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 Jan 2021

Set Run Time Parameters, Initialize Arrays and Tables

```
In[ ]:= trials = 10 000 000;
trialDeg = 720;
s = ConstantArray[0, trials];
λ = ConstantArray[0, trials];
CA = Table[{0, 0}, trials];
CB = Table[{0, 0}, trials];
a1 = ConstantArray[0, trials];
b1 = ConstantArray[0, trials];
A1 = ConstantArray[0, trials];
B1 = ConstantArray[0, trials];
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 with 3 Do Loops

```
In[ ]:= Do[e = RandomReal[{0, 360}];  (*Singlet vector angle*)
 s[[j]] = e;
 λ[[j]] = (Cos[e/2]^2)/2,  (*Hidden Variable*){j, trials}]

Do[a = RandomInteger[{0, 360}];  (*Detector vector angle 1 degree increments*)
 ar = RandomInteger[{1, 360}];
 If[0 < Cos[(a - s[[j]]) Degree] < λ[[j]] || 0 > Cos[(a - s[[j]]) Degree] > -λ[[j]],
  A = -Sign[((Cos[(ar - s[[j]]) Degree]))], A = -Sign[((Cos[(a - s[[j]]) Degree]))]];
 If[Abs[Cos[(a - s[[j]]) Degree]] < λ[[j]], aa = ar, aa = a];
 CA[[j]] = {aa, A}, {j, trials}]

Do[b = RandomInteger[{0, 360}];  (*Detector vector angle 1 degree increments*)
 br = RandomInteger[{1, 360}];
 If[0 < Cos[(b - s[[j]]) Degree] < λ[[j]] || 0 > Cos[(b - s[[j]]) Degree] > -λ[[j]],
  B = Sign[((Cos[(br - s[[j]]) Degree]))], B = Sign[((Cos[(b - s[[j]]) Degree]))]];
 If[Abs[Cos[(b - s[[j]]) Degree]] < λ[[j]], bb = br, bb = b];
 CB[[j]] = {bb, B}, {j, trials}]
(*Export["CA300K.csv",CA];
Export["CB300K.csv",CB];*)
```

Statistical Analysis of Particle Data

```
In[•]:= a1 = CA[[All, 1]]; b1 = CB[[All, 1]]; A1 = CA[[All, 2]]; B1 = CB[[All, 2]];
Do[θ = a1[[j]] - b1[[j]] + 360;  (*All angles are shifted by 2π since θ is an index*)
 aliceD = A1[[j]]; bobD = B1[[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

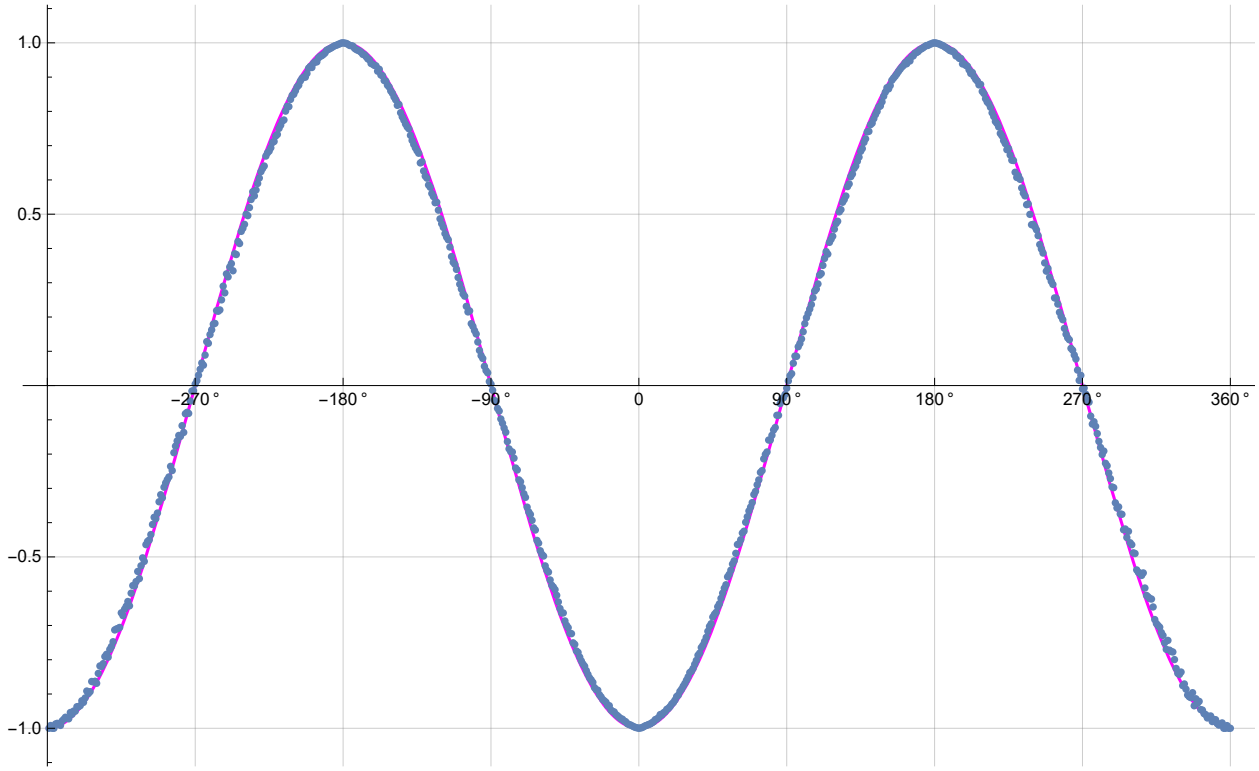
```
In[•]:= pPP = 0; pPN = 0; pNP = 0; pNN = 0;
mean = ConstantArray[0, trialDeg];
Do[sum = nPP[[i]] + nPN[[i]] + nNP[[i]] + nNN[[i]];
If[sum == 0, Goto[jump],
  {pPP = nPP[[i]] / sum;
   pNP = nNP[[i]] / sum;
   pPN = nPN[[i]] / sum;
   pNN = nNN[[i]] / sum;
mean[[i]] = pPP + pNN - pPN - pNP}];
 Label[jump],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

simulation = ListPlot[mean, PlotMarkers → {Automatic, Tiny}];
  negcos = Plot[-Cos[x Degree], {x, 0, 720}, PlotStyle → {Magenta},
    Ticks → {{{0, -360 °}, {90, -270 °}, {180, -180 °}, {270, -90 °}, {360, 0 °}, {450, 90 °},
       {540, 180 °}, {630, 270 °}, {720, 360 °}}, Automatic}, GridLines → Automatic];
```

Compare mean values with -Cosine Curve and compute averages

In[ ]:=
```
Show[negcos, simulation]
```

Out[ ]=



In[ ]:=
```
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[[315]]] - N[mean[[225]]] + N[mean[[405]]] + N[mean[[315]]]]
```

```
AveA = -0.0001106

AveB = -0.000193

P(A+) = 0.499944

P(B+) = 0.499904
```

Out[ ]= 9 999 924

Out[ ]= 0.249846

Out[ ]= 0.249998

Out[ ]= 0.250098

Out[ ]= 0.250058

Out[ ]= 2.73384