```
In[ ]:= SetDirectory@NotebookDirectory[];
        设置目录          当前笔记本的目录
        Import["Qubits_package.m"];
        导入
        Import["ExactKrylov_package.m"];
        导入
```

Parameters

```
In[ ]:= Nq = 10;
        HamType = "Heisenberg";
        (*HamType="FermiHubbard";*)
        GraphType = "Chain";
        (*GraphType="Ladder";*)
        (*GraphType=ToString[2];*)
                    转换为字符串

In[ ]:= d = 5;
        logηList = Table[-0.1 * j, {j, -50, 150}];
                   表格
        Ide = IdentityMatrix[d];
              单位矩阵
        PR = {{1.*^-3, 1.*^12}, {1.*^-14, 1.*^1}};

In[ ]:= seed = RandomInteger[{1, 1000000}];
              伪随机整数
        seed = 123;
        SeedRandom[seed];
        随机种子
```

Model

```
In[ ]:= If[StringMatchQ[HamType, "Heisenberg"], (
        ··· 字符串匹配判定
            EL = funGraph[Nq, GraphType];
            Model = funHeisenberg[Nq, EL]
          )];
        If[StringMatchQ[HamType, "FermiHubbard"], (
        ··· 字符串匹配判定
            EL = funGraph[Nq / 2, GraphType];
            u = 1.;
            Model = funFermiHubbard[Nq, EL, u]
          )];
        Graph[EL]
        图
```

Out[ ]=


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

Spectrum

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

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[ ]= 1.58524
```

Reference state

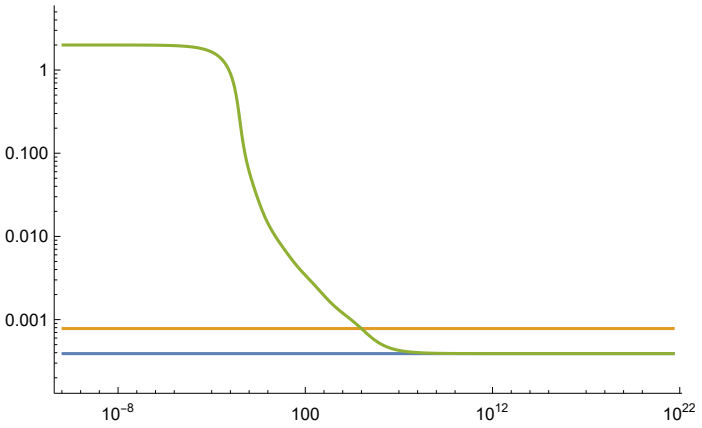
```
In[ ]:= If[StringMatchQ[HamType, "Heisenberg"], (
        ...  字符串匹配判定
            ψ = funPairwiseSinglet[Nq]
          )];
        If[StringMatchQ[HamType, "FermiHubbard"], (
        ...  字符串匹配判定
            ψ = funHartreeFock[Nq, EL]
          )];
        ψ = Conjugate[ES].ψ;
            共轭
  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

In[ ]:= pg = Proψ[[1]]
        ER = Total[Proψ * EE];
             总计
        ϵR = ER - Eg
Out[ ]= 0.682614

Out[ ]= 0.119312
```

Power

```
In[ ]:= E0 = Eg + 1.;
        {Hmat, Smat} = funMatPower[EE, Proψ, d, E0];
```

```
In[•]:= EB = Hmat[[d, d]] / Smat[[d, d]];
        ϵB = EB - Eg
Out[•]= 0.0072981

In[•]:= logη = -15;
        η = 10.^logη;
        {EK, cn} = funDiagonalisation[Hmat + 2. * η * Ide, Smat + 2. * η * Ide];
        ϵK = EK - Eg
Out[•]= 0.000390209

In[•]:= {γList, ϵList} = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, 1., 1., logηList];

