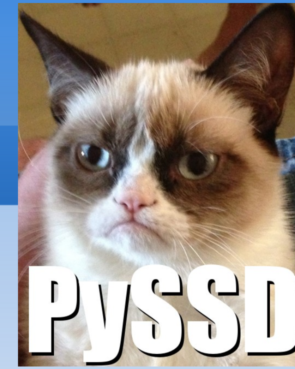
PySSD and BimBim

X. Buffat

- Python Solver for Stability Diagram → Landau damping
  - Physics
  - Examples
  - Implementation
  - Needs and future plans
- Beam-Beam and Impedance → Coherent mode analysis
  - Physics
  - Examples
  - Implementation
  - Needs and future plans

ABP-CWG 27.04.2017

Stability diagram



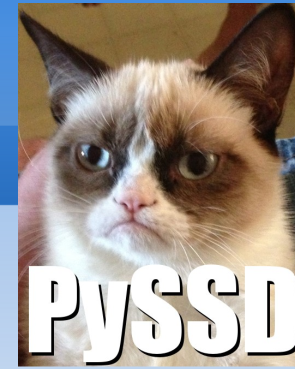
The stability of a coherent mode driven by the impedance through Landau damping is estimated through the following dispersion integral :

J.Scott Berg, F. Ruggiero, Landau damping with two-dimensional betatron tune spread, CERN-SL-96-071-AP

Valid for head-tail modes that are :

- Uncoupled
- weakly perturbed

Stability diagram



The stability of a coherent mode driven by the impedance through Landau damping is estimated through the following dispersion integral :

J.Scott Berg, F. Ruggiero, Landau damping with two-dimensional betatron tune spread, CERN-SL-96-071-AP

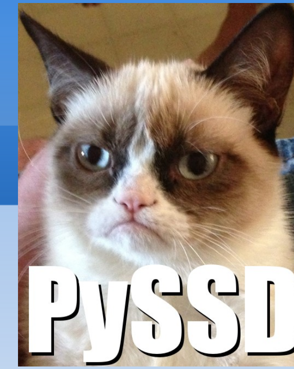
→ Valid for head-tail modes that are :

- Uncoupled
- weakly perturbed

$$\frac{-1}{\Delta Q_w} = \iint_0^{\infty} \frac{J_w \frac{d\Psi_w(J_x, J_y)}{dJ_w}}{Q - q_w(J_x, J_y) - i\epsilon} dJ_x dJ_y, w = x, y$$

- Realistic configurations (lattice imperfections, coupling, non-linearities, beam-beam interactions) lead to complex expressions of the amplitude detuning

Stability diagram



The stability of a coherent mode driven by the impedance through Landau damping is estimated through the following dispersion integral :

J.Scott Berg, F. Ruggiero, Landau damping with two-dimensional betatron tune spread, CERN-SL-96-071-AP

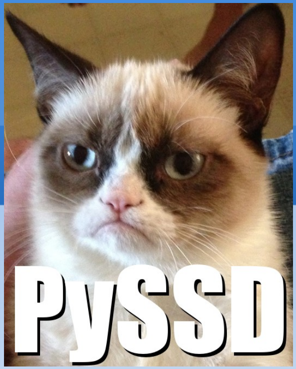
→ Valid for head-tail modes that are :

- Uncoupled
- weakly perturbed

$$\frac{-1}{\Delta Q_w} = \iint_0^{\infty} \frac{J_w \frac{d\Psi_w(J_x, J_y)}{dJ_w}}{Q - q_w(J_x, J_y) - i\epsilon} dJ_x dJ_y, \ w = x, y$$

- Realistic configurations (lattice imperfections, coupling, non-linearities, beam-beam interactions) lead to complex expressions of the amplitude detuning
- Different mechanisms may affect the distributions in a non-trivial way (non-linearities, noise, collimations)

Stability diagram



The stability of a coherent mode driven by the impedance through Landau damping is estimated through the following dispersion integral :

J.Scott Berg, F. Ruggiero, Landau damping with two-dimensional betatron tune spread, CERN-SL-96-071-AP

→ Valid for head-tail modes that are :

- Uncoupled
- weakly perturbed

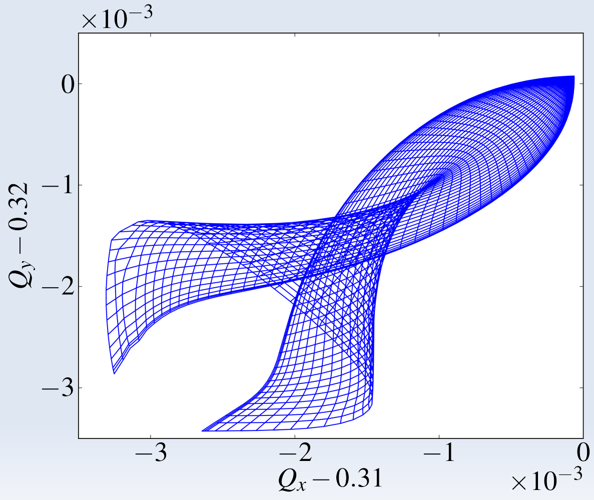
$$\frac{-1}{\Delta Q_w} = \iint_0^\infty \frac{J_w \frac{d\Psi_w(J_x,J_y)}{dJ_w}}{Q - q_w(J_x,J_y) - i\epsilon} \, dJ_x \, dJ_y , \quad w = x, y$$

- Realistic configurations (lattice imperfections, coupling, non-linearities, beam-beam interactions) lead to complex expressions of the amplitude detuning
- Different mechanisms may affect the distributions in a non-trivial way (non-linearities, noise, collimations)

→ Solve the dispersion integral numerically

(W. Herr, L. Vos, Tune distributions and effective tune spread from beam-beam interactions and the consequences for Landau damping in the LHC, LHC-PROJECT-NOTE-316)

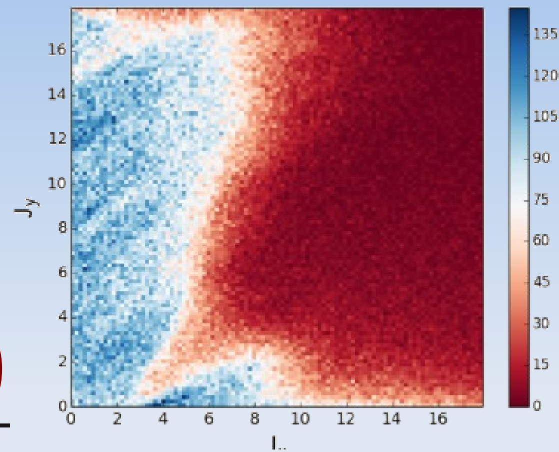
Stability diagram



Amplitude detuning from an analytical formula or from tracking with MAD-X (X. Buffat, PhD thesis, 2015)

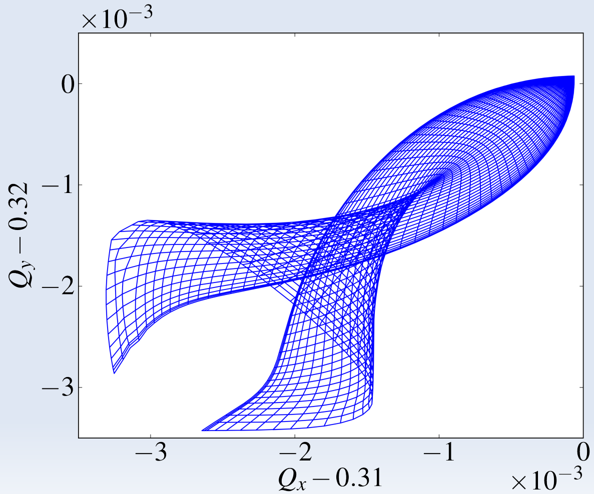
Stability diagram

Transverse distribution from an anlytical formula or from tracking with SixTrack (C. Tambaso, PhD thesis, in prep.)



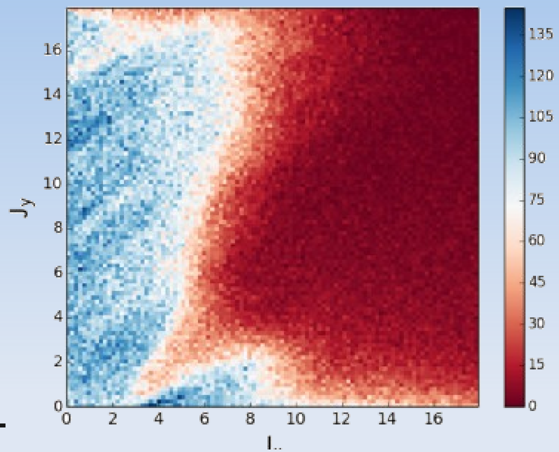
$$\frac{-1}{\Delta Q_w} = \iint_0^\infty \frac{J_w \frac{d\Psi_w(J_x, J_y)}{dJ_w}}{Q - q_w(J_x, J_y) - i\epsilon} dJ_x dJ_y, w = x, y$$

Amplitude detuning from an analytical formula or from tracking with MAD-X (X. Buffat, PhD thesis, 2015)

