First, apologies to Johann - for transforming his beautiful code and coding into whatever mash this is.

I use ROOT Ntuples and since for my case they are small, I usually end up working with them interactively. I also place the "pcor.txt" file in the same directory as my root code.

Root wants to compile an ACLiC dictionary before it will let me use maps of vectors of vectors of doubles. Even after doing this, It would not work for me. I spent a day trying to get that to work and in the end just resorted to the following method.

```
Before I loop over my events, I call from my root code:
#include <g12_pcor_alt.h>
.
.
.
double pcor_array[5][6][2];
  read_pcor( pcor_array );

This reads in all the corrections in a pseudo-pid array where the format is :
pcor_array[pseudo_pid][sector][poln]
and
pseudo_pid is
//// proton = 0
//// KPlus = 1
//// KMinus = 2
//// PiPlus = 3
//// PiMinus = 4
```

Again, I used a temporary pseudo-PID scheme since I cannot get map of vectors of vectors of doubles in the gl2 namespace of clas6 to compile right for me in root.

So I switched to arrays and redefine the Geant_PID as PID_M.

Later in the code when I loop over my events I read in the eloss corrected particle fourvectors into a TLorentzVector "P4vector[..]".

I call the "Calc_new_fourVec(P4vector[..], PID_M)" function to get the new and improved momentum corrected fourVectors for proton, K+ and K- as follows:

```
pc_P4vector[0] = Calc_new_fourVec(P4vector[0], 0);
pc_P4vector[1] = Calc_new_fourVec(P4vector[1], 1);
pc P4vector[2] = Calc_new_fourVec(P4vector[2], 2);
```

The angle phi obtained from using the root function TVector3::Phi() gives me the phi in radians and in the range (-pi,pi).

Momentum Correction code requires the phi angle to be between (0,360) degrees.

Since I am still unsure about how the momentum corrections sector notation matches up to the phi that I get from my four-vectors, I considered both cases. When we have it figured out, We can pick the right case.

For now the two cases are:

Case 1:

If sector-1 center is at "-pi" in my four-vector notation, I use :

new_phi = phi_degrees + 180. , to translate it into the correct momentum
correction notation.

Case 2:

If sector-1 center is at "0" in my four-vector notations, I use :

```
if ( phi_degrees < -30. ) {
    phi_degrees = 360 + phi_degrees;
    //cout << "new phi :: " << phi_degrees << endl;
}</pre>
```

to translate it into the correct momentum correction notation.

Note: My four-vectors are obtained initially from BOS data using the CLASEvent functions - CLASParticle and fourVec.

I have double-checked that these kludge works for all the cases I could come up with.

This is obviously not ideal. If anyone can guide me as to how I can use that vector map and get it to work with root, that would be good.

This can at least serve as a template for the people using root, in case they cannot get this to work either. Also, We get the corrected four-momentum with the same orientation as the initial four-momentum, and the invariant mass for each particle type is enforced.