**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[632]:=
```
<< 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 = 200000;
trialDeg = 721;
Ls1 = ConstantArray[0, m];
Ls2 = ConstantArray[0, m];
λ1 = ConstantArray[0, m];
λ2 = ConstantArray[0, m];
outAa = Table[{0, 0, 0, 0, 0, 0, 0}, m];
outBb = Table[{0, 0, 0, 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 = 3.1; β = 0.32; ξ = -4π/45;        (*Adustable parameters for fine tuning*)
```

Generating Particle Data with Three Independent Do-Loops

In[660]:=
```
Do[θ = RandomReal[{-179, 180}]; (*Singlet vector angle*) (*Hidden Variable*)
 λ1[[i]] = β (Cos[θ/ϕ]^2);
 λ2[[i]] = Sign[θ];
 θθ = N[Flatten[{FromPolarCoordinates[{1, θ * π/180}], 0}]];
 Ls1[[i]] = λ2[[i]] * θθ.Qcoordinates;
 Ls2[[i]] = -Ls1[[i]], {i, m}]
```

```
In[661]:= Do[a = RandomInteger[{-179, 180}]; (*Detector vector angle 1 degree increments*)
  aa = N[Flatten[{FromPolarCoordinates[{1, a * π / 180}], 0}]];
  Da = aa.Qcoordinates; (*Convert to quaternion coordinates*)
  qa = Da ** Ls1[[i]];
  aq = -(Ls1[[i]] ** Da);
  If[Abs[Re[qa]] < λ1[[i]], C1 = f1, C1 = g1];
  If[Abs[Re[qa]] > λ1[[i]],
   Aa = Re[Da ** Limit[ Ls1[[i]], Ls1[[i]] -> Sign[Re[Da ** Ls1[[i]]]] Da]],
   Aa = λ2[[i]] * Sign[qa[[4]] + ξ]];
  A5 = λ2[[i]] * Sign[qa[[4]] + ξ];
  outAa[[i]] = {a, Aa, i, C1, A5, qa, aq}, {i, m}]
 outA1 = Select[outAa, MemberQ[#, g1] &]; (*Split outAa into outA1 and outA2*)
 outA2 = Select[outAa, MemberQ[#, f1] &];

In[663]:= Do[b = RandomInteger[{-179, 180}]; (*Detector vector angle 1 degree increments*)
  bb = N[Flatten[{FromPolarCoordinates[{1, b * π / 180}], 0}]];
  Db = bb.Qcoordinates; (*Convert to quaternion coordinates*)
  qb = Ls2[[i]] ** Db;
  bq = -(Db ** Ls2[[i]]);
  If[Abs[Re[qb]] < λ1[[i]], C2 = f2, C2 = g2];
  If[Abs[Re[qb]] > λ1[[i]],
   B = Re[Db ** Limit[ Ls2[[i]], Ls2[[i]] -> Sign[Re[Db ** Ls2[[i]]]] Db]],
   B = -λ2[[i]] * Sign[qb[[4]] + ξ]];
  B5 = -λ2[[i]] * Sign[qb[[4]] + ξ];
  outBb[[i]] = {b, B, i, C2, B5, qb, bq}, {i, m}]
 outB1 = Select[outBb, MemberQ[#, g2] &]; (*Split outBb into outB1 and outB2*)
 outB2 = Select[outBb, MemberQ[#, f2] &];
```

## Matching Events Observed by Alice and Bob using Trial Numbers

```
In[665]:= list13 = outA1[[All, 3]]; (*Two lists of only trial numbers used for matching*)
 list23 = outB1[[All, 3]];
```

