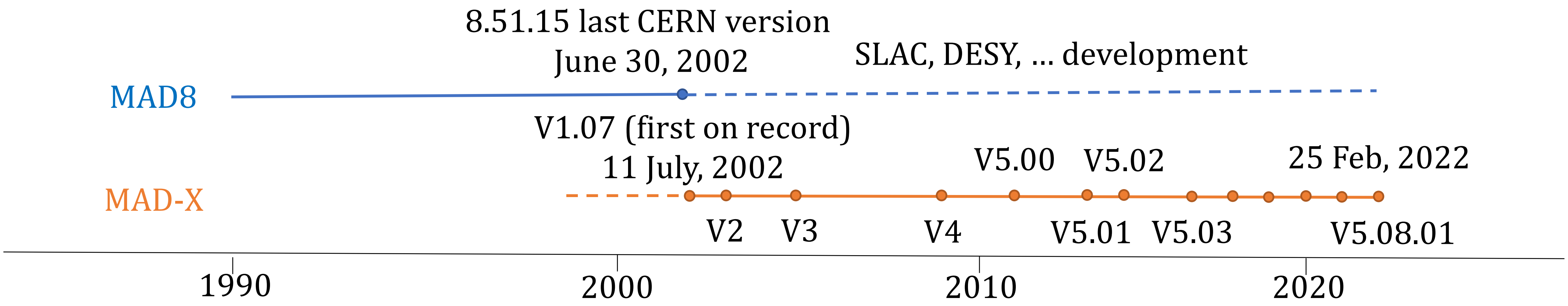
MAD-X status and progress: June 2022

R. De Maria for the MAD-X team

Thanks to Helmut Burkhardt, Tessa Charles, Laurent Deniau , Joshua Dilly, Gianni Iadarola, Jacqueline Keintzel, Andrea Latina, Tobias Persson, Ghislain Roy, Piotr Skoworonski, Frank Schmidt, Rogelio Tomas (CERN), Thomas Glasse (HIT), Scot Berg(BNL), Angeles Faus-Golfe, Guillaume Simon (CERN, CNRS-IN2P3-UPSaclay), Felix Carlier, Leon Van Riesen-Haupt (CERN, EPFL), Tatiana Pieloni (EPFL), CHART program

Timeline



MAD-X development was started by H. Grote by re-using MAD8 (Fortran) code and wrapping in a C shell.

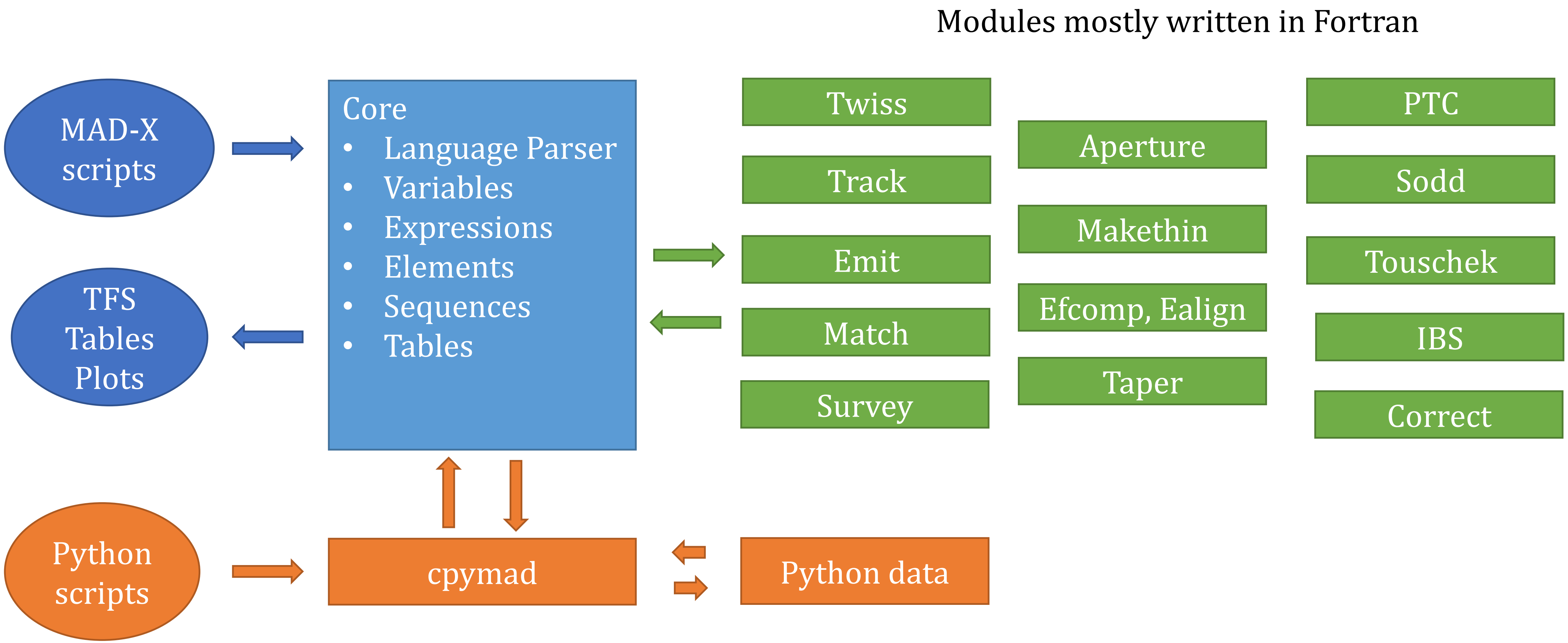
MAD-X development was then coordinated by F. Schmidt, L. Deniau, T. Persson at CERN and, from 2022, by R. De Maria.

