Improving Synchrotron Radiation in MADX

Guillaume Simon (IJCLab – CERN), Angeles Faus-Golfe (IJCLab) & Riccardo De Maria (CERN).

Outline

1) Synchrotron Radiation and tapering : FCCee issues.

2) Realistic simulation of SR : implementations in MADX

3) Preliminary Results

4) Summary

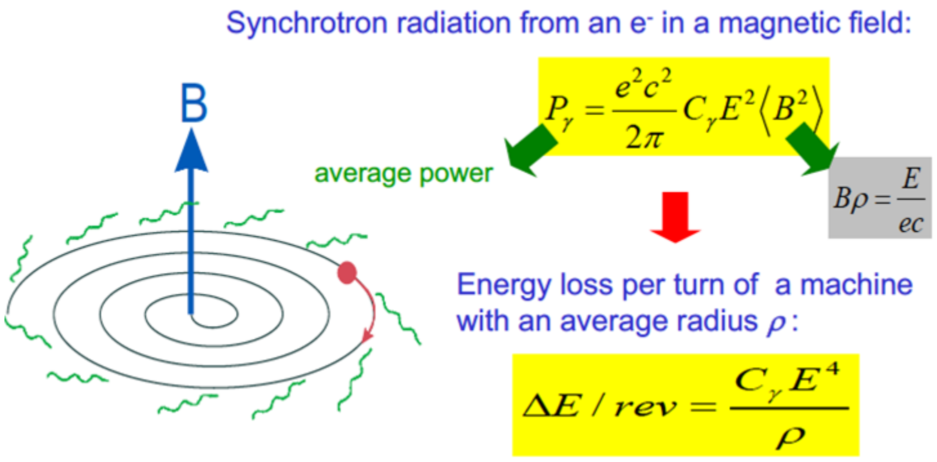
Synchrotron Radiation

Synchrotron radiation is an electromagnetic radiation emitted when relativistic charged particles are subject to an acceleration perpendicular to their velocity (**a** ⊥ **v**).

At high energy, synchrotron radiation losses lead to local deviation from the nominal energy. These deviations cause orbit offsets and combined with the gain of energy in the RF cavities, create a sawtooth effect and optics distortions.

The energy loss due to SR is proportional to $\frac{E^4}{\rho}$. To limit it we need to increase the circumference of the ring. That's why at high energies, high circumferences are needed.

For the FCCee, the energies aimed are high, so the SR is a huge issue. The average power loss has been fixed at 50MW, so the circumference is fixed to meet this power loss. Hence the high circumference of the FCCee.



Energy loss per turn has to be be replaced by the RF system, which is the major cost factor for a collider.

The Sawtooth effect

Energy loss due to SR and energy gain in RF cavities lead to what's call the "sawtooth effect". This sawtooth effects occurs both for the energy of the beam (a succession of loss and gain of energy), but also for the orbit of the particle (a succession of deviations and corrections of the orbit).

Because of the high energies involved in the FCCee and its large circumference. The sawtooth effect can't be neglected. Also, the loss of energy isn't the same at the IP, which is also why the sawtooth effect has to be corrected.

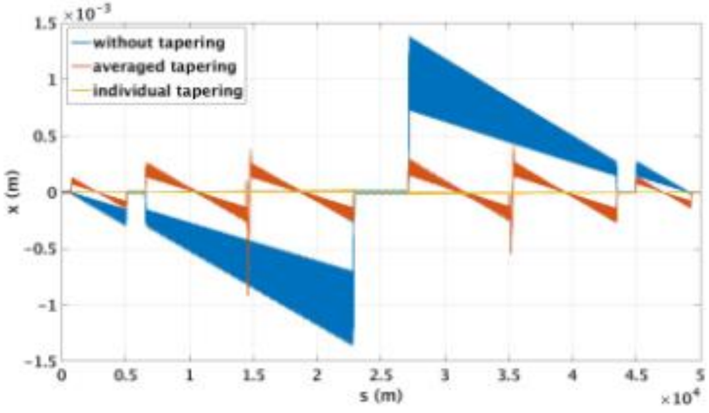


Figure 7: Same scale comparison of the orbit without tapering (blue), with individual tapering (yellow) and with averaged tapering (orange).

Example of sawtooth effect.

