The European Synchrotron

Update on Accelerator Toolbox Code and Computations for the ESRF EBS

2 December 2016,
LER design workshop
LUND

B. Nash
ESRF

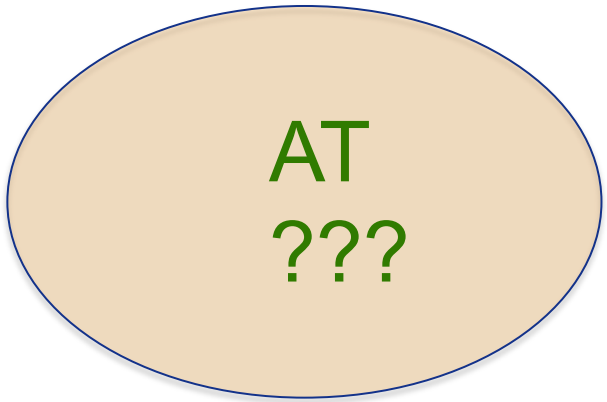
OUTLINE

- History of AT, atcollab
- Structure of AT
- integrators
- Application to ESRF EBS
- Conclusions

WIDELY USED CODES FOR ELECTRON RINGS

Elegant
APS-Borland et. al.

MAD-X/PTC
CERN- module structure
with organizers



BMAD
Cornell- D. Sagan
http://www.lepp.cornell.edu/~dcs/bmad/

(OPA, Tracy, …?)

HISTORY OF AT

AT is a Matlab based tool box for accelerator simulation
originally created by Andrei Terebilo during the late 1990's at SLAC.

He left acc. phys. 2010

Some interaction with Tracy code (Nishimura et al)
at early stage
http://inspirehep.net/record/264780?ln=en

AT is used within MML and LOCO (Portmann, Safranek),
but may also be used as a stand alone modeling code.

No one was coordinating the AT development, after Andrei stopped,

After I joined ESRF in 2009, I decided to try to coordinate AT development in an open source collaboration.

atcollab

ATCOLLAB

How to organize AT development across multiple labs, when it is already in use by so many?

Can we support an open source collaborative development for the AT code?

Needed parts of open source project:

*) Repository in version control system: SourceForge chosen with svn
*) communication amongst users and developers: (mailing list, wiki, website…)
*) documentation, bug reporting, etc
*) well defined releases

http://sourceforge.net/projects/atcollab/

considerations:
> copyright issues? (I asked A. Terebillo and he suggested we go ahead)
> branching and forking of code?
> How to find common ground and bring different development together?

ATCOLLAB COMMUNICATION

mailing list: atcollab-general@lists.sourceforge.net

27 members from 15 institutes

ESRF (5)
NSRC, Taiwan (3)
LBL (3)
IHEP, China (3)
ALBA (3)
SLAC
Australian Synchrotron
Mayo Clinic
INFN
Diamond
KIT, ANKA
SOLEIL
ELETTRA
Canadian Light Source
LAL

on Twitter:
@atcollab_tweets

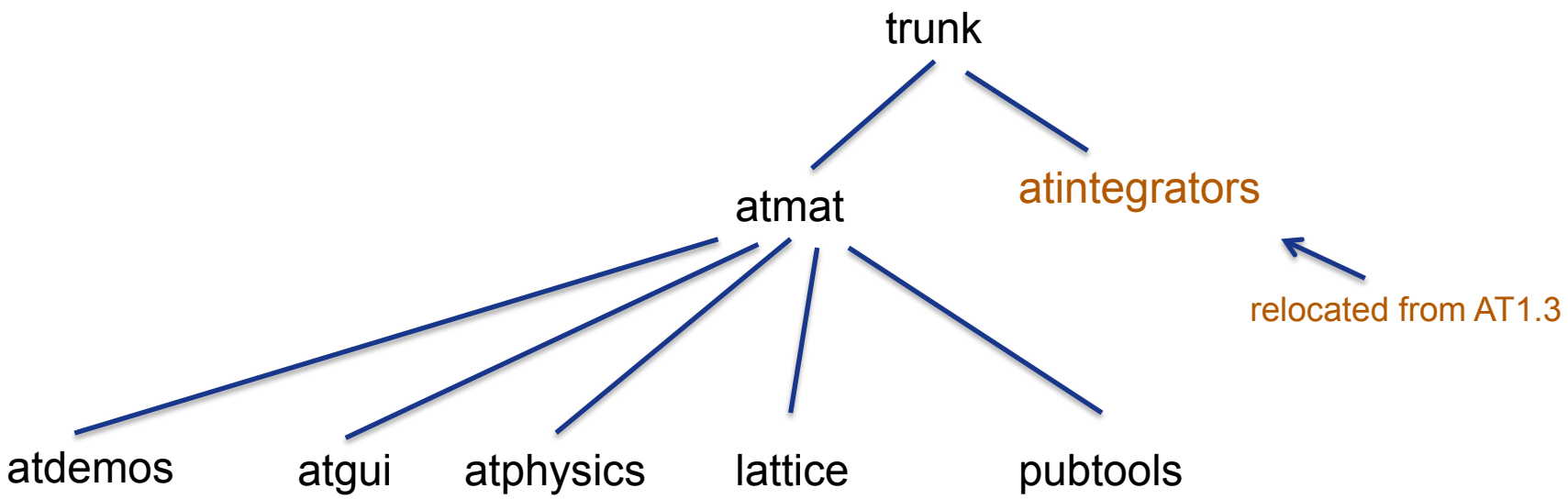
Archives and subscription info at

https://lists.sourceforge.net/lists/listinfo/atcollab-general

A better website is needed!

ATCOLLAB SVN REPOSITORY

http://sourceforge.net/projects/atcollab/



get copy of repository with:

svn checkout https://svn.code.sf.net/p/atcollab/code-0/trunk atcollab

Having repository online facilitates email discussion. Can refer to code with URL. e.g. ringpass function is here:
http://sourceforge.net/p/atcollab/code-0/HEAD/tree/trunk/atmat/attrack/ringpass.m

AT is also distributed with MML by Greg Portmann.
MiddleLayer/Release/at

If possible, we should bring these versions together!

PRINCIPLES WE FOLLOW IN OUR DEVELOPMENT

Backwards compatibility. Previous lattice construction methods (using global variables) and integrators written with AT1.3 should still work.