MAD-X collected contributions from dozens of accelerator physicists and computer scientists from CERN and other laboratories.

Home web page: https://cern.ch/madx

Development: https://github.com/MethodicalAcceleratorDesign/MAD-X

Code structure



MAD-X example

MAD-X Language

```
fodo: sequence,  l    =8.4;
ba:   sbend,     a
qf:   quadrupole,k
bb:   sbend,     a
qd:   quadrupole,k
rf:   rfcavity,  v
      harmon=20, l
endsequence;

beam, particle=pos
use, sequence=fodo

kqf= 0.8;
kqd:=-kqf;
vrf= 1;

twiss;

value, qf->k1;
value, qd->k1;
value, table(summm,q1);
value, table(summ,q2);
```

Python

```
# to install do: pip install cpymad
from cpymad.madx import Madx

madx = Madx()
madx.input("""
fodo: sequence,  l   =8.4;
```

```
print(madx.elements.qf.k1)
print(madx.elements.qd.k1)
print(tt.summary.q1)
print(tt.summary.q2)
```

MAD-X language

+ fast element manipulation
+ re-use vast existing scripts
+ concise language
- parser is brittle
- often silent errors, crashes
- limited control flow and data structures

Python

+ well known, flexible and robust
+ easy to run multiple instance of MAD-X
+ seamless integration with the vast Python scientific ecosystem
- slow element manipulation for large machines
- little more verbose

Python interface more verbose but very close to the MAD-X syntax and much more flexible

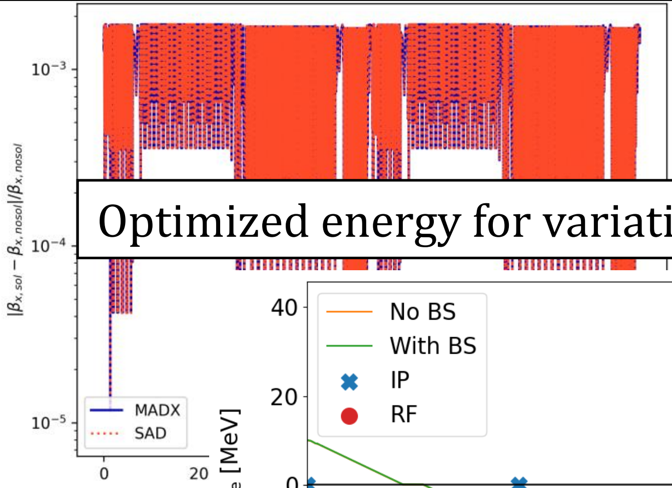
MAD-X for FCC

MAD-X is used for all circular accelerators in the CERN complex as well as for linacs.

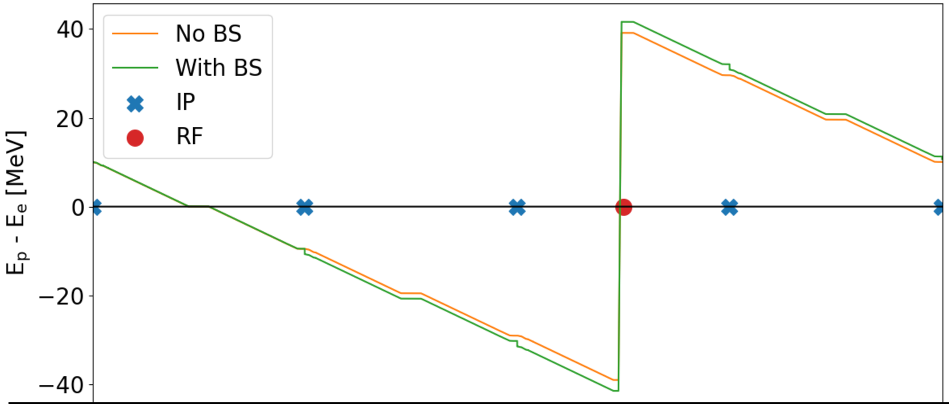
For FCC, MAD-X should be able to calculated:

- Closed orbit with energy loss with/without tapering (see TWISS, TAPER, CORRECT)
- Undamped lattice functions (TWISS)
- Damping times, equilibrium emittance (EMIT)
- Tracking with energy (without damping, with damping, with quantum excitation
- Build PTC universe and run PTC physics: normal forms, spin (PTC)

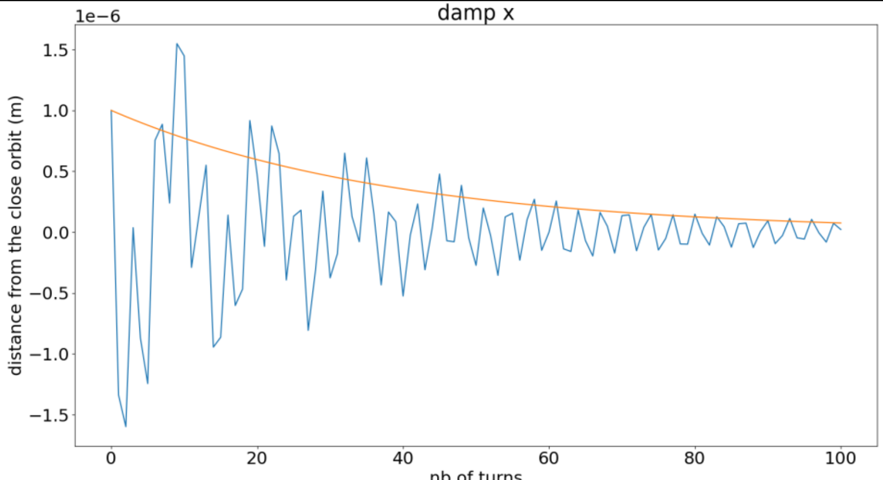
Solenoid induced beta-beating [L. Van Riesen-Haupt]



Optimized energy for variation for Z-model [J. Keintsel]



Damping tines EMIT vs TRACK [G. Simon]



Status and plan

We are currently:

1) Reviewing TWISS, EMIT and TRACK as we observed some inconsistent results in particular involving Solenoid and Multipoles

2) At the same time, update and release the MAD-X physics manual, based on the unpublished MAD-8 physics manual. The goal is to (re-)derive and document equations from first principles. It is of critical importance to enforce consistency and correctness throughout the code.

3) Extending test suite with test cases for which the reference is not an old result but an analytical estimate.

4) Extend tapering functionality to include octupoles and multipoles with an unified approach.

MAD-X Variables and Hamiltonian

MAD-X uses canonical coordinates using MAD-8 conventions [see reference in back-up]

Unscaled coordinates

$$P_x = \partial_{\dot{x}} L = m\gamma\dot{x} + qA_x$$
$$P_y = \partial_{\dot{y}} L = m\gamma\dot{y} + qA_y$$
$$P_s = \partial_{\dot{s}} L = (1+hx)^2 m\gamma\dot{s} + (1+hx)qA_s$$

MAD-X coordinates scaled by $P_s = m_0 c\beta_s\gamma_s = P_0(1+\delta_s)$

$$p_x = \frac{P_x}{P_s} \quad p_y = \frac{P_y}{P_s} \quad p_t = \frac{E - E_s}{P_s c}$$
$$t = \frac{1+\eta\delta_s}{\beta_s} s - cT \qquad a_{x,y,s} = \frac{q}{P_0} A_{x,y,s}$$

P_0 =PC in the BEAM definition. δ_s = DELTAP in TWISS command.

$$H = \frac{1+\eta\delta_s}{\beta_s} p_t - (1+hx)\left(\sqrt{p_t^2 - \frac{2p_t}{\beta_s} + 1 - \left(p_x - \frac{a_x}{1+\delta_s}\right)^2 - \left(p_y - \frac{a_y}{1+\delta_s}\right)^2} + \frac{a_s}{1+\delta_s}\right)$$

P_0 is used to set the magnetic fields proportional to the design momentum.

