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[747]:=
```
m = 100000;
trialDeg = 721;
s1 = ConstantArray[0, m];
s2 = ConstantArray[0, m];
λ1 = ConstantArray[0, m];
A12 = Table[{0, 0, 0, 0}, m];
A22 = Table[{0, 0, 0, 0}, m];
B12 = Table[{0, 0, 0, 0}, m];
B22 = Table[{0, 0, 0, 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 = 3.1; β = 0.32; ξ = -16;  (*Adustable parameters for fine tuning*)
```

Generating Particle Data with Three Independent Do-Loops

In[769]:=
```
Do[θ = RandomReal[{0, 360}];  (*Singlet vector angle*)  (*Hidden Variable*)
 λ1[[i]] = β (Cos[θ/ϕ]^2);
 s1[[i]] = θ;
 s2[[i]] = θ + 180, {i, m}]     (*Conservation of angular momentum*)
```

In[770]:=
```
Do[a = RandomInteger[{0, 360}];  (*Detector vector angle 1 degree increments*)
 If[Abs[Cos[(a - s1[[i]]) Degree]] > λ1[[i]], Aa1 = -Sign[Cos[(a - s1[[i]]) Degree]], Aa1 = 0];
 If[Abs[Cos[(a - s1[[i]]) Degree]] < λ1[[i]],
  Aa2 = -Sign[Sin[(a - s1[[i]] + ξ) Degree]], Aa2 = 0];
 A0 = -Sign[Sin[(a - s1[[i]] + ξ) Degree]];
 A12[[i]] = {a, Aa1, i, A0};
 A22[[i]] = {a, Aa2, i, A0}, {i, m}]
outA1 = DeleteCases[A12, {_, 0, _, _}];  (*Split into outA1 and outA2*)
outA2 = DeleteCases[A22, {_, 0, _, _}];
```

In[772]:=
```
Do[b = RandomInteger[{0, 360}];  (*Detector vector angle 1 degree increments*)
  If[Abs[Cos[(b - s2[[i]]) Degree]] > λ1[[i]], Bb1 = -Sign[Cos[(b - s2[[i]]) Degree]], Bb1 = 0];
  If[Abs[Cos[(b - s2[[i]]) Degree]] < λ1[[i]],
   Bb2 = -Sign[Sin[(b - s2[[i]] + ξ) Degree]], Bb2 = 0];
  B0 = -Sign[Sin[(b - s2[[i]] + ξ) Degree]];
  B12[[i]] = {b, Bb1, i, B0};
  B22[[i]] = {b, Bb2, i, B0}, {i, m}]
outB1 = DeleteCases[B12, {_, 0, _, _}];    (*Split into outB1 and outB2*)
outB2 = DeleteCases[B22, {_, 0, _, _}];
```

Matching Events Observed by Alice and Bob using Trial Numbers

In[774]:=
```
kB1 = outA1[[All, 3]]; (*Two lists of only trial numbers used for matching*)
kA1 = outB1[[All, 3]];
```

Local Detection Analysis of the Events Observed by Alice

In[776]:=
```
listA3 = Select[outA1, Intersection[{#[[3]]}, kA1] != {#[[3]]} &];
M1 = Length[listA3];
tna = listA3[[All, 3]];
outA4 = Select[outA1, Intersection[{#[[3]]}, tna] != {#[[3]]} &];
outA5 = Table[{0, 0, 0, 0}, M1];
ssca = ConstantArray[0, M1];
```

For the spinorial sign changes we will need eq. (12).

$$\mathbf{q}(\eta_{\mathbf{sn}} + \delta\,\pi,\, \mathbf{r}) = (-1)^{\delta}\,\mathbf{q}(\eta_{\mathbf{sn}},\, \mathbf{r}) \quad \text{for } \delta = 0, 1, 2, 3, \ldots$$

In[782]:=
```
Do[If[listA3[[i]][[2]] == listA3[[i]][[4]], δ = 0, δ = 1];
  If[δ == 0, ssca[[i]] = 1, ssca[[i]] = -1];  (*spinorial sign change*)
outA5[[i]] = {listA3[[i]][[1]],
    ssca[[i]] * listA3[[i]][[2]], listA3[[i]][[3]], listA3[[i]][[4]]}, {i, M1}]
outA = Sort[Catenate[{outA2, outA4, outA5}], #1[[3]] < #2[[3]] &];
(*Combine lists and sort*)
a1 = outA[[All, 1]];  (*These results are what Alice observes as defined in Eq.(??)*)
A =  outA[[All, 2]];
```

Local Detection Analysis of the Events Observed by Bob

In[784]:=
```
listB3 = Select[outB1, Intersection[{#[[3]]}, kB1] != {#[[3]]} &];
M2 = Length[listB3];
tnb = listB3[[All, 3]];
outB4 = Select[outB1, Intersection[{#[[3]]}, tnb] != {#[[3]]} &];
outB5 = Table[{0, 0, 0, 0}, M2];
sscb = ConstantArray[0, M2];
```

In[790]:=
```
Do[If[listB3[[i]][[2]] == listB3[[i]][[4]], δ = 0, δ = 1];
  If[δ == 0, sscb[[i]] = 1, sscb[[i]] = -1];
  (*spinorial sign change*)outB5[[i]] = {listB3[[i]][[1]],
    sscb[[i]] * listB3[[i]][[2]], listB3[[i]][[3]], listB3[[i]][[4]]}, {i, M2}]
outB = Sort[Catenate[{outB2, outB4, outB5}], #1[[3]] < #2[[3]] &];
(*Combine lists and sort*)
b1 = outB[[All, 1]];  (*These results are what Bob observes as defined in Eq.(??)*)
B =  outB[[All, 2]];
```

