

Frozen SC Model in MADX-SC – Developments

Valery KAPIN

Fermilab, Batavia, IL

Space Charge Collaboration Meeting 2014
CERN, Geneva, 20-21-May-2014

Abstract

- **MADX-SC code realizes the frozen space charge model within MADX code utilizing multiple BB elements.**
- **The code allows to treat several types of applied tasks for intensive beams.**
- **Presently, the code is mainly integrated into MADX source code.**
- **Initially (2006-12) SC simulations with MADX was implemented by V.Kapin via external MADX-scripts & macros written for a particular tasks.**
- **Normally, the code developments have been driven by particular simulation tasks existed in one of labs**
- **In this talk, the past and feasible MADX-SC developments are discussed by overviewing different simulation tasks**

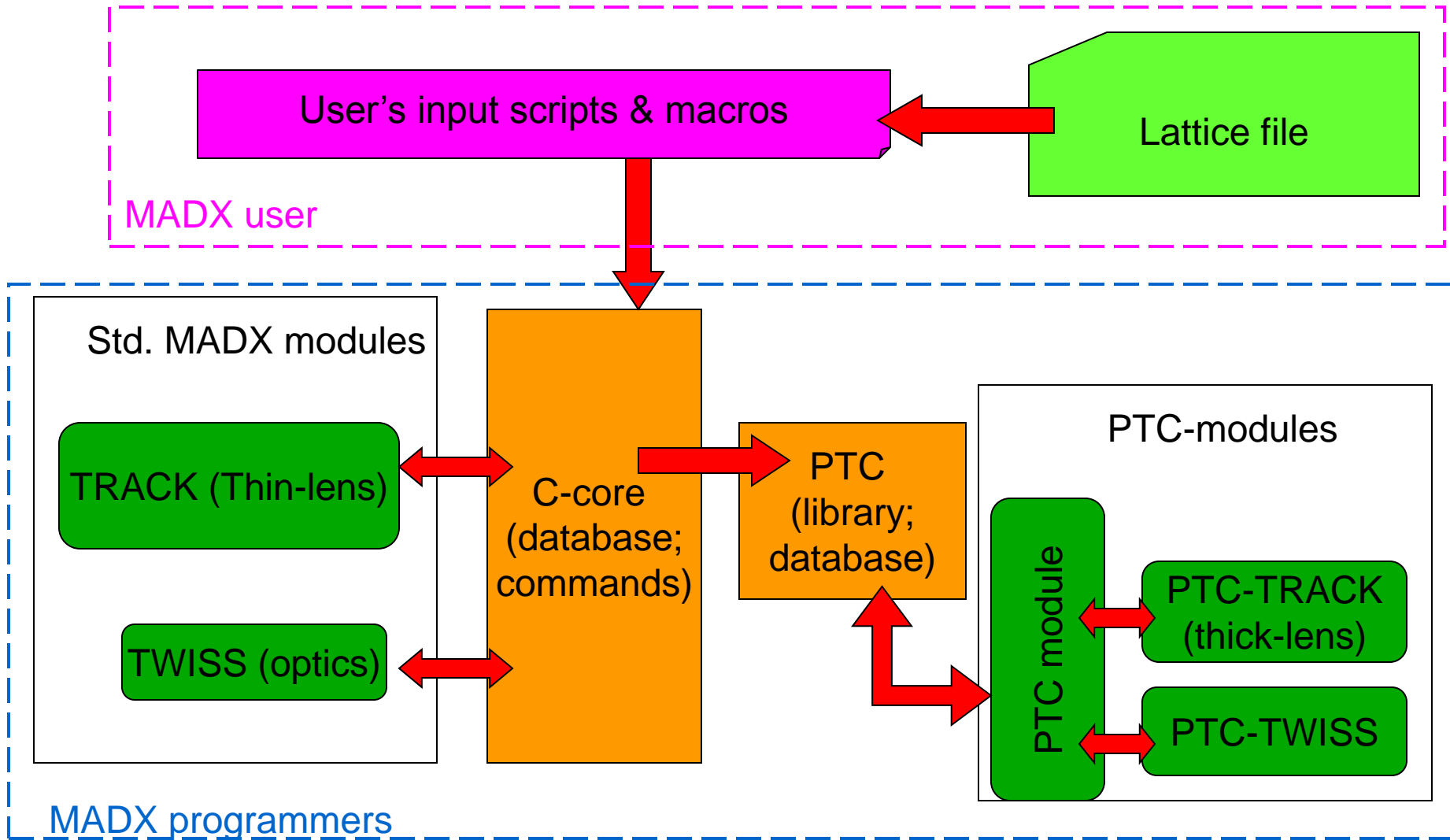
My acknowledgements (contributors 2006-2013):

| Name | Theory/ Algorithms | Coding | Application supply | ~years |
|---------------------|-----------------------|--------|-----------------------|-----------|
| Ya. Alexahin (FNAL) | + | | + | 2008-now |
| A. Bolshakov (ITEP) | | | + | 2006-2008 |
| L. Deniau (CERN) | | + | | 2013-now |
| G. Franchetti (GSI) | + | | + | 2008-2010 |
| V. Kornilov (GSI) | | | + | 2009 |
| V. Nagaslaev (FNAL) | | | + | 2010 |
| F. Schmidt (CERN) | | + | | 2005-now |
| S. Sorge (GSI) | | | + | 2009 |
| P. Zenkevich (ITEP) | + | | + | 2005-2008 |

Past/New Tasks&Extensions for SC

| | |
|---|------------------------------------|
| DA with SC and field errors for coasting beam | TWAC 2007 |
| DA under SC for coasting beam (reg. & random multipole errors, c.o., +damp. octupoles) | SIS100 2009 |
| Beam losses under SC & fake synchr. Oscillation. | SIS100 2009 |
| ... plus 4D & 6D benchmarking for SIS18 lattice | GSI, 2009 |
| Slow extraction under SC with real 6D tracking, adaptive emittance and sigmas (betas from TWISS) | Debuncher FNAL, 2010 |
| <i>Multi-turn injection: add injected beam to stored beam; fringe-field maps for large aperture magnets</i> | <i>FNAL proton driver; booster</i> |
| <i>Wake-fields; longitudinal SC</i> | - - |
| <i>Thin-track with energy variation via adaptive emittance calculations w/o TWISS (new YA-algorithm)</i> | - - |

Schematic MADX-structure



ITEP (Moscow)-CERN collaboration on MADX in 2004-2006

- MAD-X is the successor of MAD-8 (frozen in 2002).
- MAD-X has a modular organization =>
Team in 200x: custodian (F.Schmidt) + Module Keepers
- “PTC-TRACK module” is developed by
V.Kapin (ITEP) & F.Schmidt (CERN)
- In ~2006 P. Zenkevich (ITEP) has suggested to introduce
sp. ch. effect in MADX simulations via its BB-elements
- The idea is not new (e.g. *M. Furman, PAC-1987*). It has
been implemented in other beam dynamics codes.
- My task was its step-by-step adaptation to MADX

Initial plans for S.C. in 2006

Two options are discussed

I) **thin-lens-tracking** II) thick-lens (PTC-library):

I.a) usage a “linear” kick-matrix for Twiss parameters;

I.b) a space-charge kicks from "frozen beam" “BB-kicks” during tracking;

II) Model using many BBs in PTC

Presently: BB are in PTC-library (done by E.Forest), but interfacing with MADX -> ??? (out of usage)

