

学习记录

Bo Zhao (赵博)
(Dated: June 21, 2024)

蓝色是以后会用的/需要注意的, 红色是待完成项。

2024/06/17

手上压的学习任务有点多, 之后先专心过基础知识, 之后有多余时间把之前看的几篇文章好好整理一下, 之后顺着看一下Wave-Function Tomography of Topological Dimer Chains with Long-Range Couplings, 这篇应该和之前看的声子晶体Large winding number比较相似, 顺便可以尝试一下用comsol复现。都了结之后做comsol官网声子晶体的案例(包含色散关系), 尝试复现Large winding number仿真部分。看一下Ref. [1] (Realization of fractional quantum Hall state with interacting photons)。

重复comsol案例: 光子晶体和光子晶体带隙分析。注意光子晶体案例解出的电场是有虚部的, 所以最后的线图直接用默认的normE和二维图对不上。comsol默认Ez显示实部解, 因此ewfd.normE对应的不是 $\sqrt{\text{ewfd.Ex}^2 + \text{ewfd.E}^2 + \text{ewfd.Ez}^2}$, 而是 $\sqrt{\text{real}(\text{ewfd.Ex})^2 + \text{imag}(\text{ewfd.Ex})^2 + \text{real}(\text{ewfd.Ey})^2 + \text{imag}(\text{ewfd.Ey})^2 + \text{real}(\text{ewfd.Ez})^2 + \text{imag}(\text{ewfd.Ez})^2}$ 。

2024/06/15 & 2024/06/16

休息。

2024/06/20 2024/06/21

复现宇宸师兄在群里发的文章(基于紧束缚模型求解螺旋波导阵列)Ref. [2]。Ref. [2]是Nature文章Photonic Floquet topological insulators拓扑理论部分的学习笔记 [3]。Ref. [3]的亮点在于: 1. 光波段实现(基于打破时间反演)。2. 新的拓扑绝缘体实现方式。3. 无磁, 更易小型化集成化(作者没说, 我认为的)。螺旋波导电磁理论: Ref. [4]。拓扑绝缘体的哈密顿量是这样实现的: 构建模型中的第三维度z映射时间。变换到旋转框架, 波导螺旋周期需足够小, 以实现z方向的绝热性(不懂, 过完量力的绝热近似再说)。根据紧束缚方法, 得到电场的动力学演化(看完紧束缚模型再说)。由动力学演化, 用海森堡方程, 或者量子郎之万方程反向变回去(文章上说用Peierls substitution, 不过我大致估算了一下, 海森堡方程就能得到相同结果。)就可以得到Hamiltonian。

This week

2024/06/14

看完Griffiths量子力学第四章。看高量课程, 补画了个别图片。

2024/06/13

看Griffiths量子力学第四章。听了严以京关于开放系统中量子力学的报告。

2024/06/12

看Griffiths量子力学第四章。

2024/06/19

2024/06/11

感觉无心学习, 稍微提前回去休息了。18, 19号休息周六日补上。

完成Griffiths量子力学第三章, 读部分第四章。

2024/06/18

2024/06/10

17号弄得太晚, 18号一直很困就回去休息了。

下午: 读Griffiths量子力学§3.1和§3.2并完成重点*习题。

2024/06/08 & 2024/06/09

从8号中午看到9号早上。

总结：完成习题1.3，浏览复现文章Ref. [5] (Large Winding number) Winding number图片(其他图片套路和之前基本一致，就没画)。关于comsol仿真部分复现失败，对仿真掌握的知识太少，学过后可以回来尝试复现一下。

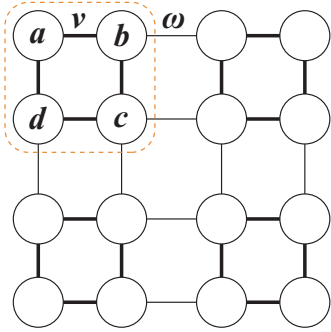


FIG. 1. 二维SSH模型。

习题1.3 A possible generalization to two dimensions- Consider a two dimensional generalization of the SSH model. Take parallel copies of the SSH chain and couple them without breaking chiral symmetry. What will happen with the edge states?

答：习题1.3的答案有人在2017年发过prl (该书是2016年出版的) [6]。设二维SSH模型cell中子晶格顺时针标号为abcd, x(y)方向cell标号为i(j),如FIG. 1所示, Hamiltonian为:

$$\begin{aligned} \hat{H} = & v \sum_{i,j}^{N_i, N_j} (\hat{a}_{i,j}^\dagger \hat{b}_{i,j} + \hat{b}_{i,j}^\dagger \hat{c}_{i,j} + \hat{c}_{i,j}^\dagger \hat{d}_{i,j} + \hat{d}_{i,j}^\dagger \hat{a}_{i,j}) \\ & + w \sum_{i,j}^{N_{i-1}, N_j} (\hat{b}_{i,j}^\dagger \hat{a}_{i+1,j} + \hat{c}_{i,j}^\dagger \hat{d}_{i+1,j}) \\ & + w \sum_{i,j}^{N_i, N_{j-1}} (\hat{d}_{i,j}^\dagger \hat{a}_{i,j+1} + \hat{c}_{i,j}^\dagger \hat{b}_{i,j+1}) + \text{H.c.} \end{aligned}$$

动量空间的Hamiltonian:

$$\hat{H}(k) = \begin{pmatrix} 0 & v + we^{-ik_x} & 0 & v + we^{-ik_y} \\ v + we^{ik_x} & 0 & v + we^{-ik_y} & 0 \\ 0 & v + we^{ik_y} & 0 & v + we^{-ik_x} \\ v + we^{ik_y} & 0 & v + we^{ik_x} & 0 \end{pmatrix}.$$

先画个色散关系, 如FIG. 2所示。

按Ref. [7]第一章的思路继续写不出了, 因为不知道如何把 $\hat{H}(k)$ 写成实空间的 \hat{H} 来画能谱图以及波函数分布。

注意到Ref. [6]限制一个方向(y方向)无边界, 另一个方向有界, 利用该思想可以得到部分实空间

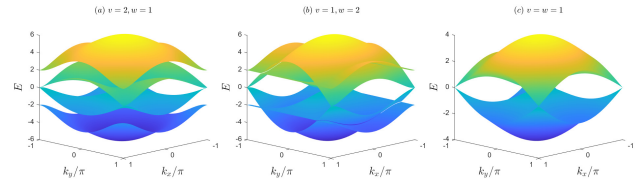


FIG. 2. 2D SSH模型色散关系。

的Hamiltonian:

$$\hat{H}(k) = \begin{pmatrix} A & B & 0 & \cdots & 0 \\ B^\dagger & A & B & \cdots & 0 \\ 0 & B^\dagger & A & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & B \\ 0 & 0 & 0 & \cdots & A \end{pmatrix}.$$

即主对角线是A矩阵, 超对角线是B矩阵, 次对角线是 B^\dagger 矩阵。其中

$$A = \begin{pmatrix} 0 & v & 0 & v + we^{-ik_y} \\ v & 0 & v + we^{-ik_y} & 0 \\ 0 & v + we^{ik_y} & 0 & v \\ v + we^{ik_y} & 0 & v & 0 \end{pmatrix},$$

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ w & 0 & 0 & 0 \\ 0 & 0 & 0 & w \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

可以据此画出能谱图, 如FIG. 3所示。原文似乎有些问题, 跟据我所绘的图, 蓝色能量对应的波函数并非边界态。

关于参考文献Ref. [6], 过完基础知识之后回来仔细看一看, 了解物理图像。

下图是Ref. [5]的复现图, 相比于我构想的winding number=2的模型, 该模型动量空间的哈密顿量e指数符号不同, 因此体现在实空间中的耦合项相差甚远。特别是, 该模型还考虑的B模式与上一cell中A模式的耦合。但是在代码上几乎相同, 此处不给出复现代码(用之前的调调参数, Hamiltonian改改符号即可)。Ref. [5]中Appendix A中动量空间Hamiltonian系数v1对应e指数少了个负号(复现图片会出现差异)。

能谱图如FIG. 5所示。但是看到文中标明4个边界态, 同时就尝试在我的模型Winding number=2情况看一下, 也出现了4个边界态, 如FIG. 6所示。很神奇, 后面继续学的时候关注一下相关知识。

