# MADX-PTC Review and Fixes after Cross-Checks with MAD-NG

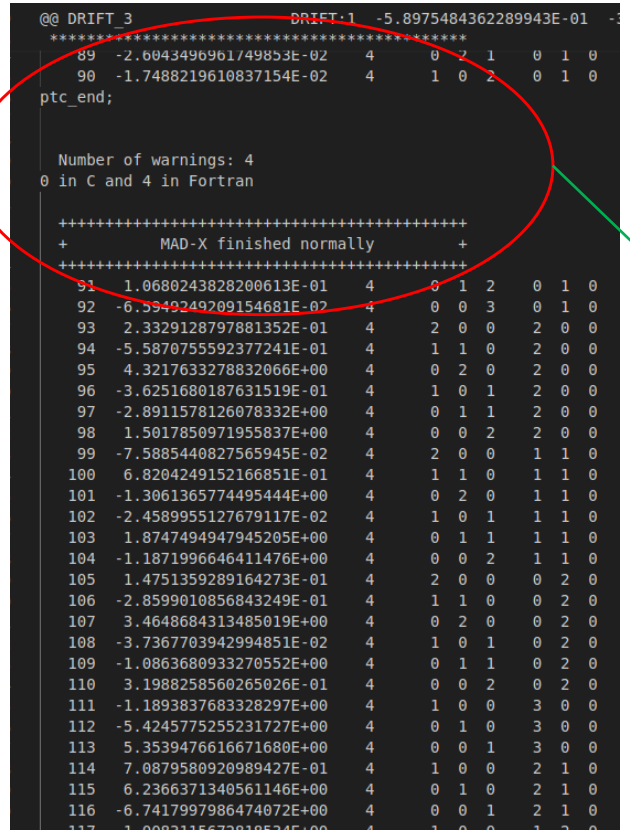
Joshua Gray & Laurent Deniau

# Context

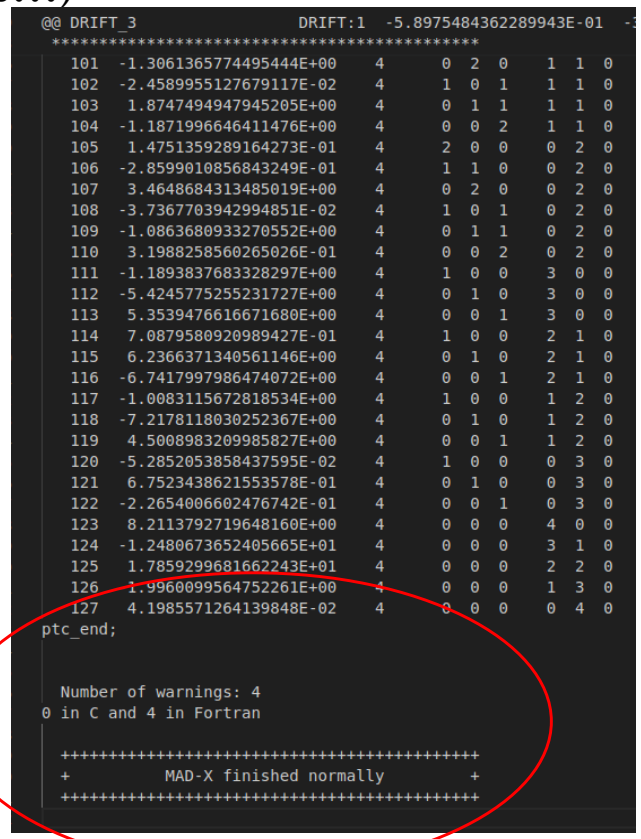
- I will refer to all indexing of arrays in the Fortran style
  i.e., knl[1] refers to the first index of the array knl.
- Red bullet points or numbers refer to a bug
  - A bug
  - An improvement
- In total there are **9** bugs fixed (closes 2 issues) **19** improvements (closes 1 issue), after cross-checks with MAD-NG.
- I have made all these changes and will open a pull request; this presentation is to list the changes I would like to make.

# DA Map Output

- Flush the DA map output (cleans up debugging etc…)



