Implementation of fully analytic orbit response analysis in python

Formulas used in this slides are derived from: https://arxiv.org/abs/1711.06589v2

S.Liuzzo, A.Franchi, February 13th 2024

ESRF | The European Synchrotron

Analytic formulas for the rapid evaluation of the orbit response matrix and chromatic functions from lattice parameters in circular accelerators

Andrea Franchi and Simone Maria Liuzzo
ESRF, CS 40220, 38043 Grenoble Cedex 9, France

Zeus Martí
CELLS, 08193 Bellaterra, Spain
(Dated: September 23, 2018)

OUTLINE

- Part 1: The implementation (12’)
- Part 2: The repository (1’)
- Part 3: outlook (3’)

OUTLINE

- Part 1: The implementation (12’)
- Part 2: The repository (1’)
- Part 3: outlook (3’)

JACOBIAN OF THE ORBIT RESPONSE MATRIX

- In order to perform a linear lattice modelling and correction, the Jacobian of the ORM needs to be computed, SVD pseudo-inverted & applied to the measured ORM & dispersion.

$$\begin{pmatrix} \delta\vec{O}^{(xx)} \\ \delta\vec{O}^{(yy)} \\ \delta\vec{D}_x \end{pmatrix} = \mathbf{N} \begin{pmatrix} \delta\vec{K}_1 \\ \delta\vec{K}_0 \end{pmatrix}$$

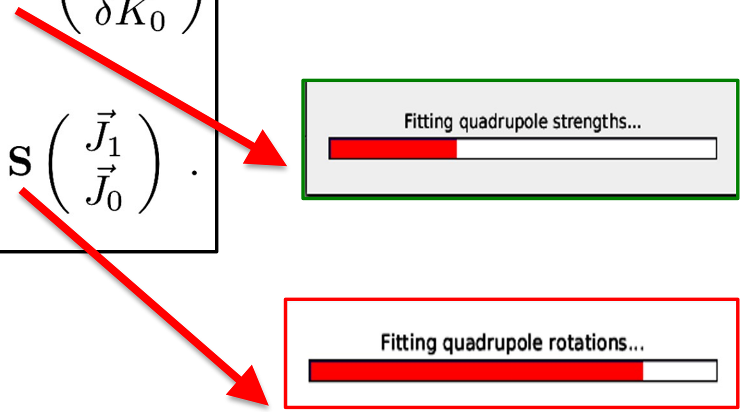
$$\begin{pmatrix} \delta\vec{O}^{(xy)} \\ \delta\vec{O}^{(yx)} \\ \delta\vec{D}_y \end{pmatrix} = \mathbf{S} \begin{pmatrix} \vec{J}_1 \\ \vec{J}_0 \end{pmatrix} .$$

JACOBIAN OF THE ORBIT RESPONSE MATRIX

- **In order to perform a linear lattice modelling and correction, the Jacobian of the ORM needs to be computed, SVD pseudo-inverted & applied to the measured ORM & dispersion.**
- **Numerical Jacobian: replicate in simulation the measurement: For each variation of the quadrupole strength, compute ORM from orbit calculations @ each steerer variation.**
  - ✓ Pros: accurate, can be parallelized
  - ✓ Cons: time consuming, if optics or orbit unstable for quadrupole variation it can fail.

$$\begin{pmatrix} \delta\vec{O}^{(xx)} \\ \delta\vec{O}^{(yy)} \\ \delta\vec{D}_x \end{pmatrix} = \mathbf{N} \begin{pmatrix} \delta\vec{K}_1 \\ \delta\vec{K}_0 \end{pmatrix}$$

$$\begin{pmatrix} \delta\vec{O}^{(xy)} \\ \delta\vec{O}^{(yx)} \\ \delta\vec{D}_y \end{pmatrix} = \mathbf{S} \begin{pmatrix} \vec{J}_1 \\ \vec{J}_0 \end{pmatrix} .$$



JACOBIAN OF THE ORBIT RESPONSE MATRIX

- **In order to perform a linear lattice modelling and correction, the Jacobian of the ORM needs to be computed, SVD pseudo-inverted & applied to the measured ORM & dispersion.**
- **Numerical Jacobian: replicate in simulation the measurement: For each variation of the quadrupole strength, compute ORM from orbit calculations @ each steerer variation.**
  - ✓ Pros: accurate, can be parallelized
  - ✓ Cons: time consuming, if optics or orbit unstable for quadrupole variation it can fail.
- **Pseudo-analytic Jacobian: use textbook formulas to evaluate ORM after computing Twiss parameters at each quadrupole variation.**
  - ✓ Pros: faster than numerical, can be parallelized
  - ✓ Cons: no analytic formulas for off-diagonal ORM blocks (coupling), it can fail if optics or orbit unstable for quadrupole variation.

$$M_{ij} = \frac{\sqrt{\beta_j \beta_i}}{2 \sin \pi Q} \cos\left(|\phi_j - \phi_i| - \pi Q\right).$$

JACOBIAN OF THE ORBIT RESPONSE MATRIX

- **In order to perform a linear lattice modelling and correction, the Jacobian of the ORM needs to be computed, SVD pseudo-inverted & applied to the measured ORM & dispersion.**
- **Numerical Jacobian: replicate in simulation the measurement: For each variation of the quadrupole strength, compute ORM from orbit calculations @ each steerer variation.**
  - ✓ Pros: accurate, can be parallelized
  - ✓ Cons: time consuming, if optics or orbit unstable for quadrupole variation it can fail.
- **Pseudo-analytic Jacobian: use textbook formulas to evaluate ORM after computing Twiss parameters at each quadrupole variation.**
  - ✓ Pros: faster than numerical, can be parallelized
  - ✓ Cons: no analytic formulas for off-diagonal ORM blocks (coupling), it can fail if optics or orbit unstable for quadrupole variation.
- **Fully analytic Jacobian: evaluate directly the Jacobian from Twiss parameters of the initial model (ideal or from beam-based measurements)**
  - ✓ Pros: coupling & dispersion included, only one computation of Twiss parameters needed, no orbit calculation needed, faster than pseudo-analytic, can be parallelized
  - ✓ Cons: tedious to code.

JACOBIAN OF THE ORBIT RESPONSE MATRIX

- **In order to perform a linear lattice modelling and correction, the Jacobian of the ORM needs to be comput**

$$N_{wj,m}^{(xx)} \simeq -\frac{\sqrt{\beta_{j,x}^{(mod)}\beta_{w,x}^{(mod)}}\beta_{m,x}^{(mod)}}{2\sin\left(\pi Q_x^{(mod)}\right)}\left\{\frac{\cos\left(\tau_{x,wj}^{(mod)}\right)}{4\sin\left(2\pi Q_x^{(mod)}\right)}\left[\cos\left(2\tau_{x,mj}^{(mod)}\right)+\cos\left(2\tau_{x,mw}^{(mod)}\right)\right]\right.$$

$$+\frac{\sin\left(\tau_{x,wj}^{(mod)}\right)}{4\sin\left(2\pi Q_x^{(mod)}\right)}\left[\sin\left(2\tau_{x,mj}^{(mod)}\right)-\sin\left(2\tau_{x,mw}^{(mod)}\right)\right]$$

$$\left.+\frac{1}{2}\sin\left(\tau_{x,wj}^{(mod)}\right)\left[\Pi(m,j)-\Pi(m,w)+\Pi(j,w)\right]+\frac{\cos\left(\Delta\phi_{x,wj}^{(mod)}\right)}{4\sin\left(\pi Q_x^{(mod)}\right)}\right\},$$

w -> steerer

j -> BPM

m -> magnet

- **Numeri strengt**
  - ✓ Pros: ac
  - ✓ Cons: ti
- **Pseudo each qu** **t**
  - ✓ Pros: fa
  - ✓ Cons: n
- **Fully analytic Jacobian: evaluate directly the Jacobian** **or from beam-based measurements)**
  - ✓ Pros: coupling & dispersion included, only one computation of Twiss pseudo-analytic, can be parallelized
  - ✓ Cons: tedious to code.

$$\Pi(a,b)=1 \quad \text{if } s_a < s_b\ , \qquad \Pi(a,b)=0 \quad \text{if } s_a \geq s_b$$

$$\tau_{z,ab}=\Delta\phi_{z,ab}-\pi Q_z\ , \qquad z=x,y$$

JACOBIAN OF THE ORBIT RESPONSE MATRIX

- **In order to perform a linear lattice modelling and correction, the Jacobian of the ORM needs to be computed, SVD pseudo-inverted & applied to the measured ORM & dispersion.**
- **Numerical Jacobian: replicate in simulation the strength, compute ORM from orbit calculations**
  - ✓ Pros: accurate, can be parallelized
  - ✓ Cons: time consuming, if optics or orbit unstable for quadru
- **Pseudo-analytic Jacobian: use textbook formul each quadrupole variation.**
  - ✓ Pros: faster than numerical, can be parallelized
  - ✓ Cons: no analytic formulas for off-diagonal ORM blocks (co
- **Fully analytic Jacobian: evaluate directly the Ja or from beam-based measurements)**
  - ✓ Pros: coupling & dispersion included, only one computation pseudo-analytic, can be parallelized
  - ✓ Cons: tedious to code.

$$\beta_m \longrightarrow I_{\beta,m} = \frac{1}{L_m}\int_0^{L_m} \beta(s)\ ds\ ,$$

$$\beta_m \sin(2\tau_{mj}) \longrightarrow I_{S,mj} = \frac{1}{L_m}\int_0^{L_m} \beta(s)\sin(2\tau_{sj})\ ds$$

$$\beta_m \cos(2\tau_{mj}) \longrightarrow I_{C,mj} = \frac{1}{L_m}\int_0^{L_m} \beta(s)\cos(2\tau_{sj})\ ds$$

$$\sqrt{\beta_m}\sin(\tau_{mj}) \longrightarrow J_{C,mj} = \frac{1}{L_m}\int_0^{L_m} \sqrt{\beta(s)}\sin(\tau_{sj})\ ds$$

$$\sqrt{\beta_m}\cos(\tau_{mj}) \longrightarrow J_{C,mj} = \frac{1}{L_m}\int_0^{L_m} \sqrt{\beta(s)}\cos(\tau_{sj})\ ds$$

