Michel Fodje's epr-simple simulation translated from Python to Mathematica by John Reed 13 Nov 2013
Modified for QM Local Complete States 5 Aug 2019
No Zeroes in A and B outputs

Set Run Time Parameters, Initialize Arrays and Tables

In[1815]:=
```
trialcs = 10 000 000;
trialDeg = 360;
```

In[1817]:=
```
CS1 = Table[{0, 0, 0, 0, 0}, trialcs];
CS2 = Table[{0, 0, 0, 0, 0}, trialcs];
CA = Table[{0, 0, 0}, trialcs];
CB = Table[{0, 0}, trialcs];
a1 = ConstantArray[0, trialcs];
b1 = ConstantArray[0, trialcs];
aliceDeg = ConstantArray[0, trialcs];
bobDeg = ConstantArray[0, trialcs];
A1 = ConstantArray[0, trialcs];
B1 = ConstantArray[0, trialcs];
a2 = ConstantArray[0, trialcs];
b2 = ConstantArray[0, trialcs];
ss1 = ConstantArray[0, trialcs];
aliceD1 = ConstantArray[0, trialcs];
bobD1 = ConstantArray[0, trialcs];
```

In[1832]:=
```
nPP = ConstantArray[0, trialDeg];
nNN = ConstantArray[0, trialDeg];
nPN = ConstantArray[0, trialDeg];
nNP = ConstantArray[0, trialDeg];
nAP = ConstantArray[0, trialDeg];
nBP = ConstantArray[0, trialDeg];
nAN = ConstantArray[0, trialDeg];
nBN = ConstantArray[0, trialDeg];
```

Generate Particle Data

In[1840]:=
```
Do[
 s = Normalize@RandomVariate[NormalDistribution[], 3];  (*3D uniform random unit vectors*)
 a = Normalize@RandomVariate[NormalDistribution[], 3];
 b = Normalize@RandomVariate[NormalDistribution[], 3];
 x1 = Part[a, 1]; y1 = Part[a, 2];
 aliceDeg[[j]] = ArcTan[x1, y1];
 x2 = Part[b, 1]; y2 = Part[b, 2];
 bobDeg[[j]] = ArcTan[y2, x2];
 λ = RandomChoice[{-1, 1}];
 If[a.s > 0, s1 = a, s1 = -a];  (*Polarizer Functions*)
 If[b.s > 0, s2 = b, s2 = -b];
 (*Measurement Functions*)
 A = λ (a.(-s1));
 B = λ (b.s2);
 CA[[j]] = {a, A, s};
 CB[[j]] = {b, B},
 {j, trialcs}]
```

Complete States Selection and Statistical Analysis of Particle Data

In[1841]:=
```
a1 = CA[[All, 1]];
b1 = CB[[All, 1]];
ss1 = CA[[All, 3]];
A1 = CA[[All, 2]];
B1 = CB[[All, 2]];
Do[
 t = RandomReal[{0, π}];
 z = (2/Sqrt[1 + 3 t/π] - 1);
 If[Abs[a1[[j]].ss1[[j]]] < z || Abs[b1[[j]].ss1[[j]]] < z, CS = 0, CS = 1];
 CS1[[j]] = {A1[[j]], B1[[j]], a1[[j]], b1[[j]], CS},
 {j, trialcs}]
CS2 = Select[CS1, Last[#] == 1 &];
trials = Length[CS2];
a2 = CS2[[All, 3]];  b2 = CS2[[All, 4]];
aliceD1 = CS2[[All, 1]];  bobD1 = CS2[[All, 2]];
```

In[1847]:=
```
Do[
 If[(aliceDeg[[j]] * bobDeg[[j]]) > 0, theta = ArcCos[a2[[j]].b2[[j]]] * 180/π,
  theta = -ArcCos[a2[[j]].b2[[j]]] * 180/π + 360];
 ϴ = Round[theta];
 aliceD = aliceD1[[j]]; bobD = bobD1[[j]];
 If[aliceD == 1, nAP[[ϴ]]++];
 If[bobD == 1, nBP[[ϴ]]++];
 If[aliceD == -1, nAN[[ϴ]]++];
 If[bobD == -1, nBN[[ϴ]]++];
 If[aliceD == 1 && bobD == 1, nPP[[ϴ]]++];
 If[aliceD == 1 && bobD == -1, nPN[[ϴ]]++];
 If[aliceD == -1 && bobD == 1, nNP[[ϴ]]++];
 If[aliceD == -1 && bobD == -1, nNN[[ϴ]]++],
 {j, trials}]
```

Calculate mean values and plot

In[1848]:=
```
pPP = 0;  pPN = 0;  pNP = 0; pNN = 0;
```

In[1849]:=
```
mean = ConstantArray[0, trialDeg];
```

In[1850]:=
```
Do[
 sum = nPP[[i]] + nPN[[i]] + nNP[[i]] + nNN[[i]];
If[sum == 0, Goto[jump],
  {pPP = nPP[[i]] / sum;
   pNP = nNP[[i]] / sum;
   pPN = nPN[[i]] / sum;
   pNN = nNN[[i]] / sum;
mean[[i]] = pPP + pNN - pPN - pNP}];
 Label[jump],
 {i, trialDeg}]
```