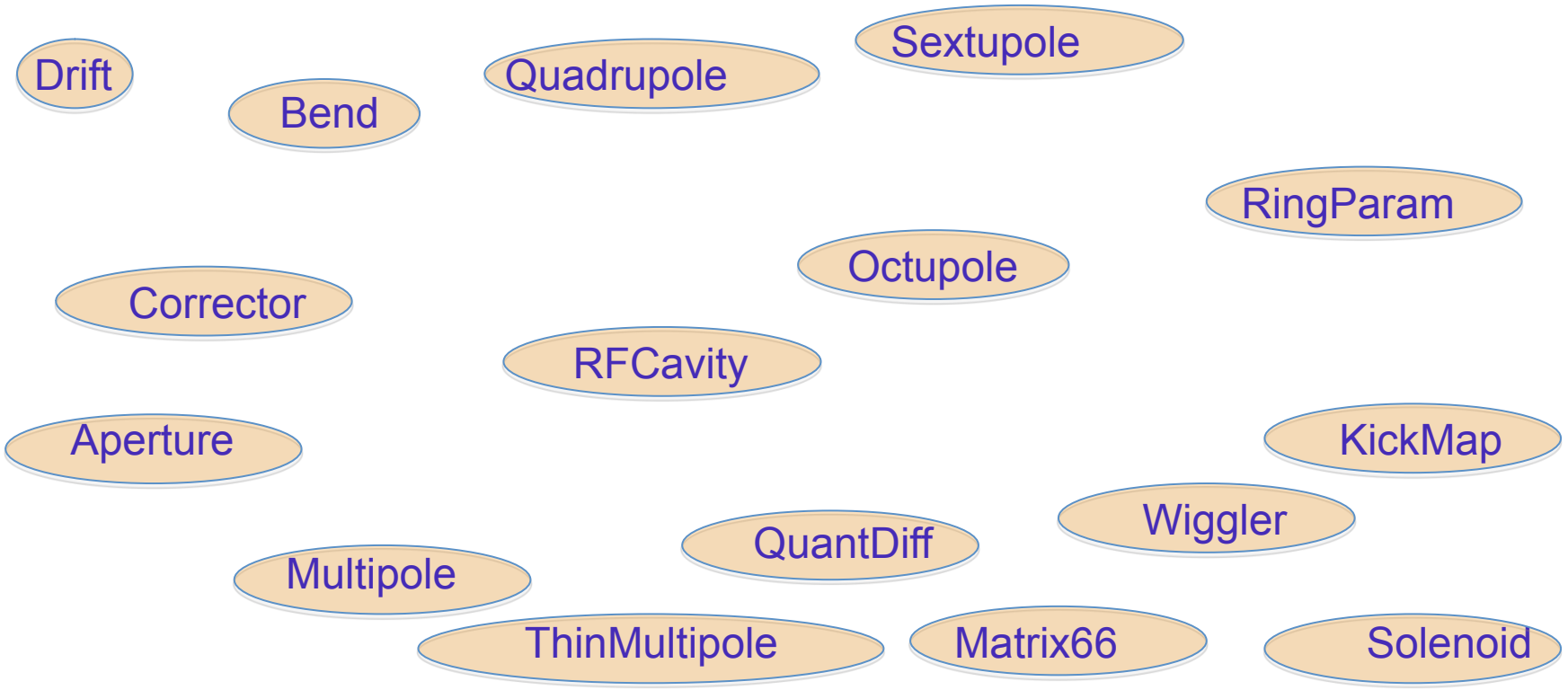
Some tests with MML integration show no issues. More tests are needed, and feedback greatly appreciated!
(Any problems with LOCO?)

Compilation should be simply done by calling atmexall.
We test this on as many different versions of Matlab, and as many platforms (Linux, Windows, Mac) as we can.

Tools that are working but not fully integrated or adequately tested go into the pubtools directory. After further testing, we migrate these tools into the other main AT directories.

CLASSES

Issue: how to write functions that identify particular elements that will work for lattices at different labs?
Solution: add a “Class” field to all elements



Example: indsext=findcells(‘ring’,’Class’,’Sextupole’)
gives indices of all sextupoles

INTEGRATORS

•AT has the following integrators in the atintegrators directory:

AperturePass.c

BendLinearPass.c

BndMPoleSymplectic4E2Pass.c

BndMPoleSymplectic4E2RadPass.c

BndMPoleSymplectic4Pass.c

BndMPoleSymplectic4RadPass.c

CavityPass.c

CorrectorPass.c

DriftPass.c

QuantDiffPass.c

EAperturePass.c

IdTablePass.c

IdentityPass.c

Matrix66Pass.c

QuadLinearPass.c

SolenoidLinearPass.c

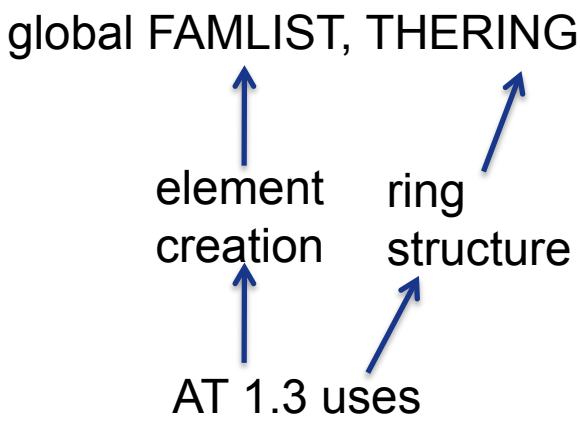
StrMPoleSymplectic4Pass.c

WiggLinearPass.c

GWigSymplecticPass.c

Others have developed different integrators.
We should find the most accurate and fastest integrators.