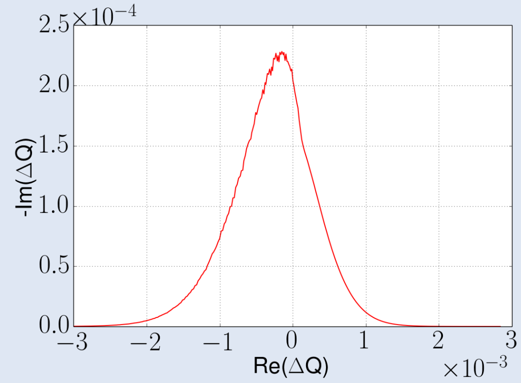


Stability diagram

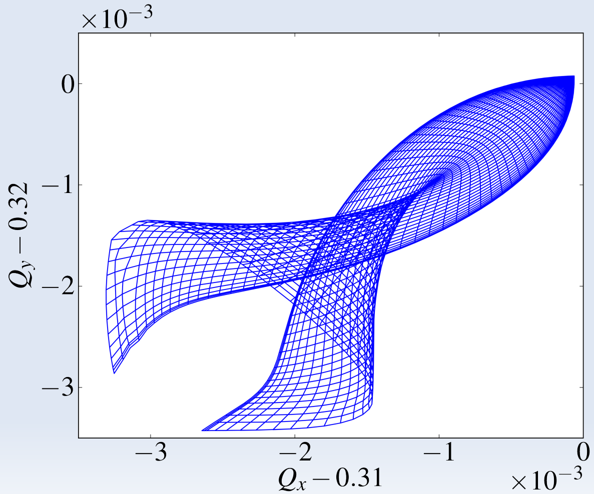
Transverse distribution from an anlytical formula or from tracking with SixTrack (C. Tambaso, PhD thesis, in prep.)



$$\frac{-1}{\Delta Q_w} = \iint_0^\infty \frac{J_w \frac{d\Psi_w(J_x,J_y)}{dJ_w}}{Q - q_w(J_x,J_y) - i\epsilon} dJ_x dJ_y, w = x, y$$

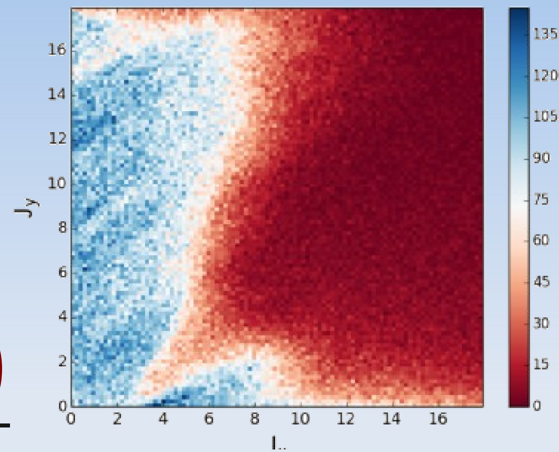


Amplitude detuning from an analytical formula or from tracking with MAD-X (X. Buffat, PhD thesis, 2015)

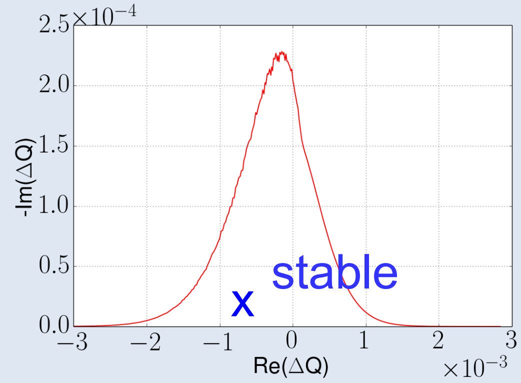


Stability diagram

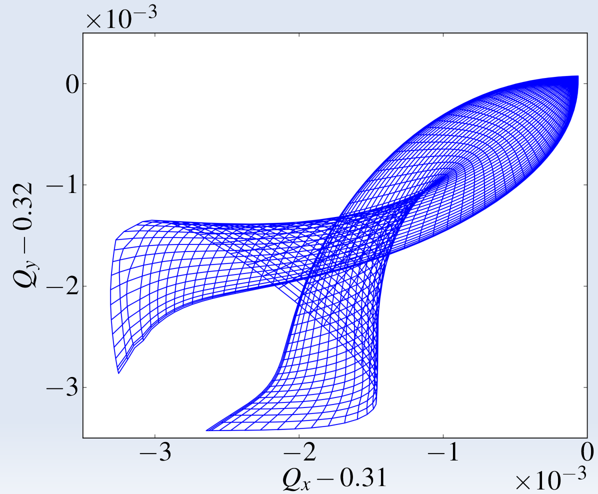
Transverse distribution from an anlytical formula or from tracking with SixTrack (C. Tambaso, PhD thesis, in prep.)



$$\frac{-1}{\Delta Q_w} = \iint_0^\infty \frac{J_w \frac{d\Psi_w(J_x,J_y)}{dJ_w}}{Q - q_w(J_x,J_y) - i\epsilon}\, dJ_x\, dJ_y, \quad w = x, y$$

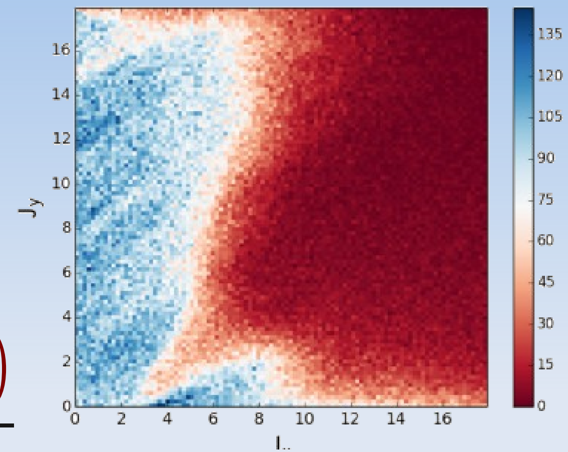


Amplitude detuning from an analytical formula or from tracking with MAD-X (X. Buffat, PhD thesis, 2015)

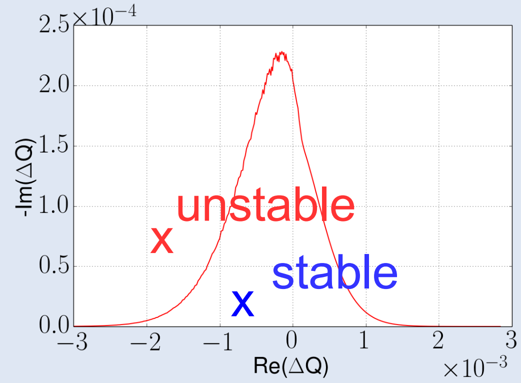


Stability diagram

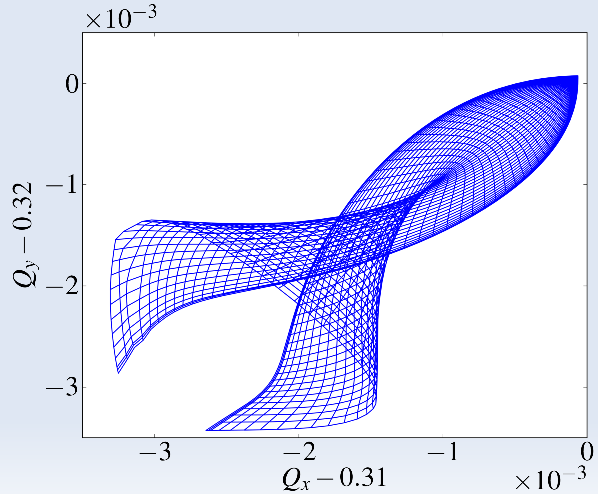
Transverse distribution from an anlytical formula or from tracking with SixTrack (C. Tambaso, PhD thesis, in prep.)



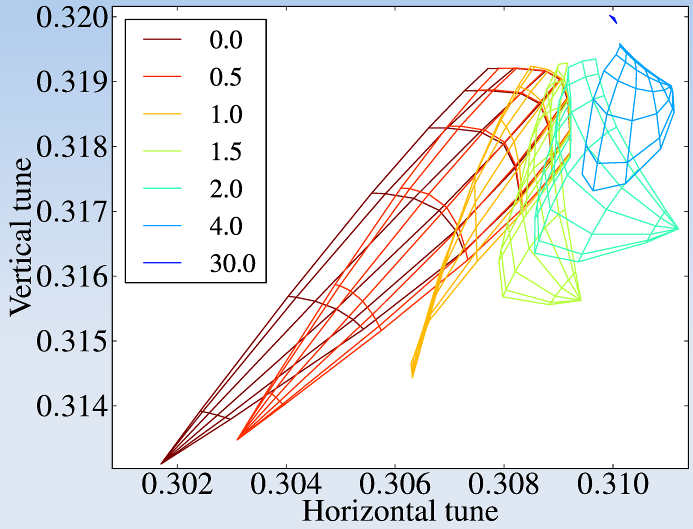
$$\frac{-1}{\Delta Q_w} = \iint_0^{\infty} \frac{J_w \frac{d\Psi_w(J_x, J_y)}{dJ_w}}{Q - q_w(J_x, J_y) - i\epsilon} dJ_x dJ_y, w = x, y$$

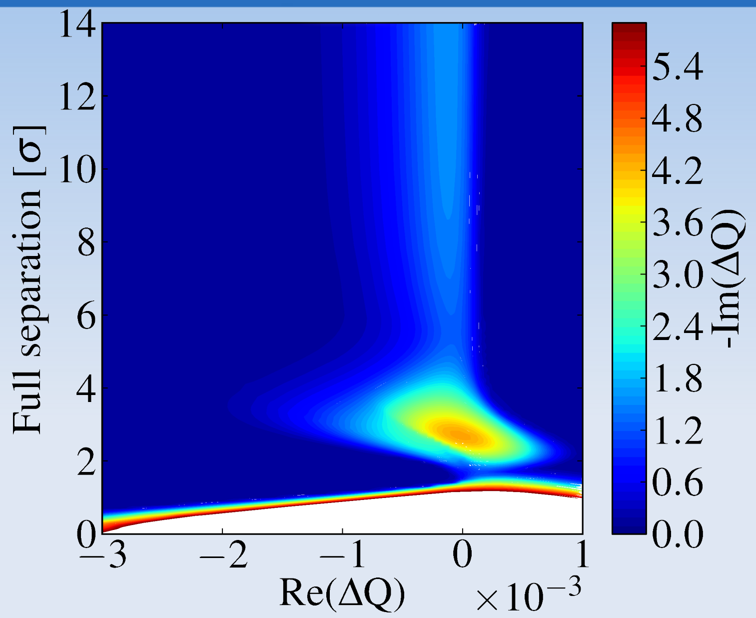


Amplitude detuning from an analytical formula or from tracking with MAD-X (X. Buffat, PhD thesis, 2015)