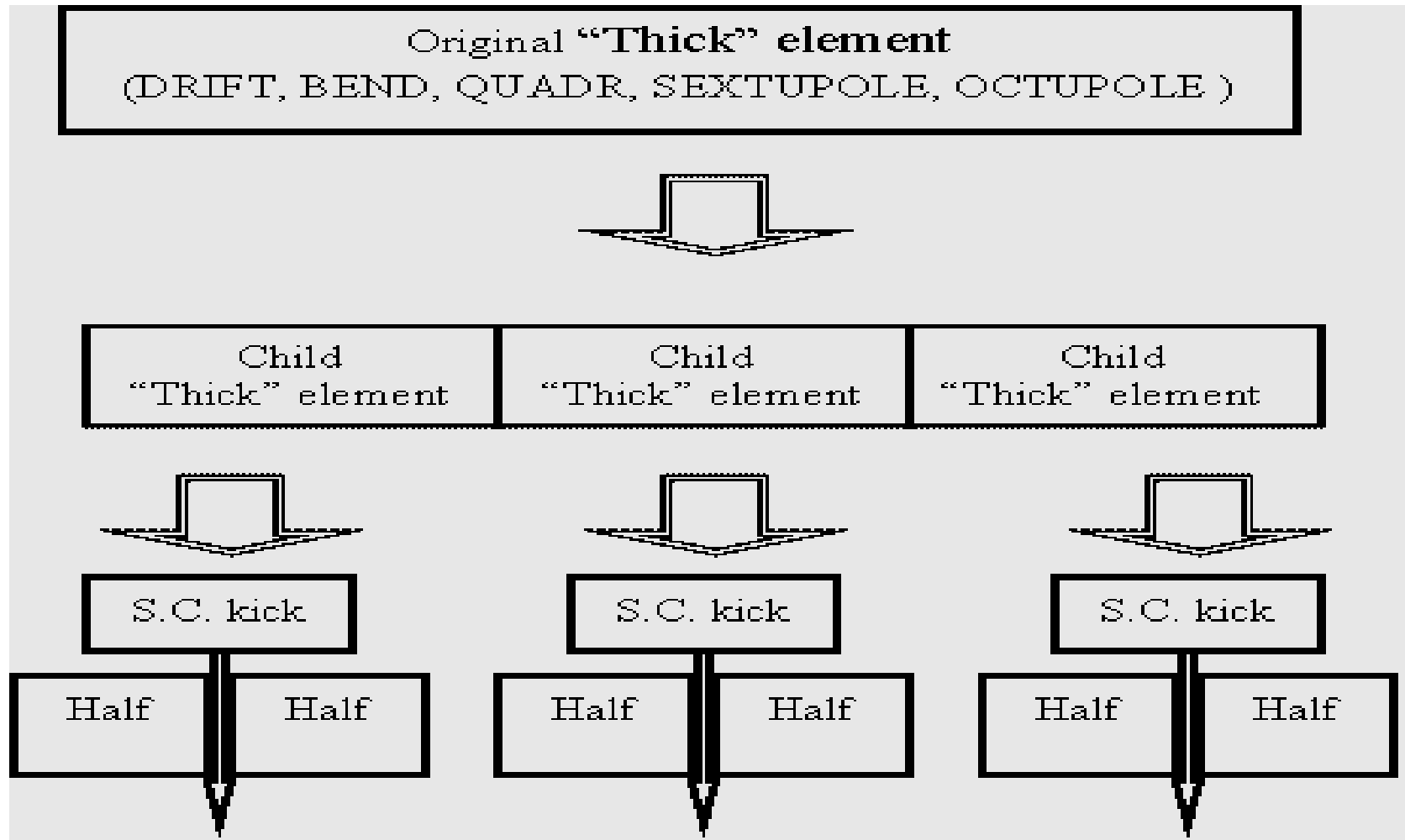
WE USE ONLY 1-st option (THIN-TRACK) for MADX with Sp.Ch. !!!

Lattice preparation for Direct S.C. simulations with MADX*

- 4D (coasting beam) is considered
- most work is done using macros of MADX input scripts and some external FORTRAN subrs.
- Only 2D(transv.) space-charge fields
- "Frozen" charge distribution
either linear (MATRIX) or Gaussian (BB);
- Several space-charge kicks within every thick element (bends, quads, drifts etc.);

* Presently (>2012) – it is core of preparatory process of MADX-SC

Element-by-element splitting obeying the *2nd order ray tracing integrator* for a number of S-C kicks



Reminder about integrators

1) A. Chao, “Adv. Topics”, USPAS, 2000:

Thick element with the foc. strength S and the length L :

1st order “ $O(L)$ ”:

a) drift(L)+kick(SL) ; b) kick(SL)+drift(L)

2nd order “ $O(L^2)$ ”: drift($L/2$) + kick(SL) + drift($L/2$)

2) MAD-9 project (EPAC’00, TUP6B11):

Hamiltonian: $H=H_{\text{ext}}+H_{\text{sc}} \Rightarrow$ 2nd order map:

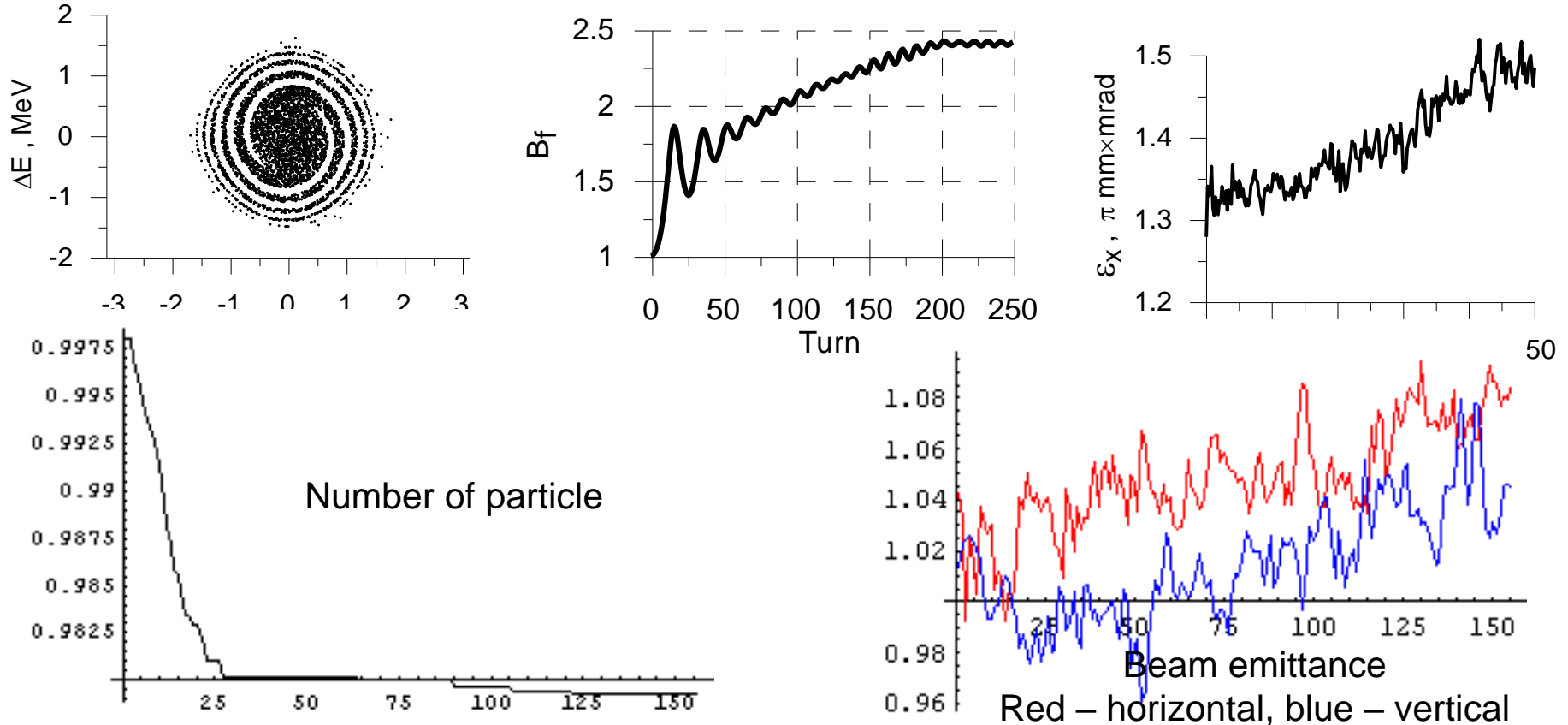
$$M = M_{\text{ext}}(t/2) M_{\text{sc}}(t) M_{\text{ext}}(t/2)$$

Ray Tracing: $(L/2n)(SL/n)(L/2n)$ repeat n times

Example of S.C. with MAD-8 (the 1st order integrator)

Y.Alexahin, A.Drozhdin, N.Kazarinov,

“direct space charge in Booster with **MAD8**”, Beams-doc-2609-v1, 2007



Unfortunately TRACK module of MAD8 (frozen, only executable file) does not permit to have more than 200 BB elements. Thereby the particle tracking has been fulfilled with 197 BB elements with average distance between \sim to 2.4 m.

Basics for SC simulations in that Ref.

1) direct space charge in MAD using set of BEAMBEAM (BB) elements.

Tune shift:

$$\Delta \nu = -\frac{1}{4\pi} \sum_{i=1}^{N_{bb}} \beta_i K_i$$

where β – the Twiss beta-function at BB location, K – kick acting on the particle, N_{bb} – number of BB elements.

2) The number of particle N' in fictitious colliding beam must be set as:

$$N' = B_f \frac{NL_i}{C(\gamma^2 - 1)}$$

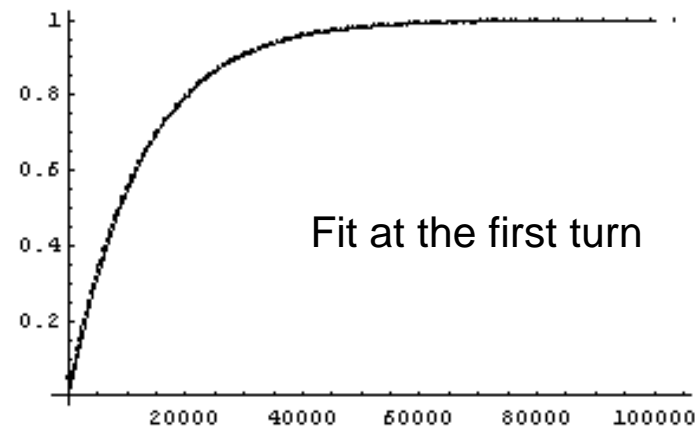
Here B_f – bunching factor, N – number of particle, C – circumference, L_i – distance between successive BB elements, γ – relativistic factor.

3) Simulation with changing emittance from turn to turn of the beam.

The emittance have been evaluated by fitting the integral of distribution function with:

$$F(I) = 1 - \exp(-I / \varepsilon)$$

where I – is action variable.