MOVING AWAY FROM GLOBAL VARIABLES
ELEMENT AND RING CREATION



Issue: We want to be able to create many lattices in one Matlab session

For the element creation, output elements directly. For example

```
DR1=atdrift('D1',0.7)
QF=atquadrupole('QF1',0.94,0.39,'QuadMPoleFringePass')
SH=atsextupole('SH',0,0,'StrMPoleSymplectic4Pass');
B1=atsbend('B1',0,0,0,'BndMPoleSymplectic4Pass');
```

```
QF =

      FamName: 'QF1'
   PassMethod: 'QuadMPoleFringePass'
       Length: 0.9400
        Class: 'Quadrupole'
            K: 0.3900
   PolynomB: [0 0.3900]
   PolynomA: [0 0]
   MaxOrder: 1
NumIntSteps: 10
```

INTEGRATORS

Many models out there.
We have a magnetic field.
Usually, fourth order symplectic integrator (Ruth, Forest, Yoshida…)
Depends on Hamiltonian, should be separable into two solvable parts

$$H = -(1+hx)\frac{A_s}{B\rho} - (1+\delta)hx + (1+hx)\frac{p_x^2+p_y^2}{2(1+\delta)},$$

BndMPoleSymplectic4Pass.c

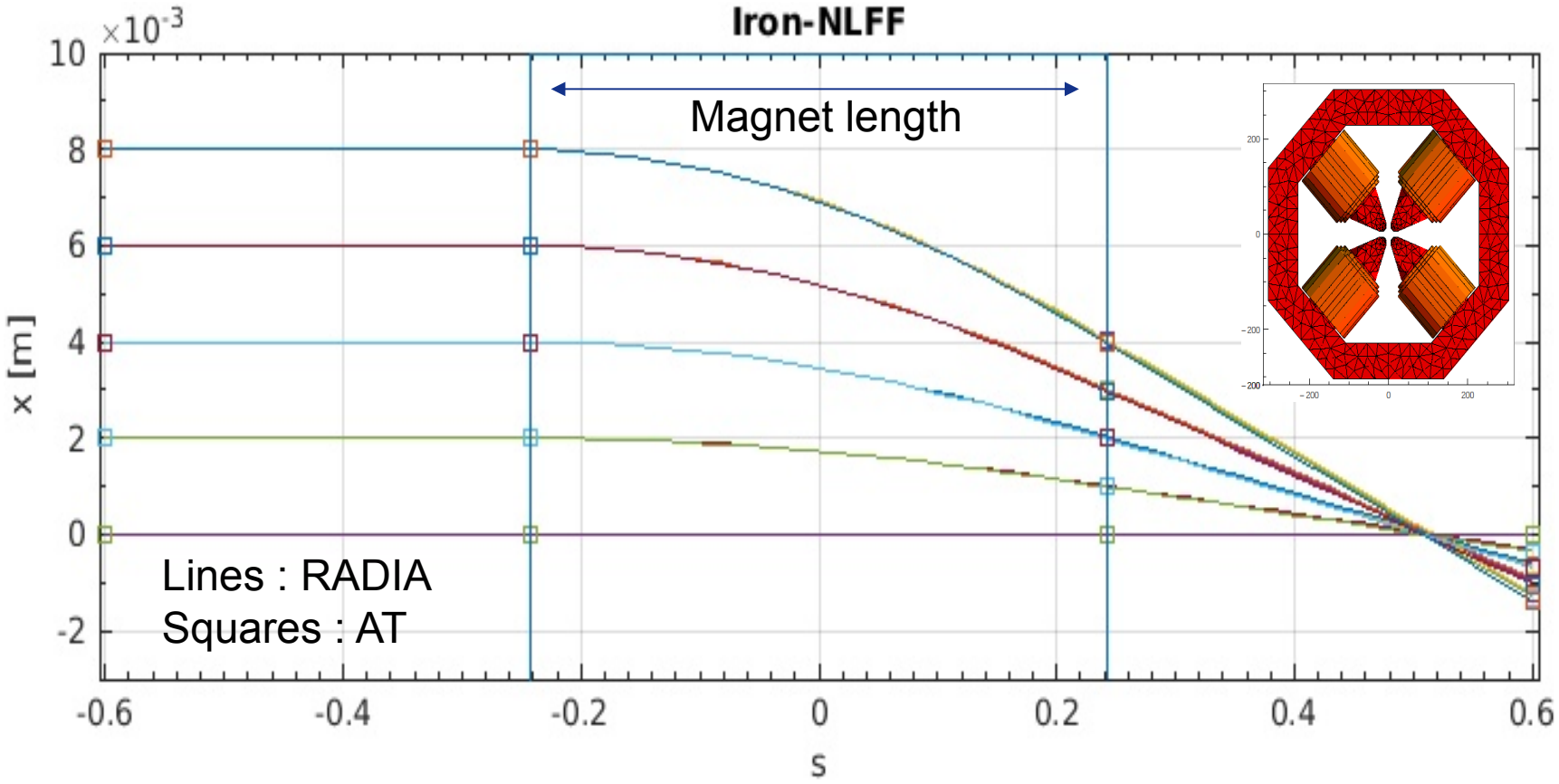
BndMPoleSymplectic4E2Pass.c

X. Huang (improves drift in dipole)

```
void StrMPoleSymplectic4Pass(double *r, double le, double *A, double *B,
        int max_order, int num_int_steps,
        double *T1, double *T2,
        double *R1, double *R2,
        double *RApertures, double *EApertures, int num_particles)
{                       int c,m;
   double norm, NormL1, NormL2;
   double *r6;
   double SL, L1, L2, K1, K2;
   SL = le/num_int_steps;
   L1 = SL*DRIFT1;
   L2 = SL*DRIFT2;
   K1 = SL*KICK1;
   K2 = SL*KICK2;

   for (c = 0;c<num_particles;c++)          {   /*Loop over particles  */
      r6 = r+c*6;
      if(!atIsNaN(r6[0])) {
         /*  misalignment at entrance  */
         if (T1) ATaddvv(r6,T1);
         if (R1) ATmultmv(r6,R1);
         /* Check physical apertures at the entrance of the magnet */
         if (RApertures) checkiflostRectangularAp(r6,RApertures);
         if (EApertures) checkiflostEllipticalAp(r6,EApertures);
         /*  integrator  */
         for (m=0; m < num_int_steps; m++) {  /*  Loop over slices */
                        r6 = r+c*6;
            norm = 1/(1+r6[4]);
            NormL1 = L1*norm;
            NormL2 = L2*norm;
            fastdrift(r6, NormL1);
            strthinkick(r6, A, B,  K1, max_order);
            fastdrift(r6, NormL2);
            strthinkick(r6, A, B, K2, max_order);
            fastdrift(r6, NormL2);
            strthinkick(r6, A, B,  K1, max_order);
            fastdrift(r6, NormL1);
         }
         /* Check physical apertures at the exit of the magnet */
         if (RApertures) checkiflostRectangularAp(r6,RApertures);
         if (EApertures) checkiflostEllipticalAp(r6,EApertures);
         /* Misalignment at exit */
         if (R2) ATmultmv(r6,R2);
         if (T2) ATaddvv(r6,T2);
      }
   }
}
```

TEST INTEGRATOR WITH RADIA MODEL



RADIA model script provided by G. Le Bec. Performs tracking of electrons in the complete field generated by the magnet (full line). Squares represent the particles tracking in AT for the equivalent hard edge, iron length, straight multipole model.