Beam-beam interactions



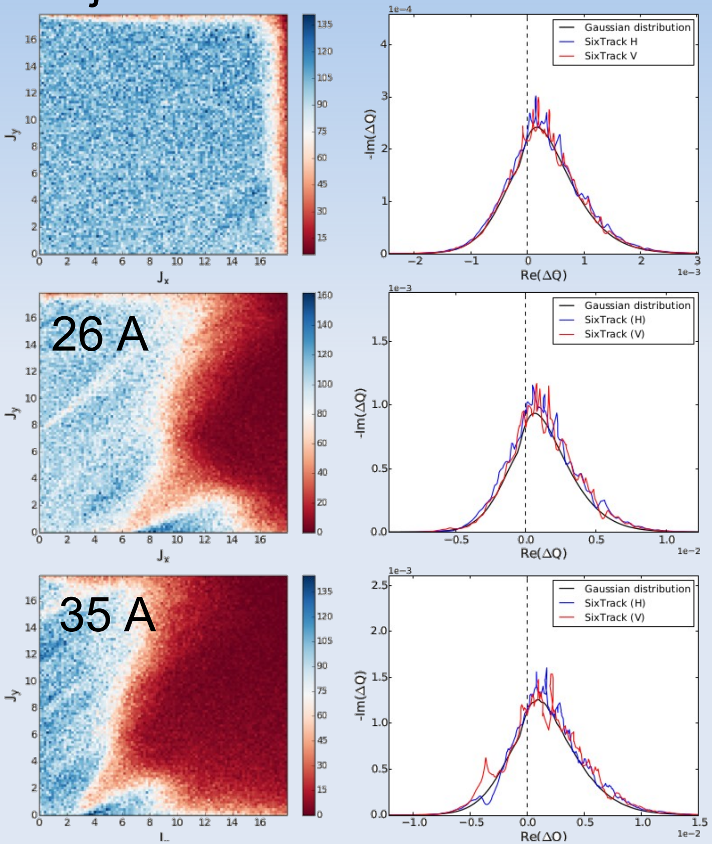


- The behaviour of the amplitude detuning when the beams collide with an offset is non-trivial and leads to important modification of the Landau damping

  → Possible loss of Landau damping when bringing the beams into collision or when leveling the luminosity with a transverse offset (X. Buffat et al., Stability diagram of colliding beams in the LHC, Phys. Rev. ST Accel. Beams 17, 111002, Nov 2014)

Non-Gaussian distributions

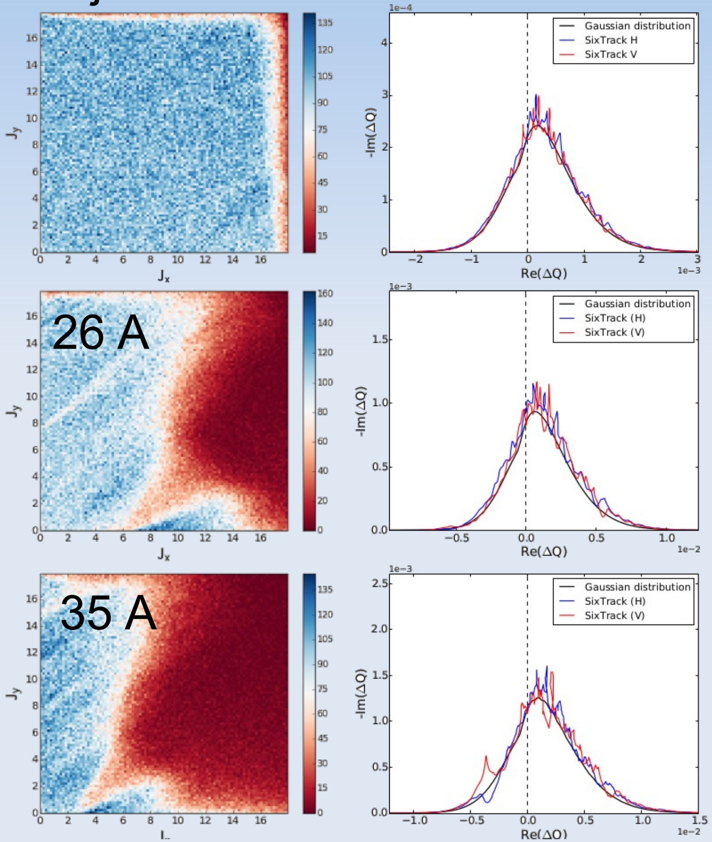
Injection Ioct = 6.5 A



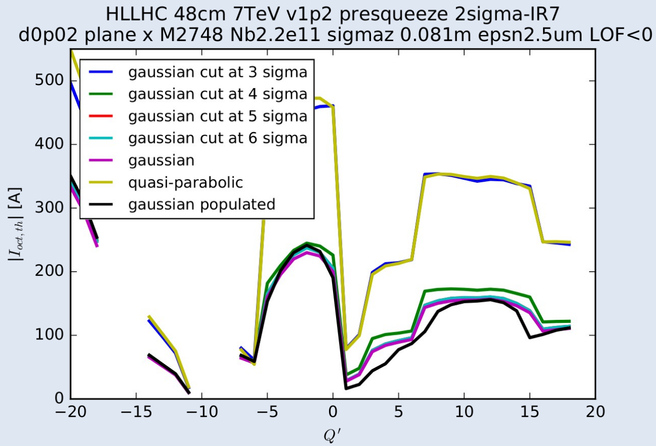
- The impact on the deformation of the distribution can be evaluated in realistic configurations (C. Tambasco et al., Impact of incoherent effects on Landau stability diagram at the LHC, IPAC17)

Non-Gaussian distributions

Injection Ioct = 6.5 A

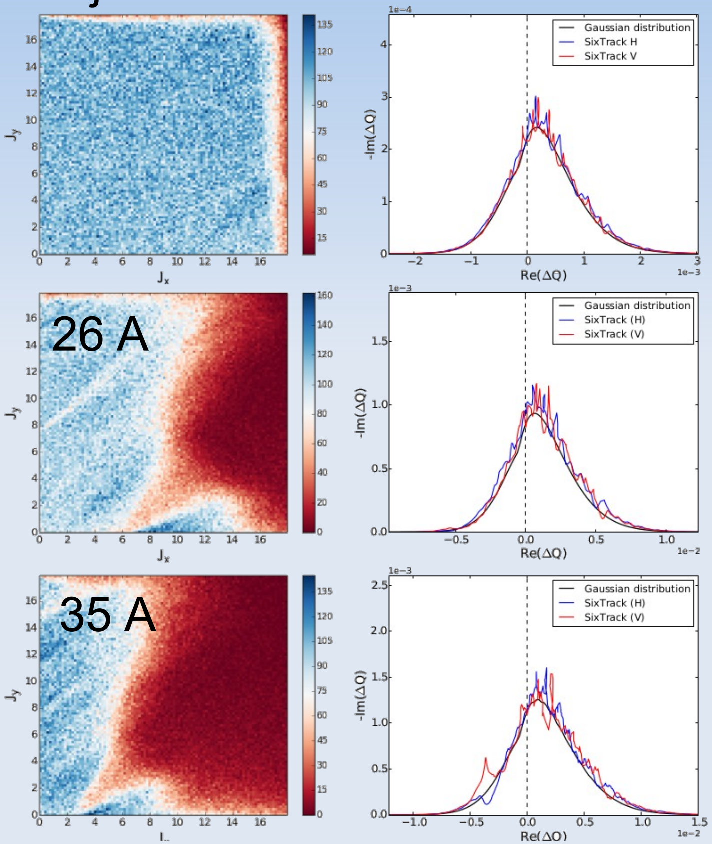


- The impact on the deformation of the distribution can be evaluated in realistic configurations (C. Tambasco et al., Impact of incoherent effects on Landau stability diagram at the LHC, IPAC17)
- Effect of cutting the tails can be evaluated (N. Biancacci et al,. Effect of tail cut and tail population on octupole stability threshold in the HL-LHC, WP2 meeting, 03-10-2016)

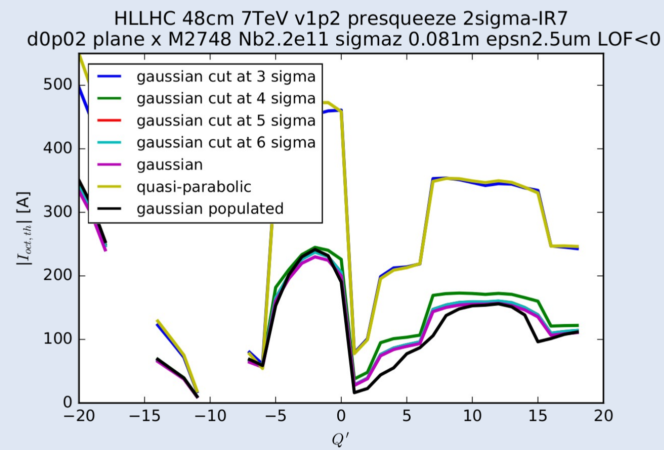


Non-Gaussian distributions

Injection Ioct = 6.5 A



- The impact on the deformation of the distribution can be evaluated in realistic configurations (C. Tambasco et al., Impact of incoherent effects on Landau stability diagram at the LHC, IPAC17)
- Effect of cutting the tails can be evaluated (N. Biancacci et al,. Effect of tail cut and tail population on octupole stability threshold in the HL-LHC, WP2 meeting, 03-10-2016)