Local Detection Analysis of the Events Observed by Alice

```
In[667]:= listA4 = Select[outA1, Intersection[{#[[3]]}, list23] == {#[[3]]} &];
 (*Events in outA1 that match go to listA4*)
 listad3 = listA4[[All, 3]];
 listA3 = Select[outA1, Intersection[{#[[3]]}, listad3] != {#[[3]]} &];
 (*Events in outA1 that do not match go to listA3*)

In[670]:= M = Length[listA3];
 listA6 = Table[{0, 0, 0, 0, 0, 0, 0}, M]; (*Quaternion sign change*)
 qaaq = ConstantArray[0, M];
 listA36 = listA3[[All, 6]];
 listA37 = listA3[[All, 7]];
 Do[If[listA3[[i]][[2]] == listA3[[i]][[5]],
   qaaq[[i]] = 1, qaaq[[i]] = Re[listA36[[i]] ** listA37[[i]]]];
  listA6[[i]] = {listA3[[i]][[1]], qaaq[[i]] * listA3[[i]][[2]], listA3[[i]][[3]],
    listA3[[i]][[4]], listA3[[i]][[5]], listA3[[i]][[6]], listA3[[i]][[7]]}, {i, M}]

In[676]:= outA = Sort[Catenate[{listA4, outA2, listA6}], #1[[3]] < #2[[3]] &];
 (*Combine lists and sort*)
 a1 = outA[[All, 1]];
 A = outA[[All, 2]]; (*These results are what Alice observes as defined in Eq.(13)*)
```

Local Detection Analysis of the Events Observed by Bob

```
In[679]:= listB4 = Select[outB1, Intersection[{#[[3]]}, list13] == {#[[3]]} &];
(*Events in outB1 that match go to listB4*)
listbd3 = listB4[[All, 3]];
listB3 = Select[outB1, Intersection[{#[[3]]}, listbd3] != {#[[3]]} &];
(*Events in outB1 that do not match go to listB3*)

In[682]:= M2 = Length[listB3];
listB6 = Table[{0, 0, 0, 0, 0, 0, 0}, M2]; (*Quaternion sign change*)
qbbq = ConstantArray[0, M2];
listB36 = listB3[[All, 6]];
listB37 = listB3[[All, 7]];
Do[If[listB3[[i]][[2]] == listB3[[i]][[5]],
  qbbq[[i]] = 1, qbbq[[i]] = Re[listB36[[i]] ** listB37[[i]]]];
 listB6[[i]] = {listB3[[i]][[1]], qbbq[[i]] * listB3[[i]][[2]], listB3[[i]][[3]],
   listB3[[i]][[4]], listB3[[i]][[5]], listB3[[i]][[6]], listB3[[i]][[7]]}, {i, M2}]

In[688]:= outB = Sort[Catenate[{listB4, outB2, listB6}], #1[[3]] < #2[[3]] &];
(*Combine lists and sort*)
b1 = outB[[All, 1]];
B =  outB[[All, 2]]; (*These results are what Bob observes as defined in Eq.(20)*)
```

## Statistical Analysis of the Particle Data Received from Alice and Bob

```
In[691]:= theta = ConstantArray[0, m];
Do[θ1 = a1[[j]] - b1[[j]] + 361;
 (*All angles are shifted by 361 degrees since θ1 is an index*)
 theta[[j]] = θ1;
 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

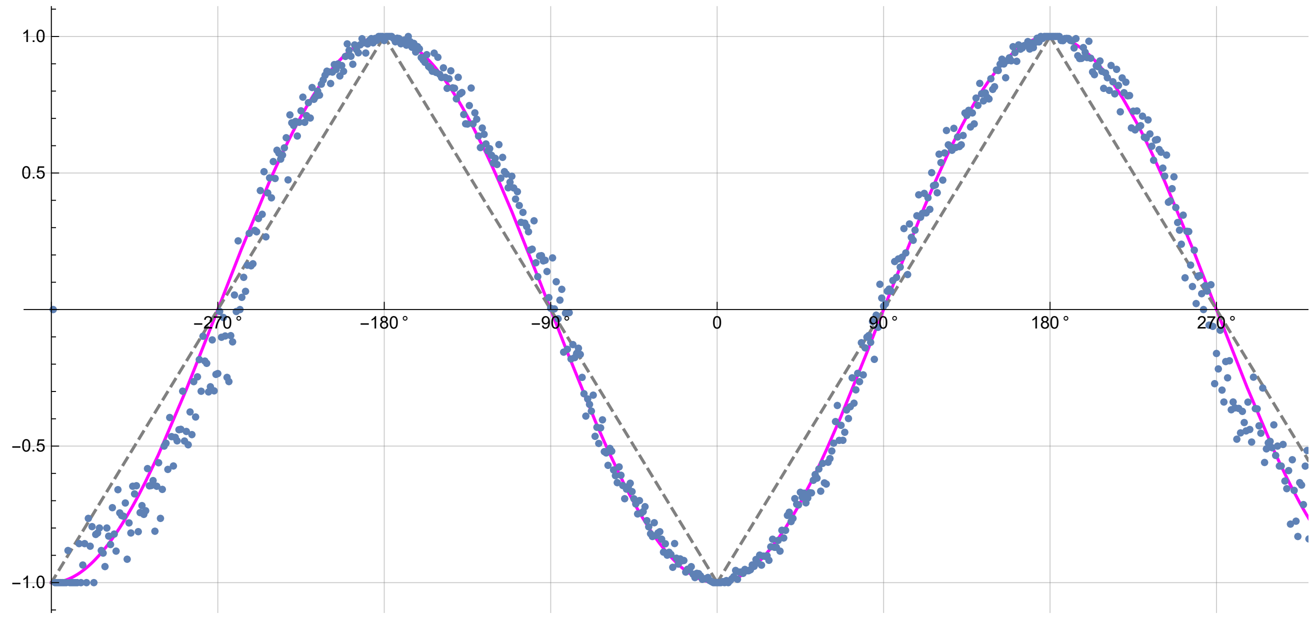
```
In[693]:= pPP = 0; pPN = 0; pNP = 0; pNN = 0;
mean = ConstantArray[0, trialDeg];
Do[sum = nPP[[i]] + nPN[[i]] + nNP[[i]] + nNN[[i]];
If[sum == 0, Goto[jump],
  {pPP = nPP[[i]] / sum;
   pNP = nNP[[i]] / sum;
   pPN = nPN[[i]] / sum;
   pNN = nNN[[i]] / sum;
mean[[i]] = pPP + pNN - pPN - pNP}];
 Label[jump], {i, trialDeg}]
```

In[696]:=
```
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[702]:=
```
Show[negcos, p1, p2, p3, p4, simulation]
```