Self-consistent (linear) BB-sizes

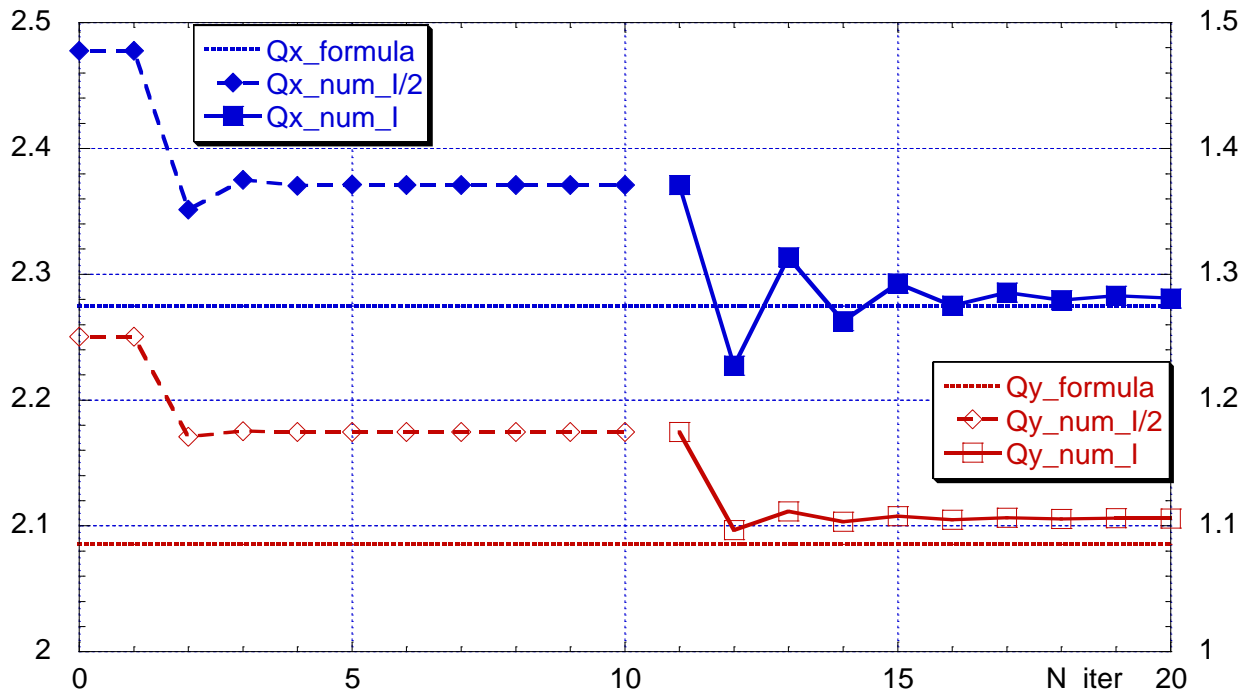
- Space-charge kicks simulated by the 1st order MATRIX for linear TWISS calculations;
- Linearly self-consistent beam sizes calculated by iterations with the TWISS;
- Analytical Laslett's formula and numerical iterations provide near the same tune shifts (for coasting beam !!!)

$$\Delta \nu_{x,y} \approx -\frac{r_0^{\text{part}} N_{\text{part}}}{2\pi B} \frac{1}{\gamma^3 \beta^2} \frac{\beta_{x,y}}{\sigma_{x,y} (\sigma_x + \sigma_y)}$$

$$\sigma_{x,y}^0 = \sqrt{\varepsilon_{x,y} \beta_{x,y}^0}$$

Iterations to find beam sizes at non-zero beam current*

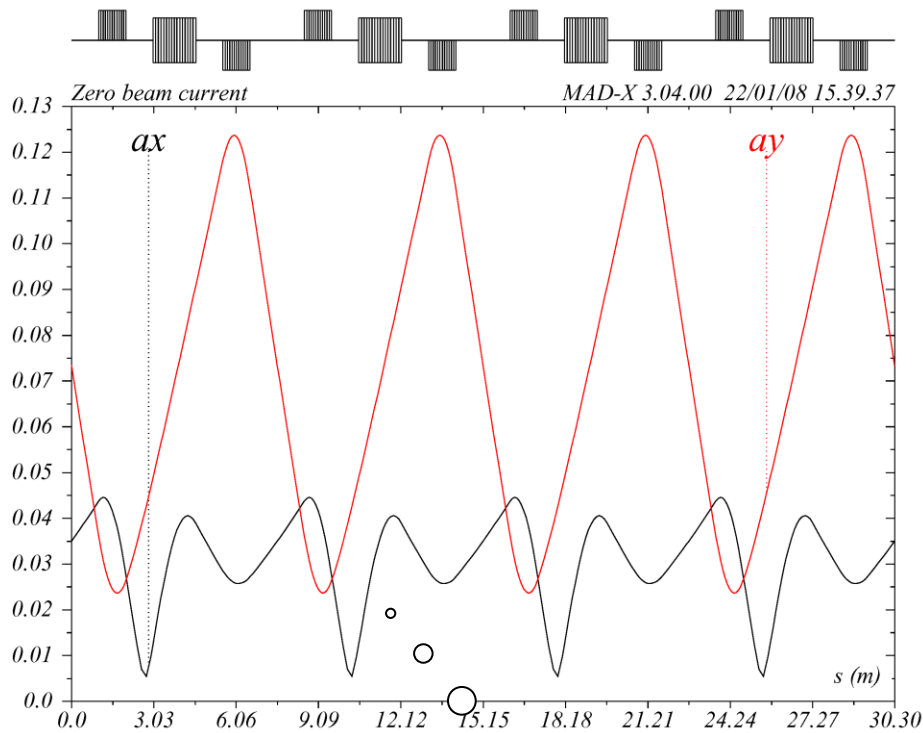
- No Equilibrium solution $Q_{x,y} = \text{int} \Rightarrow$ (TWISS \Rightarrow ERR)
- Tune value oscillates around a final value;
 \Rightarrow near $Q = n$ iterations with steps for the beam current



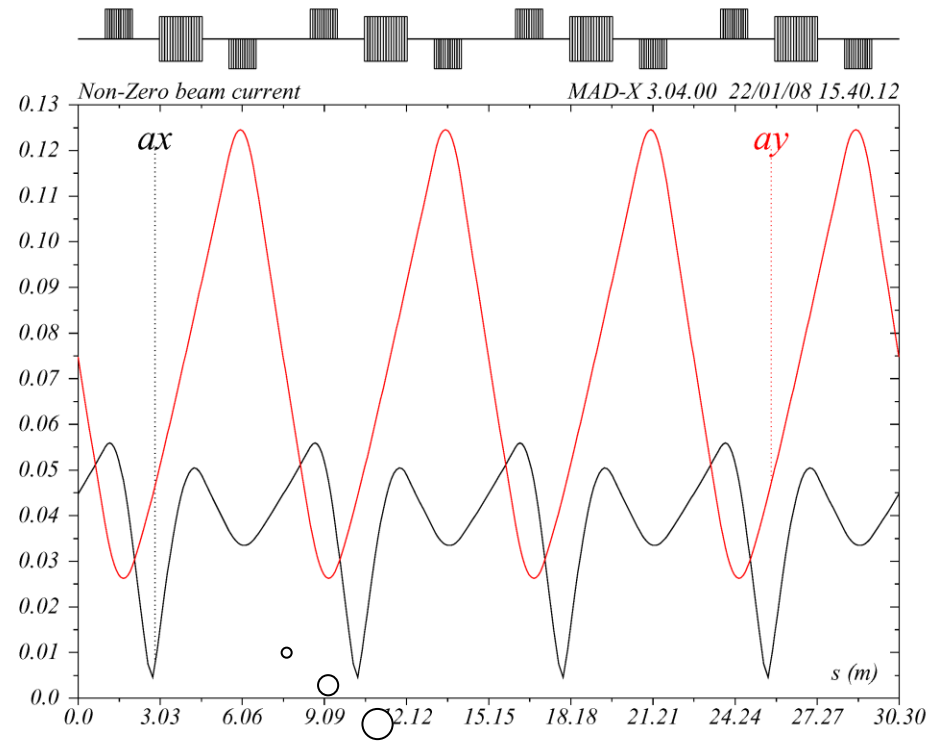
Example of iterations in two steps
Straight lines shows analytical values according to the Laslett's formula for tune shifts.

