Simulation Based on Michel Fodje's epr-simple simulation translated from Python to Mathematica by John Reed 13 Nov 2013 and Quaternions Modified by Fred Diether for Completely Local-Realistic Sep 2021 Some parts by Bill Nelson. Includes Joy's S^3 Quaternion Model.

Load Quaternion Package, Set Run Time Parameters, Initialize Arrays and Tables

In[147]:=
```
<< Quaternions`;
β0 = Quaternion[1, 0, 0, 0];
β1 = Quaternion[0, 1, 0, 0];
β2 = Quaternion[0, 0, 1, 0];
β3 = Quaternion[0, 0, 0, 1];
Qcoordinates = {β1, β2, β3};
m = 2000000;
trialDeg = 721;
Ls1 = ConstantArray[0, m];
Ls2 = ConstantArray[0, m];
λ1 = ConstantArray[0, m];
λ2 = ConstantArray[0, m];
qa = ConstantArray[0, m];
aq1 = ConstantArray[0, m];
qa1 = ConstantArray[0, m];
qb = ConstantArray[0, m];
bq1 = ConstantArray[0, m];
qb1 = ConstantArray[0, m];
aa1 = ConstantArray[0, m];
bb1 = ConstantArray[0, m];
outA1 = Table[{0, 0, 0, 0}, m];
outA2 = Table[{0, 0, 0, 0}, m];
outB1 = Table[{0, 0, 0, 0}, m];
outB2 = Table[{0, 0, 0, 0}, m];
plotq = Table[{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; β = 0.32; ξ = 0.933;  (*Adustable parameters for fine tuning*)
```

Generating Particle Data with Three Independent Do-Loops

```
In[185]:= Do[θ = RandomReal[{-179, 180}]; (*Singlet vector angle*) (*Hidden Variable*)
  λ1[[i]] = β (Cos[θ/3]^2);
  λ2[[i]] = ξ (Cos[θ/3]^2);
  θθ = N[Flatten[{FromPolarCoordinates[{1, θ * π / 180}], 0}]];
  Ls1[[i]] = θθ.Qcoordinates;
  Ls2[[i]] = -θθ.Qcoordinates, {i, m}]

In[186]:= Do[a = RandomInteger[{-179, 180}]; (*Detector vector angle 1 degree increments*)
  aa1[[i]] = a;
  aa = N[Flatten[{FromPolarCoordinates[{1, a * π / 180}], 0}]];
  Da = aa.Qcoordinates; (*Convert to quaternion coordinates*)
  qa = Da ** Ls1[[i]];
  qa1[[i]] = qa;
  aq = - (-Da ** Ls1[[i]]);
  aq1[[i]] = aq;
  If[Abs[Re[qa]] > λ1[[i]], Aa1 = Sign[Re[qa]], Aa1 = Re[aq]];
  outA1[[i]] = {a, Aa1, i, qa}, {i, m}]
 Do[If[Abs[Re[qa1[[i]]]] < λ2[[i]], Aa2 = -Sign[qa1[[i]][[4]]], Aa2 = Re[aq1[[i]]]];
  outA2[[i]] = {aa1[[i]], Aa2, i + m, qa1[[i]]}, {i, m}]
 outA = Catenate[{outA1, outA2}];

In[189]:= Do[b = RandomInteger[{-179, 180}]; (*Detector vector angle 1 degree increments*)
  bb1[[i]] = b;
  bb = N[Flatten[{FromPolarCoordinates[{1, b * π / 180}], 0}]];
  Db = bb.Qcoordinates; (*Convert to quaternion coordinates*)
  qb = Ls2[[i]] ** Db;
  qb1[[i]] = qb;
  bq = - (-Ls2[[i]] ** Db);
  bq1[[i]] = bq;
  If[Abs[Re[qb]] > λ1[[i]], Bb1 = Sign[Re[qb]], Bb1 = Re[bq]];
  outB1[[i]] = {b, Bb1, i, qb}, {i, m}]
 Do[If[Abs[Re[qb1[[i]]]] < λ2[[i]], Bb2 = Sign[qb1[[i]][[4]]], Bb2 = Re[bq1[[i]]]];
  outB2[[i]] = {bb1[[i]], Bb2, i + m, qb1[[i]]}, {i, m}]
 outB = Catenate[{outB1, outB2}];
```