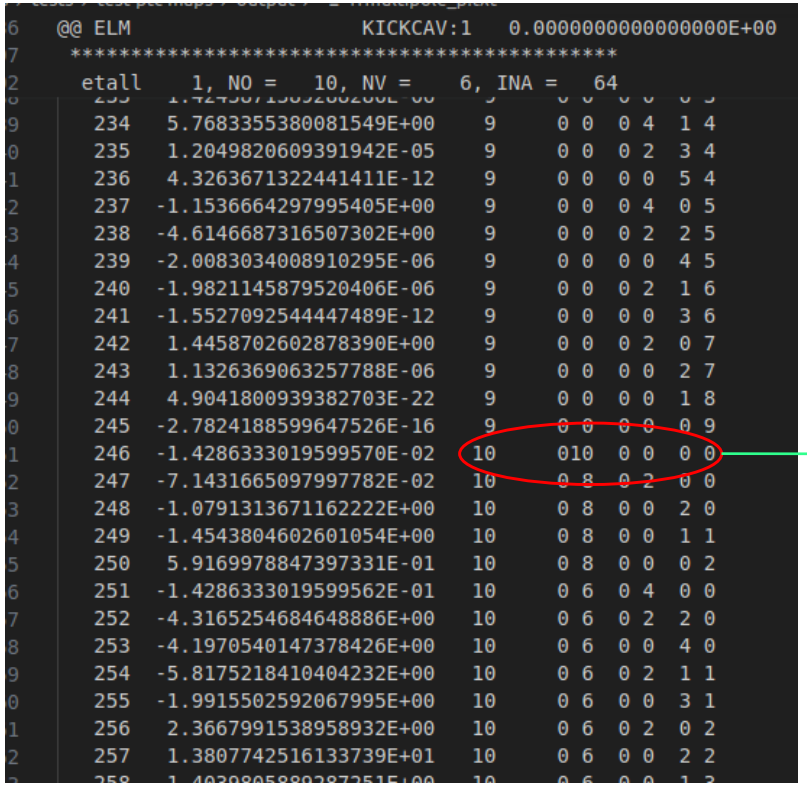

Before




After

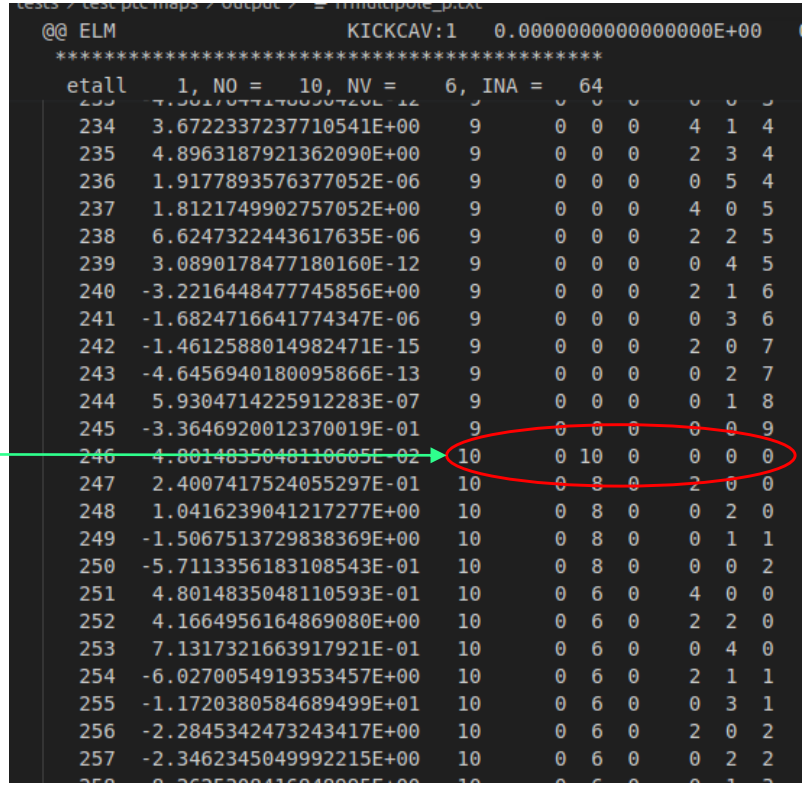
# DA Map Output

- Expand the monomial number output



```
@@ ELM                    KICKCAV:1   0.0000000000000000E+00
 ******************************************
  etall    1, NO =   10, NV =    6, INA =   64
   234   5.7683355380081549E+00    9      0 0  0 4  1 4
   235   1.2049820609391942E-05    9      0 0  0 2  3 4
   236   4.3263671322441411E-12    9      0 0  0 0  5 4
   237  -1.1536664297995405E+00    9      0 0  0 4  0 5
   238  -4.6146687316507302E+00    9      0 0  0 2  2 5
   239  -2.0083034008910295E-06    9      0 0  0 0  4 5
   240  -1.9821145879520406E-06    9      0 0  0 2  1 6
   241  -1.5527092544447489E-12    9      0 0  0 0  3 6
   242   1.4458702602878390E+00    9      0 0  0 2  0 7
   243   1.1326369063257788E-06    9      0 0  0 0  2 7
   244   4.9041800939382703E-22    9      0 0  0 0  1 8
   245  -2.7824188599647526E-16    9      0 0  0 0  0 9
   246  -1.4286333019599570E-02   10     010  0 0  0 0
   247  -7.1431665097997782E-02   10      0 8  0 2  0 0
   248  -1.0791313671162222E+00   10      0 8  0 0  2 0
   249  -1.4543804602601054E+00   10      0 8  0 0  1 1
   250   5.9169978847397331E-01   10      0 8  0 0  0 2
   251  -1.4286333019599562E-01   10      0 6  0 4  0 0
   252  -4.3165254684648886E+00   10      0 6  0 2  2 0
   253  -4.1970540147378426E+00   10      0 6  0 0  4 0
   254  -5.8175218410404232E+00   10      0 6  0 2  1 1
   255  -1.9915502592067995E+00   10      0 6  0 0  3 1
   256   2.3667991538958932E+00   10      0 6  0 2  0 2
   257   1.3807742516133739E+01   10      0 6  0 0  2 2
```

Before



```
@@ ELM                    KICKCAV:1   0.0000000000000000E+00
 ******************************************
  etall    1, NO =   10, NV =    6, INA =   64
   234   3.6722337237710541E+00    9      0  0  0    4  1  4
   235   4.8963187921362090E+00    9      0  0  0    2  3  4
   236   1.9177893576377052E-06    9      0  0  0    0  5  4
   237   1.8121749902757052E+00    9      0  0  0    4  0  5
   238   6.6247322443617635E-06    9      0  0  0    2  2  5
   239   3.0890178477180160E-12    9      0  0  0    0  4  5
   240  -3.2216448477745856E+00    9      0  0  0    2  1  6
   241  -1.6824716641774347E-06    9      0  0  0    0  3  6
   242  -1.4612588014982471E-15    9      0  0  0    2  0  7
   243  -4.6456940180095866E-13    9      0  0  0    0  2  7
   244   5.9304714225912283E-07    9      0  0  0    0  1  8
   245  -3.3646920012370019E-01    9      0  0  0    0  0  9
   246   4.8014835048110605E-02   10      0 10  0    0  0  0
   247   2.4007417524055297E-01   10      0  8  0    2  0  0
   248   1.0416239041217277E+00   10      0  8  0    0  2  0
   249  -1.5067513729838369E+00   10      0  8  0    0  1  1
   250  -5.7113356183108543E-01   10      0  8  0    0  0  2
   251   4.8014835048110593E-01   10      0  6  0    4  0  0
   252   4.1664956164869080E+00   10      0  6  0    2  2  0
   253   7.1317321663917921E-01   10      0  6  0    0  4  0
   254  -6.0270054919353457E+00   10      0  6  0    2  1  1
   255  -1.1720380584689499E+01   10      0  6  0    0  3  1
   256  -2.2845342473243417E+00   10      0  6  0    2  0  2
   257  -2.3462345049992215E+00   10      0  6  0    0  2  2
```

After

# PTC Setup

MADX-PTC (global setup)

- **sector_nmul_max, sector_nmul**: (≠ ➠ **segfault**…)
  default in PTC is 22
  default in MADX-PTC
  k0 ➠ snm = 3 for n=0
  kn, kns ➠ **snm = 3,4,5** for n=1,2,3
  (**inconsistent**: **should have snm = 4,5,6**).
  knl[1..n], ksl[1..n] ➠ snm = n+2 for n=1..

