```
In[503]:= (*Validation of the Local QM Product Calculation Using Pauli Matrices*)
ss = 0; t = 0; u = 0;
m = 100 000;
plotArray = Table[{0, 0}, m];
For[nn = 1, nn ≤ m, nn += 1,
 {a = Normalize@RandomVariate[NormalDistribution[], 3]; (*Uniform Unit 3D Vectors*)
  b = Normalize@RandomVariate[NormalDistribution[], 3];
  s = Normalize@RandomVariate[NormalDistribution[], 3];
  If[a.s > 0, s1 = a, s1 = -a]; (*Polarizer Functions; note s1=s2*)
  If[b.s > 0, s2 = b, s2 = -b];
  λ = RandomChoice[{-1, 1}];
  (*Measurement Functions*)
  A = λ/2 ((1 0).(PauliMatrix[1] * a[[1]] + PauliMatrix[2] * a[[2]] + PauliMatrix[3] * a[[3]]).
       (PauliMatrix[1] * (-s1[[1]]) + PauliMatrix[2] * (-s1[[2]]) + PauliMatrix[3] * (-s1[[3]])).(1
                                                                                                 0) +
      (0 1).(PauliMatrix[1] * a[[1]] + PauliMatrix[2] * a[[2]] + PauliMatrix[3] * a[[3]]).
       (PauliMatrix[1] * (-s1[[1]]) + PauliMatrix[2] * (-s1[[2]]) + PauliMatrix[3] * (-s1[[3]])).(0
                                                                                                 1));
  B = λ/2 ((1 0).(PauliMatrix[1] * s2[[1]] + PauliMatrix[2] * s2[[2]] + PauliMatrix[3] * s2[[3]]).
       (PauliMatrix[1] * b[[1]] + PauliMatrix[2] * b[[2]] + PauliMatrix[3] * b[[3]]).(1
                                                                                     0) +
      (0 1).(PauliMatrix[1] * s2[[1]] + PauliMatrix[2] * s2[[2]] + PauliMatrix[3] * s2[[3]]).
       (PauliMatrix[1] * b[[1]] + PauliMatrix[2] * b[[2]] + PauliMatrix[3] * b[[3]]).(0
                                                                                     1));
  q = 0;
  (*And now... we have the Product Calculation conforming to eq. (16) with s1=s2*)
  If[λ == 1, q = λ/2 ((1 0).(PauliMatrix[1] * a[[1]] + PauliMatrix[2] * a[[2]] + PauliMatrix[3] * a[[3]]).
        (PauliMatrix[1] * (-s2[[1]]) + PauliMatrix[2] * (-s2[[2]]) + PauliMatrix[3] * (-s2[[3]])).(1
                                                                                                  0) +
       (0 1).(PauliMatrix[1] * a[[1]] + PauliMatrix[2] * a[[2]] + PauliMatrix[3] * a[[3]]).
        (PauliMatrix[1] * (-s2[[1]]) + PauliMatrix[2] * (-s2[[2]]) + PauliMatrix[3] * (-s2[[3]])).(0
                                                                                                  1)) *
     λ/2 ((1 0).(PauliMatrix[1] * s2[[1]] + PauliMatrix[2] * s2[[2]] + PauliMatrix[3] * s2[[3]]).
        (PauliMatrix[1] * b[[1]] + PauliMatrix[2] * b[[2]] + PauliMatrix[3] * b[[3]]).(1
                                                                                      0) +
       (0 1).(PauliMatrix[1] * s2[[1]] + PauliMatrix[2] * s2[[2]] + PauliMatrix[3] * s2[[3]]).
        (PauliMatrix[1] * b[[1]] + PauliMatrix[2] * b[[2]] + PauliMatrix[3] * b[[3]]).(0
                                                                                      1)),
```

```
q = λ/2 ((1 0).(PauliMatrix[1] * b[[1]] + PauliMatrix[2] * b[[2]] + PauliMatrix[3] * b[[3]]).
      (PauliMatrix[1] * s1[[1]] + PauliMatrix[2] * s1[[2]] + PauliMatrix[3] * s1[[3]]).(1
                                                                                     0) +
     (0 1).(PauliMatrix[1] * b[[1]] + PauliMatrix[2] * b[[2]] + PauliMatrix[3] * b[[3]]).
      (PauliMatrix[1] * s1[[1]] + PauliMatrix[2] * s1[[2]] + PauliMatrix[3] * s1[[3]]).(0
                                                                                     1)) *
   λ/2 ((1 0).(PauliMatrix[1] * (-s1[[1]]) + PauliMatrix[2] * (-s1[[2]]) + PauliMatrix[3] * (-s1[[3]])).
      (PauliMatrix[1] * a[[1]] + PauliMatrix[2] * a[[2]] + PauliMatrix[3] * a[[3]]).(1
                                                                                  0) +
     (0 1).(PauliMatrix[1] * (-s1[[1]]) + PauliMatrix[2] * (-s1[[2]]) + PauliMatrix[3] * (-s1[[3]])).
      (PauliMatrix[1] * a[[1]] + PauliMatrix[2] * a[[2]] + PauliMatrix[3] * a[[3]]).(0
                                                                                  1))];
 ss = ss + q;
 t = t + A;
 u = u + B;
 ϕA = ArcTan[a[[2]], a[[1]]];
 ϕB = ArcTan[b[[2]], b[[1]]];
 If[ϕA * ϕB > 0, angle = ArcCos[a.b] / Degree, angle = (-ArcCos[a.b] + 2 π) / Degree];
 plotArray[[nn]] = {angle, Extract[Flatten[Re[q]], 1]};};]
mean = ss / m (*shows vanishing of the non-real part*)
avA = t / m;
avB = u / m;
Print[" <A> = ", avA, "  <B> = ", avB]
ListPlot[plotArray]
```

Out[507]= {{0.00089322 + 0. i}}

<A> = {{-0.00092 + 0. i}}  <B> = {{0.0003 + 0. i}}