Solenoid in MAD-X Thin-Lens Tracking – Implementation

- Straightforward implementation of formulae from **G.Ripken, F.Schmidt**; *A symplectic Six-Dimensional Thin-Lens Formalism For Tracking*; CERN/SL/95-12(AP) and DESY 95-063; April 5, 1995; p.24–31
- Fortran 77 subroutine (simply because it was the fastest solution to be implemented)
- Not in Matrix-Notation (matrix formalism is nice to read, but would create only unnecessary overhead in this context, the TWISS module however uses matrix notation, which is ok there)
- Full 6D formulae, however there is a drawback:
  5D (transverse coordinates + $\Delta p/p$) only treated correctly if $\delta = \Delta p/p$ is absorbed into p_T variable.

Solenoid in MAD-X Thin-Lens Tracking – Implementation, Formulae

$$x^f = x^i \cdot \cos \Delta\Theta + y^i \cdot \sin \Delta\Theta,$$

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

$$y^f = -x^i \cdot \sin \Delta\Theta + y^i \cdot \cos \Delta\Theta,$$

$$p_y^f = -\hat{p}_x^f \cdot \sin \Delta\Theta + \hat{p}_y^f \cdot \cos \Delta\Theta,$$

$$\sigma^f = \hat{\sigma}^f + \left\{ x^i \cdot \hat{p}_y^f + y^i \cdot \hat{p}_x^f \right\} \cdot \frac{H(s_0) \cdot \Delta s}{[1 + f(p_\sigma^i)]^2} \cdot f'(p_\sigma^i)$$

$$p_\sigma^f = p_\sigma^i$$

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

$$\hat{p}_y^f = p_y^i - \frac{y^i}{[1 + f(p_\sigma^i)]} \cdot H(s_0)^2 \cdot \Delta s,$$

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

$$H(s_0) = \frac{1}{2} \frac{e}{p_0 \cdot c} \cdot B_s(0, 0, s_0) = \frac{1}{2} \cdot k_s,$$

$$\Delta\Theta = \frac{H(s_0) \cdot \Delta s}{[1 + f(p_\sigma^i)]}, \quad H(s_0) \cdot \Delta s = \frac{1}{2} k_{sl}$$

$$f(p_\sigma) = \sqrt{1 + 2p_\sigma + \beta^2 p_\sigma^2} - 1$$

$$f'(p_\sigma) = \frac{1 + \beta^2 p_\sigma}{\sqrt{1 + 2p_\sigma + \beta^2 p_\sigma^2}}$$

$$p_\sigma = \frac{p_T}{\beta}$$

$$\sigma = \beta \cdot T$$

Solenoid in MAD-X Thin-Lens Tracking – Sequence definition, THICK solenoid

```
// Define sequence, single solenoid of 10m length
lsol  = 10.;
epfac = 0.299792458;
solfield = 4.;
pbeam = 450.;
nsol  = 10;

------------------------------------------------------------

// Define a thick solenoid
ksol:= solfield*epfac/pbeam;
msol: solenoid, l:= lsol, ks:=ksol;

// Thick Sequence
myseqT : SEQUENCE, REFER= entry, l= 10.0;
mStart: MARKER, AT= 0.0;
mysol : msol,   AT= 0.0;
mEnd  : MARKER, AT= 10.0;
ENDSEQUENCE;
```

Solenoid in MAD-X Thin-Lens Tracking – Sequence definition, THIN solenoid

```
// Define sequence, single solenoid of 10m length
lsol  = 10.;
epfac = 0.299792458;
solfield = 4.;
pbeam = 450.;
nsol  = 10;
------------------------------------------------------------
// Define a thin solenoid
ksol:= solfield*epfac/pbeam;
msol: solenoid, l:= 0, lrad:= lsol/nsol, ks:=ksol, ksl:= ksol*lsol/nsol;
ddd : DRIFT, l= 10.0/nsol;

// Thin Sequence
myseq : SEQUENCE, REFER= entry, l= 10.0;
mStart: MARKER, AT= 0.0;
ddd1  : DRIFT,  AT= 0.0, l= 10.0/nsol/2;
n = 1;
while (n < nsol) {
  msol   : msol,   AT= (n-1)*lsol/nsol+10.0/nsol/2;
  ddd    : ddd,    AT= (n-1)*lsol/nsol+10.0/nsol/2;
  n = n + 1;
}
msol  : msol,   AT= (n-1)*lsol/nsol+10.0/nsol/2;
ddd2  : DRIFT,  AT= (nsol-1)*lsol/nsol+10.0/nsol/2, l= 10.0/nsol/2;
mEnd  : MARKER, AT= 10.0;
ENDSEQUENCE;
```

Solenoid in MAD-X Thin-Lens Tracking – Tests against TWISS, PTC_TRACK

```
TWISS thick solenoid:
x   =    0.01999704128229284 ;
px  =  0.0009996449677422605 ;
y   = -0.0002664585310928548 ;
py  = -1.332016701664907e-05 ;
t   = -5.000887659439013e-06 ;
pt  =                      0 ;
s   =                     10 ;

PTC_TRACK thick solenoid:
xf  =    0.01999704128051779 ;
xpf =  0.0009996449673429625 ;
yf  = -0.0002664585310692027 ;
ypf = -1.332016701132847e-05 ;
tf  =                      0 ;
tpf =                      0 ;
sf  =                      0 ;

TRACK 10 thin solenoids:
xf  =    0.01999703980108315 ;
xpf =   0.000999644967277321 ;
yf  = -0.0002664585113558872 ;
ypf = -1.332016701045381e-05 ;
tf  = -5.000884190944135e-06 ;
tpf =                      0 ;
sf  =                     10 ;
```