- Fix sector_nmul_max inconsistency reported here
- Added warning when sector_nmul_max < order + 1
- Enable integration order 8 (method = 8) (validified in same report as above)
- Fix model check in quadrupole so that unnecessary warning is not emitted

- Add to ptc_setswitch in MAD-X, called nocharge, which defaults to true.
- Stop charge defaulting to 1 when nocharge is false and retrieve charge from the beam.

# Fringe Maps – Current

How to activate the fringe maps in PTC **(it's complicated)**

- A checkmark (✓) indicates that for the map (row) to run, the column title must be true. Therefore, to activate the map, all checkmarks on the same row need to be true.
- A slash (/) means the map will be run independent of the column, but if the column title is false, the map will have no effect.
- Greyed out ( ) means that this attribute (column) has absolutely no effect on the map.

| Map | Kill_e(nt/xi) = False | ptc_set_switch fringe = True | El%fringe 00 → 0, 01 → 1, 10 → 2, 11 → 3 | Bend_fringe = True | k0 ≠ 0 | k1 ≠ 0 | nmul > 1 | fint(x) > 0 | hgap > 0 | Exact = True |
|---|---|---|---|---|---|---|---|---|---|---|
| fringe_dipole | ✓ | | | ✓ | / | | | ✓ | / | ✓ |
| multipole_fringe | ✓ | ✓ | or $x1$ | | | | ✓ | ✓ | | |
| fringe2quad | ✓ | | $1x$ | | | / | | ✓ | | |

fint > 0 and fintx > 0 → Error

Highest multipole in the fringe → 2

El%fringe > 3 equivalent to el%fringe = 0

# Fringe Maps – Proposed (MAD-NG)

How to activate the fringe maps in PTC (it's complicated)

- A checkmark (✓) indicates that for the map (row) to run, the column title must be true.
  Therefore, to activate the map, all checkmarks on the same row need to be true.
- A slash (/) means the map will be run independent of the column, but if the column title is false, the map will have no effect.
- Greyed out ( ) means that this attribute (column) has absolutely no effect on the map.

| Map | Kill_e(nt/xi)<br>= False | ptc_set_switch<br>fringe = True | El%fringe<br>000 → 0<br>001 → 1<br>010 → 2 100 → 4 | Bend_fringe<br>= True | k0<br>≠ 0 | k1<br>≠ 0 | nmul<br>> 1 | fint(x)<br>> 0 | hgap<br>> 0 | Exact<br>= True | el%frngmax<br>> 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| fringe_dipole | ✓ | | $xx1$ | | / | | | / | / | ✓ | |
| multipole_fringe | ✓ | ✓ | or $x1x$ | | | | ✓ | | | | / |
| fringe2quad | ✓ | | $1xx$ | | | / | | | | | |

Highest multipole in the fringe → el%frngmax (default = 2)

```
fringe_none    = 0
fringe_bend    = 1
fringe_mult    = 2
fringe_qsad    = 6
fringe_rfcav   = 8

fringe_comb    = 3
fringe_combqs  = 7
fringe_all     =-1
```

# Fringe Maps – Proposal (MAD-NG)

How to activate the fringe maps in PTC (it's complicated)

- A checkmark (✓) indicates that for the map (row) to run, the column title must be true. Therefore, to activate the map, all checkmarks on the same row need to be true.
- A slash (/) means the map will be run independent of the column, but if the column title is false, the map will have no effect.
- Greyed out ( ) means that this attribute (column) has absolutely no effect on the map.

| Map | Kill_e(nt/xi) = False | ptc_set_switch fringe = -> | El%fringe 0000 → 0 0001 → 1 0010 → 2 … | k0 ≠ 0 | k1 ≠ 0 | nmul > 1 | fint(x) > 0 | hgap > 0 | Exact = True | el%frngmax > 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| fringe_dipole | ✓ | $xxx1$ | $xxx1$ | / | | | / | / | ✓ | |
| multipole_fringe | ✓ | $xx1x$ | $xx1x$ | | | ✓ | | | | / |
| fringe2quad | ✓ | $x1xx$ | $x1xx$ | | / | | | | | |
| rfcav_fringe | ✓ | $1xxx$ | $1xxx$ | | | | | | | |

ptc_set_switch, fringe = fringe_all; (default)

# Fringe and Pole Face

- Speed-up PTC fringe fields · Issue #1160
  - Sets n_wedge = 0, instead of 5, this will save 150 yrotation done for each element
  - Running ptc_normal through 20 rbends 100 times, n_wedge = 0 ~ 5x faster
- Fringe executed twice on exit of magnet in PTC · Issue #1135 – Comment out a single line
- Add and connect the following:
  - Sbend
    - f1, f2 → fringe2quad
  - Rbend
    - f1, f2 → fringe2quad
  - Quadrupole, Sextupole, Octupole
    - e1, e2
    - h1, h2 → face
    - fint, fintx, hgap → fringe_dipole
    - f1, f2 → fringe2quad
  - Solenoid
    - fint, fintx, hgap → fringe_dipole
    - f1, f2 → fringe2quad
- Connect the following for the rbend[1]:
  - tilt
  - fint(x), hgap
  - h1, h2
  - kill_ent_fringe, kill_exi_fringe
  - fringe, bend_fringe
  - nst
  - aperture
  - knl, ksl

[1] These attributes were accepted, yet had no effect

# Normal and Skew Multipole Components

- From the MAD-X manual:

  To preserve the reference orbit of straight elements, the dipole components for those elements are ignored and must be specified as zero: `KNL(0)=0`, `KSL(0)=0`.

