Followings are off-line (after the meeting by e-mail) Questions by K.Kubo and Answers by J.Jones on his presentation.

First round (Nov. 17, 2006):

Page 2:
Q: You mentioned "previous work", where can I find that? Any papers or talk slides?
A: I produced an EPAC paper showing some simulation work http://accelconf.web.cern.ch/AccelConf/e06/PAPERS/MOPLS117.PDF

Page 3:
Q: "Tuning knobs optimised using 1-Dimensional Nelder-Mead Simplex algorithm."
"1-Dimensional" means optimization is done for each knob one by one. Correct?
A: Correct

Q: What do x and y in the expression of "luminosity" represent? Beam sizes?
A: Yes, the beam sizes.

Page 4:
Q: "4 Linear knobs, 4 coupling knobs and 12 2nd order knobs created using all 4 degrees of freedom." What are these knobs?
A: The 4 linear knobs are the beta waists and the 2 dispersions. The coupling knobs are y'x, yx, y'x' and yx'. The 2nd order knobs are generally related to the 4 transverse coordinates.

Q: You mention "translations/rotations/field changes in the final 5 sextupoles".
The knobs are combinations of these changes? What does translation mean? (transverse position?)
A: Each knob uses one dimension only. But different knobs use different dimenisons.
Yes, transverse position.

Q: Same question for "non-linear, or nonorthogonal knobs"
A: Some of the knobs are either non-linear (in say T436) so the response is non-linear, some are not orthogonal to other important terms (i.e. T436 knob may vary the T442 term...)

Q: Are there total 42 knobs?
I guess every knob is optimized one by one, maximizing "luminosity", or looking at beam size at IP. Correct?
A: Theres about 20, but in the simulation I only used 14 due to nonlinear/non-orthogonal knobs. Yes, minimising the "luminosity" equation I give.

Q: These optimizations are done after orbit correction. Correct?
A: Yes. The jump in luminosity after each set of iterations (i.e. after 14 knobs) is because I then recorrect the trajectory, which changes the correction at the IP due to different orbit in sextupoles.

Page 5:
Q: beam_0 is the 6x6 transfer matrix from the entrance of BDS to IP without errors. beam_err is the 6x6 transfer matrix with machine errors. Correct?
A: No. I track several thousant particles down the line with no errors and record their final coordinates. This is beam_0, and is 6*no particles long. I then track the same beam with errors in the lattice and record final coordinates, this is beam_err. So the final matrices are (assuming 5000 particles) 5000*6 big. The R matrix is then a least squares fit of the transformation from beam_err to beam_0 using SVD.

Q: "4 response matrices (one for each degree of freedom),"
I thought there are only three degrees of freedom (x, y and z). Why four?
A: Sorry, thats x, y, field change and rotation of sextupoles.

Q: What is the dimension of the matrices (6x6 or 4x4 or 2x2)? Can you explain how to create tuning knobs from the matrices?
A: The matrices are 6x6 which I flatten to 1x36.I then use an SVD inversion of the 20 knobs(4 degrees of freedom * 5 sextupoles) * 36 matrix elements to create each of the knobs i.e. A= 20*36 matrix. Tuning knob 1(which is xx) would be K = A^-1 . {1,0,0,0,0,0....0}

Q: Can you explain why you can improve orthogonality by choosing 17? Why you do not need 20 knobs (4x5: we do not care energy and z at IP, then 4. z at the entrance can be ignored, then 5.) ? (In other words, Why you do not need yx, yy' and y'x' ?)
A: I chose all of my knobs just based on empirical results. i.e. I didnt choose yx etc because they didnt seem to improve things. In theory probably the 20 knobs would be better, but it maybe that they were not orthogonal so didnt help to improve the final luminosity so I ignored them. This needs more work on making the knobs more orthogonal by some other means...

Q: Same question as in page 4:
What are these knobs? You mention "translations/rotations/field changes in the final 5 sextupoles". The knobs are combinations of these changes? What does translation mean? (transverse position?)
A: Yes, these knobs are different combinations of each different changes. Yes, translation = transverse position

Q: Can you introduce any references for Rotation Matrix?
A: I sort of made it up, but that obviously doesnt mean someone else hasn't done it, I've just never seen it.

Page 7:
Q: Simplex algorithm means: minimizing beam size (maximizing luminosity) changing each knob one by one. Correct?
A: In this section I use the simplex to change each of the basic knobs but as one big optimisation. So it is a simplex with 21 dimensions with each point being a combination of the 4 different degrees of freedom and the 5 sextupoles. So each point in the optimisation is 20 different numbers each number the value of , say, the x position of the first sextupole. So the optimiser has 20 values * 21 different points. You can also run it by having say 5 numbers for the x position of each sextupole, then optimise the luminosity, then another optimisation of the y values of each sextupole etc...

Q: "Optimise all degrees of freedom at once"
"Optimise each degree of freedom separately"
Can you explain what "degree of freedom" means? How many freedoms?
I thought you just maximized "luminosity".

A: Sorry, in this case I mean optimise either the x position AND the y position AND the field strength AND the rotation of ALL 5 sextupoles at once, or do the x position THEN the y position etc. The same explanation as for the last question. But I always use the luminosity as the result to optimise.

Page 8:
Q: I guess you used 21 knobs, ten iterations. Correct?
A: No. As in the last question, I used all of the different sextupoles x, y, field and rotation values in a big simplex. It just happens to have ~210 iterations to converge...

Q: Same question as in page 4:
What are these knobs? You mention "translations/rotations/field changes in the final 5 sextupoles". The knobs are combinations of these changes?
A: Yes.

Second round (Nov. 20, 2006):

Q:
Am I correct if I say as follows?

There are 20 different beam_err and R (4 degrees of freedom * 5 sextupoles), say beam_err-n and R_n (n=1,2,3,, ,20). You set, e.g., x offset of 1st sextupole and get 5000*6 dimensional "vector", beam_err-1, then 6x6 matrix R_1. Set y offset of 1st sextupole and get another "vector" beam_err-2 and R_2. And so on. Flattening R_n and get 20x36 matrix A, where, n-th row of A is flattened R_n.

Then (if it is correct), two more questions:
How to choose the initial parameters of the 5000 particles?
How to choose the amount of "errors", e.g. how large offset was set for each sextupole to get beam_err?

A:
Yes, that sounds correct. I choose the initial beam coordinates just as a Gaussian distributed beam at the entrance of the FF section, created with the correct twiss parameters.
The error amplitude is a bit empirical. So for x & y I use 0.5mm, 0.5mrad for rotation, and 0.05% field error. This is bigger than what is required from the simulation results, which are about 10 times lower values. I may need to optimise this number more, but so far the tuning knobs seem to be fairly linear over the region required, so I have yet to do this fully.

– - - - - - - - - end - - - - - - -
–