**From Joe 10/27/20 discussion**

List of topics to think about:

1. Assigning spin component information to a particle (or macroparticle)
2. Implementing the Bargman-Michel-Telegdi time-dependent spin-equation of motion
3. Generally rate of spin motion (ds/dt) is similar to physical motion (dx/dt), but maybe consider how time steps are computed, e.g. when an adaptive time step is implemented
4. Implement methods to define initial spin distribution a) internally (e.g. set sx,sy,sz of particles very useful for testing) or b) externally (e.g. to accept a distribution originating from another code like Geant4 output)
5. Evaluate numerical test cases of 'pure fields' (e.g. B\_perp, B\_axial, E\_perp, E\_axial) or single elements (e.g. dipole, solenoid, e-field deflector) against analytic solutions
6. Evaluate numerical test case of field map (e.g. we have ExB wien filter field maps, solenoids, cavity)
7. Saving the spin component information, similar to other arrays, in output file
8. Ability to display spin component (or polarization = average of sx, sy or sz) in GUI would be useful

**From Bas 11/9/20 discussion**

Let’s already try to agree on a few things (very important technical details):

1. The spin coordiantes will be called ’sx’, ’sy’ and ’sz’. This will be a real pain if we need to change this later. Choose wisely.
2. They can be read from file using (to be modified) setfile if the corresponding columns are available
3. You let me know what other methods you would like to see for initialization. This is NOT trivial.
4. Output will be with the same ’sx’, ’sy’ and ’sz’ columns if you agree with the name
5. I create the differential equations for spin, but I am not sure if they will slowly diverge and get a length != 1. Maybe we have to (re)-normalise them after each timestep. No big deal, also not at all a performance issue, but we should not forget about this.
6. Display is already done since I programmed a GDF to ParaView converted and this allows you to for example plot particle coordinates as spheres, velocity vectors as 3D arrows and 3D spin vectors as whatever 3D shape/color you like.