2024/06/07

来了办公室但是超级困, 周六日再来学吧。

2024/06/06

上午: 睡觉(前两天熬的太晚)。

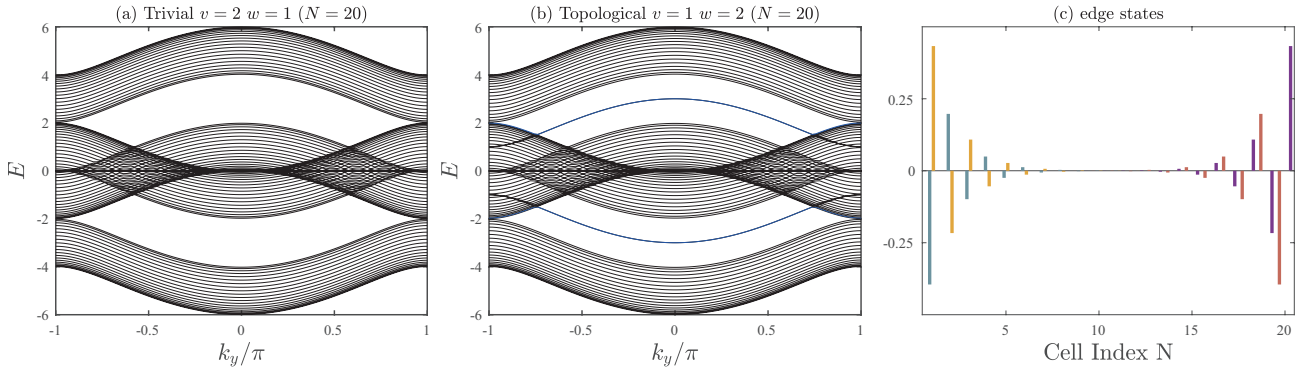


FIG. 3. 2D SSH模型能带(a)(b)以及边界态波函数分布(c)。

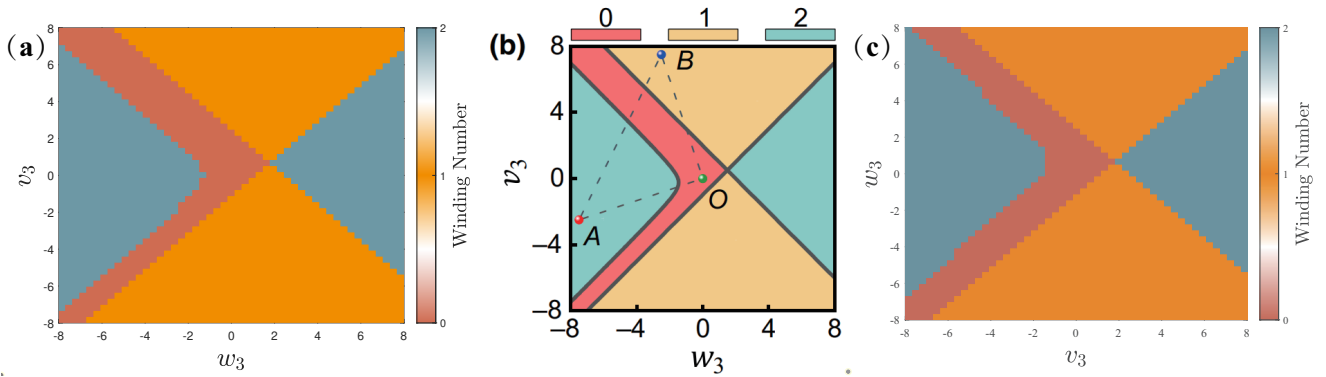


FIG. 4. Winding Number of Ref. [5], (a) 使用修正后Hamiltonian。 (b) 原图。 (c) 使用原文Hamiltonian。

