Dynamic

- Description

We study the time evolution of the system for a particular initial state corresponding to one of the basis vectors.
We plot the magnetization of each site, with a magnetization of 0.5 corresponding to an upspin, and a magnetization
of -0.5 corresponding to a downspin.
We also plot the probability of finding each of the basis vectors is time.
(The code provided here is used to obtain Figure 2 in the paper).

- Parameters explained

*) initialstate =initial state corresponding to one of the site-basis vectors
*) endtime = how many times the loop for time is repeated (choice depends on the system)
*) increment = dt = discrete interval of time (choice depends on the system)
*) Psi[t] = evolved state

- Code for the Dynamics
- Hamiltonian, eigenvalues and eigenstates

```
(*Parameters of the Hamiltonian*)
Clear[chainsize, upspins, downspins, dim, Jxy, Jz, open];
chainsize = 4;
upspins = 2;
downspins = chainsize - upspins;
dim = chainsize! / (upspins! downspins!);
Jxy = 1.0;
Jz = 0.5;
open = 1;
(*Creating the basis*)
Clear[onebasisvector, basis];
onebasisvector =
  Flatten[{Table[1, {k, 1, upspins}], Table[0, {k, 1, downspins}]}];
basis = Permutations[onebasisvector];

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

(*Diagonal elements-Ising interaction*)
Do[
  Do[
    HH[i, i] = HH[i, i] + (Jz / 4.) * (-1.) ^ (basis[[i, k]] + basis[[i, k + 1]]);
```

```
    , {k, 1, chainsize - 1}];
, {i, 1, dim}];
(*Term included in the Ising interaction if the chain is closed*)
If[open == 0,
  Do[HH[i, i] = HH[i, i] + (Jz / 4.) * (-1.)^(basis[[i, chainsize]] + basis[[i, 1]]),
   {i, 1, dim}]];

(*Off-diagonal elements-flip-flop term*)
Clear[howmany, site];
Do[
  Do[
    (*Initialization*)
howmany = 0
Do[site[z] = 0, {z, 1, chainsize}];
(*Sites where states i and j differ*)
Do[If[basis[[i, k]] ≠ basis[[j, k]], {howmany = howmany + 1, site[howmany] = k}];,
     {k, 1, chainsize}];
(*Coupling matrix element-when only two neighbor sites differ*)
If[howmany == 2, If[site[2] - site[1] == 1,
       {HH[i, j] = HH[i, j] + Jxy / 2., HH[j, i] = HH[j, i] + Jxy / 2.}]];
(*Additional term for closed system*)If[open == 0, If[site[2] - site[1] ==
        chainsize - 1, {HH[i, j] = HH[i, j] + Jxy / 2., HH[j, i] = HH[j, i] + Jxy / 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];

(*Choose an initial state*)
Do[
  If[basis[[i, 1]] == 1 && basis[[i, 2]] == 1, initialstate1 = i];
  If[basis[[i, 3]] == 1 && basis[[i, 4]] == 1, initialstate2 = i];
, {i, 1, dim}];
InState = (Vector[[All, initialstate1]] - Vector[[All, initialstate2]]) / Sqrt[2];
(*DYNAMICS*)
Clear[endtime, PSI1, PSI2, PSI, increment];
endtime = 201;
increment = 0.1;
Do[
  (*PSI1[t]=Sum[Vector[[j,initialstate1]]
     Vector[[j]] Exp[-I Energy[[j]](t-1) increment],{j,1,dim}];
  PSI2[t]=Sum[Vector[[j,initialstate2]]Vector[[j]]
     Exp[-I Energy[[j]](t-1) increment],{j,1,dim}];
  PSI[t]=(PSI1[t]-PSI2[t])/Sqrt[2];*)
```

```
  PSI[t] =
    Sum[InState[[j]] Vector[[j]] Exp[-I Energy[[j]] (t - 1) increment], {j, 1, dim}];
, {t, 1, endtime}];

(*Call Package for Legends*)
<< PlotLegends`
Clear[ProbT];
Do[ProbT[k] = Table[{(t - 1) increment, Abs[PSI[t][[k]]]^2}, {t, 1, endtime}],
  {k, 1, dim}];

Print[];
Print["Probability of each site-basis"];
ListPlot[Table[ProbT[k], {k, 1, dim}], PlotRange → All,
 Joined → True, PlotStyle → {{Thick, Red}, {Thick, Blue}, {Thick, Brown},
   {Thick, Darker[Green]}, {Thick, Magenta}, {Thick, Dashed, Black} },
 LabelStyle → Directive[Black, Bold, Medium],
 PlotLegend → Table[basis[[k]], {k, 1, dim}], LegendPosition → {1, 0},
 LegendSize → {1, 1}, AxesLabel → {"time", "Probability"}]
```

```
Probability of each site-basis
```