

## 导入

## Parameters

```
PR = {{1.*^-3, 1.*^12}, {1.*^-14, 1.*^1}};
```

# Model

图



```
In[•]:= Ham = funHamiltonianQubit[Model];
```

## Spectrum

```

In[ ]:= {EE, ES} = funSpectrum[Ham];
HamNorm = Max[Abs[EE]];
           |... |绝对值
EE = EE / HamNorm;
Eg = EE[[1]]

Out[ ]:= -0.721501

In[ ]:= If[StringMatchQ[HamType, "Heisenberg"], (
           |... |字符串匹配判定
           htot = 3. * Length[EL] / HamNorm
               |长度
           )]
If[StringMatchQ[HamType, "FermiHubbard"], (
           |... |字符串匹配判定
           htot = (2. * Length[EL] + u / 4. * Nq / 2.) / HamNorm
               |长度
           )]

Out[ ]:= 1.17428

```

## Reference state

```

In[ ]:= If[StringMatchQ[HamType, "Heisenberg"], (
           |... |字符串匹配判定
            $\psi$  = funPairwiseSinglet[Nq]
           )];
If[StringMatchQ[HamType, "FermiHubbard"], (
           |... |字符串匹配判定
            $\psi$  = funHartreeFock[Nq, EL]
           )];
 $\psi$  = Conjugate[ES]. $\psi$ ;
           |共轭
 $\psi$  = Flatten[ $\psi$ ];
           |压平
Pro $\psi$  = Abs[ $\psi$ ] ^ 2;
           |绝对值

In[ ]:= pg = Pro $\psi$ [[1]]
ER = Total[Pro $\psi$  * EE];
           |总计
eR = ER - Eg

Out[ ]:= 0.60369

Out[ ]:= 0.0223459

```

# Run

```

In[ ]:= dList = Table[d, {d, 2, 30}];
      | 表格

plotsRTE = {};
plotsF = {};
plots = {};
γ2εK = {};

In[ ]:= Do[
      | Do循环

      d = dList[[1]];
      Ide = IdentityMatrix[d];
      | 单位矩阵

      E0 = Eg + 1.;
      {Hmat, Smat} = funMatPower[EE, Proψ, d, E0];
      EB = Hmat[[d, d]] / Smat[[d, d]];
      εB = EB - Eg;
      logη = -15;
      η = 10.logη;
      {EK, cn} = funDiagonalisation[Hmat + 2. * η * Ide, Smat + 2. * η * Ide];
      εK = EK - Eg;
      If[εK > 1.*^-2 || εK < 1.*^-9, (
      | 如果

        Print[{d, εK}];
        | 打印

        Goto[end];
        | 转到

      )];
      {γList, εList} = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, 1., 1., logηList];
      γListP = γList;
      εListP = εList;
      εMin = Min[εList];
      | 最小值

      If[εMin < 2. * εK, (
      | 如果

        γ = funInterpolation[εList, γList, 2. * εK];
        ), (
        γ = 0.;
        )];
      γP = γ;

      E0 = 0;
      {Hmat, Smat} = funMatChebyshev[EE, Proψ, d, htot, E0];
      {γList, εList} = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, 1., 1., logηList];
      γListCP = γList;
      εListCP = εList;
      εMin = Min[εList];
      | 最小值