NEW TOOLS

atmatch (S. Liuzzo)

atfastring (L. Farvacque)
linear matrix, chromatic and amplitude tune shift
quantum diffusion element
used for collective effects, spin dynamics studies

atplot (L. Farvacque)
allows arbitrary function
together with synoptic

collective effects- multiparticle tracking
with impedance (S. White)

Apertures in elements for improved loss mapping
(L. Farvacque, R. Versteegen)

quantum diffusion element
QuantDiffPass
(N. Carmignani, B. Nash)

see Nash et. al. IPAC 14 for some details
http://inspirehep.net/record/1417208

recently added by S. Liuzzo:
error setting and correction tools

FUNCTIONS IN PUBTOOLS, STARTED DOCUMENTATION PROCESS

Available functions, documentation in progress, draft available in atmat/pubtools

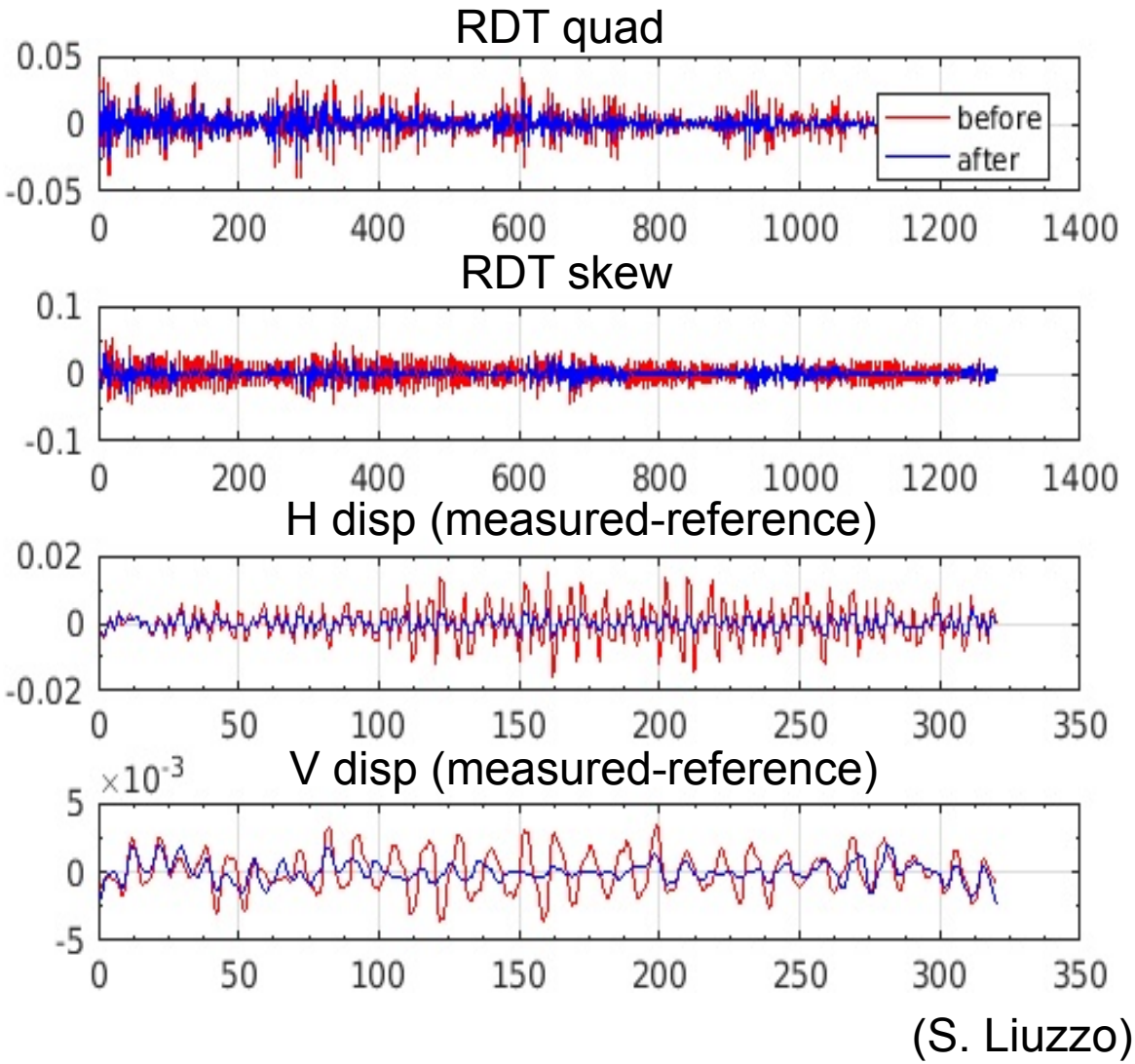

```
[...
   rcor,...          % corrected lattice
   ch,...            % final H cor values
   cv,...            % final V cor values
   cq,...            % final Quad cor values
   cs...             % final Skew Quad cor values
   ]=CorrectionChain(...
   rerr,...          %1  initial lattice with errors
   r0,...            %2  model lattice
   indBPM,...        %3  bpm index
   indHCor,...       %4  h steerers index
   indVCor,...       %5  v steerers index
   indSkewCor,...    %6  skew quad index
   indQuadCor,...    %7  quadrupole correctors index
   Neig,...          %8  number of eigen vectors
   corrorder,...     %9  correction order 1: orbit, 2: tune,…
   ModelRM,...       %10 response matrices
   speclab,...       %11 label
   verbose)          %12 verbose (false)
```

