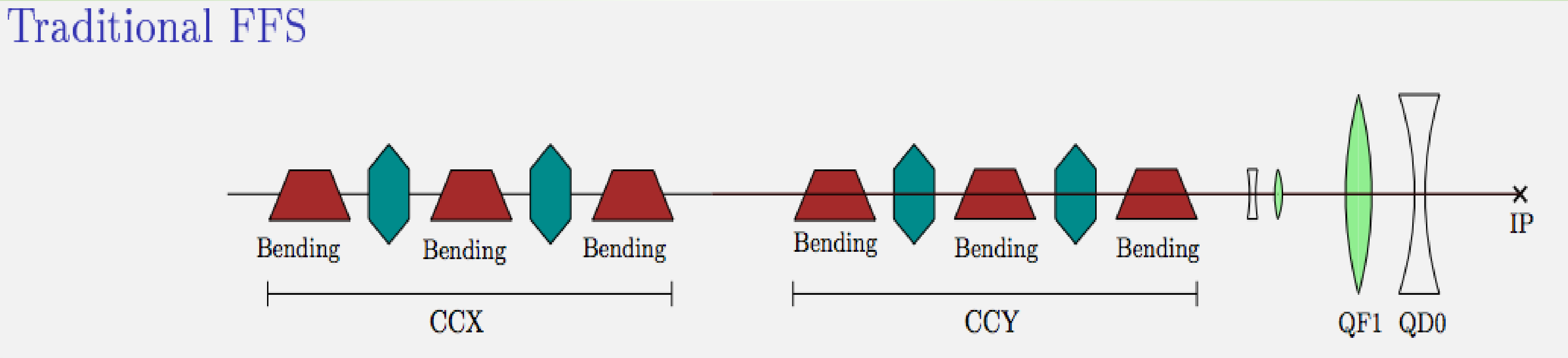
Some background

- Two separate sections for chromaticity correction
- Lattice by Hector Garcia, see e.g. his talk at [CLIC WS 2014]
- Relatively simple system for design and analysis



Some background

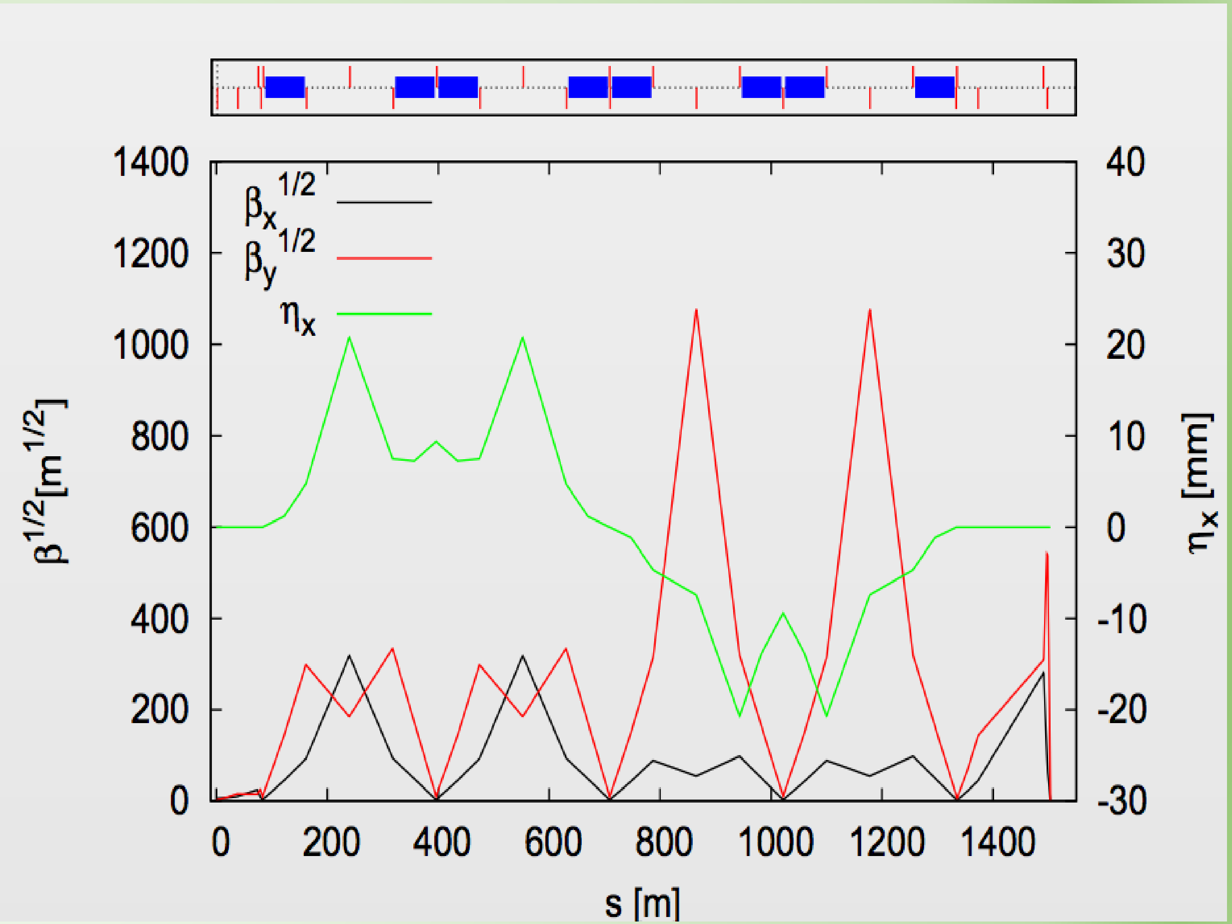
- Two separate sections for chromaticity correction
- Lattice by Hector Garcia, see e.g. his talk at [CLIC WS 2014]
- Relatively simple system for design and analysis



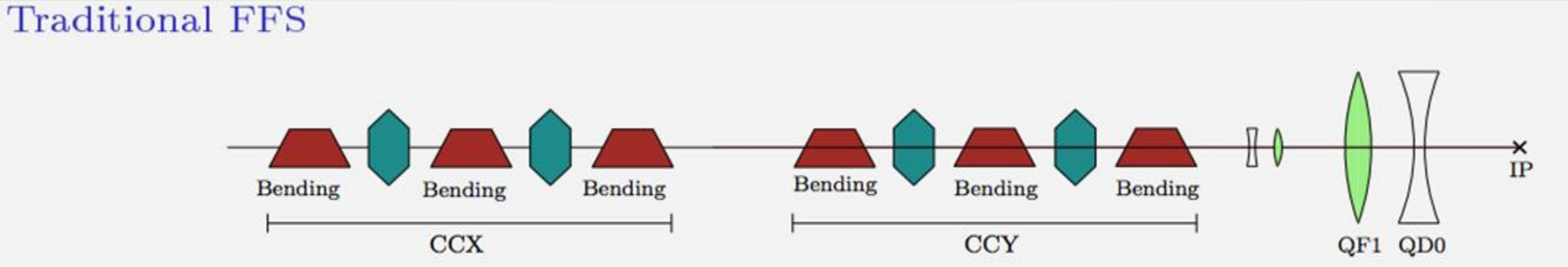
Traditional Final Focus

Parameter	Unit	Traditional "Optimized"	Local
Length	m	1460	450
Total Luminosity	$cm^{-2}s^{-1}$	$7.5 * 10^{34}$	$7.8 * 10^{34}$
Peak (1%) Luminosity	$cm^{-2}s^{-1}$	$2.4 * 10^{34}$	$2.4 * 10^{34}$

Optimized lattice achieves similar luminosity as local scheme.



Historical Tuning Procedure

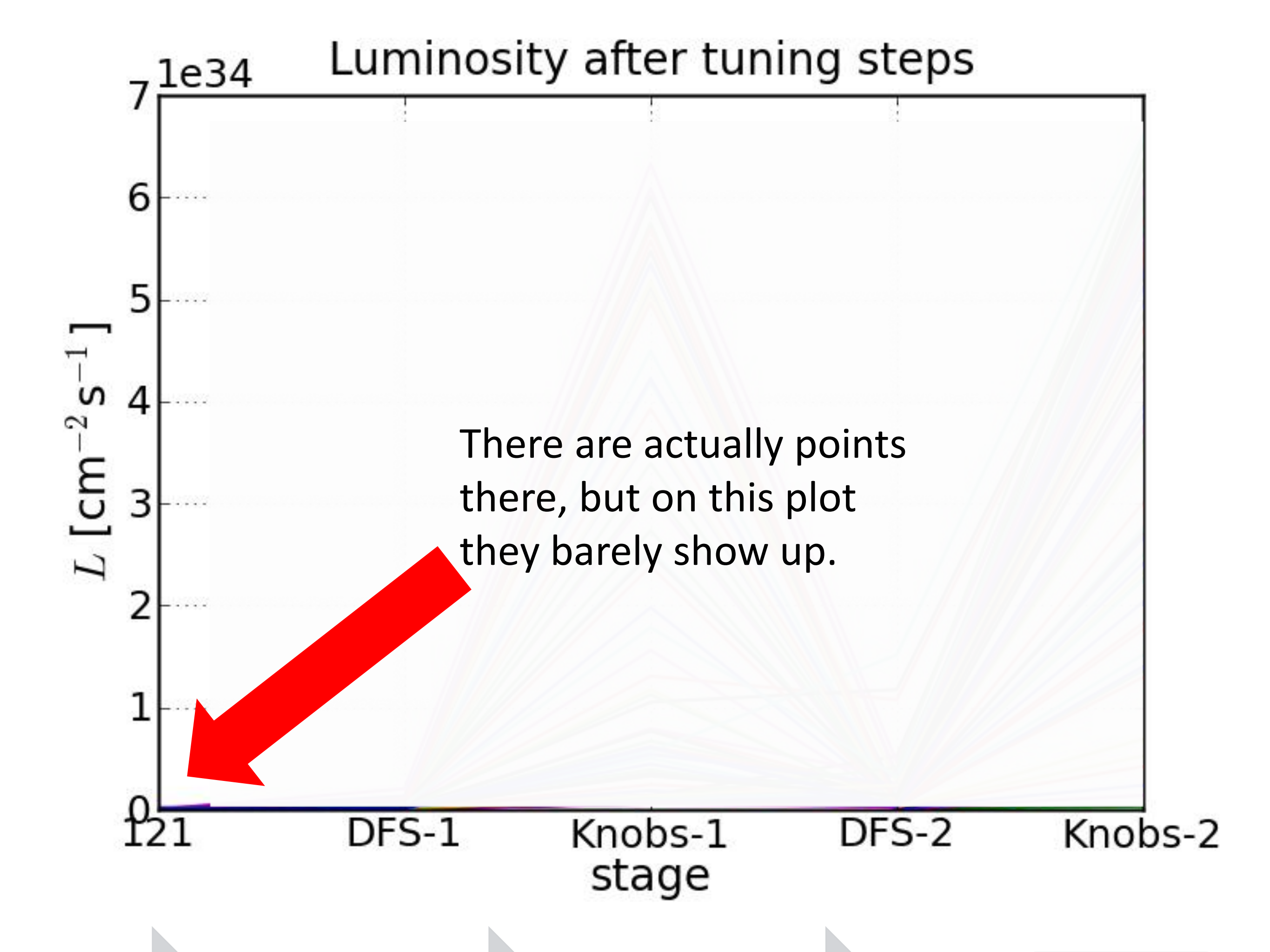


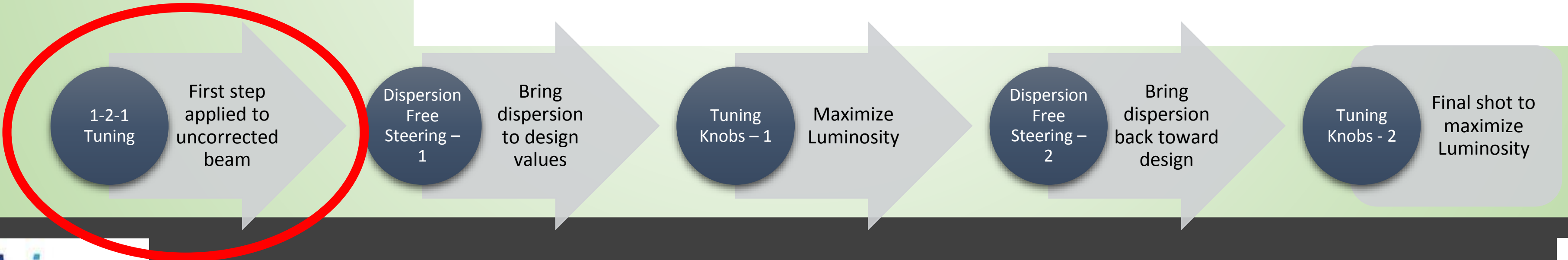
- Looking at the traditional final focus system, with 3 TeV collision energy.
- Simulations using PLACET and GUINEA-PIG
- Apply static offsets in x and y plane (10 µm RMS, 10 nm BPM resolution), then:

1. **1-2-1 Tuning**: First step applied to uncorrected beam
2. **Dispersion Free Steering – 1**: Bring dispersion to design values
3. **Tuning Knobs – 1**: Maximize Luminosity
4. **Dispersion Free Steering – 2**: Bring dispersion back toward design
5. **Tuning Knobs - 2**: Final shot to maximize Luminosity

Step by step…

1. Apply the first corrections to the uncorrected beam

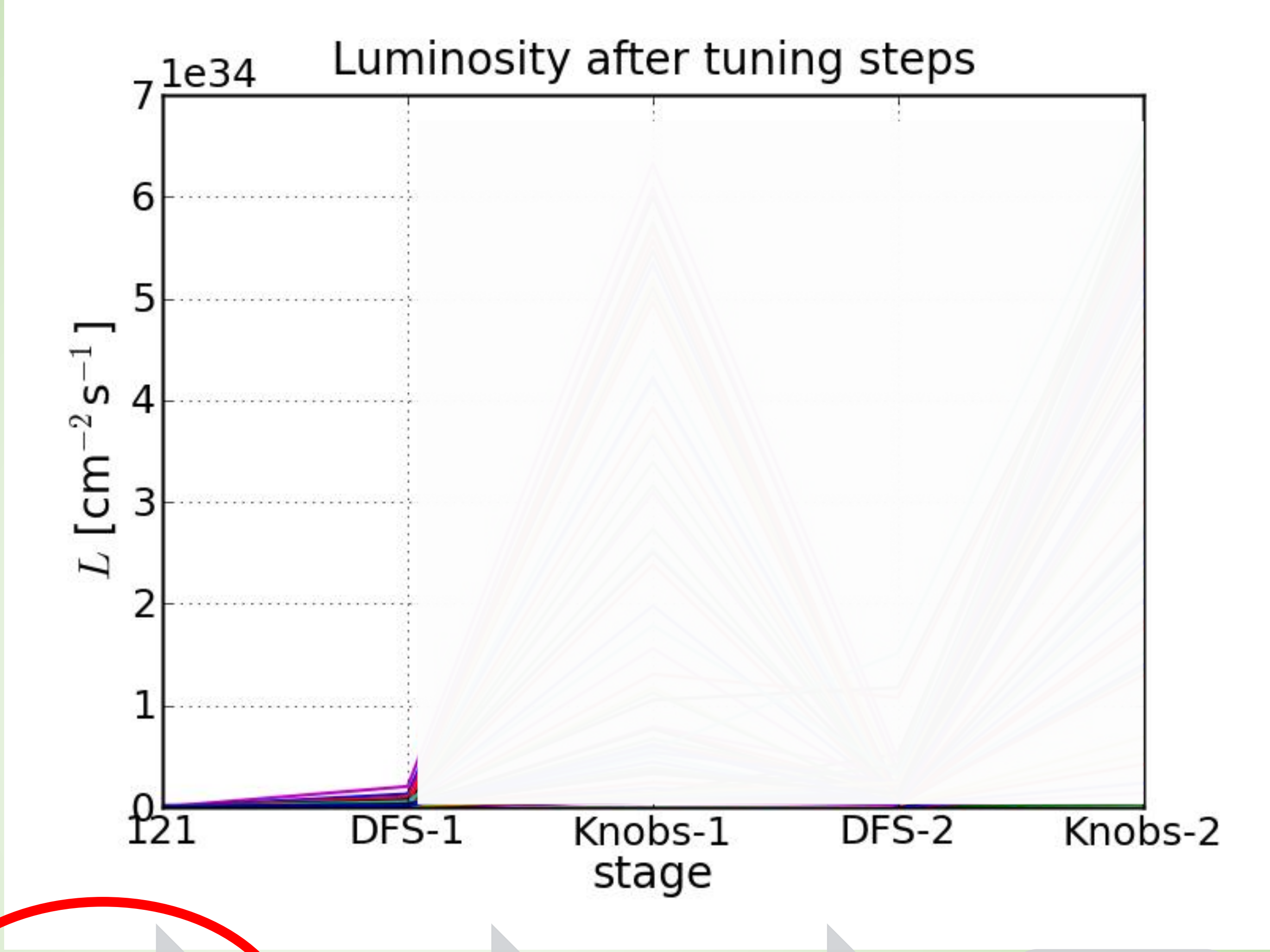


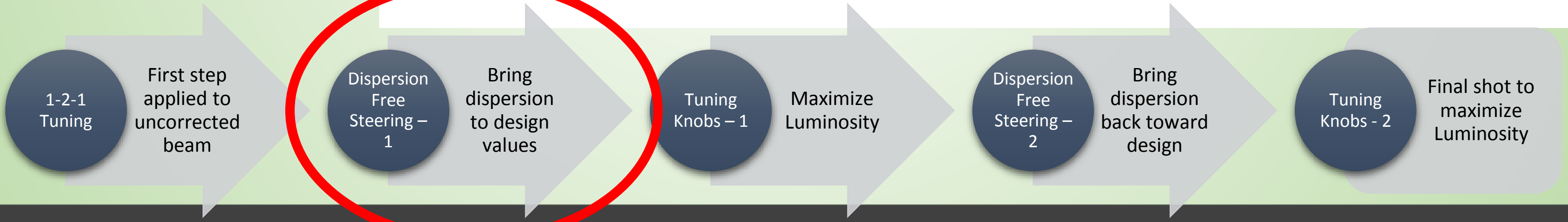


Step by step…

1. Apply the first corrections to the uncorrected beam

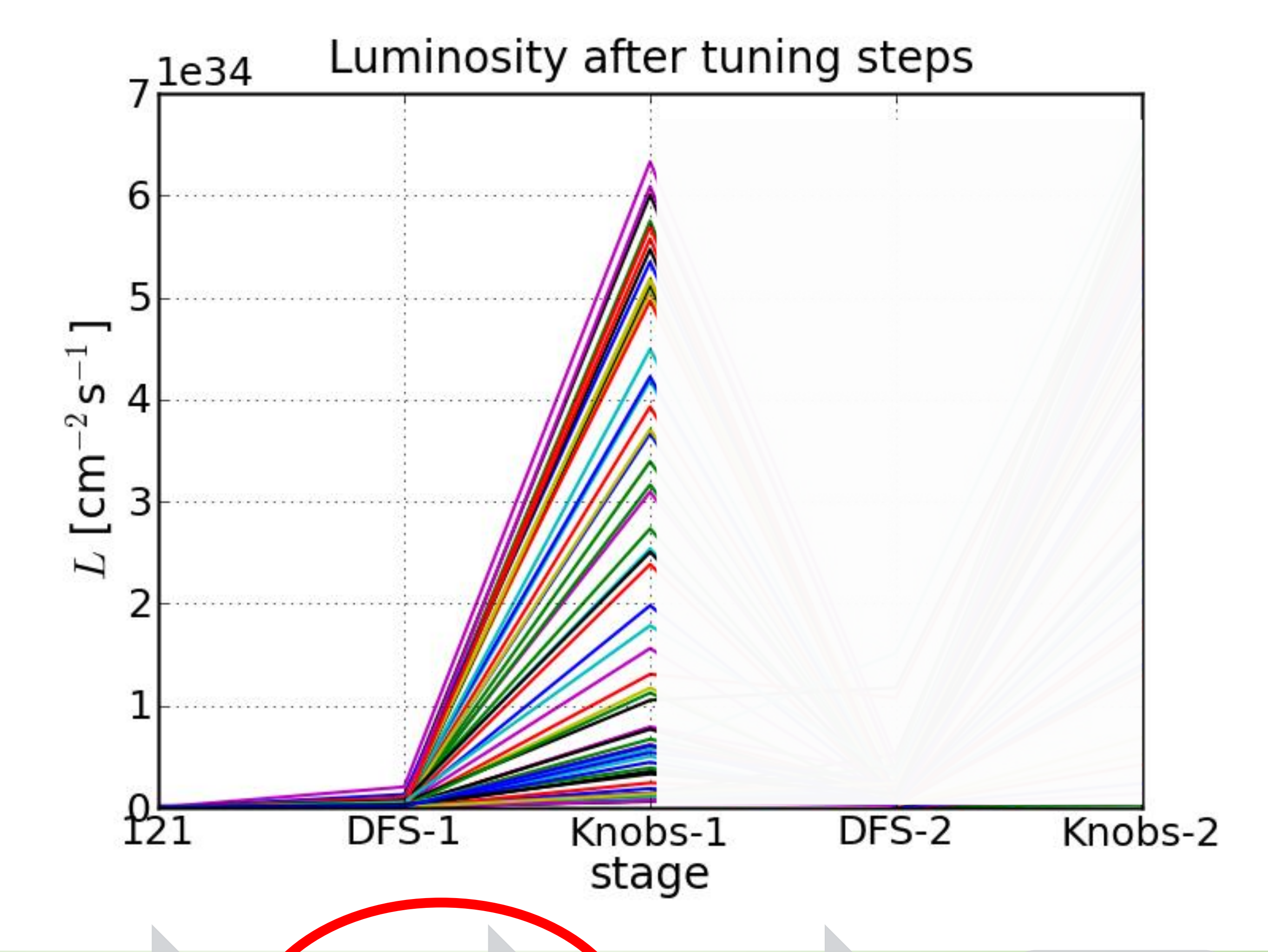
2. Correct the dispersion to approach design values





Step by step…

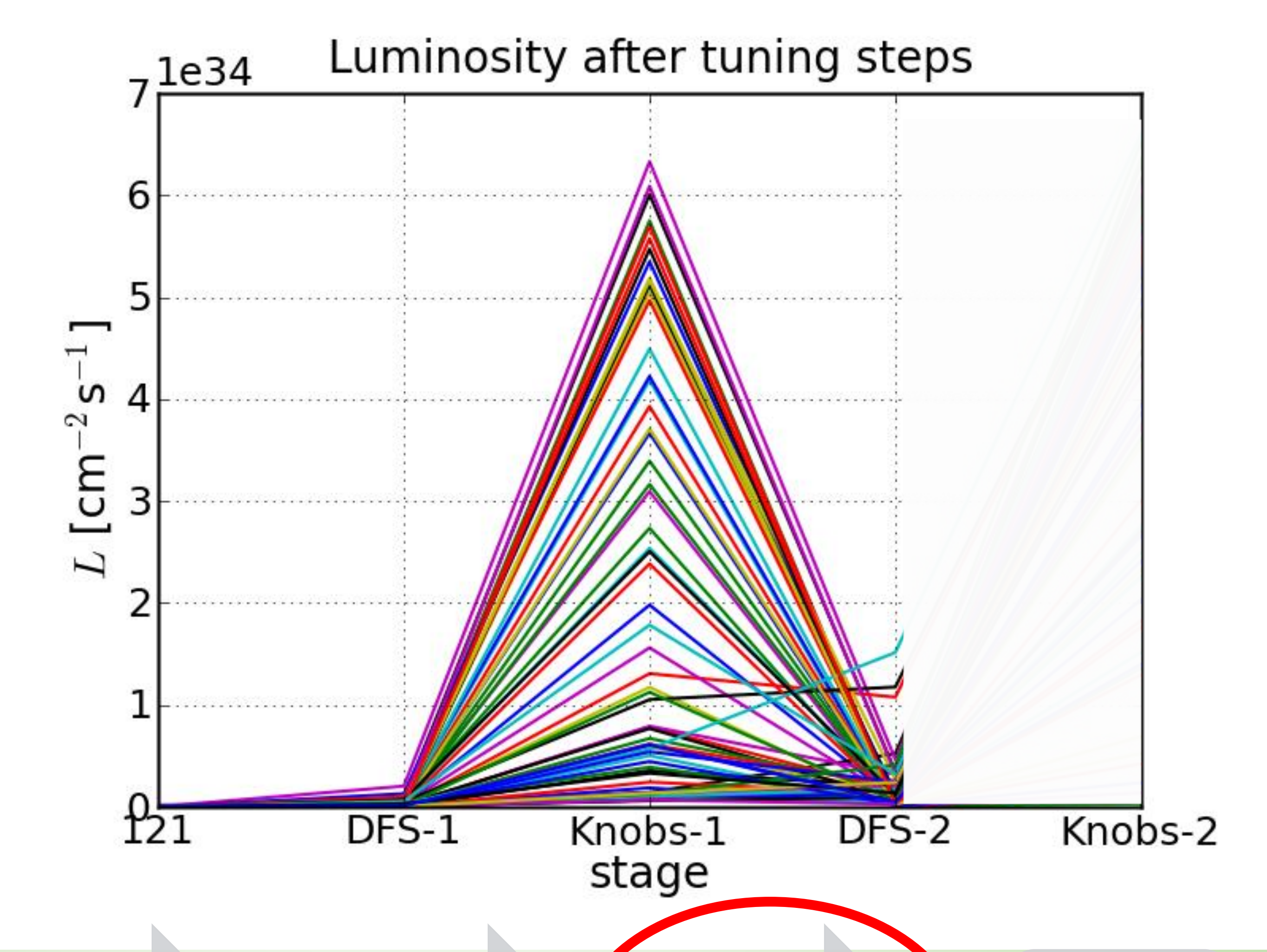
1. Apply the first corrections to the uncorrected beam
2. Correct the dispersion to approach design values
3. Use the tuning knobs to maximize luminosity