In[•]:= ListLogLogPlot[{Transpose[{γList, 0. * ϵList + ϵK}],
        点集的双对数图  转置
          Transpose[{γList, 0. * ϵList + 2. * ϵK}], Transpose[{γList, ϵList}]},
          转置  转置
         PlotRange → {{Min[γList] / 3, Max[γList] * 3},
         绘制范围  最小值  最大值
           {Min[Join[ϵList, {ϵR, ϵK}]] / 3, Max[Join[ϵList, {ϵR, ϵK}]] * 3}}, Joined → True]
            …  连接  …  连接  连接点  真
```

Out[•]=

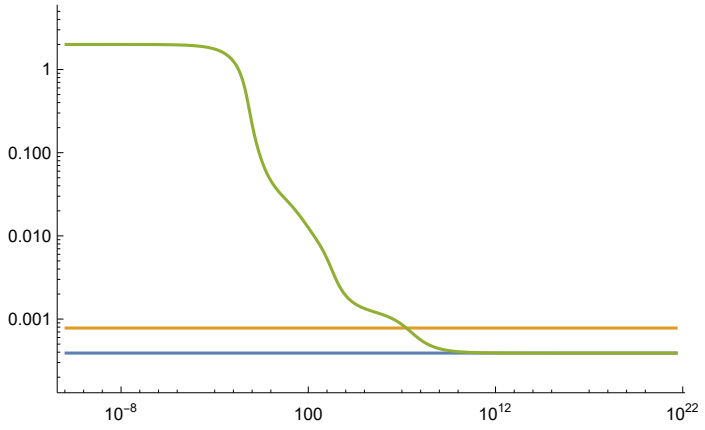



```
In[•]:= γListP = γList;
        ϵListP = ϵList;
        ϵMin = Min[ϵList];
               最小值
        If[ϵMin < 2. * ϵK, (
        如果
          γ = funInterpolation[ϵList, γList, 2. * ϵK];
         ), (
          γ = 0.;
         )]
        γP = γ
Out[•]= 96 610.8
```

Chebyshev polynomial

```
In[•]:= E0 = 0;
        {Hmat, Smat} = funMatChebyshev[EE, Proψ, d, htot, E0];
        {γList, ϵList} = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, 1., 1., logηList];

In[•]:= ListLogLogPlot[{Transpose[{γList, 0. * ϵList + ϵK}],
          点集的双对数图        转置
          Transpose[{γList, 0. * ϵList + 2. * ϵK}], Transpose[{γList, ϵList}]},
          转置                                      转置
         PlotRange → {{Min[γList] / 3, Max[γList] * 3},
          绘制范围        最小值              最大值
           {Min[Join[ϵList, {ϵR, ϵK}]] / 3, Max[Join[ϵList, {ϵR, ϵK}]] * 3}}, Joined → True]
            … 连接                          … 连接                          连接点  真
```

Out[•]=




```
In[•]:= γListCP = γList;
        ϵListCP = ϵList;
        ϵMin = Min[ϵList];
                最小值
        If[ϵMin < 2. * ϵK, (
        如果
          γ = funInterpolation[ϵList, γList, 2. * ϵK];
         ), (
          γ = 0.;
         )]
        γCP = γ
```

Out[•]= 1.71996×10^{7}

Gaussian-Power

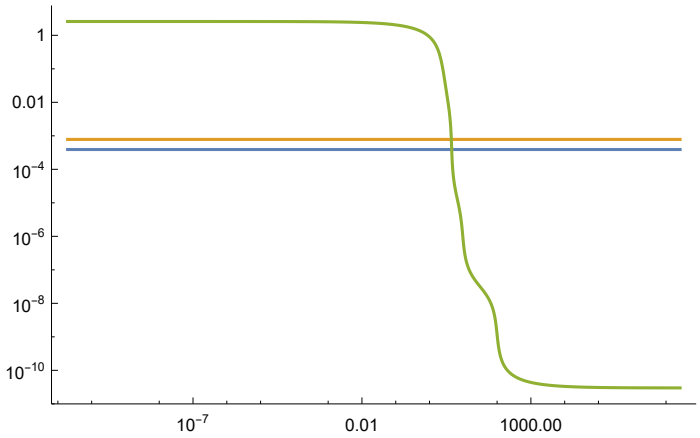
In[ ]:=
```
E0 = Eg;
τMIN = 0;
τMAX = 64;
Do[(
 Do循环
    τ = (τMIN + τMAX) / 2.;