Finally, a single function does the whole commissioning like procedure

(S. Liuzzo)

RDT AND DISPERSION CORRECTION

This function can be used on a fitted lattice or on the lattice with errors.



alpha: 0.80 (h disp) 0.10 *(tune) 0.80 (v disp)
RDT Disp. Tune Steering iter 1, n-eig: 150, 50,
RDT Disp. Tune Steering iter 2, n-eig: 250, 100,
before --> after
rq: 9.174 -> 5.711
rs: 15.024 -> 7.079
dX: 5.839 -> 1.810mm
dY: 1.466 -> 0.715mm
tX: 0.008 -> 0.007
tY: -0.000 -> -0.004

	min	mean	max	
qs:	-0.03	-0.00	0.03	1/m2
ss:	-0.01	-0.00	0.01	1/m2

dpp: 0

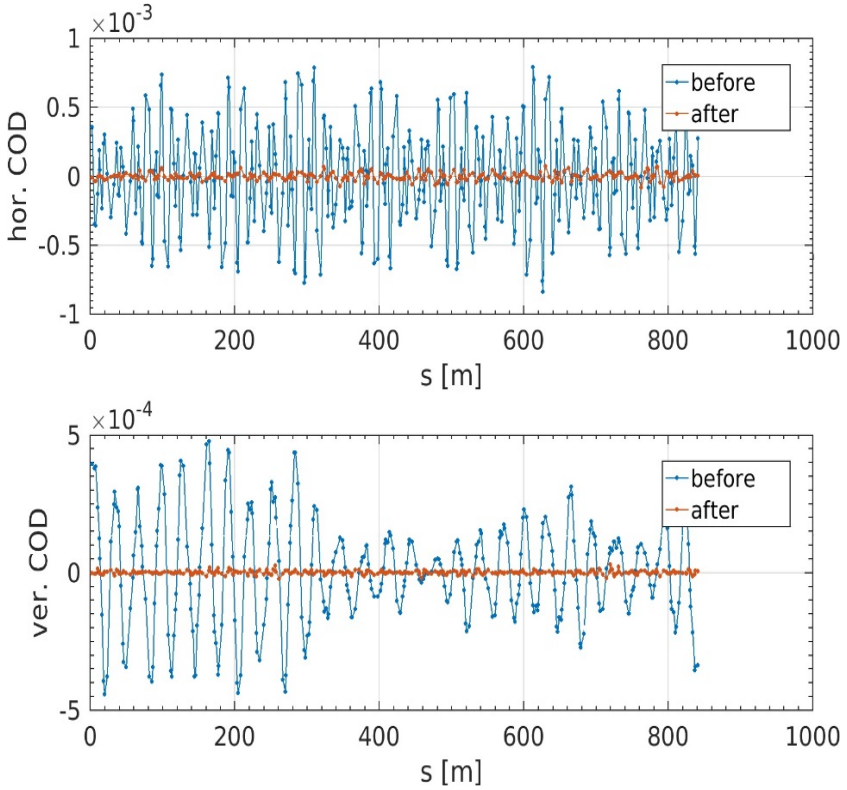
The normal and skew quadrupole components are computed from gradient errors and from alignment errors.

(S. Liuzzo)

ORBIT CORRECTION

```
[rcor,inCOD,hs,vs]=atcorrectorbit(…
    rcor,...
    indBPM,...
    indHCor,...
    indVCor,...
    inCOD,...
    [
    [50 50];...
    [100 100];...
    [150 150];...
    [200 200];...
    [220 220];...
    [240 240];...
    [260 260]...
    ],...
    [true true],...
    1.0,...
    ModelRM,...
    zeros(2,length(indBPM)),...
    [0.5e-3 0.5e-3],...
    true);
```

several correction iterations with different number of eigenvectors



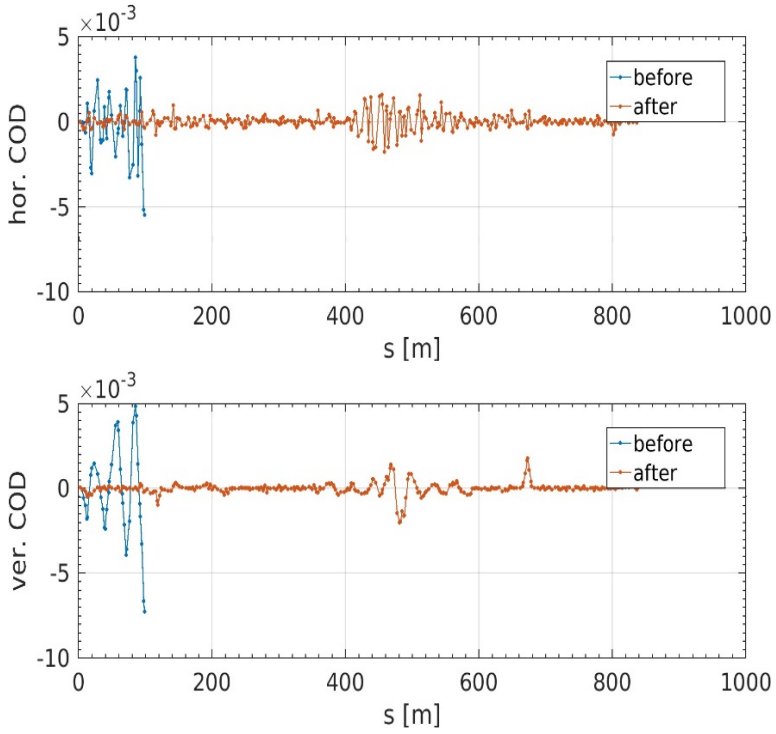

```
Orbit correction iter 1,n-eig: 50, 50,
Orbit correction iter 2,n-eig: 100, 100,
Orbit correction iter 3,n-eig: 150, 150,
Orbit correction iter 4,n-eig: 200, 200,
Orbit correction iter 5,n-eig: 220, 220,
Orbit correction iter 6,n-eig: 240, 240,
Orbit correction iter 7,n-eig: 260, 260,
      before    -->    after
oX: 128.917 -> 26.728um
oY: 65.781 -> 6.688um
    min    mean    max
hs:-0.01   0.00   0.01 mrad
vs:-0.02  -0.00   0.01 mrad
dpp: 3.2212e-25
```

Possible to correct to reference orbit
Usable for bumps
Defaulted RM computation from model
Correctors limits
Average of correctors to zero and DPP correction
Easy iteration varying the number of eigenvectors.