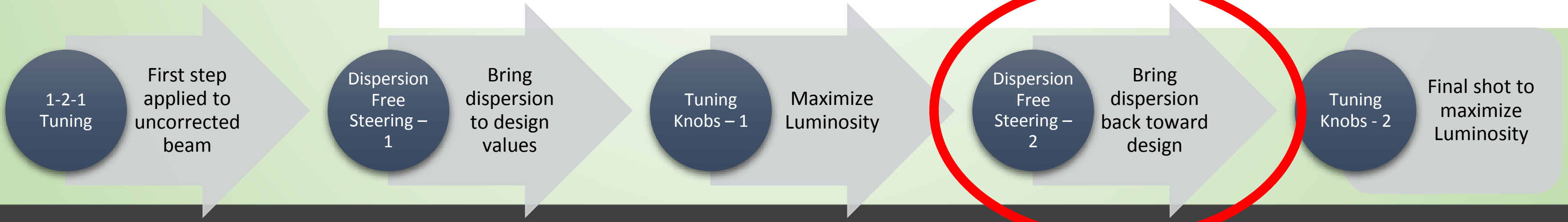


1-2-1 Tuning	First step applied to uncorrected beam
Dispersion Free Steering – 1	Bring dispersion to design values
Tuning Knobs – 1	Maximize Luminosity
Dispersion Free Steering – 2	Bring dispersion back toward design
Tuning Knobs - 2	Final shot to maximize Luminosity

Step by step…

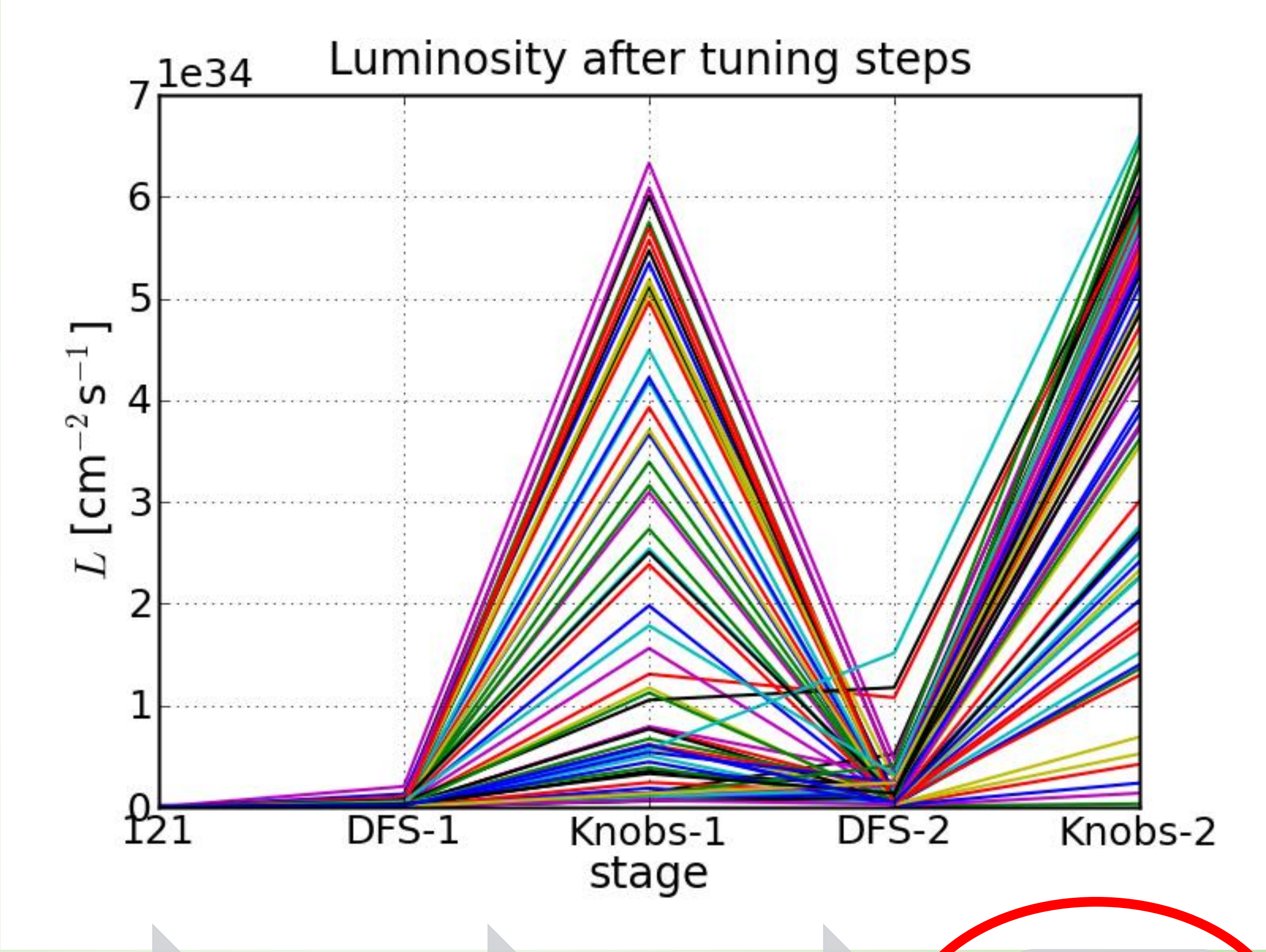
1. Apply the first corrections to the uncorrected beam
2. Correct the dispersion to approach design values
3. Use the tuning knobs to maximize luminosity
4. Attempt to optimize dispersion again

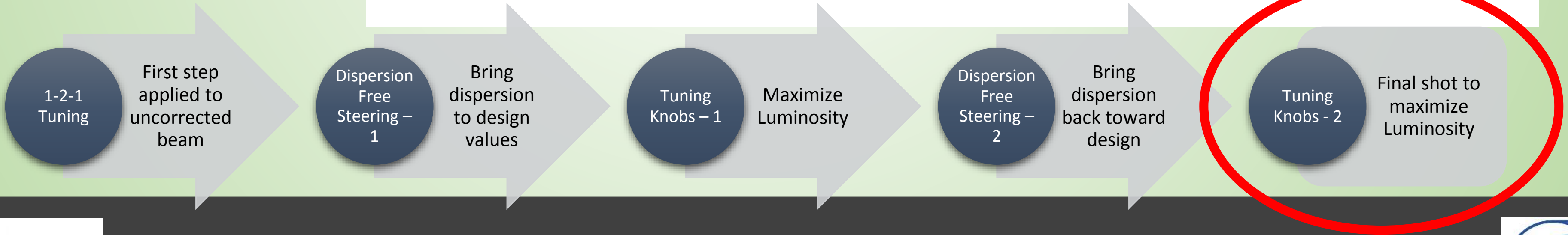




Step by step…

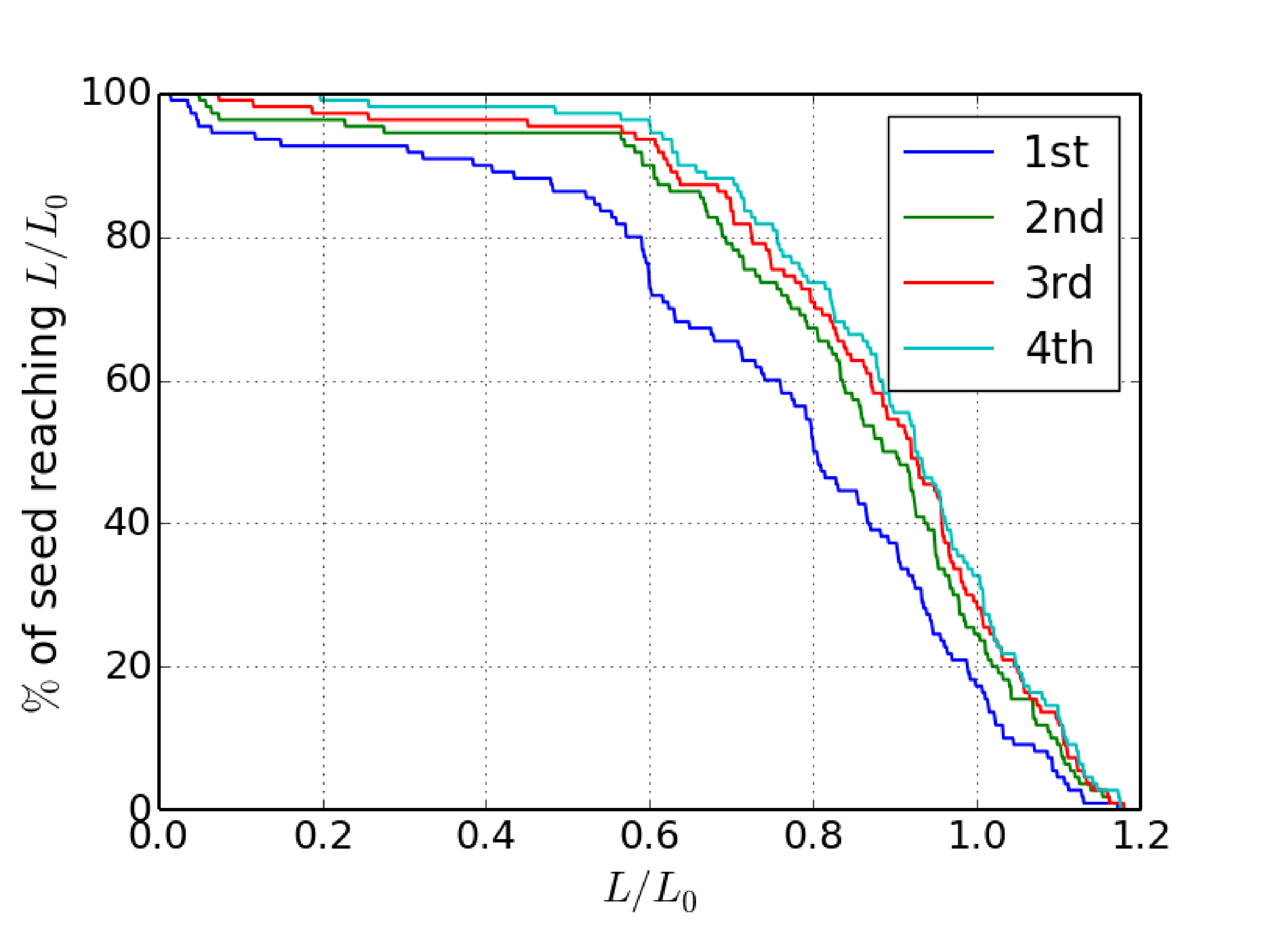
1. Apply the first corrections to the uncorrected beam
2. Correct the dispersion to approach design values
3. Use the tuning knobs to maximize luminosity
4. Attempt to optimize dispersion again
5. Maximize luminosity once more





Performing multiple iterations:

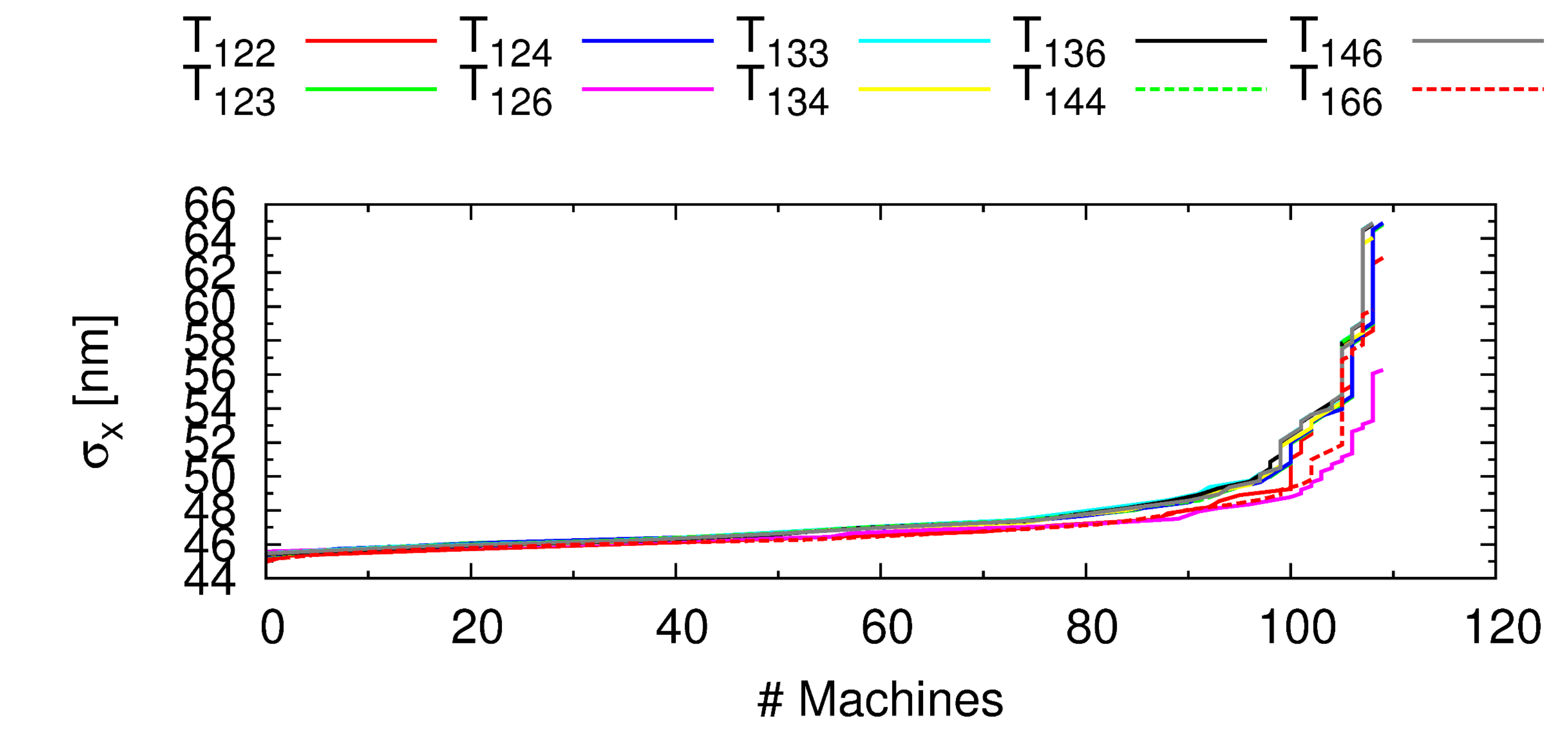
- Goal is 90% of seeds reaching 110% of the nominal luminosity.
- Good improvement for 2nd iteration.
  - Only small improvements with further iterations
  - Seeds with low luminosity improving
  - Need to address 2nd order corrections and correct specific aberrations



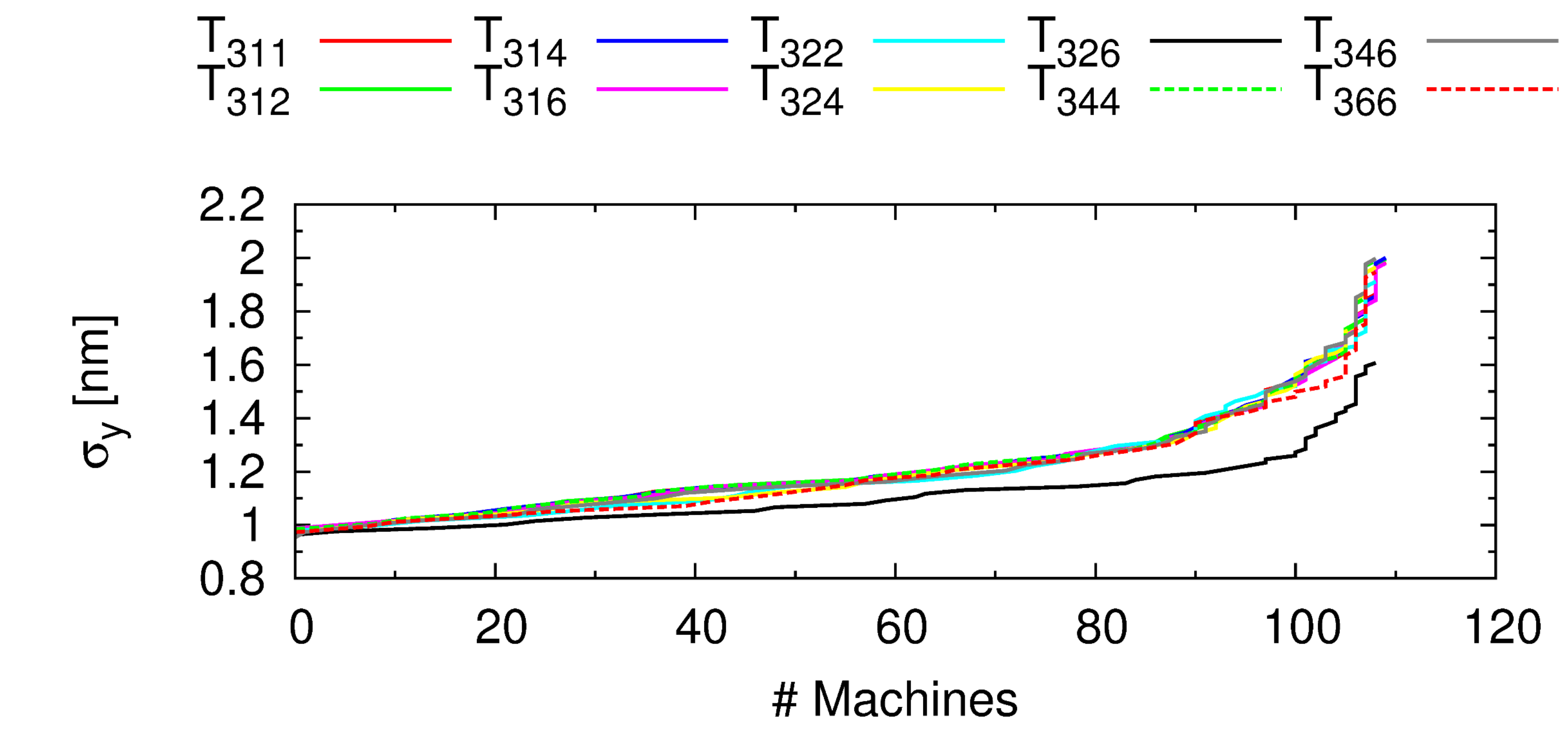
Aberration-targeted knobs

- Still nowhere near the goal of 90% of machines reaching 110% of the nominal luminosity
- To address this, need to design knobs which can correct for specific aberrations
  - Edu Marin Lacoma developed this method
- Analysis of the IP beam distributions identifies high order aberrations which can make further improvements of the luminosity measurement
  - In the X plane, these are: T126, T166, T122
  - In the Y plane, these are: T326
- Added dimensionless skew sextupoles to address aberrations