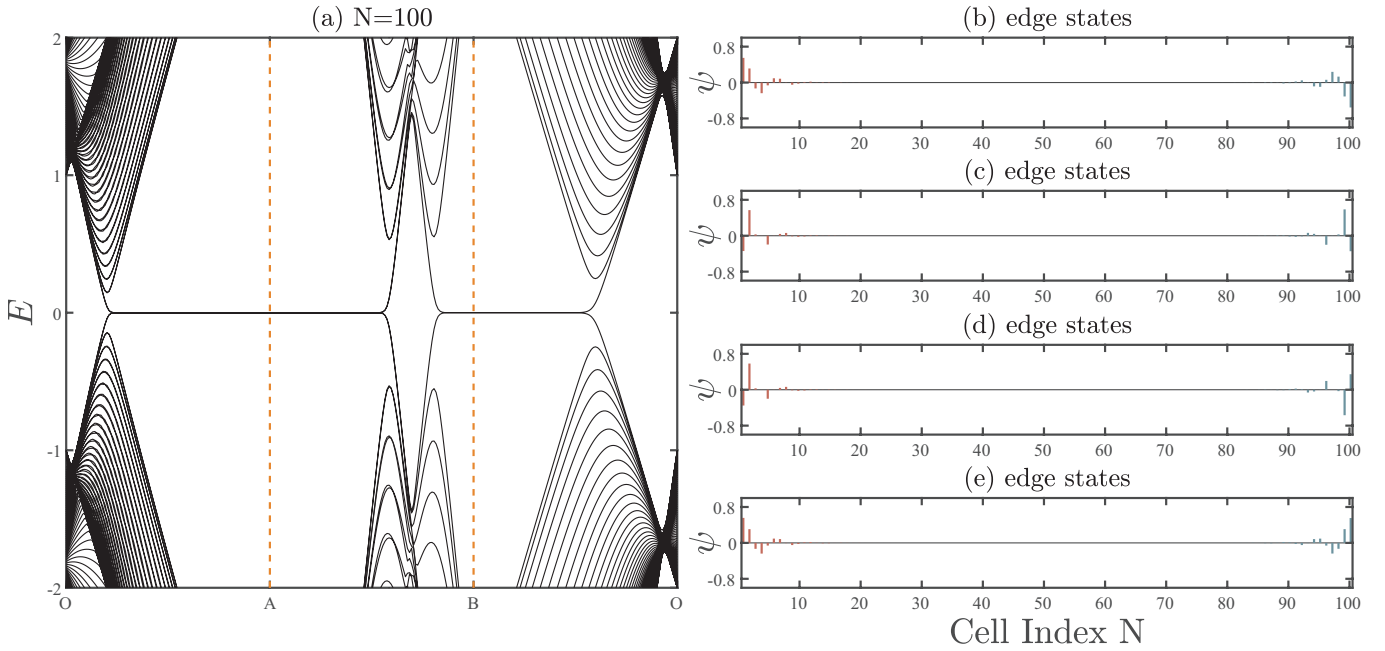


FIG. 5. Winding Number=2存在四个边界态,原文复现 (原文 $N=120$,但结果相同), 2024/06/14补画

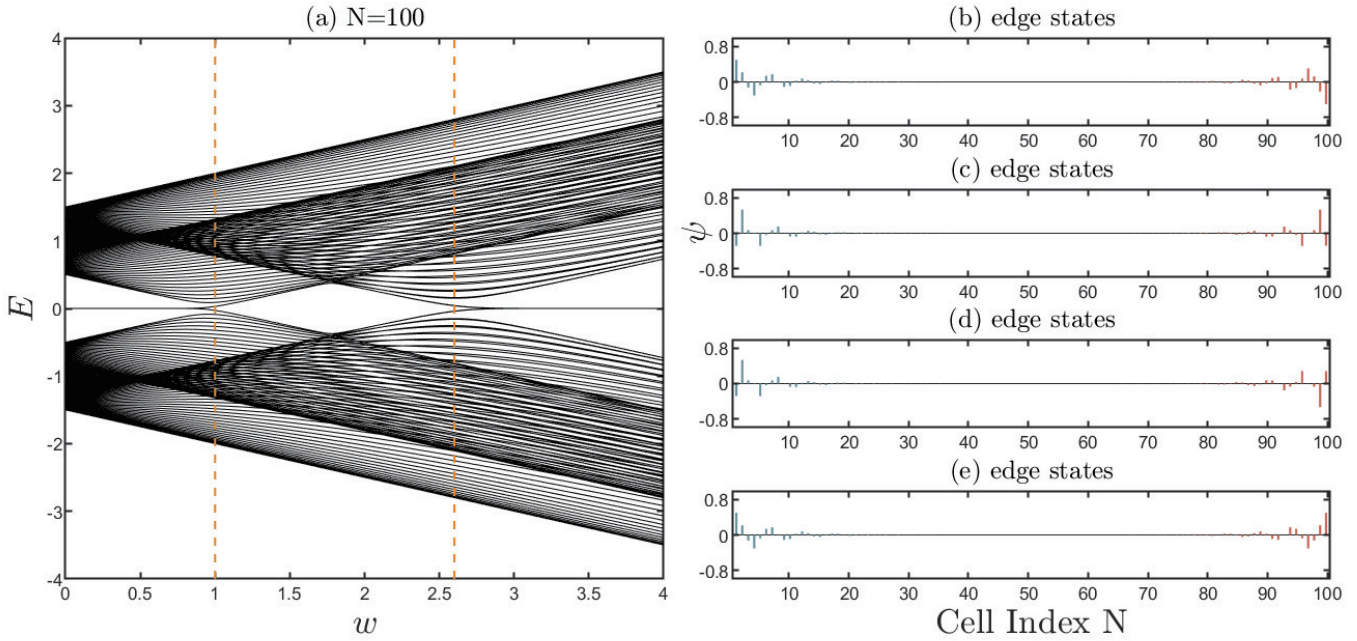


FIG. 6. Winding Number=2存在四个边界态

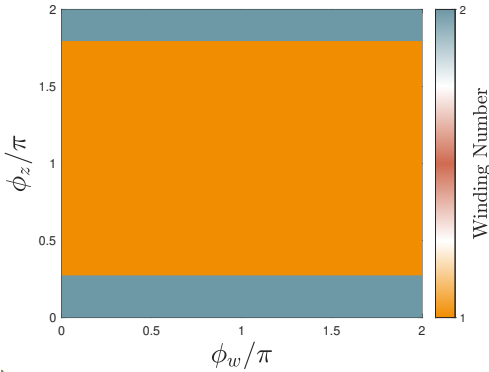


FIG. 7. 相位控制的Winding number。

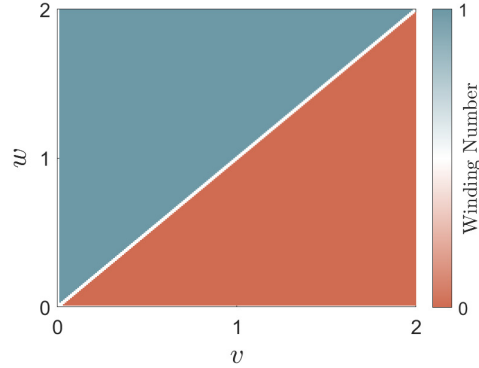


FIG. 8. SSH模型的Winding number。

下午：整理重绘复现图片，完成习题1.2（处理winding number图片变化交界处的方法不一样，所以三张图风格不统一，分别使用的是手动取消网格，光滑处理，光滑处理）。

习题1.2: *Complex-valued hopping amplitudes*- Generalize the SSH model in the following way. Assume that the hopping amplitudes $v = |v|e^{i\psi_v}$ and $w = |w|e^{i\psi_w}$ are complex, and include a third complex-valued hopping amplitude $z = |z|e^{i\psi_z}$ between the states $|m, A\rangle$ and $|m+1, B\rangle$ for every m . Provide a specific example where the tuning of one of the phases changes the bulk winding number.

答：和我设计的结构很像，但是远耦合方向相反且差个系数，并考虑相位。相位控制的Winding number如图7所示， $v=0.5$ 。 $w=0.5$ 。代码如CODE4所示。

2024/06/04 & 2024/06/05

读完 *A Short Course on Topological Insulators* 的第一章并复现图片 [7]，如图8所示，代码位于CODE3。写了一个自动算 2×2 Hamiltonian（频域）系统Winding number的函数，复现代码中可查。完成习题1.1并绘制图片。

习题1.1: *Higher winding numbers*- The SSH model is one-dimensional in space, and has a two-dimensional internal Hilbert space. Construct a lattice model that has these properties of the SSH model, but which has a bulk winding number of 2. Generalize the construction for an arbitrary integer bulk winding number.

答：如FIG. 9所示。代码如CODE2所示。

2024/05/27

$$\hat{H}(k) = \begin{pmatrix} 0 & v + e^{-ik} + 0.5we^{2ik} \\ v + e^{ik} + 0.5we^{-2ik} & 0 \end{pmatrix}.$$

重新制定基础知识学习计划，如FIG. 11所示。图书馆借相关书籍，看高量课程。

2024/05/25 & 2024/05/26

2024/06/03

上午 & 下午：完成Griffths量子力学第二章重点 (*) 习题。

周六日休息。

2024/06/02

休息 & 读Griffths量子力学第二章。

2024/05/24

2024/06/01

休息。

收拾房间，初步了解金属自由电子气体模型。

2024/05/31

摸鱼了。

2024/05/23

2024/05/30

上午：做学习记录。
下午：完成Griffths量子力学第一章重点 (*, **) 习题。

办理校园卡，登记入住宿舍，收拾房间，选择工位。

2024/05/29

上午：做Griffths量子力学 $P_{1.1} - P_{1.7}$ 习题。
下午：读*A Short Course on Topological Insulators*的§ 1.1-§ 1.3并复现如FIG. 10所示（直积不懂不会算，返回复习基础看过Griffths第三章再往下阅读复现） [7]。MATLAB代码如CODE1所示 (B、C、DE)。

- [1] C. Wang, *et al.*, Realization of fractional quantum Hall state with interacting photons, *Science* **384**, 579 (2024), <https://www.science.org/doi/pdf/10.1126/science.ado3912>.
- [2] H. ZHONG, D. MIHALACHE, S. SHEN, and Y. ZHANG, THE BAND STRUCTURE OF HELICAL WAVEGUIDE ARRAYS IN TOPOLOGICAL PHOTONICS: A TUTORIAL, .
- [3] M. C. Rechtsman, *et al.*, Photonic Floquet topological insulators, *Nature* **496**, 196 (2013).
- [4] F. Lederer, *et al.*, Discrete solitons in optics, *Physics Reports* **463**, 1 (2008).
- [5] H. Liu, *et al.*, Acoustic Topological Metamaterials of Large Winding Number, *Phys. Rev. Appl.* **19**, 054028 (2023).
- [6] F. Liu and K. Wakabayashi, Novel Topological Phase with a Zero Berry Curvature, *Phys. Rev. Lett.* **118**, 076803 (2017).
- [7] J. K. Asb'ath, L. Oroszl'ny, and A. P'lyi, *A Short Course on Topological Insulators* (Springer International Publishing, 2016).

