Table of Contents

Matlab script to calculate Z_2 invariant

This matlab routine uses the self-energy output of the DMFT+ CTAUX code for self-energy and define the effective topological Hamiltonian and diginalise the effective Hamiltonian with spin twisted boundary condition along x- direction and periodic boundary condition along y- directions.

```
clear all;
clc;
```

Diagonalization of the effective Hamiltonian

```
disp('QSH calculation......')
%plot 1
Nk=100; % Number of grid points in periodic boundary condition
lambda =3.5; % staggering potential
count=1;
tx=1.0; % Hopping matrix element along x- directions
ty=1.0; % Hopping matrix element a
gama=0.25; % Spin-mixing term
n=6;          % Size of the magnetic Brillouin zone; it is 6 presently
 as p/q=1/6
Ntheta=20;  % Number of grid points in twisted boundary condition

nn=2.23; % self-energy at zero frequency for full self-energy and at
 large frequency for Hartree fock
U=7.0; % interaction strength
A=0.0;
%initialise the matrix
sigmaup=zeros(n,n);
sigmadw=zeros(n,n);
sigmaud=zeros(n,n);
sigmadu=zeros(n,n);
gap=zeros(2*n,2*n,2*Ntheta+1,2*Nk+1);
size(gap);
gap_abs=zeros(2*Ntheta+1,2*Nk+1);
%full matrix initialization
comp=zeros(2*n,2*n);


for nth=1:2*Ntheta+1

for nky=1:2*Nk+1
```

```
    theta1=pi/Ntheta*(nth-Ntheta-1);
    ky = pi/Nk*(nky-Nk-1);



 for q=1:1:n
 for   p=1:1:n
     x=q;
     y=q;
     if p==q


 sigmaup(q,p)=-
ty*(exp(1i*ky)*exp(1i*2.0*(x)*pi/6)+exp(-1i*ky)*exp(-1i*2.0*(x)*pi/6))+(-1)^x*lamb
+A-(-1)^y*nn;
     end
 end
end


 %Now off diagonal entries
 for q=1:1:n
 for   p=1:1:n
     if (p-q)==1
 sigmaup(q,p)=-tx*cos(2*pi*gama);
     end
 end
 end
 for q=1:1:n
 for   p=1:1:n
     if (q-p)==1
 sigmaup(q,p)=-tx*cos(2*pi*gama);
     end
 end
 end

%corner term
sigmaup(1,q)=-tx*cos(2*pi*gama)*exp(-1j*theta1);
sigmaup(q,1)=-tx*cos(2*pi*gama)*exp(1j*theta1);

%sigma down term
for q=1:1:n
 for   p=1:1:n
     x=q;
     y=q;
     if p==q


 sigmadw(q,p)=-
ty*(exp(1i*ky)*exp(-1i*2.0*(x)*pi/6)+exp(-1i*ky)*exp(1i*2.0*(x)*pi/6))+(-1)^x*lamb
+A-(-1)^y*nn;
     end
 end
end
```

```
 %Now off diagonal entries
 for q=1:1:n
 for   p=1:1:n
     if (p-q)==1
 sigmadw(q,p)=-tx*cos(2*pi*gama);
     end
 end
 end
 for q=1:1:n
 for   p=1:1:n
     if (q-p)==1
 sigmadw(q,p)=-tx*cos(2*pi*gama);
     end
 end
 end
 %corner term
sigmadw(1,6)=-tx*cos(2*pi*gama)*exp(1j*theta1);
sigmadw(6,1)=-tx*cos(2*pi*gama)*exp(-1j*theta1);



 %sigma up down
for q=1:1:n
 for   p=1:1:n
     if (p-q)==1
 sigmaud(q,p)=-tx*(1j*sin(2*pi*gama));
     end
 end
 end
 for q=1:1:n
 for   p=1:1:n
     if (q-p)==1
 sigmaud(q,p)=-tx*(-1j*sin(2*pi*gama));
     end
 end
 end

%corner term
sigmaud(1,p)=-tx*(-1j*sin(2*pi*gama))*exp(-1j*theta1);
sigmaud(q,1)=-tx*(1j*sin(2*pi*gama))*exp(-1j*theta1);

 %sigma down up
 for q=1:1:n
 for   p=1:1:n
     if (p-q)==1
 sigmadu(q,p)=-tx*(1j*sin(2*pi*gama));
     end
 end
 end
 for q=1:1:n
 for   p=1:1:n
     if (q-p)==1
```

```
 sigmadu(q,p)=-tx*(-1j*sin(2*pi*gama));
     end
 end
 end

sigmadu(1,q)=-tx*(-1j*sin(2*pi*gama))*exp(1j*theta1);
sigmadu(q,1)=-tx*(1j*sin(2*pi*gama))*exp(1j*theta1);


 for q=1:1:n
 for   p=1:1:n
 comp(2*q,2*p)=sigmaup(q,p);
 end
end
for q=1:1:n
 for   p=1:1:n
 comp(2*q-1,2*p-1)=sigmadw(q,p);
 end
end
for q=1:1:n
 for   p=1:1:n
 comp(2*q,2*p-1)=sigmaud(q,p);
 end
end
for q=1:1:n
 for   p=1:1:n
 comp(2*q-1,2*p)=sigmadu(q,p);
 end
end
 Ekk=comp;

  [evec,eval]=eig(Ekk);


  [evec,eval]=sortem(evec,eval);

 gap(:,:,nth,nky)=evec;

 eval=diag(eval);
%  plot(ky,eval,'r--o')
%  hold on
 gap_abs(nth,nky)=min(abs(eval));

end
end
```