δ_s introduces a momentum error by forcing the revolution period: $t_\mathrm{f} = t_\mathrm{i} \rightarrow T_\mathrm{rev} = \frac{L_\mathrm{ring}(1+\eta)\delta_s}{c\beta_s}$.

δ_s is not, in general, $\delta_0 = \frac{P-P_0}{P_0}$ nor $\delta = \frac{P-P_s}{P_s} = \sqrt{p_t + \frac{2p_t}{\beta_s} + 1} - 1$, but controls the energy deviation of the closed orbit.

MAD-X variables: consequences

Momentun scaling $P_s = m_0 c \beta_s \gamma_s = P_0(1+\delta_s)$

DELTAP in TWISS command

PC in BEAM command

Coordinates scaled by P_s $\quad p_x = \frac{P_x}{P_s} \quad p_y = \frac{P_y}{P_s} \quad p_t = \frac{E - E_s}{P_s c} \quad t = \frac{1+\eta\delta_s}{\beta_s} s - cT$

Fields scaled by P_0 $\quad a_{x,y,s} = \frac{q}{P_0} A_{x,y,s} \quad B_y(x,y) - iB_x(x,y) = -\frac{q}{P_0} \sum \frac{k_n^N + i k_n^S}{n!}(x+iy)^n$

<u>Implications</u>:

1. For any δ_s, on the closed orbit $\delta_0 = \frac{P - P_0}{P_0} \approx \delta_s$ and $\delta \approx 0, p_t \approx 0, t \approx 0$, good to keep approximations in p_t, t small.

2. In general $\frac{dx}{ds} = \frac{(1+h\,x)\left(p_x - \frac{a_x}{1+\delta_s}\right)}{\sqrt{p_t^2 + \frac{2p_t}{\beta_s} + 1 - \left(p_x - \frac{a_x}{1+\delta_s}\right)^2 - \left(p_y - \frac{a_y}{1+\delta_s}\right)^2}} \neq p_x$. Careful when using p_x, p_y inside dipoles and solenoids!

TWISS: Exact Hamiltonian, but maps truncated to 2nd order. δ_s dependency is exact, p_t dependency is approximated.
TRACK and EMIT: <u>δ_s forced to 0</u> (but inconsistencies found), maps are generally symplectic solutions of approximated Hamiltonians. DELTA in TRACK changes δ and not δ_s.
<u>Considering aligning behaviour of TRACK and EMIT to TWISS in the next MAD-X version.</u>

Radiation effect: average power loss

$$-\frac{dE}{dT} = \frac{q^2}{6\pi\epsilon_0 m^2 c^3}\frac{dP^\mu}{d\tau}\frac{dP_\mu}{d\tau} = \frac{2}{3}\frac{r_q}{mc}\frac{dP^\mu}{d\tau}\frac{dP_\mu}{d\tau}$$

For static magnetic fields $\frac{dP^\mu}{d\tau}\frac{dP_\mu}{d\tau} = (q\gamma\beta c B_\perp)^2 = \left(\frac{PP_0 \mathrm{b}_\perp}{m}\right)^2$ where $\mathrm{b}_\perp = \frac{q}{P_0}B_\perp$

$$-\frac{dE}{dT} = \frac{2}{3}\frac{r_q P_0^2 P^2}{m^3 c} b_\perp^2 \qquad \frac{dp_t}{ds} = -\frac{dE}{dT}\frac{dt}{ds}\frac{1}{P_s c} = \frac{2}{3}\frac{r_q P_s^3}{m^3 c^3}\frac{(1+\delta)^2}{(1+\delta_s)^2} b_\perp^2 \frac{dH}{dp_t} = \frac{2}{3} r_q \beta_s^3 \gamma_s^3 \frac{(1+\delta)^2}{(1+\delta_s)^2} b_\perp^2 \frac{dH}{dp_t}$$

In dipoles for instance $\frac{dH}{dp_t} = \frac{d(-ct)}{ds} = -\frac{(1+\mathrm{hx})\left(\mathrm{p_t} - \frac{1}{\beta_s}\right)}{\sqrt{(1+\delta)^2 - p_x^2 - p_y^2}} \approx -\frac{(1+\mathrm{hx})}{\beta}\left(1 - \frac{1}{2}\frac{\mathrm{p_x^2 + p_y^2}}{(1+\delta)^2}\right)$