Credits to :B. Härer, A. Doblhammer, and B.J. Holzer, "Tapering Options and Emittance Fine Tuning for the FCC-ee Collider", in Proc. 7th Int. Particle Accelerator Conf. (IPAC'16), Busan, Korea, May 2016, paper THPOR003, pp. 3767-3770, doi:10.18429/JACoW-IPAC2016-THPOR003

Tapering

To correct the orbit offset due to energy loss by SR, we can adjust the dipoles strength's k factor to the local beam energy. This is called "dipole tapering".

$$k = \frac{\alpha}{l} = \frac{1}{\rho} = \frac{e}{\beta E} B$$

k : dipole strength factor
α : bending angle
l : dipole's length
ρ : bending radius
e : electric charge
β : relativistic β
E : total energy
B : Magnetic field

There are two ways to optimize the dipoles' strength :

1) Individual tapering for each dipole thanks to an individual mechanic system. But for a machine the size of the FCCee, it is expensive.
2) Depending on how large the orbit offset is acceptable, families of dipoles can be given an "average tapering strength".

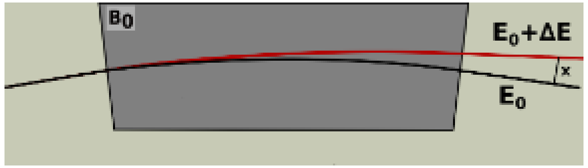


Figure 1: Dipole before tapering. A particle with an energy deviation ΔE is forced away from the ideal orbit.

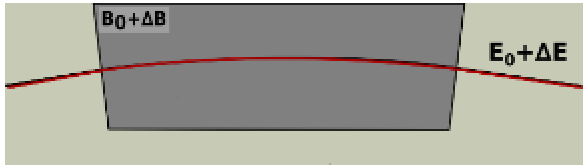


Figure 2: Dipole after tapering. A particle with an energy deviation ΔE now moves on the ideal orbit.

Credits to :B. Härer, A. Doblhammer, and B.J. Holzer, "Tapering Options and Emittance Fine Tuning for the FCC-ee Collider", in Proc. 7th Int. Particle Accelerator Conf. (IPAC'16), Busan, Korea, May 2016, paper THPOR003, pp. 3767-3770, doi:10.18429/JACoW-IPAC2016-THPOR003

MADX

Methodical Accelerator Design – X is a project with a long history, aiming to be at the forefront of computational physics in the field of particle accelerator design and simulation. Its scripting language is *de facto* the standard to describe particle accelerators, simulate beam dynamics and optimize beam optics at CERN.

MADX evolves thanks to dynamical programming. The software is constantly updated by teams of physicists. Up until a few years ago, MADX was adapted to the protons, now we update it to the physics of the FCCee involving electrons and positons.

MADX is now up to the version 5.08.01.

Tapering in MADX

In MADX the tapering is calculated as follow :

1) First we calculate pt at the entrance of the element, with : $p_t = \frac{E - E_0}{P_0 c}$

E is the total energy of the particle, E_0 is the energy of the reference particle and P_0 is the reference of the momentum particle.

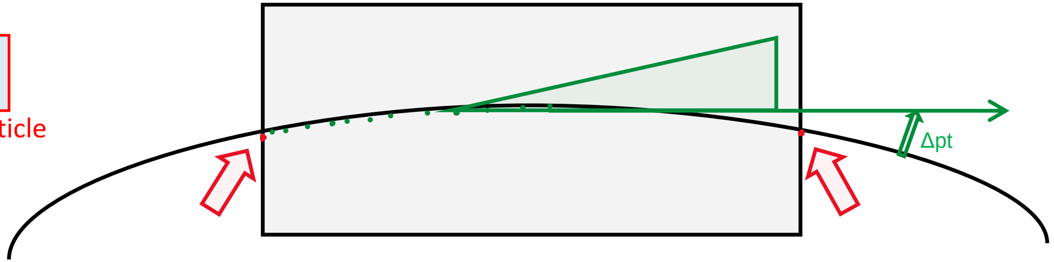
2) Then we track the particle through the element with the radiation.

3) We record the pt at the exit of the element.