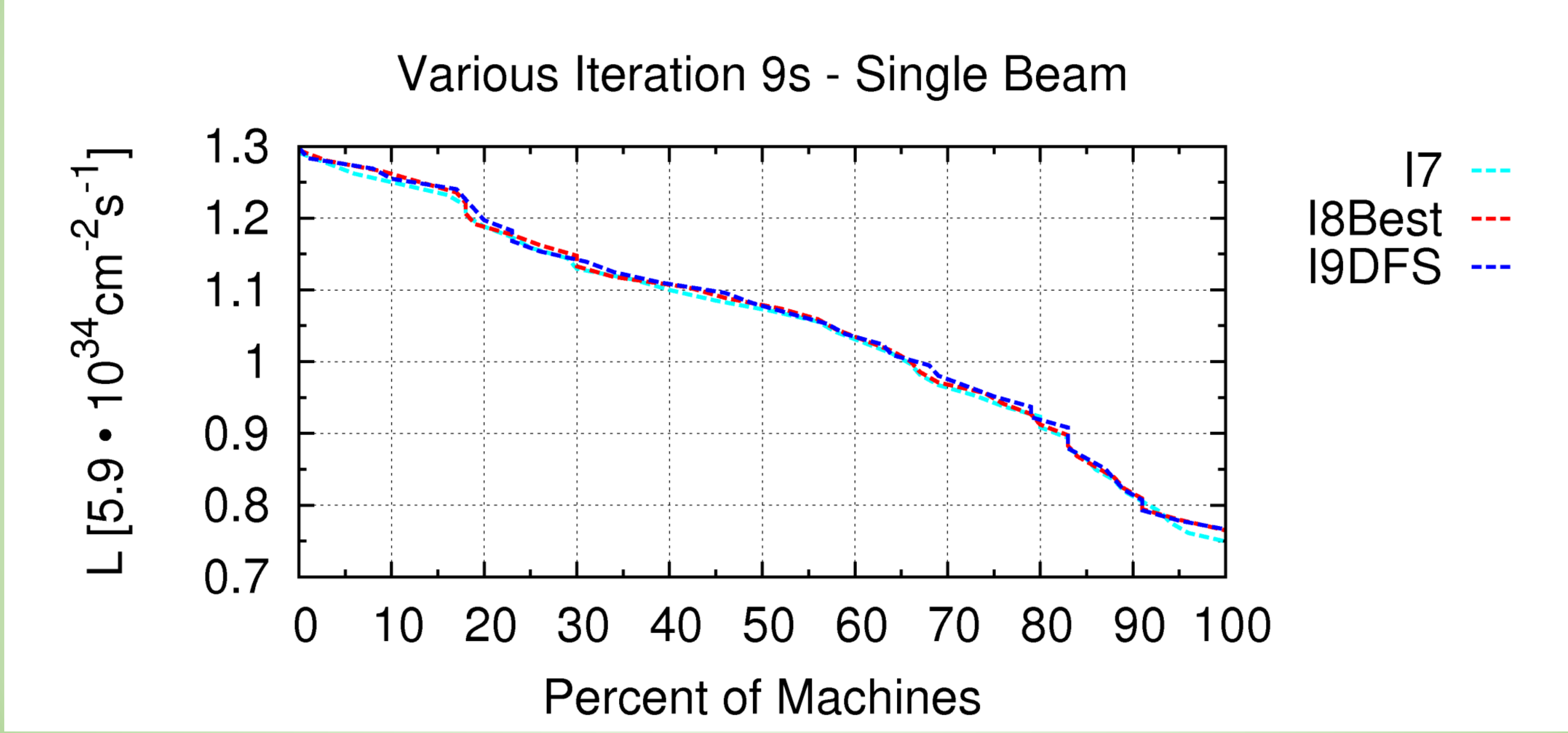
In the horizontal plane



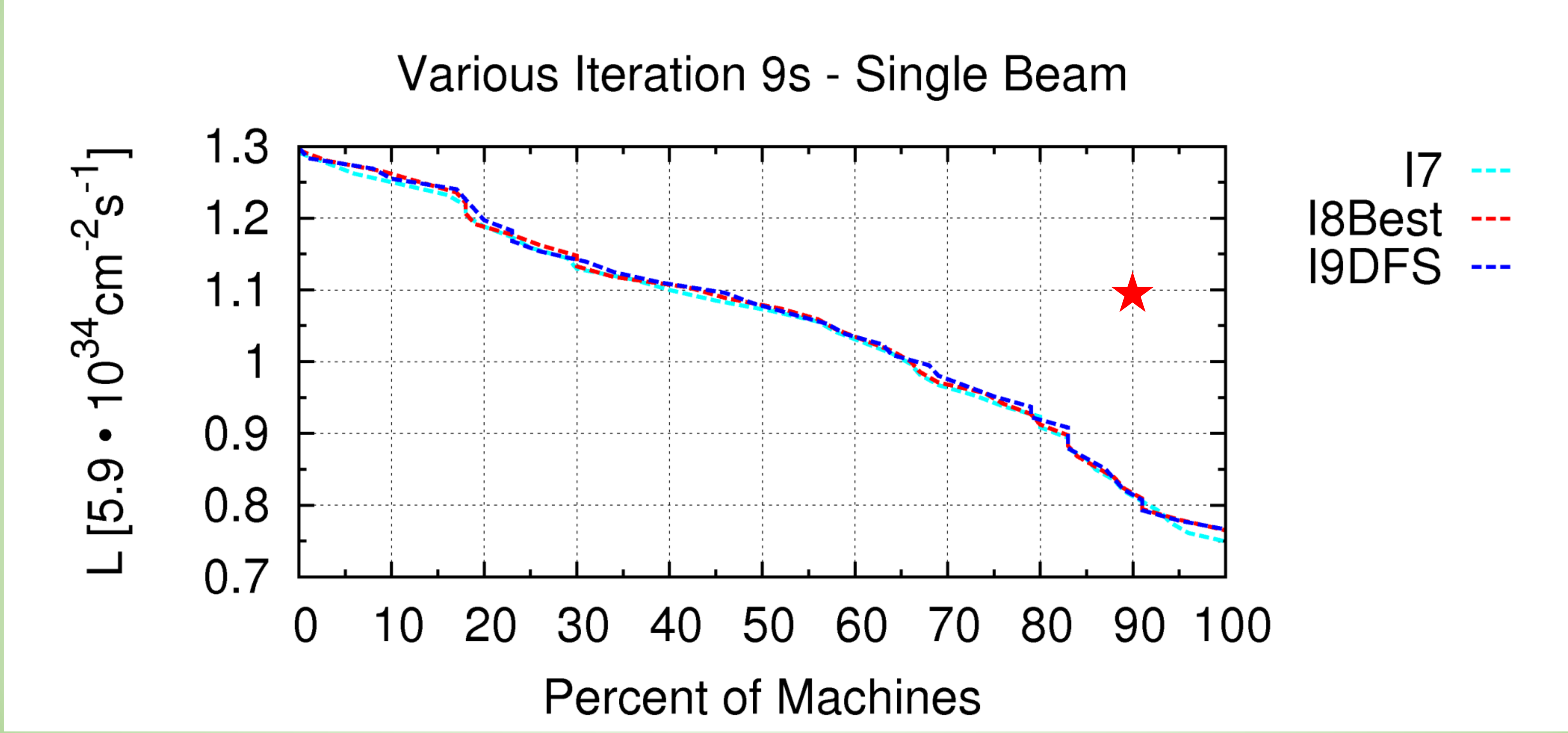
In the vertical plane



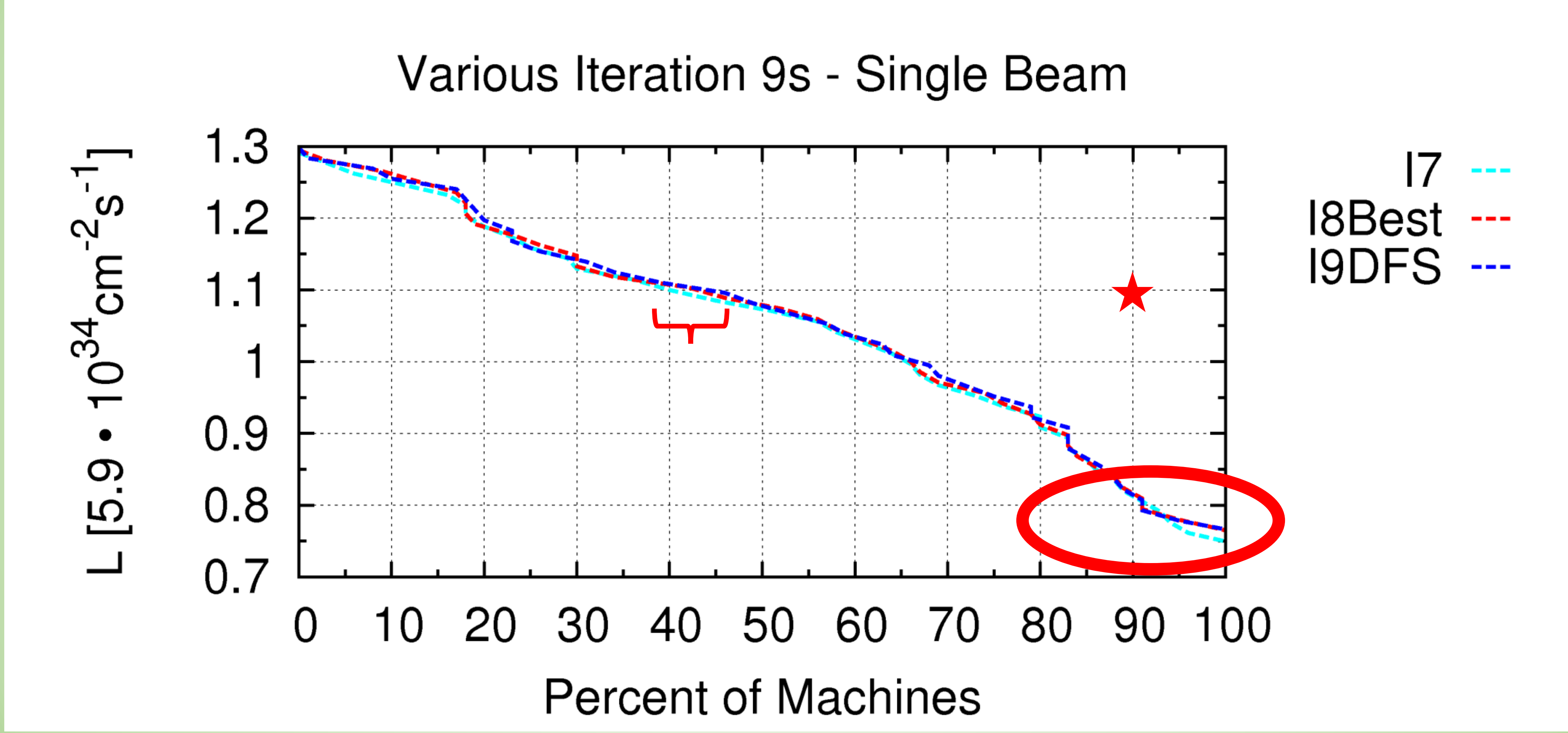
So, how did these changes perform?



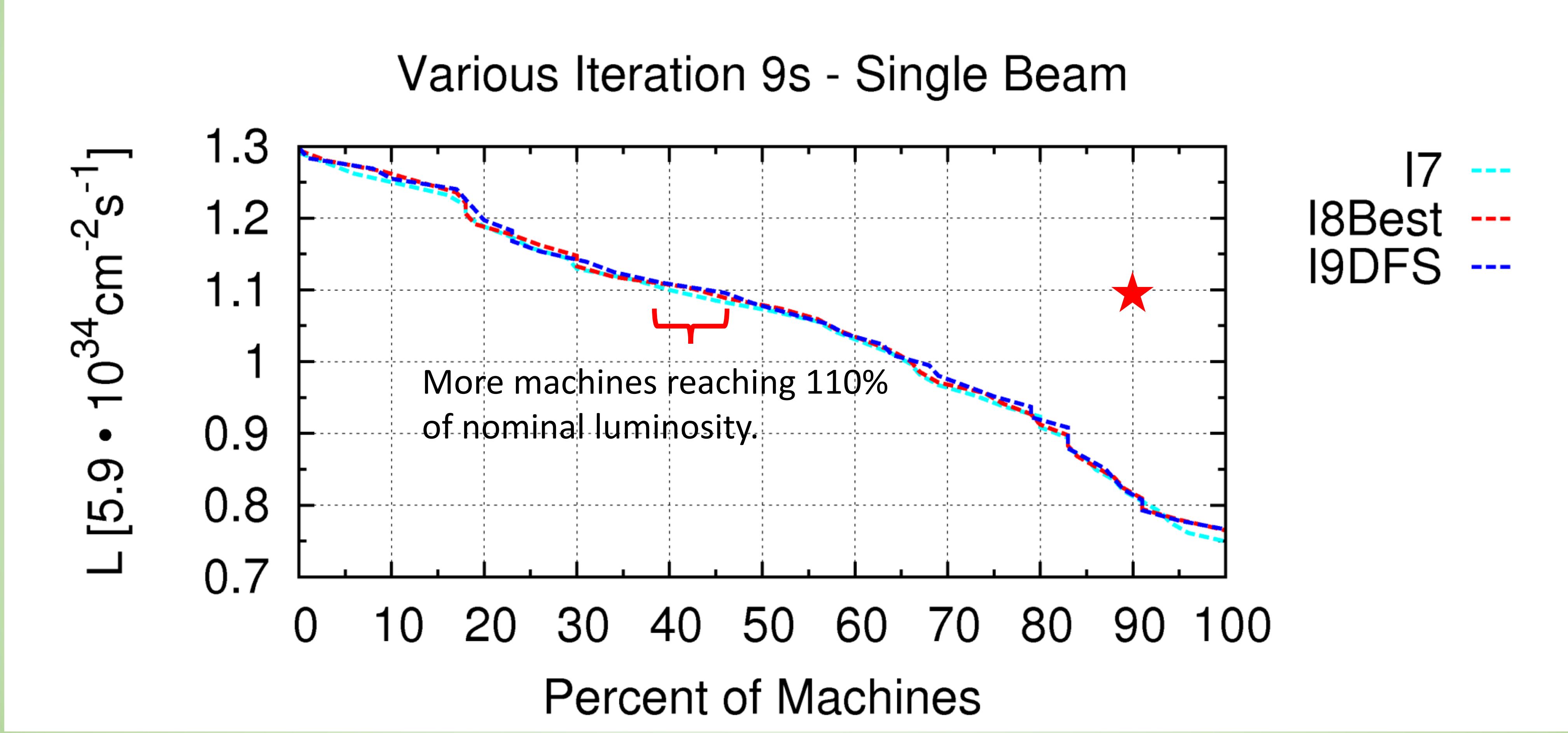
So, how did these changes perform?



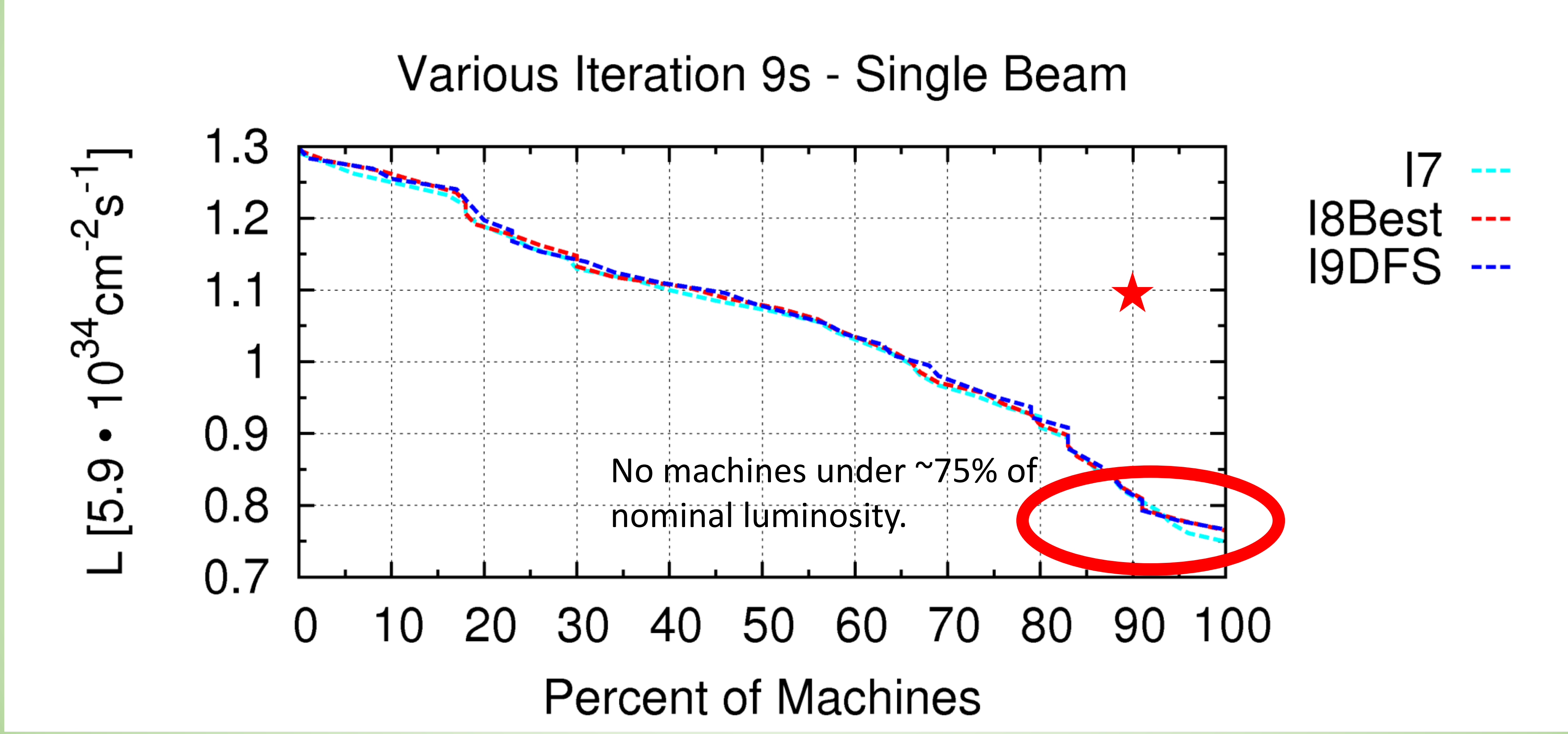
So, how did these changes perform?