→ PySSD is currently used to determine the stability thresholds in all phases of the LHC and HL-LHC cycles where the amplitude detuning is not linear (e.g. beam-beam interactions, coupling) or when the distribution is not Gaussian (e.g. diffusion mechanisms, hollow e-lens)

Implementation

- Python2.6 and python3 compatible, numpy (tested 1.7.0 and above)
- Object oriented
- The sources are available at https://github.com/PyCOMPLETE/PySSD/ (sixtrack interface is not there yet)
- No license
- No parallelism implemented
- No speed optimisation
- No documentation (except for the references mentionned)
- Several integrators available, but only the slowest most robust one is used (fixed grid rectangular) to avoid numerical issues with badly behaving footprints / distributions

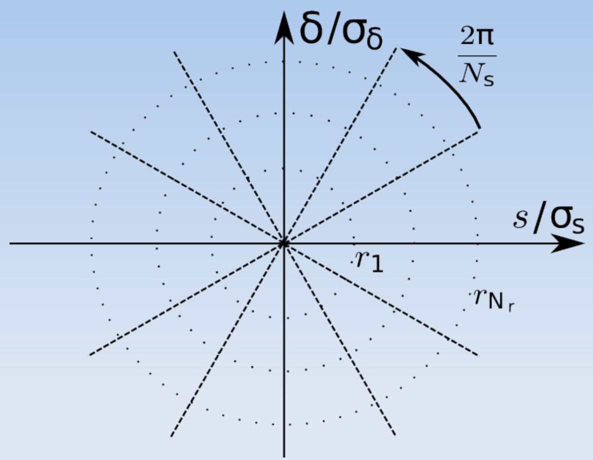
Needs and future plans

- PySSD is run usually on lxplus (MADnPySSD : 1 MAD-X HD footprint 8nh + 2 PySSD (horizontal and vertical) 2nd each
- Each study consists of tens of MADnPySSD run → Current resources (LSF / HTCondor) are appropriate to cover the needs

- Include M. Schenk's implementation of longitudinal to transverse Landau damping based on the dispersion integral derived by A. Maillard
- Efforts on speed improvement would be welcome, but not urgent needs

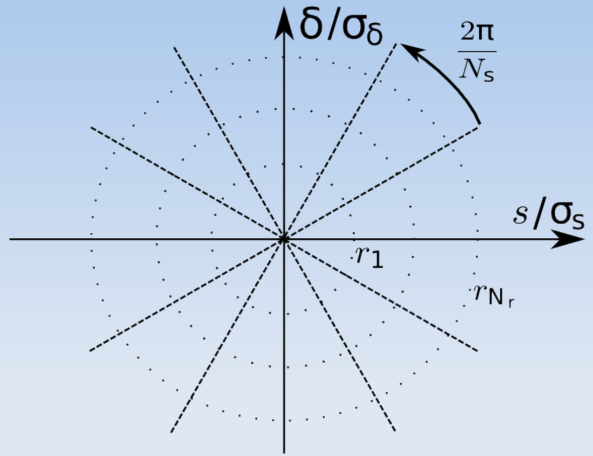
The circulant matrix model

- Initially developed for VEPP-2M (E. A. Perevedenstev and A. A. Valishev, “Simulation of the head-tail instability of colliding bunches,” Phys. Rev. ST Accel. Beams 4, 024403, 2001)
  - One bunch / one beam-beam interaction
- Derive the transverse linearised equation of motion for a discretised longitudinal distribution, including :
  - Linear synchrotron and betatron motion
  - First and second order chromaticity
  - Dipolar and quadrupolar, single/multibunch beam coupling impedance
  - Linearised coherent beam-beam forces (round, 4D, 6D, arbitrary configurations)
  - Perfect transverse feedback, ADT model
  - RFQ (M. Schenk)



The circulant matrix model

- Initially developed for VEPP-2M (E. A. Perevedenstev and A. A. Valishev, “Simulation of the head-tail instability of colliding bunches,” Phys. Rev. ST Accel. Beams 4, 024403, 2001)
  - One bunch / one beam-beam interaction
- Derive the transverse linearised equation of motion for a discretised longitudinal distribution, including :
  - Linear synchrotron and betatron motion
  - First and second order chromaticity
  - Dipolar and quadrupolar, single/multibunch beam coupling impedance
  - Linearised coherent beam-beam forces (round, 4D, 6D, arbitrary configurations)
  - Perfect transverse feedback, ADT model
  - RFQ (M. Schenk)



$$\underline{x}(t) = M^{t}_{One\,turn}\,\underline{x}(0) = \sum_j e^{-2\pi i Q_j t}\,\underline{v}_j$$

→ Analyse the stability of the one turn matrix with normal mode analysis

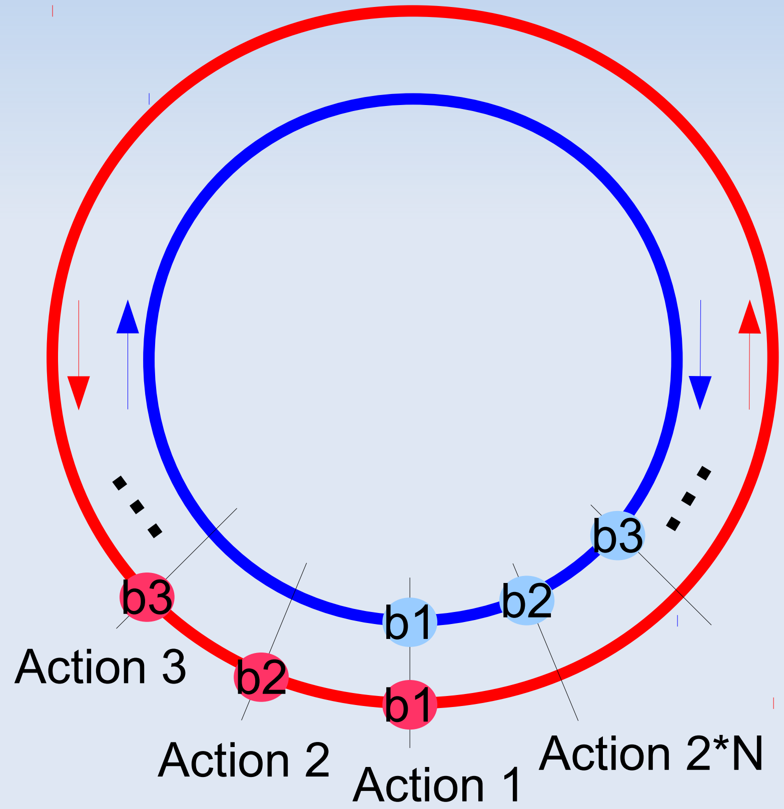
The circulant matrix model

- The equivalence between the CMM and Vlasov treatement (As in D. Amorim et al., DELPHI and Vlasov solvers used at CERN, ABP-CWG 16 Mar. 2017) was demonstrated recently by A. Maillard (paper in preparation)
- Intrinsic limitations :
  - Transverse Landau damping (the transverse motion has to be linear)
  - Multiturn wakes (derivation of a one turn matrix)
  - Non-normality of the matrices (multibunch instabilities)

→ Vlasov solvers also have intrinsic limitations, the circulant matrix model is often complementary

Building the one turn matrix

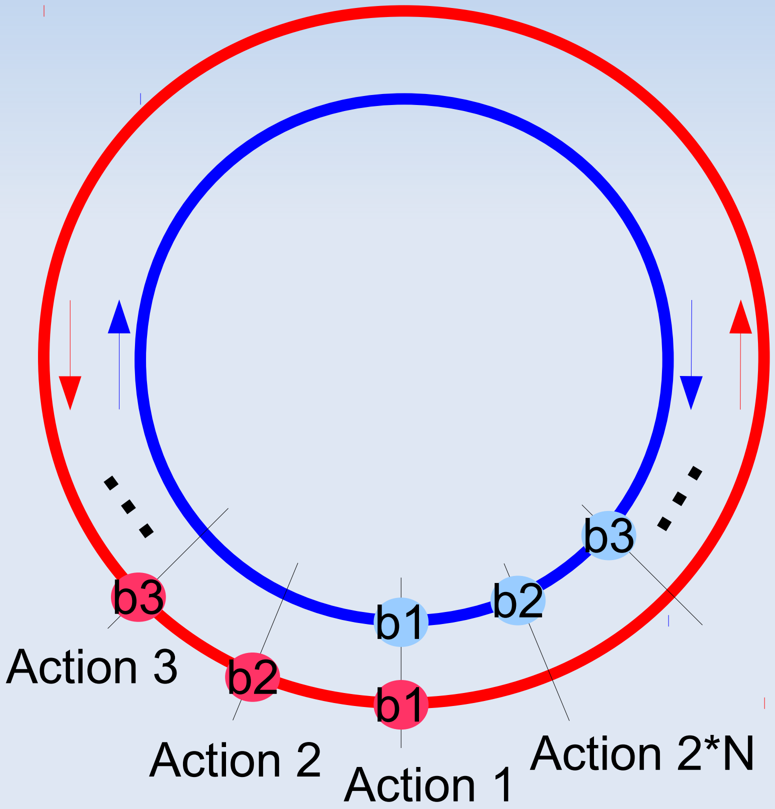
Full system basis : Beam ⊗ Bunch ⊗ Ring ⊗ Slice ⊗ Transverse dof



