**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 July 2021**
**Some parts by Bill Nelson**

## Set Run Time Parameters, Initialize Arrays and Tables

In[274]:=
```
m = 10 000;
trialDeg = 720;
s = ConstantArray[0, m];
λ = ConstantArray[0, m];
outA1 = Table[{0, 0, 0, 0, 0}, m];
outB1 = Table[{0, 0, 0, 0, 0}, m];
outA2 = Table[{0, 0, 0, 0, 0}, m];
outB2 = Table[{0, 0, 0, 0, 0}, m];
CA1 = Table[{0, 0, 0, 0, 0}, m];
CB1 = Table[{0, 0, 0, 0, 0}, m];
listAa1 = Table[{0, 0, 0, 0, 0}, m];
listBb1 = Table[{0, 0, 0, 0, 0}, m];
listAa2 = Table[{0, 0, 0, 0, 0}, m];
listBb2 = Table[{0, 0, 0, 0, 0}, m];
listAa6 = Table[{0, 0, 0, 0, 0}, m];
listBb6 = Table[{0, 0, 0, 0, 0}, m];
listAa7 = Table[{0, 0, 0, 0, 0}, m];
listBb7 = Table[{0, 0, 0, 0, 0}, m];
a1 = ConstantArray[0, m];
b1 = ConstantArray[0, m];
A1 = ConstantArray[0, m];
B1 = 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];
```

## Generate Particle Data with 3 Do Loops

In[304]:=
```
Do[e = RandomReal[{0, 360}];  (*Singlet vector angle*)
 s[[i]] = e;
 λ[[i]] = 0.25 (Cos[e * 90 / 2]^2), (*Hidden Variables*)
 {i, m}]
```

In[305]:=
```
Do[a = RandomChoice[{0, 90}];  (*Detector vector angle*)
 If[If[Abs[Cos[((a - s[[i]]) Degree)]] < λ[[i]], C1 = f1, C1 = g1];
  Abs[Cos[((a - s[[i]]) Degree)]] > λ[[i]], A = -Sign[Cos[((a - s[[i]]) Degree)]],
  A = -Sign[Cos[((a - s[[i]]) Degree)]] Sign[Cot[((a - s[[i]]) Degree)]]];
 AA = -Sign[Cos[((a - s[[i]]) Degree)]] Sign[Cot[((a - s[[i]]) Degree)]];
 CA1[[i]] = {a, A, i, C1, AA}, {i, m}]
outA1 = Select[CA1, MemberQ[#, g1] &];
outA2 = Select[CA1, MemberQ[#, f1] &];
```

In[308]:=
```
Do[b = RandomChoice[{45, 135}];  (*Detector vector angle*)
 If[If[Abs[Cos[((b - s[[i]]) Degree)]] < λ[[i]], C2 = f2, C2 = g2];
  Abs[Cos[(b - s[[i]]) Degree]] > λ[[i]], B = Sign[((Cos[(b - s[[i]]) Degree]))],
  B = Sign[((Cos[(b - s[[i]]) Degree]))] Sign[((Cot[(b - s[[i]]) Degree]))]];
 BB = Sign[((Cos[(b - s[[i]]) Degree]))] Sign[((Cot[(b - s[[i]]) Degree]))];
 CB1[[i]] = {b, B, i, C2, BB}, {i, m}]
outB1 = Select[CB1, MemberQ[#, g2] &];
outB2 = Select[CB1, MemberQ[#, f2] &];
```