(S. Liuzzo)

OPEN TRAJECTORY CORRECTION

```
[rcor,inCOD]=atfirstturntrajectory(...
    rerr,...
    inCOD,...
    indBPM,...
    indHCor,...
    indVCor,...
    2.0e-3,...
    60,...
    [true true],...
    ModelRM,...
    zeros(2,length(indBPM)),...
    [5e-3 5e-3]);
```



correcting trajectory with model rm
Search closed orbit
Trajectory correction: **nbpms= 19 ncor: 16, 16,**
computing ORM for available trajectory
H PLANE
correcting available H trajectory
V PLANE
correcting available V trajectory
X: 1393.790 -> 806.879 um
Y: 1136.597 -> 420.950 um
Search closed orbit
Trajectory correction: nbpms= 34 ncor: 31, 31,
computing ORM for available trajectory
H PLANE
correcting available H trajectory
V PLANE
correcting available V trajectory
X: 1330.159 -> 575.785 um
Y: 779.406 -> 262.396 um
……. …. … …. … … … ….
Search closed orbit
Trajectory correction: nbpms= 319 ncor: 286, 286,
computing ORM for available trajectory
H PLANE
correcting available H trajectory
V PLANE
correcting available V trajectory
X: 184.243 -> 178.645 um
Y: 65.737 -> 52.012 um
Search closed orbit
no improvement from last step. increasing lim to: 0.0025
Trajectory correction: **nbpms= 320 ncor: 287, 287,**
computing ORM for available trajectory
H PLANE
correcting available H trajectory
V PLANE
correcting available V trajectory
X: 195.664 -> 190.196 um
Y: 95.250 -> 84.276 um
Search closed orbit
Found closed orbit

Possible to correct to reference trajectory
Defaulted RM computation from model
Correctors limits
Average of correctors to zero and DPP correction

(S. Liuzzo)

TO DO FOR AT AND ON THE HORIZON

Improve documentation

continue standardization of lattice format

merge physics tools and integrators into common repository

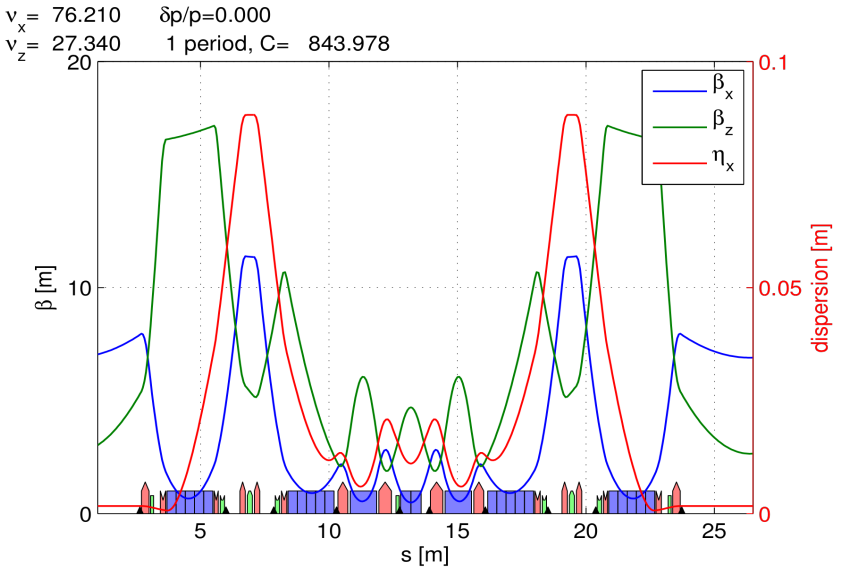
more tests of compatibility with MML

Python AT- pyAT is in development in collaboration with Will Rogers from Diamond

https://github.com/willrogers/at

CALCULATIONS FOR ESRF EBS

	current ESRF	EBS
circumference (m)	844.39	843.979
emittance (pm)	4000	135
beta_x,y	37/3	6.9/2.6
damping times (msec)	7/7/3.5	8.5/13/8.8
momentum compaction	1.78e-4	0.87e-4
tunes	36.44/13.39	75.6/27.6
Energy loss/turn (MeV)	4.88	2.60
Natural chrom	-130/-58	-100/-84



(atplot)

CALCULATING PARAMETERS IN AT FOR ESRF1, EBS

function atx() gives linear optics plus
Ohmi Envelope results

[linopt,p]=atx(esrf)

[linoptS28D,pS28D]=atx(S28D)