2024/05/28

上午：看高量课程，复习经典力学。
下午：阅读Griffths量子力学。

CODE4

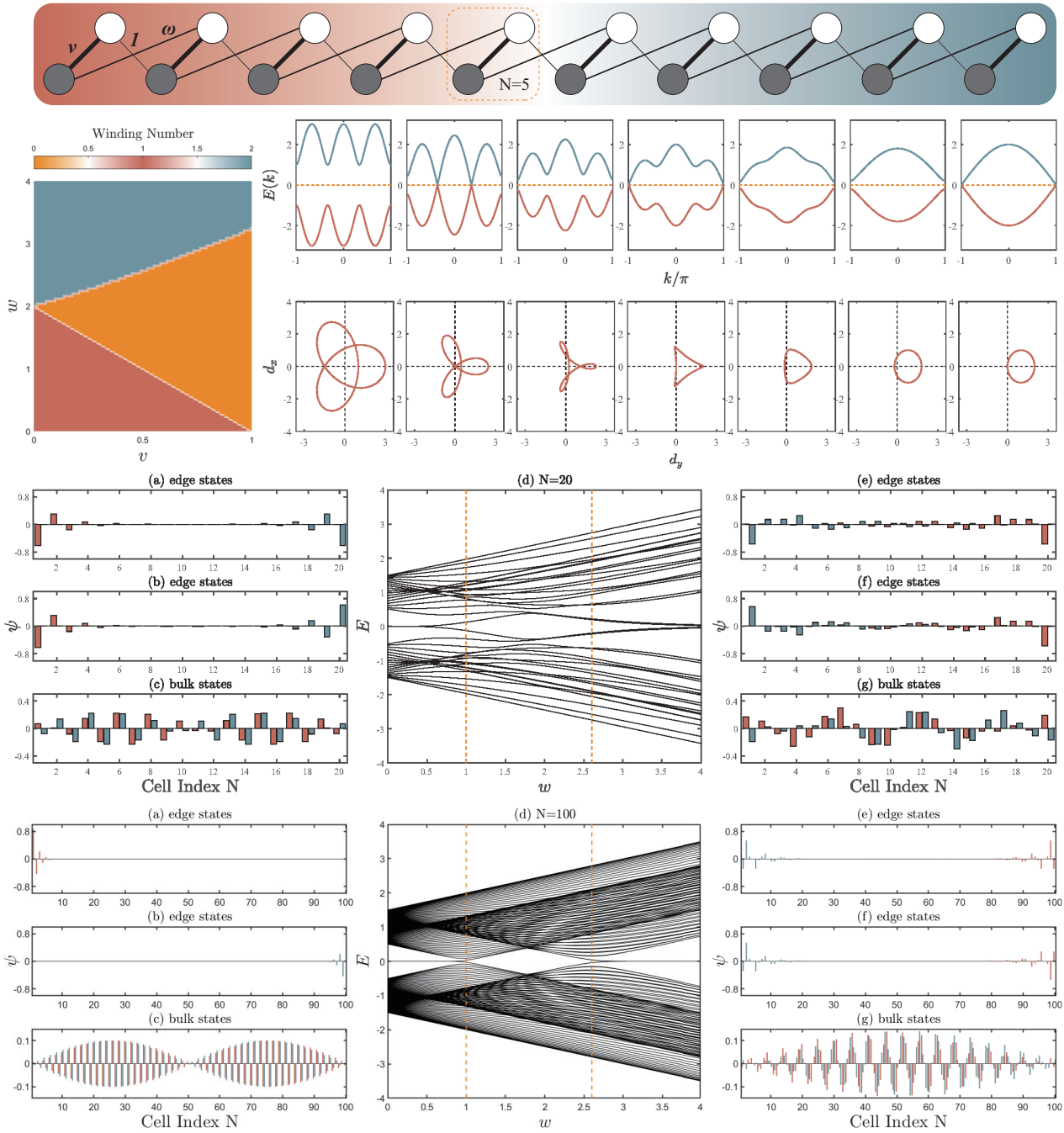


FIG. 9. 设计的Winding number=2的拓扑绝缘体。(Hypocycloid, 圆内螺线)

```

2  clc
3  clear all
4
5  syms k;
6
7  rate=30;
8  parfor_progress(rate);
9
10 n=0;
11 vj=linspace(0,0*pi,rate);

```

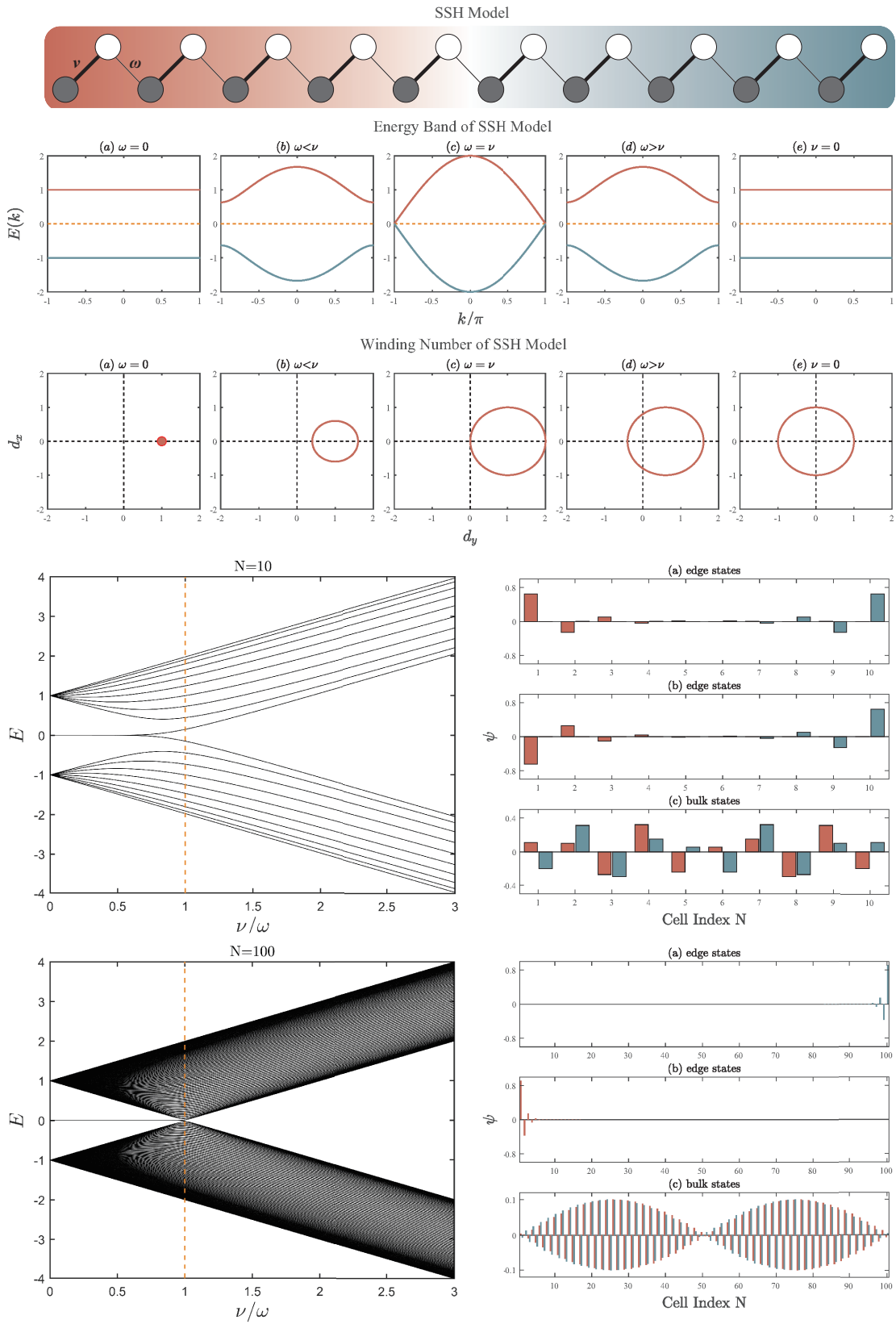


FIG. 10. SSH模型复现图片。自上而下分别为：A. SSH模型图 (cell index $N=10$)。B. SSH模型的能带 (bulk部分)。C. SSH模型的Winding number。D. $N=10$ 时SSH模型的能谱及波函数。 $(\nu = 0.4, \text{对应自下而上第10, 11, 8条能带})$ E. $N=100$ 时SSH模型的能谱及波函数。 $(\nu = 0.4, \text{对应自下而上第100, 101, 98条能带})$

```

12 wj=linspace(0,2*pi,rate);
13 zj=linspace(0,2*pi,rate);
14 v0=linspace(0.5,0.5,rate);
15 w0=linspace(1,1,rate);
16 z0=linspace(1,1,rate);
17
18 w=1;
19
20 parfor a = 1
21     pause(rand); % Replace with real code
22     parfor_progress;
23         v=v0(a)*exp(1i*vj(a));
24     for b = 1 : rate
25
26         w=w0(b)*exp(1i*wj(b));
27         z=z0(b)*exp(1i*zj(b));
28         H_k = [0 v+w*exp(-1i*k)+z*exp(-2i*k); conj(v)+conj(w)*exp(1i*k)+conj(z)*
                exp(2i*k) 0];
29         [winding_number, d, d_0] = calculateWindingNumber(H_k);
30         wn(a,b) = abs(winding_number);
31         if abs(wn(a,b)) <= 1e-3 || (abs(wn(a,b))-1) <= 1e-3 || (abs(wn(a,b))-2)
            <= 1e-3
32             else
33                 wn(a,b) = NaN;
34             end
35         end
36     end
37 parfor_progress(0);
38 toc
39 %%
40 color0=[1,1,1];
41 color2 = [207, 108, 82] / 255;
42 color1 = [241,141,0]/255;
43 color3 = [109, 152, 165] / 255;
44
45 Map = [linspace(color1(1), color0(1), 128)', linspace(color1(2), color0(2), 128)
        ', linspace(color1(3), color0(3), 128)';
        linspace(color0(1), color2(1), 128)', linspace(color0(2), color2(2), 128)',
        linspace(color0(3), color2(3), 128)';
        linspace(color2(1), color0(1), 128)', linspace(color2(2), color0(2), 128)',
        linspace(color2(3), color0(3), 128)';
        linspace(color0(1), color3(1), 128)', linspace(color0(2), color3(2), 128)',
        linspace(color0(3), color3(3), 128)'];
46
47
48
49
50 %%
51 pcolor(wj/(pi),zj/(pi),abs(wn)')
52 xlabel('$\phi_{w}/\pi$', 'interpreter', 'latex', 'FontSize', 22)
53 ylabel('$\phi_{z}/\pi$', 'interpreter', 'latex', 'FontSize', 22)
54 % axis([0 2 0 2])
55 c = colorbar('Ticks',[1 2], 'TickLabels',{ '1', '2' });;%('ticks',linspace(-3,21,5),
    'FontSize',11);
56 c.Label.FontSize=16;
57 c.Label.Interpreter='Latex';
58 c.Label.String = '$\mathrm{Winding\ -Number}$';
59 c.Label.HorizontalAlignment = "center";
60 colormap(Map);