```

```

    取'J'IE
If[ $\epsilon_{\text{Min}} < 2. * \epsilon_K$ , (
  如果
     $\gamma = \text{funInterpolation}[\epsilon_{\text{List}}, \gamma_{\text{List}}, 2. * \epsilon_K]$ ;
  ), (
     $\gamma = 0.$ ;
  )];
 $\gamma_{\text{CP}} = \gamma$ ;

E0 = Eg;
 $\tau_{\text{MIN}} = 0$ ;
 $\tau_{\text{MAX}} = 64$ ;
Do[ (
  Do循环
     $\tau = (\tau_{\text{MIN}} + \tau_{\text{MAX}}) / 2.$ ;
    {Hmat, Smat} = funMatGaussianPower[EE, Pro $\psi$ , 1, htot,  $\tau$ , E0];
    EK = Hmat[[1, 1]] / Smat[[1, 1]];
    err = EK - Eg;
    If[err >  $\epsilon_B$ ,  $\tau_{\text{MIN}} = \tau$ ];
    如果
      If[err <  $\epsilon_B$ ,  $\tau_{\text{MAX}} = \tau$ ];
    如果
  ), {j, 1, 30}];
 $\tau = (\tau_{\text{MIN}} + \tau_{\text{MAX}}) / 2.$ ;
 $\delta = \text{RandomReal}[\{-0.1, 0.1\}]$ ;
  伪随机实数
E0 = Eg +  $\delta$ ;
{Hmat, Smat} = funMatGaussianPower[EE, Pro $\psi$ , d, htot,  $\tau$ , E0];
{ $\gamma_{\text{List}}$ ,  $\epsilon_{\text{List}}$ } = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, htot, 1., log $\eta_{\text{List}}$ ];
 $\gamma_{\text{ListGP}} = \gamma_{\text{List}}$ ;
 $\epsilon_{\text{ListGP}} = \epsilon_{\text{List}}$ ;
 $\epsilon_{\text{Min}} = \text{Min}[\epsilon_{\text{List}}]$ ;
  最小值
If[ $\epsilon_{\text{Min}} < 2. * \epsilon_K$ , (
  如果
     $\gamma = \text{funInterpolation}[\epsilon_{\text{List}}, \gamma_{\text{List}}, 2. * \epsilon_K]$ ;
  ), (
     $\gamma = 0.$ ;
  )];
 $\gamma_{\text{GP}} = \gamma$ ;

E0 = Eg - 1.;
{Hmat, Smat} = funMatInversePower[EE, Pro $\psi$ , d, E0];
{ $\gamma_{\text{List}}$ ,  $\epsilon_{\text{List}}$ } = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, 1., 1., log $\eta_{\text{List}}$ ];
 $\gamma_{\text{ListIP}} = \gamma_{\text{List}}$ ;
 $\epsilon_{\text{ListIP}} = \epsilon_{\text{List}}$ ;
 $\epsilon_{\text{Min}} = \text{Min}[\epsilon_{\text{List}}]$ ;
  最小值
If[ $\epsilon_{\text{Min}} < 2. * \epsilon_K$ , (
  如果
     $\gamma = \text{funInterpolation}[\epsilon_{\text{List}}, \gamma_{\text{List}}, 2. * \epsilon_K]$ ;
  ), (

```

```

    γ = 0.;
  ]];
γIP = γ;

E0 = Eg;
τMIN = 0;
τMAX = 64;
Do[ (
  |Do循环
    τ = (τMIN + τMAX) / 2.;
    {Hmat, Smat} = funMatITE[EE, Proψ, d, τ, E0];
    EK = Hmat[[d, d]] / Smat[[d, d]];
    err = EK - Eg;
    If[err > εB, τMIN = τ];
    |如果
    If[err < εB, τMAX = τ];
    |如果
  ), {j, 1, 30}]];
τ = (τMIN + τMAX) / 2.;
E0 = Eg;
{Hmat, Smat} = funMatITE[EE, Proψ, d, τ, E0];
{γList, εList} = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, 1., 1., logηList];
γListITE = γList;
εListITE = εList;
εMin = Min[εList];
|最小值
If[εMin < 2. * εK, (
  |如果
    γ = funInterpolation[εList, γList, 2. * εK];
  ), (
    γ = 0.;
  )];
γITE = γ;

ΔtList = Table[ $\frac{2. * \text{PI}}{100}$  j, {j, 1, 100}];
|表格

εKList = ΔtList;
Do[ (
  |Do循环
    Δt = ΔtList[[j]];

    E0 = Eg;
    {Hmat, Smat} = funMatRTE[EE, Proψ, d, Δt, E0];
    logη = -15;
    η = 10.logη;
    {EK, cn} = funDiagonalisation[Hmat + 2. * η * Ide, Smat + 2. * η * Ide];
    err = EK - Eg;

    εKList[[j]] = err
  ), {j, 1, Length[ΔtList]};
  |长度