- We did not think this was good enough argument to remove it from knl and ksl i.e., multipole and rfmultipole. Therefore, we now allow it.

- Stop removing corresponding strength in knl and ksl.
  Instead, behaviour is to follow the philosophy of the (s/r)bend.

Quadrupole:
$k1_{final} = k1 + knl[2]/l$
$k1s_{final} = k1s + ksl[2]/l$

Sextupole:
$k2_{final} = k2 + knl[3]/l$
$k2s_{final} = k2s + ksl[3]/l$

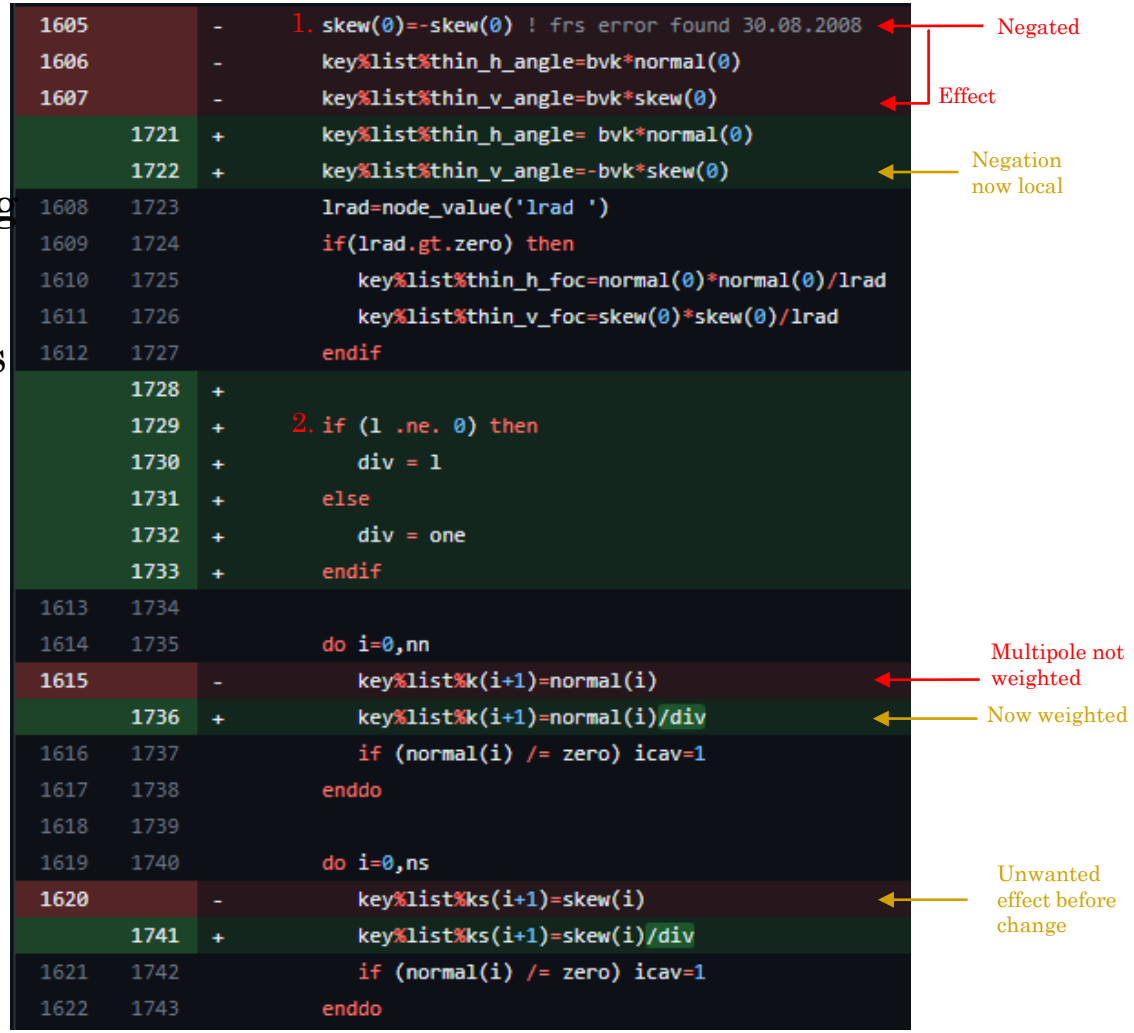
Octupole:
$k3_{final} = k3 + knl[4]/l$
$k3s_{final} = k3s + ksl[4]/l$

In above, if $l == 0$, then $l$ is set to 1.

- Stop MADX-PTC reversing the sign of the dipole skew component

# RF Multipole

1. Fix ksl[1] in rfmultipole
2. Fix thick rfmultipole (l≠0) from not dividing knl and ksl by length (l)
3. Add no_cavity_totalpath to match attributes of rfcavity[1].




# Crab Cavity

1. Add no_cavity_totalpath to match attributes of rfcavity.
2. Add harmon to match attributes of rfcavity.

[1] This attribute was dealt with in layer between MADX and PTC but not in MAD-X element dictionary

# Sbend and Rbend Skew Components

- MADX-PTC attempts to convert the skew component to a tilt, like in MAD-X.
- This calculation is performed

The angle used by PTC

$$\theta = l \times \sqrt{\left(\frac{\mathrm{knl}[1]}{l}\right)^2 + \left(\frac{\mathrm{ksl}[1]}{l}\right)^2} + \mathrm{angle}$$