* Test with a simple 4-bend FODO structure

Resulting beam sizes for a simple 4-bend FODO structure



Zero beam current



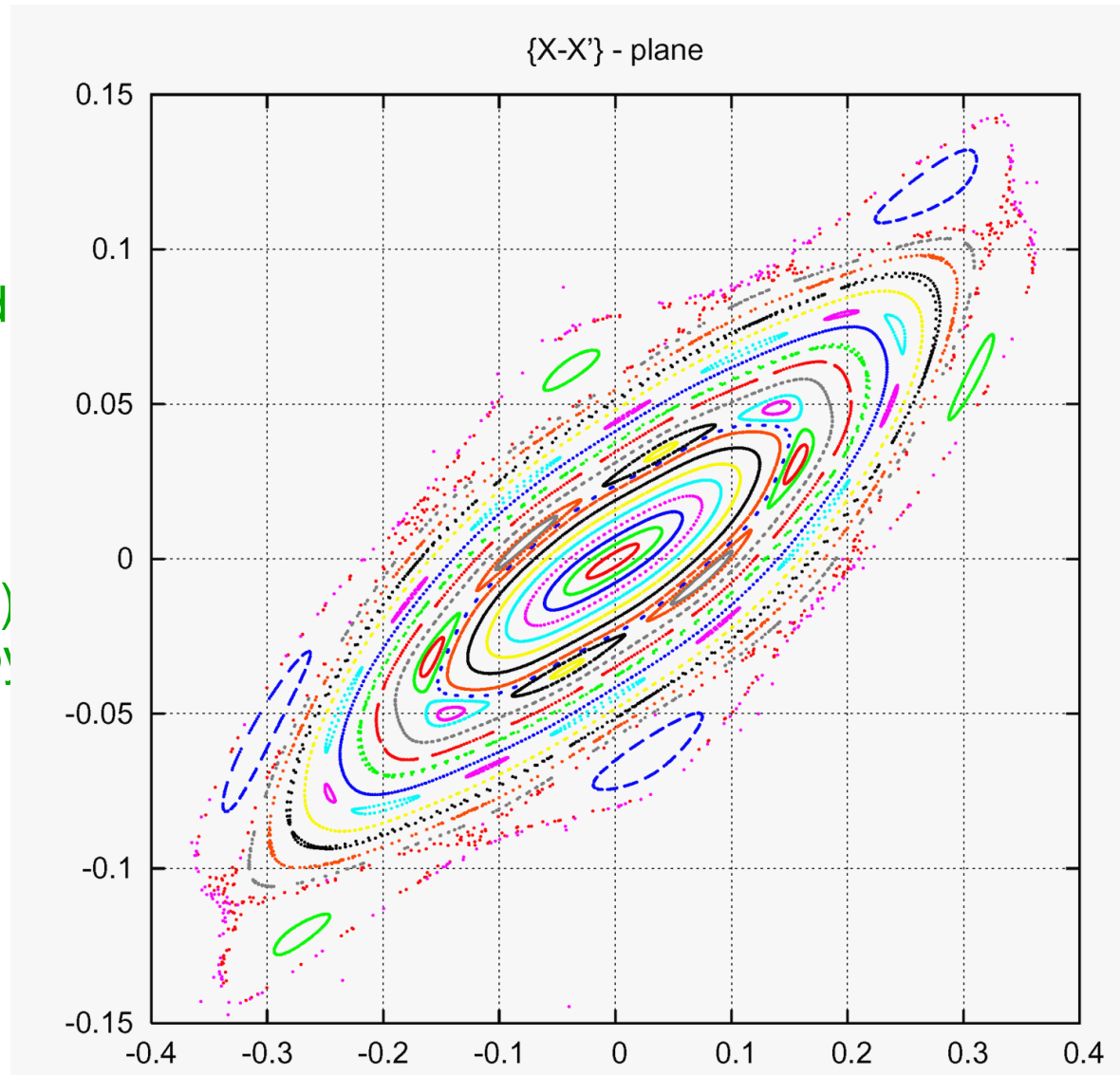
Non-zero beam current

The self-consistent horizontal beam size ax is increased essentially

Tracking with many BB

Example for a simple linear lattice

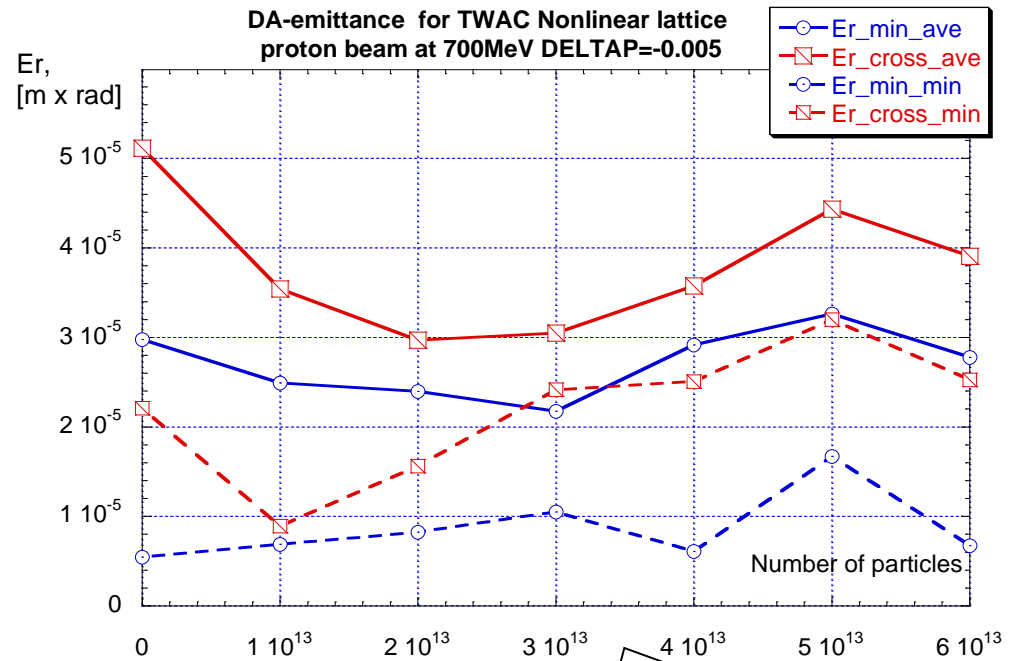
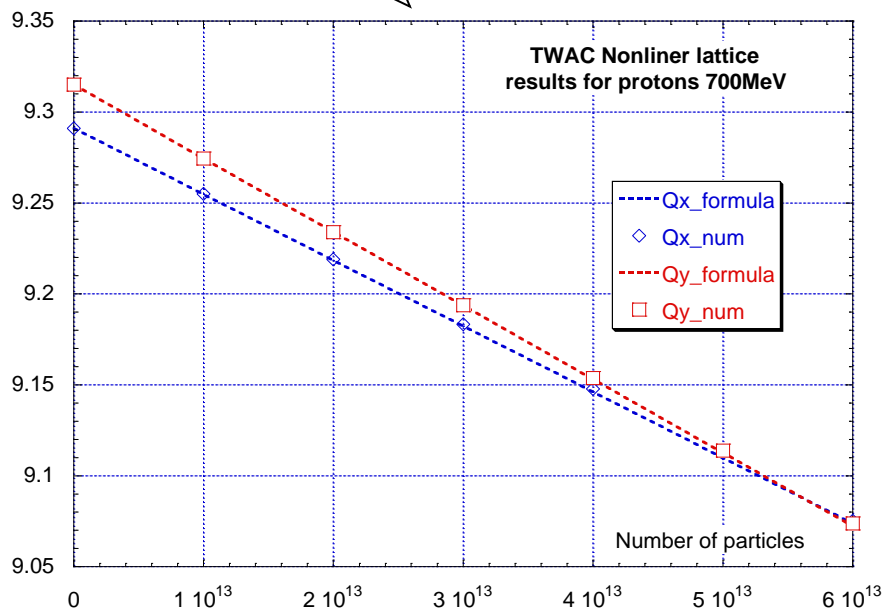
- S.C kicks by BB-elements for non-linear tracking; (C.O. shifts are included a total number BB-elements is not limited);
- thin-lens tracking with MADX (similar to MAD8) with lattice conversion by MAKETHIN command



Application to real ring – ITEP’s TWAC

4D simulations

Dependence of analytical (Laslett) and computed (MADX) betatron tunes on intensity.



Dependence of DA on the beam intensity for relative momentum offset DELTAP= -0.005 (simulations done for high-order nonlinearities in all magnets and essential fluctuations in “E-blocks”)

DA of SIS100 study (4D) following G.Franchetti-report (Micromap)

**Complicated work with
the “cold” lattice!**

- Syst. & rand. Errors in D&Q
- RMS closed orbit – 1.5/1.0 mm
- (+ octupoles for Landau damping and effect of sp.ch.)