- **Observation: the accuracy of the two analytic approaches can be poor if thin-element model is used. Corrections to account for the variation of Twiss parameters across magnets have been included which reduce dramatically the errors w.r.t. the numerical version (see next slide).**

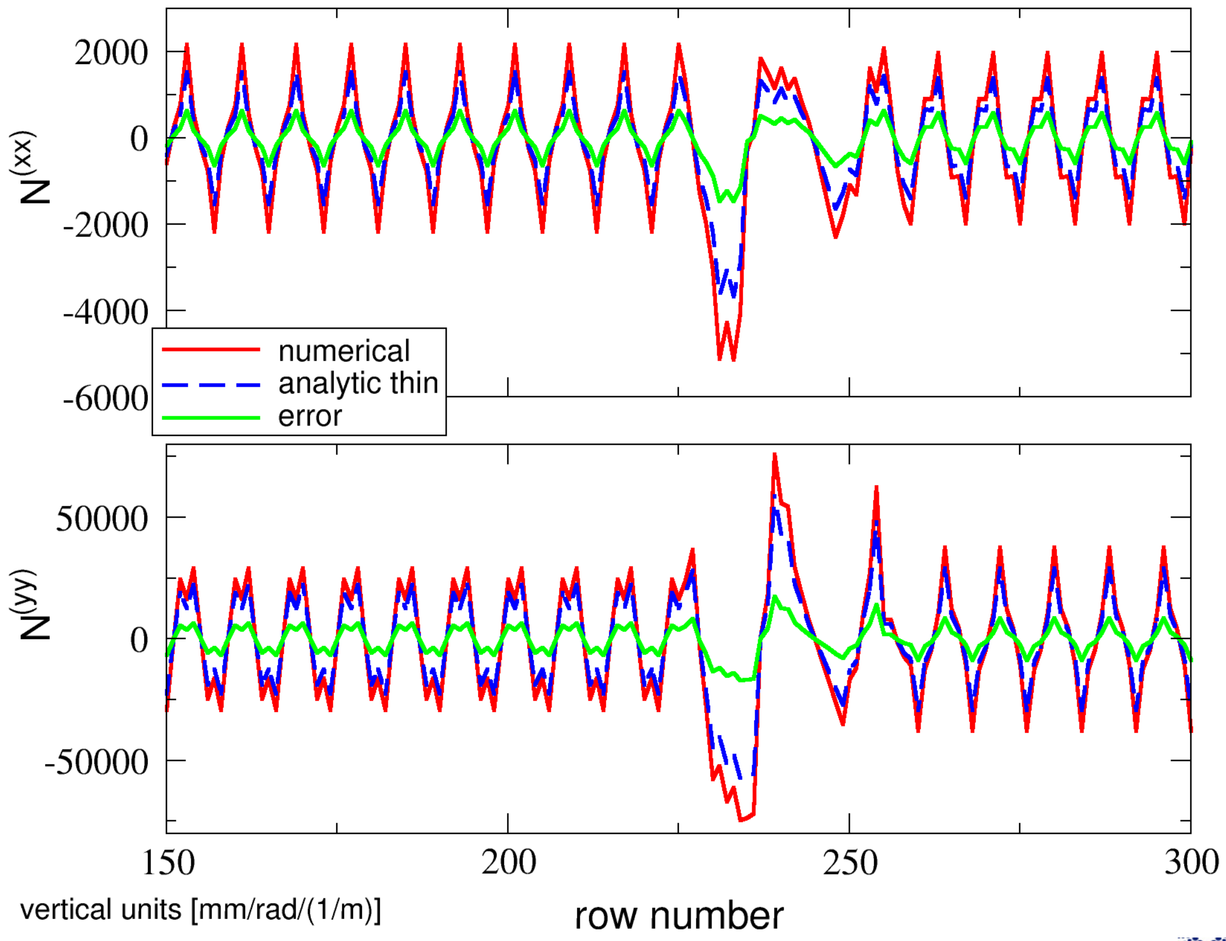
NUMERICAL VS ANALYTIC JACOBIAN OF THE ORM: ACCURACY

Example: FCC quadrupole ORM Jacobian with

- **1600 BPMs**
- **8 steerers**
- **1 quadrupole QC1L1_1**

$$\begin{pmatrix} \delta\vec{O}^{(xx)} \\ \delta\vec{O}^{(yy)} \\ \sout{\delta\vec{D}_x} \end{pmatrix} = \mathbf{N} \begin{pmatrix} \delta\vec{K}_1 \\ \sout{\delta\vec{K}_0} \end{pmatrix}$$

THIN



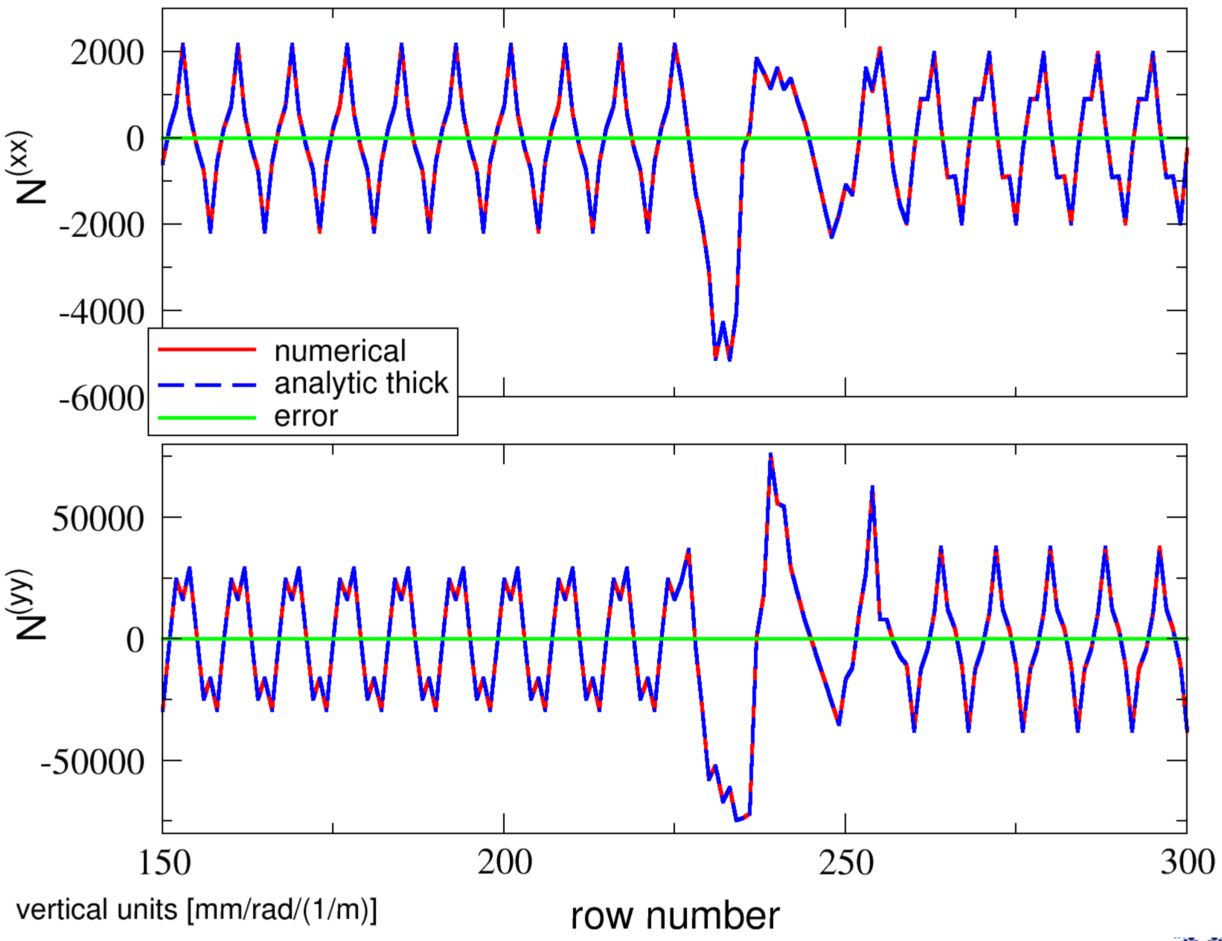
NUMERICAL VS ANALYTIC JACOBIAN OF THE ORM: ACCURACY

Example: FCC quadrupole ORM Jacobian with

- **1600 BPMs**
- **8 steerers**
- **1 quadrupole QC1L1_1**

$$\begin{pmatrix} \delta\vec{O}^{(xx)} \\ \delta\vec{O}^{(yy)} \\ \sout{\delta\vec{D}_x} \end{pmatrix} = \mathbf{N} \begin{pmatrix} \delta\vec{K}_1 \\ \sout{\delta\vec{K}_0} \end{pmatrix}$$