Building the one turn matrix

Full system basis : Beam ⊗ Bunch ⊗ Ring ⊗ Slice ⊗ Transverse dof

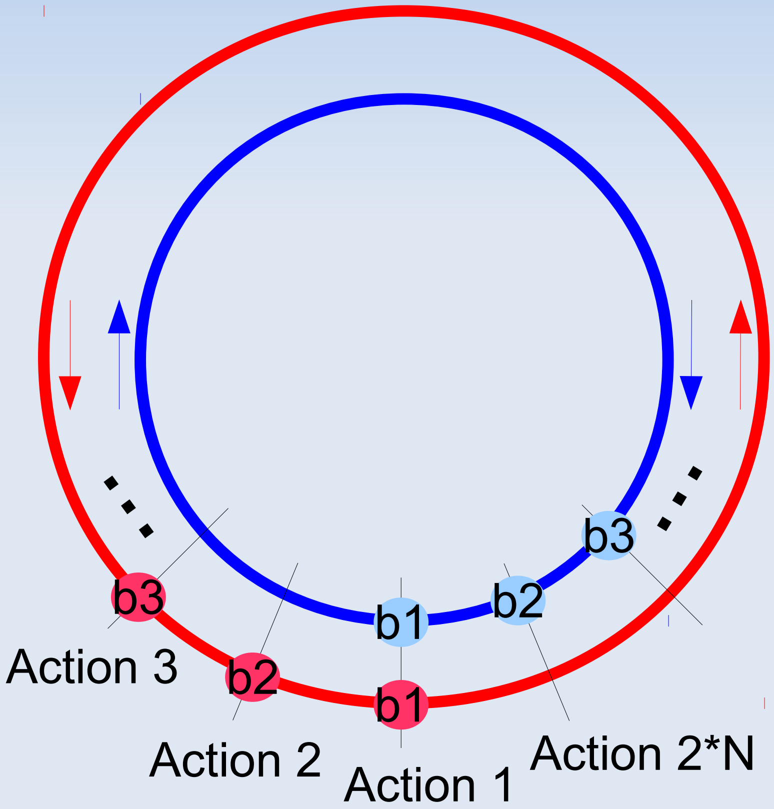
- The layout of the two beams are described by an action sequence



Building the one turn matrix

Full system basis : Beam ⊗ Bunch ⊗ Ring ⊗ Slice ⊗ Transverse dof

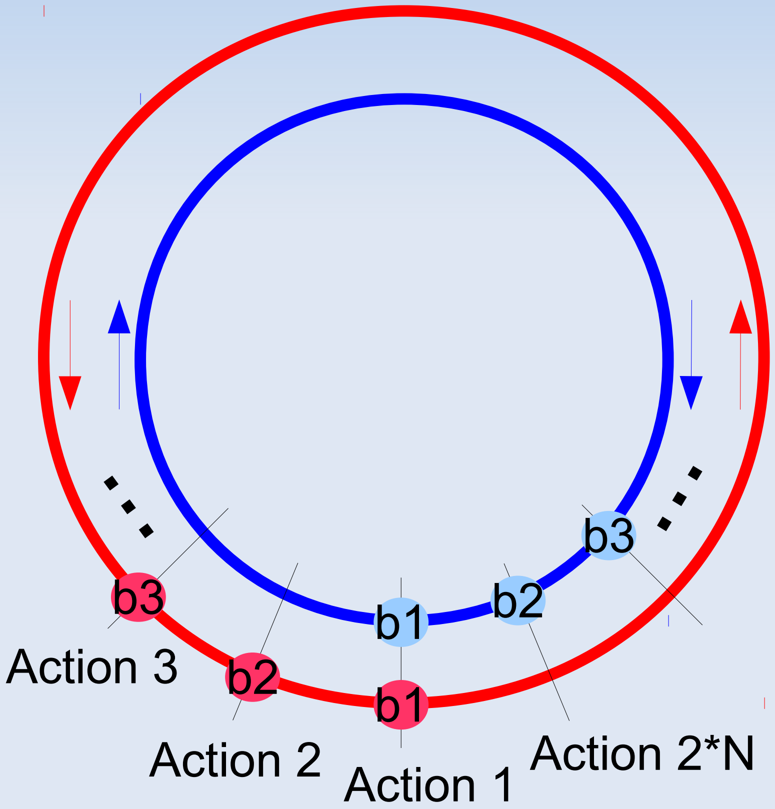
- The layout of the two beams are described by an action sequence
- The bunch of the two beams move towards each other in steps



Building the one turn matrix

Full system basis : Beam ⊗ Bunch ⊗ Ring ⊗ Slice ⊗ Transverse dof

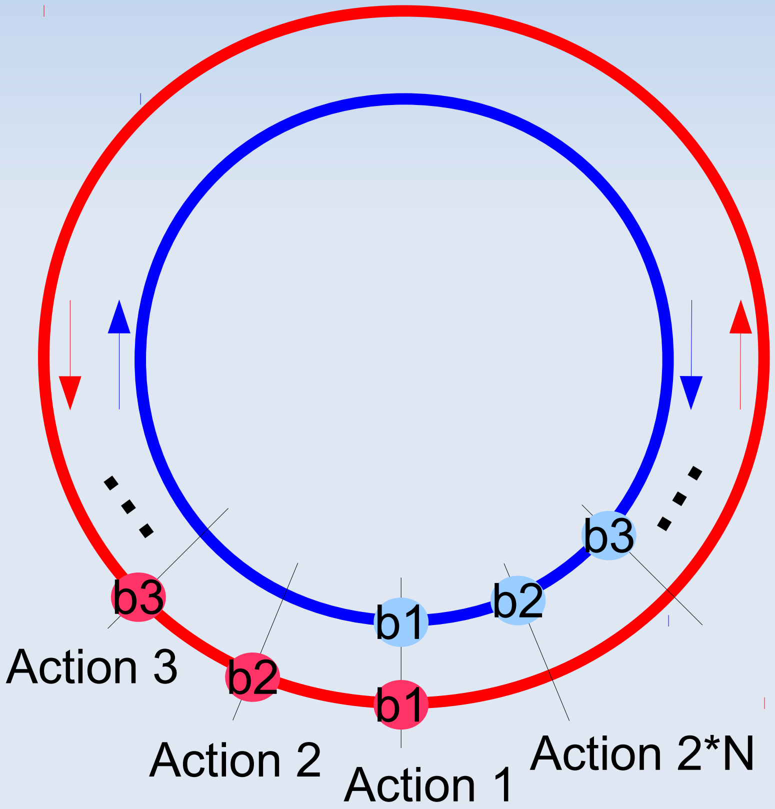
- The layout of the two beams are described by an action sequence
- The bunch of the two beams move towards each other in steps
- Starting with the identity expressed in the full sytem basis



Building the one turn matrix

Full system basis : Beam ⊗ Bunch ⊗ Ring ⊗ Slice ⊗ Transverse dof

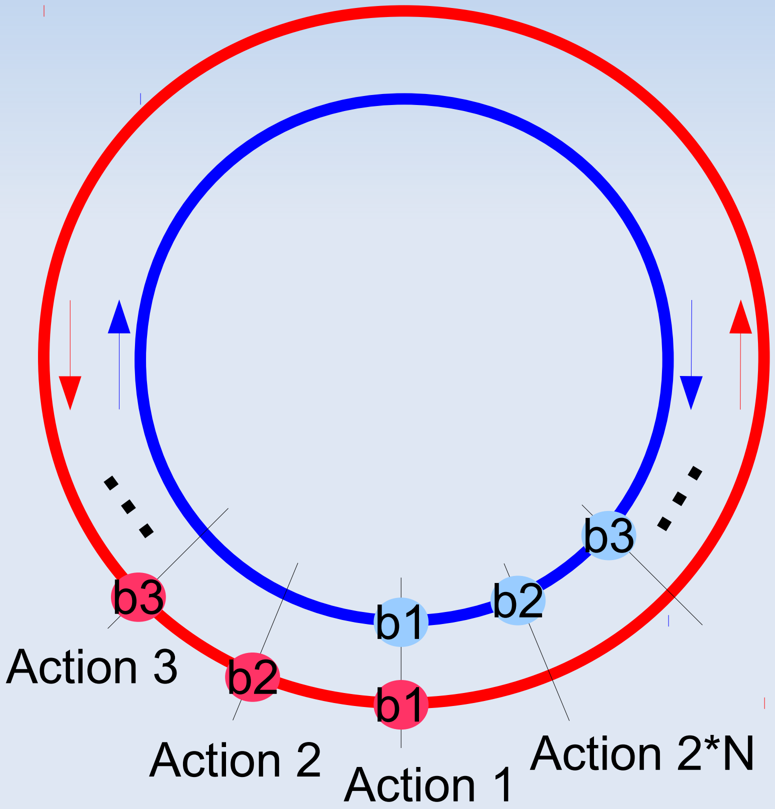
- The layout of the two beams are described by an action sequence
- The bunch of the two beams move towards each other in steps
- Starting with the identity expressed in the full sytem basis
- At each step the matrix corresponding a given action is computed for each bunch of the two beams



Building the one turn matrix

Full system basis : Beam ⊗ Bunch ⊗ Ring ⊗ Slice ⊗ Transverse dof

- The layout of the two beams are described by an action sequence
- The bunch of the two beams move towards each other in steps
- Starting with the identity expressed in the full sytem basis
- At each step the matrix corresponding a given action is computed for each bunch of the two beams
- The matrices are projected in the full system basis, and multiplied to matrix obtained at the previous step



