Simulation Based on Michel Fodje's epr-simple simulation translated from Python to Mathematica by John Reed 13 Nov 2013
Modified by Fred Diether for Completely Local-Realistic Sep 2021
Some parts by Bill Nelson.

Set Run Time Parameters, Initialize Arrays and Tables

In[2106]:=
```
m = 5000000;
trialDeg = 721;
s1 = ConstantArray[0, m];
s2 = ConstantArray[0, m];
λ1 = ConstantArray[0, m];
λ2 = ConstantArray[0, m];
outA1 = Table[{0, 0}, m];
outA2 = Table[{0, 0}, m];
outB1 = Table[{0, 0}, m];
outB2 = Table[{0, 0}, m];
aa = ConstantArray[0, m];
bb = ConstantArray[0, m];
a1 = ConstantArray[0, m];
b1 = ConstantArray[0, m];
A = ConstantArray[0, m];
B = ConstantArray[0, m];
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];
ϕ = 3; β = 0.32; ξ = 0;  (*Adustable parameters for fine tuning*)
```

Generating Particle Data with Three Independent Do-Loops

In[2131]:=
```
Do[θ = RandomReal[{0, 360}];  (*Singlet vector angle*)  (*Hidden Variable*)
 λ1[[i]] = β (Cos[θ / ϕ]^2);
 λ2[[i]] = 0.938 (Cos[θ / ϕ]^2);
 s1[[i]] = θ;
 s2[[i]] = θ + 180, {i, m}]     (*Conservation of angular momentum*)
```

In[2132]:=
```
Do[a = RandomInteger[{0, 360}];  (*Detector vector angle 1 degree increments*)
 aa[[i]] = a;
 If[Abs[Cos[(a - s1[[i]]) Degree]] > λ1[[i]], Aa1 = Sign[Cos[(a - s1[[i]]) Degree]], Aa1 = 0];
 outA1[[i]] = {a, Aa1}, {i, m}]
Do[If[Abs[Cos[(aa[[i]] - s1[[i]]) Degree]] < λ2[[i]],
   Aa2 = Sign[Sin[(aa[[i]] - s1[[i]] + ξ) Degree]], Aa2 = 0];
 outA2[[i]] = {aa[[i]], Aa2}, {i, m}]
outA = Catenate[{outA1, outA2}];
a1 = outA[[All, 1]];
A = outA[[All, 2]];
```

In[2137]:=
```
Do[b = RandomInteger[{0, 360}];  (*Detector vector angle 1 degree increments*)
 bb[[i]] = b;
 If[Abs[Cos[(b - s2[[i]]) Degree]] > λ1[[i]], Bb1 = Sign[Cos[(b - s2[[i]]) Degree]], Bb1 = 0];
 outB1[[i]] = {b, Bb1}, {i, m}]
Do[If[Abs[Cos[(bb[[i]] - s2[[i]]) Degree]] < λ2[[i]],
  Bb2 = Sign[Sin[(bb[[i]] - s2[[i]] + ξ) Degree]], Bb2 = 0];
 outB2[[i]] = {bb[[i]], Bb2}, {i, m}]
outB = Catenate[{outB1, outB2}];
b1 = outB[[All, 1]];
B = outB[[All, 2]];
```

Statistical Analysis of the Particle Data Received from Alice and Bob

In[2142]:=
```
m2 = 2 m;
Do[θ1 = a1[[j]] - b1[[j]] + 361;
 (*All angles are shifted by 361 degrees since θ1 is an index*)
 aliceD = A[[j]]; bobD = B[[j]];
 If[aliceD == 1, nAP[[θ1]]++];
 If[bobD == 1, nBP[[θ1]]++];
 If[aliceD == -1, nAN[[θ1]]++];
 If[bobD == -1, nBN[[θ1]]++];
 If[aliceD == 1 && bobD == 1, nPP[[θ1]]++];
 If[aliceD == 1 && bobD == -1, nPN[[θ1]]++];
 If[aliceD == -1 && bobD == 1, nNP[[θ1]]++];
 If[aliceD == -1 && bobD == -1, nNN[[θ1]]++], {j, m2}]
```

Calculating Mean Values of A, B, and AB, and Plotting the Results

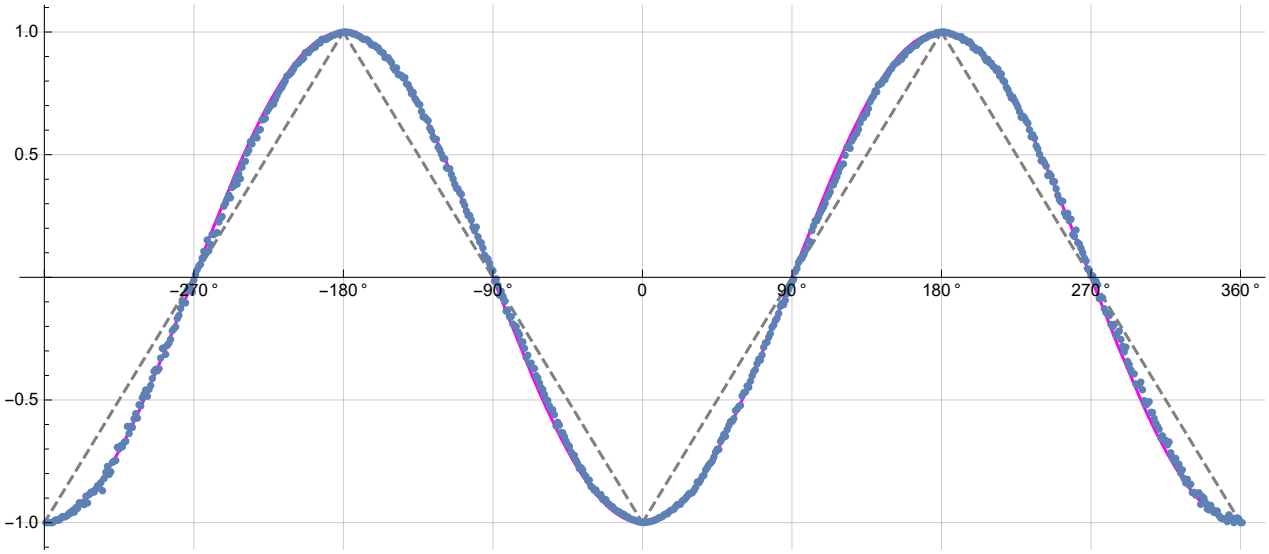
In[2144]:=
```
pPP = 0;  pPN = 0;  pNP = 0; pNN = 0;
mean = ConstantArray[0, trialDeg];
Do[sum = nPP[[i]] + nPN[[i]] + nNP[[i]] + nNN[[i]];
If[sum == 0, Goto[jump],
  {pPP = nPP[[i]] / sum;
   pNP = nNP[[i]] / sum;
   pPN = nPN[[i]] / sum;
   pNN = nNN[[i]] / sum;
mean[[i]] = pPP + pNN - pPN - pNP}];
 Label[jump], {i, trialDeg}]
```