Out[702]=



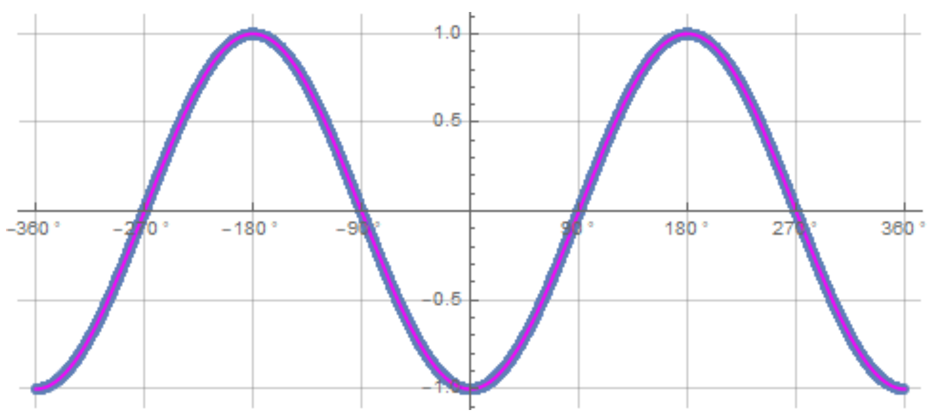

In[703]:=
```
AveA = N[Sum[A[[i]], {i, m}] / m];
AveB = N[Sum[B[[i]], {i, m}] / m];
Print["AveA = ", AveA]
Print["AveB = ", AveB]
PAP = N[Sum[nAP[[i]], {i, trialDeg}]];
PBP = N[Sum[nBP[[i]], {i, trialDeg}]];
PAN = N[Sum[nAN[[i]], {i, trialDeg}]];
PBN = N[Sum[nBN[[i]], {i, trialDeg}]];
PA1 = PAP / (PAP + 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 = 0.00271

AveB = 0.00062

P(A+) = 0.501355

P(B+) = 0.50031

Out[715]= 200 000

Out[716]= 0.250695

Out[717]= 0.24903

Out[718]= 0.25066

Out[719]= 0.249615

Out[720]= 2.71645
```

## Product Calculation

```
q = 0;
q2a = outA[[All, 6]];
q2b = outB[[All, 6]];
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]
```

```
In[738]:= Import["prodcalc200k.png"]
```

Out[738]=



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

```
(*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]*)
```