```

```

Δt = ΔtList[[Position[εKList, Min[εKList]][[1, 1]]];
      位置      最小值
AppendTo[plotsRTE, ListLogPlot[Transpose[{ΔtList, εKList}], PlotRange → Full]];
      附加      点集的对数图      转置      绘制范围      全范围
E0 = Eg;
{Hmat, Smat} = funMatRTE[EE, Proψ, d, Δt, E0];
{γList, εList} = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, 1., 1., logηList];
γListRTE = γList;
εListRTE = εList;
εMin = Min[εList];
      最小值
If[εMin < 2. * εK, (
  如果
    γ = funInterpolation[εList, γList, 2. * εK];
  ), (
    γ = 0.;
  )];
γRTE = γ;

ΔE = 0.;
E0 = Eg;
τMIN = 0;
τMAX = 64;
Do[(
  Do循环
    T = (τMIN + τMAX) / 2.;
    {Hmat, Smat} = funMatFilter[EE, Proψ, d, T, E0, ΔE];
    EK = Hmat[[1, 1]] / Smat[[1, 1]];
    err = EK - Eg;
    If[err > εB, τMIN = T];
      如果
    If[err < εB, τMAX = T];
      如果
  ), {j, 1, 30}];
T = (τMIN + τMAX) / 2.;
ΔEList = Table[ $\frac{2.}{d * 100} * j$ , {j, 1, 100}];
      表格

εKList = ΔEList;
Do[(
  Do循环
    ΔE = ΔEList[[j]];

    E0 = Eg;
    {Hmat, Smat} = funMatFilter[EE, Proψ, d, T, E0, ΔE];
    logη = -15;
    η = 10.logη;
    {EK, cn} = funDiagonalisation[Hmat + 2. * η * Ide, Smat + 2. * η * Ide];
    err = EK - Eg;

    εKList[[j]] = err;

```

```

    ), {j, 1, Length[ΔEList]};
    ΔE = ΔEList[[Position[εKList, Min[εKList]][[1, 1]]];
    AppendTo[plotsF, ListLogPlot[Transpose[{ΔEList, εKList}], PlotRange → Full]];
    E0 = Eg;
    {Hmat, Smat} = funMatFilter[EE, Proψ, d, T, E0, ΔE];
    {γList, εList} = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, 1., 1., logηList];
    γListF = γList;
    εListF = εList;
    εMin = Min[εList];
    If[εMin < 2. * εK, (
        γ = funInterpolation[εList, γList, 2. * εK];
    ), (
        γ = 0.;
    )];
    γF = γ;

    AppendTo[plots, ListLogLogPlot[
        {Transpose[{γListP, 0. * εListP + εK}], Transpose[{γListP, 0. * εListP + 2. * εK}]},
        Transpose[{γListP, εListP}], Transpose[{γListCP, εListCP}],
        Transpose[{γListGP, εListGP}], Transpose[{γListIP, εListIP}],
        Transpose[{γListITE, εListITE}], Transpose[{γListRTE, εListRTE}],
        Transpose[{γListF, εListF}]}, PlotRange → PR, Joined → True,
        PlotStyle → {Cyan, Brown, Red, Yellow, Blue, Green, Orange, Purple, Magenta}]];

    AppendTo[γ2εK, {γP, γCP, γGP, γIP, γITE, γRTE, γF}];

    Label[end];
    Print[{"d", d, εK, ToString[Now]}];
    {1, 1, Length[dList]}
    γ2εK = Transpose[γ2εK];
    {2, 0.0110074}
    {d, 2, 0.0110074, DateObject[{2022, 12, 10, 11, 49, 51.0045883}, Instant, Gregorian, 8.]}
    {d, 3, 0.00516331, DateObject[{2022, 12, 10, 11, 49, 51.6409754}, Instant, Gregorian, 8.]}
    {d, 4, 0.00192308, DateObject[{2022, 12, 10, 11, 49, 52.4794042}, Instant, Gregorian, 8.]}