In[2147]:=
```
simulation = ListPlot[mean, PlotMarkers → {Automatic, Tiny}];
negcos = Plot[-Cos[x Degree], {x, 0, 720}, PlotStyle → {Magenta}, AspectRatio → 7 / 16,
   Ticks → {{{0, -360 °}, {90, -270 °}, {180, -180 °}, {270, -90 °}, {360, 0 °}, {450, 90 °},
      {540, 180 °}, {630, 270 °}, {720, 360 °}}, Automatic}, GridLines → Automatic];
p1 = Plot[-1 + 2 x Degree / π, {x, 0, 180}, PlotStyle → {Gray, Dashed}];
p2 = Plot[3 - 2 x Degree / π, {x, 180, 360}, PlotStyle → {Gray, Dashed}];
p3 = Plot[-5 + 2 x Degree / π, {x, 360, 540}, PlotStyle → {Gray, Dashed}];
p4 = Plot[7 - 2 x Degree / π, {x, 540, 720}, PlotStyle → {Gray, Dashed}];
```

Comparing Mean Values with -Cosine Function and Computing Averages

In[2153]:=
```
Show[negcos, p1, p2, p3, p4, simulation]
```

Out[2153]=



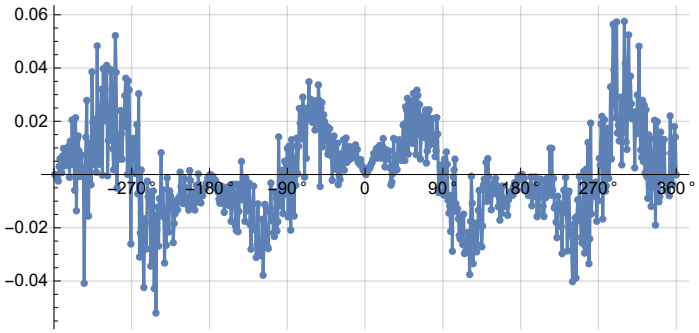
In[2154]:=
```
AveA = N[Sum[A[[i]], {i, m}] / m];
AveB = N[Sum[B[[i]], {i, m}] / m];
Print["AveA = ", AveA]
Print["AveB = ", AveB]
PAP = N[Sum[nAP[[i]], {i, trialDeg}]];
PBP = N[Sum[nBP[[i]], {i, trialDeg}]];
PAN = N[Sum[nAN[[i]], {i, trialDeg}]];
PBN = N[Sum[nBN[[i]], {i, trialDeg}]];
PA1 = PAP / (PAP + 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}];
Print["Total Events = ", totAB]
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];
Print["Ave ++ = ", PP]
Print["Ave -- = ", NN]
Print["Ave +- = ", PN]
Print["Ave -+ = ", NP]
CHSH = Abs[N[mean[[315]]] - N[mean[[225]]] + N[mean[[405]]] + N[mean[[45]]]];
Print["Approx. CHSH = ", CHSH]
```

```
AveA = 0.0003344

AveB = -0.0003312

P(A+) = 0.500103

P(B+) = 0.499816

Total Events = 4991174

Ave ++ = 0.250137

Ave -- = 0.249953

Ave +- = 0.250086

Ave -+ = 0.249824

Approx. CHSH = 2.78929
```

Deviation from negative cosine curve

In[2178]:=
```
dev1 = ConstantArray[2, 720];
dev2 = ConstantArray[2, 720];
dev3 = ConstantArray[2, 720];
Do[dev1 = mean[[i]]; dev2[[i]] = {dev1, i}, {i, 720}]
devang = dev2[[All, 2]] - 361;
Do[dev3[[i]] = mean[[i]] + Cos[devang[[i]] Degree], {i, 720}]
ListPlot[N[dev3], PlotMarkers → {Automatic, Tiny}, Joined → True, AspectRatio → 1 / 2,
 Ticks → {{{0, -360 °}, {90, -270 °}, {180, -180 °}, {270, -90 °}, {360, 0 °}, {450, 90 °},
    {540, 180 °}, {630, 270 °}, {720, 360 °}}, Automatic}, GridLines → Automatic]
```

Out[2184]=



In[2185]:=
```
Mean[N[dev3]]
Mean[N[Abs[dev3]]]
```

Out[2185]= 0.000579506

Out[2186]= 0.012917