THICK



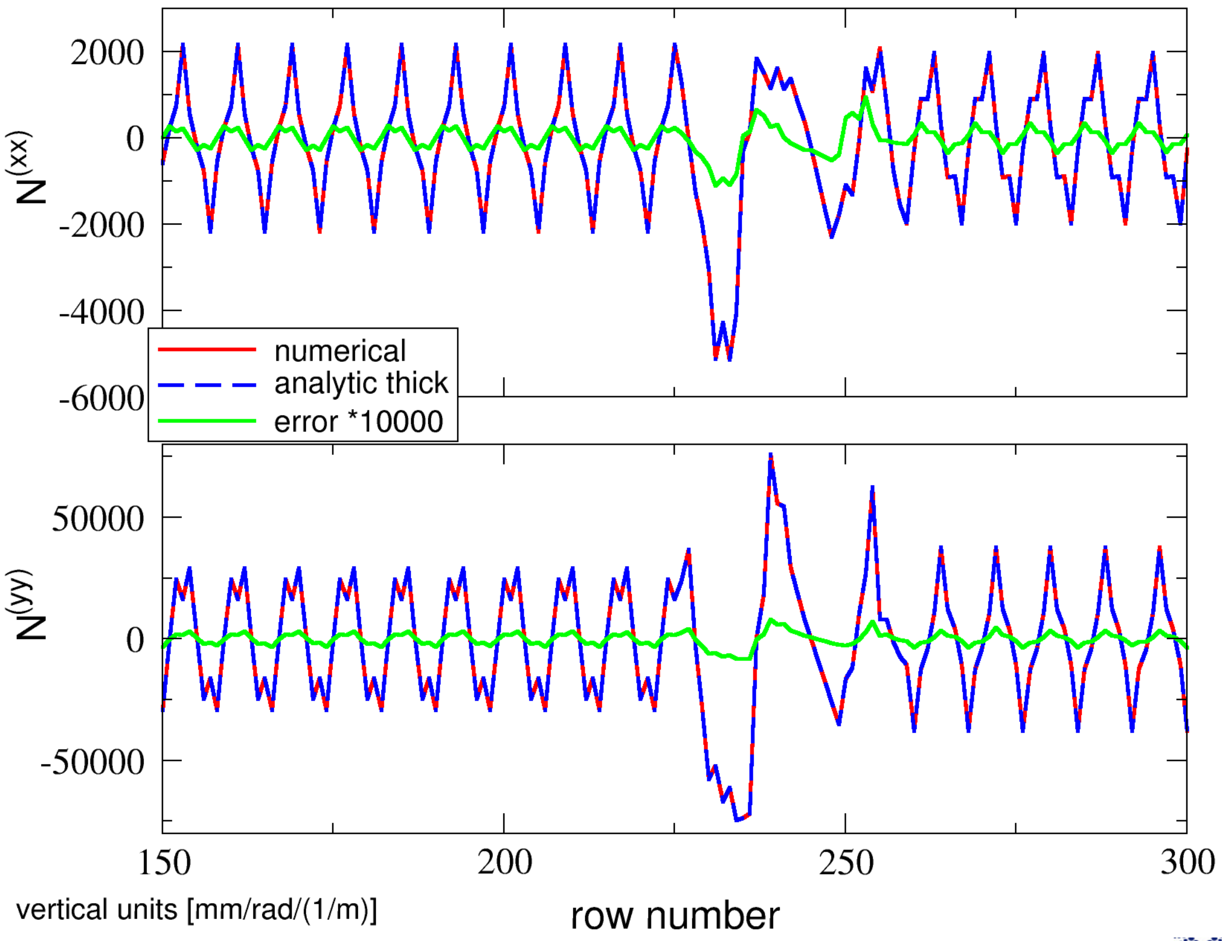
NUMERICAL VS ANALYTIC JACOBIAN OF THE ORM: ACCURACY

Example: FCC quadrupole ORM Jacobian with

- **1600 BPMs**
- **8 steerers**
- **1 quadrupole QC1L1_1**

$$\begin{pmatrix} \delta\vec{O}^{(xx)} \\ \delta\vec{O}^{(yy)} \\ \delta\vec{D}_x \end{pmatrix} = \mathbf{N} \begin{pmatrix} \delta\vec{K}_1 \\ \delta\vec{K}_0 \end{pmatrix}$$

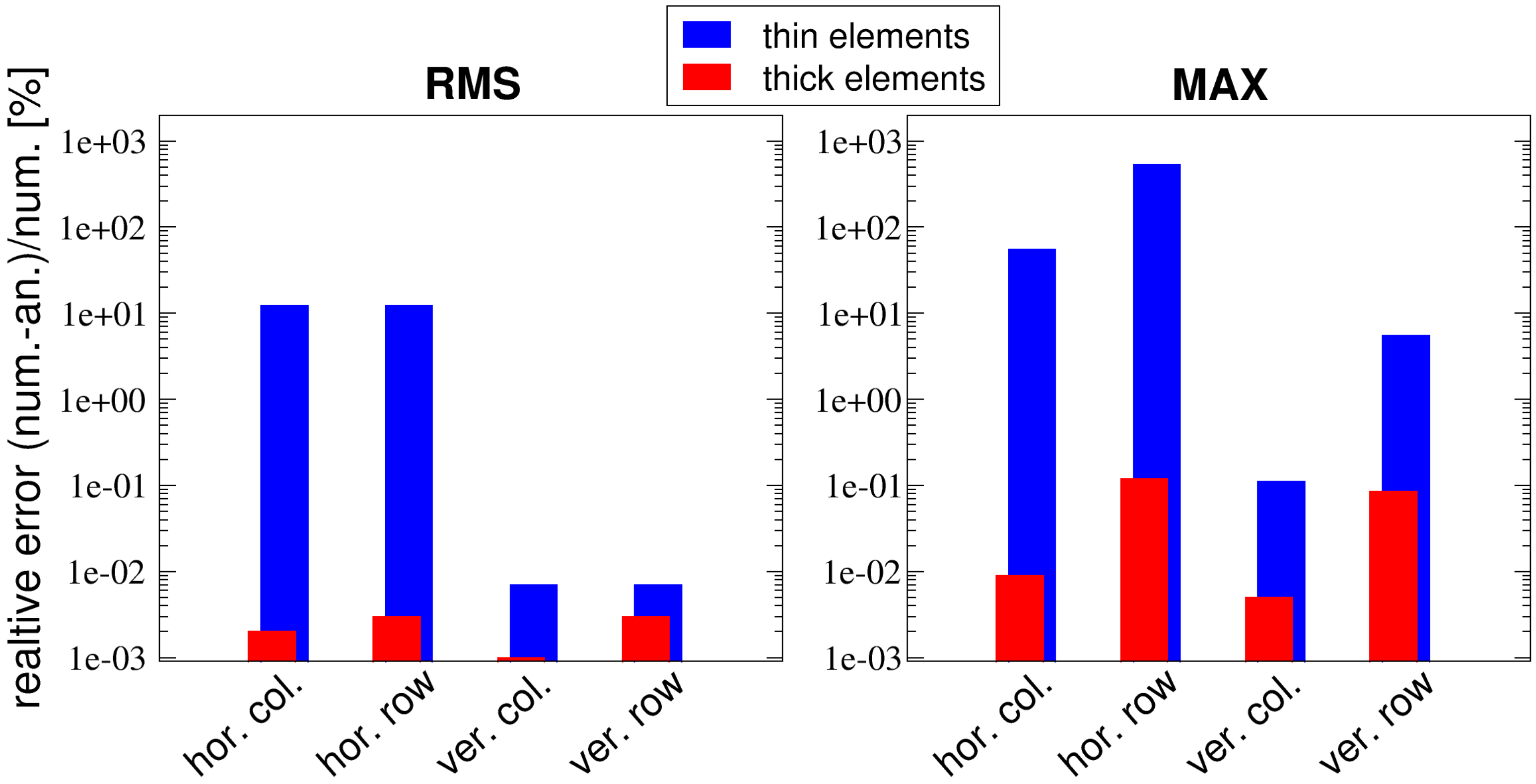
THICK



NUMERICAL VS ANALYTIC JACOBIAN OF THE ORM: ACCURACY

Example: FCC quadrupole ORM Jacobian with

- **1600 BPMs**
- **8 steerers**
- **1 quadrupole QC1L1_1**



$$\mathbf{ORM} = \begin{pmatrix} \mathbf{O^{(xx)}} & \mathbf{O^{(xy)}} \\ \mathbf{O^{(yx)}} & \mathbf{O^{(yy)}} \end{pmatrix}$$

- **RMS & MAX error computed over all columns & rows of the diagonal ORM blocks $O^{(xx)}$ & $O^{(yy)}$.**

NUMERICAL VS ANALYTIC JACOBIAN OF THE ORM: CPU TIME

Example: FCC quadrupole ORM Jacobian *N* (diagonal blocks only) with

- 1600 BPMs
- 8 steerers
- 360 quadrupoles parallelized over 64 cores CPUs (for both numerical and analytic tests)

Results

- Numeric: 1807.1 s [100%]
- fully analytic: 221.1 s [12%] (room for further optimization)

NUMERICAL VS ANALYTIC JACOBIAN OF THE ORM: OPTICS CORRECTION

Table 2: β-beating, dispersion and emittances after correction of 10 µm random alignment errors on dipole quadrupole and sextupole magnets for the EBS and FCC-ee lattices using analytic or numeric ORM derivative.

$\langle std \rangle_{50}$ units	$\frac{\Delta\beta_h}{\beta_{h,0}}$ %	$\frac{\Delta\beta_v}{\beta_{v,0}}$ %	$\Delta\eta_h$ mm	$\Delta\eta_v$ mm	$\Delta\epsilon_v$ pm rad
			EBS		
err.	19.37	11.08	17.33	6.91	94.17
ana.	0.2	0.2	0.18	0.05	0.003
num.	0.2	0.2	0.18	0.05	0.003
			FCC-ee Z		
err.	3.6	59.4	120.5	82.45	-
ana.	0.81	4.29	26.0	9.57	0.17
num.	0.82	4.30	25.98	9.64	0.18

From IPAC23 MOPL069

Table 3: β-beating, dispersion and emittances after correction of 10 µm random alignment errors on dipole quadrupole and sextupole magnets for the FCC-ee lattice using analytic ORM derivative (1856 BPMs, 18 steerers). The input lattice is tested: without radiation, with radiation and with radiation and tapering. Reference lattice is in all cases without radiation.