MICROMAP:

a) COD=0:

central DA=4.05 with 3- σ (3.5-4.6)

b) COD=1.5/1.0mm:

central 3.3 with 3- σ (2.7-4.0)

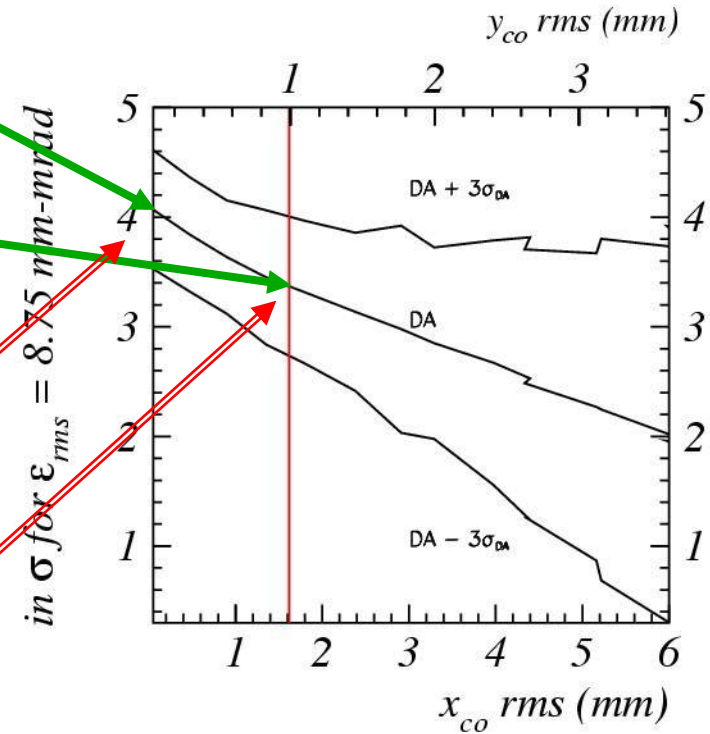
MADX:

a) COD=0:

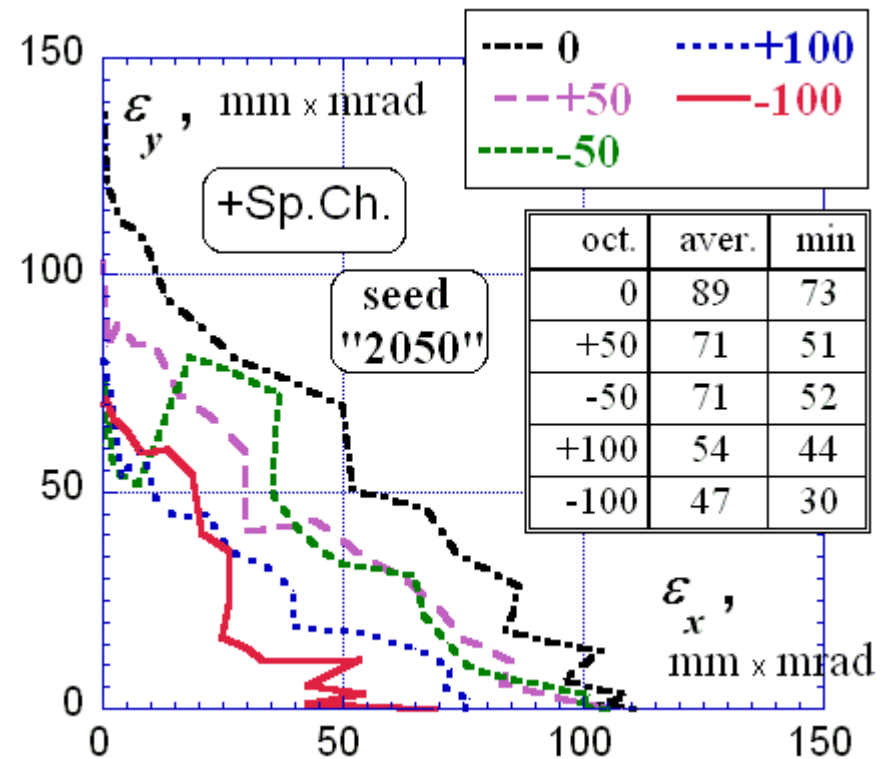
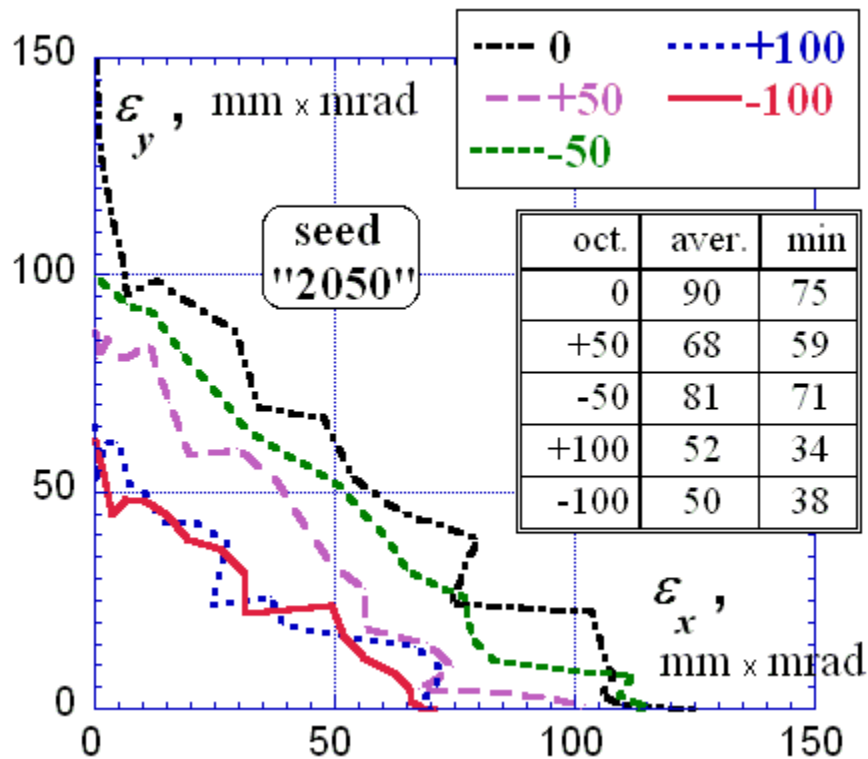
DA=> (3.8-4.0)

b) COD=1.5/1.0mm:

DA => (3.2-3.3)



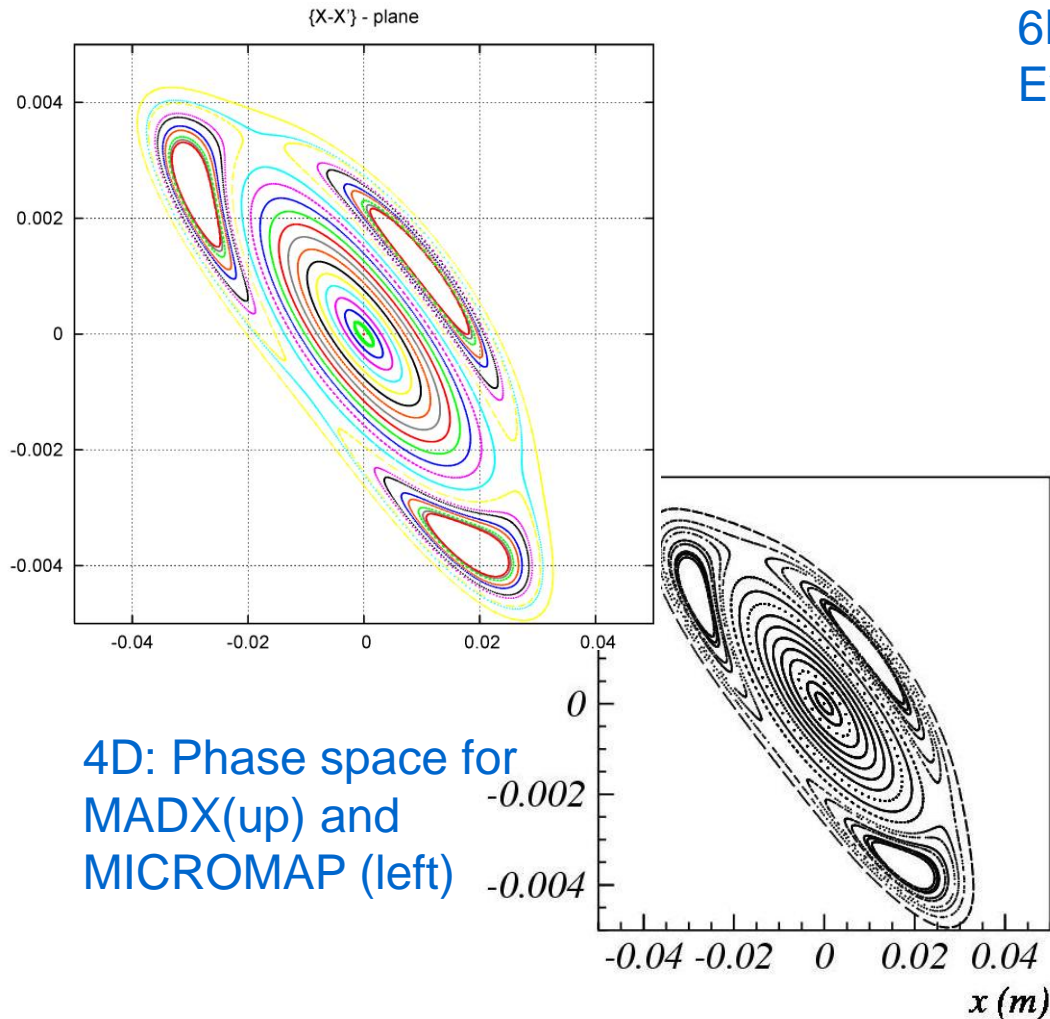
SIS100:Oct. for Landau damping & SC effect (4D)*