```



```

61
62 delete(gcf);
63
64
65 %%
66 function [winding_number, d, d_0] = calculateWindingNumber(H_k)
67 syms k real;
68
69 sigma_0 = [1 0; 0 1];
70 sigma_x = [0 1; 1 0];
71 sigma_y = [0 -1i; 1i 0];
72 sigma_z = [1 0; 0 -1];
73
74 A = [reshape(sigma_0.', [], 1), reshape(sigma_x.', [], 1), reshape(sigma_y.', [],
75     1), reshape(sigma_z.', [], 1)];
76 b = reshape(H_k.', [], 1);
77
78 coefficients = A \ b;
79
80 d_0 = coefficients(1);
81 d_x = coefficients(2);
82 d_y = coefficients(3);
83 d_z = coefficients(4);
84
85 d = [d_x; d_y; d_z];
86 d = simplify(d);
87
88 d_dk = diff(d, k);
89
90 cross_product = simplify(cross(d, d_dk));
91 norm_d_squared = simplify(norm(d)^2);
92 integrand = cross_product(3) / norm_d_squared;
93
94 func = matlabFunction(integrand, 'Vars', {k});
95
96 winding_number = integral(func, -pi, pi) / (2 * pi);
97
98 %%
99 function percent = parfor_progress(N)
100 %PARFOR_PROGRESS Progress monitor (progress bar) that works with parfor.
101 % PARFOR_PROGRESS works by creating a file called parfor_progress.txt in
102 % your working directory, and then keeping track of the parfor loop's
103 % progress within that file. This workaround is necessary because parfor
104 % workers cannot communicate with one another so there is no simple way
105 % to know which iterations have finished and which haven't.
106 %
107 % PARFOR_PROGRESS(N) initializes the progress monitor for a set of N
108 % upcoming calculations.
109 %
110 % PARFOR_PROGRESS updates the progress inside your parfor loop and
111 % displays an updated progress bar.
112 %
113 % PARFOR_PROGRESS(0) deletes parfor_progress.txt and finalizes progress
114 % bar.
115 %

```

```

116 % To suppress output from any of these functions, just ask for a return
117 % variable from the function calls, like PERCENT = PARFOR_PROGRESS which
118 % returns the percentage of completion.
119 %
120 % Example:
121 %
122 %     N = 100;
123 %     parfor_progress(N);
124 %     parfor i=1:N
125 %         pause(rand); % Replace with real code
126 %         parfor_progress;
127 %     end
128 %     parfor_progress(0);
129 %
130 % See also PARFOR.
131
132 % By Jeremy Scheff - jdscheff@gmail.com - http://www.jeremyscheff.com/
133
134 error(nargchk(0, 1, nargin, 'struct'));
135
136 if nargin < 1
137     N = -1;
138 end
139
140 percent = 0;
141 w = 50; % Width of progress bar
142
143 if N > 0
144     f = fopen('parfor_progress.txt', 'w');
145     if f < 0
146         error('Do you have write permissions for %s?', pwd);
147     end
148     fprintf(f, '%d\n', N); % Save N at the top of progress.txt
149     fclose(f);
150
151     if nargout == 0
152         disp(['--0%[>', repmat('-', 1, w), '']]);
153     end
154 elseif N == 0
155     delete('parfor_progress.txt');
156     percent = 100;
157
158     if nargout == 0
159         disp([repmat(char(8), 1, (w+9)), char(10), '100%[' , repmat('=', 1, w+1),
160             '']]);
161     end
162 else
163     if ~exist('parfor_progress.txt', 'file')
164         error('parfor_progress.txt not found. Run PARFOR_PROGRESS(N) before PARFOR_PROGRESS to initialize parfor_progress.txt.');
```

```

170     f = fopen('parfor_progress.txt', 'r');
171     progress = fscanf(f, '%d');
172     fclose(f);
173     percent = (length(progress)-1)/progress(1)*100;
174
175     if nargin == 0
176         perc = sprintf('%3.0f%%', percent); % 4 characters wide, percentage
177         disp([repmat(char(8), 1, (w+9)), char(10), perc, '[', repmat('=', 1,
178                 round(percent*w/100)), '>', repmat('-', 1, w - round(percent*w/100)),
179                 '']]);
178     end
179 end

```

CODE3

```

1  tic
2  clc
3  clear all
4
5  syms k;
6
7  rate=20;
8  % parfor_progress(rate);
9
10 n=0;
11 va = linspace(0,2,rate+1);
12 va(1) = [];
13 wb = linspace(0,2,rate+1);
14 wb(1) = [];
15 wn = NaN(rate, rate);
16
17 parpool;
18 parfor a = 1 : rate
19     % pause(rand); % Replace with real code
20     % parfor_progress;
21     v=va(a);
22     for b = 1 : rate
23         w=wb(b);
24         H_k = [0 v + w*exp(-1i*k); v + w*exp(1i*k) 0];
25         [winding_number, d, d_0] = calculateWindingNumber(H_k);
26         if v == w
27             wn(a,b) = NaN;
28         else
29             wn(a,b) = winding_number;
30         end
31     end
32 end
33 delete(gcf);
34 % parfor_progress(0);
35 toc
36 %%
37 color1 = [207, 108, 82] / 255;
38 color2 = [1, 1, 1];

```

```

39 color3 = [109, 152, 165] / 255;
40
41 Map = [ linspace(color1(1), color2(1), 128)', linspace(color1(2), color2(2), 128)
42         ', linspace(color1(3), color2(3), 128) '];
43         linspace(color2(1), color3(1), 128)', linspace(color2(2), color3(2), 128)',
44         ', linspace(color2(3), color3(3), 128) '];
45
46 %%
47 pcolor(va,wb,wn')
48 shading interp
49 set(gca,'XTick',[0,1,2],'FontSize',16)
50 set(gca,'YTick',[0,1,2],'FontSize',16)
51 xlabel('$v$', 'interpreter','latex','FontSize',22)
52 ylabel('$w$', 'interpreter','latex','FontSize',22)
53 axis([0 2 0 2])
54 c = colorbar('Ticks',[0 1]);;%('ticks',linspace(-3,21,5),'FontSize',11);
55 c.Label.FontSize=16;
56 c.Label.Interpreter='Latex';
57 c.Label.String = '$\mathrm{Winding}\text{-Number}$';
58 c.Label.HorizontalAlignment = "center";
59 colormap(Map);
60 %save("Phase_diagram_winding_number",'-mat')
61 %%
62 function [winding_number, d, d_0] = calculateWindingNumber(H,k)
63 syms k real;
64
65 sigma_0 = [1 0; 0 1];
66 sigma_x = [0 1; 1 0];
67 sigma_y = [0 -1i; 1i 0];
68 sigma_z = [1 0; 0 -1];
69
70 A = [reshape(sigma_0.', [], 1), reshape(sigma_x.', [], 1), reshape(sigma_y.', [],
71         1), reshape(sigma_z.', [], 1)];
72 b = reshape(H.k.', [], 1);
73
74 coefficients = A \ b;
75
76 d_0 = coefficients(1);
77 d_x = coefficients(2);
78 d_y = coefficients(3);
79 d_z = coefficients(4);
80
81 d = [d_x; d_y; d_z];
82 d = simplify(d);
83
84 d_dk = diff(d, k);
85
86 cross_product = simplify(cross(d, d_dk));
87 norm_d_squared = simplify(norm(d)^2);
88 integrand = cross_product(3) / norm_d_squared;
89
90 func = matlabFunction(integrand, 'Vars', {k});
91
92 winding_number = integral(func, -pi, pi) / (2 * pi);
93 end