Sometimes missing in the code

In MAD-X we calculate

$$r = \frac{E - E_{new}}{E} = -\frac{2}{3} r_q \beta_s^3 \gamma_s^3 \frac{(1+\delta)}{(1+\delta_s)^2}\beta \int_0^L b_\perp^2 \frac{dH}{dp_t} ds$$

$$f = \frac{P_{new}}{P} = \sqrt{\frac{r(r-2)}{\beta^2} + 1}$$

The energy is lost in the direction of the momentum of particle and not the canonical momentum:

$$p_x^{\mathrm{new}} = f\,(p_x - a_x) + a_x$$
$$p_y^{\mathrm{new}} = f(p_y - a_y) + a_y$$
$$p_t^{new} = p_t(1 - r) - \frac{r}{\beta_s}$$

Conclusion

- Invest on MAD-X code to create a solid platform for FCC-ee:
  - MAD-X already used for FCC-ee studies, has synergies with many other projects.
  - Improve interface between MAD-X and other codes such as Xsuite relying on Python.
- Radiation effects in MAD-X used already in many FCC-ee studies:
  - Few calculations have shown inconsistent results mostly related to thin multipole elements and solenoid.
  - Some usability issues have been identified related to tapering in the last version.
- Stabilize radiation calculations in MAD-X:
  - Review and document radiation related physics applied to MAD-X.
  - Compare calculations on FCC-ee or other test lattices using different methods such as direct tracking, map formalism, radiation integral formalism.
  - Fix issues in the code.
  - Coordinate with optics studies for defining priorities such as interaction region modelling and vertical emittance studies.

References

- MAD8 Physics guide: https://cern.ch/mad8/doc/phys_guide.pdf

Still most complete reference. Unpublished, some typos

We are in the process of correcting typos and re-release for MAD-X

- For radiation effects:

J. Jowett, Introductory statistical mechanics for electron storage rings https://doi.org/10.1063/1.36374

Back-up

MAD-X Variables and Hamiltonian

$$L = -\frac{mc^2}{\gamma} + q\dot{\boldsymbol{R}} \cdot \boldsymbol{A} - q\mathrm{V}$$

For piecewise constant curvatures!

$$\dot{\boldsymbol{R}} = \dot{s}(1 + hx)\boldsymbol{e}_s + \dot{x}\boldsymbol{e}_x + \dot{y}\boldsymbol{e}_y$$

$$\boldsymbol{R}(s) = \boldsymbol{R}_0(s) + x(s)\boldsymbol{e}_x(s) + y(s)\boldsymbol{e}_y(s)$$

$$P_x = \partial_{\dot{x}}L = m\gamma\dot{x} + qA_x$$
$$P_y = \partial_{\dot{y}}L = m\gamma\dot{y} + qA_y$$
$$P_s = \partial_{\dot{s}}L = (1 + hx)^2 m\gamma\dot{s} + (1 + hx)qA_s$$

$$H_E = P_x\dot{x} + P_y\dot{y} + P_s\dot{s} - L = c\sqrt{(P_x - qA_x)^2 + \left(P_y - qA_y\right)^2 + \left(\frac{P_s}{1 + hx} - qA_s\right)^2 + m^2c^2} + q\mathrm{V}$$

$$H_{-P_s} = -(1 + hx)\left(\sqrt{\left(\frac{E}{c} - \frac{qV}{c}\right)^2 - (P_x - qA_x)^2 - \left(P_y - qA_y\right)^2 - m^2c^2} + q\mathrm{A_s}\right)$$

Scaled coordinates by

$P_s = m_0 c\beta_s\gamma_s = P_0(1 + \delta_s)$

$$p_x = \frac{P_x}{P_s} \quad p_y = \frac{P_y}{P_s} \quad p_t = \frac{E - E_s}{P_s c} \quad t = \frac{1 + \eta\delta_s}{\beta_s}s - cT \quad a_{x,y,s} = \frac{q}{P_0}A_{x,y,s}$$

$$H = \frac{1 + \eta\delta_s}{\beta_s}p_t - (1 + hx)\sqrt{p_t^2 - \frac{2p_t}{\beta_s} + 1 - \left(p_x - \frac{a_x}{1 + \delta_s}\right)^2 - \left(p_y - \frac{a_y}{1 + \delta_s}\right)^2} - \frac{(1 + hx)a_s}{1 + \delta_s}$$