```

```

{d, 5, 0.000602236, DateObject[{2022, 12, 10, 11, 49, 53.4082409}, Instant, Gregorian, 8.]}
{d, 6, 0.000229127, DateObject[{2022, 12, 10, 11, 49, 54.5744971}, Instant, Gregorian, 8.]}
{d, 7, 0.0000830542, DateObject[{2022, 12, 10, 11, 49, 56.0325346}, Instant, Gregorian, 8.]}
{d, 8, 0.0000279942, DateObject[{2022, 12, 10, 11, 49, 57.7839061}, Instant, Gregorian, 8.]}
{d, 9, 0.0000111243, DateObject[{2022, 12, 10, 11, 49, 59.8469732}, Instant, Gregorian, 8.]}
{d, 10, 4.36578×10-6, DateObject[{2022, 12, 10, 11, 50, 2.2534840}, Instant, Gregorian, 8.]}
{d, 11, 1.69961×10-6, DateObject[{2022, 12, 10, 11, 50, 5.0665694}, Instant, Gregorian, 8.]}
{d, 12, 6.82016×10-7, DateObject[{2022, 12, 10, 11, 50, 8.4258071}, Instant, Gregorian, 8.]}
{d, 13, 2.8792×10-7, DateObject[{2022, 12, 10, 11, 50, 12.6528769}, Instant, Gregorian, 8.]}
{d, 14, 1.61108×10-7, DateObject[{2022, 12, 10, 11, 50, 20.5740825}, Instant, Gregorian, 8.]}
{d, 15, 8.56406×10-8, DateObject[{2022, 12, 10, 11, 50, 30.2744736}, Instant, Gregorian, 8.]}
{d, 16, 4.4359×10-8, DateObject[{2022, 12, 10, 11, 50, 40.6851590}, Instant, Gregorian, 8.]}
{d, 17, 2.01177×10-8, DateObject[{2022, 12, 10, 11, 50, 52.9903669}, Instant, Gregorian, 8.]}
{d, 18, 1.18725×10-8, DateObject[{2022, 12, 10, 11, 51, 9.0967416}, Instant, Gregorian, 8.]}