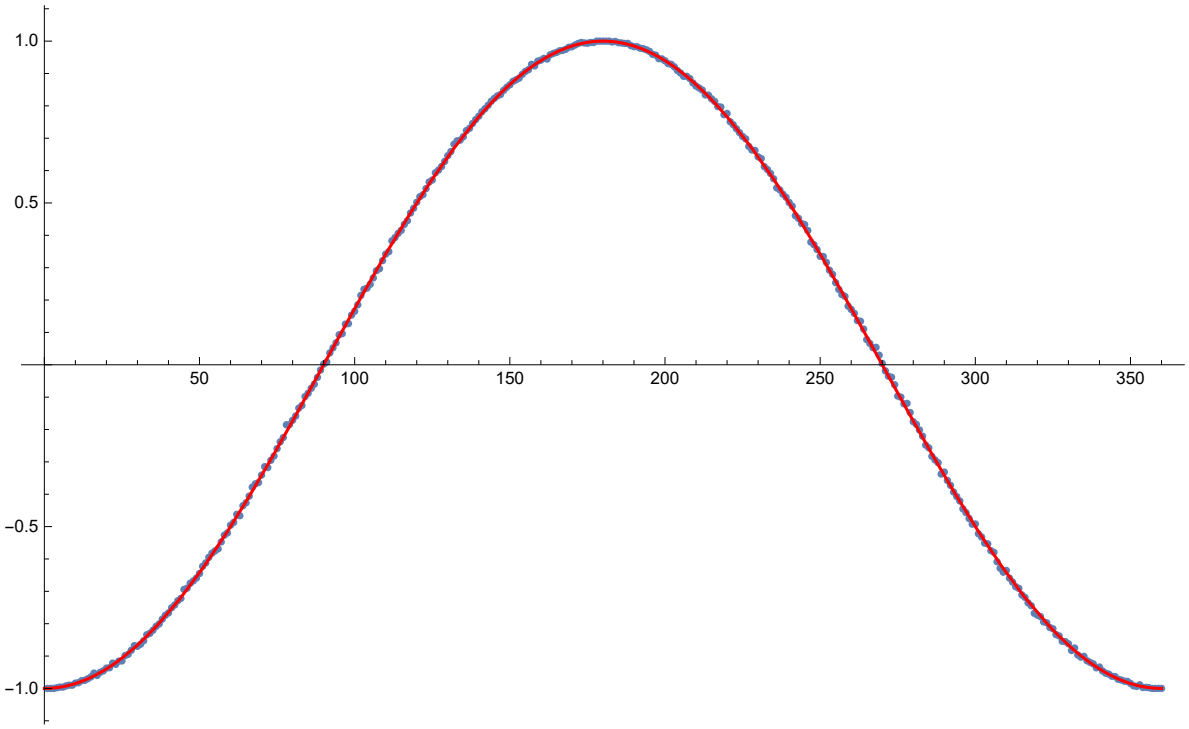
In[1851]:=
```
simulation = ListPlot[mean, PlotMarkers → {Automatic, Tiny}];
```

In[1852]:=
```
negcos = Plot[-Cos[x Degree], {x, 0, 360}, PlotStyle → {Red}];
```

Compare mean values with -Cosine Curve and compute averages

In[1853]:=
```
Show[simulation, negcos]
AveA = N[Sum[A1[[i]], {i, trials}] / trials];
AveB = N[Sum[B1[[i]], {i, trials}] / trials];
Print["AveA = ", AveA]
Print["AveB = ", AveB]
PAP = N[Sum[nAP[[i]], {i, trialDeg}]];
PBP = N[Sum[nBP[[i]], {i, trialDeg}]];
PAN = N[Sum[nAN[[i]], {i, trialDeg}]];
PBN = N[Sum[nBN[[i]], {i, trialDeg}]];
PA1 = PAP / (PAP + PAN);
PB1 = PBP / (PBP + PBN);
Print["P(A+)= ", PA1]
Print["P(B+)= ", PB1]
totAB = Sum[nPP[[i]] + nNN[[i]] + nPN[[i]] + nNP[[i]], {i, trialDeg}];
PP = N[Sum[nPP[[i]], {i, trialDeg}] / totAB]
NN = N[Sum[nNN[[i]], {i, trialDeg}] / totAB]
PN = N[Sum[nPN[[i]], {i, trialDeg}] / totAB]
NP = N[Sum[nNP[[i]], {i, trialDeg}] / totAB]
CHSH = Abs[N[mean[[22]]] + N[mean[[67]]] - N[mean[[135]]] + N[mean[[315]]]]
```

Out[1853]=




```
AveA = -0.000186972
AveB = 0.000599358
P(A+)= 0.499994
P(B+)= 0.500371
```

Out[1867]= 0.250242

Out[1868]= 0.249876

Out[1869]= 0.249753

Out[1870]= 0.25013

Out[1871]= 2.71548