```

CODE2

```

1  tic
2  clc
3  clear all
4
5  syms k;
6
7  rate=20;
8  parfor_progress(rate);
9
10 n=0;
11 va = linspace(0.0001,1,rate);
12 wb = linspace(0.0001,4,rate);
13 wn = NaN(rate, rate);
14
15 parfor a = 1 : rate
16     parfor_progress;
17     v=va(a);
18     for b = 1 : rate
19         w=wb(b);
20         H.k = [0 v+1()*exp(-1i*k) + 0.5*w*exp(1i*2*k); v+1*exp(1i*k) + 0.5*w*exp
                (-1i*2*k) 0];
21         [winding_number, d, d_0] = calculateWindingNumber(H.k);
22         wn(a,b) = abs(winding_number);
23         if abs(wn(a,b)) <= 1e-3 || (abs(wn(a,b))-1) <= 1e-3 || (abs(wn(a,b))-2)
                <= 1e-3
24             else
25                 wn(a,b) = NaN;
26             end
27         end
28     end
29 parfor_progress(0);
30 toc
31 %%
32 color0=[1,1,1];
33 color2 = [207, 108, 82] / 255;
34 color1 = [241,141,0]/255;
35 color3 = [109, 152, 165] / 255;
36
37 Map = [linspace(color1(1), color0(1), 128)', linspace(color1(2), color0(2), 128)
        ', linspace(color1(3), color0(3), 128)';
        linspace(color0(1), color2(1), 128)', linspace(color0(2), color2(2), 128)',
        linspace(color0(3), color2(3), 128)';
        linspace(color2(1), color0(1), 128)', linspace(color2(2), color0(2), 128)',
        linspace(color2(3), color0(3), 128)';
        linspace(color0(1), color3(1), 128)', linspace(color0(2), color3(2), 128)',
        linspace(color0(3), color3(3), 128)'];
38
39
40
41
42 %%
43 pcolor(va,wb,abs(wn)')
44 shading interp
45 % set(gca,'XTick',[0,1,2],'FontSize',16)
46 % set(gca,'YTick',[0,1,2],'FontSize',16)
47 xlabel('$v$', 'interpreter', 'latex', 'FontSize',22)

```

```

48 ylabel('$w$', 'interpreter', 'latex', 'FontSize', 22)
49 % axis([0 2 0 2])
50 c = colorbar;%('ticks', linspace(-3,21,5), 'FontSize', 11);
51 c.Label.FontSize=16;
52 c.Label.Interpreter='Latex';
53 c.Label.String = '$\mathrm{Winding\{-Number\}}$';
54 c.Label.HorizontalAlignment = "center";
55 colormap(Map);
56 % save('WN_012.mat')
57 delete(gcf);
58
59 %%
60 function [winding_number, d, d_0] = calculateWindingNumber(H_k)
61 syms k real;
62
63 sigma_0 = [1 0; 0 1];
64 sigma_x = [0 1; 1 0];
65 sigma_y = [0 -1i; 1i 0];
66 sigma_z = [1 0; 0 -1];
67
68 A = [reshape(sigma_0.', [], 1), reshape(sigma_x.', [], 1), reshape(sigma_y.', [],
69     1), reshape(sigma_z.', [], 1)];
70 b = reshape(H_k.', [], 1);
71
72 coefficients = A \ b;
73
74 d_0 = coefficients(1);
75 d_x = coefficients(2);
76 d_y = coefficients(3);
77 d_z = coefficients(4);
78
79 d = [d_x; d_y; d_z];
80 d = simplify(d);
81
82 d_dk = diff(d, k);
83
84 cross_product = simplify(cross(d, d_dk));
85 norm_d_squared = simplify(norm(d)^2);
86 integrand = cross_product(3) / norm_d_squared;
87
88 func = matlabFunction(integrand, 'Vars', {k});
89
90 winding_number = integral(func, -pi, pi) / (2 * pi);
91 end
92
93 %%
94 function percent = parfor_progress(N)
95 %PARFOR_PROGRESS Progress monitor (progress bar) that works with parfor.
96 % PARFOR_PROGRESS works by creating a file called parfor_progress.txt in
97 % your working directory, and then keeping track of the parfor loop's
98 % progress within that file. This workaround is necessary because parfor
99 % workers cannot communicate with one another so there is no simple way
100 % to know which iterations have finished and which haven't.
101 %
102 % PARFOR_PROGRESS(N) initializes the progress monitor for a set of N
    upcoming calculations.

```

```

103 %
104 % PARFOR_PROGRESS updates the progress inside your parfor loop and
105 % displays an updated progress bar.
106 %
107 % PARFOR_PROGRESS(0) deletes parfor_progress.txt and finalizes progress
108 % bar.
109 %
110 % To suppress output from any of these functions, just ask for a return
111 % variable from the function calls, like PERCENT = PARFOR_PROGRESS which
112 % returns the percentage of completion.
113 %
114 % Example:
115 %
116 %     N = 100;
117 %     parfor_progress(N);
118 %     parfor i=1:N
119 %         pause(rand); % Replace with real code
120 %         parfor_progress;
121 %     end
122 %     parfor_progress(0);
123 %
124 % See also PARFOR.
125
126 % By Jeremy Scheff - jdscheff@gmail.com - http://www.jeremyscheff.com/
127
128 error(nargchk(0, 1, nargin, 'struct'));
129
130 if nargin < 1
131     N = -1;
132 end
133
134 percent = 0;
135 w = 50; % Width of progress bar
136
137 if N > 0
138     f = fopen('parfor_progress.txt', 'w');
139     if f < 0
140         error('Do you have write permissions for %s?', pwd);
141     end
142     fprintf(f, '%d\n', N); % Save N at the top of progress.txt
143     fclose(f);
144
145     if nargout == 0
146         disp(['--0%[>', repmat('-', 1, w), '']]);
147     end
148 elseif N == 0
149     delete('parfor_progress.txt');
150     percent = 100;
151
152     if nargout == 0
153         disp([repmat(char(8), 1, (w+9)), char(10), '100%[' , repmat('=', 1, w+1),
154             '']]);
155     end
156 else
157     if ~exist('parfor_progress.txt', 'file')

```

```

157         error('parfor_progress.txt not found.-Run-PARFOR_PROGRESS(N)-before-
           PARFOR_PROGRESS-to-initialize-parfor_progress.txt.');
```

158 end

```

159
160 f = fopen('parfor_progress.txt', 'a');
161 fprintf(f, '1\n');
162 fclose(f);
163
164 f = fopen('parfor_progress.txt', 'r');
165 progress = fscanf(f, '%d');
166 fclose(f);
167 percent = (length(progress)-1)/progress(1)*100;
168
169 if nargin == 0
170     perc = sprintf('%3.0f%%', percent); % 4 characters wide, percentage
171     disp([repmat(char(8), 1, (w+9)), char(10), perc, '[', repmat('=', 1,
172         round(percent*w/100)), '>', repmat('-', 1, w - round(percent*w/100)),
173         ']'']);
172 end
173 end
```

```

1 clear all
2 clc
3 f1=figure(1);
4 position0=get(f1, 'position');
5 set(f1, 'position', position0+[-0.8*position0(3), 0, 1.6*position0(3), -0.3*position0
   (4)]);
6 tiledlayout("horizontal", "TileSpacing", "tight");
7 titlej={'$(a)\-\omega=0$'; '$(b)\-\omega\textless\ \nu$'; '$(c)\-\omega=\nu$'; '$(d)\-\
   \omega\textgreater\ \nu$'; '$(e)\-\nu=0$'};
8
9 rate_k=1000;
10
11 vj=[0, 0.3, 0.5, 0.6, 0.7, 0.8, 1];
12 wj=[4, 2.3, 1.5, 0.8, 0.3, 0, 0];
13 ki=linspace(-pi, pi, rate_k);
14
15 for j = 1 : 7
16     v=vj(j);
17     w=wj(j);
18     for i = 1 : rate_k
19         k=ki(i);
20         H_k = [0 v+1*exp(-1i*k) + 0.5*w*exp(1i*2*k); v+1*exp(1i*k) + 0.5*w*exp(-1
21             i*2*k) 0];
22         [d, d_0] = WindingNumberFigure(H_k);
23         d_x(i)=d(1);
24         d_y(i)=d(2);
25     end
26     nexttile
27     plot(linspace(-3.6, 3.6, 100), linspace(0, 0, 100), 'LineWidth', 1, 'linestyle', '—',
28         'Color', 'k')
29     hold on
30     plot(linspace(0, 0, 100), linspace(-4, 4, 100), 'LineWidth', 1, 'linestyle', '—',
31         'Color', 'k')
32     hold on
```



```

30     plot(d_x,d_y,'LineWidth',1.5,'Color',[207,108,82]/255);
31     hold on
32     set(gca,'LineWidth',1);
33     axis([-3.6,3.6,-4,4])
34     set(gca,'ytick',[-4 -2 0 2 4],'FontSize',12);
35     set(gca,'xtick',[-3 0 3],'FontSize',12);
36     % title(titlej{j},'interpreter','latex','FontSize',12);
37 end
38 % title(t,'Winding Number of SSH Model','FontSize',18);
39 xlabel(t,'$d_{y}$','interpreter','latex','FontSize',18);
40 ylabel(t,'$d_{x}$','interpreter','latex','FontSize',18);
41
42 function [d, d_0] = WindingNumberFigure(H,k)
43 syms k v w
44
45 sigma_0 = [1 0; 0 1];
46 sigma_x = [0 1; 1 0];
47 sigma_y = [0 -1i; 1i 0];
48 sigma_z = [1 0; 0 -1];
49
50 A = [reshape(sigma_0.', [], 1), reshape(sigma_x.', [], 1), reshape(sigma_y.', [],
51     1), reshape(sigma_z.', [], 1)];
52 b = reshape(H.k.', [], 1);
53
54 coefficients = A \ b;
55
56 d_0 = coefficients(1);
57 d_x = coefficients(2);
58 d_y = coefficients(3);
59 d_z = coefficients(4);
60
61 d = [d_x; d_y; d_z];
62 %d = simplify(d);
63 end

