Exercise 11
Level Spacing Distribution for impurity model

Hamiltonian

```
(* PARAMETERS OF THE HAMILTONIAN *)
Clear[L, upspins, downspins, dim, Jxy, Jz, ϵ, defectsite, alpha, JJxy, JJz];
L = 15;
upspins = L / 3;
downspins = L - upspins;
dim = L! / (upspins! downspins!);
(* Impurity *)
ϵ = 0.5;
defectsite = Floor[L / 2];
(* Parameters for nearest-neighbor couplings *)
Jxy = 1.0;
Jz = 0.5;
(* Parameters for next-nearest-neighbor couplings *)
(* Set alpha=0 if NNN couplings do not exist *)
alpha = 0.0;
JJxy = 1.0;
JJz = 0.5;

(* BASIS *)
Clear[onebasisvector, basis];
onebasisvector = Flatten[{Table[1, {k, 1, upspins}], Table[0, {k, 1, downspins}]}];
basis = Permutations[onebasisvector];

(* ELEMENTS OF THE HAMILTONIAN *)
(* Impurity and NN couplings *)
Clear[HH];
(* Initialization *)
Do[Do[HH[i, j] = 0., {i, 1, dim}], {j, 1, dim}];

(* Diagonal elements *)
Do[
  (* Impurity *)
  If[basis[[i, defectsite]] == 1,
   HH[i, i] = HH[i, i] + ϵ / 2.,  HH[i, i] = HH[i, i] - ϵ / 2.];
  (* Ising interaction *)
  Do[If[basis[[i, j]] == basis[[i, j + 1]],
     HH[i, i] = HH[i, i] + Jz / 4., HH[i, i] = HH[i, i] - Jz / 4.];
   , {j, 1, L - 1}];
  , {i, 1, dim}];

(* Off-diagonal elements *)
Clear[howmany, site];
Do[
  Do[
    (* Initialization *)
```

```
    howmany = 0;
    Do[site[kk] = 0, {kk, 1, L}];
    (* Sites where states i and j differ *)
    Do[
     If[basis[[i, k]] ≠ basis[[j, k]], {howmany = howmany + 1, site[howmany] = k}];
     , {k, 1, L}];
    (* If only two neighbor sites differ, there is a coupling matrix element *)
    If[howmany == 2,
     If[site[2] - site[1] == 1, {HH[i, j] = Jxy / 2., HH[j, i] = Jxy / 2.}]];
    , {j, i + 1, dim}];
  , {i, 1, dim - 1}];

(* ELEMENTS OF THE HAMILTONIAN WITH NNN COUPLINGS *)
If[alpha > 0,

(* Diagonal elements *)
  Do[
   (* Ising interaction *)
   Do[If[basis[[i, j]] == basis[[i, j + 2]],
       HH[i, i] = HH[i, i] + alpha JJz / 4., HH[i, i] = HH[i, i] - alpha JJz / 4.];
     , {j, 1, L - 2}];
   , {i, 1, dim}];

  (* Off-diagonal elements *)
  Clear[howmany, site];
  Do[
   Do[
     (* Initialization *)
     howmany = 0;
     Do[site[kk] = 0, {kk, 1, L}];
     (* Sites where states i and j differ *)
     Do[
      If[basis[[i, k]] ≠ basis[[j, k]], {howmany = howmany + 1, site[howmany] = k}];
      , {k, 1, L}];
     (* If only two next-neighbor sites differ,
     there is a coupling matrix element *)
     If[howmany == 2,
      If[site[2] - site[1] == 2,
       {HH[i, j] = alpha JJxy / 2., HH[j, i] = alpha JJxy / 2.}]];
     , {j, i + 1, dim}];
   , {i, 1, dim - 1}];

 ];

(* TOTAL HAMILTONIAN AND DIAGONALIZATION *)
Clear[Hamiltonian, Energy, Vector];
Hamiltonian = Table[Table[HH[i, j], {j, 1, dim}], {i, dim}];
Energy = Eigenvalues[Hamiltonian];
Vector = Eigenvectors[Hamiltonian];
```

Level spacing distribution for NN+defect (parity is not taken into account)

```
(* LEVEL SPACINGS OF THE UNFOLDED SPECTRUM *)
(* Order the eigenvalues from lowest to highest values *)
```

```
Clear[Ener];
Ener = Sort[Table[Energy[[k]], {k, 1, dim}]];

(* Discard ~10% of the eigenvalues located at the borders of the spectrum *)
Clear[percentage, half, spacing];
percentage = 0.1 dim;
half = Floor[percentage / 2.];
Do[
  Clear[average];
  (* Compute the neighboring level spacings
   for the remaining eigenvalues after unfolding them *)
  (* Unfolding here means that the average of each
    group of 10 level spacings = 1 *)
  average = (Ener[[half + 10 j]] - Ener[[half + 10 (j - 1)]]) / 10.;
  Do[spacing[i] = (Ener[[half + i]] - Ener[[(half - 1) + i]]) / average;
   , {i, 1 + 10 (j - 1), 10 j}];
  , {j, 1, Floor[(dim - percentage) / 10]}];

(* HISTOGRAM *)
Clear[spcmin, spcmax, bin, Nofbins];
spcmin = 0.;
spcmax = 8.;
bin = 0.1;
Nofbins = IntegerPart[(spcmax - spcmin) / bin];

Clear[SPChist, Nhist];
SPChist[1] = spcmin;
Do[SPChist[i + 1] = SPChist[i] + bin, {i, 1, Nofbins}];
Do[Nhist[j] = 0., {j, 1, Nofbins}];

(* Nhist[j] gives how many spacings we
 have in the interval SPChist[j+1] and SPChist[j] *)
Do[
  Do[
    If[SPChist[j] ≤ spacing[k] < SPChist[j + 1], Nhist[j] = Nhist[j] + 1];
    , {j, 1, Nofbins}];
  , {k, 1, 10 Floor[(dim - percentage) / 10]}];

(* Normalization *)
Clear[Norma];
Norma = Sum[bin Nhist[j], {j, 1, Nofbins}];
Do[Nhist[j] = Nhist[j] / Norma, {j, 1, Nofbins}];

(* ListPlot with the obtained data *)
Clear[jj, nl];
jj = 0;
nl = {};
Do[jj += 1;
  nl = Append[nl, {SPChist[jj], Nhist[jj]}];
  nl = Append[nl, {SPChist[jj + 1], Nhist[jj]}];
  , {j, 1, Nofbins - 1}];
DataPlot =
  ListPlot[nl, Joined → True, PlotRange → {{0, 8}, {0, 1}}, PlotStyle → {Black, Thick},
   LabelStyle → Directive[Black, Bold, Medium], AxesLabel → {"s", "P"}];
(* Theoretical curves *)
```

```
WignerDyson = Plot[Pi s / 2. Exp[-Pi s^2 / 4.],
   {s, 0, 8}, PlotRange → {0, 1}, PlotStyle → {Red, Thick},
   LabelStyle → Directive[Black, Bold, Medium], AxesLabel → {"s", "P"}];
Poisson = Plot[Exp[-s], {s, 0, 8}, PlotRange → {0, 1}, PlotStyle → {Blue, Thick},
   LabelStyle → Directive[Black, Bold, Medium], AxesLabel → {"s", "P"}];
(* The three curves together *)
Show[{DataPlot, WignerDyson, Poisson}, PlotRange → {{0, 4}, {0, 1.1}}]
```