

# Su-Schrieffer-Heeger model

---

## Hamiltonian

```
In[*]:=  $\sigma_0 = \{\{1, 0\}, \{0, 1\}\};$   
         $\sigma_x = \{\{0, 1\}, \{1, 0\}\};$   
         $\sigma_y = \{\{0, -i\}, \{i, 0\}\};$   
         $\sigma_z = \{\{1, 0\}, \{0, -1\}\};$   
         $\sigma_p = \{\{0, 1\}, \{0, 0\}\};$   
         $\sigma_m = \{\{0, 0\}, \{1, 0\}\};$   
        ChainLength = 20;  
        CMat1 = SparseArray[[疎な配列] {Band[{1, 1]} → 1}, [帯対角行列] {ChainLength, ChainLength}];  
        CMat2 = SparseArray[[疎な配列] {Band[{1, 2]} → 1}, [帯対角行列] {ChainLength, ChainLength}];  
        CMat3 = SparseArray[[疎な配列] {Band[{2, 1]} → 1}, [帯対角行列] {ChainLength, ChainLength}];  
        t = 1.0;  
        v = 0.2;  
        HamSSH = KroneckerProduct[CMat1, v *  $\sigma_x$ ] +  
                [クロネッカー積]  
                KroneckerProduct[CMat2, t *  $\sigma_m$ ] + KroneckerProduct[CMat3, t *  $\sigma_p$ ];  
                [クロネッカー積] [クロネッカー積]
```



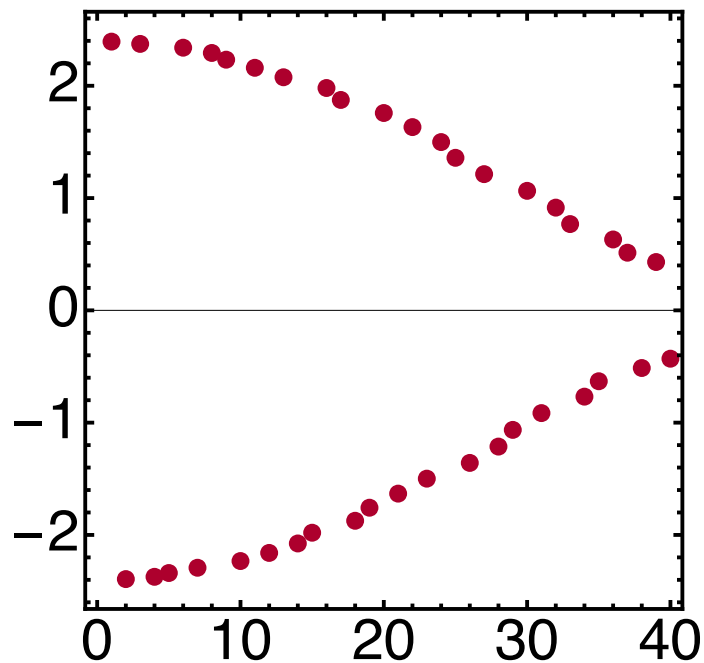
## Spectrum & eigenstates

```
In[*]:= SetOptions[ListPlot,
  [オプション… [リストプロット
    Frame → True, Joined → False, FrameStyle → Thick, FrameTicksStyle → 30,
    [枠 [真 [点の結合 [偽 [枠のスタイル [太い [枠目盛のスタイル
    BaseStyle → {FontFamily → "Helvetica", FontSize → 20}, PlotStyle →
    [ベーススタイル [フォントファミリー [フォントサイズ [プロットスタイル
    {{RGBColor[173 / 255, 0, 45 / 255], PointSize[.03]}}, AspectRatio → 1.0];
    [RGBカラー [ポイントサイズ [縦横比
```