Two examples of matrices expressed in the full system basis

Two bunches colliding, 1 slice, 1 ring :

$$B_0(\delta) = \begin{pmatrix} cos(\phi_x(\delta)) & \beta_x sin(\phi_x(\delta)) & 0 & 0 \\ -\frac{1}{\beta_x} sin(\phi_x(\delta)) & cos(\phi_x(\delta)) & 0 & 0 \\ 0 & 0 & cos(\phi_y(\delta)) & \beta_y sin(\phi_y(\delta)) \\ 0 & 0 & -\frac{1}{\beta_y} sin(\phi_y(\delta)) & cos(\phi_y(\delta)) \end{pmatrix}$$

$$C_{\mathrm{BB}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{\partial \Delta x'_{\mathrm{coh}}}{\partial x}(x_0, y_0) & 1 & \frac{\partial \Delta x'_{\mathrm{coh}}}{\partial x}(x_0, y_0) & 0 \\ 0 & 0 & 1 & 0 \\ \frac{\partial \Delta x'_{\mathrm{coh}}}{\partial x}(x_0, y_0) & 0 & -\frac{\partial \Delta x'_{\mathrm{coh}}}{\partial x}(x_0, y_0) & 1 \end{pmatrix}$$

Two examples of matrices expressed in the full system basis

Two bunches colliding, 1 slice, 1 ring :

$$B_0(\delta)=\begin{pmatrix} cos(\phi_x(\delta)) & \beta_x sin(\phi_x(\delta)) & 0 & 0 \\ -\frac{1}{\beta_x} sin(\phi_x(\delta)) & cos(\phi_x(\delta)) & 0 & 0 \\ 0 & 0 & cos(\phi_y(\delta)) & \beta_y sin(\phi_y(\delta)) \\ 0 & 0 & -\frac{1}{\beta_y} sin(\phi_y(\delta)) & cos(\phi_y(\delta)) \end{pmatrix}$$

$$C_{BB}=\begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{\partial \Delta x'_{coh}}{\partial x}(x_0, y_0) & 1 & \frac{\partial \Delta x'_{coh}}{\partial x}(x_0, y_0) & 0 \\ 0 & 0 & 1 & 0 \\ \frac{\partial \Delta x'_{coh}}{\partial x}(x_0, y_0) & 0 & -\frac{\partial \Delta x'_{coh}}{\partial x}(x_0, y_0) & 1 \end{pmatrix}$$

The effect of synchotron motion is introduced via a circulant matrix :

Betatron motion for multiple slices:

$$B_\beta=\begin{pmatrix} w_1 B_0(\delta_0) & & & \\ & w_2 B_0(\delta_1) & & \\ & & \ddots & \\ & & & w_{N_c} B_0(\delta_{N_c}) \end{pmatrix}$$

Two examples of matrices expressed in the full system basis

Two bunches colliding, 1 slice, 1 ring :

$$B_0(\delta)=\begin{pmatrix} cos(\phi_x(\delta)) & \beta_x sin(\phi_x(\delta)) & 0 & 0 \\ -\frac{1}{\beta_x} sin(\phi_x(\delta)) & cos(\phi_x(\delta)) & 0 & 0 \\ 0 & 0 & cos(\phi_y(\delta)) & \beta_y sin(\phi_y(\delta)) \\ 0 & 0 & -\frac{1}{\beta_y} sin(\phi_y(\delta)) & cos(\phi_y(\delta)) \end{pmatrix}$$

$$C_{\mathrm{BB}}=\begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{\partial \Delta x'_{\mathrm{coh}}}{\partial x}(x_0,y_0) & 1 & \frac{\partial \Delta x'_{\mathrm{coh}}}{\partial x}(x_0,y_0) & 0 \\ 0 & 0 & 1 & 0 \\ \frac{\partial \Delta x'_{\mathrm{coh}}}{\partial x}(x_0,y_0) & 0 & -\frac{\partial \Delta x'_{\mathrm{coh}}}{\partial x}(x_0,y_0) & 1 \end{pmatrix}$$

The effect of synchotron motion is introduced via a circulant matrix :

Betatron motion for multiple slices:

$$B_\beta=\begin{pmatrix} w_1 B_0(\delta_0) & & & \\ & w_2 B_0(\delta_1) & & \\ & & \ddots & \\ & & & w_{N_c} B_0(\delta_{N_c}) \end{pmatrix}$$

Weighting for equipopulated / equidistant slices

Two examples of matrices expressed in the full system basis

Two bunches colliding, 1 slice, 1 ring :

$$B_0(\delta) = \begin{pmatrix} cos(\phi_x(\delta)) & \beta_x sin(\phi_x(\delta)) & 0 & 0 \\ -\frac{1}{\beta_x} sin(\phi_x(\delta)) & cos(\phi_x(\delta)) & 0 & 0 \\ 0 & 0 & cos(\phi_y(\delta)) & \beta_y sin(\phi_y(\delta)) \\ 0 & 0 & -\frac{1}{\beta_y} sin(\phi_y(\delta)) & cos(\phi_y(\delta)) \end{pmatrix}$$

$$C_{BB} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{\partial \Delta x'_{coh}}{\partial x}(x_0, y_0) & 1 & \frac{\partial \Delta x'_{coh}}{\partial x}(x_0, y_0) & 0 \\ 0 & 0 & 1 & 0 \\ \frac{\partial \Delta x'_{coh}}{\partial x}(x_0, y_0) & 0 & -\frac{\partial \Delta x'_{coh}}{\partial x}(x_0, y_0) & 1 \end{pmatrix}$$

The effect of synchotron motion is introduced via a circulant matrix :

Betatron motion for multiple slices:

$$B_\beta = \begin{pmatrix} w_1 B_0(\delta_0) & & & \\ & w_2 B_0(\delta_1) & & \\ & & \ddots & \\ & & & w_{N_c} B_0(\delta_{N_c}) \end{pmatrix}$$

Weighting for equipopulated / equidistant slices

Chromaticity effect depend on the Δp/p of a given ring

Two examples of matrices expressed in the full system basis

Two bunches colliding, 1 slice, 1 ring :

$$B_0(\delta)=\begin{pmatrix} cos(\phi_x(\delta)) & \beta_x sin(\phi_x(\delta)) & 0 & 0 \\ -\frac{1}{\beta_x} sin(\phi_x(\delta)) & cos(\phi_x(\delta)) & 0 & 0 \\ 0 & 0 & cos(\phi_y(\delta)) & \beta_y sin(\phi_y(\delta)) \\ 0 & 0 & -\frac{1}{\beta_y} sin(\phi_y(\delta)) & cos(\phi_y(\delta)) \end{pmatrix}$$

$$C_{BB}=\begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{\partial \Delta x'_{coh}}{\partial x}(x_0,y_0) & 1 & \frac{\partial \Delta x'_{coh}}{\partial x}(x_0,y_0) & 0 \\ 0 & 0 & 1 & 0 \\ \frac{\partial \Delta x'_{coh}}{\partial x}(x_0,y_0) & 0 & -\frac{\partial \Delta x'_{coh}}{\partial x}(x_0,y_0) & 1 \end{pmatrix}$$

The effect of synchotron motion is introduced via a circulant matrix :

Circulant matrix dependent on Qs :

$$S_r = P_{N_s}^{N_s Q_s}$$

$$P_{N_s}=\begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ 1 & & & 0 & 1 \end{pmatrix}$$

Betatron motion for multiple slices:

$$B_\beta=\begin{pmatrix} w_1 B_0(\delta_0) & & & \\ & w_2 B_0(\delta_1) & & \\ & & \ddots & \\ & & & w_{N_c} B_0(\delta_{N_c}) \end{pmatrix}$$

Weighting for equipopulated / equidistant slices

Chromaticity effect depend on the Δp/p of a given ring

Two examples of matrices expressed in the full system basis

Two bunches colliding, 1 slice, 1 ring :

$$B_0(\delta)=\begin{pmatrix} cos(\phi_x(\delta)) & \beta_x sin(\phi_x(\delta)) & 0 & 0 \\ -\frac{1}{\beta_x} sin(\phi_x(\delta)) & cos(\phi_x(\delta)) & 0 & 0 \\ 0 & 0 & cos(\phi_y(\delta)) & \beta_y sin(\phi_y(\delta)) \\ 0 & 0 & -\frac{1}{\beta_y} sin(\phi_y(\delta)) & cos(\phi_y(\delta)) \end{pmatrix}$$

$$C_{BB}=\begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{\partial \Delta x'_{coh}}{\partial x}(x_0,y_0) & 1 & \frac{\partial \Delta x'_{coh}}{\partial x}(x_0,y_0) & 0 \\ 0 & 0 & 1 & 0 \\ \frac{\partial \Delta x'_{coh}}{\partial x}(x_0,y_0) & 0 & -\frac{\partial \Delta x'_{coh}}{\partial x}(x_0,y_0) & 1 \end{pmatrix}$$

The effect of synchotron motion is introduced via a circulant matrix :

Circulant matrix dependent on Qs :

$$S_r = P_{N_s}^{N_s Q_s}$$

$$P_{N_s}=\begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ 1 & & & 0 & 1 \end{pmatrix}$$

Betatron motion for multiple slices:

$$B_\beta=\begin{pmatrix} w_1 B_0(\delta_0) & & & \\ & w_2 B_0(\delta_1) & & \\ & & \ddots & \\ & & & w_{N_c} B_0(\delta_{N_c}) \end{pmatrix}$$

The effect is identical in each ring :

$$S_0 = \mathbb{I}_{N_r} \otimes S_r \, .$$

Weighting for equipopulated / equidistant slices

Chromaticity effect depend on the Δp/p of a given ring

Two examples of matrices expressed in the full system basis

Two bunches colliding, 1 slice, 1 ring :

$$B_0(\delta)=\begin{pmatrix} cos(\phi_x(\delta)) & \beta_x sin(\phi_x(\delta)) & 0 & 0 \\ -\frac{1}{\beta_x} sin(\phi_x(\delta)) & cos(\phi_x(\delta)) & 0 & 0 \\ 0 & 0 & cos(\phi_y(\delta)) & \beta_y sin(\phi_y(\delta)) \\ 0 & 0 & -\frac{1}{\beta_y} sin(\phi_y(\delta)) & cos(\phi_y(\delta)) \end{pmatrix}$$

$$C_{BB}=\begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{\partial \Delta x'_{coh}}{\partial x}(x_0,y_0) & 1 & \frac{\partial \Delta x'_{coh}}{\partial x}(x_0,y_0) & 0 \\ 0 & 0 & 1 & 0 \\ \frac{\partial \Delta x'_{coh}}{\partial x}(x_0,y_0) & 0 & -\frac{\partial \Delta x'_{coh}}{\partial x}(x_0,y_0) & 1 \end{pmatrix}$$