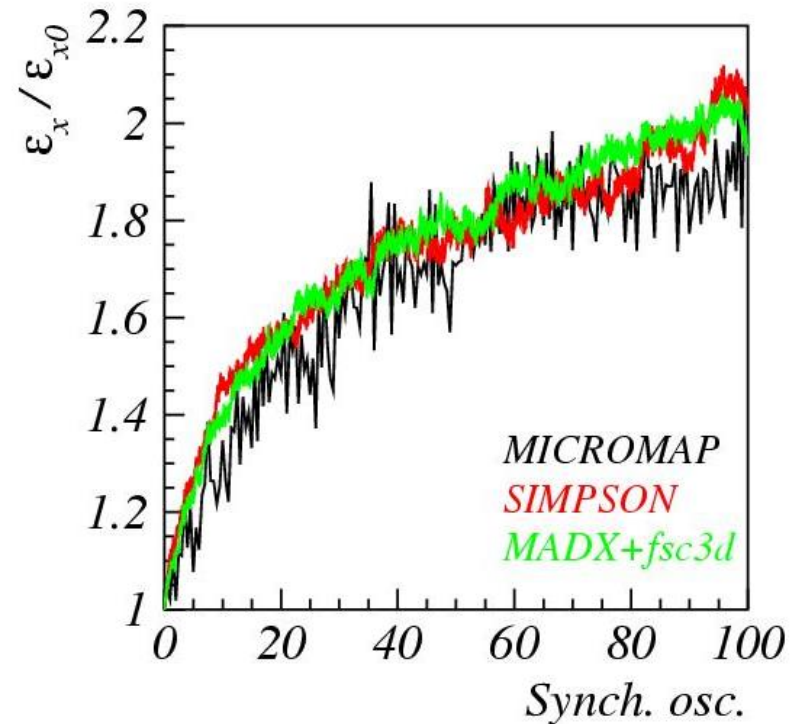
Boundary of the stability domain for the seed "2050"
 (the lattice with systematic errors and random errors and the nonzero c.o.)
without (left) and with (right) the space charge.

Benchmarking with SIS18-lattice

http://www-linux.gsi.de/~giuliano/research_activity/trapping_benchmarking/main.html



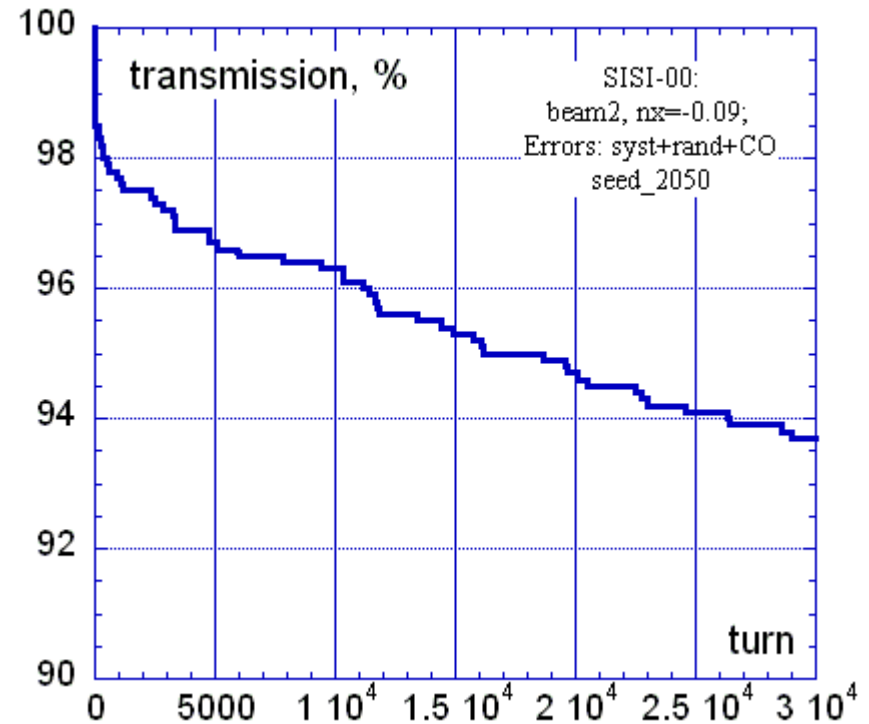
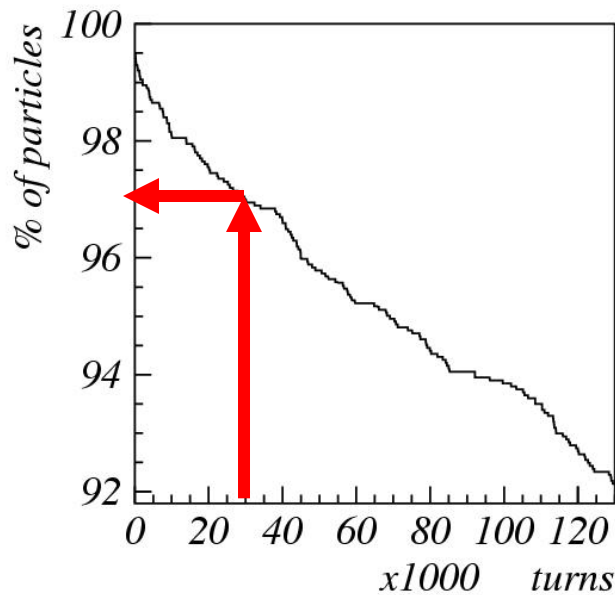
6D: Fake longitudinal motion (BM-9):
Evolution transv. Rms-emittance



Fake 6D Beam loss in SIS100 (GSI)

MADX-Lattice with 840 BBs for *Sp-charge*

"BEAM2" for JF (nu_x=-0.090)



CDR-2008: Fig.11 Beam loss with space charge for Beam2 (2k macroparticles; Micromap).

97% vs 93.7% (?)

Initial Mismatching CO + other diff. (seed, sp-ch-center).

Initial loss->1.5% => **97% vs 95.2%**

Real 6D tracking with transverse sp.ch. for FNAL Debuncher (already presented at SC-2013)

- Utilizing the **BB** elements to create a frozen space charge model but **adapting** emittances and **Twiss** parameters for the sigma determination (Y.Alexahin's algorithms).
- Using the **MAD-X Macro technique** for all operations
- **Phase-I: Lattice Preparation**
 - Splitting the elements (at least once) and introduce **SC** kicks in between
 - Making sure that the lattice is stable including **SC** and self-consistent (linearly)
 - Transferring thick lattices into thin ones (**symplectic and only way to track in MAD-X**)
- **Phase-II: Running**
 - Multi-runs of one-turn tracking (MADX runs lattices with constant parameters)
 - Time varying magnet strengths, phase trombone after every turn
 - Emittance and sigma recalculation after every turn via **TWISS**
 - Output (losses, emittances, etc)

MADX-script algorithm briefly

- In more details **for every turn**:
 - a) varying lattice parameters; b) collecting surviving particles and lost particles; c) calculating some “integral” beam parameters; d) reading and filling tables etc.
- Tool: **madx-script language** (C-like) with a quite poor set of commands (***no manuals, only a few examples***) => a time consuming code developments;
CPU memory consuming => run portions with only 25 turns (restarts with a simple external linux-script).
- The developed script has **all attributes of a simple particle tracking code**, which can be written and tested with a usual language like FORTRAN in a few weeks.
- It consists of main code with about **7 hundreds lines**, plus **macro file with 5 hundreds lines**, while the lattice file (MADX-sequence) prepared separately.

2012: new code MADX-SC*

- The **MACRO** technique of **MAD-X** is working fine but typically one ends up with **very complex logic** → Therefore quite **unpractical** for new applications.
- **Macros** are inherently slow!
- Therefore the idea was to reduce the use of **Macros** as much as possible and to modify the **code** to do most of the work directly within **MAD-X**.
- In detail:
 - Time varying multipoles via turn-by-turn **TFS** tables
 - Time varying phase trombone Time varying **RF cavity voltage**
 - Include the sigma and emittance determination after each turn
 - Allow for stop&go for the **MAD-X** tracking routine to allow intermediate **TWISS** calculation at start and the locations of the **SC** elements
 - Input via several **TFS** tables.

Since 2012 MADX-SC is developed & coordinated by F.Schmidt (while Phase-I is used in the old macro-based style)

* Schmidt & Kapin, "MADX-SC Flags description", CERN-ACC-NOTE-2013-006
D'Imperio, "Experience wit OpenMP for MDX-SC", CERN report No.xxx, 2014

Possible alternative for MAD-SC with YA-algorithm*

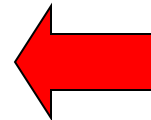
A typical run example for MADX-SC:

```
TRACK, onepass, recloss, APERTURE;  
CALL, FILE="start_1k_particles.dat";  
deltap_rms=0.000555; deltap_max=5*deltap_rms;  
  
RUN, turns=1, maxaper={ap_max,0,0,0,0,0},  
n_part_gain=1, sigma_z=9.59,  
track_harmon=2, deltap_rms=deltap_rms,  
deltap_max=deltap_max;  
  
n=1;  
while (n < 10)  
{  
twiss,table="spch_bb",file="spch_bb_twiss.dat";  
readtable, file="spch_bb_twiss.dat";  
RUN, turns=100, maxaper={ap_max,0,0,0,0,0};  
n = n + 1;  
};  
ENDTRACK;
```

New YA-algorithm potentially
will allow
to exclude TWISS calls.

TWISS calls can be used sometimes
for (diagnostics/benchmarking).