```

```

1 clear all
2 clc
3
4 %%
5 f2=figure(1);
6 position0=get(f2,'position');
7 set(f2,'position',position0+[-0.85*position0(3),-0.3*position0(4),1.7*position0
8     (3),0*position0(4)]);
9 t=tiledlayout(3,3,"TileSpacing","tight");
10
11 f2=figure(1);
12 position0=get(f2,'position');
13 set(f2,'position',position0+[-0*position0(3),-0*position0(4),-0.2*position0(3)
14     ,0.34*position0(4)]);
15
16 n=100;
17 %%
18 rate_v=500;
19
20 v=0.5;

```

```

19 wj=linspace(0,4,rate_v);
20 for j = 1 : rate_v
21     w=wj(j);
22     v1=[v,1]'*ones(1,n);
23     v1=v1(:)';
24     v1(2*n)=[];
25     w0=[0.5*w,0]'*ones(1,n);
26     w0=w0(:)';
27     w0(2*n)=[];
28     w0(2*n-1)=[];
29     w0(2*n-2)=[];
30     w0(2*n-3)=[];
31     w0(2*n-4)=[];
32     H=diag(v1,1)+diag(v1,-1)+diag(w0,5)+diag(w0,-5);
33     [V,D]=eig(H);
34     E(j,:)=diag(D);
35     %psi(:,j)=V;
36 end
37 nexttile(2,[3,1]);
38 for j = 1 : size(E,2)
39     plot(wj,E(:,j),'Color','black');
40     hold on;
41 end
42 plot(linspace(1,1,100),linspace(-4,4,100),'LineWidth',1,'linestyle','—','Color',
,[241,141,0]/255)
43 hold on;
44 plot(linspace(2.6,2.6,100),linspace(-4,4,100),'LineWidth',1,'linestyle','—','
Color',[241,141,0]/255)
45 hold on;
46 set(gca,'LineWidth',1);
47 xlabel('$w$', 'interpreter','latex','FontSize',16);
48 ylabel('$E$', 'interpreter','latex','FontSize',16);
49 title(sprintf("(d) N=%d", n), 'interpreter','latex','FontSize',12);
50
51 %ratio=400;
52
53 v=0.5;
54 w=0;
55 v1=[v,1]'*ones(1,n);
56 v1=v1(:)';
57 v1(2*n)=[];
58 w0=[0.5*w,0]'*ones(1,n);
59 w0=w0(:)';
60 w0(2*n)=[];
61 w0(2*n-1)=[];
62 w0(2*n-2)=[];
63 w0(2*n-3)=[];
64 w0(2*n-4)=[];
65 H=diag(v1,1)+diag(v1,-1)+diag(w0,5)+diag(w0,-5);
66 [V,D]=eig(H);
67 psi=V;
68
69 x=linspace(0.75,10.25,20);
70 nexttile(1);
71 %bar(x, reshape(psi(:,10,ratio),[2,10]))
72 b1=bar(reshape(psi(:,n),[2,n]),0.9,'GroupWidth',0.8);

```

```

73 set(gca,'ytick',[-0.8,0,0.8])
74 axis([0.5,n+0.5,-1,1])
75 title({'(a)-edge-states'},'VerticalAlignment','bottom','interpreter','latex','',
       'FontSize',12);
76 set(b1(1),'FaceColor',[207,108,82]/255);
77 set(b1(2),'FaceColor',[109,152,165]/255);
78 set(gca,'LineWidth',1);
79
80 nexttile(4);
81 %bar(x,reshape(psi(:,11,ratio),[2,10]))
82 b2=bar(reshape(psi(:,n+1),[2,n]),0.9,'GroupWidth',0.8);
83 set(gca,'ytick',[-0.8,0,0.8])
84 axis([0.5,n+0.5,-1,1])
85 title({'(b)-edge-states'},'VerticalAlignment','bottom','interpreter','latex','',
       'FontSize',12);
86 set(b2(1),'FaceColor',[207,108,82]/255);
87 set(b2(2),'FaceColor',[109,152,165]/255);
88 set(gca,'LineWidth',1);
89 ylabel('\psi','interpreter','latex','FontSize',16,'VerticalAlignment','middle')
90 ;
91 nexttile(7);
92 %bar(x,reshape(psi(:,8,ratio),[2,10]))
93 b3=bar(reshape(psi(:,n-2),[2,n]),0.9,'GroupWidth',0.8);
94 if n >= 100
95     set(gca,'ytick',[-0.1,0,0.1])
96     axis([0.5,n+0.5,-0.15,0.15])
97 else
98     set(gca,'ytick',[-0.4,0,0.4])
99     axis([0.5,n+0.5,-0.5,0.5])
100 end
101 title({'(c)-bulk-states'},'VerticalAlignment','bottom','interpreter','latex','',
       'FontSize',12);
102 set(b3(1),'FaceColor',[207,108,82]/255);
103 set(b3(2),'FaceColor',[109,152,165]/255);
104 set(gca,'LineWidth',1);
105
106 xlabel('Cell-Index-N','interpreter','latex','FontSize',16);
107
108 %%
109
110 v=0.5;
111 w=4;
112 v1=[v,1]*ones(1,n);
113 v1=v1(:)';
114 v1(2*n)=[];
115 w0=[0.5*w,0]*ones(1,n);
116 w0=w0(:)';
117 w0(2*n)=[];
118 w0(2*n-1)=[];
119 w0(2*n-2)=[];
120 w0(2*n-3)=[];
121 w0(2*n-4)=[];
122 H=diag(v1,1)+diag(v1,-1)+diag(w0,5)+diag(w0,-5);
123 [V,D]=eig(H);
124 psi=V;

```

```

125
126 x=linspace(0.75,10.25,20);
127 nexttile(3);
128 %bar(x, reshape(psi(:,10,ratio),[2,10]))
129 b1=bar(reshape(psi(:,n),[2,n]),0.9,'GroupWidth',0.8);
130 set(gca,'ytick',[-0.8,0,0.8])
131 axis([0.5,n+0.5,-1,1])
132 title({'(e)-edge-states'},'VerticalAlignment','bottom','interpreter','latex','
        FontSize',12);
133 set(b1(1),'FaceColor',[207,108,82]/255);
134 set(b1(2),'FaceColor',[109,152,165]/255);
135 set(gca,'LineWidth',1);
136
137 nexttile(6);
138 %bar(x, reshape(psi(:,11,ratio),[2,10]))
139 b2=bar(reshape(psi(:,n+1),[2,n]),0.9,'GroupWidth',0.8);
140 set(gca,'ytick',[-0.8,0,0.8])
141 axis([0.5,n+0.5,-1,1])
142 title({'(f)-edge-states'},'VerticalAlignment','bottom','interpreter','latex','
        FontSize',12);
143 set(b2(1),'FaceColor',[207,108,82]/255);
144 set(b2(2),'FaceColor',[109,152,165]/255);
145 set(gca,'LineWidth',1);
146 ylabel('$\psi$', 'interpreter','latex','FontSize',16,'VerticalAlignment','middle')
        ;
147
148 nexttile(9);
149 %bar(x, reshape(psi(:,8,ratio),[2,10]))
150 b3=bar(reshape(psi(:,n-2),[2,n]),0.9,'GroupWidth',0.8);
151 if n >= 100
152     set(gca,'ytick',[-0.1,0,0.1])
153     axis([0.5,n+0.5,-0.15,0.15])
154 else
155     set(gca,'ytick',[-0.4,0,0.4])
156     axis([0.5,n+0.5,-0.5,0.5])
157 end
158 title({'(g)-bulk-states'},'VerticalAlignment','bottom','interpreter','latex','
        FontSize',12);
159 set(b3(1),'FaceColor',[207,108,82]/255);
160 set(b3(2),'FaceColor',[109,152,165]/255);
161 set(gca,'LineWidth',1);
162
163 xlabel('Cell-Index-N','interpreter','latex','FontSize',16);