$\langle std \rangle_{50}$ units	$\frac{\Delta\beta_h}{\beta_{h,0}}$ %	$\frac{\Delta\beta_v}{\beta_{v,0}}$ %	$\Delta\eta_h$ mm	$\Delta\eta_v$ mm	$\Delta\epsilon_v$ pm rad
4D err	3.63	61.37	118.7	82.36	-
4D cor	0.84	4.24	25.67	9.58	0.71
6D err	3.60	59.45	120.54	82.45	-
6D cor	0.81	4.29	26.0	9.57	0.17
6D err + tapering	3.61	61.33	119.59	82.96	-
6D cor + tapering	0.82	4.22	26.03	9.65	0.18

ANALYTIC TUNE VARIATION WITH THICK QUADRUPOLES

EBS

Quad: ΔQF1J=0.023%

error ΔQH (num-ana): -0.16%

error ΔQv (num-ana): 0.14%

THIN

$$\frac{\Delta Q}{\Delta K_j} = \frac{\beta_j}{4\pi}$$

average β function across quadrupole

$$\frac{\Delta Q}{\Delta K_j} = \frac{\frac{1}{2}\left[\beta_j + \frac{\gamma_j}{K_j}\right] + \frac{\sin(2\sqrt{K_j}L_j)}{4\sqrt{K_j}L_j}\left[\beta_j - \frac{\gamma_j}{K_j}\right] + \frac{\alpha_j}{2K_jL_j}\left[\cos(2\sqrt{K_j}L_j) - 1\right]}{4\pi}$$

error ΔQH (num-ana): −0.0005 %

error ΔQv (num-ana): 0.0024 %

THICK

From IPAC23 MOPL069

(“Thick modelling” original idea and mathematical approach : Zeus Martí , ALBA)

ANALYTIC TUNE VARIATION WITH THICK QUADRUPOLES

FCC-ee

Quad: ΔQFG2-1 = 0.184 %

error ΔQH (num-ana): 2.2%

error ΔQv (num-ana): 0.38%

THIN

$$\frac{\Delta Q}{\Delta K_j} = \frac{\beta_j}{4\pi}$$

average β function across quadrupole

$$\frac{\Delta Q}{\Delta K_j} = \frac{\frac{1}{2}\left[\beta_j + \frac{\gamma_j}{K_j}\right] + \frac{\sin(2\sqrt{K_j}L_j)}{4\sqrt{K_j}L_j}\left[\beta_j - \frac{\gamma_j}{K_j}\right] + \frac{\alpha_j}{2K_jL_j}\left[\cos(2\sqrt{K_j}L_j) - 1\right]}{4\pi}$$

error ΔQH (num-ana): 0.0003 %
error ΔQv (num-ana): −0.0008 %

THICK

From IPAC23 MOPL069

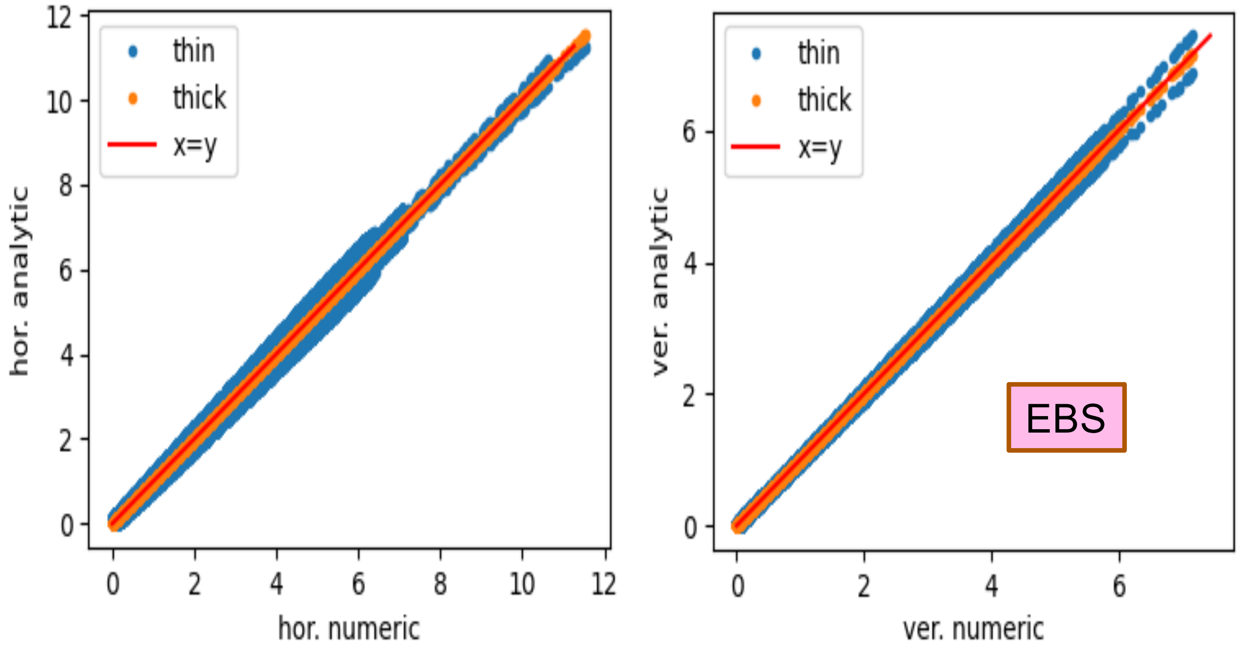
(“Thick modelling” original idea and mathematical approach : Zeus Martí , ALBA)

THICK STEERERS IN CLASSIC ANALYTIC ORM FORMULA

THIN steerers ORM

$$\vec{x} = M\vec{\theta} \quad ,$$

$$M_{ij} = \frac{\sqrt{\beta_j \beta_i}}{2 \sin \pi Q} \cos\left(|\phi_j - \phi_i| - \pi Q\right) + \frac{\eta_j \eta_i}{L_0 \alpha_c}$$



THICK steerers ORM

$$M_{i,j} = \frac{\sqrt{\beta_i}}{2 \sin \pi Q} \left[\left(\sqrt{\beta_j} - \frac{\alpha_j L_j}{2\sqrt{\beta_j}} \right) \cos\left(|\phi_j - \phi_i| - \pi Q\right) + \frac{L_j}{2\sqrt{\beta_j}} \sin\left(|\phi_j - \phi_i| - \pi Q\right) \right] + \frac{\eta_i \eta_j}{L_0 \alpha_c}$$

OUTLINE

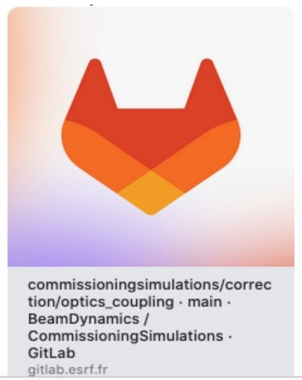
- Part 1: The implementation (12’)
- **Part 2: The repository (1’)**
- Part 3: outlook (3’)

WHERE YOU CAN FIND US

https://gitlab.esrf.fr/BeamDynamics/commissioningsimulations

Commissioning tools are still poor in terms of documentation, and debugging. The inclusion into the pyAT repository is pending such extensive validation tests.

Modules for the fully analytic ORM Jacobian can be found here:



https://gitlab.esrf.fr/BeamDynamics/commissioningsimulations/-/tree/main/commissioningsimulations/correction/optics_coupling

OUTLINE

- Part 1: The implementation (12')
- Part 2: The repository (1')
- **Part 3: outlook (3')**

OUTLOOK: THICK-ELEMENT CORRECTION FOR RDTS

$$f_{jklm}(s) \propto \sum_{m} \beta_{m,x}^{(J+k)/2} \beta_{m,y}^{(l+m)/2} \, e^{i[(j-k)\Delta\phi_{w,x}^{(s)} + (l-m)\Delta\phi_{w,y}^{(s)}]}$$

$$f_{1001,j} = \frac{\sum_{m=1}^{M} J_{m,1}\sqrt{\beta_{m,x}\beta_{m,y}} e^{i(\Delta\phi_{x,mj} - \Delta\phi_{y,mj})}}{4\left[1 - e^{2\pi i(Q_x - Q_y)}\right]} + O(J_1^2)$$

$$f_{1010,j} = \frac{\sum_{m=1}^{M} J_{m,1}\sqrt{\beta_{m,x}\beta_{m,y}} e^{i(\Delta\phi_{x,mj} + \Delta\phi_{y,mj})}}{4\left[1 - e^{2\pi i(Q_x + Q_y)}\right]} + O(J_1^2)$$

$$f_{2000,j} = -\frac{\sum_{m=1}^{M} \beta_{m,x}^{(mod)} \delta K_{m,1} e^{2i\Delta\phi_{x,mj}^{(mod)}}}{1 - e^{4\pi i Q_x^{(mod)}}} + O(\delta K_1^2)$$

$$f_{0020,j} = \frac{\sum_{m=1}^{M} \beta_{m,y}^{(mod)} \delta K_{m,1} e^{2i\Delta\phi_{y,mj}^{(mod)}}}{1 - e^{4\pi i Q_y^{(mod)}}} + O(\delta K_1^2)$$

$$\beta_m^{\mathbf{A}} \sin\left(\mathbf{B}\,\Delta\phi_{mj}\right) \longrightarrow \frac{1}{L_m}\int_0^{L_m} \beta(s)^{\mathbf{A}} \sin\left(\mathbf{B}\,\Delta\phi_s\right)\, ds$$

$$\beta_m^{\mathbf{A}} \cos\left(\mathbf{B}\,\Delta\phi_{mj}\right) \longrightarrow \frac{1}{L_m}\int_0^{L_m} \beta(s)^{\mathbf{A}} \cos\left(\mathbf{B}\,\Delta\phi_s\right)\, ds$$

Thick-element corrections are being implemented to RDTs

OUTLOOK: A NEW WAY TO COMPUTE LINEAR RDTS

From analytic formulas

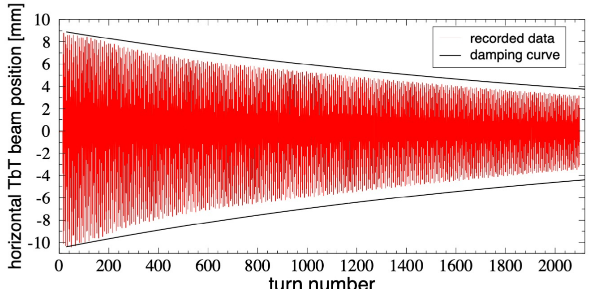
$$f_{2000,j} = -\frac{\sum_{m=1}^{M} \beta_{m,x}^{(mod)} \delta K_{m,1} e^{2i\Delta\phi_{x,mj}^{(mod)}}}{1 - e^{4\pi i Q_x^{(mod)}}} + O(\delta K_1^2)$$