The user input

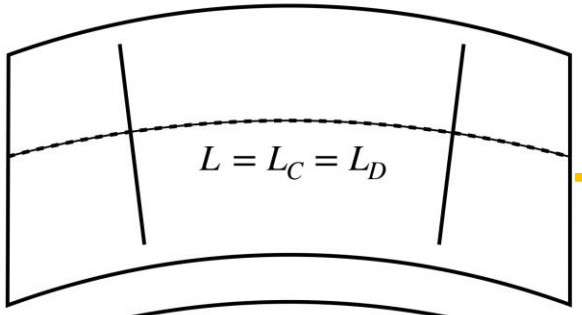
- If the square root is non-zero, below is performed

$$\mathrm{tilt} = \mathrm{atan2}\left(-\frac{\mathrm{ksl}[1]}{l}, \frac{\mathrm{knl}[1]}{l}\right)$$

- Problem 1: This was before the modification to MAD-X that allowed $k_{0l} \neq \theta$
- Problem 2: Formula is wrong.
- **Solution: Remove these calculations and let PTC integrate this component in the kick.**

# RBends (local frame)

RBEND ($L_D$)
true_rbend = false
[sbend_thick, sbend_kick]
$E_1 = \theta/2 + e_1$ ; $E_2 = \theta/2 + e_2$

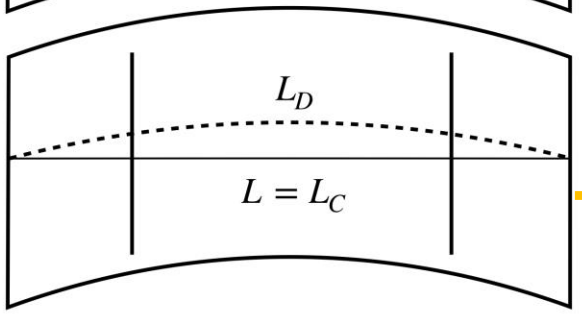



[S/R]BENDs input length is always $L_D$

Multipoles length are always related to $L$

→ Curved rbend

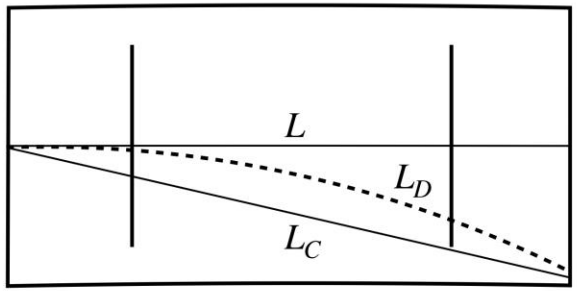
RBEND ($L_C$)
true_rbend = true
[rbend_thick, rbend_kick]
$E_1 = \theta/2 + e_1$ ; $E_2 = \theta/2 + e_2$

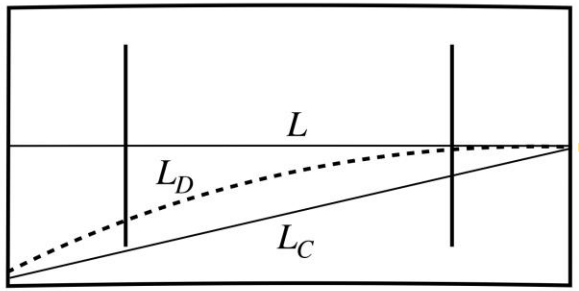



$e_1$ and $e_2$ are user's pole face angles
$E_1$ and $E_2$ are angles used internally
$L_C = L_D \operatorname{sinc} \theta/2$
$L = L_C \cos(\theta/2 - \alpha)$

→ Straight rbend

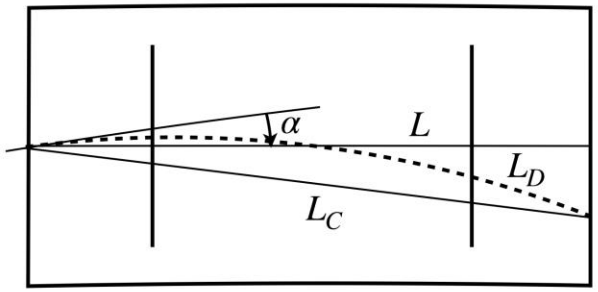
RBEND ($L$)
true_rbend = true
[rbend_thick, rbend_kick, **patch**]
$E_1 + E_2 = \theta$, def. only $e_1$ or $e_2$
Left: $E_1 = e_1 = 0$ ; $E_2 = \theta$
Right: $E_2 = e_2 = 0$ ; $E_1 = \theta$

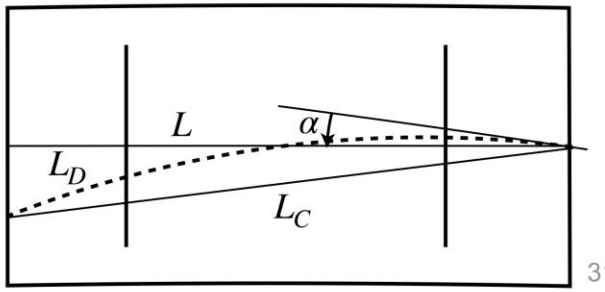



→ True Parallel rbend

$E_1 + E_2 = \theta$, def. only $e_1$ or $e_2$
Left: $E_1 = e_1 = \alpha$ ; $E_2 = \theta - \alpha$
Right: $E_2 = e_2 = \alpha$ ; $E_1 = \theta - \alpha$




31

From here

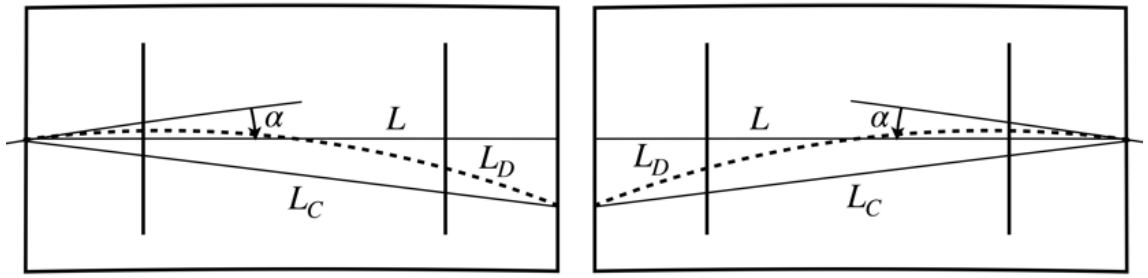


Laurent Deniau, CERN BE/ABP, 1211 Geneva 23, laurent.deniau@cern.ch

# Restore True Parallel Rbend

## Current

- To activate straight rbend (see definition here):
  - ptcrbend = true and truerbend = false
  - e1, e2 are considered as expected
- To activate true parallel rbend
  - ptcrbend = true and truerbend = true
  - If e1≠0 and e2≠0 sets e2 to 0
  - This setup goes to straight rbend **(misuse)**
  - Therefore, **no** true parallel rbend