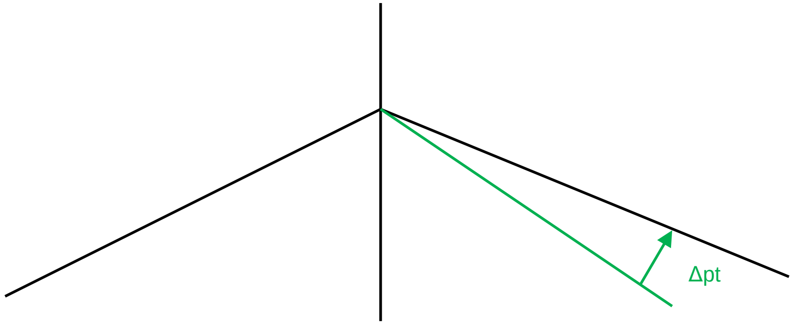
4) In the end we scale the dipole's strength by the average pt calculated to correct the offset of the particle's orbit.

Note that this is done for thousands of magnets or all magnets simultaneously, given the fact that in FCCee there are thousands and thousands of magnets, doing it for all magnets individually would be very expensive.

This model works both for thick and thin lattices.
N.B : MAD-X cannot taper untapered thin lattices, but tapering information is transfered to a thin lattice using the module MAKETHIN.



Scheme for thick lattice



Scheme for thin lattice

Implementation in MADX

$$k_{new} = \frac{k}{1-p_t}$$ Not correct

$$k_{new} = k(1+p_t)$$ Correct

- For the tapering, MADX does the following approximation :
- $R_{ij}(k_{new}, p_t = a) = R_{ij}(k_{new}, p_t = 0) + T_{ijk}(k_{new}, p_t = 0) * a$
- This equation can be rewritten as :
- $M_{ij}(Z_c) = M_{ij}(0) + \Sigma_k T_{ijk} Z_k$
- with Z_c a coordinate of the closed orbit, Z_k a coordinate, M_{ij} a transfer matrix of an element and T_{ijk} the matrix of second order terms.
- $\Sigma_k T_{ijk} Z_k$ corresponds to « sum 1 » in the code on the right.

- With tapering, we want that :
- $R_{ij}(k_{new}, p_t) = R_{ij}(k, p_t = 0)$
- This equation can be rewritten as :
- $Z_i = K_i + \Sigma_j M_{ij} X_j + \Sigma_{jk} T_{ijk} X_j X_k = K_i + [\Sigma_j (\mathrm{M_{ij}} + \Sigma_k T_{ijk} X_k) X_j]$
- With Z_i the final transfer map, K_i a constant and X a coordinate. This equation correspond to « sum 2 » in the code on the right.

- This map corresponds to a tracking map, and in order to have better results with tapering, we must find a way to add an higher order term to the last equation.

```
SUBROUTINE tmtrak(ek,re,te,orb1,orb2)
  use math_constfi, only : zero
  implicit none
  !----------------------------------------------------------------------*
  !     Purpose:                                                         *
  !     Track orbit and change reference for RE matrix.                  *
  !     Input:                                                           *
  !     ek(6)      (double)  kick on orbit.                              *
  !     re(6,6)    (double)  transfer matrix before update.              *
  !     te(6,6,6)  (double)  second order terms.                         *
  !     orb1(6)    (double)  orbit before element.                       *
  !     Output:                                                          *
  !     orb2(6)    (double)  orbit after element.                        *
  !     re(6,6)    (double)  transfer matrix after update.               *
  !----------------------------------------------------------------------*
  double precision, intent(IN)     :: ek(6), te(6,6,6), orb1(6)
  double precision, intent(IN OUT) :: re(6,6)
  double precision, intent(OUT)    :: orb2(6)

  integer :: i, k, l
  double precision :: sum1, sum2, temp(6)

  integer, external :: get_option

  do i = 1, 6
     sum2 = ek(i)
     do k = 1, 6
        sum1 = zero
        do l = 1, 6
           sum1 = sum1 + te(i,k,l) * orb1(l)
        enddo
        sum2 = sum2 + (re(i,k) + sum1) * orb1(k)
        re(i,k) = re(i,k) + sum1 + sum1
     enddo
     temp(i) = sum2
  enddo
```

$\Sigma_k T_{ijk} Z_k$

$Z_i = K_i + [\Sigma_j (\mathrm{M_{ij}} + \Sigma_k T_{ijk} X_k) X_j]$

Preliminary Results

TWISS : The TWISS command calculates the linear lattice functions and optionally the chromatic functions. The linear lattice functions are analytically calculated.