    {Hmat, Smat} = funMatGaussianPower[EE, Proψ, 1, htot, τ, E0];
    EK = Hmat[[1, 1]] / Smat[[1, 1]];
    err = EK - Eg;

    If[err > ϵB, τMIN = τ];
    如果
    If[err < ϵB, τMAX = τ];
    如果
    (*Print[{err,τMIN,τMAX,ToString[Now]}];*)
      打印                    转换为…   此刻
  ), {j, 1, 30}];
τ = (τMIN + τMAX) / 2.
```

Out[ ]= 6.34638

In[ ]:=
```
E0 = Eg;
{Hmat, Smat} = funMatGaussianPower[EE, Proψ, d, htot, τ, E0];
{γList, ϵList} = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, htot, 1., logηList];
```

In[ ]:=
```
ListLogLogPlot[{Transpose[{γList, 0. * ϵList + ϵK}],
 点集的双对数图          转置
   Transpose[{γList, 0. * ϵList + 2. * ϵK}], Transpose[{γList, ϵList}]},
   转置                                      转置
 PlotRange → {{Min[γList] / 3, Max[γList] * 3},
 绘制范围          最小值          最大值
   {Min[Join[ϵList, {ϵR, ϵK}]] / 3, Max[Join[ϵList, {ϵR, ϵK}]] * 3}}, Joined → True]
    …  连接                    …  连接                  连接点  真
```

Out[ ]=

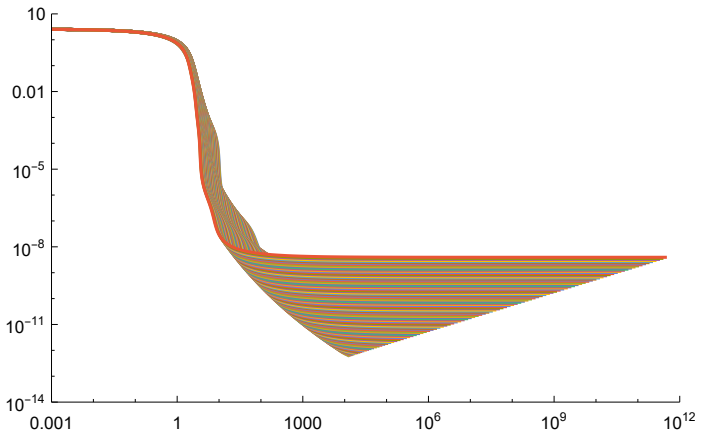


In[ ]:=
```
γListGP = γList;
ϵListGP = ϵList;
ϵMin = Min[ϵList];
       最小值
If[ϵMin < 2. * ϵK, (
如果
  γ = funInterpolation[ϵList, γList, 2. * ϵK];
 ), (
  γ = 0.;
 )]
γGP = γ
```

Out[ ]= 4.41987

In[ ]:=
```
δList = Table[0.002 * j, {j, -50, 50}];
        表格
δCurves = {};
Do[(
Do循环
  δ = δList[[j]];

  E0 = Eg + δ;
  {Hmat, Smat} = funMatGaussianPower[EE, Proψ, d, htot, τ, E0];
  {γList, ϵList} = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, htot, 1., logηList];

