

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[1362]:= m = 300 000;  
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[1392]:= Do[e = RandomReal[{0, 360}]; (*Singlet vector angle*)  
s[[i]] = e;  
λ[[i]] = 0.25 (Cos[ $\frac{e}{2}$ ] ^ 2), (*Hidden Variables*)  
{i, m}]
```

```

In[1393]:= Do[a = RandomInteger[{0, 360}]; (*Detector vector angle 1 degree increments*)
  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[Sin[(a - s[[i]]) Degree]]];
  AA = -Sign[Sin[(a - s[[i]]) Degree]];
  CA1[[i]] = {a, A, i, C1, AA}, {i, m}
outA1 = Select[CA1, MemberQ[#, g1] &];
outA2 = Select[CA1, MemberQ[#, f1] &];

In[1396]:= Do[b = RandomInteger[{0, 360}]; (*Detector vector angle 1 degree increments*)
  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[Sin[(b - s[[i]]) Degree]]];
  BB = Sign[Sin[(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 and do Statistical Analysis of Particle Data

```

In[1399]:= 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 = Sort[Catenate[{listAa1, outA4}], #1[[3]] < #2[[3]] &];
outB5 = Sort[Catenate[{listBb1, outB4}], #1[[3]] < #2[[3]] &];
trials2 = Length[outA5];
a1 = outA5[[All, 1]];
b1 = outB5[[All, 1]];
A1 = outA5[[All, 2]];
B1 = outB5[[All, 2]];
Do[ $\theta$  = a1[[j]] - b1[[j]] + 360; (*All angles are shifted by  $2\pi$  since  $\theta$  is an index*)
  aliceD = A1[[j]]; bobD = B1[[j]];
  If[aliceD == 1, nAP[[ $\theta$ ]] ++];
  If[bobD == 1, nBP[[ $\theta$ ]] ++];
  If[aliceD == -1, nAN[[ $\theta$ ]] ++];
  If[bobD == -1, nBN[[ $\theta$ ]] ++];
  If[aliceD == 1 && bobD == 1, nPP[[ $\theta$ ]] ++];
  If[aliceD == 1 && bobD == -1, nPN[[ $\theta$ ]] ++];
  If[aliceD == -1 && bobD == 1, nNP[[ $\theta$ ]] ++];
  If[aliceD == -1 && bobD == -1, nNN[[ $\theta$ ]] ++], {j, trials2}

```

Calculate Mean Values and Plot

```

In[1446]:= 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}

```

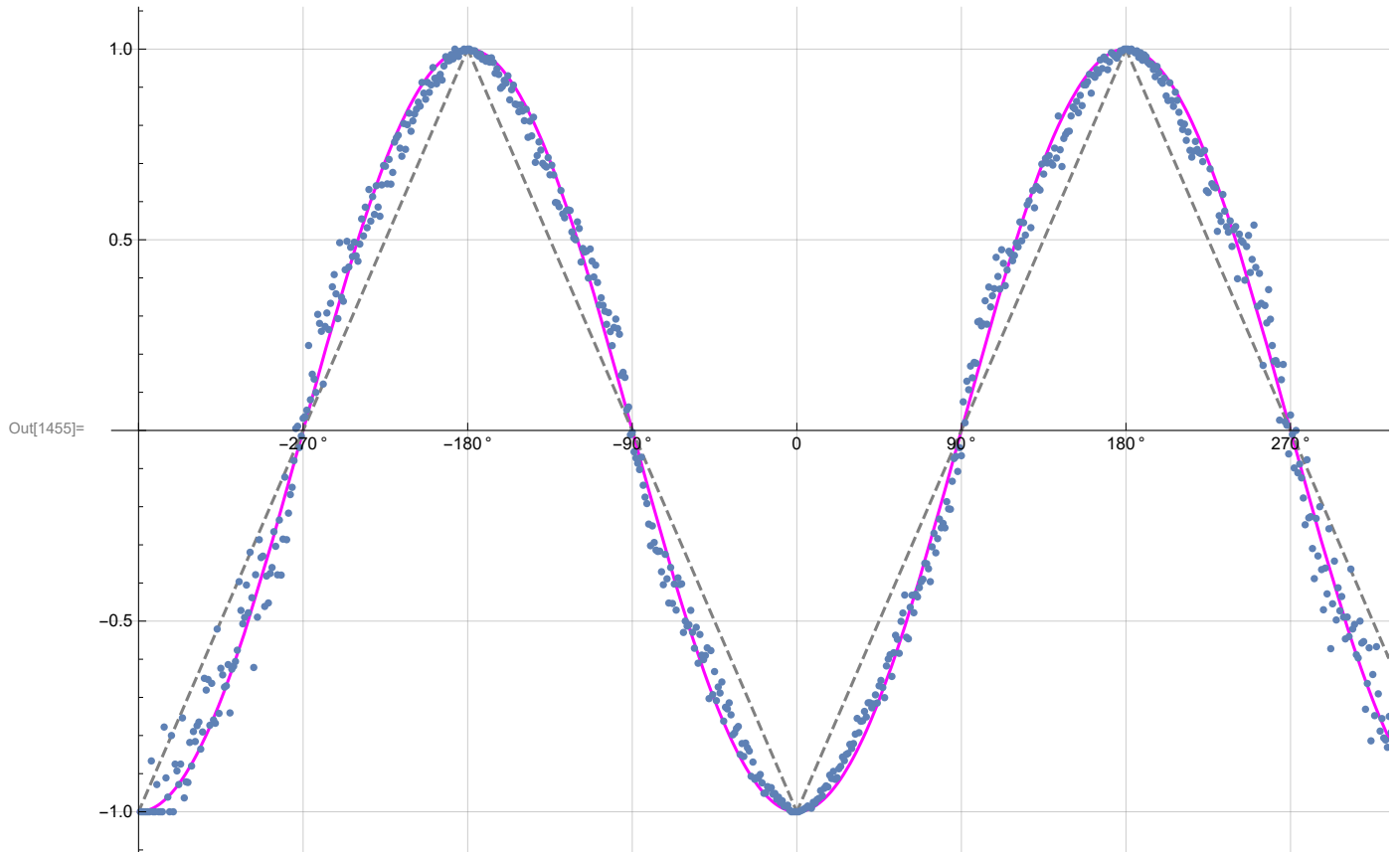
```

In[1449]:= 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];
p1 = Plot[-1 + (2 x Degree) /  $\pi$ , {x, 0, 180}, PlotStyle -> {Gray, Dashed}];
p2 = Plot[3 - 2 x Degree /  $\pi$ , {x, 180, 360}, PlotStyle -> {Gray, Dashed}];
p3 = Plot[-5 + 2 x Degree /  $\pi$ , {x, 360, 540}, PlotStyle -> {Gray, Dashed}];
p4 = Plot[7 - 2 x Degree /  $\pi$ , {x, 540, 720}, PlotStyle -> {Gray, Dashed}];

