```
(*
SpinCorrSim.nb
  Copyright 2011 by Robert Close

  Find correlation between spins obtained by sampling points
  at different angles on a sphere divided into + and - hemispheres.

  History:  Bell_Sine.nb written 2/14/2011 by Robert Close (RC):
    orientation expressed as rotation of initial z-axis.
               SpinCorr.nb written 9/5/2011 by RC:
                device A samples points on a sphere with weighting for density of states.
                 SpinCorrSim.nb
        written 9/12/2011 by RC: uniform sampling of points on sphere.
                           Modified 9/16/2011 by RC to clarify variable names.
*)

thetabi = .;
theta = .;
phi = .;
thetaAB = .;
corrAB = .;
np = .;
na = .;

(* Sampling of thetab *)
np = 40;
(* Sampling of spin rotation angles *)
na = 80;

Array[thetaAB, np];
Array[corrAB, np];
(* Loop through thetab values 0 to Pi *)
For[i = 1, i ≤ np, i++,
  thetaAB[i] = Pi (i - 1 / 2) / np;
  thetaABi = thetaAB[i];

  snorm = 0;
  corrAB[i] = 0;
  corra = 0;
  corrb = 0;

  (* Get systematic orientation of rotation
   axes using polar coordinates: 0<theta<Pi/2; 0<phi<2Pi *)
  For[nth = 1, nth ≤ na, nth++,
   thetaA = Pi (nth - 1 / 2) / na;

   nphi = IntegerPart[2 na Abs[Sin[thetaA]] + 0.5];
   For[nph = 1, nph ≤ 2 nphi, nph++,
    phi = 2 Pi (nph - 1 / 2) / (2 nphi);

    spinA = N[Sign[Cos[phi] Sin[thetaA]]];
```

```
        spinB = N[Sign[Cos[phi] Sin[thetaA + thetaABi]]];

        corrAB[i] += N[spinA spinB ];

        (* Density of rotation axes is Sin[theta] dtheta dphi *)
        snorm += 1;
        corra += N[spinA];
        corrb += N[spinB];
       ]; (* For nph *)
      ]; (* For nth *)
      (* Print["snorm= ",snorm]; *)
      corrAB[i] = corrAB[i] / snorm;
      (* Uncomment to check values *)
      (*
      Print["thetabi = ",thetabi];
      Print["corrab = ",corrab[i]];
      Print["Avg spin A = ",N[corra/snorm]];
      Print["Avg spin B = ",N[corrb/snorm]];
      *)
     ]; (* For[i *)

    rmserr = 0;
    For[i = 1, i ≤ np, i++,
      thetabi = Pi (i - 1 / 2) / np;
      rmserr += N[(corrAB[i] - Cos[thetaAB[i]])^2];
     ];
    rmserr = N[Sqrt[rmserr / np]];
    Print["rmserr=", rmserr];

    ListPlot[Table[{thetaAB[i], corrAB[i]}, {i, 0, np}]]
```

```
rmserr=0.00236736
```