{d, 19, 7.04174×10-9, DateObject[{2022, 12, 10, 11, 51, 37.5559958}, Instant, Gregorian, 8.]}
{d, 20, 3.74192×10-9, DateObject[{2022, 12, 10, 11, 52, 8.5206633}, Instant, Gregorian, 8.]}
{d, 21, 2.0856×10-9, DateObject[{2022, 12, 10, 11, 52, 41.5691291}, Instant, Gregorian, 8.]}
{d, 22, 1.37236×10-9, DateObject[{2022, 12, 10, 11, 53, 24.0047987}, Instant, Gregorian, 8.]}
{23, 8.43686×10-10}
{d, 23, 8.43686×10-10, DateObject[{2022, 12, 10, 11, 53, 24.0792446}, Instant, Gregorian, 8.]}
{24, 5.08429×10-10}
{d, 24, 5.08429×10-10, DateObject[{2022, 12, 10, 11, 53, 24.2023464}, Instant, Gregorian, 8.]}
{25, 3.31254×10-10}
{d, 25, 3.31254×10-10, DateObject[{2022, 12, 10, 11, 53, 24.3205044}, Instant, Gregorian, 8.]}
{26, 2.25562×10-10}
{d, 26, 2.25562×10-10, DateObject[{2022, 12, 10, 11, 53, 24.4426243}, Instant, Gregorian, 8.]}
{27, 1.31702×10-10}
{d, 27, 1.31702×10-10, DateObject[{2022, 12, 10, 11, 53, 24.5630324}, Instant, Gregorian, 8.]}
{28, 8.07604×10-11}
{d, 28, 8.07604×10-11, DateObject[{2022, 12, 10, 11, 53, 24.6967698}, Instant, Gregorian, 8.]}
{29, 5.79942×10-11}
{d, 29, 5.79942×10-11, DateObject[{2022, 12, 10, 11, 53, 24.8707436}, Instant, Gregorian, 8.]}
{30, 4.40111×10-11}
{d, 30, 4.40111×10-11, DateObject[{2022, 12, 10, 11, 53, 25.0564772}, Instant, Gregorian, 8.]}

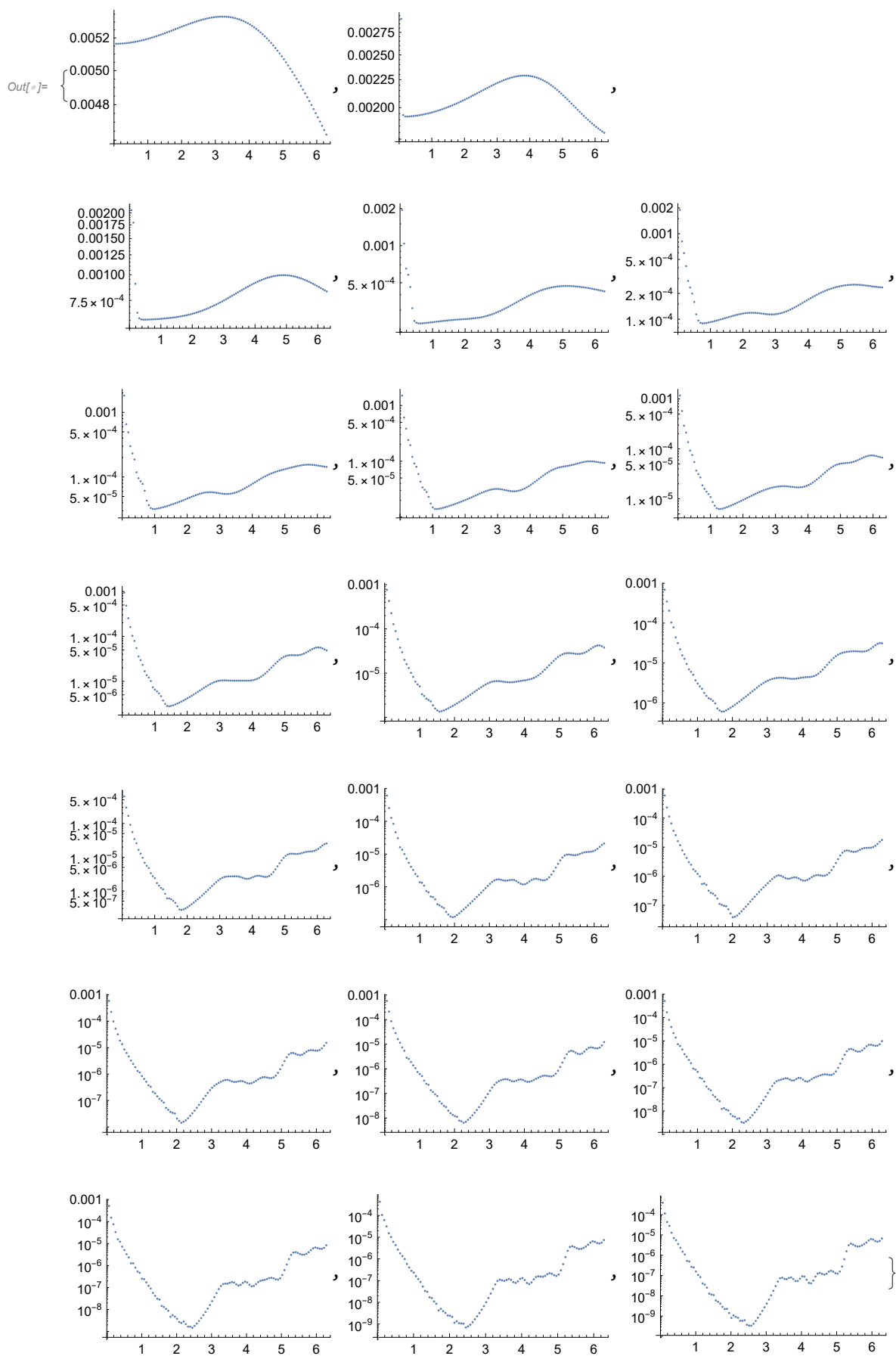
```