```

Compare mean values with -Cosine Curve and compute averages

In[1455]:= Show[negcos, p1, p2, p3, p4, simulation]



```
In[1456]:= AveA = N[Sum[A1[[i]], {i, trials2}]/trials2];
AveB = N[Sum[B1[[i]], {i, trials2}]/trials2];
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 = 0.00209333

AveB = 0.00016

P (A+) = 0.501048

P (B+) = 0.500078

Out[1468]= 299 999

Out[1469]= 0.251224

Out[1470]= 0.250098

Out[1471]= 0.249824

Out[1472]= 0.248854

Out[1473]= 2.59608

In[1474]:= trials2

Out[1474]= 300 000

In[1477]:= outA5 [[All, 3]]

outB5 [[All, 3]]

Out[]:=

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44,
 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65,
 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, ... 299 840 ..., 299 921,
 299 922, 299 923, 299 924, 299 925, 299 926, 299 927, 299 928, 299 929, 299 930, 299 931,
 299 932, 299 933, 299 934, 299 935, 299 936, 299 937, 299 938, 299 939, 299 940, 299 941,
 299 942, 299 943, 299 944, 299 945, 299 946, 299 947, 299 948, 299 949, 299 950, 299 951,
 299 952, 299 953, 299 954, 299 955, 299 956, 299 957, 299 958, 299 959, 299 960, 299 961,
 299 962, 299 963, 299 964, 299 965, 299 966, 299 967, 299 968, 299 969, 299 970, 299 971,
 299 972, 299 973, 299 974, 299 975, 299 976, 299 977, 299 978, 299 979, 299 980, 299 981,
 299 982, 299 983, 299 984, 299 985, 299 986, 299 987, 299 988, 299 989, 299 990, 299 991,
 299 992, 299 993, 299 994, 299 995, 299 996, 299 997, 299 998, 299 999, 300 000}
```

large output

show less

show more

show all

set size limit...

Out[]:=

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44,
 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65,
 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, ... 299 840 ..., 299 921,
 299 922, 299 923, 299 924, 299 925, 299 926, 299 927, 299 928, 299 929, 299 930, 299 931,
 299 932, 299 933, 299 934, 299 935, 299 936, 299 937, 299 938, 299 939, 299 940, 299 941,
 299 942, 299 943, 299 944, 299 945, 299 946, 299 947, 299 948, 299 949, 299 950, 299 951,
 299 952, 299 953, 299 954, 299 955, 299 956, 299 957, 299 958, 299 959, 299 960, 299 961,
 299 962, 299 963, 299 964, 299 965, 299 966, 299 967, 299 968, 299 969, 299 970, 299 971,
 299 972, 299 973, 299 974, 299 975, 299 976, 299 977, 299 978, 299 979, 299 980, 299 981,
 299 982, 299 983, 299 984, 299 985, 299 986, 299 987, 299 988, 299 989, 299 990, 299 991,
 299 992, 299 993, 299 994, 299 995, 299 996, 299 997, 299 998, 299 999, 300 000}
```

large output

show less

show more

show all

set size limit...