MAD-X: useful relations

$$p_t^{new} = p_t(1-r) - \frac{r}{\beta_s} = \frac{E_{new} - E_s}{\beta_s E_s} = \frac{E - E_s}{\beta_s E_s}(1-r) - \frac{E_s r}{E_s \beta_s} = \frac{E - E_s - rE + E_s r - E_s r}{E_s \beta_s} \rightarrow E_{new} - E = -rE$$

$$f = \sqrt{\frac{r(r-2)}{\beta^2} + 1} = \sqrt{\frac{(E - E_{new})(-E - E_{new})}{E^2 \beta^2} + 1} = \frac{\sqrt{E_{new}^2 - E^2 + Pc^2}}{Pc} = \frac{P_{new}}{P}$$

$$p_x^{\text{new}} = f\,(p_x - a_x) + a_x$$
$$p_y^{\text{new}} = f(p_y - a_y) + a_y$$
$$p_t^{new} = p_t(1-r) - \frac{r}{\beta_s}$$

$$df = \frac{\frac{(r-1)dr}{\beta^2} - r(r-2)d(1/\beta^2)}{f}$$

$$dp_x = df\,(p_x - a_x) + f(dp_x - da_x) + da_x$$
$$dp_y = df\,\left(p_y - a_y\right) + f\left(dp_y - da_y\right) + da_y$$
$$dp_t = (p_t - 1/\beta_s)dr$$

$$\frac{1}{\beta^2} = \frac{\left(\mathrm{p_t} + \frac{1}{\beta_s}\right)^2}{p_t^2 + \frac{2p_t}{\beta_s} + 1}$$

$$\frac{d(1/\beta^2)}{dp_t} = \frac{2\left(\mathrm{p_t} + \frac{1}{\beta_s}\right)}{p_t^2 + \frac{2p_t}{\beta_s} + 1} - \frac{2\left(\mathrm{p_t} + \frac{1}{\beta_s}\right)^3}{\left(p_t^2 + \frac{2p_t}{\beta_s} + 1\right)^2} = \frac{2(\beta_s - 1)}{\gamma_s^2\left(p_t^2 + \frac{2p_t}{\beta_s} + 1\right)^2}$$

$$\frac{1}{4\pi\epsilon_0} = 1.00000000055(15)\ 10^{-7}c^2 \text{(since 2019)}$$

Fixes in track

https://github.com/MethodicalAcceleratorDesign/MAD-X/pull/1079

arad=ten_m_16*charge*charge*get_variable("qelect")*clight*clight/mass;

```
const = arad * (betas * gammas)**3 / three
```

$$2\text{ const} = \frac{q^2\beta_s^3\gamma_s^3}{6\pi\epsilon_0 mc^2} = \frac{10^{-7}c^2q_e^2e}{10^9 m_{[GeV]}}\frac{2}{3}\beta_s^3\gamma_s^3$$

```
          do jtrk = 1,ktrack
+            x = track(1,jtrk)
+            pt = track(6,jtrk)
             curv = sqrt((dipr + dxt(jtrk))**2 + (dipi + dyt(jtrk))**2) / elrad
             if (quantum) then
                call trphot(elrad,curv,rfac,pt)
             else
-               rfac = const * curv**2 * elrad
+               delta_plus_1 = sqrt(pt*pt + two*pt*beti + one);
+               rfac = const * curv**2 * delta_plus_1 * elrad * (one + dipr/elrad * x)
             endif
```

