



REVIEW OF MAD-X FOR FCC-EE STUDIES

Supervisors : R. De Maria, A. Faus-Golfe, F. Schmidt
Ph.D student : G. Simon

Special thanks to : T. Pieloni and L. Van Riesen-Haupt

Abstract

The design of the electron-positron Future Circular Collider (FCC-ee) requires a special effort on optics codes (like **MAD-X**) in terms of accuracy, consistency, and performance. In particular the impact of **Synchrotron Radiation (sawtooth and tapering)** has to be carefully evaluated in terms of consistency, absolute accuracy and stability.

I will analyze the MAD-X TWISS, TRACK and EMIT modules and I will make improvement proposals.

Physics context: Synchrotron Radiation

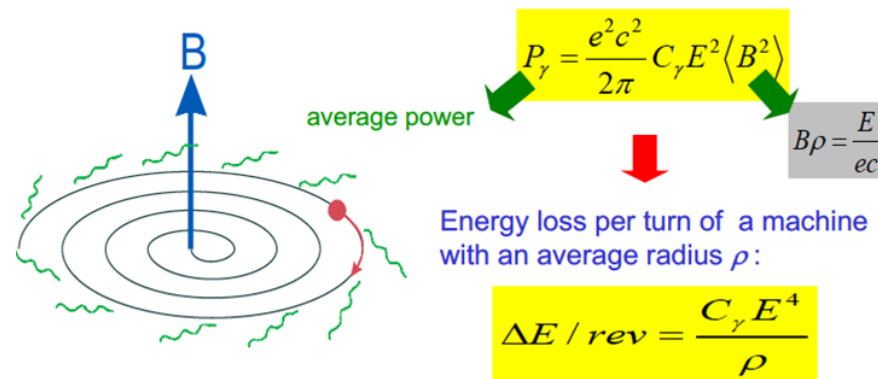
Synchrotron radiation is an electromagnetic radiation emitted when relativistic charged particles are subject to an acceleration perpendicular to their velocity ($\mathbf{a} \perp \mathbf{v}$).

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 per beam has been fixed at 50MW, so the circumference is fixed to meet this power loss. Hence the high circumference of the FCCee.

Synchrotron radiation from an e^- in a magnetic field:



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

Sawtooth effect and tapering

Sawtooth effect

Energy loss due to SR and energy gain in RF cavities lead to what's called 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).

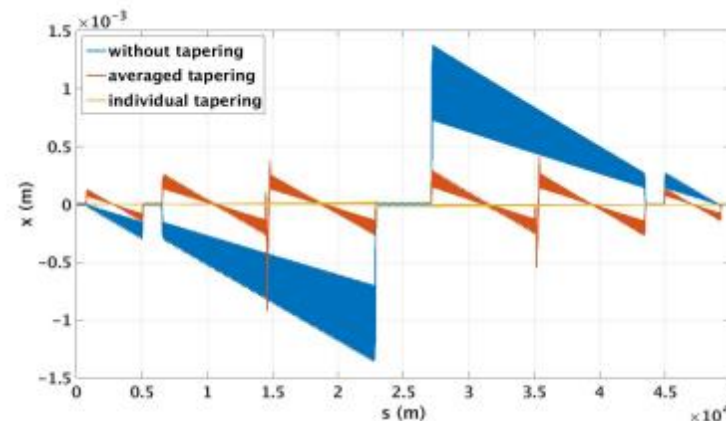
Because of the **high energies** involved in the **FCCee** and its large circumference. The sawtooth effect can't be neglected. Also, the loss of energy is even more important at the IP due to local chromaticity correction, which is an other reason why the sawtooth effect has to be corrected.

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”.

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”.



Tapering in MAD-X

In MADX the tapering is calculated as follow :

- 1) First, we calculate p_t 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.