Norad : Beam simulated without any radiations nor losses due to said radiations.

Rad : Beam simulated with radiations and the losses linked to the radiations.

Exact : If this is used the dirft is expanded around the actual closed orbit instead of the ideal orbit.

Taper : TAPER calculates the adjustment to the strengths of elements to account for small momentum variations through RF cavities or synchrotron radiation.

	5.07.00	5.08.01	(5.0x)
rbend/sbend	k0->k0*(1+pt/beta0)	k=k(1+ktap) Ktap=pt/beta0	K0=k0(1+ktap) Ktap=pt/beta0
Twiss FCC notaper (rad)	402.22401 394.36428	402.30484 394.39673	402.19001 394.28686
Twiss FCC notaper (norad)	402.22400 394.36000	402.22400 394.36000	402.22400 394.36000
Twiss FCC taper (exact)	x	x	402.22389 394.35931
Twiss FCC taper (no exact)	402.23424 394.40915	402.21536 394.35092	402.20084 394.30421

Preliminary Results

TRACK : The TRACK command initiates trajectory tracking.

DYNAP : The DYNAP command calculates tunes, tune footprints from tracking data. DYNAP tracks two close-by particles over a selected number of turns (minimum 64 and maximum 1024), from which it obtains the betatron tunes with error. Many such companion particle-pairs can be tracked at the same time, which speeds up the calculation.

	5.07.00	5.08.01	(5.0x)
Track Dynap FCC rad (unknown)	1,2893.10^-11	6,56737.10^-10	0,358.10^-12
Track Dynap FCC norad	0.223874	0.223874	0.223874
Track Dynap FCC rad notaper	0.333333	~0	~0
Track Dynap FCC rad taper	x	1,42269.10^-10	~0
Track Dynap FCC rad taper no exact	1,2893.10^-11	6,56737.10^-10	~0
Track FCC rad notaper	0.333166	0.083499	0.251666
Track FCC rad taper exact	x	0.256666	0.245833
Track FCC rad taper no exact	0.480166	0.256666	0.245833
Track FCC rad (unknown)	0.372166	0.256666	0.245833
Track FCC norad	0.223666	0.223833	0.223666

Summary

- Tapering aim to correct the orbit offset du to synchrotron radiaton.
- Was implemented in MADX 5.07.00 but with a approximated equation giving good tune results in Twiss because it matches Twiss approximations but supposedly worse tracking results.
- The equation actually implemented in MADX 5.08.01 is correct but gives worse tune results in Twiss because of the lack of higher order terms in Twiss calculation.
- Next : Implementing the higher order terms in Twiss to get more accurate results.

Thank you !

Special thanks to Léon Van Riesen-Haupt for his help.

Annexe

- With tapering, we want that : $R_{ij}(k_{new}, p_t) = R_{ij}(k, p_t = 0)$, [eq 1],

with R a transfer matrix of an element.

- But MADX does the following approximation : $R_{ij}(k_{new}, p_t = a) = R_{ij}(k_{new}, p_t = 0) + T_{ijk}(k_{new}, p_t = 0) * a$, [eq 2],

where T is the matrix of the second order terms of R.

- For example, in a thin quadrupole, assuming $\beta_0 = 0$, we know that : $R_{21}(k, 0) = -k$ and $T_{216}(\mathrm{k}, 0) = +\mathrm{k}$,

so we obtain the following equation : $-k_{new} + k_{new} * p_t = -k \Rightarrow k_{new} = \frac{k}{1-p_t}$, [eq 3],

which works well in MADX 5.07.00. But the equation 2 is an approximation and the fact that we have good results (especially for tuning) is a coincidence.

- The correct equation being : $R_{21}(k, p_t) = \frac{k}{1+p_t}$, we obtain for the equation 1 :
- $\frac{k_{new}}{1+p_t} = k \Rightarrow k_{new} = k(1 + p_t) = \frac{k}{1-p_t+p_t^2-p_t^3 \ldots}$, [eq 4]

Showing that the equation 3 is an approximation of the equation 4. The equation 4 is the one actually implemented in MADX 5.08.01, while being the correct one mathematically and giving better results at tracking, it gives worse results for the tunes.