## Match Trial Numbers

In[311]:=
```
listad = outA1[[All, 3]]; (*Match Trial Numbers*)
listbd = outB1[[All, 3]];
listAa1 = Select[outA1, Intersection[{#[[3]]}, listbd] == {#[[3]]} &];
listBb1 = Select[outB1, Intersection[{#[[3]]}, listad] == {#[[3]]} &];
listad2 = outA1[[All, 3]];
listad3 = listAa1[[All, 3]];
listAa3 = Select[outA1, Intersection[{#[[3]]}, listad3] != {#[[3]]} &];
listAa4 = Select[listAa1, Intersection[{#[[3]]}, listad2] != {#[[3]]} &];
listbd2 = outB1[[All, 3]];
listbd3 = listBb1[[All, 3]];
listBb3 = Select[outB1, Intersection[{#[[3]]}, listbd3] != {#[[3]]} &];
listBb4 = Select[listBb1, Intersection[{#[[3]]}, listbd2] != {#[[3]]} &];
M = Length[listAa3];
listAa7 = Table[{0, 0, 0, 0, 0}, M];
a2 = ConstantArray[0, M];
A2 = ConstantArray[0, M];
ind2 = ConstantArray[0, M];
A3 = ConstantArray[0, M];
A5 = ConstantArray[0, M];
A4 = ConstantArray[0, M];
A6 = ConstantArray[0, M];
a2 = listAa3[[All, 1]];
A2 = listAa3[[All, 2]];
ind2 = listAa3[[All, 3]];
A5 = listAa3[[All, 5]];
Do[A4 = A2[[i]]; A6 = A5[[i]];
 If[A4 == A6, A2 = A2, A2 = A5];
 listAa7[[i]] = {a2[[i]], A2[[i]], ind2[[i]], f1, A5[[i]]}, {i, M}]
M2 = Length[listBb3];
listBb7 = Table[{0, 0, 0, 0, 0}, M2];
b2 = ConstantArray[0, M2];
B2 = ConstantArray[0, M2];
ind3 = ConstantArray[0, M2];
B3 = ConstantArray[0, M2];
B5 = ConstantArray[0, M2];
B4 = ConstantArray[0, M2];
B6 = ConstantArray[0, M2];
b2 = listBb3[[All, 1]];
B2 = listBb3[[All, 2]];
ind3 = listBb3[[All, 3]];
B5 = listBb3[[All, 5]];
Do[B4 = B2[[i]]; B6 = B5[[i]];
 If[B4 == B6, B2 = B2, B2 = B5];
 listBb7[[i]] = {b2[[i]], B2[[i]], ind3[[i]], f1, B5[[i]]}, {i, M2}]
outA4 = Sort[Catenate[{outA2, listAa7}], #1[[3]] < #2[[3]] &];
outB4 = Sort[Catenate[{outB2, listBb7}], #1[[3]] < #2[[3]] &];
outA5 = Catenate[{listAa1, outA4}];
outB5 = Catenate[{listBb1, outB4}];
a2 = outA5[[All, 1]];
b2 = outB5[[All, 1]];
A1 = outA5[[All, 2]];
B1 = outB5[[All, 2]];
```

## CHSH Analysis of Particle Data

In[356]:=
```
nP1 = 0; nN1 = 0; nP2 = 0; nN2 = 0; nP3 = 0; nN3 = 0; nP4 = 0; nN4 = 0;
M = Length[outA5];
Do[a1 = a2[[j]];
 b1 = b2[[j]];
 aliceD = A1[[j]]; bobD = B1[[j]];
 If[(b1 == 45) && (a1 - b1 == -45) && aliceD * bobD == 1, nP1++];
 If[(b1 == 45) && (a1 - b1 == -45) && aliceD * bobD == -1, nN1++];
 If[(a1 - b1) == -135 && aliceD * bobD == 1, nP2++];
 If[(a1 - b1) == -135 && aliceD * bobD == -1, nN2++];
 If[(a1 - b1) == 45 && aliceD * bobD == 1, nP3++];
 If[(a1 - b1) == 45 && aliceD * bobD == -1, nN3++];
 If[a1 == 90 && (a1 - b1) == -45 && aliceD * bobD == 1, nP4++];
 If[a1 == 90 && (a1 - b1) == -45 && aliceD * bobD == -1, nN4++], {j, M}]
E1 = N[(nP1 - nN1) / (nP1 + nN1)];
E2 = N[(nP2 - nN2) / (nP2 + nN2)];
E3 = N[(nP3 - nN3) / (nP3 + nN3)];
E4 = N[(nP4 - nN4) / (nP4 + nN4)];
CHSH = Abs[E1 - E2 + E3 + E4];
Print["CHSH = ", CHSH]
```

```
CHSH = 2.64946
```