```

CODE1

```

1 clear all
2 clc
3 f1=figure(1);
4 position0=get(f1,'position');
5 set(f1,'position',position0+[-0.75*position0(3),0,1.5*position0(3),-0.27*
    position0(4)]);
6 t=tilelayout("horizontal","TileSpacing","tight");

```

```

7  titlej={'$(a)\-\omega=0$'; '$(b)\-\omega\textless\nu$'; '$(c)\-\omega=\nu$'; '$(d)\-\omega\textgreater\nu$'; '$(e)\-\nu=0$'};
8
9  rate_k=1000;
10
11 vj=[1,1,1,0.6,0];
12 wj=[0,0.6,1,1,1];
13 ki=linspace(-pi,pi,rate_k);
14
15 for j = 1 : 5
16     v=vj(j);
17     w=wj(j);
18     for i = 1 : rate_k
19         k=ki(i);
20         E_1(i)=sqrt(v+w+2*v*w*cos(k));
21         E_2(i)=-sqrt(v+w+2*v*w*cos(k));
22     end
23     nexttile
24     plot(ki/pi,linspace(0,0,rate_k),'LineWidth',1,'linestyle','—','Color',
25         ,[241,141,0]/255)
26     hold on
27     plot(ki/pi,E_1,'LineWidth',1.5,'Color',[207,108,82]/255);
28     hold on
29     plot(ki/pi,E_2,'LineWidth',1.5,'Color',[109,152,165]/255);
30     hold on
31     set(gca,'LineWidth',1);
32     axis([-1,1,-2,2])
33     title(titlej{j},'interpreter','latex','FontSize',12);
34 end
35 title(t,'Energy Band of SSH Model','FontSize',16);
36 xlabel(t,'$k/\pi$','interpreter','latex','FontSize',16);
37 ylabel(t,'$E(k)$','interpreter','latex','FontSize',16);

```

```

1  clear all
2  clc
3  f1=figure(1);
4  position0=get(f1,'position');
5  set(f1,'position',position0+[-0.75*position0(3),0,1.5*position0(3),-0.27*
6     position0(4)]);
7  t=tiledlayout("horizontal","TileSpacing","tight");
8  titlej={'$(a)\-\omega=0$'; '$(b)\-\omega\textless\nu$'; '$(c)\-\omega=\nu$'; '$(d)\-\omega\textgreater\nu$'; '$(e)\-\nu=0$'};
9
10 rate_k=1000;
11
12 vj=[1,1,1,0.6,0];
13 wj=[0,0.6,1,1,1];
14 ki=linspace(-pi,pi,rate_k);
15
16 for j = 1 : 5
17     v=vj(j);
18     w=wj(j);
19     for i = 1 : rate_k
20         k=ki(i);
21         d_x(i)=v+w*cos(k);

```

```

21     d_y(i)=w*sin(k);
22 end
23 nexttile
24 plot(linspace(-2,2,100),linspace(0,0,100),'LineWidth',1,'linestyle','—','Color','k')
25 hold on
26 plot(linspace(0,0,100),linspace(-2,2,100),'LineWidth',1,'linestyle','—','Color','k')
27 hold on
28 if j == 1
29     plot(d_x,d_y,'ro','MarkerFaceColor',[207,108,82]/255,'MarkerSize',9);
30 else
31     plot(d_x,d_y,'LineWidth',1.5,'Color',[207,108,82]/255);
32 end
33 hold on
34 set(gca,'LineWidth',1);
35 axis([-2,2,-2,2])
36 title(titlej{j},'interpreter','latex','FontSize',12);
37 end
38 title(t,'Winding-Number-of-SSH-Model','FontSize',16);
39 xlabel(t,'$d_{y}$','interpreter','latex','FontSize',16);
40 ylabel(t,'$d_{x}$','interpreter','latex','FontSize',16);

```

```

1 clear all
2 clc
3
4 rate_v=30000;
5 n=100;
6
7 w=1;
8 vj=linspace(0,3,rate_v);
9
10 for j = 1 : rate_v
11     v=vj(j);
12     vw=[v,w]'*ones(1,n);
13     vw=vw(:)';
14     vw(2*n)=[];
15     H=diag(vw,1)+diag(vw,-1);
16     [V,D]=eig(H);
17     E(j,:)=diag(D);
18     %psi(:,j)=V;
19 end
20 for j = 1 : size(E,2)
21     plot(vj,E(:,j),'Color','black');
22     hold on;
23 end
24 plot(linspace(1,1,100),linspace(-4,4,100),'LineWidth',1,'linestyle','—','Color',[241,141,0]/255)
25 set(gca,'LineWidth',1);
26 xlabel('$\nu/\omega$','interpreter','latex','FontSize',16);
27 ylabel('$E$','interpreter','latex','FontSize',16);
28 title(sprintf("N=%d",n),'interpreter','latex','FontSize',12);
29
30 %%
31 f2=figure(2);

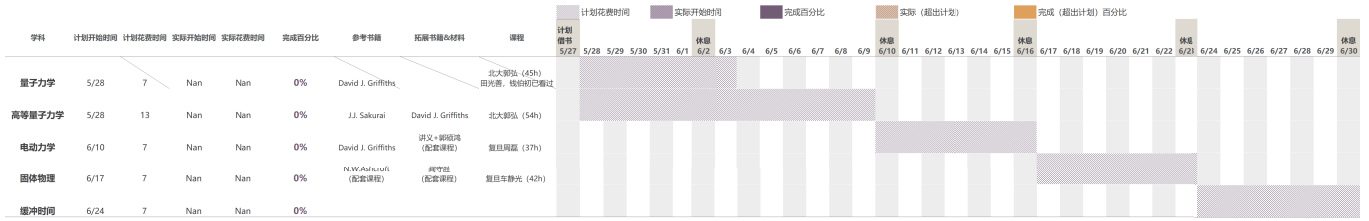
```

```

32 position0=get(f2,'position');
33 set(f2,'position',position0+[-0.4*position0(3),-0.3*position0(4),0.2*position0(3),
    0.3*position0(4)]);
34 t= tiledlayout("vertical","TileSpacing","tight");
35
36 %ratio=400;
37
38 v=0.4;
39 w=1;
40 vw=[v,w]*ones(1,n);
41 vw=vw(:)';
42 vw(2*n)=[];
43 H=diag(vw,1)+diag(vw,-1);
44 [V,D]=eig(H);
45 psi=V;
46
47 x=linspace(0.75,10.25,20);
48 nexttile
49 %bar(x,reshape(psi(:,10,ratio),[2,10]))
50 b1=bar(reshape(psi(:,n),[2,n]),0.9,'GroupWidth',0.8);
51 set(gca,'ytick',[-0.8,0,0.8])
52 axis([0.5,n+0.5,-1,1])
53 title({'(a)-edge-states'},'VerticalAlignment','bottom','interpreter','latex','
    FontSize',12);
54 set(b1(1),'FaceColor',[207,108,82]/255);
55 set(b1(2),'FaceColor',[109,152,165]/255);
56
57 nexttile
58 %bar(x,reshape(psi(:,11,ratio),[2,10]))
59 b2=bar(reshape(psi(:,n+1),[2,n]),0.9,'GroupWidth',0.8);
60 set(gca,'ytick',[-0.8,0,0.8])
61 axis([0.5,n+0.5,-1,1])
62 title({'(b)-edge-states'},'VerticalAlignment','bottom','interpreter','latex','
    FontSize',12);
63 set(b2(1),'FaceColor',[207,108,82]/255);
64 set(b2(2),'FaceColor',[109,152,165]/255);
65
66 nexttile
67 %bar(x,reshape(psi(:,8,ratio),[2,10]))
68 b3=bar(reshape(psi(:,n-2),[2,n]),0.9,'GroupWidth',0.8);
69 if n >= 100
70     set(gca,'ytick',[-0.1,0,0.1])
71     axis([0.5,n+0.5,-0.12,0.12])
72 else
73     set(gca,'ytick',[-0.4,0,0.4])
74     axis([0.5,n+0.5,-0.5,0.5])
75 end
76 title({'(c)-bulk-states'},'VerticalAlignment','bottom','interpreter','latex','
    FontSize',12);
77 set(b3(1),'FaceColor',[207,108,82]/255);
78 set(b3(2),'FaceColor',[109,152,165]/255);
79
80 ylabel(t,'$\psi$','interpreter','latex','FontSize',16);
81 xlabel(t,'Cell-Index-N','interpreter','latex','FontSize',16);

```

学习规划



P. S. 李海萌只推荐量子力学一并学习, 基础过完之后看沈晓清的Topological insulator; 钱伯初推荐刘文科的半导体物理以及一些高等凝聚态讲义, 计划过完基础看一看。

FIG. 11. 基础知识学习计划