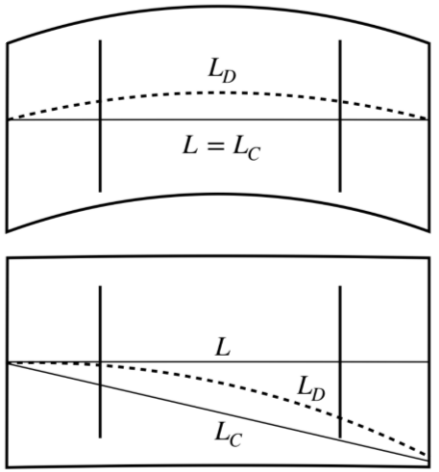

True Parallel Rbend with angle $\alpha$

## Proposed

- To activate straight rbend (see definition here):
  - ptcrbend = true and truerbend = false
  - e1, e2 are considered as expected
- To activate true parallel rbend
  - ptcrbend = true and truerbend = true
  - If e1 > 2π, patch will be performed on the entry of the rbend
  - If e2 > 2π, patch will be performed on the exit of the rbend
  - If e1 > 2π and e2 > 2π, an error is emitted (like in MAD-NG)

# Rbend Length Scaling

- For knl and ksl components, PTC/MAD-NG integrates using $L$ as the length
- PTC only accepts non-integrated strengths, i.e., MAD-X needs to convert knl and ksl components for PTC.
- MADX-PTC is unaware of $L$ so divides the integrated strength by $L_D$.
- Therefore, the integrated strength in PTC is $\frac{\text{knl}[n+1]}{L_D} L$[1]
- If you use $k_n$ instead, the integrated strength is $k_n L$




The lengths and path lengths for different rbends

- **Solution: Make MADX-PTC aware of $L$ so that the integrated strength in PTC is $\frac{\text{knl}[n+1]}{L} L$**

[1] The $[n+1]$ is because of the indexing from 1.

# Solenoid

## Current

```
case(code_solenoid) ! case(9) ! PTC accepts mults
   key%magnet="solenoid"
   ks=node_value('ks ')
   if(l.ne.zero) then
      key%list%bsol=bvk*ks
   else
      ksi=node_value('ksi ')
      lrad=node_value('lrad ')
      if(lrad.eq.zero.and.ks.ne.zero) lrad=ksi/ks
      if(ksi.eq.zero.or.lrad.eq.zero) then
         key%magnet="marker"
         print*,"Thin solenoid: ",name," has no strength - set to marker"
      else
         key%list%bsol=bvk*ksi/lrad
         key%list%ls=lrad
      endif
   endif
   !VK
   CALL SUMM_MULTIPOLES_AND_ERRORS (l, key, normal_0123,skew_0123,ord_max)
```

Remove, as it could have multipoles

## Proposed

```
case(code_solenoid) ! case(9) ! PTC accepts mults
   key%magnet="solenoid"
  !VK
   CALL SUMM_MULTIPOLES_AND_ERRORS (l, key, normal_0123,skew_0123,ord_max)
   ks=node_value('ks ')
   if(l.ne.zero) then ! JG 21.04.2023 L != 0 means thick solenoid
      key%list%bsol=bvk*ks
      !JG: 18.04.2023 Allowed k0 and k0s for solenoid
      key%list%k (1)=normal_0123(0)
      key%list%ks(1)=  skew_0123(0)
   else
     ! JG 21.04.2023 L = 0 means thin solenoid, therefore needs multipole
     ! thin_h_angle, thin_v_angle, thin_h_foc, thin_v_foc
      ksi=node_value('ksi ')
      lrad=node_value('lrad ')
      if(lrad.eq.zero.and.ks.ne.zero) lrad=ksi/ks
      key%list%thin_h_angle= bvk*normal_0123(0)
      key%list%thin_v_angle=-bvk*skew_0123(0)
      if(lrad.gt.zero) then
         key%list%thin_h_foc=normal_0123(0)*normal_0123(0)/lrad
         key%list%thin_v_foc=skew_0123(0)*skew_0123(0)/lrad
         key%list%bsol=bvk*ksi/lrad
      endif
      key%list%ls=lrad
   endif
```

thick

thin

Add dipole component to thick and thin solenoid

# Multipole Solenoid Component

- Add and connect solenoid component to a multipole
  - Changes nothing by default
  - If ksi specified but lrad is not, a warning is emitted

```
! JG 18.04.2023 Added solenoid to multipole
      ksi=node_value('ksi ')
      if (ksi.ne.zero.and.lrad.ne.zero) then
        key%list%bsol=bvk*ksi/lrad
        key%list%ls=lrad
      elseif (ksi.ne.zero) then
        write(msg,*) "Multipole solenoid component ignored as lrad=0, &
                     the combination of ks and ksi is not supported in a multipole. &
                     Please use the SOLENOID element with knl and ksl or specify lrad"
        call fort_warn("ptc_input",msg(:len_trim(msg)))
      endif
```

# Solenoid in MAD-X

- Solenoid in MAD-X Thin-Lens Tracking – Implementation.pdf (cern.ch)

- Affects the t variable in trsol (MAD-X track) when l=0

- Testing the thin map vs the thick map in MAD-NG[1] we find the error by order of the t variable to be[1]:
  - $0^{th}$ order – PTC eq. **57** times **better** than MAD-X eq. (PTC error $4.9 \times 10^{-11}$)
  - $1^{st}$ order – PTC eq. **25** times **better** than MAD-X eq. (PTC error $2.8 \times 10^{-7}$)
  - $2^{nd}$ order – PTC eq. **6** times **better** than MAD-X eq. (PTC error $4.7 \times 10^{-4}$)
  - $3^{rd}$ order – PTC eq. **8** times **better** than MAD-X eq. (PTC error $9.4 \times 10^{-4}$)
  - $4^{th}$ order – No difference (Error $7.5 \times 10^{-2}$)

- The wrong equation is being used by MAD-X or PTC – Given above, I would suggest MAD-X is incorrect.