$$\text{curv} = b = \frac{\sqrt{\Delta p_x^2 + \Delta p_y^2}}{l}$$

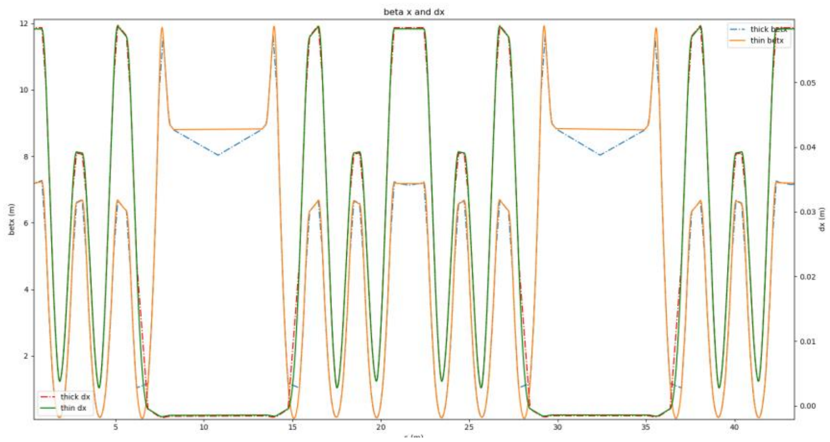
Missing $(1+\delta)(1+hx)$ dependence

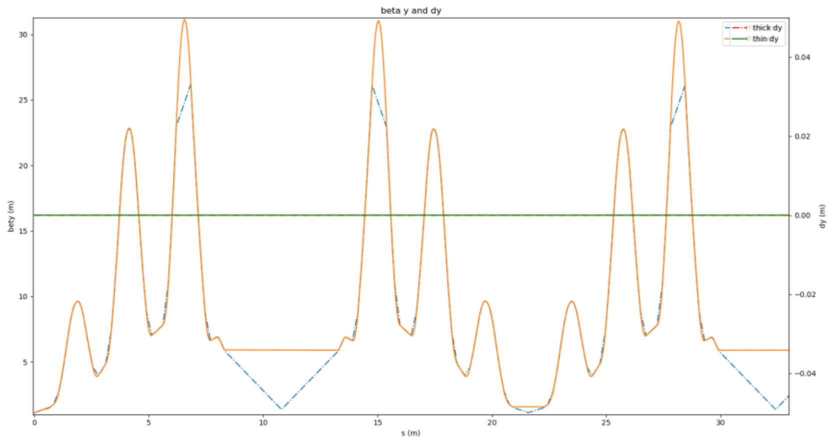
```
-
-          DXT(:ktrack) = dipr*dipr*TRACK(1,:ktrack)/elrad
-          DYT(:ktrack) = dipi*dipi*TRACK(3,:ktrack)/elrad
+          !!! terms should scale with h_c k0 therefore (dipr-dbr) dipr
+          DXT(:ktrack) = (dipr-dbr)*dipr*TRACK(1,:ktrack)/elrad
+          DYT(:ktrack) = (dipi-dbi)*dipi*TRACK(3,:ktrack)/elrad
```

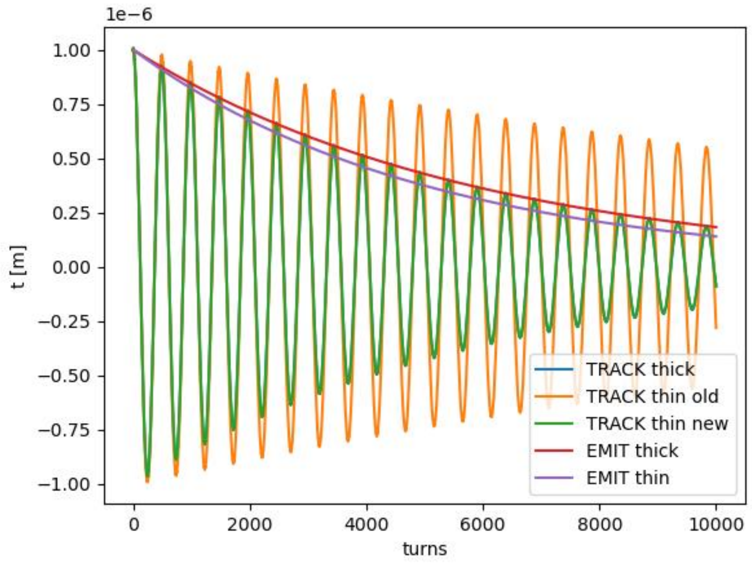
Not directly related to radiation damping but still need fix when k0l different from angle

Damping times of Electra lattice

Taking Electra lattice as an example of small lattice with few cavities







$$t[turns] = t_0 e^{-\alpha_t T_0\, turns}$$

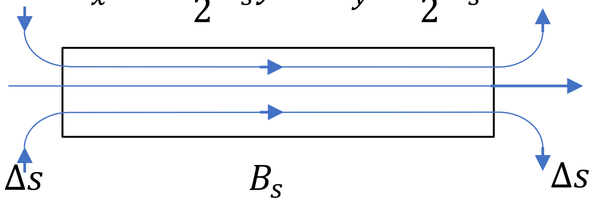
Method	**Damping constant $\alpha_t[1/s]$**
EMIT Thick	196.3
EMIT Thin	227.4
TRACK Thick	196.3
TRACK Thin (before fix)	70.12
TRACK Thin (after fix)	198.4
Twiss thin using $D = \frac{\oint k_0 D_x (k_1 + k_0^2)\, ds}{\oint k_0^2\, ds}$ $\alpha_t = \frac{W_0}{2E_0T_0}(2 + D)$	198.2