  AppendTo[δCurves, Transpose[{γList, ϵList}]]
  附加                转置
 ), {j, 1, Length[δList]}]
          长度
ListLogLogPlot[δCurves, PlotRange → PR, Joined → True]
点集的双对数图              绘制范围          连接点   真
```

Out[ ]=

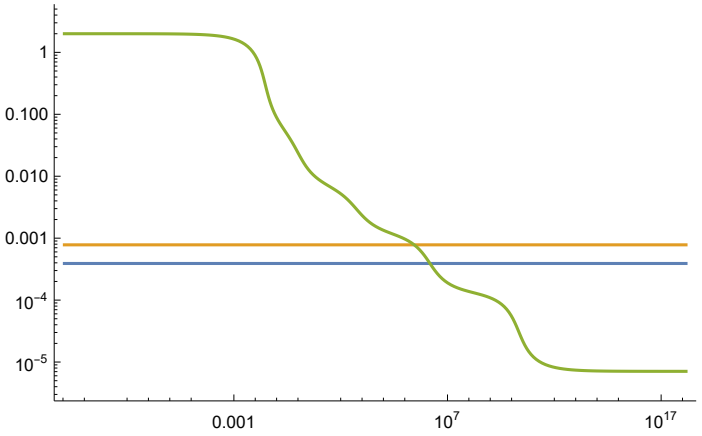


Inverse Power

In[ ]:=
```
E0 = Eg - 1.;
{Hmat, Smat} = funMatInversePower[EE, Proψ, d, E0];
{γList, ϵList} = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, 1., 1., logηList];
```

In[•]:=
```
ListLogLogPlot[{Transpose[{γList, 0. * ϵList + ϵK}],
 Transpose[{γList, 0. * ϵList + 2. * ϵK}], Transpose[{γList, ϵList}]},
 PlotRange → {{Min[γList] / 3, Max[γList] * 3},
   {Min[Join[ϵList, {ϵR, ϵK}]] / 3, Max[Join[ϵList, {ϵR, ϵK}]] * 3}}, Joined → True]
```
点集的双对数图 转置 转置 转置 绘制范围 最小值 最大值 … 连接 … 连接 连接点 真

Out[•]=



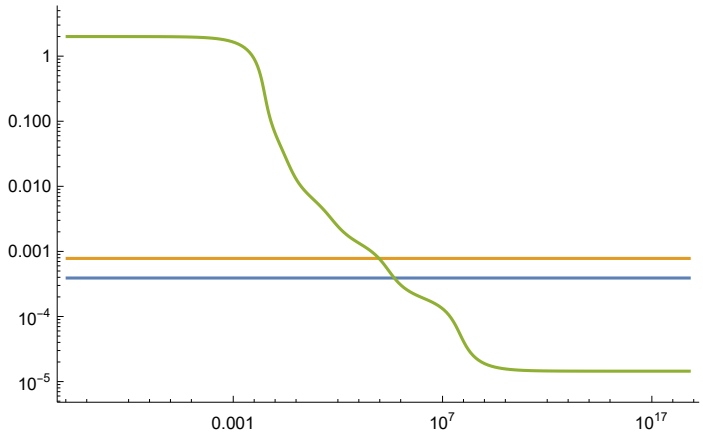
In[•]:=
```
γListIP = γList;
ϵListIP = ϵList;
ϵMin = Min[ϵList];
```
最小值
```
If[ϵMin < 2. * ϵK, (
```
如果
```
  γ = funInterpolation[ϵList, γList, 2. * ϵK];
 ), (
  γ = 0.;
 )]
γIP = γ
```

Out[•]= 276 085.

Imaginary-time evolution

```
In[•]:= 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
            (*Print[{err,τMIN,τMAX,ToString[Now]}];*)
              打印                    转换为…   此刻
          ), {j, 1, 30}];
        τ = (τMIN + τMAX) / 2.
Out[•]= 1.1768

In[•]:= E0 = Eg;
        {Hmat, Smat} = funMatITE[EE, Proψ, d, τ, E0];
        {γList, ϵList} = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, 1., 1., logηList];

In[•]:= ListLogLogPlot[{Transpose[{γList, 0. * ϵList + ϵK}],
        点集的双对数图          转置
          Transpose[{γList, 0. * ϵList + 2. * ϵK}], Transpose[{γList, ϵList}]},
          转置                                    转置
         PlotRange → {{Min[γList] / 3, Max[γList] * 3},
         绘制范围          最小值          最大值
           {Min[Join[ϵList, {ϵR, ϵK}]] / 3, Max[Join[ϵList, {ϵR, ϵK}]] * 3}}, Joined → True]
            …  连接                    …  连接                    连接点  真
```

Out[•]=



In[ ]:=
```
γListITE = γList;
ϵListITE = ϵList;
ϵMin = Min[ϵList];
       最小値
If[ϵMin < 2. * ϵK, (
如果
  γ = funInterpolation[ϵList, γList, 2. * ϵK];
 ), (
  γ = 0.;
 )]
γITE = γ
```

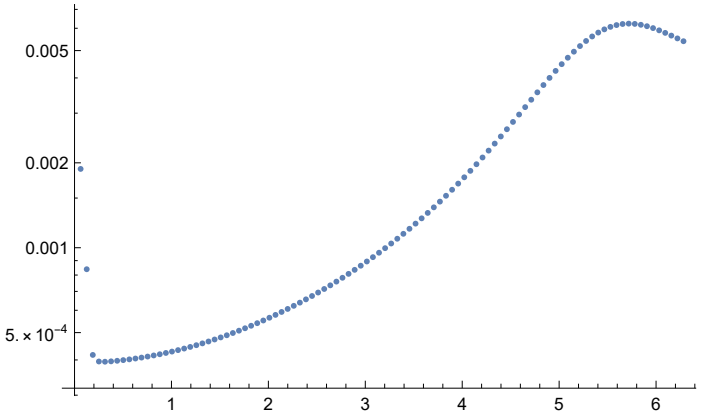
Out[ ]=
```
8964.89
```

Real-time evolution