The effect of synchotron motion is introduced via a circulant matrix :

Betatron motion for multiple slices:

$$B_\beta=\begin{pmatrix} w_1 B_0(\delta_0) & & & \\ & w_2 B_0(\delta_1) & & \\ & & \ddots & \\ & & & w_{N_c} B_0(\delta_{N_c}) \end{pmatrix}$$

Weighting for equipopulated / equidistant slices

Chromaticity effect depend on the Δp/p of a given ring

Circulant matrix dependent on Qs :

$$S_r = P_{N_s}^{N_s Q_s}$$

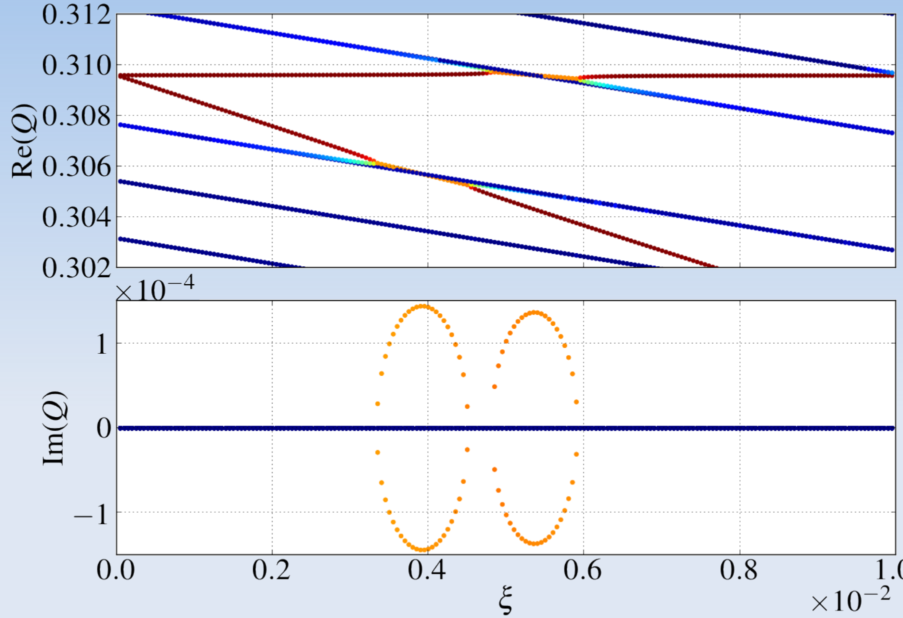
$$P_{N_s}=\begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ 1 & & & 0 & 1 \end{pmatrix}$$

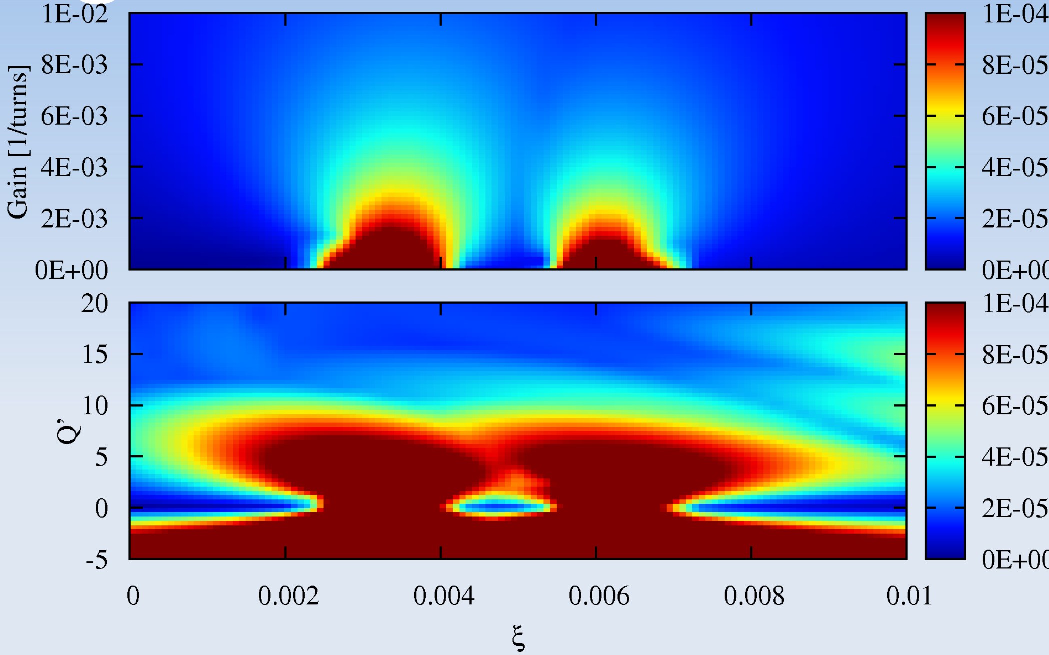
The effect is identical in each ring :

$$S_0 = \mathbb{I}_{N_r} \otimes S_r \;\longrightarrow\; M_0 = S_0 \otimes B_0$$

(More details in X. Buffat, PhD thesis, 2015)

Mode coupling instability of colliding beams



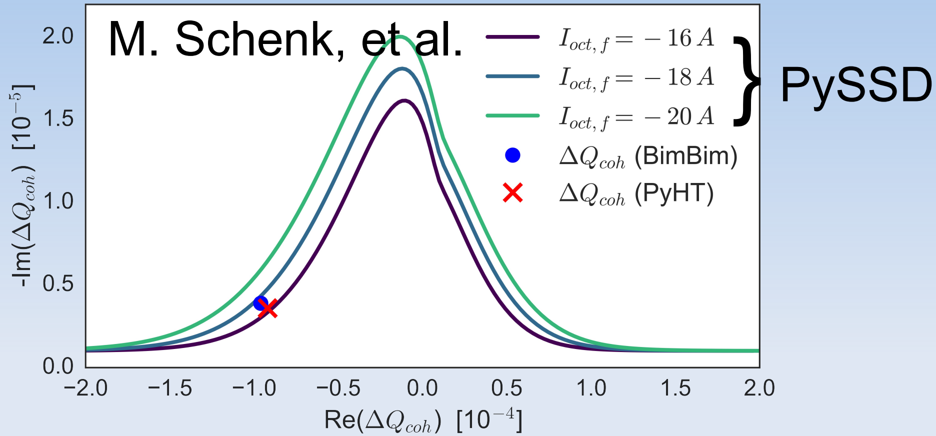


- The coupling of beam-beam and head-tail modes leads to strong instabilities
- They are well mitigated by a transverse feedback and/or high chromaticity

→ These mitigations (at least one) are needed in the LHC to bring the beams into collision and level the luminosity with a transverse offset (e.g IPs 2 and 8)

(S. White et al., Transverse mode coupling instability of colliding beams, Phys. Rev. ST Accel. Beams 17, 041002, 2014)

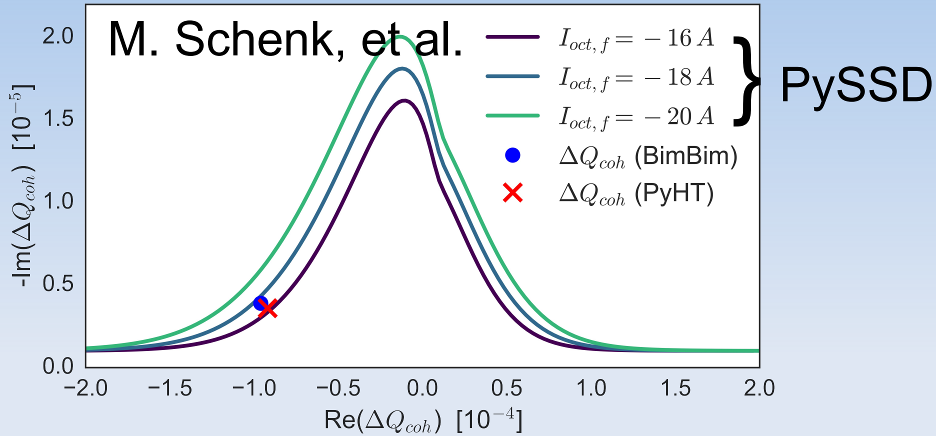
Stability of head-tail modes

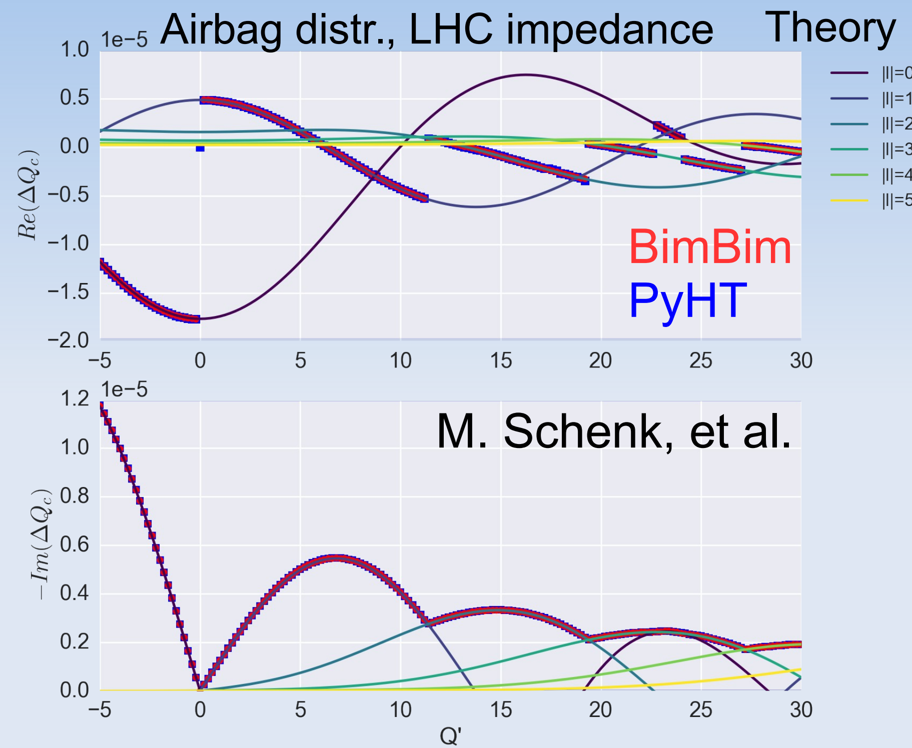


→ Octupole threshold consistent with PyHT octupole scan and measurements at the LHC