[1]Performed in 5D with $x = 3 \times 10^{-3}, px = -2 \times 10^{-4}, y = -2 \times 10^{-3}, py = 3 \times 10^{-4}, t = 0, pt = 2 \times 10^{-5}$, default beam, ksi = 0.1

# Equation Direct Comparison

Construction of Nonlinear Symplectic Six-Dimensional Thin - Lens Maps by Exponentiation K. Heinemann, G. Ripken, F. Schmidt 30 Oct 1995 (cern.ch)

PTC/MAD-NG

$$\Delta\Theta = \frac{H(s_0)\cdot\Delta s}{[1+f(p_\sigma^i)]}$$

$$\sigma^f = \sigma^i - \frac{f'(p_\sigma^i)}{[1+f(p_\sigma^i)]}\cdot\Delta\Theta \times\left\{\frac{1}{2}H(s_0)\cdot\left[(x^i)^2+(z^i)^2\right]+\left[p_x^i\cdot z^i - p_z^i\cdot x^i\right]\right\}; \qquad (4.35b)$$

$$\hat{\sigma}^f = \hat{\sigma}^i - \frac{f'(\hat{p}_\sigma^i)}{[1+f(\hat{p}_\sigma^i)]^2}\cdot[H(s_0)]^2\cdot\Delta s\cdot\frac{1}{2}\left[(\hat{x}^i)^2+(\hat{z}^i)^2\right];$$

$$\sigma^f = \hat{\sigma}^f + \left\{\hat{x}^f\cdot\hat{p}_z^f - \hat{z}^f\cdot\hat{p}_x^f\right\}\cdot\frac{H(s_0)\cdot\Delta s}{\left[1+f(\hat{p}_\sigma^f)\right]^2}\cdot f'(\hat{p}_\sigma^f);$$

MAD-X

A symplectic six-dimensional thin-lens formalism for tracking, G. Ripken, F. Schmidt 5 April 1995 (cern.ch)

# Patches

- Fix changeref from throwing memory access outside program range error

```
1354  1449        key%magnet="CHANGEREF"
1355  1450        PATCH_ANG = zero
1356  1451        PATCH_TRANS = zero
1357        -     call get_node_vector('patch_ang ',3,patch_ang)
1358        -     call get_node_vector('patch_trans ',3,patch_trans)
      1452  +     np = 3
      1453  +     call get_node_vector('patch_ang ',np,patch_ang)
      1454  +     call get_node_vector('patch_trans ',np,patch_trans)
```

- Add shortcuts to patches so that identity transformations are not performed – increasing accuracy

```
@@ -390,6 +390,8 @@ SUBROUTINE ROT_YZP(A,X,b,EXACT,ctime)
390  390        real(dp),INTENT(IN):: A,b
391  391        LOGICAL(lp),INTENT(IN):: EXACT,ctime
392  392
     393  +       if (A == 0.0_dp) return
     394  +
393  395        call PRTP("ROT_YZ:0", X)
394  396
395  397        CALL ALLOC(XN,6)
@@ -456,6 +458,7 @@ SUBROUTINE TRANSP(A,X,b,EXACT,ctime)
456  458        real(dp),INTENT(IN):: A(3),b
457  459        LOGICAL(lp),INTENT(IN):: EXACT,ctime
458  460
     461  +     if (A(1) == 0.0_dp .and. A(2) == 0.0_dp .and. A(3) == 0.0_dp) return
```

```
@@ -527,6 +530,7 @@ SUBROUTINE ROT_XYP(A,X)
527  530        real(dp),INTENT(IN):: A
528  531        real(dp)           :: cosa, sina
529  532
     533  +     if (A == 0.0_dp) return
530  534        call PRTP("ROT_XY:0", X)
531  535
532  536        cosa = COS(A)
@@ -610,6 +614,8 @@ SUBROUTINE ROT_XZP(A,X,b,EXACT,ctime)
610  614        real(dp) sina, cosa, tana
611  615        LOGICAL(lp),INTENT(IN):: EXACT,ctime
612  616
     617  +       if (A == 0.0_dp) return
     618  +
613  619        call PRTP("ROT_XZ:0", X)
614  620
615  621        IF(EXACT) THEN
```

# Tilting Elements

- Connect the tilt for the elseparator (currently does nothing).
- Connect the tilt for the rbend (already mentioned).