```
In[ ]:= ΔtList = Table[2. * 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
        ListLogPlot[Transpose[{ΔtList, ϵKList}], PlotRange → Full]
        点集的对数图  转置                        绘制范围      全范围
```

```
Out[ ]= 0.314159
```

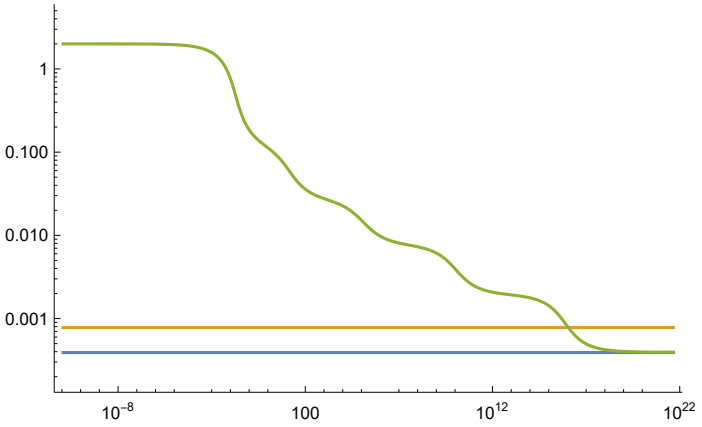
```
Out[ ]=
```




```
In[ ]:= E0 = Eg;
        {Hmat, Smat} = funMatRTE[EE, Proψ, d, Δt, E0];
        {γList, ϵList} = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, 1., 1., logηList];
```

In[ ]:=
```
ListLogLogPlot[{Transpose[{γList, 0. * ϵList + ϵK}],
 Transpose[{γList, 0. * ϵList + 2. * ϵK}], Transpose[{γList, ϵList}]},
 PlotRange → {{Min[γList] / 3, Max[γList] * 3},
  {Min[Join[ϵList, {ϵR, ϵK}]] / 3, Max[Join[ϵList, {ϵR, ϵK}]] * 3}}, Joined → True]
```
点集的双对数图 转置 转置 转置 绘制范围 最小值 最大值 … 连接 … 连接 连接点 真

Out[ ]=



In[ ]:=
```
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
```

Out[ ]= 1.11593×10^{16}

Filter

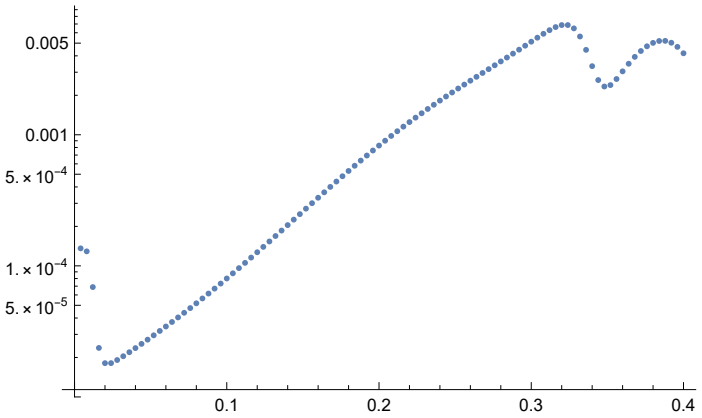
In[ ]:=
```
ΔE = 0.;
E0 = Eg;
τMIN = 0;
τMAX = 64;
Do[(
 Do循环
    T = (τMIN + τMAX) / 2.;