NB. EMIT thick and thin gives the same T_0 and W_0

Solenoid

$$A_x = -\frac{1}{2}B_s y \qquad A_y = \frac{1}{2}B_s x$$

$$B_x = -\frac{B_s x}{2\Delta s} \qquad B_y = -\frac{B_s y}{2\Delta s}$$



$$B_x = \frac{B_s x}{2\Delta s} \qquad B_y = \frac{B_s y}{2\Delta s}$$

$$k_s = \frac{qB_s}{2\,P_0}$$

Exit Fringe

$\Delta x' = -k_s y \quad \Delta p_x = 0$

$\Delta y = k_s x \quad \Delta p_y = 0$

Entry Fringe

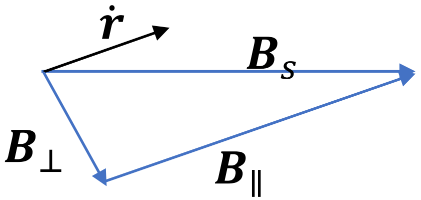
$\Delta x' = k_s y \quad \Delta p_x = 0$

$\Delta y = -k_s x \quad \Delta p_y = 0$

$$x' = \frac{(1 + h\,x)\left(p_x - \frac{a_x}{1+\delta_s}\right)}{\sqrt{p_t^2 + \frac{2p_t}{\beta_s} + 1 - \left(p_x - \frac{a_x}{1+\delta_s}\right)^2 - \left(p_y - \frac{a_y}{1+\delta_s}\right)^2}} \approx \frac{p_x + k_s y}{1+\delta}$$

$$y' = \frac{(1 + h\,x)\left(p_y - \frac{a_y}{1+\delta_s}\right)}{\sqrt{p_t^2 + \frac{2p_t}{\beta_s} + 1 - \left(p_x - \frac{a_x}{1+\delta_s}\right)^2 - \left(p_y - \frac{a_y}{1+\delta_s}\right)^2}} \approx \frac{p_y - k_s x}{1+\delta}$$

N.B. In MAD-X, KS = $2k_s = qB_s/P_0$



$$B_\perp = B_s\sqrt{x'^2 + y'^2} \qquad b_\perp = 2k_s\sqrt{x'^2 + y'^2}$$

Multipole map

General equation for average energy loss $-\frac{dE}{dt} = \frac{2}{3}\frac{r_e}{m_0 c}\frac{dP^\mu}{d\tau}\frac{dP_\mu}{d\tau}$ with $r_e = \frac{e^2}{m_0 c^2}$

With $\frac{dP_\mu}{d\tau} = \gamma e(0, v \times B) = \frac{e}{m_0 c}(0, P \times B)$ we get $-\frac{dE}{dt} = \frac{2}{3}\frac{e^2 r_e P^2}{(m_0 c)^3} B^2$

Integrating on the integration length $l(1 + hx)$ with the integrated multipole kick $\Delta p = \sqrt{\Delta p_x^2 + \Delta p_y^2}$ we get:

$$-\Delta E = \frac{2}{3}\frac{e^2 r_e P^2 P_0^2}{(m_0 c)^3}\left(\frac{\Delta p}{l}\right)^2 \frac{l(1+hx)}{\beta c} = \frac{2}{3}\frac{e^2 r_e E\, P_0^3 (1+\delta)}{c(m_0 c)^3}\left(\frac{\Delta p}{l}\right)^2 l(1+hx)$$

with $\delta = \frac{P - P_0}{P_0 c}$ $\quad p_t = \frac{E - E_0}{P_0 c}$ $\quad p_x = \frac{P_x}{P_0}$ $\quad p_x = \frac{P_y}{P_0}$

Assuming $r = \frac{\Delta E}{E}$, $f = \frac{\Delta P}{P} = \sqrt{1 + r(r-2)/\beta}$

then

$$p_x^{new} = f\, p_x$$
$$p_y^{new} = f\, p_y$$
$$p_t^{new} = p_t(1 - r) - \frac{r}{\beta_0}$$

Magnetostatic

Curved frame traverse magnetic fields

$$B_x(s,0,s) = 0$$

$$B_y(x,0,s) = \sum_{n=0}^{N} B_n \frac{x^n}{n!}$$

$$B_s(x,0,s) = 0$$

$$A_x = 0$$

$$A_y = 0$$

$$A_s = -B_0\left(x - \frac{h^2}{2(1+hx)}\right) - B_1\left(\frac{1}{2}(x^2 - y^2) - \frac{h}{6}x^3\right) - B_2 \ldots$$

Straight frame traverse magnetic fields

$$h = 0 \;\rightarrow\; B_y(x,y) - iB_x(x,y) = -\frac{q}{P_0}\sum \frac{k_n^N + ik_n^S}{n!}(x+iy)^n$$

Straight frame solenoidal fields

$$A_x = -yU$$

$$A_y = xU$$

$$A_s = 0$$

$$U = \sum_{n=0}^{\infty} \frac{(-1)^n r^{2n} \partial_z^{2n} B_z(0,0,z)}{2^{2n+1} n!\,(n+1)!}$$