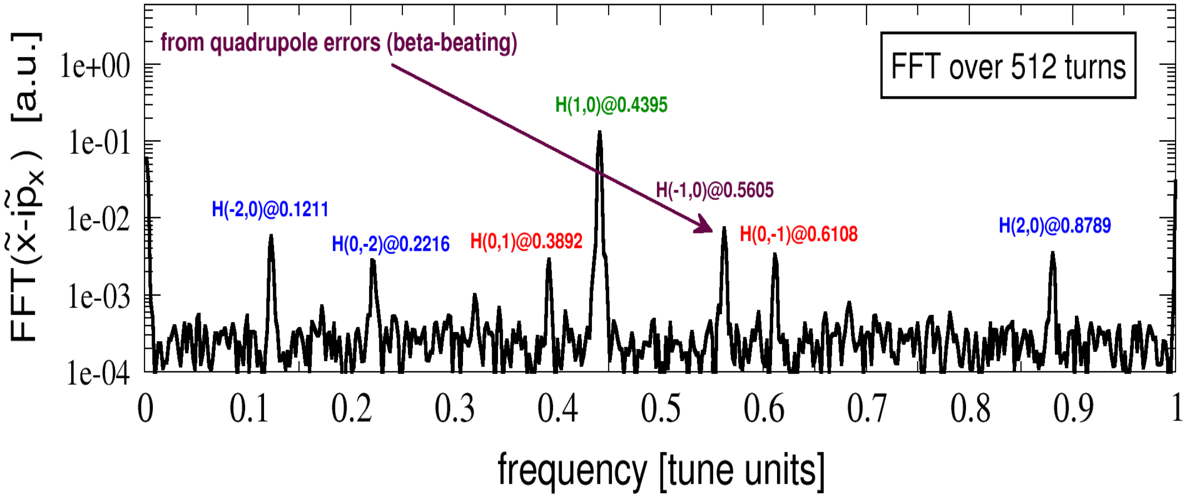
$$f_{0020,j} = \frac{\sum_{m=1}^{M} \beta_{m,y}^{(mod)} \delta K_{m,1} e^{2i\Delta\phi_{y,mj}^{(mod)}}}{1 - e^{4\pi i Q_y^{(mod)}}} + O(\delta K_1^2)$$

New formulas from OTM

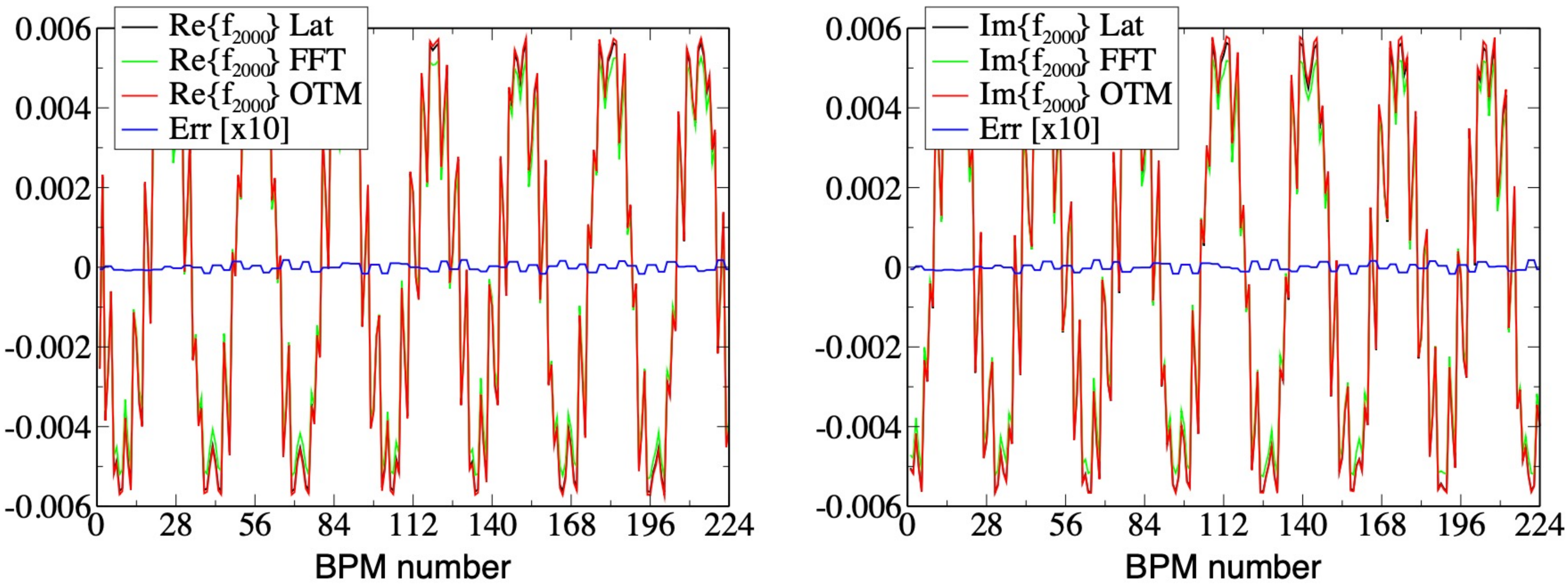
$$\vec{X}^{(N+1)} = \mathbf{M}\vec{X}^{(N)} , \qquad \vec{X} = \begin{pmatrix} x \\ p_x \\ y \\ p_y \end{pmatrix}$$

From particle tracking + harmonic analysis



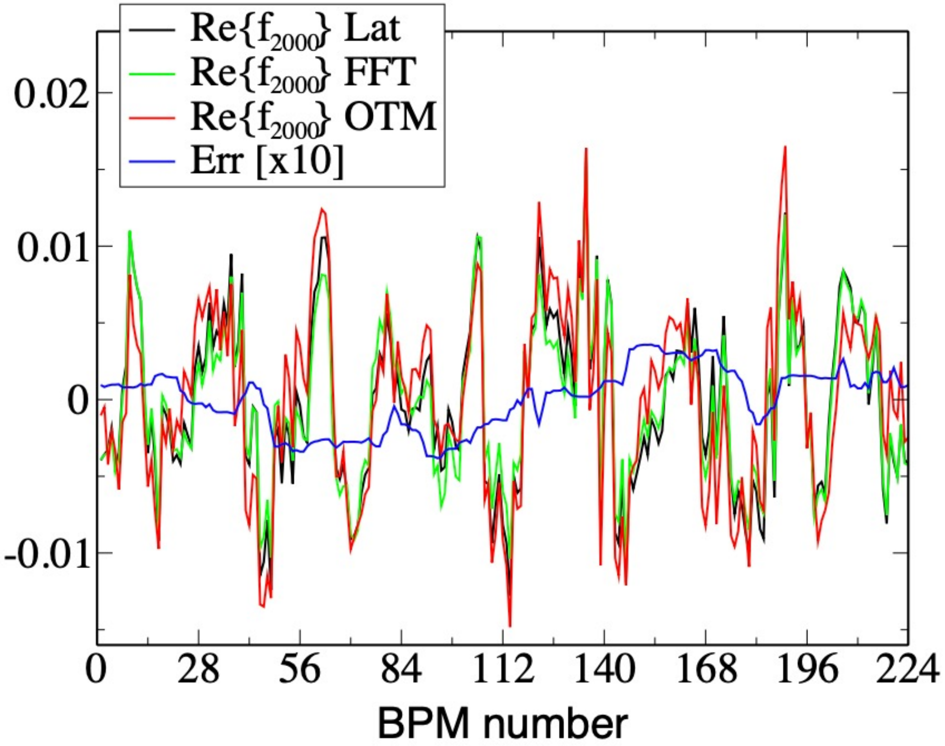


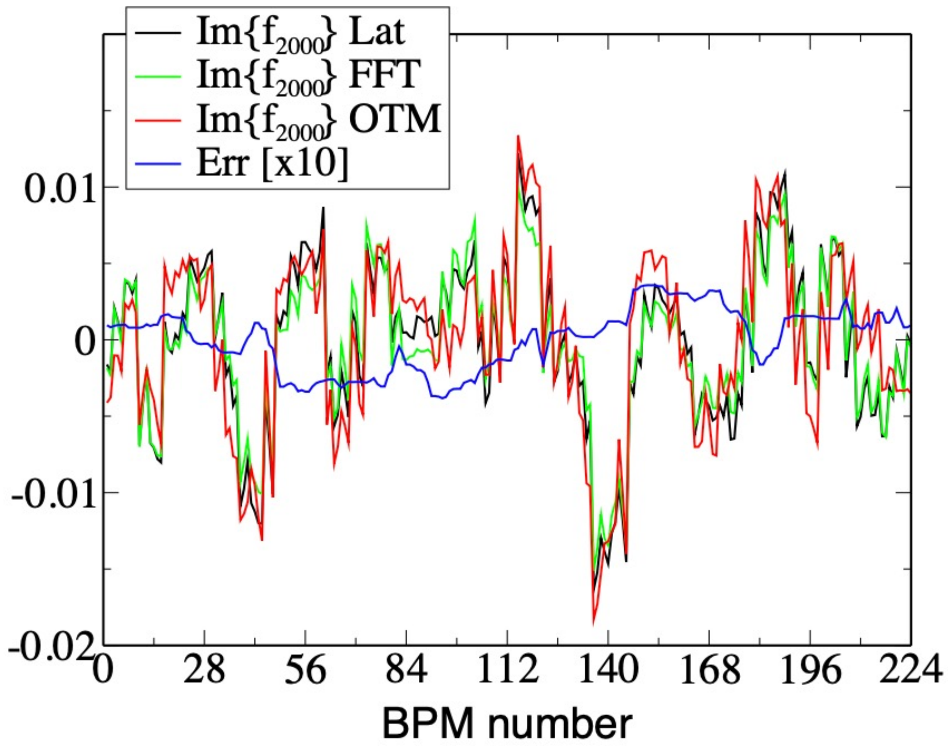
OUTLOOK: A NEW WAY TO COMPUTE LINEAR RDTS



Very accurate for single-quad error (thus ok for response matrix)

OUTLOOK: A NEW WAY TO COMPUTE LINEAR RDTS





Less accurate for distributed errors (2nd order & coupling terms)

OUTLOOK: A NEW WAY TO COMPUTE LINEAR RDTS

New formulas for coupling RDTs from OTM are still to be derived

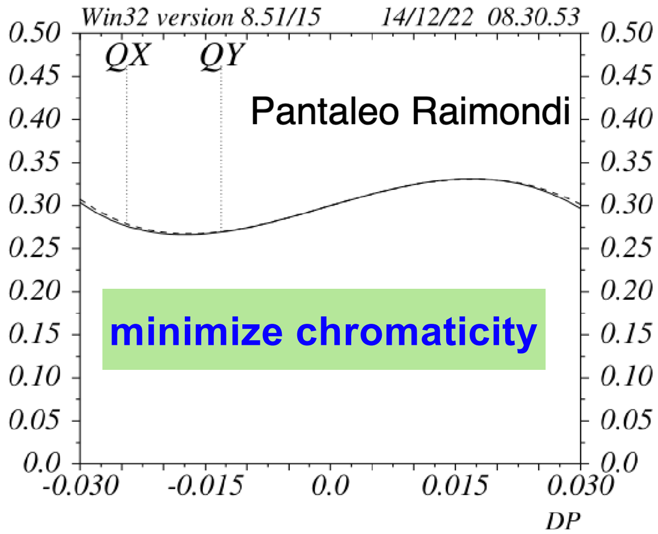
$$\vec{X}^{(N+1)} = \mathbf{M}\vec{X}^{(N)} \,, \qquad \vec{X} = \begin{pmatrix} x \\ p_x \\ y \\ p_y \end{pmatrix}$$

OUTLINE

- Part 1: The implementation (12’)
- Part 2: The repository (1’)
- Part 3: outlook (3’)

Thank you!