    {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
    (*Print[{err,τMIN,τMAX,ToString[Now]}];*)
      打印                    转换为…   此刻
  ), {j, 1, 30}];
T = (τMIN + τMAX) / 2.
```

Out[ ]= 11.2447

In[ ]:=
```
ΔEList = Table[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]}]
              长度
ΔE = ΔEList[[Position[ϵKList, Min[ϵKList]][[1, 1]]]]
              位置              最小值
ListLogPlot[Transpose[{ΔEList, ϵKList}], PlotRange → Full]
点集的对数图  转置                        绘制范围      全范围
```

Out[ ]=
```
0.02
```

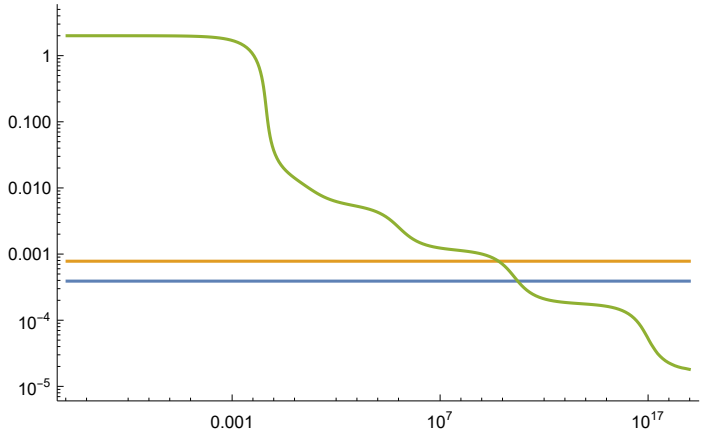
Out[ ]=



In[ ]:=
```
E0 = Eg;
{Hmat, Smat} = funMatFilter[EE, Proψ, d, T, E0, ΔE];
{γList, ϵList} = funGammaEpsilon[Eg, pg, Hmat, Smat, Ide, 1., 1., logηList];
```

In[•]:=
```
ListLogLogPlot[{Transpose[{γList, 0. * ϵList + ϵK}],
 Transpose[{γList, 0. * ϵList + 2. * ϵK}], Transpose[{γList, ϵList}]},
 PlotRange → {{Min[γList] / 3, Max[γList] * 3},
   {Min[Join[ϵList, {ϵR, ϵK}]] / 3, Max[Join[ϵList, {ϵR, ϵK}]] * 3}}, Joined → True]
```
点集的双对数图 转置 转置 转置 绘制范围 最小值 最大值 … 连接 … 连接 连接点 真

Out[•]=



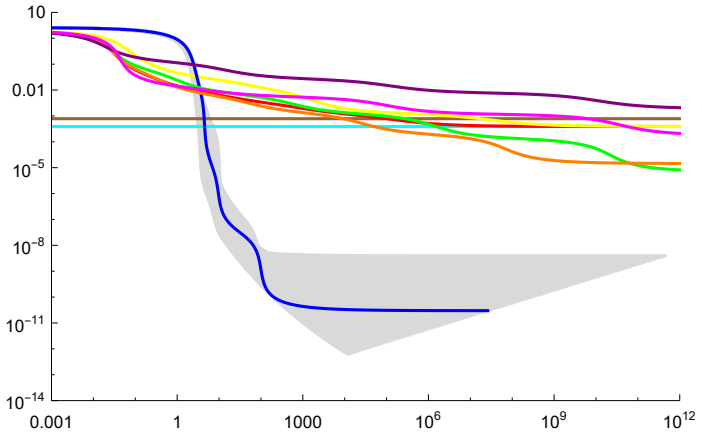
In[•]:=
```
γListF = γList;
ϵListF = ϵList;
ϵMin = Min[ϵList];
```
最小值
```
If[ϵMin < 2. * ϵK, (
```
如果
```
  γ = funInterpolation[ϵList, γList, 2. * ϵK];
 ), (
  γ = 0.;
 )]
γF = γ
```

Out[•]= 6.4194×10^{9}

Plot

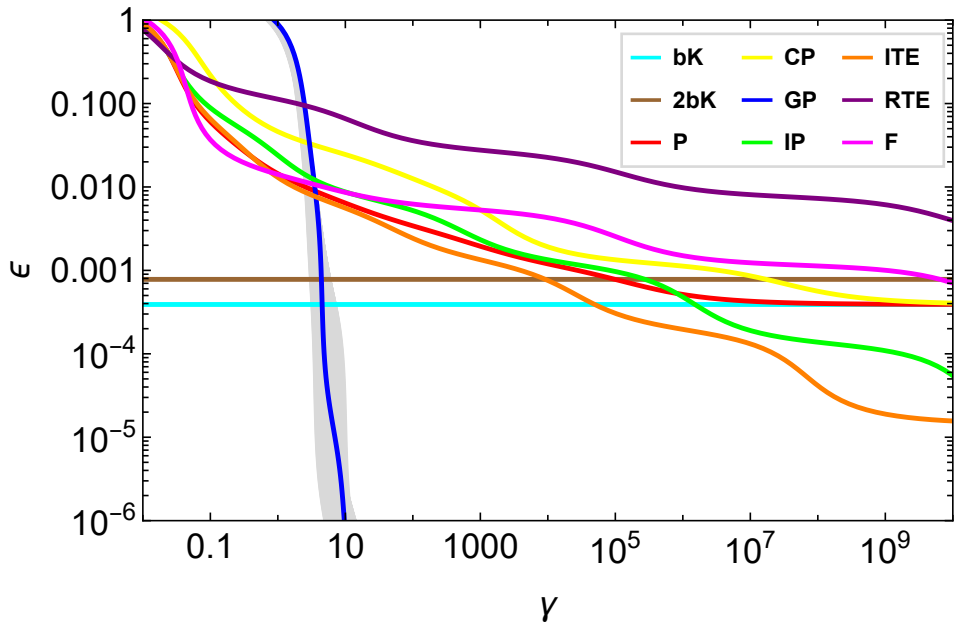
In[ ]:=
```
ListLogLogPlot[Join[δCurves, {Transpose[{γListP, 0. * ϵListP + ϵK}],
  Transpose[{γListP, 0. * ϵListP + 2. * ϵK}], Transpose[{γListP, ϵListP}],
  Transpose[{γListCP, ϵListCP}], Transpose[{γListGP, ϵListGP}],
  Transpose[{γListIP, ϵListIP}], Transpose[{γListITE, ϵListITE}],
  Transpose[{γListRTE, ϵListRTE}], Transpose[{γListF, ϵListF}]}],
 PlotRange → PR, Joined → True, PlotStyle → Join[Table[LightGray, Length[δCurves]],
  {Cyan, Brown, Red, Yellow, Blue, Green, Orange, Purple, Magenta}]]
```