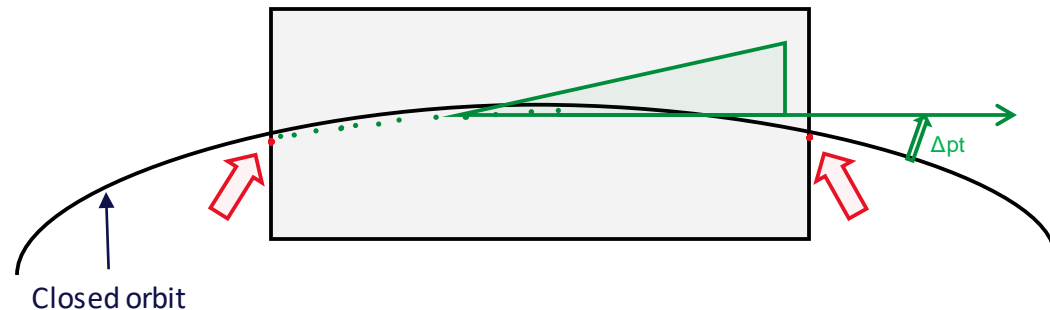
- 2) Then we track the particle through the element with the radiation.

- 3) We record the p_t at the exit of the element.

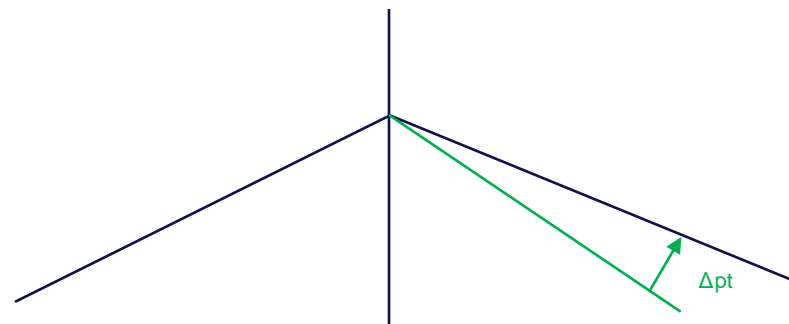
- 4) At the end we scale the dipole's strength by the average p_t calculated to correct the offset of the particle's orbit.

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, implementing this for all individual magnets 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 transferred to a thin lattice using the module MAKETHIN.



Scheme for thick lattice



Scheme for thin lattice

TRACK and TWISS code

Here we will compare the two tracking codes in TWISS and TRACK. EMIT on the other hand complements the linear transfer matrix from TWISS with the radiation. The parameter compared between the three modules is the tune because it can be precisely measured unlike the orbit. Which is a perfect tool to control the orbit offset of the beam. If the tune isn't consistent with what we expect, then it means that the offset of the beam is too large.

TRACK

```

subroutine ttddf(el,track,ktrack)
  use trackfi
  use math_constfi, only : one, two
  implicit none
  !-----*
  ! Purpose:
  ! Track a set of particle through a drift space.
  ! Input/output:
  ! TRACK(6,*) (double) Track coordinates: (X, PX, Y, PY, T, PT).
  ! KTRACK (integer) number of surviving tracks.
  ! Output:
  ! EL (double) Length of drift.
  !-----*
  double precision :: el, track(6,*)
  integer :: ktrack

  double precision :: pt, px, py, l_pz
  integer :: i

! picked from trturn in madx.ss
!$OMP PARALLEL PRIVATE(i, px, py, pt, l_pz)
!$OMP DO
  do i = 1, ktrack
    px = track(2,i)
    py = track(4,i)
    pt = track(6,i)
    ! L/pz
    l_pz = el / sqrt( one + two*pt*beti + pt**2 - px**2 - py**2 )
    l_pz = el / sqrt( one + two*pt*beti + pt*pt - px*px - py*py ) !speedup hrr Nov 2021
    track(1,i) = track(1,i) + l_pz*px
    track(3,i) = track(3,i) + l_pz*py
    ! track(5,i) = track(5,i) + el*(beti + pt*dtbys) - (beti+pt)*l_pz
    !---- AK 20060413
    !---- Ripken DESY-95-189 p.36
    track(5,i) = track(5,i) + (el*beti - (beti + pt) * l_pz)
  enddo
!$OMP END DO
!$OMP END PARALLEL
end subroutine ttddf

```

TWISS

```

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

```

LINEAR OPTICS CALCULATIONS WITH *pt* OFFSETS

- 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 :

$$M_{ij}(Z_c) = M_{ij}(0) + \sum_k T_{ijk} Z_k$$

- 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.