### (1) trivial phase ( $t=1.0, v=1.4$ )

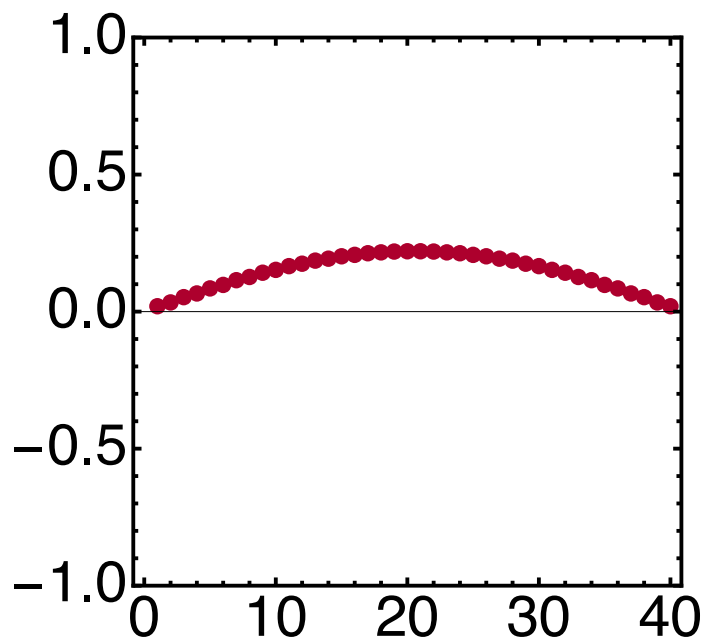
```
In[*]:= ListPlot[eigen[[1]]
  [リストプロット
```

Out[\*]=



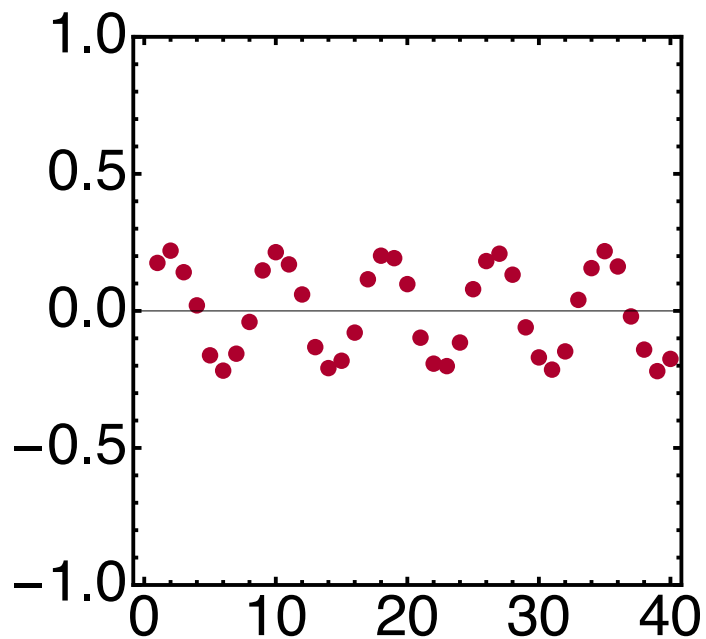
```
In[*]:= ListPlot[eigen[[2, 1]], PlotRange → {-1, 1}]  
[リストプロット [プロット範囲]
```

Out[\*]=



```
In[*]:= ListPlot[eigen[[2, ChainLength]], PlotRange → {-1, 1}]  
[リストプロット [プロット範囲]
```

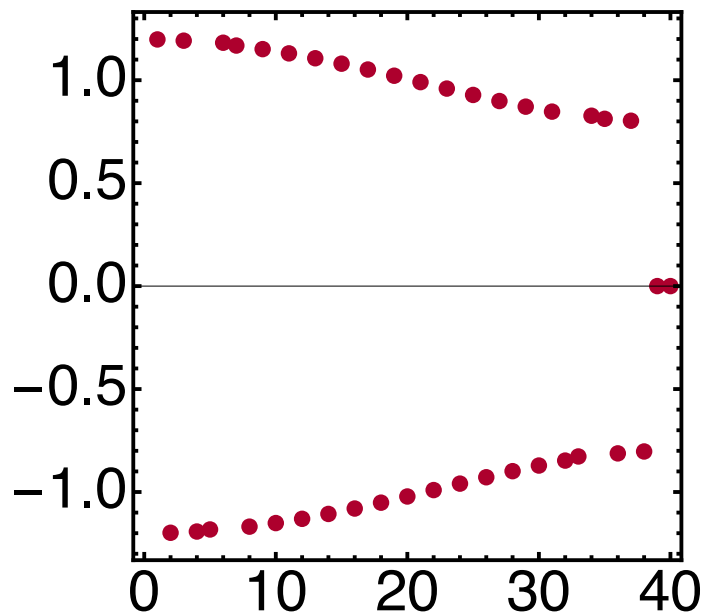
Out[\*]=



(2) topological phase ( $t=1.0, v=0.2$ )