点集的双对数图 连接 转置 转置 转置 转置 转置 转置 转置 转置 绘制范围 连接点 真 绘图样式 连接 表格 浅灰色 长度 蓝绿色 棕色 红色 黄色 蓝色 绿色 橙色 紫色 品红色

Out[ ]=



In[ ]:=
```
PR = {{1.*^-2, 1.*^10}, {1.*^-6, 1.*^0}};
plot1 = ListLogLogPlot[δCurves, PlotRange → PR,
        点集的双对数图          绘制范围
    Joined → True, PlotStyle → Table[LightGray, Length[δCurves]],
    连接点   真    绘图样式      表格   浅灰色      长度
    Frame → True, FrameStyle → Directive[Black, Thickness[0.002]],
    边框    真    边框样式      指令       黑色   粗细
    FrameTicksStyle → Directive[Black, Thickness[0.002]], FrameLabel → {"γ", "ϵ"}];
    边框刻度样式          指令       黑色   粗细                  边框标签
plot2 = ListLogLogPlot[{Transpose[{γListP, 0. * ϵListP + ϵK}],
        点集的双对数图      转置
     Transpose[{γListP, 0. * ϵListP + 2. * ϵK}], Transpose[{γListP, ϵListP}],
     转置                                        转置
     Transpose[{γListCP, ϵListCP}], Transpose[{γListGP, ϵListGP}],
     转置                           转置
     Transpose[{γListIP, ϵListIP}], Transpose[{γListITE, ϵListITE}],
     转置                           转置
     Transpose[{γListRTE, ϵListRTE}], Transpose[{γListF, ϵListF}]},
     转置                             转置
    PlotRange → PR, Joined → True, PlotStyle → {{Thickness[0.006], Cyan},
    绘制范围        连接点   真    绘图样式      粗细                蓝绿色
      {Thickness[0.006], Brown}, {Thickness[0.006], Red}, {Thickness[0.006], Yellow},
       粗细                棕色      粗细                红色      粗细                黄色
      {Thickness[0.006], Blue}, {Thickness[0.006], Green}, {Thickness[0.006], Orange},
       粗细                蓝色      粗细                绿色      粗细                橙色
      {Thickness[0.006], Purple}, {Thickness[0.006], Magenta}},
       粗细                紫色      粗细                品红色
    Frame → True, FrameStyle → Directive[Black, Thickness[0.002]],
    边框    真    边框样式      指令       黑色   粗细
    FrameTicksStyle → Directive[Black, Thickness[0.002]],