Statistical Analysis of the Particle Data Received from Alice and Bob

In[792]:=
```
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, m}]
```

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

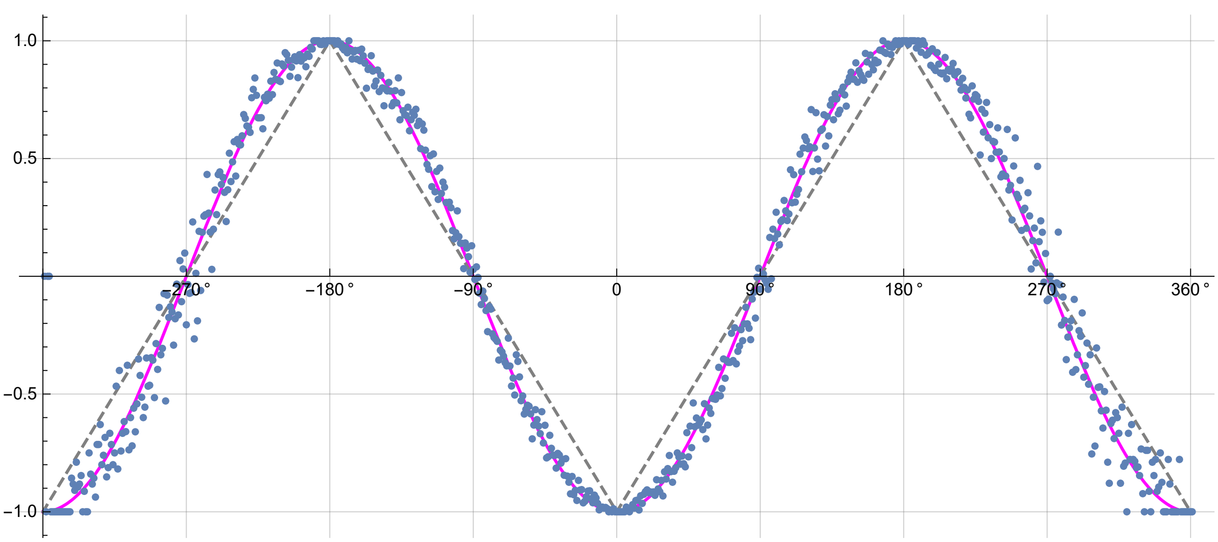
In[793]:=
```
pPP = 0; pPN = 0; pNP = 0; pNN = 0;
mean = ConstantArray[0, trialDeg];
Do[sum = nPP[[i]] + nPN[[i]] + nNP[[i]] + nNN[[i]];
If[sum == 0, Goto[jump],
   {pPP = nPP[[i]] / sum;
    pNP = nNP[[i]] / sum;
    pPN = nPN[[i]] / sum;
    pNN = nNN[[i]] / sum;
mean[[i]] = pPP + pNN - pPN - pNP}];
 Label[jump], {i, trialDeg}]
```

In[796]:=
```
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[802]:=
```
Show[negcos, p1, p2, p3, p4, simulation]
```

Out[802]=



In[803]:=
```
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}]
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[[45]]]]
```

```
AveA = -0.00142

AveB = -0.0007

P(A+)= 0.49929

P(B+)= 0.49965
```

Out[815]= 100 000

Out[816]= 0.24915

Out[817]= 0.25021

Out[818]= 0.25014

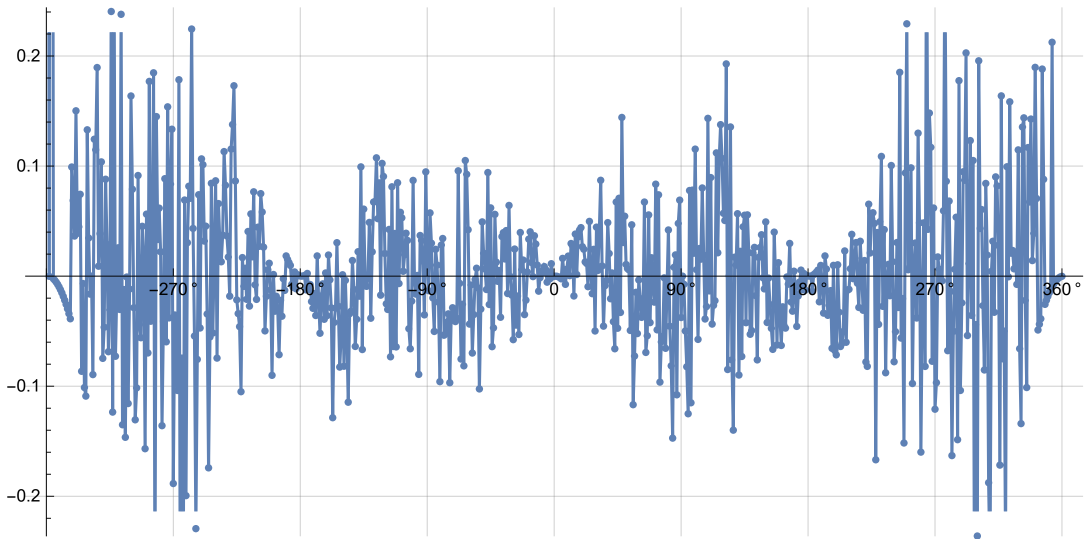
Out[819]= 0.2505

Out[820]= 2.92954

Deviation from negative cosine curve

In[821]:=
```
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[827]=



In[828]:=
```
Mean[N[dev3]]
Mean[N[Abs[dev3]]]
```

Out[828]= 0.00641541

Out[829]= 0.0564116