*It can be realized as FORTRAN
subroutines inside TRACK module*



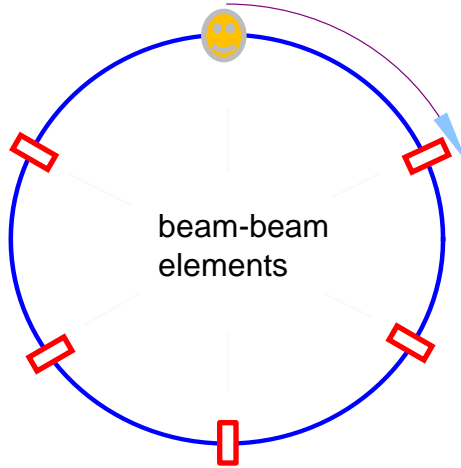
Very time –consuming TWISS calls,
which are necessary for some
applications after every turn

*read as Yuri Alexahin's algorithm

Courtesy Yuri Alexahin

Beam sizes from tracking data*

observation point,
 Σ_0 and/or ε_m



Present MADX-SC algorithm requires knowledge of β -functions =>
Turn-by-turn call of TWISS module (CPU-time consuming!);
In a rough approximation - fixed β -s can be used

$$\sigma_u^2 = \sum_m \beta_{um} \varepsilon_m, \quad u = x, y, z \quad m = 1, 2, 3$$

New YA-algorithm*: allow to avoid TWISS calls to

$$\Sigma_{(k)} = \mathbf{M}_{0 \rightarrow k} \cdot \Sigma_{(0)} \cdot \mathbf{M}_{0 \rightarrow k}^T$$

Tracking of sigma matrix throughout BB-elements using sigma matrix value at START Σ_0 and transfer matrices $M_{i \rightarrow j}$ between BB-elements, while $M_{i \rightarrow j}$ includes matrix for preceding BB $M_{\text{BB}(i)}$.

Such tracking does not require knowledge of Σ_k in a simple kick-rotate scheme (but does require $\Sigma_i, i=0, \dots, k-1$); =>

It requires calculation of Σ_0 at every turn with dedicated YA-algorithm !

* Y.Alexahin, "Computation of Eigen-Emittances (and Optics Functions!) from Tracking Data", SC-2013, Cern, Geneva; "Computing Eigen-Emittances from Tracking Data", MAP doc 4358-v-1, 2013

Calculation of Sigma-matrix at START

- suppress halo contribution to covariance matrix in a self-consistent way to obtain right sizes for the space charge forces computation
- Iterative procedure for nonlinear fit of particle distribution in the phase space with a **Gaussian smooth function**

Initial definition of covariance matrix (Σ - matrix) for a set of N particles

$$\Sigma_{i,j} = \frac{1}{N} \sum_{k=1}^N \zeta_i^{(k)} \zeta_j^{(k)}, \quad \zeta_i^{(k)} = z_i^{(k)} - \bar{z}_i, \quad \bar{z}_i = \frac{1}{N} \sum_{k=1}^N z_i^{(k)}, \quad i = 1, \dots, 6$$

Basic assumption: particle distribution is a function of quadratic form

$$\Phi(\underline{\zeta}) = (\underline{\zeta}, \Sigma^{-1} \underline{\zeta}) \equiv \sum_{i=1}^6 \zeta_i (\Sigma^{-1} \underline{\zeta})_i = \sum_{i,j=1}^6 \Sigma_{ij}^{-1} \zeta_i \zeta_j$$

Example of **simple iterative procedure** for Σ values with $\eta=1$ (eq.(41) or slide-7 in Refs*):

$$\Sigma_{ij} = \frac{1}{N} \sum_{k=1}^N \zeta_i^{(k)} \zeta_j^{(k)} \exp\left[-\frac{1}{2}(\underline{\zeta}^{(k)}, \Sigma^{-1} \underline{\zeta}^{(k)})\right] / \left\{ \frac{1}{N} \sum_{k=1}^N \exp\left[-\frac{1}{2}(\underline{\zeta}^{(k)}, \Sigma^{-1} \underline{\zeta}^{(k)})\right] - \frac{\eta^{-1}}{2^{n/2+1}} \right\} \quad (41)$$

For $n=6$ in all cases just 20-30 iterations are required to achieve precision $\leq 10^{-6}$, it takes *Mathematica* ~13 seconds with $N=104$ on my home PC. For a Fortran or C code it will be a fraction of a second.

Possibility: Fitting with two Gaussian distributions

Some distributions with essential halos can be fitted with two Gaussian distributions. using eqs. (41-43) in Ref*:

$$\Sigma_{ij} = \frac{1}{N} \sum_{k=1}^N \zeta_i^{(k)} \zeta_j^{(k)} \exp\left[-\frac{1}{2}(\underline{\zeta}^{(k)}, \Sigma^{-1} \underline{\zeta}^{(k)})\right] / \left\{ \frac{1}{N} \sum_{k=1}^N \exp\left[-\frac{1}{2}(\underline{\zeta}^{(k)}, \Sigma^{-1} \underline{\zeta}^{(k)})\right] - \frac{\eta^{-1}}{2^{n/2+1}} \right\} \quad (41)$$

$$\eta = \frac{2^{n/2}}{N} \sum_{k=1}^N \exp\left[-\frac{1}{2}(\underline{\zeta}^{(k)}, \Sigma^{-1} \underline{\zeta}^{(k)})\right] \quad (42)$$

1) the “main”(core) Gaussian distribution:

Σ -matrix is calculated from eq. (41) with $\eta=1$ and the fraction of particles in this “main” distribution from eq.(42)

2) to obtain the “halo” Gaussian distribution subtract the “main”(core) Gaussian distribution

$$F(\underline{\zeta}) = \frac{\eta}{(2\pi)^{n/2} \sqrt{\det \Sigma}} \exp\left[-\frac{1}{2}(\underline{\zeta}, \Sigma^{-1} \underline{\zeta})\right] \quad (34)$$

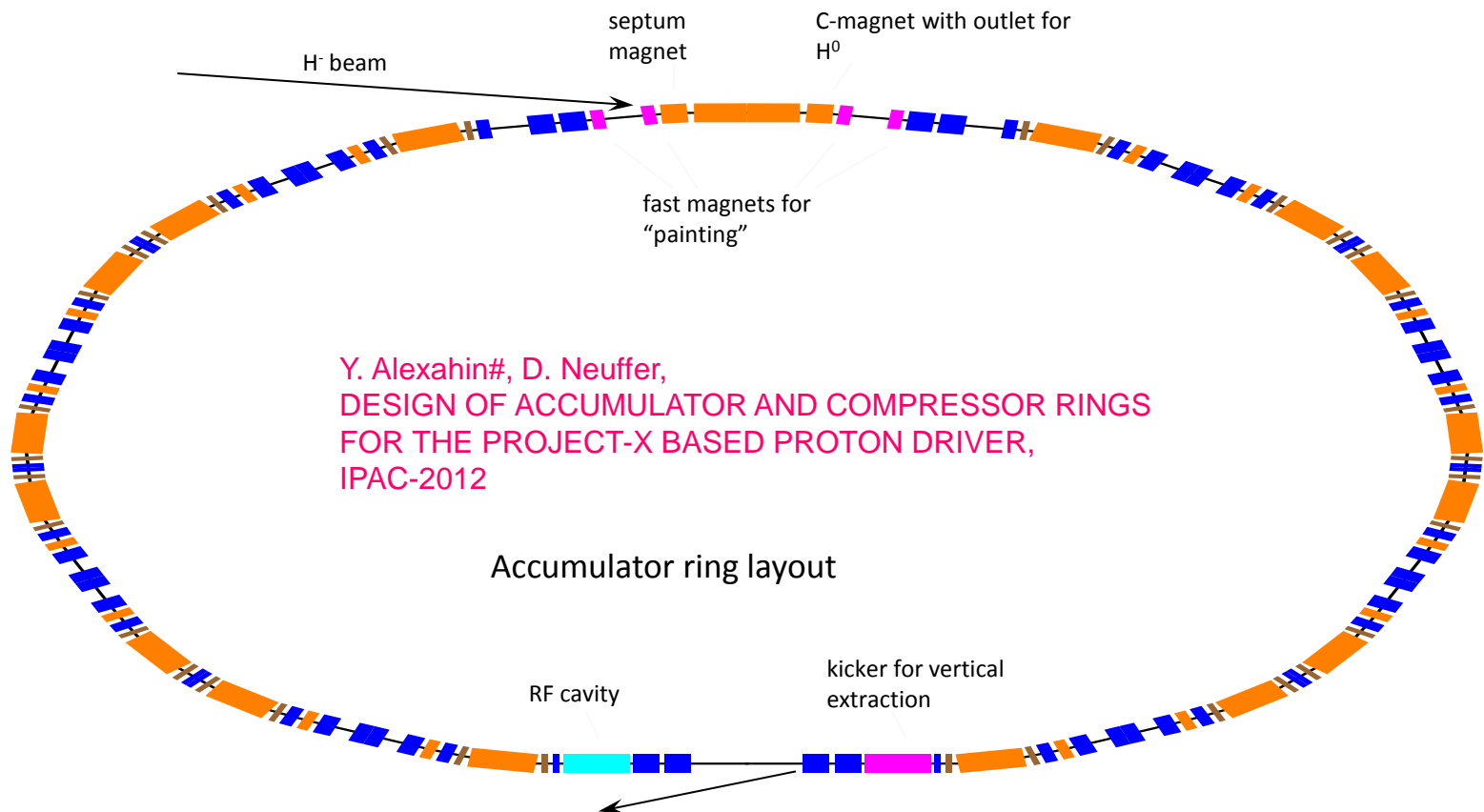
from original one and repeat the above procedure for remaining “halo” particles
(*details yet to be developed*)