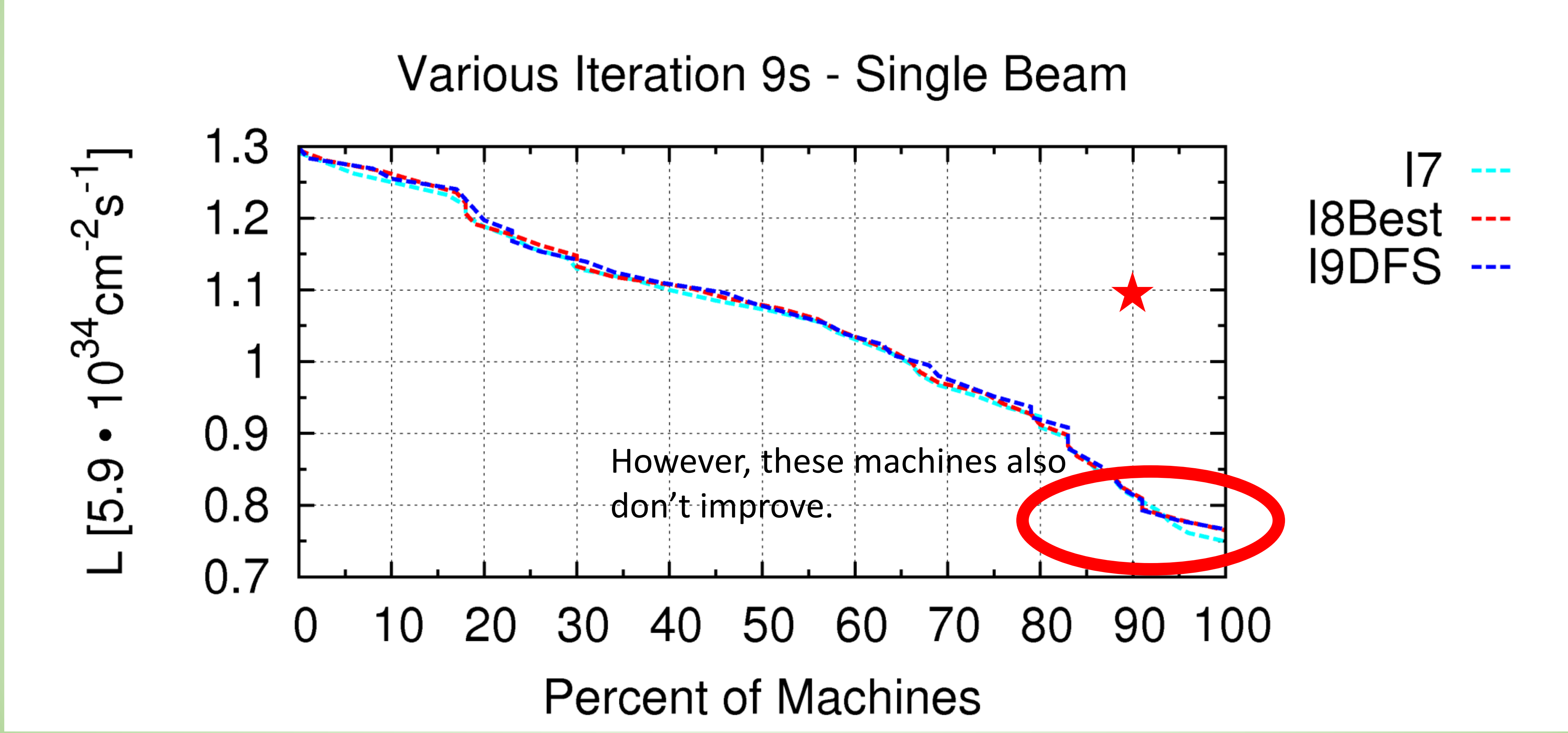
So, how did these changes perform?



So, how did these changes perform?



So, how did these changes perform?



Where does one go from here?

Starting at the beginning

```
##############################################################################
#
# Tuning studies for single beam
#
##############################################################################

# Disable multi-core option
#ParallelThreads -num 1

set e_initial 1500
set e0 $e_initial

set script_dir /afs/cern.ch/user/r/rbodenst/tuning/traditionaltuning_modular/traditionaltuning/
if {![file exist $script_dir]} {
    set script_dir [pwd]/..
}

# command-line options

# sr          : synchrotron radiation on/off
# gain1       : gain 1-to-1 steering
# gain2       : gain dfs
# iter1       : nr of iterations 1-to-1 steering
# iter2       : nr of iterations dfs
# sigma       : misalignment in um
# bpmres      : bpm resolution in um
# deltae      : energy difference for dfs
# beta        : normalisation parameter for dfs
# w1          : weight for first stage dfs
# w2          : weight for second stage dfs
# machine     : machine seed
# loadmachine : load machine status from file on/off
# repeat      : multipole alignment parameter
# dfs1        : first stage dfs on/off
# dfs2/dfs2knobs/dfs2hybrid  : second stage dfs: either dfs / dfs knobs / hybrid method
# dfsmeasure  : measure response matrix
# secondorder : apply second order knobs
```

```
array set args {
    nm 5
    sr 1
    gain1 0.5
    gain2 0.2
    iter1 100
    iter2 36
    sigma 10.0
    bpmres 0.000
    deltae 0.001
    beta 11
    w1 73
    w2 5
    machine all
    loadmachine 0
    repeat 1
    dfs1 1
    dfs2 0
    dfs2knobs 0
    dfs2hybrid 1
    dfsmeasure 0
    secondorder 0
}

array set args $argv

set nm $args(nm)
set gain1 $args(gain1)
set gain2 $args(gain2)
set iter1 $args(iter1)
set iter2 $args(iter2)
set sigma $args(sigma)
set bpmres $args(bpmres)
set deltae $args(deltae)
set w1 $args(w1)
set w2 $args(w2)
set beta $args(beta)
set sr $args(sr)
set machine $args(machine)
set loadmachine $args(loadmachine)
set repeat $args(repeat)
set dfs1 $args(dfs1)
set dfs2 $args(dfs2)
set dfs2knobs $args(dfs2knobs)
set dfs2hybrid $args(dfs2hybrid)
```

Starting at the beginning

```
##############################################################################
#
# Tuning studies for single beam
#
##############################################################################

# Disable multi-core option
#ParallelThreads -num 1

set e_initial 1500
set e0 $e_initial

set script_dir /afs/cern.ch/user/r/rbodenst/tuning/traditionaltuning_modular/traditionaltuning/
if {![file exist $script_dir]} {
    set script_dir [pwd]/..
}

# command-line options

# sr          : synchrotron radiation on/off
# gain1       : gain 1-to-1 steering
# gain2       : gain dfs
# iter1       : nr of iterations 1-to-1 steering
# iter2       : nr of iterations dfs
# sigma       : misalignment in um
# bpmres      : bpm resolution in um
# deltae      : energy difference for dfs
# beta        : normalisation parameter for dfs
# w1          : weight for first stage dfs
# w2          : weight for second stage dfs
# machine     : machine seed
# loadmachine : load machine status from file on/off
# repeat      : multipole alignment parameter
# dfs1        : first stage dfs on/off
# dfs2/dfs2knobs/dfs2hybrid  : second stage dfs: either dfs / dfs knobs / hybrid method
# dfsmeasure  : measure response matrix
# secondorder : apply second order knobs
```

```
array set args {
    nm 5
    sr 1
    gain1 0.5
    gain2 0.2
    iter1 100
    iter2 36
    sigma 10.0
    bpmres 0.000
    deltae 0.001
    beta 11
    w1 73
    w2 5
    machine all
    loadmachine 0
    repeat 1
    dfs1 1
    dfs2 0
    dfs2knobs 0
    dfs2hybrid 1
    dfsmeasure 0
    secondorder 0
}

array set args $argv

set nm $args(nm)
set gain1 $args(gain1)
set gain2 $args(gain2)
set iter1 $args(iter1)
set iter2 $args(iter2)
set sigma $args(sigma)
set bpmres $args(bpmres)
set deltae $args(deltae)
set w1 $args(w1)
set w2 $args(w2)
set beta $args(beta)
set sr $args(sr)
set machine $args(machine)
set loadmachine $args(loadmachine)
set repeat $args(repeat)
set dfs1 $args(dfs1)
set dfs2 $args(dfs2)
set dfs2knobs $args(dfs2knobs)
set dfs2hybrid $args(dfs2hybrid)
```

Starting at the beginning

```
################################################################################
#
# Tuning studies for single beam
#
################################################################################

# Disable multi-core option
#ParallelThreads -num 1

set e_initial 1500
set e0 $e_initial

set script_dir /afs/cern.ch/user/r/rbodenst/tuning/traditionaltuning_modular/traditionaltuning/
if {![file exist $script_dir]} {
    set script_dir [pwd]/..
}

# command-line options

# sr          : synchrotron radiation on/off
# gain1       : gain 1-to-1 steering
# gain2       : gain dfs
# iter1       : nr of iterations 1-to-1 steering
# iter2       : nr of iterations dfs
# sigma       : misalignment in um
# bpmres      : bpm resolution in um
# beta        : normalisation parameter for dfs
# w1          : weight for first stage dfs
# w2          : weight for second stage dfs
# loadmachine : load machine status from file on/off
# repeat      : multipole alignment parameter
# dfs1        : first stage dfs on/off
# dfs2/dfs2knobs/dfs2hybrid  : second stage dfs: either dfs / dfs knobs / hybrid method
# dfsmeasure  : measure response matrix
# secondorder : apply second order knobs
```

```
array set args {
    nm 5
    sr 1
    gain1 0.5
    gain2 0.2
    iter1 100
    iter2 36
    sigma 10.0
    bpmres 0.000
    beta 11
    w1 73
    w2 5
    loadmachine 0
    repeat 1
    dfs1 1
    dfs2 0
    dfs2knobs 0
    dfs2hybrid 1
    dfsmeasure 0
    secondorder 0
}

array set args $argv

set nm $args(nm)
set gain1 $args(gain1)
set gain2 $args(gain2)
set iter1 $args(iter1)
set iter2 $args(iter2)
set sigma $args(sigma)
set bpmres $args(bpmres)
set deltae $args(deltae)
set w1 $args(w1)
set w2 $args(w2)
set beta $args(beta)
set sr $args(sr)
set machine $args(machine)
set loadmachine $args(loadmachine)
set repeat $args(repeat)
set dfs1 $args(dfs1)
set dfs2 $args(dfs2)
set dfs2knobs $args(dfs2knobs)
set dfs2hybrid $args(dfs2hybrid)
```

UNIVERSITY OF OXFORD
JAI John Adams Institute for Accelerator Science
CERN
clic

Here’s the 121 Code:

Here’s the first place β is used

- This is essentially a weighting function for the kicks being applied
- Should be nonzero, or 121 tuning will not occur

```
puts "1:1 correction"

Octave {
    # multipoles off
    disp("Multipoles off")
    placet_element_set_attribute("test", MI, "strength", complex(0.0, 0.0));

    # 1:1 correction
    #disp("Start of 1:1 correction...")
    K0 = [ R0 ; $beta * I0 ];
    #disp("Strengths matrix ok!")

    for i=1:$iter1

    placet_test_no_correction("test", "beam0", "None");

    Log.EmittX(Log.index) += str2num(Tcl_GetVar("beam_emitt_x"));
    Log.EmittY(Log.index) += str2num(Tcl_GetVar("beam_emitt_y"));
    Log.SizeX(Log.index) += str2num(Tcl_GetVar("beam_size_x"));
    Log.SizeY(Log.index) += str2num(Tcl_GetVar("beam_size_y"));

    Measure0.R = placet_get_bpm_readings("test");

    b0 = [
          Measure0.R(:,1) - Response0.Model_R(:,1)
          Measure0.R(:,2) - Response0.Model_R(:,2)
         ];

    k0 = -[ R0 ; $beta * I0 ] \ [ b0 ; i0 ];

    placet_element_vary_attribute("test", DI, "strength_x", $gain1 * k0(1:M));
    placet_element_vary_attribute("test", DI, "strength_y", $gain1 * k0((M+1):end));

    chi2 = sum(b0 .** 2)

    Log.Chi(Log.index) += chi2;
    Log.Stage(Log.index) = 1;
    Log.index++;

    end
    disp("End of 1:1 correction")
    # end of 1:1

    # track a "fat" bunch beam0t to measure the beam size and the emittance
    disp("Multipoles on")
    placet_element_set_attribute("test", MI, "strength", complex(MS,0.0));
#     saveLumi(1);

#     STATUS = placet_element_get_attributes("test");
#     save -text machine_status_121.dat STATUS;
}
```

Here's the DFS1 Code:

Here both β and ω_1 are used

- Again, these are weighting factors, applied to a different correction scheme
- If ω_1 is zero, then DFS1 cannot occur, but 121 still can

```
puts "DFS, multipoles off"

Octave {
    disp("Multipoles off")
    placet_element_set_attribute("test", MI, "strength", complex(0.0,0.0));
    disp("Start of DFS")
    # DFS (if weight != 0.)
    if $w1 != 0

    #A0 = [ R0 ; $w1 * ETA0 ; $beta * I0 ];

    for i=1:$iter2
    placet_test_no_correction("test", "beam0", "None");

    Log.EmittX(Log.index) += str2num(Tcl_GetVar("beam_emitt_x"));
    Log.EmittY(Log.index) += str2num(Tcl_GetVar("beam_emitt_y"));
    Log.SizeX(Log.index) += str2num(Tcl_GetVar("beam_size_x"));
    Log.SizeY(Log.index) += str2num(Tcl_GetVar("beam_size_y"));

    Measure0.R = placet_get_bpm_readings("test");
    Measure0.Eta = MeasureDispersion("test", "beam1", "beam2", $deltae);

    b0 = [
          Measure0.R(:,1) - Response0.Model_R(:,1)
          Measure0.R(:,2) - Response0.Model_R(:,2)
         ];

    eta0 = [
            Measure0.Eta(:,1) - Response0.Model_Eta(:,1)
            Measure0.Eta(:,2) - Response0.Model_Eta(:,2)
           ];

    y0 =  [ b0 ; $w1 * eta0 ; $beta * i0 ];
    k0 = -[ R0 ; $w1 * ETA0 ; $beta * I0 ] \ y0;

    placet_element_vary_attribute("test", DI, "strength_x", $gain2 * k0(1:M));
    placet_element_vary_attribute("test", DI, "strength_y", $gain2 * k0((M+1):end));

    chi2 = sum(y0 .** 2)
    chi_R = sum(b0 .** 2);
    chi_ETA = sum(eta0 .** 2);

    Log.Chi(Log.index) += chi2;
    Log.Chi_R(Log.index) += chi_R;
    Log.Chi_ETA(Log.index) += chi_ETA;
    Log.Stage(Log.index) = 2;
    Log.index++;
    end
    end
    disp("End of DFS")

    disp("Multipoles on")
    placet_element_set_attribute("test", MI, "strength", complex(MS,0.0));

    # measure the beam size and the emittance

#    saveLumi(2);
#    STATUS = placet_element_get_attributes("test");
#    save -text machine_status_dfs1.dat STATUS;
}
```

Need to scan the parameters!