- $\sum_k T_{ijk} Z_k$ 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 :

$$Z_i = K_i + \sum_j M_{ij} X_j + \sum_{jk} T_{ijk} X_j X_k = K_i + [\sum_j (M_{ij} + \sum_k T_{ijk} X_k) X_j]$$

- 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.

$$k_{new} = \frac{k}{1 - \delta(p_t)}$$

Accurate to first order

$$k_{new} = k(1 + k_{tap}) = k(1 + \delta(p_t))$$

Accurate to any order

```

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

```

$\sum_k T_{ijk} Z_k$

$$Z_i = K_i + [\sum_j (M_{ij} + \sum_k T_{ijk} X_k) X_j]$$

Review of MAD-X 5.07.00, 5.08.01 and 5.09.00

We've been reviewing tunes calculations through MAD-X 5.07.00, 5.08.01 and 5.09.00. The idea here being to see the consistency between the previous version before working on a new model of the tapering.

As explained before, we concentrated on the tunes because it's a precise parameter measurable which gives information about the orbit offset.

To calculate the tunes, TWISS computes transfer matrices using second order expanded maps from the origin, calculating the eigenvalues of the matrices. TRACK tracks one turn and use the finite difference to compute the linear matrix, then the eigenvalues are extracted to calculate the tunes and EMIT uses the same method as TWISS but adding the radiation effects.

The tapering calculations are different between the version 5.07.00 and 5.08.01/5.09.00 of MAD-X :

5.07.00 : $k_{new} = \frac{k}{1-\delta(p_t)}$ which is accurate only to first order

5.08.01 and 5.09.00 : $k_{new} = k(1 + k_{tap}) = k(1 + \delta(p_t))$
which is accurate to any order.

TWISS : TWISS calculates the linear lattice functions and optionally the chromatic functions.

TRACK : TRACK initiates trajectory tracking.

EMIT : EMIT calculates the equilibrium emittances.

EXACT (TWISS option) : If this is used the dirft is expanded around the actual closed orbit instead of the ideal orbit.

Tapering formulas in MAD-X with old versions

MAD version	5.07.00	5.08.01	5.09.00
rbend/sbend	$K0 = k0 / (1 - pt / \beta0)$	$k = k(1 + ktap)$ $Ktap = pt / \beta0$	$K0 = k0(1 + ktap)$ $Ktap = pt / \beta0$
quadrupole	$K1 = k1 + k1tap$ $k1tap = 1 / (1 - pt / \beta0) - 1$	$K1 = k1(1 + ktap)$ $Ktap = pt / \beta0$	$K1 = k1(1 + ktap)$ $Ktap = pt / \beta0$
sextupole	$K2 = k2 + k2tap$ $k2tap = 1 / (1 - pt / \beta0) - 1$	$K2 = k2(1 + ktap)$ $Ktap = pt / \beta0$	$K2 = k2(1 + ktap)$ $Ktap = pt / \beta0$
octupole	x	x	$K3 = k3(1 + ktap)$ $Ktap = pt / \beta0$
multipole	x	x	$Kn = kn(1 + ktap)$ $Ktap = pt / \beta0$

MAD-X coordinates and parameters

In the model we're currently testing (for now called PR1163), we want to stay at the closed orbit, in order to do so we want $p_t = 0$. To obtain this, we introduce δ_s which is the δ for $p_t = 0$. We redefine our parameters according to δ_s . We firstly focused our work on quadrupole because they're the main sources of optics errors and the easiest first approach to our problem.