Statistical Analysis of the Particle Data Received from Alice and Bob

```
In[192]:= m2 = 2 m;
 theta = ConstantArray[0, m2];
 a1 = outA[[All, 1]]; A = outA[[All, 2]]; b1 = outB[[All, 1]]; B = outB[[All, 2]];
 Do[th = a1[[i]] - b1[[i]] + 361;
  (*All angles are shifted by 361 degrees since θ1 is an index*)
  theta[[i]] = th;
  aliceD = A[[i]]; bobD = B[[i]];
  If[aliceD == 1, nAP[[th]]++];
  If[bobD == 1, nBP[[th]]++];
  If[aliceD == -1, nAN[[th]]++];
  If[bobD == -1, nBN[[th]]++];
  If[aliceD == 1 && bobD == 1, nPP[[th]]++];
  If[aliceD == 1 && bobD == -1, nPN[[th]]++];
  If[aliceD == -1 && bobD == 1, nNP[[th]]++];
  If[aliceD == -1 && bobD == -1, nNN[[th]]++], {i, m2}]
```

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

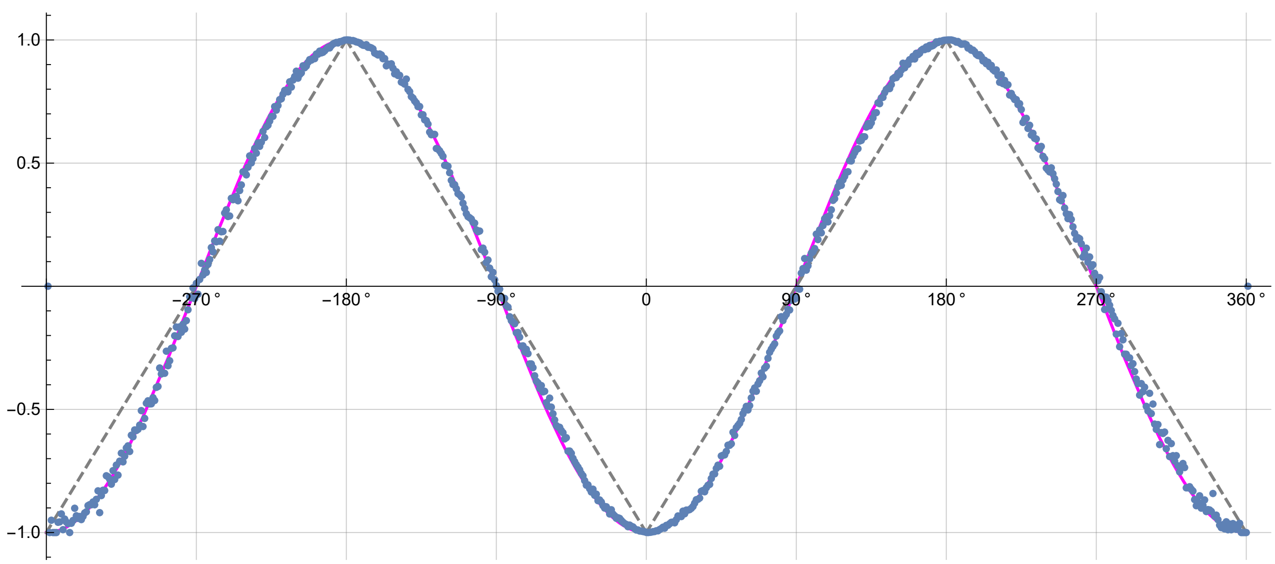
In[196]:=
```
mean = ConstantArray[0, trialDeg];
sum1 = ConstantArray[0, trialDeg];
sum2 = ConstantArray[0, trialDeg];
Do[sum1[[i]] = (nPP[[i]] + nNN[[i]] - nPN[[i]] - nNP[[i]]);
 sum2[[i]] = nPP[[i]] + nPN[[i]] + nNP[[i]] + nNN[[i]] + 0.0000001;
mean[[i]] = sum1[[i]] / sum2[[i]], {i, trialDeg}]
```