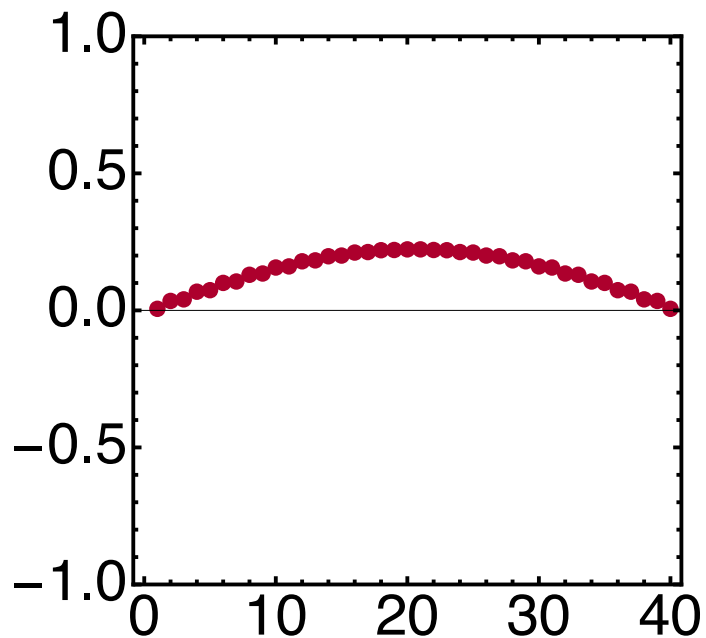
```
In[*]:= ListPlot[eigen[[1]]]
      ↳リストプロット
```

```
Out[*]=
```



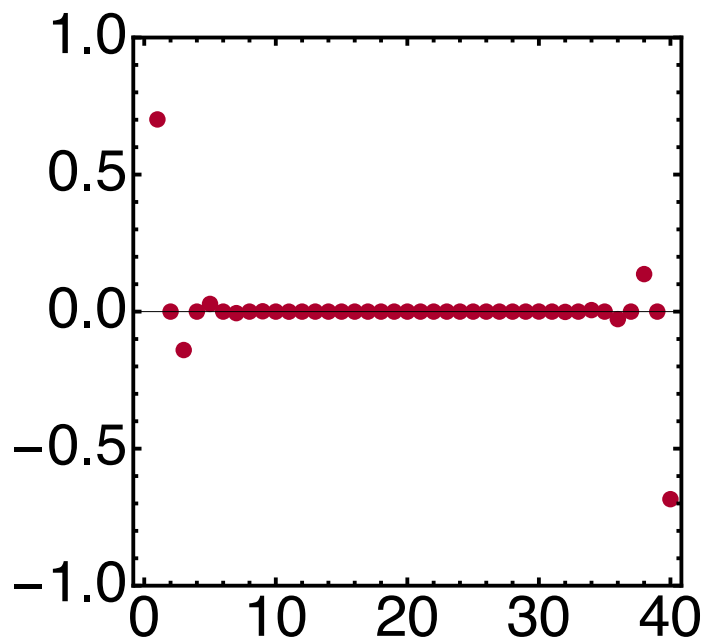
```
In[*]:= ListPlot[eigen[[2, 1]], PlotRange → {-1, 1}]
      ↳リストプロット      ↳プロット範囲
```

```
Out[*]=
```



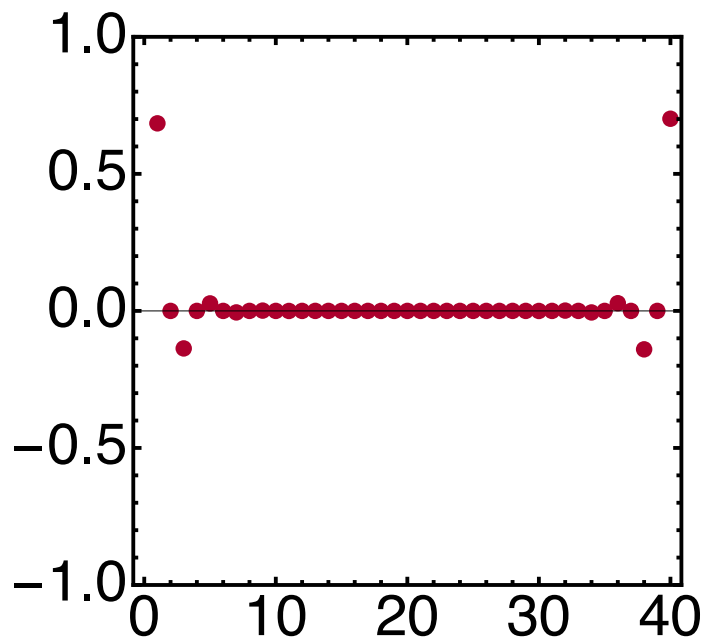
```
In[*]:= ListPlot[eigen[[2, 2 ChainLength - 1]], PlotRange → {-1, 1}]  
[リストプロット [プロット範囲]
```