$$p_x = \frac{P_x}{P_s} \quad p_y = \frac{P_y}{P_s} \quad p_t = \frac{E - E_s}{cP_s} \quad t = \frac{s}{\beta_0}(1 + \eta\delta_s) - cT$$

$$E_0 = \beta_0 \gamma_0 m_0 c^2 \quad P_s = P_0 (1 + \delta_s) \quad E_s = \beta_s \gamma_s m_0 c^2$$

$$\delta(p_t) = \frac{P - P_s}{P_s} = \sqrt{1 + \frac{2p_t}{\beta_s} + p_t^2} - 1 \quad \delta_s \neq \delta \text{ in the general case}$$

δ_s is a **parameter** that defines the scaled momenta, $\delta(p_t)$ is a **function of** p_t .

For high energy machine, $\delta(p_t) \rightarrow p_t$ for $\beta_s \rightarrow 1$

δ_s is used only in TWISS but not in TRACK for $\delta_s = 0$

New model implementation in MAD-X (work in progress)

As explained before we want $p_t = 0$ so we introduced δ_s . To do so we proceed to a change of variable in the calculation of the transport map of the quadrupoles as a first test.

```
+ double precision :: x,px,y,py,t,pt,deltaplusone,nk1
+
+ if (exact_expansion) then
+   x= orbit(1)
+   px= orbit(2)
+   y= orbit(3)
+   py= orbit(4)
+   t= orbit(5)
+   pt= orbit(6)
+   deltaplusone=sqrt(pt**2+2*pt/beta+1)
+   nk1=sk1/deltaplusone
+ else
+   nk1= sk1
+ endif
```

```
+ if (exact_expansion) then
+   !---- First-order terms.
+   re(1,1) = cx
+   re(1,2) = sx/deltaplusone
+   re(2,1) = - nk1 * sx * deltaplusone
+   re(2,2) = cx
+   re(3,3) = cy
+   re(3,4) = sy /deltaplusone
+   re(4,3) = + nk1 * sy *deltaplusone
+   re(4,4) = cy
+   re(5,6) = el/(beta*gamma)**2
+
+   ek(5) = el*dtbyds ! to be checked
+
+ else
```

!---- Track orbit.

```
+ if (exact_expansion) then
+
+   orbit(1)=cx*x + sx*px/deltaplusone
+   orbit(2)=-nk1 * sx * x*deltaplusone + cx*px
+   orbit(3)=cy*y + sy*py/deltaplusone
+   orbit(4)=nk1 * sy * y*deltaplusone + cy*py
+   orbit(5)=el/(beta*gamma)**2*pt
+   re(5,1)=re(5,1) + te(5,1,1)*x + te(5,1,2)*px + te(5,1,3)*y + te(5,1,4)*py +
+   te(5,1,5)*t + te(5,1,6)*pt
+ else
+   if (ftrk) call tmtrak(ek,re,te,orbit,orbit)
+ endif
```

Quadrupole map

Quadrupole map (truncated H)

$$\begin{aligned}
 x_f &= C_x x + S_x \frac{p_x}{1 + \delta} \\
 p_{x,f} &= -k(1 + \delta) S_x x + C_x p_x \\
 y_f &= C_y y + S_y \frac{p_y}{1 + \delta} \\
 p_{y,f} &= k(1 + \delta) S_y y + C_y p_y \\
 t_f &= \dots
 \end{aligned}$$

$$k = \frac{k_1 + k_0 h}{1 + \delta}$$

$$\begin{aligned}
 C_{x,y} &= \cos \sqrt{|k|} l \text{ or } \cosh \sqrt{|k|} l \\
 S_{x,y} &= \frac{\sin \sqrt{|k|} l}{\sqrt{|k|}} \text{ or } \sinh \frac{\sqrt{|k|} l}{\sqrt{|k|}}
 \end{aligned}$$