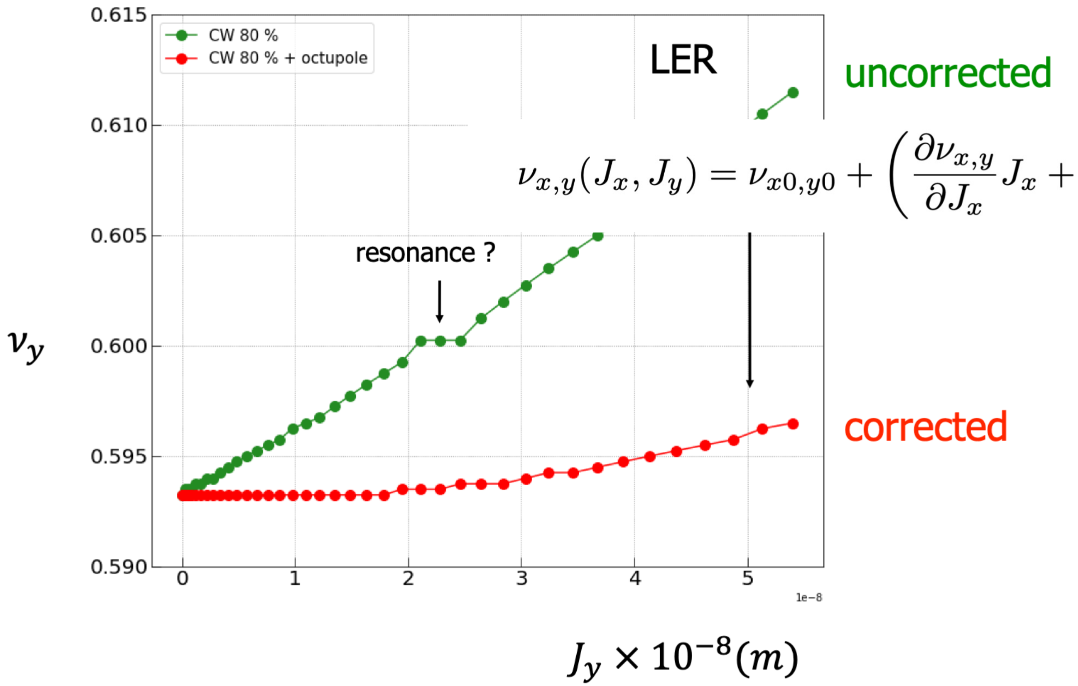
SOME THOUGHTS ON NON-LINEAR OPTIMIZATION



minimize chromaticity

Y. Ohnishi / KEK

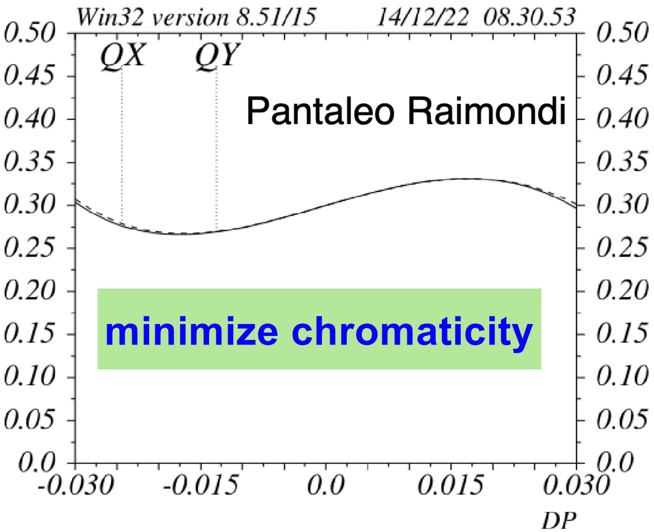
The ARC+LSS has zero second order chromaticity in both planes, only third and higher orders remain.



$$\nu_{x,y}(J_x, J_y) = \nu_{x0,y0} + \left(\frac{\partial \nu_{x,y}}{\partial J_x} J_x + \frac{\partial \nu_{x,y}}{\partial J_y} J_y \right) + \cdots$$

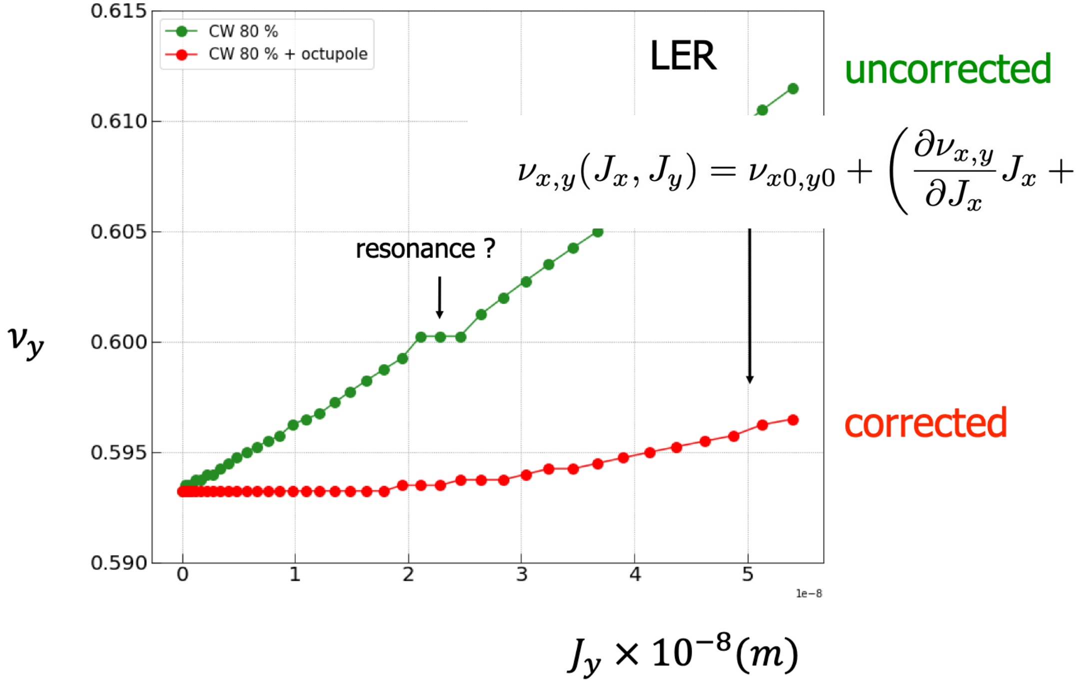
minimize detuning with amplitude

SOME THOUGHTS ON NON-LINEAR OPTIMIZATION



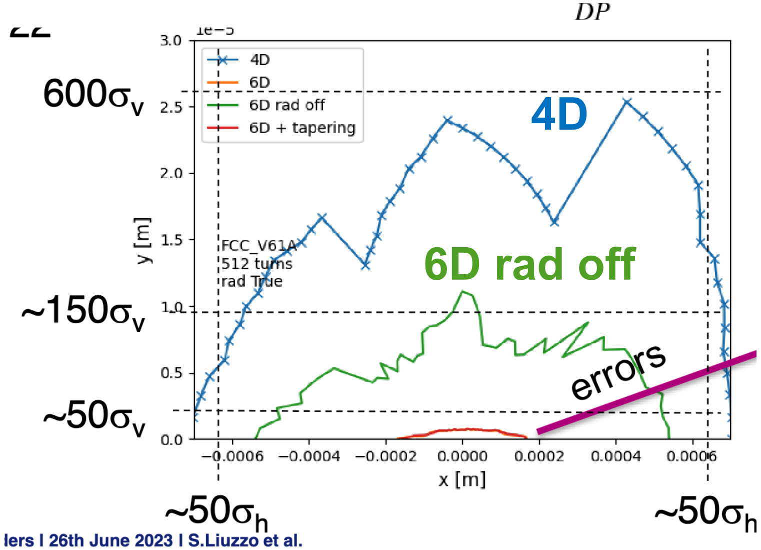
The ARC+LSS has zero second order chromaticity in both planes, only third and higher orders remain.

Y. Ohnishi / KEK



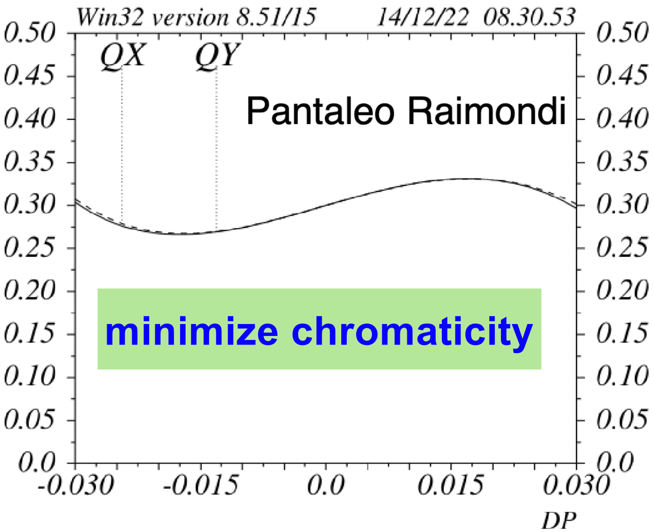
$$\nu_{x,y}(J_x, J_y) = \nu_{x0,y0} + \left(\frac{\partial \nu_{x,y}}{\partial J_x} J_x + \frac{\partial \nu_{x,y}}{\partial J_y} J_y\right) + \cdots$$