# Thanks for Listening!

# Solenoid Equations (PTC/MAD-NG)

$$\underline{\hat{A}}_{01} = \Delta s \cdot \frac{H^2}{[1+f(\hat{p}_\sigma)]} \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix} \tag{4.29a}$$

$$\longrightarrow \underline{M}_0 = \left[\underline{1} + \underline{\hat{A}}_{01}\right] \cdot \begin{pmatrix} \cos(\Delta\Theta) & 0 & +\sin(\Delta\Theta) & 0 \\ 0 & \cos(\Delta\Theta) & 0 & +\sin(\Delta\Theta) \\ -\sin(\Delta\Theta) & 0 & \cos(\Delta\Theta) & 0 \\ 0 & -\sin(\Delta\Theta) & 0 & \cos(\Delta\Theta) \end{pmatrix} . \tag{4.33}$$

$$\vec{y}_0 = \begin{pmatrix} x \\ p_x \\ z \\ p \end{pmatrix} \longrightarrow \vec{y}_0^{\,f} = \underline{M}_0 \, \vec{y}_0^{\,i} \; ; \tag{4.35a}$$

$$\sigma^f = \sigma^i - \frac{f'(p_\sigma^i)}{[1+f(p_\sigma^i)]} \cdot \Delta\Theta \times \left\{ \frac{1}{2} H(s_0) \cdot \left[(x^i)^2 + (z^i)^2\right] + \left[p_x^i \cdot z^i - p_z^i \cdot x^i\right] \right\} \; ; \tag{4.35b}$$

$$p_\sigma^f = p_\sigma^i \tag{4.35c}$$

with

$$\Delta\Theta = \frac{H(s_0) \cdot \Delta s}{[1+f(p_\sigma^i)]} \tag{4.35d}$$

[Construction of Nonlinear Symplectic Six-Dimensional Thin - Lens Maps by Exponentiation K. Heinemann, G. Ripken, F. Schmidt 30 Oct 1995 (cern.ch)]

# Solenoid Equations (MAD-X)

$$\hat{x}^f = \hat{x}^i ;$$

$$\hat{p}_x^f = \hat{p}_x^i - \frac{\hat{x}^i}{[1+f(\hat{p}_\sigma^i)]} \cdot [H(s_0)]^2 \cdot \Delta s ;$$

$$\hat{z}^f = \hat{z}^i ;$$

$$\hat{p}_z^f = \hat{p}_z^i - \frac{\hat{z}^i}{[1+f(\hat{p}_\sigma^i)]} \cdot [H(s_0)]^2 \cdot \Delta s ;$$

$$\hat{\sigma}^f = \hat{\sigma}^i - \frac{f'(\hat{p}_\sigma^i)}{[1+f(\hat{p}_\sigma^i)]^2} \cdot [H(s_0)]^2 \cdot \Delta s \cdot \frac{1}{2}\left[(\hat{x}^i)^2 + (\hat{z}^i)^2\right] ;$$

$$\hat{p}_\sigma^f = \hat{p}_\sigma^i .$$

Extra terms in **t** ($\sigma^f$)

$$x^f = \hat{x}^f \cdot \cos\Delta\Theta + \hat{z}^f \cdot \sin\Delta\Theta ;$$

$$p_x^f = \hat{p}_x^f \cdot \cos\Delta\Theta + \hat{p}_z^f \cdot \sin\Delta\Theta ;$$

$$z^f = -\hat{x}^f \cdot \sin\Delta\Theta + \hat{z}^f \cdot \cos\Delta\Theta ;$$

$$p_z^f = -\hat{p}_x^f \cdot \sin\Delta\Theta + \hat{p}_z^f \cdot \cos\Delta\Theta ;$$

$$\sigma^f = \hat{\sigma}^f + \left\{\hat{x}^f \cdot \hat{p}_z^f - \hat{z}^f \cdot \hat{p}_x^f\right\} \cdot \frac{\int_{s_0-0}^{s_0+0} d\tilde{s} \cdot H(\tilde{s})}{\left[1+f(\hat{p}_\sigma^f)\right]^2} \cdot f'(\hat{p}_\sigma^f)$$

$$= \hat{\sigma}^f + \left\{\hat{x}^f \cdot \hat{p}_z^f - \hat{z}^f \cdot \hat{p}_x^f\right\} \cdot \frac{H(s_0)\cdot\Delta s}{\left[1+f(\hat{p}_\sigma^f)\right]^2} \cdot f'(\hat{p}_\sigma^f) ;$$

$$p_\sigma^f = \hat{p}_\sigma^f$$

[A symplectic six-dimensional thin-lens formalism for tracking, G. Ripken, F. Schmidt 5 April 1995 (cern.ch)]

# List of Bug Fixes

1. Multiple Fringes on exit of magnet in PTC · Issue #1135 – Comment out a single line
2. Allow tilt, fint, hgap, h1, h2, kill_fringe, fringe, etc… to be communicated to the rbend.
3. If ksi == 0 or lrad == 0, stop solenoid being turned into a marker, due to possible multipole components.
4. Fixed ksl[1] (k0s) in rfmultipole
5. Fix tilt not being communicated to PTC in elseparator
6. Fix sector_nmul_max inconsistency reported here
7. Fix changeref from throwing ++ memory access outside program range ++ error
8. Fix thick rfmultipole from not dividing knl and ksl by length
9. Stop MAD-X-PTC attempt to convert the skew component to a tilt

# List of Improvements

1. Speed-up PTC fringe fields · Issue #1160
    1. n_wedge = 0
    2. instead of 5, this will save 5*(8+7)*2 = 150 yrotation done for each element
2. Flush the DA map output (cleans up debugging etc…)
3. Expand the monomial number output i.e. 10 0 0 0 0 0 not 100 0 0 0 0
4. Allow the entry fint and exit fint to be different
5. Add f1 and f2 to enable the qsad_fringe
6. Add frngmax to dictate the highest multipole used in the multipole fringe
7. Add all the fringe and face parameters to sbend, rbend, quadrupole, sextupole, octupole
8. Add Yoshida method 8
9. Stop fint/fintx = 0, killing the entry/exit fringe
10. If fringe is set to > 3, then it is reduced to 3, assuming all fringes would like to be activated in PTC
11. For sextupole and octupole, stop corresponding strength in knl and ksl being removed (instead added to $k_n$ and $k_{ns}$)
12. Allowed knl[1] and ksl[1] for multipole integration.
13. Added solenoid component to a multipole
14. Add no_cavity_totalpath to crab-cavity and rf-multipole (therefore add fix for harmon)
15. Add a tilt to the multipole
16. Restore true parallel rbend
17. Allow Charge to work in PTC, therefore add option to MAD-X called nocharge, which defaults to true.
18. Fix model check in quadrupole so that unnecessary warning is not emitted
19. Add shortcuts to patches so that identity transformations are not performed