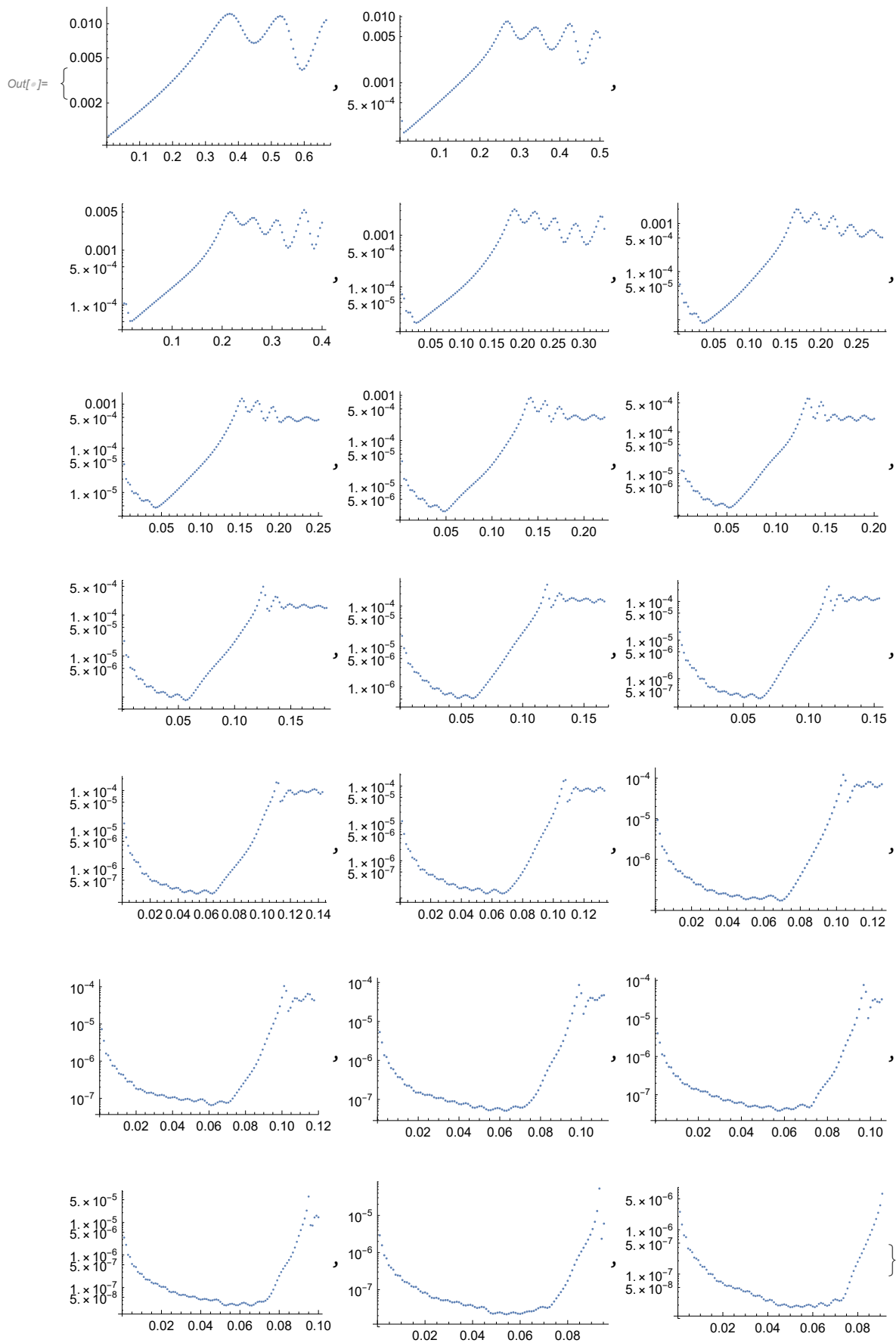


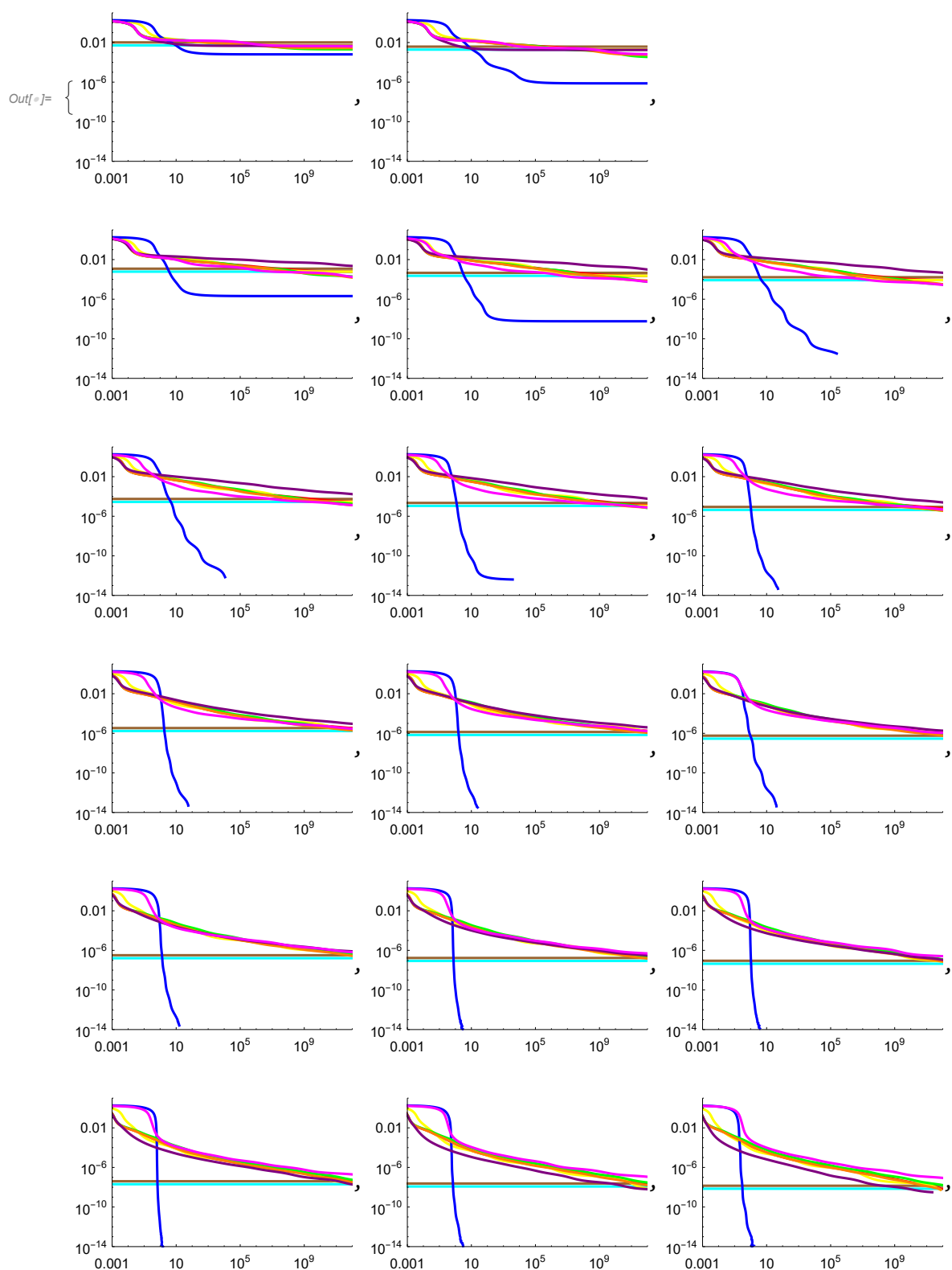
---

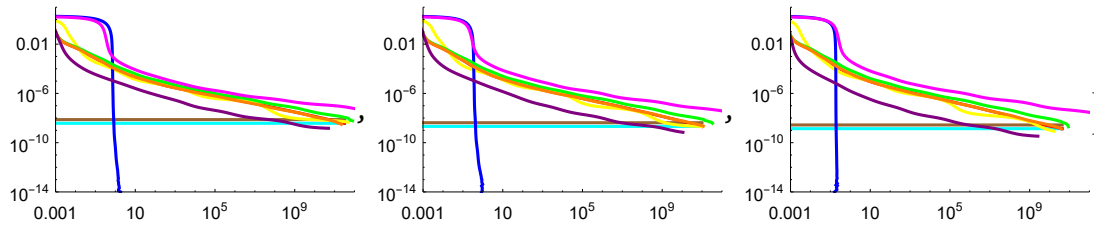
## Plots

```
In[ ]:= plotsRTE  
       plotsF  
       plots
```









## Data

```

In[ ]:= If[! StringMatchQ[GraphType, "Chain"] && ! StringMatchQ[GraphType, "Ladder"],
  |如果 |字符串匹配判定
  GraphType = "Random"];

In[ ]:= path =
  ToString[HamType] <> "-" <> ToString[GraphType] <> "-Nq=" <> ToString[Nq] <> ".dat";
  |转换为字符串 |转换为字符串 |转换为字符串
  CreateFile[path];
  |创建文件
  file = File[path];
  |文件位置的符号表示

In[ ]:= Data = {};
  AppendTo[Data, "γ2εK:"];
  |附加
  Data = Join[Data, γ2εK];
  |连接
  AppendTo[Data, ""];
  |附加
  Export[file, Data];
  |导出

```