minimize detuning with amplitude



Simone Liuzzo

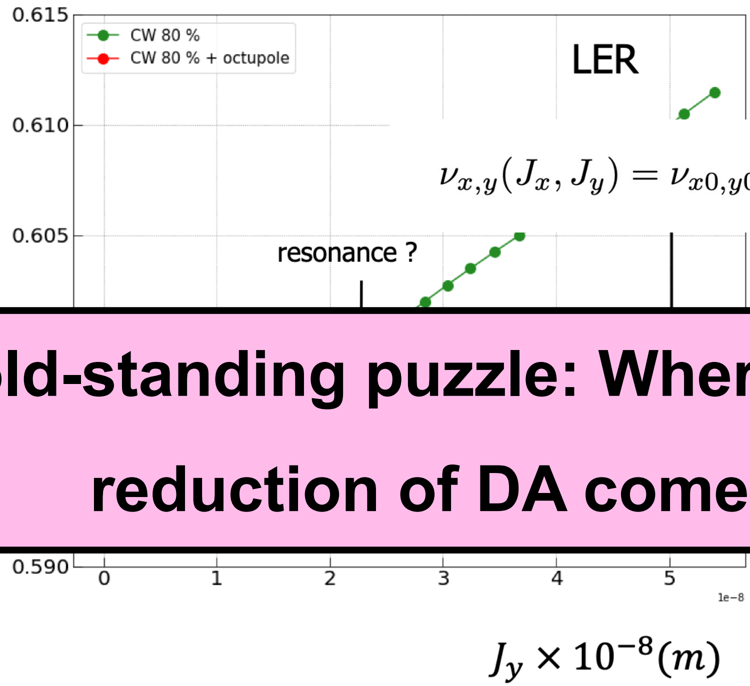
SOME THOUGHTS ON NON-LINEAR OPTIMIZATION



minimize chromaticity

The ARC+LSS has zero second order chromaticity in both planes, only third and higher orders remain.

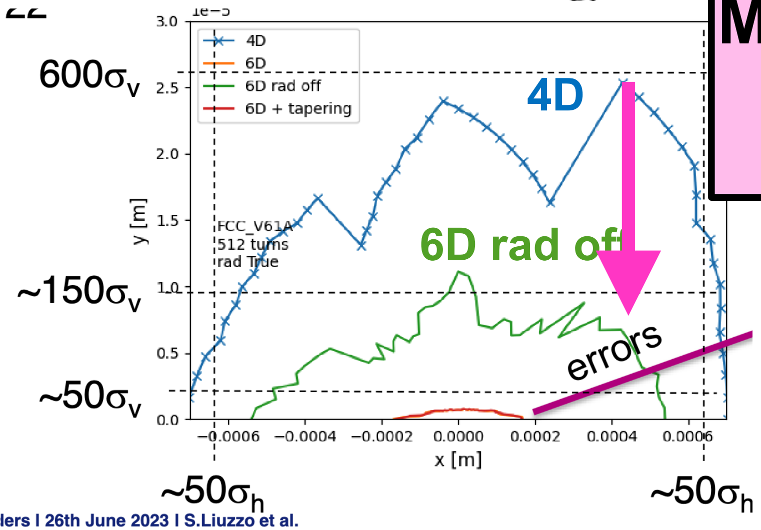
Y. Ohnishi / KEK



$$\nu_{x,y}(J_x, J_y) = \nu_{x0,y0} + \left(\frac{\partial \nu_{x,y}}{\partial J_x} J_x + \frac{\partial \nu_{x,y}}{\partial J_y} J_y\right) + \cdots$$

minimize detuning with amplitude

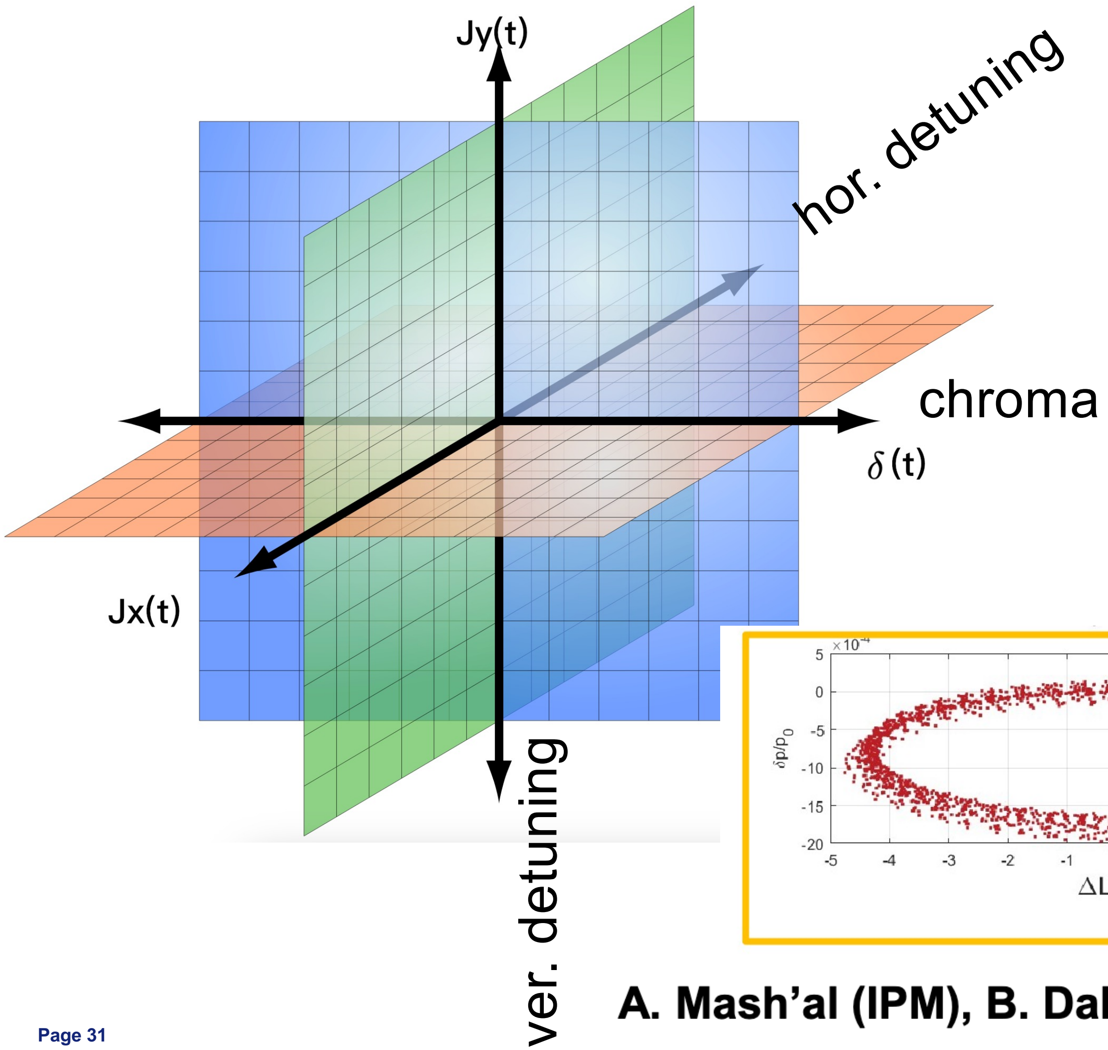
My old-standing puzzle: Where does all this reduction of DA come from?