Variable changes in
qdbody algorithm for
TWISS in order to track
particles with $p_t = 0$

Exact transfer matrix

$$\begin{aligned}
 R_{11} &= \frac{\partial x_f}{\partial x} = C_x & R_{12} &= \frac{\partial x_f}{\partial p_x} = \frac{S_x}{1 + \delta} \\
 R_{21} &= \frac{\partial p_{x,f}}{\partial x} = -k S_x (1 + \delta) & R_{22} &= \frac{\partial p_{x,f}}{\partial p_x} = C_x & \dots \\
 & \dots & & \dots & \dots
 \end{aligned}$$

Tunes results in MAD-X

TWISS	No tapering	Tapering w/o EXACT	Tapering + EXACT
5.07.00	0.2240 0.3600	0.2240 0.3593	x
5.08.01	0.2240 0.3600	0.2153 0.3509	0.2168 0.3523
5.09.00	0.2240 0.3600	0.2153 0.3509	0.2168 0.3523
PR1163	0.2240 0.3600	0.2153 0.3509	0.2171 0.3526
EMIT	No tapering	Tapering w/o EXACT	Tapering + EXACT
5.07.00	0.2240 0.3600 0.101	0.2240 0.3600 0.0815	x
5.08.01	0.2240 0.3600 0.101	0.2153 0.3515 0.0815	0.2168 0.3530 0.0815
5.09.00	0.2240 0.3600 0.101	0.2153 0.3515 0.0815	0.2168 0.3530 0.0815
PR1163	0.2240 0.3600 0.101	0.2153 0.3515 0.0815	0.2171 0.3533 0.0815
TRACK	No tapering	Tapering w/o EXACT	Tapering + EXACT
5.07.00	0.2240 0.3600 0.101	0.1797 0.3942 0.0833	x
5.08.01	0.2240 0.3600 0.101	0.2236 0.3588 0.0815	0.2236 0.3588 0.0815
5.09.00	0.2240 0.3600 0.101	0.2236 0.3588 0.0815	0.2236 0.3588 0.0815
PR1163	0.2240 0.3600 0.101	0.2236 0.3588 0.0815	0.2236 0.3588 0.0815

Conclusion

MAD-X TWISS and TRACK use two separate numerical methods that are not necessarily connected.

MAD-X TWISS computes transfer matrices using second order expanded maps from the origin. This introduces small inaccuracy that are relevant for the FCC lattices when tapering is used.

MAD-X tapering in V5.07 was fine-tuned on the TWISS calculation, giving consistent results at the expenses of the tracking model.

MAD-X tapering in V5.08 and V5.09 introduce a more coherent tapering method, giving better result in tracking, but exposing the inaccuracies of TWISS.

We are in the process of making the TWISS more accurate, using the exact option.



Thank you
for your attention !



Back up slides

Quadrupole fix?

We can write the transfer matrix $M\left(z = (x, y, p_x, p_y, p_t), \delta_s, k_1 = \frac{\partial B_y}{\partial x} \frac{q}{P_0}, l\right)$

Exact for $x = y = p_x = p_y = p_t = 0$ and any δ_s .

Approximated using $M_{ij}(z = (x, y, p_x, p_y, p_t), \delta_s, k_1, l) = M_{ij}(0, 0, 0, 0, 0, \delta_s, k_1, l) + 2 \sum_k T_{ij}(0, 0, 0, 0, 0, \delta_s, k_1, l) z_k$

The fix is computing $M(z = (x, y, p_x, p_y, p_t), \delta_s, k_1, l)$ by using $M(z = (x, y, p_x^*, p_y^*, 0), \delta_s^*, k_1, l)$ where