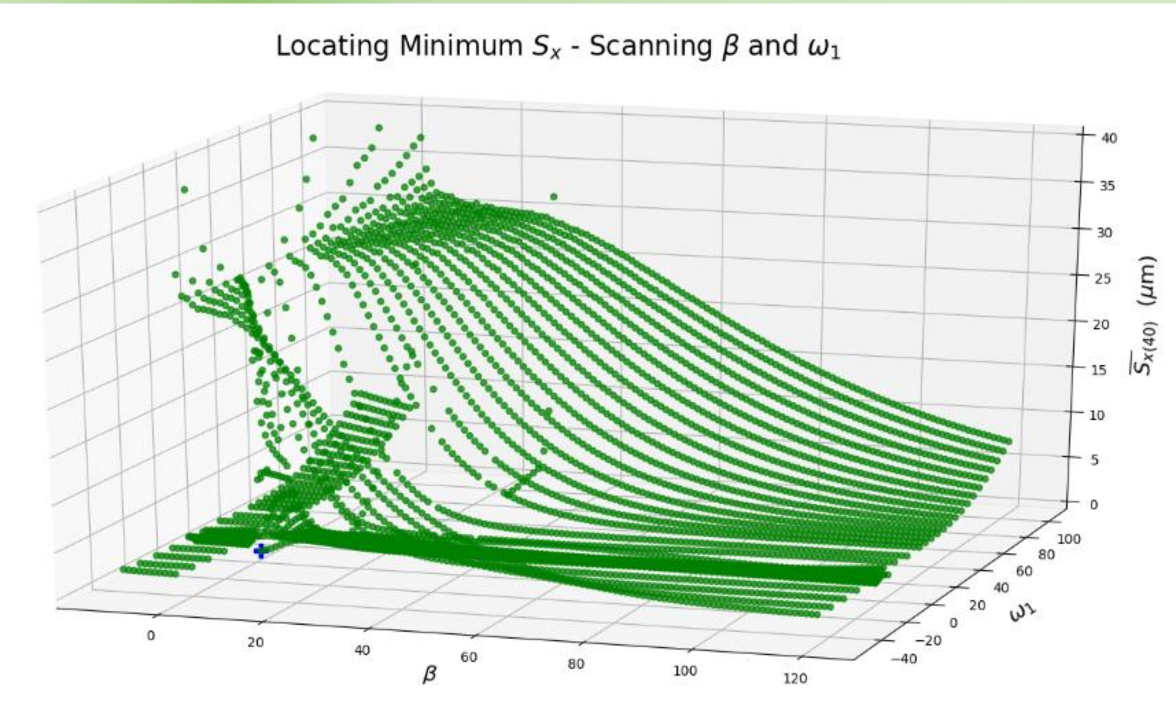
- Had assumed that these terms were already accurately determined
  - True for the Local scheme
  - Unsure if true for Traditional (evidence shows not true)
- Currently scanning β and ω_1 to see what values should actually be used
- Requires MANY batch jobs (currently 4320 jobs * 40 machines = 172,800 seeds, plus re-run failures, plus SR jobs)
- To save time, looking at beam size instead of luminosity
  - Two preliminary possibilities:
    - DFS1 may be making things worse
    - May have to tune each plane independently

$$\mathcal{L} = \frac{N_1 N_2 f n_b}{4\pi\sigma_x\sigma_y}$$

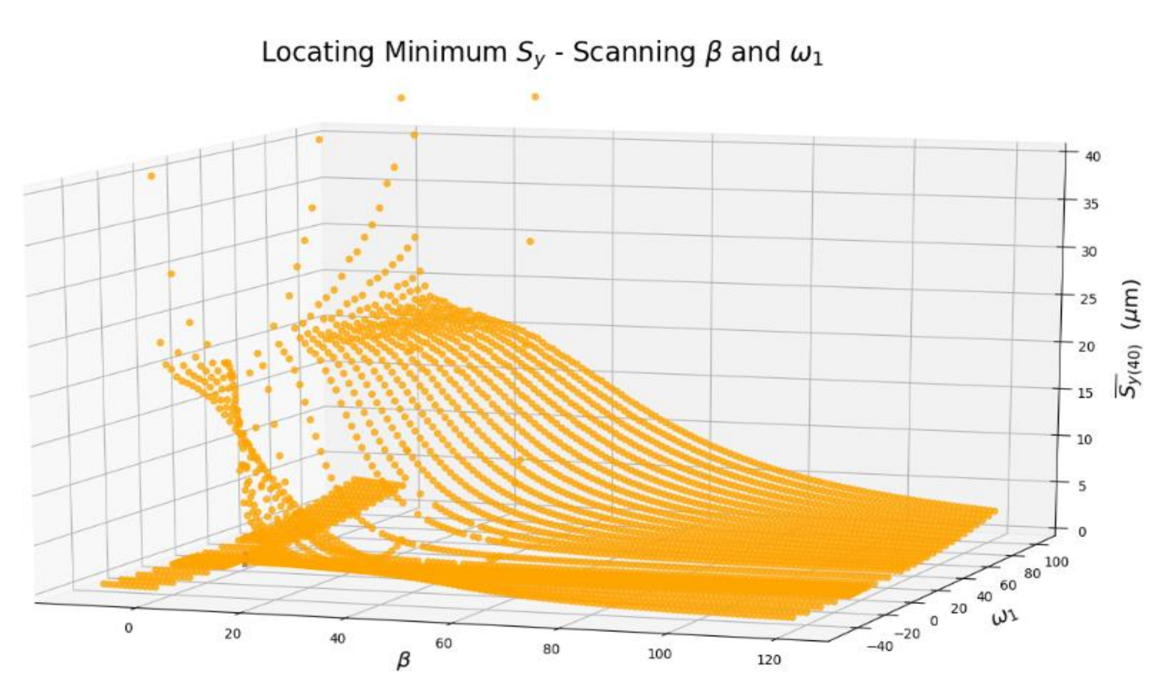
UNIVERSITY OF OXFORD

John Adams Institute for Accelerator Science

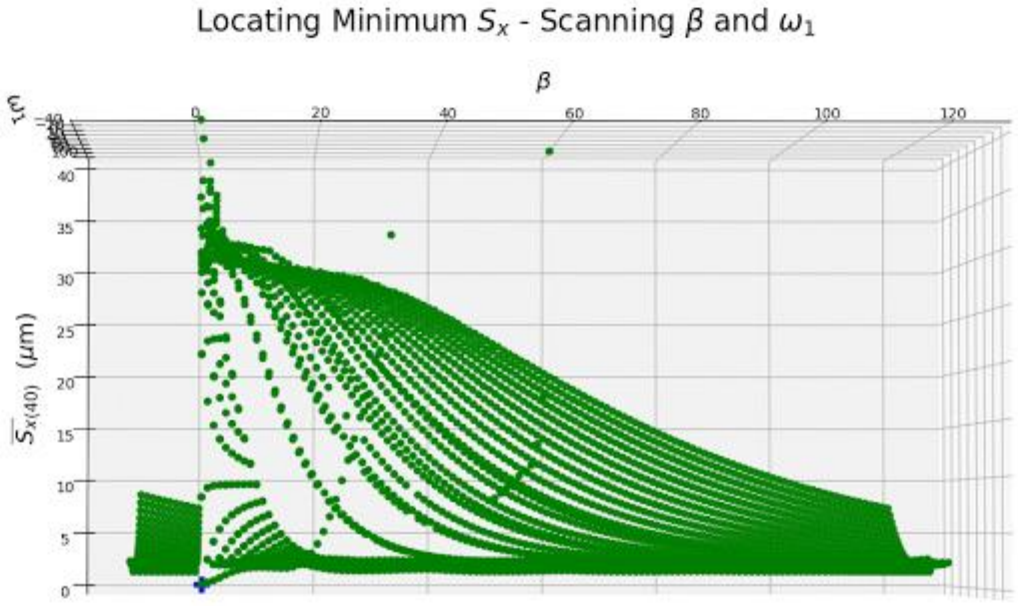
CERN



- Each point is the average of 40 machines (seeds)
- These results are preliminary – still running ~1200 jobs (40 machines each)

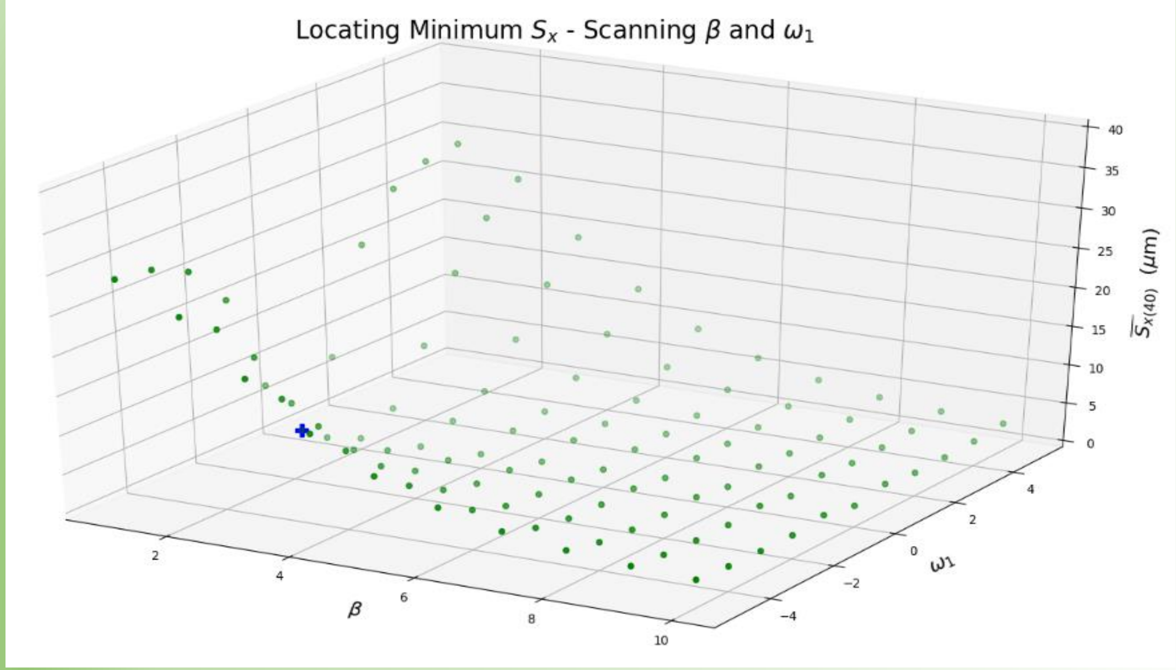


- When including $\omega_1 = 0$, best results:
  - $\beta = 1$, $\omega_1 = 0$
  - Sx = 0.462127568975
  - $\beta = 1$, $\omega_1 = 0$
  - Sy = 0.0779053202139
- When excluding $\omega_1 = 0$, best results:
  - $\beta = -2$, $\omega_1 = -5$
  - Sx = 1.56542499589
  - $\beta = 8$, $\omega_1 = 1$
  - Sy = 0.230448485582

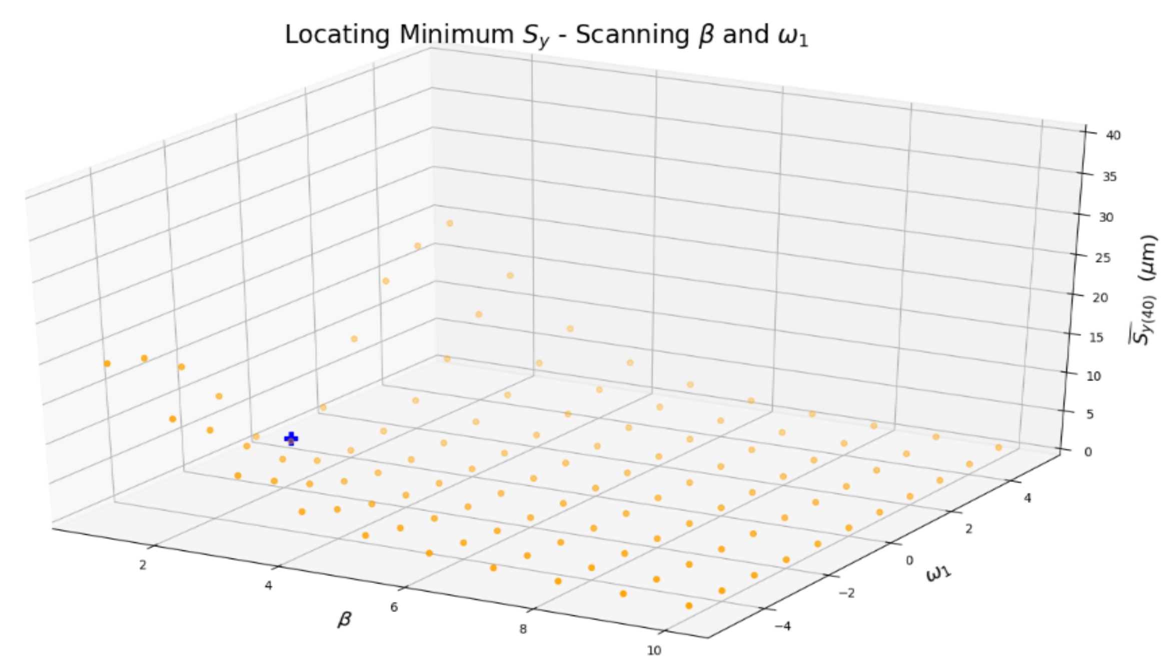


Please note: this is an animated plot in PowerPoint.

