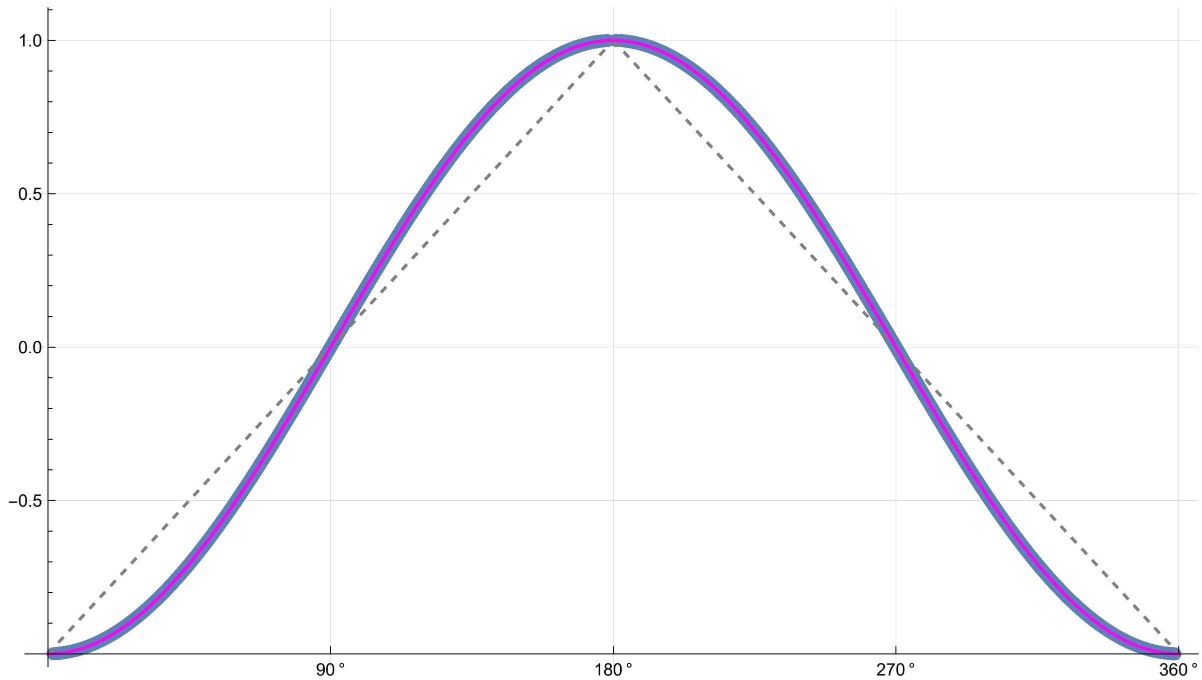
Validation of the Local QM Product Calculation Prediction Using Pauli Matrices with 3D vectors, Created by Fred Diether Feb. 2022

```
In[125]:= m = 20000;
s1 = ConstantArray[0, m];
s2 = ConstantArray[0, m];
a1 = ConstantArray[0, m];
b1 = ConstantArray[0, m];
AA = ConstantArray[0, m];
BB = ConstantArray[0, m];
r1 = ConstantArray[0, m];
r2 = ConstantArray[0, m];
A = ConstantArray[0, m];
B = ConstantArray[0, m];
plotAB = Table[{0, 0}, m];

In[137]:= Do[s = RandomPoint[Sphere[]];  (*Uniform Unit 3D Vectors; Hidden Variable*)
 s1[[i]] = s;
 s2[[i]] = -s, {i, m}]

In[138]:= Do[a = RandomPoint[Sphere[]];  (*Uniform Unit 3D Vectors*)
 a1[[i]] = a;
 σa = PauliMatrix[1] * a[[1]] + PauliMatrix[2] * a[[2]] + PauliMatrix[3] * a[[3]];
 σs1 = PauliMatrix[1] * s1[[i]][[1]] + PauliMatrix[2] * s1[[i]][[2]] + PauliMatrix[3] * s1[[i]][[3]];
 AA[[i]] = Flatten[1/2 ((1 0).σa.σs1.(1
                                      0) + (0 1).σa.σs1.(0
                                                          1))];
 (*Particle - Detecter interaction*)
 A[[i]] = Sign[Extract[Re[AA[[i]]], 1]];
 r1[[i]] = Cross[a, s1[[i]]], {i, m}]

In[139]:= Do[b = RandomPoint[Sphere[]];  (*Uniform Unit 3D Vectors*)
 b1[[i]] = b;
 σb = PauliMatrix[1] * b[[1]] + PauliMatrix[2] * b[[2]] + PauliMatrix[3] * b[[3]];
 σs2 = PauliMatrix[1] * s2[[i]][[1]] + PauliMatrix[2] * s2[[i]][[2]] + PauliMatrix[3] * s2[[i]][[3]];
 (*Particle - Detecter interaction*)
 BB[[i]] = Flatten[1/2 ((1 0).σs2.σb.(1
                                      0) + (0 1).σs2.σb.(0
                                                          1))];
 B[[i]] = Sign[Extract[Re[BB[[i]]], 1]];
 r2[[i]] = Cross[b, s2[[i]]], {i, m}]
```

In[140]:=
```
Do[AB = Extract[AA⟦i⟧ * BB⟦i⟧ + r1⟦i⟧.r2⟦i⟧, 1];  (*Product Calculation*)
 ϕa = ArcTan[a1⟦i⟧⟦1⟧, a1⟦i⟧⟦2⟧];
 ϕb = ArcTan[b1⟦i⟧⟦2⟧, b1⟦i⟧⟦1⟧];
 If[ϕa * ϕb > 0, angle = ArcCos[a1⟦i⟧.b1⟦i⟧] / Degree,
  angle = (2 π - ArcCos[a1⟦i⟧.b1⟦i⟧]) / Degree];
 plotAB⟦i⟧ = {angle, AB}, {i, m}]
```

In[141]:=
```
simulation = ListPlot[plotAB, PlotMarkers → {Automatic, Small}, AspectRatio → 9 / 16,
   Ticks → {{{90, 90 °}, {180, 180 °}, {0, 0 °}, {270, 270 °}, {360, 360 °}}, Automatic},
   GridLines → Automatic, AxesOrigin → {0, -1.0}];
negcos = Plot[-Cos[x Degree], {x, 0, 360}, PlotStyle → {Magenta}];
p1 = Plot[-1 + 2 x1 Degree / π, {x1, 0, 180}, PlotStyle → {Gray, Dashed}];
p2 = Plot[3 - 2 x2 Degree / π, {x2, 180, 360}, PlotStyle → {Gray, Dashed}];
Show[ simulation, p1, p2, negcos]
```

Out[145]=



In[146]:=
```
AveA = N[Total[A] / m];
AveB = N[Total[B] / m];
Print[" <A> = ", AveA, "  <B> = ", AveB];
meanAB = Mean[plotAB⟦All, 2⟧];
Print["meanAB = ", meanAB];
```

```
 <A> = 0.0211  <B> = -0.0111

meanAB = 0.0022129 + 0. i
```