In[200]:=
```
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[206]:=
```
Show[negcos, p1, p2, p3, p4, simulation]
```

Out[206]=



In[207]:=

```
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 = -0.000115901

AveB = 0.0014283

P(A+) = 0.500033

P(B+) = 0.500685

Total Events = 1990970

Ave ++ = 0.250471

Ave -- = 0.249677

Ave +- = 0.249462

Ave -+ = 0.25039

Approx. CHSH = 2.74057
```

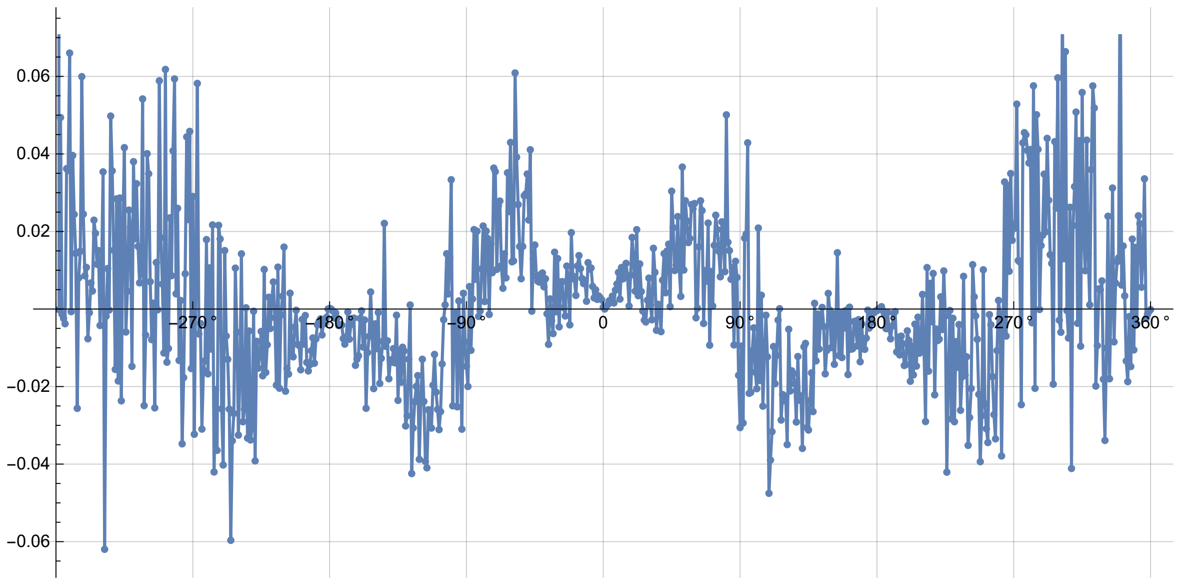
Product Calculation

In[231]:=
```
(*q=0;
q2a=outA[[All,4]];
q2b=outB[[All,4]];
plotq=Table[{0,0},m];
angle=ConstantArray[0,m];
Do[If[λ2[[i]]==1,q=q2a[[i]]**q2b[[i]],q=q2b[[i]]**q2a[[i]]];
  angle=theta[[i]]-361;
  plotq[[i]]={angle,Re[q]},{i,m}]
 sim=ListPlot[plotq,PlotMarkers→{Automatic,Small},AspectRatio→7/16,
   Ticks→{{{-360,-360°},{-270,-270°},{-180,-180°},{-90,-90°},{0,0°},{90,90°},
      {180,180°},{270,270°},{360,360°}},Automatic},GridLines→Automatic];
negcos1=Plot[-Cos[x Degree],{x,-360,360},PlotStyle→{Magenta}];
Show[sim,negcos1]*)
```

Blue is the correlation data and magenta is the -cosine curve for an exact match.

In[232]:=
```
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[238]=



In[239]:=
```
N[Mean[Abs[dev3]]]
N[Mean[dev3]]
```

Out[239]= 0.0169498

Out[240]= 0.00273285