- BimBim is powerfull in predicting the stability threshold when combined with PySSD

Stability of head-tail modes



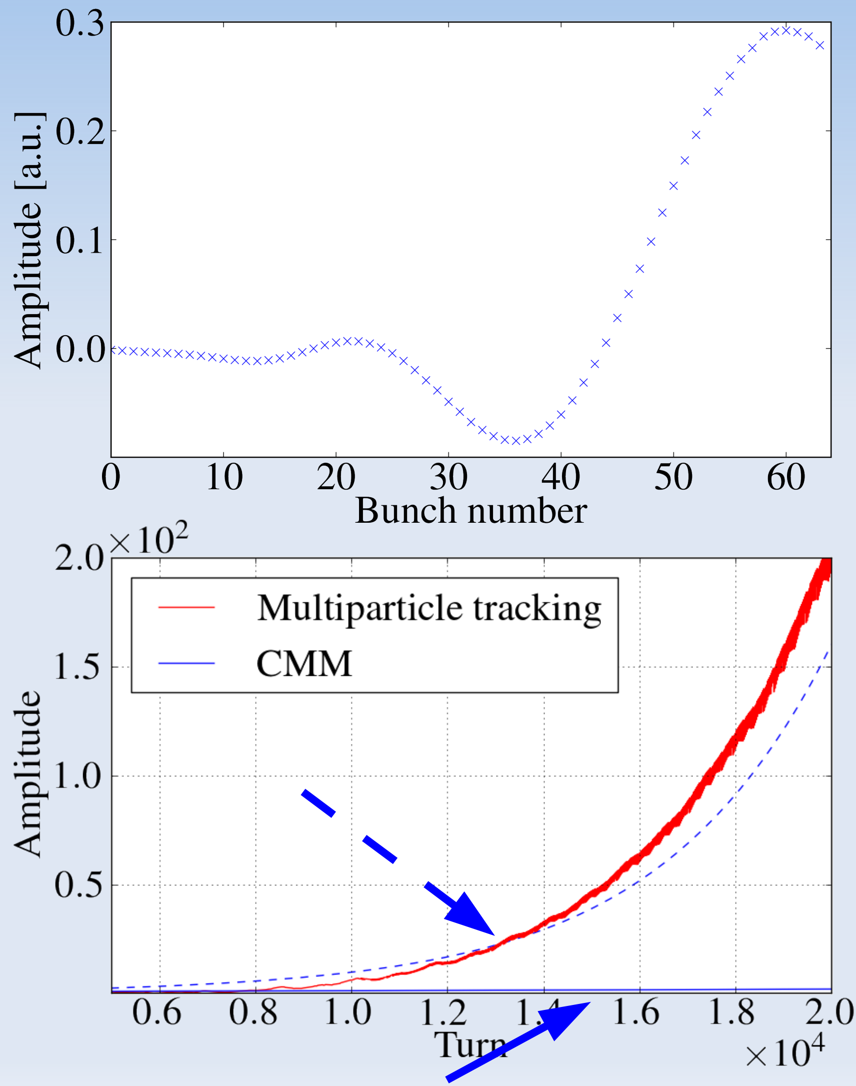


→ Octupole threshold consistent with PyHT octupole scan and measurements at the LHC

- BimBim is powerfull in predicting the stability threshold when combined with PySSD
- The flexibility of BimBim allows for a better understanding of the mechanisms and benchmark with simplied analytical models, other Vlasov solvers and tracking simulations (Sacherer, COMBI, PyHT, multbunch HT, DELPHI, NHT)

Non-normality

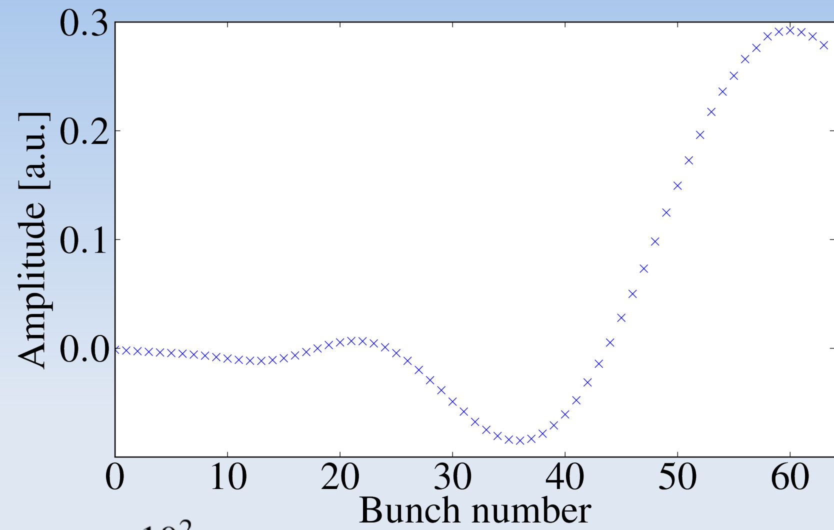
- The dynamics of bunch trains is not well described by normal mode analysis (several identical eigenvalues)

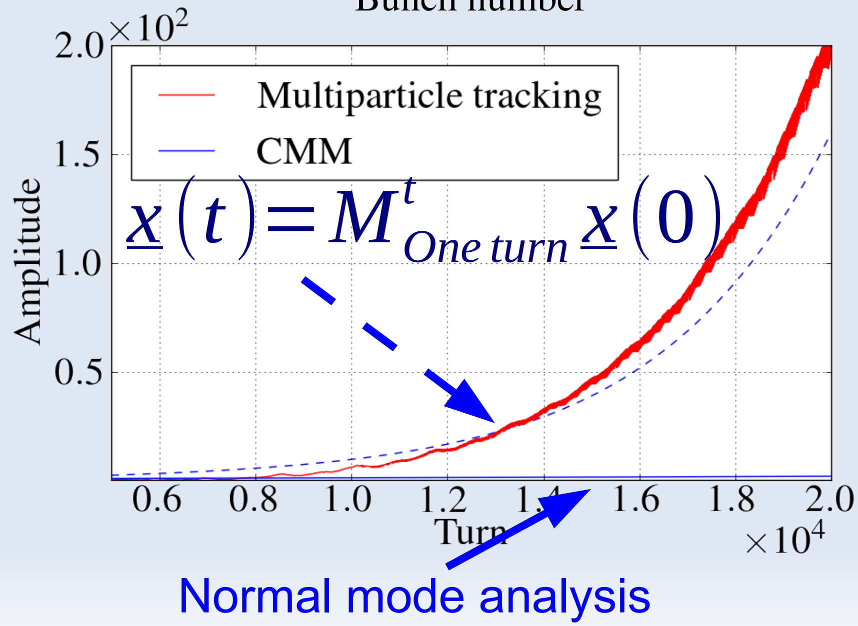


Normal mode analysis

Non-normality

- The dynamics of bunch trains is not well described by normal mode analysis (several identical eigenvalues)
  - Usually adressed by considering uniformly filled machines (e.g. DELPHI, NHT)
- Modern analysis tools are required to analyse the matrix (pseudospectrum)
  - The non-normal behavior of the beams in the LHC is well mitigated by the transverse feedback





Usage of BimBim

- BimBim is currently used at CERN to :
  - Study single/multibunch instability mechanims involving the impedance and beam-beam interactions in the LHC
  - Study instability mechanisms in the presence of second order chromaticity or an RFQ
  - Benchmark between formulas and codes
  - Complement other instability models' predixtions with different limitations

Implementation

- Python2.6 and python3 compatible, numpy (tested 1.7.0) and scipy (tested 0.12.0b1)
- The sources are available at https://svnweb.cern.ch/cern/wsvn/BimBim
- Object oriented, with a strong encapsulation (not strictly needed)
- No license
- No parallel implementation
- No documentation (except for the references mentionned)
- Memory and speed optimisation by using sparse matrices to build the intermediate matrices
- By default a regular eigenvalue solver (numpy.linalg.eig) is used since the full one turn map is usually dense

Needs and future plans

Matrix basis : Beam ⊗ Bunch ⊗ Ring ⊗ Slice ⊗ Transverse dof , Matrix size, Memory size
LHC nominal : 2 x 2808 x 20 x 40 x 4 → 10^7x 10^7 →1Pb

- The perfromance of BimBim is limited by memory to a single train of bunches
- A single run takes from <1s to a week depending on the number of bunches and the convergence requirement (number of slices and rings)
- A given mechanism is well covered with tens of runs
- The current resources are appropriate to cover the needs
- Future plans :
  - Flat beams collision
  - Space charge (A. Oeftiger)
  - Linearised dynamical model of the e-cloud ?
  - Memory optimised parallel scheme ?