Out[\*]=



```
In[*]:= ListPlot[eigen[[2, 2 ChainLength]], PlotRange → {-1, 1}]  
[リストプロット [プロット範囲]
```

Out[\*]=



## Effect of disorder

```

In[*]:= ChainLength = 20;
t0 = 1.0;
tw = 0.4;
tList = Table[t0 + tw * RandomReal[{-0.5, 0.5}], {n, 1, ChainLength - 1}];
           ↳リストを作成      ↳実数乱数
v0 = 0.2;
vw = 0.4;
vList = Table[v0 + vw * RandomReal[{-0.5, 0.5}], {n, 1, ChainLength}];
           ↳リストを作成      ↳実数乱数
HamSSHdis = Block[{thop, vhop},
                  ↳ブロック
    thop =
      SparseArray[{Band[{2, 3}, {2 ChainLength - 2, 2 ChainLength - 1}, 2] → tList},
                  ↳疎な配列      ↳帯対角行列
                  {2 ChainLength, 2 ChainLength}];
    vhop = SparseArray[{Band[{1, 2}, {2 ChainLength - 1, 2 ChainLength}, 2] → vList},
                  ↳疎な配列      ↳帯対角行列
                  {2 ChainLength, 2 ChainLength}];
    thop + Transpose[thop] + vhop + Transpose[vhop]
           ↳転置              ↳転置
];

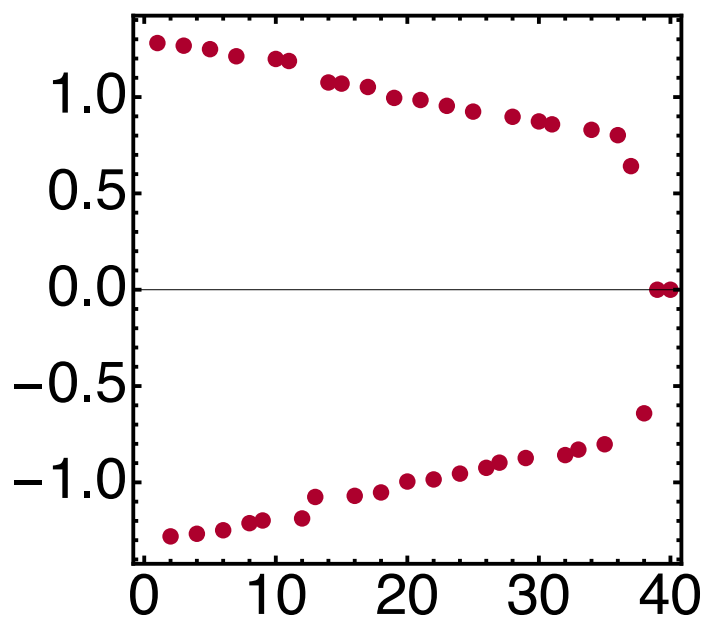
```





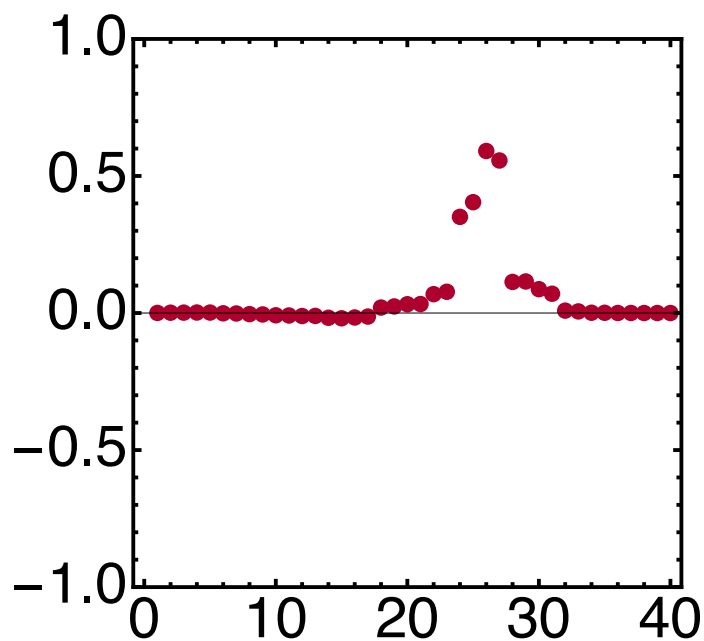
```
In[*]:= ListPlot[eigendis[[1]]  
[リストプロット
```