    边框刻度样式          指令       黑色   粗细
    FrameLabel → {"γ", "ϵ"}, PlotLegends →
    边框标签                 绘图的图例
     Placed[LineLegend[{"bK", "2bK", "P", "CP", "GP", "IP", "ITE", "RTE", "F"},
     放置    线的图例
       LegendFunction → (Framed[#, FrameStyle → LightGray] &), LegendMarkerSize →
       图例函数              加边框    边框样式      浅灰色          图例标记尺寸
        {16, 8}, LabelStyle → {Black, Bold, FontSize → 12, FontFamily → "Arial"},
                 标签样式      黑色   粗体   字体大小      字体系列
       LegendMargins → 0, LegendLayout → {"Column", 3}], {0.79, 0.84}]];
       图例边幅           图例布局          列
Show[plot1, plot2, LabelStyle → {FontSize → 18, FontFamily → "Arial"}, ImageSize → 500]
显示    标签样式      字体大小      字体系列                图像尺寸
```

Out[•]=



Data

In[•]:=
```
If[! StringMatchQ[GraphType, "Chain"] && ! StringMatchQ[GraphType, "Ladder"],
  GraphType = "Random"];
```

In[•]:=
```
path = ToString[HamType] <> "-" <> ToString[GraphType] <>
   "-Nq=" <> ToString[Nq] <> "-d=" <> ToString[d] <> ".dat";
CreateFile[path];
file = File[path];
```

In[•]:=
```
Data = {};
AppendTo[Data, "ϵR:"];
AppendTo[Data, ϵR];
AppendTo[Data, ""];
AppendTo[Data, "ϵB:"];
AppendTo[Data, ϵB];
AppendTo[Data, ""];
AppendTo[Data, "ϵK:"];
AppendTo[Data, ϵK];
AppendTo[Data, ""];
```

```
AppendTo[Data, "γP:"];
附加
AppendTo[Data, γP];
附加
AppendTo[Data, ""];
附加
AppendTo[Data, "γCP:"];
附加
AppendTo[Data, γCP];
附加
AppendTo[Data, ""];
附加
AppendTo[Data, "γGP:"];
附加
AppendTo[Data, γGP];
附加
AppendTo[Data, ""];
附加
AppendTo[Data, "γIP:"];
附加
AppendTo[Data, γIP];
附加
AppendTo[Data, ""];
附加
AppendTo[Data, "γITE:"];
附加
AppendTo[Data, γITE];
附加
AppendTo[Data, ""];
附加
AppendTo[Data, "γRTE:"];
附加
AppendTo[Data, γRTE];
附加
AppendTo[Data, ""];
附加
AppendTo[Data, "γF:"];
附加
AppendTo[Data, γF];
附加
AppendTo[Data, ""];
附加
AppendTo[Data, "seed:"];
附加
AppendTo[Data, seed];
附加
AppendTo[Data, ""];
附加
Export[file, Data];
导出
```