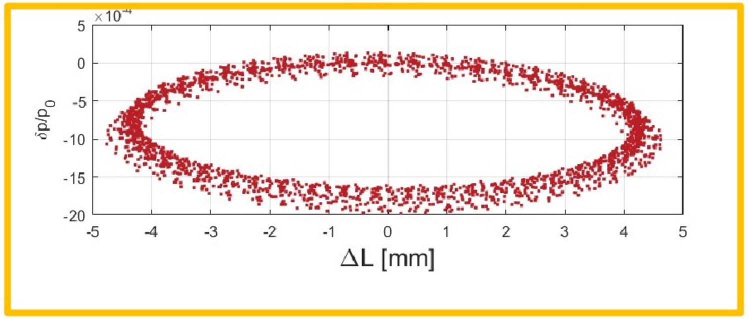


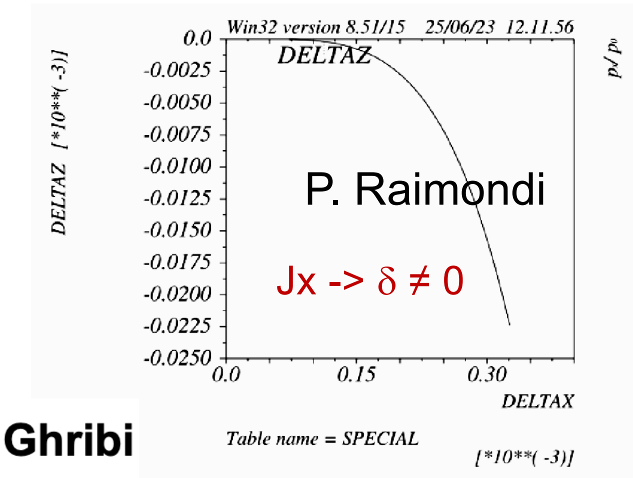
Simone Liuzzo

SOME THOUGHTS ON NON-LINEAR OPTIMIZATION

- **My guess: Blame path lengthening and synchrotron motion for that.**

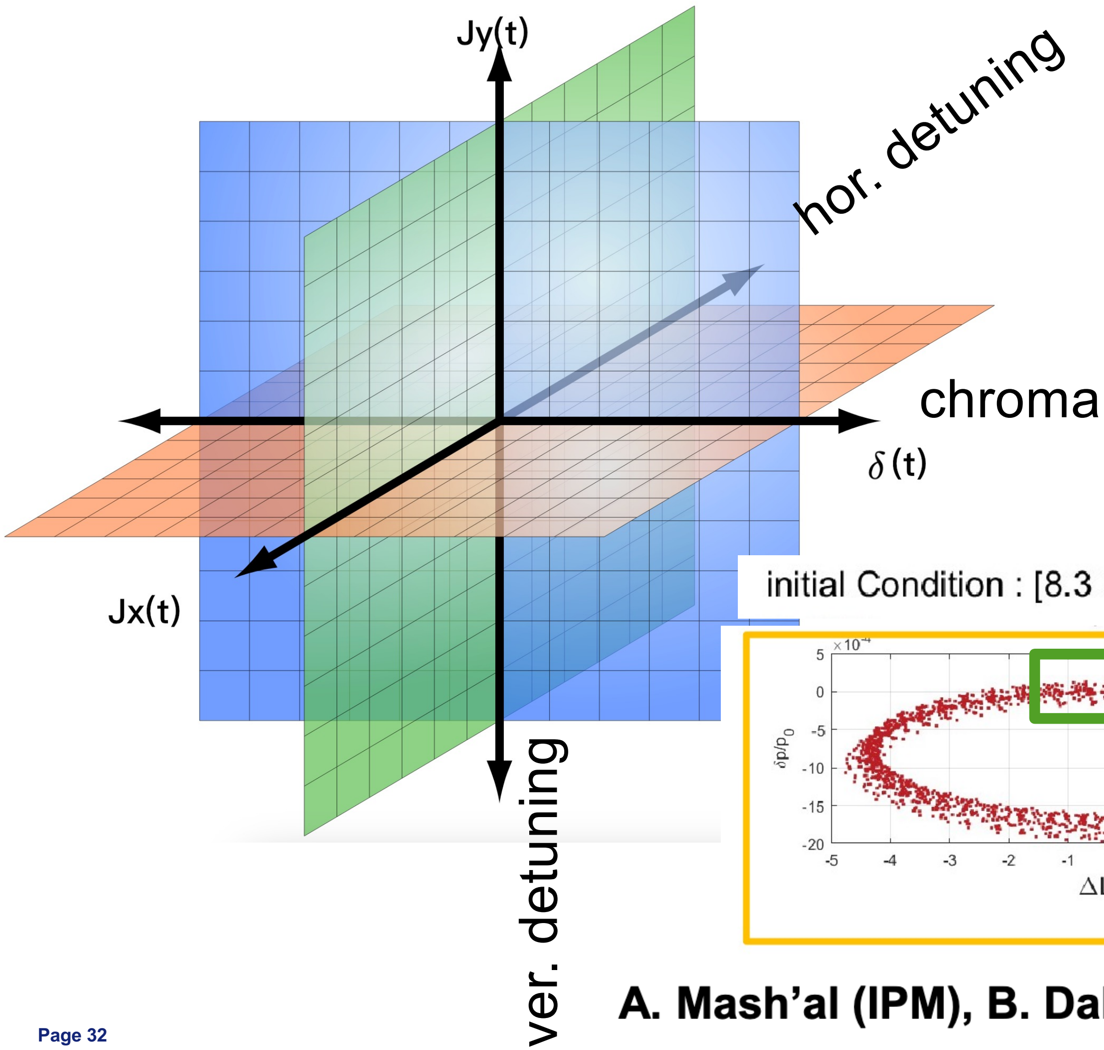




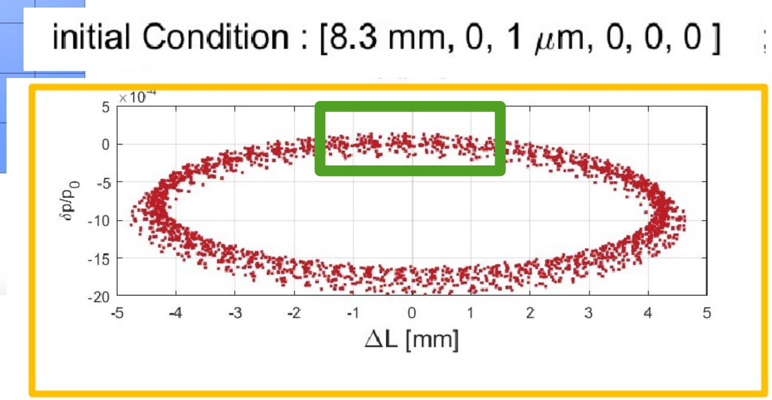


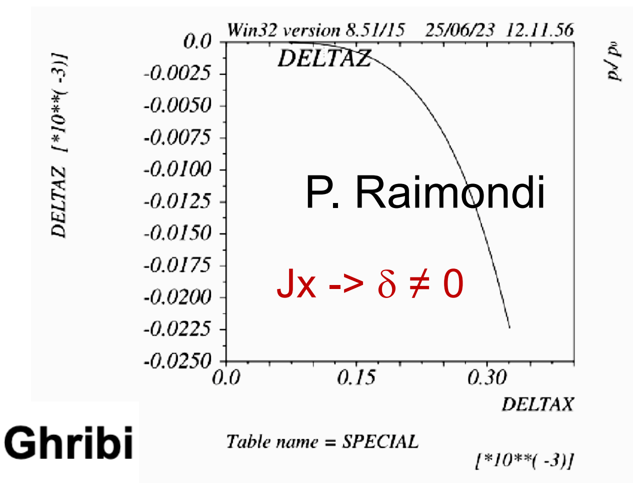
A. Mash'al (IPM), B. Dalena, A. Chance, A. Ghribi

SOME THOUGHTS ON NON-LINEAR OPTIMIZATION



- **My guess: Blame path lengthening and synchrotron motion for that.**
- **Most of the time they will be off-axis ($J_x \neq 0$) and off-energy ($\delta \neq 0$), thus, outside the 3 axis of the tune space.**





A. Mash'al (IPM), B. Dalena, A. Chance, A. Ghribi

SOME THOUGHTS ON NON-LINEAR OPTIMIZATION

Consequence: Minimizing cross-term (betatron+chromatic) detuning terms could be as much, if not more, effective than minimizing purely higher-order betatron or chromatic terms

$$Q\big(J_x(t),\delta(t)\big) = Q(J_x(0),\delta(0)) \; +$$

| ANHX, ANHY |

$$\frac{\partial Q}{\partial J_x} J_x(t) \; + \; \frac{\partial Q}{\partial \delta}\delta(t) \; +$$

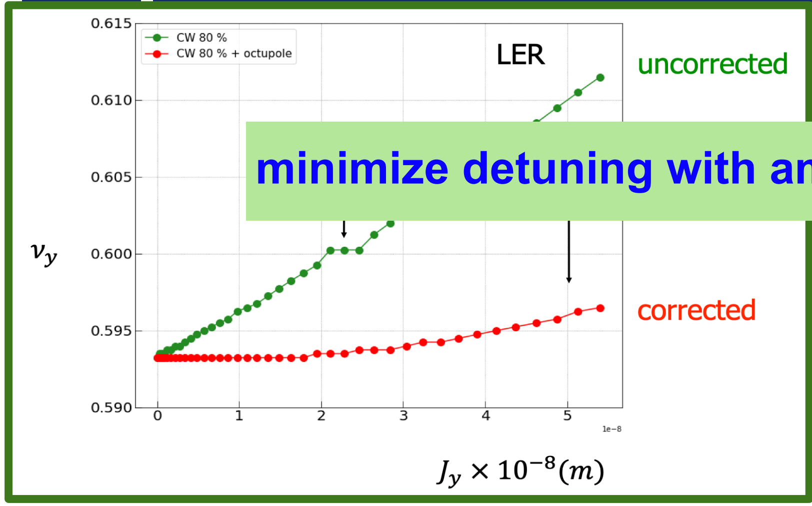
| $n(\varepsilon_1), n(\varepsilon_2), n(\delta_p)$ |

$$\frac{1}{2}\left\{\frac{\partial^2 Q}{\partial J_x^2} J_x^2(t) \; + \; \frac{\partial^2 Q}{\partial \delta^2}\delta^2(t) \; + \; \frac{\partial^2 Q}{\partial J_x \partial \delta} J_x(t)\delta(t)\right\} +$$

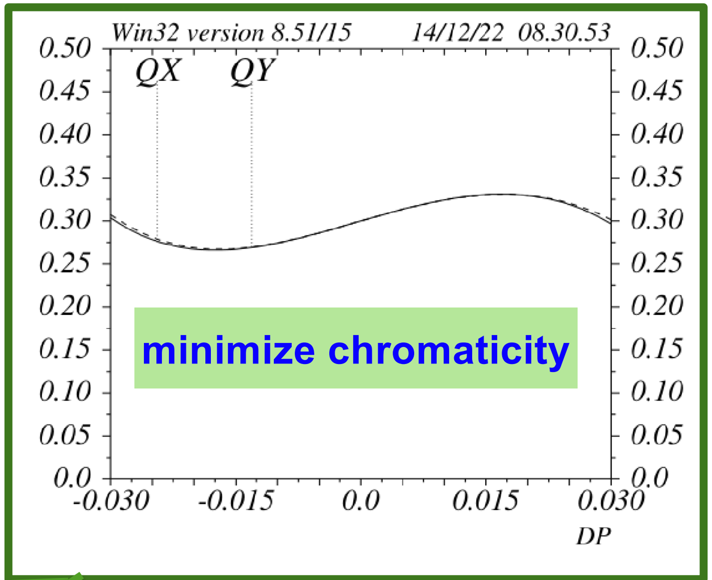
PTC_NORMAL,

....

SOME THOUGHTS ON NON-LINEAR OPTIMIZATION



minimize detuning with amplitude



minimize chromaticity

$$Q\big(J_x(t),\delta(t)\big) = Q\big(J_x(0),\delta(0)\big) + \frac{\partial Q}{\partial J_x} J_x(t) + \frac{\partial Q}{\partial \delta}\delta(t) + \frac{1}{2}\left\{\frac{\partial^2 Q}{\partial J_x^2} J_x^2(t) + \frac{\partial^2 Q}{\partial \delta^2}\delta^2(t) + \frac{\partial^2 Q}{\partial J_x \partial \delta} J_x(t)\delta(t)\right\} + \ldots.$$

Maybe we shall also minimize the cross term along with higher-order terms

SOME THOUGHTS ON NON-LINEAR OPTIMIZATION

Blame path lengthening: Actually, minimizing linear chroma is still very helpful, since path lengthening and chroma are highly correlated

$$\Delta C = -2\pi(J_X\xi_X + J_Y\xi_Y). \qquad (31)$$

$$Q\left(J_x(t), \delta(t)\right) = Q(J_x(0), \delta(0)) + \frac{\partial Q}{\partial J_x} J_x(t) + \frac{\partial Q}{\partial \delta}\delta(t) +$$

PHYSICAL REVIEW SPECIAL TOPICS - ACCELERATORS AND BEAMS **8**, 094001 (2005)

Dependence of average path length betatron motion in a storage ring

Yoshihiko Shoji*