Out[\*]=



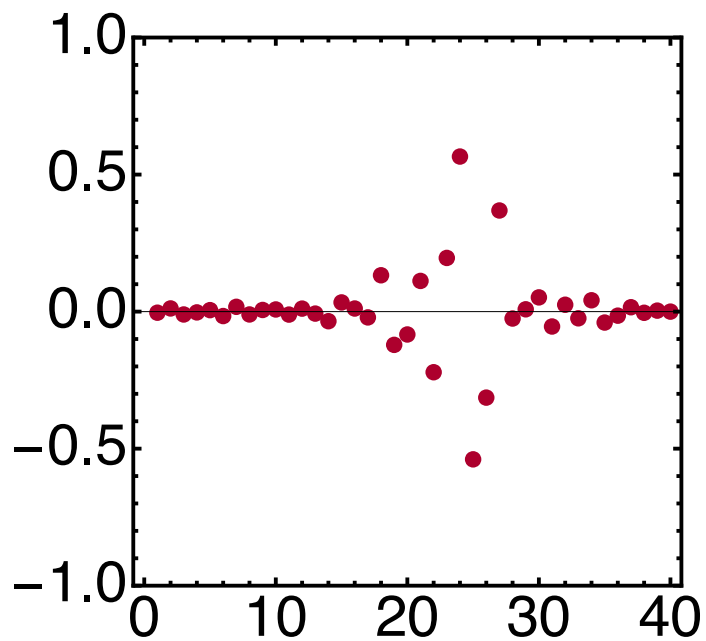
```
In[*]:= ListPlot[eigendis[[2, 5]], PlotRange → {-1, 1}]  
[リストプロット [プロット範囲
```

Out[\*]=



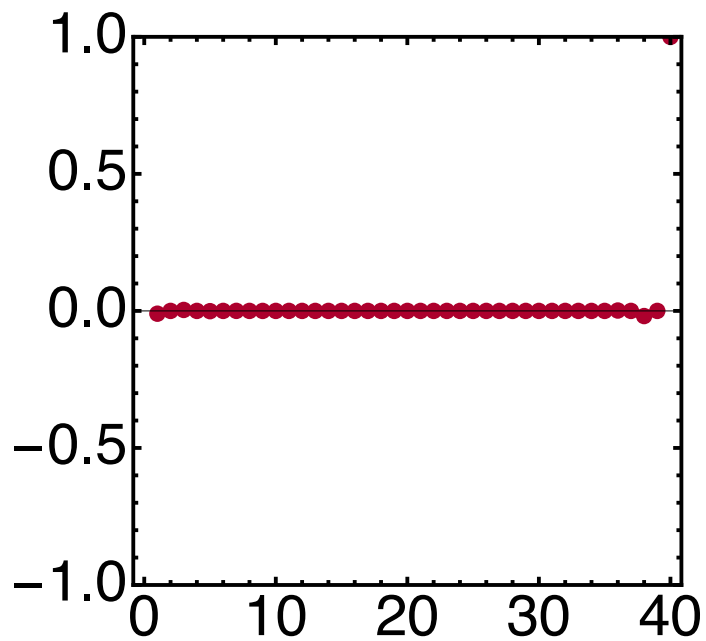
```
In[*]:= ListPlot[eigendis[[2, ChainLength]], PlotRange → {-1, 1}]
          ↳リストプロット                               ↳プロット範囲
```

Out[\*]=



```
In[*]:= ListPlot[eigendis[[2, 2 ChainLength - 1]], PlotRange → {-1, 1}]
          ↳リストプロット                               ↳プロット範囲
```

Out[\*]=



```
In[*]:= ListPlot[eigendis[[2, 2 ChainLength]], PlotRange → {-1, 1}]  
[リストプロット [プロット範囲]
```

Out[\*]=

