Status report for the Simulation Validation

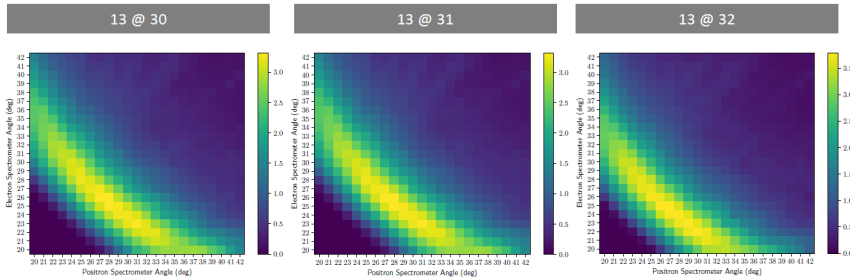
David Markus

Darklight@Ariel at TRIUMF

June 18, 2024

Previous Issue Identified

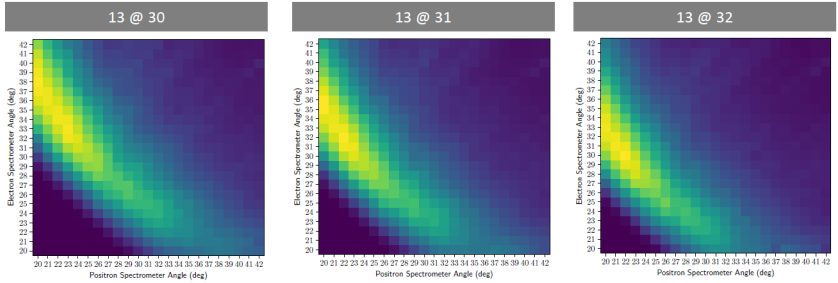
Previously Julia Azzi and Story Frantzen identified an asymmetry in simulation results:



Electron Decay Particle Results

Previous Issue Identified

Previously Julia Azzi and Story Frantzen identified an asymmetry in simulation results:



Positron Decay Particle Results

Location in Code

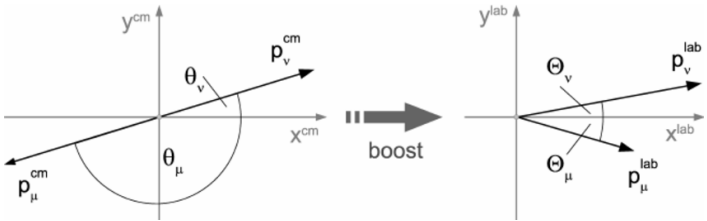
The only difference between the above two simulations is which particle is used to generate the background:

```
double thetaD = e1out.Lorentz(-q_out).rotate(q_out).theta();
double phiD   = e1out.Lorentz(-q_out).rotate(q_out).phi();
double weight = QEDBackground(e_in,e_out,q_out,m,thetaD,phiD, 0, tdata->trident, tdata->asymm)*Solidangle;
```

```
double thetaD = e2out.Lorentz(-q_out).rotate(q_out).theta();
double phiD   = e2out.Lorentz(-q_out).rotate(q_out).phi();
double weight = QEDBackground(e_in,e_out,q_out,m,thetaD,phiD, 0, tdata->trident, tdata->asymm)*Solidangle;
```

Geometry

Looking at this superficially makes this appear correct; however, geometry alone tells us it is wrong:



Two body decay kinematics

Solution

The correct solution is given by shifting the angles of the positron:

```
//Recover the decay angles from the decay particles as a check
//Electron was generated from random angles in rest frame, is now in lab frame, has to be deboosted and rotated back.
double thetaD = e1out.Lorentz(-q_out).rotate(q_out).theta();
double phiD   = e1out.Lorentz(-q_out).rotate(q_out).phi();

//Positron was generated from electron in lab frame,
double thetaD_second = M_PI - e2out.Lorentz(-q_out).rotate(q_out).theta();
double phiD_second   = M_PI + e2out.Lorentz(-q_out).rotate(q_out).phi();

//Correction check printout. When considering result remember 2pi periodicity of phi.
printf("thetaD: %f thetaD calc e-: %f thetaD calc e+: %f \n",thetaD,thetadecay,thetaD_second);
printf("phiD: %f phiD calc e-: %f phiD calc e+: %f\n",phiD,phidecay,phiD_second);

// 0 for muon - not currently being changed
double weight = QEDBackground(e_in,e_out,q_out,m,thetaD,phiD, 0, tdata->trident, tdata->asymm)*Solidangle;
```

```
thetaD: 1.711989 thetaD calc e-: 1.711989 thetaD calc e+: 1.711989
phiD: -2.660814 phiD calc e-: 3.622371 phiD calc e+: 3.622371
thetaD: 1.653865 thetaD calc e-: 1.653865 thetaD calc e+: 1.653865
phiD: 2.047389 phiD calc e-: 2.047389 phiD calc e+: 2.047389
thetaD: 1.399478 thetaD calc e-: 1.399478 thetaD calc e+: 1.399478
phiD: 0.066413 phiD calc e-: 0.066413 phiD calc e+: 0.066413
```

Additional bug found and corrected

Since the original commit, there has been in error in the particle generation:

```
//    e1out = Polar(energy(m_electron,peq), peq,
//            thetadecay, phidecay).rotateTo(q_out).Lorentz(q_out);
//
// more efficient without rotation!!!

// e1out is positron e2out is

// boosts back to lab frame (thetadecay and phidecay) are angles of ouotgoing electron (not relative to positron)
e1out = Polar(energy(lepton,peq), peq,
              thetadecay, phidecay).Lorentz(q_out);
```

It is, of course, indeed more efficient. It is unfortunately however also wrong.