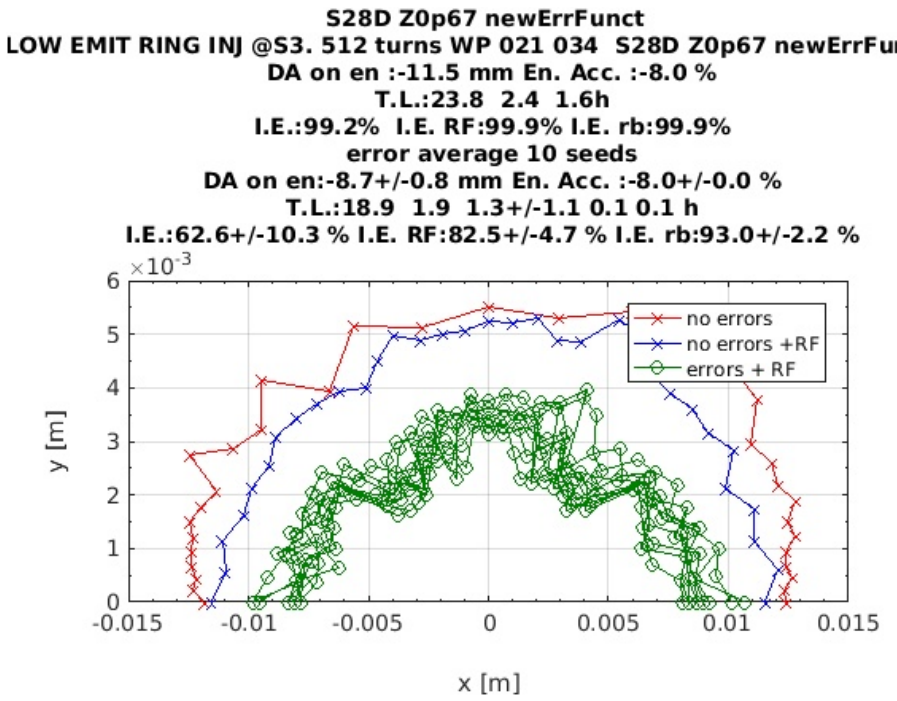
p =

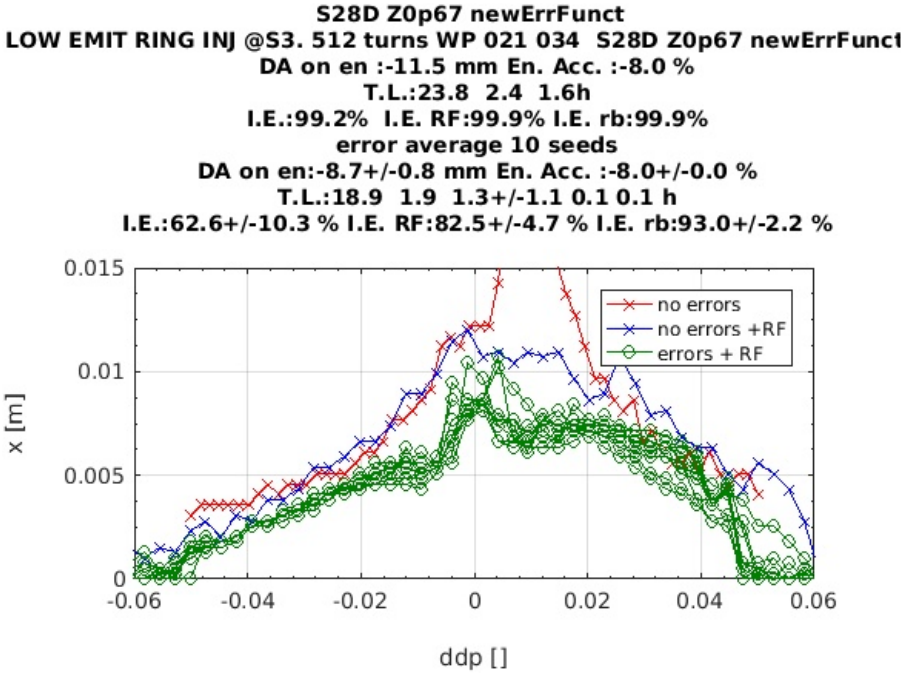
ll: 844.3904
alpha: 1.7758e-04
fractunes: [0.4396 0.3901]
fulltunes: [36.4396 13.3901]
nuh: 36.4396
nuv: 13.3901
chromaticity: [5.7983 6.1301]
dampingtime: [0.0070 0.0070 0.0035]
espread: 0.0011
blength: 0.0043
modemittance: [4.0131e-09 1.3081e-36 4.5432e-06]
energy: 6.0400e+09
fs: 2.1040e+03
eloss: 4.8806e+06
synchrophase: 0.5732
momcompact: 1.7758e-04

pS28D =

ll: 843.9773
alpha: 8.5140e-05
fractunes: [0.2100 0.3400]
fulltunes: [76.2100 27.3400]
nuh: 76.2100
nuv: 27.3400
chromaticity: [5.9996 3.9997]
dampingtime: [0.0088 0.0134 0.0091]
espread: 9.3359e-04
blength: 0.0029
modemittance: [1.3211e-10 -5.0066e-38 2.7210e-06]
energy: 6.0000e+09
fs: 1.3017e+03
eloss: 2.5224e+06
synchrophase: 0.3985
momcompact: 8.5140e-05

DYNAMIC APERTURE

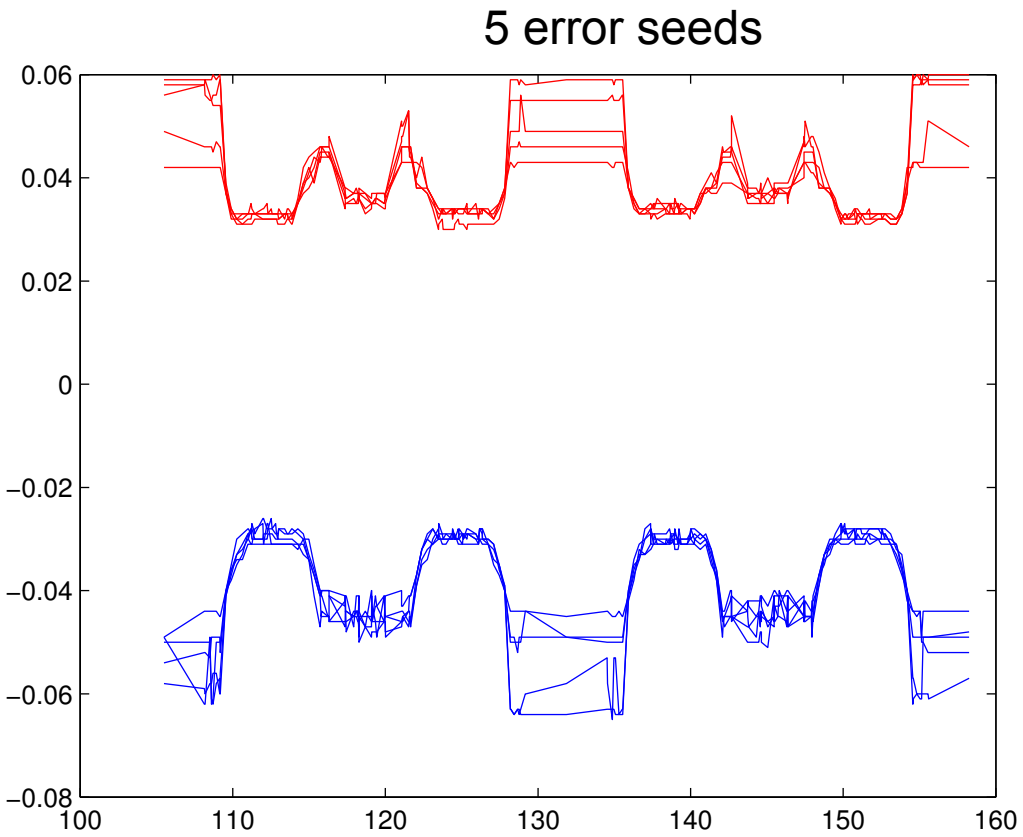




Zn = .67 ohms
for bunch lengthening

booster beam params for injection efficiency calcs:
epsx,y=(120,5) (60,5), (30,30)