Sx scan including $\omega_1 = 0$



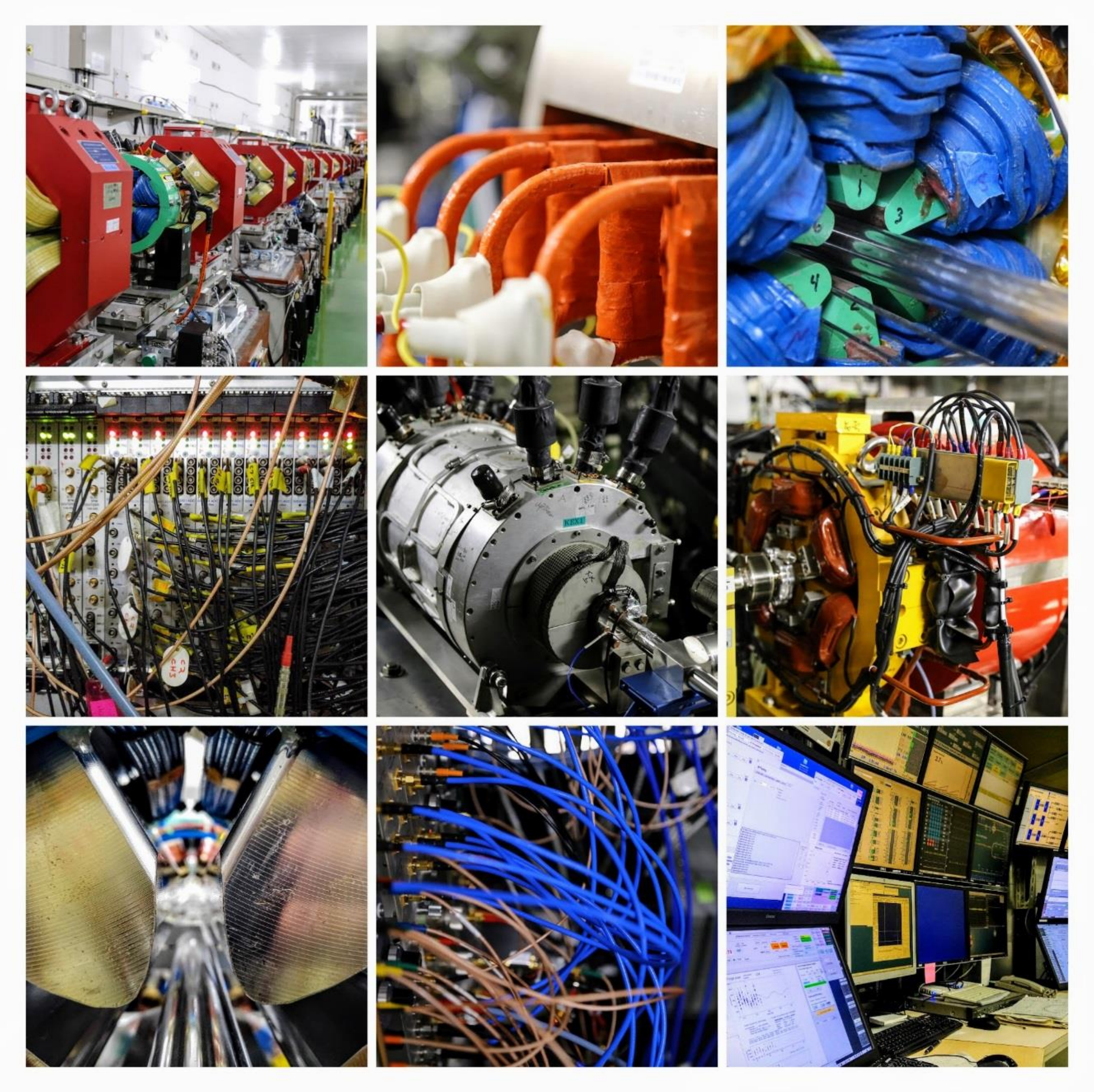
- Each point is the average of 40 machines (seeds)
- These results include synchrotron radiation and are focused on the region with the expected minimum



- Without synchrotron radiation, best results:
  - $\beta = 1$, $\omega_1 = 0$
  - Sx = 0.46127568975
  - $\beta = 1$, $\omega_1 = 0$
  - Sy = 0.0779053202139
- With synchrotron radiation, best results:
  - $\beta = 1$, $\omega_1 = 0$
  - Sx = 0.46180163949
  - $\beta = 1$, $\omega_1 = 0$
  - Sy = 0.077898236242

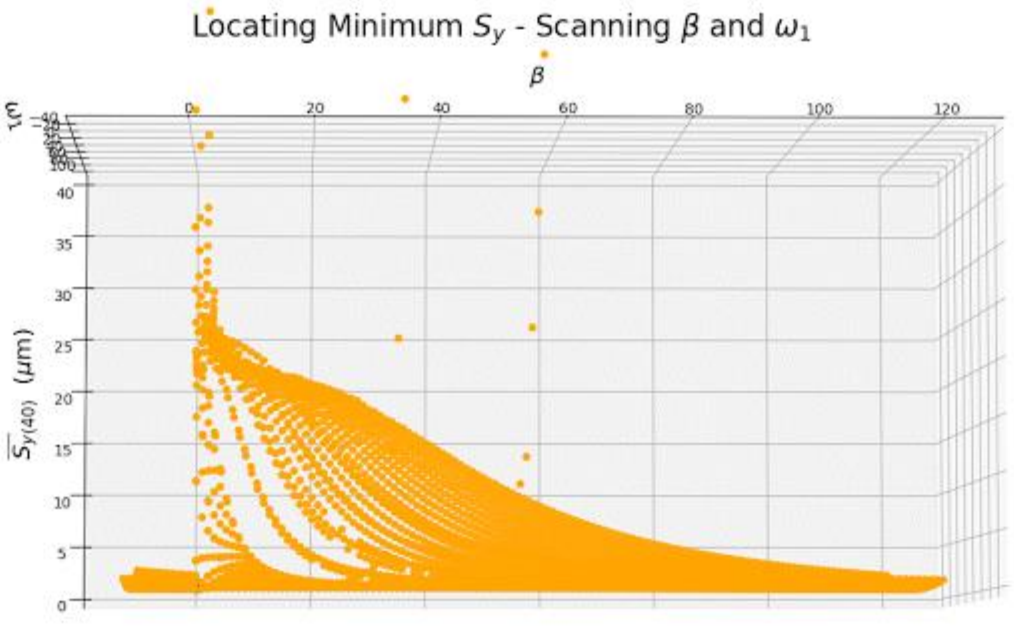
To be continued…

- It appears that the first stage DFS may be detrimental to tuning the traditional lattice, but this must be confirmed.
  - Will be running jobs with the new parameters based on luminosity measurements.
- Secondary results indicate that tuning may need to be separated into planes.
  - Must be investigated further.
- Must still scan ω_2 values for second-stage DFS tuning.
- Afterwards, will go through tuning knobs as before.

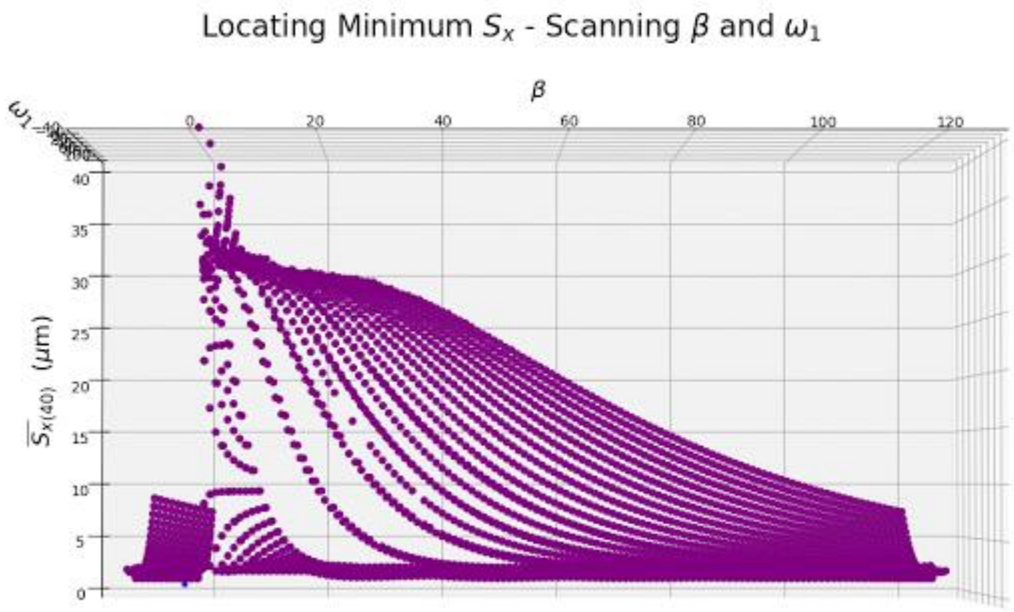


Thanks!

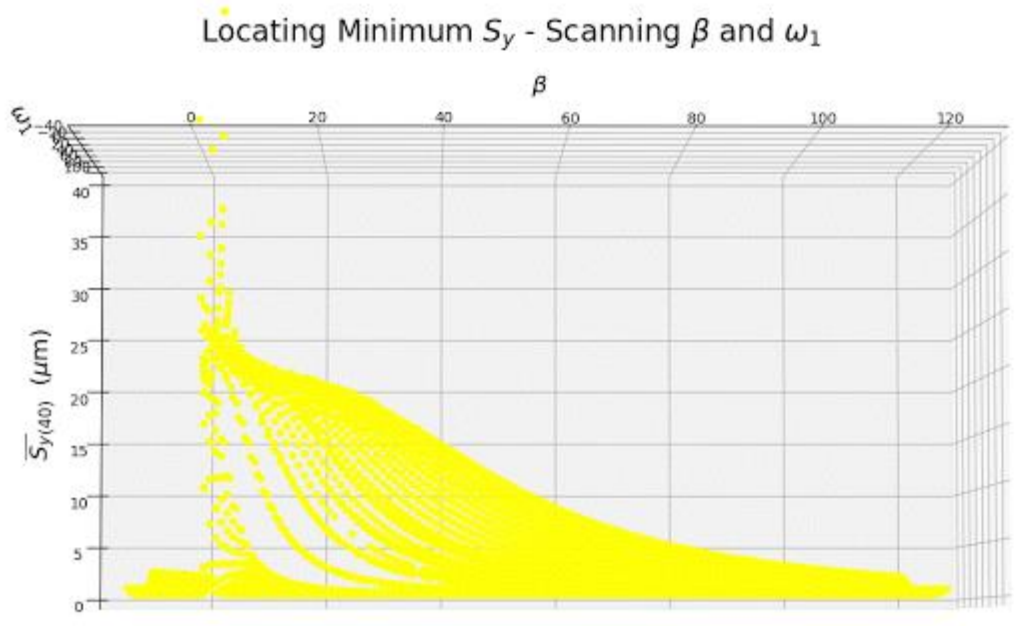
Extras



Sy scan including $\omega_1 = 0$



Sx scan excluding $\omega_1 = 0$



Sy scan excluding $\omega_1 = 0$

UNIVERSITY OF OXFORD

John Adams Institute for Accelerator Science

CERN