New task: Multi-turn injection

Examples of multi-turn injection of H-minus:

a) FNAL booster;

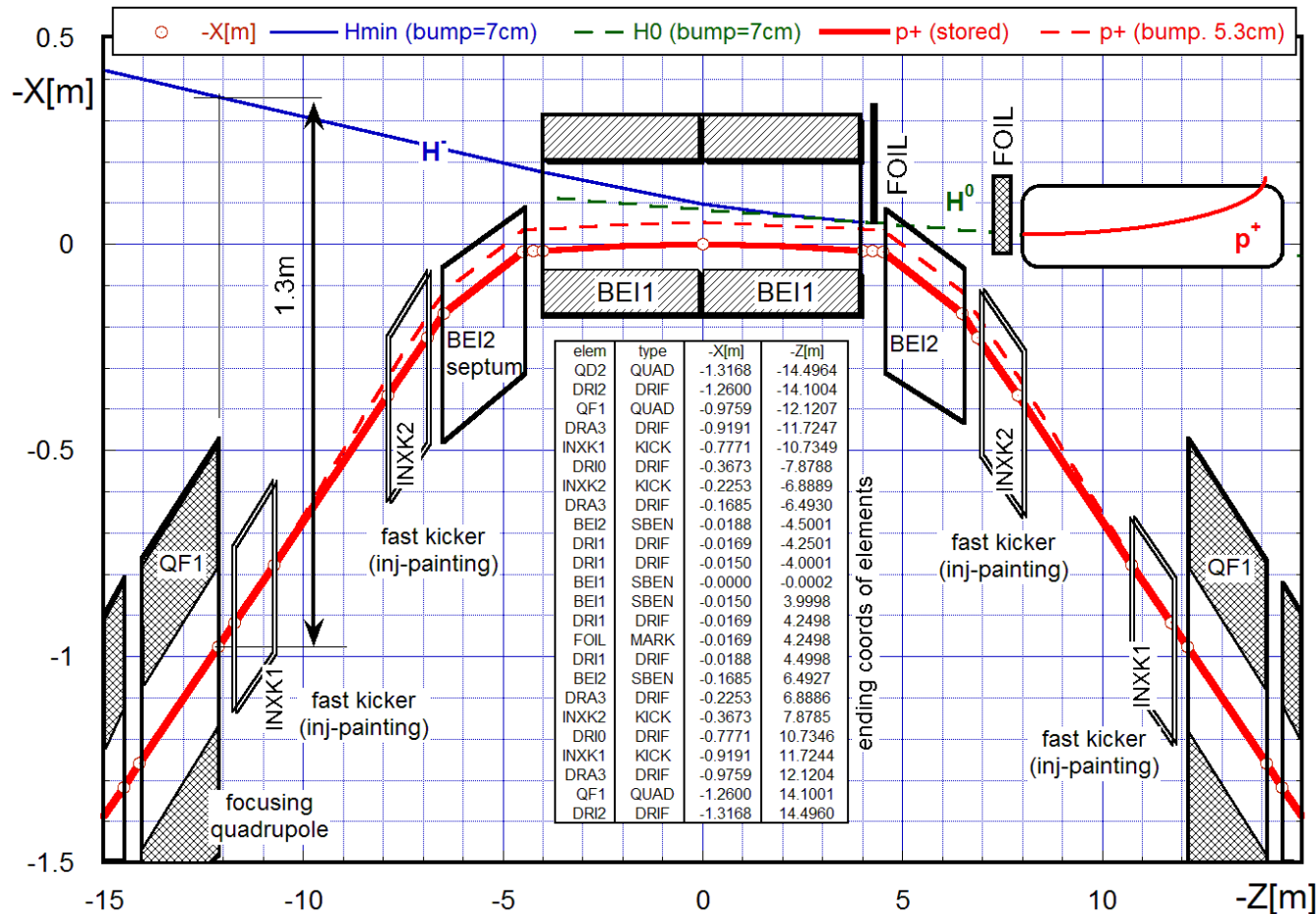
b) Accumulator Ring (AR) for a proton driver (Muon Collider & Neutrino Factory)



Detailed scheme for AR

New coding tasks:

- Injection painting with programmable fast kickers (= thin-lens multipoles ?)
- Adding additional “beamlets” at foil and “randomly-born” protons in vicinity of a strong bend BE2 ($H^0 \rightarrow p$); BB's on movable kicked orbit
- Large aperture week magnet $2 \cdot BE1 \Rightarrow$ detailed tracking with fringe-field maps



External-maps for large magnets (fringe-fields)

Example: **1.5 TeV (c.o.m.) Muon Collider (MC)** : very small values of β -function at IP $\beta^* \leq 1$ cm; distance from IP to the first Quad $> \sim 6$ m (for detector protection). \Rightarrow β -function at Quads $\rightarrow 100$ km \Rightarrow beam dynamics very sensitive to errors.

The effects on dynamic aperture (DA) of fringe-fields (FF) in both Dipoles & Quads. of MC.

FF effects are important for the MC ring as well (very large β -function; large magnet aperture $R_{ap} \sim 40$ -80mm; short length of quadrupoles < 2 m).

MAD-X PTC module treats FF effects in quadrupoles in "hard-edge" approximation

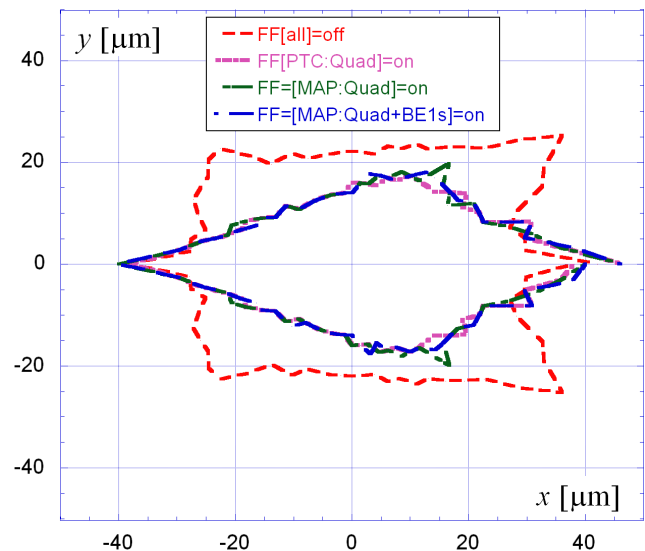
Exporting Magnet Maps from COSY has been realized with PTC-TRACK:

- ❖ to extend simulation capabilities beyond the hard-edge approximation
- ❖ usage of realistic FF falloffs generated by COSY INFINITY

The 1000-turn DA was calculated for 4 FF options:

- 1) no FF;
- 2) PTC FF in quadrupoles;
- 3) COSY FF maps in quadrupoles;
- 4) COSY FF maps in both quadrupoles and IR dipoles.

FF results in a reduction of the stable area ($\sim 45\%$).
The difference between the results obtained with PTC hard-edge approximation and COSY maps is quite small.



Wake-fields & longitudinal SC-forces

The longitudinal SC force can be introduced in MADX-SC:

- 1) approach of frozen Gaussian longitudinal distribution;*
- 2) Make bins for tracking particles in the longitudinal direction => 1D charge density => Ez component*

Borrow the wake fields calculations from “MAD-8 with acceleration”:*

- 1) Fortran subroutines are available at SLAC;*
- 2) Wake-fields for round beam must be submitted by user*
- 3) Subroutines were reviewed, debugged and detailed description of variables involved is under the way*

```
label: LCAVITY, TYPE=name, L=real, DELTAE=real, PHI0=real,  
FREQ=real,ELOSS=real, APERTURE=real, E0=real, VOLTERR=real,  
LAGERR=real,NBIN=integer, BINMAX=real,  
LFILE=string, TFILE=string
```

** H. Grote, E. Keil (CERN), T.O. Raubenheimer, M. Woodley (SLAC),
“EXTENSION OF MAD VERSION 8 TO INCLUDE BEAM
ACCELERATION”, EPAC-200*

Thin-track with energy variation (?)

- Thin-track use few elements: multipoles, cavities, BBs, ... =>
Remember and apply **energy scaling laws for few elements**
(use existing codes, e.g. STRUCT, LOCO at FNAL
with MADX-lattice)
- **New YA-algorithm** allows to **avoid TWISS calls**
(communication with functions in C-core)
- **Disconnect THINTRACK from C-code database:**
use database only before the first turn to collect all relevant
parameters and then create **own database** for variable
parameters with clear and transparent programming (f90-module)
-

Conclusions

- MADX-SC can be applied for S.C. simulations for both coasting and bunching beams
- 4D&6D benchmarking tests showed good agreements with other codes.
- Applications: beam losses and DA calculations in lattices with magnets errors and closed orbits, while lattices with variable parameters can be also used
- The code can be further extended via implementations for new simulation tasks.