Simulation Based on Michel Fodje's epr-simple simulation translated from Python to Mathematica by John Reed 13 Nov 2013
Further modified by Fred Diether Jan 2021

Set Run Time Parameters, Initialize Arrays and Tables

In[50]:=
```
trials = 3 000 000;
trialDeg = 720;
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 2 Do Loops (2 separate stations)

In[66]:=
```
Do[a = RandomInteger[{0, 360}];  (*Detector vector angle 1 degree increments*)
 If[Cos[(a) Degree] < 0, A = -1, A = 1];
 CA[[j]] = {a, A}, {j, trials}]
```

In[67]:=
```
Do[b = RandomInteger[{0, 360}];  (*Detector vector angle 1 degree increments*)
 If[Cos[(b) Degree] < 0, B = 1, B = -1];
 CB[[j]] = {b, B}, {j, trials}]
```

Statistical Analysis of Particle Data

In[68]:=
```
a1 = CA[[All, 1]];
b1 = CB[[All, 1]];
A1 = CA[[All, 2]];
B1 = CB[[All, 2]]; (*Data Extraction*)
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}]
```

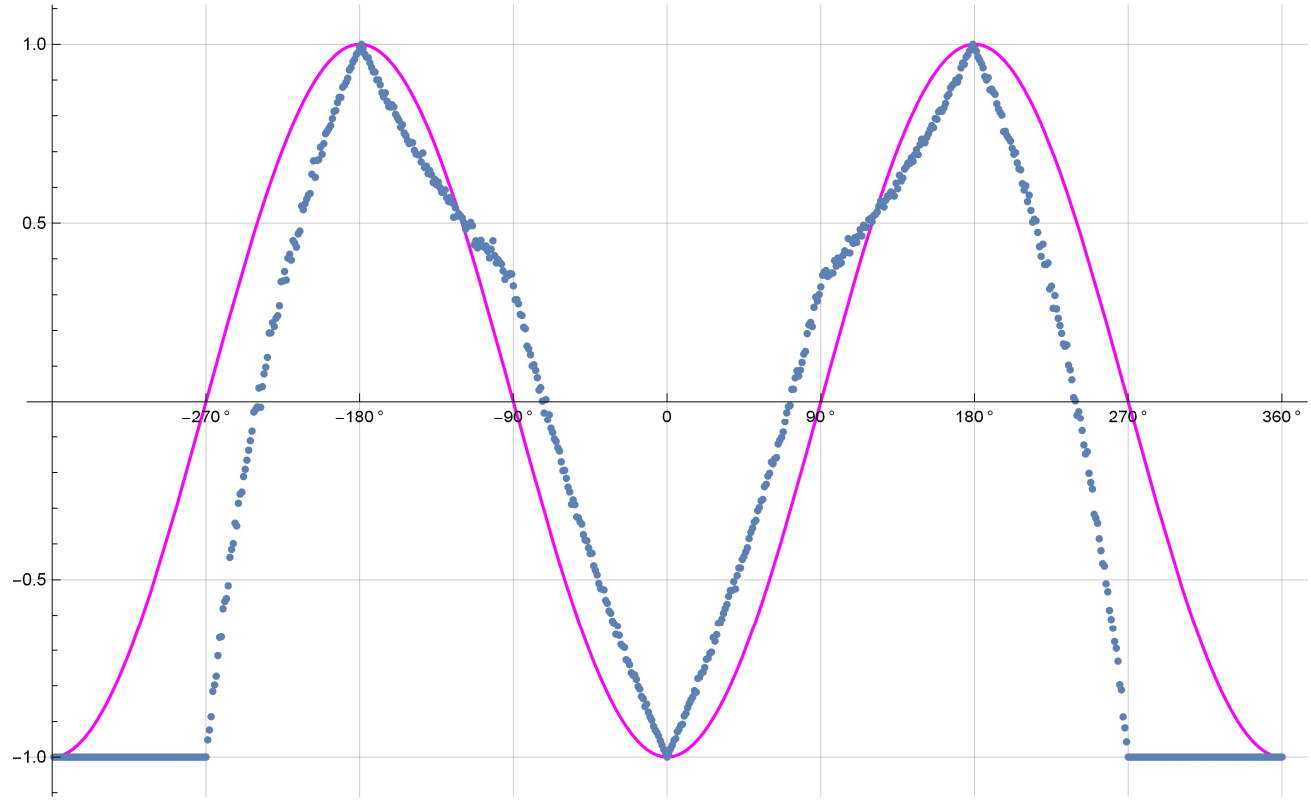
Calculate Mean Values

In[70]:=
```
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}]
```

Compare mean values with -Cosine Curve and compute averages

In[73]:=
```
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];
Show[negcos, simulation]
```

Out[75]=



In[76]:=
```
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[[333]]] + N[mean[[387]]] - N[mean[[505]]] + N[mean[[315]]]];
Print["CHSH = ", CHSH]
```

```
AveA = 0.00884267
AveB = -0.00899067
P(A+)= 0.504418
P(B+)= 0.495508
```

Out[88]= 2 999 977

Out[89]= 0.24977

Out[90]= 0.249844

Out[91]= 0.254647

Out[92]= 0.245738

```
CHSH = 2.46647
```