δ_s^* is chosen such that $p_t^* = 0$ that is when $E = E_s^*$ or $P = P_s^*$ $1 + \delta = 1 + \delta_s^*$, that is $\delta = \delta_s^*$ and

$$p_x^* = \frac{P_x}{P_s^*} = \frac{P_x P_s}{P_s P_s^*} = \frac{p_x}{1 + \delta_s^*} \quad p_y^* = p_x \frac{p_y}{1 + \delta_s^*}$$

Quadrupole map

Quadrupole map (truncated H)

$$x \rightarrow C \cdot x + \frac{S}{\sqrt{k}} \cdot \frac{p_x}{1+\delta},$$

$$p_x \rightarrow -\sqrt{k}(1+\delta)S \cdot x + C \cdot p_x,$$

$$y \rightarrow \hat{C} \cdot y + \frac{\hat{S}}{\sqrt{-k}} \cdot \frac{p_y}{1+\delta},$$

$$p_y \rightarrow -\sqrt{-k}(1+\delta)\hat{S} \cdot y + \hat{C} \cdot p_y$$

$$z \rightarrow z + \frac{1-\beta_0^2}{\beta_0^2} p_t \cdot L - \frac{1}{2} \frac{\left(p_t + \frac{1}{\beta_0}\right)}{(1+\delta)^2} \cdot$$

$$\cdot \left\{ \frac{1}{2} k_0 \left[x^2 \left(L - \frac{C \cdot S}{\sqrt{k}} \right) - y^2 \left(L - \frac{\hat{C} \cdot \hat{S}}{\sqrt{-k}} \right) \right] + \right.$$

$$+ \frac{1}{2} \frac{1}{1+\delta} \left[p_x^2 \left(L + \frac{C \cdot S}{\sqrt{k}} \right) + p_y^2 \left(L + \frac{\hat{C} \cdot \hat{S}}{\sqrt{-k}} \right) \right] +$$

$$\left. - \left[x \cdot p_x \cdot (1-C^2) + y \cdot p_y \cdot (1-\hat{C}^2) \right] \right\}$$

$$p_t \rightarrow p_t.$$

Exact transfer matrix

$$R_{11} = \frac{\partial x_f}{\partial x} = C_x$$

$$R_{12} = \frac{\partial x_f}{\partial p_x} = \frac{S_x}{1+\delta}$$

$$R_{16} = \frac{\partial x_f}{\partial p_t} = \frac{1}{2} \left(\frac{l C_x p_x}{k} - \frac{l S_x x \operatorname{sign}(k)}{2} - \frac{S_x p_x}{(1+\delta)} \left(\frac{1}{2k} + \frac{1}{1+\delta} \right) \right) \frac{\partial k}{\partial p_t}$$

$$R_{21} = \frac{\partial p_{x,f}}{\partial x} = -k S_x (1+\delta)$$

$$R_{22} = \frac{\partial p_{x,f}}{\partial p_x} = C_x$$

$$\dots$$

Useful formula

$$\delta + 1 = \sqrt{1 + \frac{2p_t}{\beta_s} + p_t^2}$$

$$x_p = \frac{p_x}{1+\delta} \quad y_p = \frac{p_y}{1+\delta}$$

$$k = \frac{k_1 + k_0 h}{1+\delta}$$

$$C_{x,y} = \cos \sqrt{|k|} l \text{ or } \cosh \sqrt{|k|} l$$

$$S_{x,y} = \frac{\sin \sqrt{|k|} l}{\sqrt{|k|}} \text{ or } \sinh \frac{\sqrt{|k|} l}{\sqrt{|k|}}$$

$$\frac{\partial \delta}{\partial p_t} = \frac{p_t + 1/\beta_s}{1+\delta} = \frac{1}{\beta}$$

$$\frac{\partial k}{\partial p_t} = -\frac{k}{1+\delta} \frac{\partial \delta}{\partial p_t}$$

$$\frac{\partial \sqrt{|k|}}{\partial p_t} = \frac{\operatorname{sign}(k)}{2\sqrt{|k|}} \frac{\partial k}{\partial p_t}$$