

Integrating solenoid field maps in PLACET

Yngve Inntjore Levinsen
Barbara Dalena
Rogelio Tomás Garcia

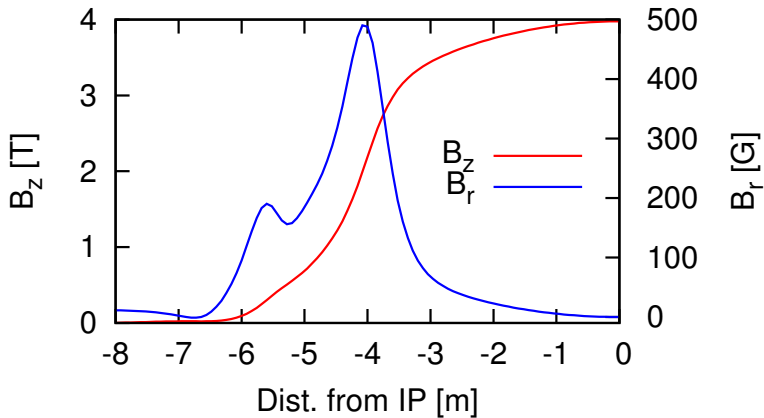
CERN

11. of May, 2012

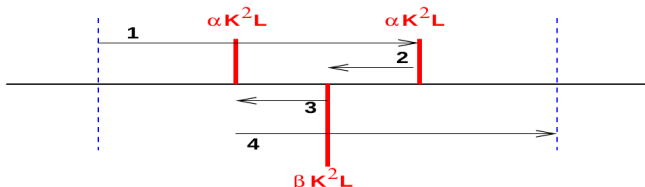
- Solenoid field maps in ASCII input format
- Integrate with lattice and track using symplectic integrator
- Integration of code into PLACET

r	z	B_r	B_z
[cm]	[cm]	[G]	[G]

- Current input format given above
- r redundant? $r = z \tan(\text{xangle})$
- Input format/units should be fixed



- New class DetectorSolenoid defined
 - Holds information about the field map
 - Holds coordinate transforms between reference systems
- New class IRTracking defined
 - Contains all other information about tracking procedure
 - 2nd and 4th order symplectic integrator available
- All elements in BDS need new function GetMagneticField() defined



Example:

```
...  
TestIntRegion -beam beam1 -emitt_file emitt.dat  
              -angle 0.001 -step 0.005 -synrad 1  
              -filename ildantinobuck.txt
```

- New tracking method TestIntRegion defined.
- $z=0$ in map assumed at end of beamline.
- If solenoid map does not cover the full length of the beamline, field assumed to be 0 elsewhere.
- **Not available** in existing tracking methods, TestFreeCorrection, TestSimpleCorrection...

- In principle, this is just a new step function
- Could instead be implemented directly into elements
- If solenoid map available, simply switch step function inside element

- In principle, this is just a new step function
- Could instead be implemented directly into elements
- If solenoid map available, simply switch step function inside element
- **Would make solenoid field available with existing tracking methods**

- In principle, this is just a new step function
- Could instead be implemented directly into elements
- If solenoid map available, simply switch step function inside element
- **Would make solenoid field available with existing tracking methods**
- Existing implementation ~700 lines, nearly finished debugging
 - Less rewrite of original code
 - Took a few weeks to implement&debug

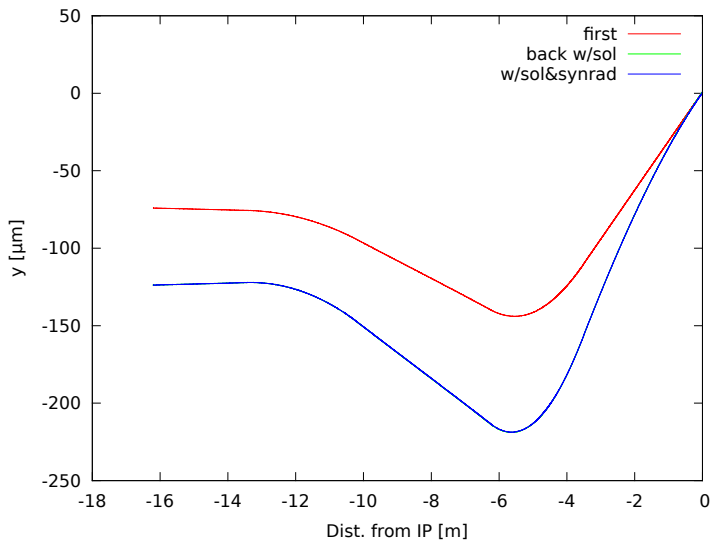
- In principle, this is just a new step function
- Could instead be implemented directly into elements
- If solenoid map available, simply switch step function inside element
- **Would make solenoid field available with existing tracking methods**
- Existing implementation ~700 lines, nearly finished debugging
 - Less rewrite of original code
 - Took a few weeks to implement&debug
- Implementation requires careful debugging; coordinate system transformations & sign conventions
- Expect **order of 1-2 weeks** to write&debug this implementation

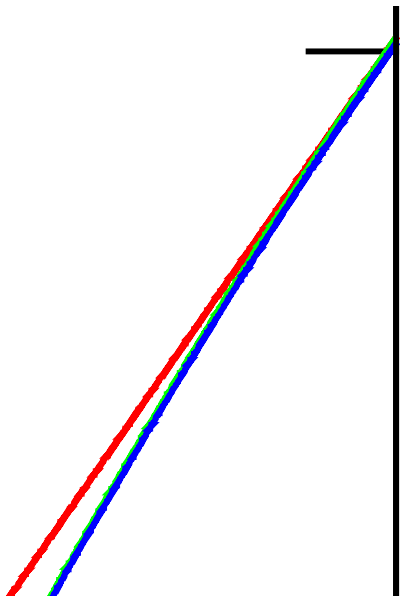
- Aim: Study the effect of synchrotron radiation from solenoid field
- Need to subtract effects from non-optimized BDS, synchrotron radiation of other elements

- Aim: Study the effect of synchrotron radiation from solenoid field
- Need to subtract effects from non-optimized BDS, synchrotron radiation of other elements
- Procedure:
 - Track electrons through **without** solenoid field or synchrotron radiation

- Aim: Study the effect of synchrotron radiation from solenoid field
- Need to subtract effects from non-optimized BDS, synchrotron radiation of other elements
- Procedure:
 - Track electrons through **without** solenoid field or synchrotron radiation
 - **Flip** distribution to positrons
 - Track positrons **backwards with** solenoid field but **without** synchrotron radiation

- Aim: Study the effect of synchrotron radiation from solenoid field
- Need to subtract effects from non-optimized BDS, synchrotron radiation of other elements
- Procedure:
 - Track electrons through **without** solenoid field or synchrotron radiation
 - **Flip** distribution to positrons
 - Track positrons **backwards with** solenoid field but **without** synchrotron radiation
 - **Flip** distribution to electrons
 - Track electrons **forwards** through the beamline **with** solenoid field & synchrotron radiation





- Solenoid implementation soon ready, mostly debugged.
- Alternative implementation has been suggested
 - Would provide some added functionality
 - Would probably be easier to maintain
 - Time estimate ~week, expect some time for debugging
- Should decide on input format, sign convention, & units