MOMENTUM ACCEPTANCE AND TOUSCHEK LIFETIME



tau=16.3+/-2.8 hrs

computed on cluster (paralellized tracking) by
N. Carmignani

MODE PARAMETERS

multi-bunch

200mA, 868 bunches

16 bunch

90 mA

4 bunch

40 mA

Mode	Intensity	I_b (mA)	Lifetime
Uniform	200	.202	8.5 hrs
16 bunch	90	5.62	2.42 hrs
4 bunch	40	10	1.78 hrs

CONCLUSIONS

AT is still widely used in light source community

An attempt has been made to coordinate development, called *atcollab*

AT is used in two ways, as an online modeling tool, usually integrated within MML, and secondly as a design tool for low emittance light sources

We (at ESRF) have added tools to make AT powerful for design purposes, and others can benefit from our work.

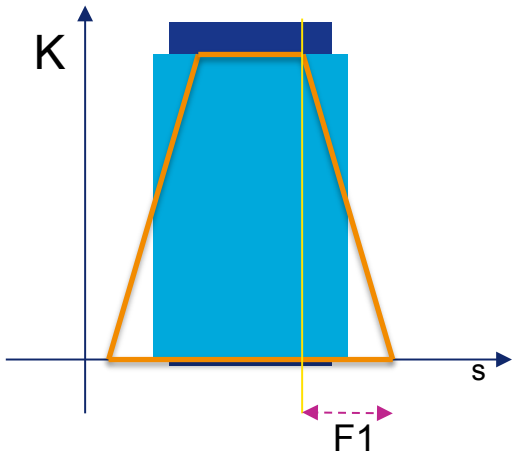
We have attempted to maintain backwards compatibility, so we hope that our version of AT can also be used widely with MML and a standard version can be maintained.

AT has been the main tool used in the modeling for the ESRF EBS.

Let's work together for a release of AT2.0!!

EXTRA SLIDES

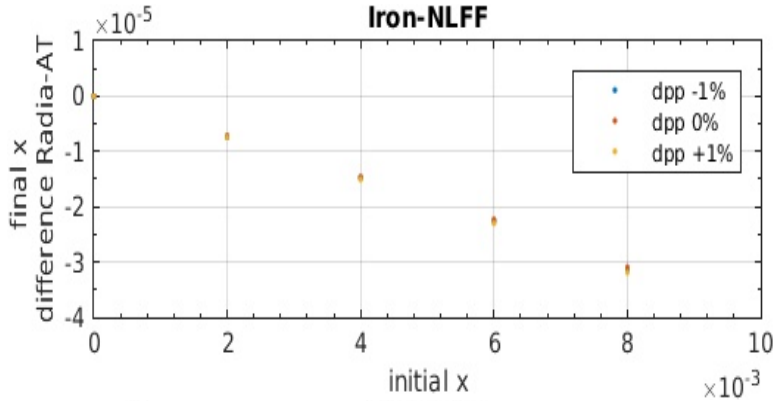
QUADRUPOLE MODEL: AT VS RADIA

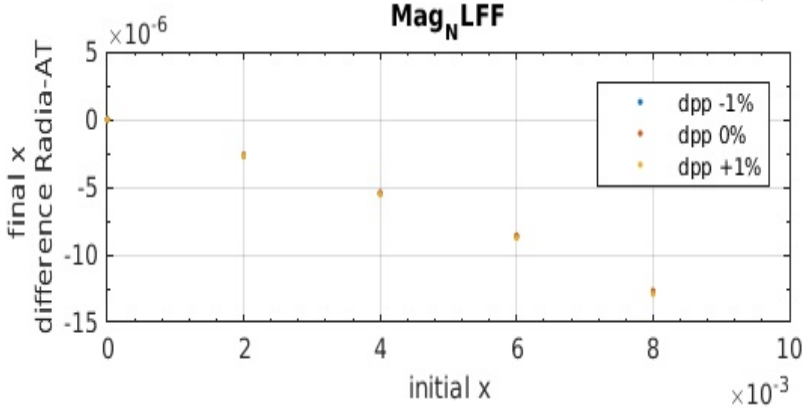


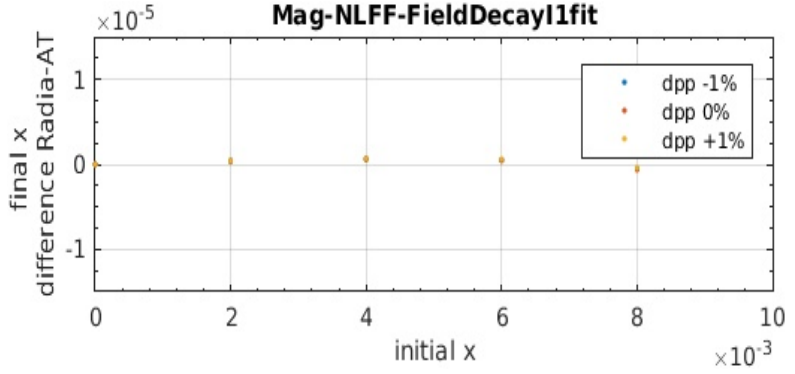
To fit at best the tracking in RADIA for QF8, with a trapezoid,
(field integral unchanged)

F1=0.079 m
(I1+ = I1- = 0.00013)

Also similar or better then a sliced model







RINGPARAM

Issue: if we want to define many different ring structures at the same time, we need a way to define global parameters for each one.

Idea: create an element in each ring to store global parameters for that ring.

ATRINGPARAM(FAMNAME,E0,NBPERIODS)

reference energy periodicity of lattice

Not many functions yet developed to take advantage of this feature.

(IdentityPass needed as PassMethod for RingPass)