Calculating the gap

```
size(gap)
gapabs=min(gap_abs);
gapabs=2.0*min(gapabs);

gap(:,:,nth,nky);
```

Calculating the Z_2 invariant using the avove eigen vectors

First define the link variable according to the notes in our preprint based on the Fukui et al work Phys. Rev. B 75, 121403 (2007)

```
U1=zeros(2*Ntheta+1,2*Nk+1); % Link variable in k_x direction

for nth=1:2*Ntheta+1
 for nky=1:2*Nk+1
%
%    %periodic boundry conditions

   if (nth < 2*Ntheta+1)
       nthh=nth+1;
   else
       nthh=1;
   end

    %for the nky terms

  %-----------------------------------------------------------------------
        %This is for U1 matrix in the draft

  %-----------------------------------------------------------------------

                g11=dot(gap(:,7,nth,nky),gap(:,7,nthh,nky));
                g12=dot(gap(:,7,nth,nky),gap(:,8,nthh,nky));
                g13=dot(gap(:,7,nth,nky),gap(:,9,nthh,nky));
                g14=dot(gap(:,7,nth,nky),gap(:,10,nthh,nky));
                g15=dot(gap(:,7,nth,nky),gap(:,11,nthh,nky));
                g16=dot(gap(:,7,nth,nky),gap(:,12,nthh,nky));

                g21=dot(gap(:,8,nth,nky),gap(:,7,nthh,nky));
                g22=dot(gap(:,8,nth,nky),gap(:,8,nthh,nky));
                g23=dot(gap(:,8,nth,nky),gap(:,9,nthh,nky));
                g24=dot(gap(:,8,nth,nky),gap(:,10,nthh,nky));
                g25=dot(gap(:,8,nth,nky),gap(:,11,nthh,nky));
                g26=dot(gap(:,8,nth,nky),gap(:,12,nthh,nky));

                g31=dot(gap(:,9,nth,nky),gap(:,7,nthh,nky));
                g32=dot(gap(:,9,nth,nky),gap(:,8,nthh,nky));
                g33=dot(gap(:,9,nth,nky),gap(:,9,nthh,nky));
                g34=dot(gap(:,9,nth,nky),gap(:,10,nthh,nky));
                g35=dot(gap(:,9,nth,nky),gap(:,11,nthh,nky));
                g36=dot(gap(:,9,nth,nky),gap(:,12,nthh,nky));

                g41=dot(gap(:,10,nth,nky),gap(:,7,nthh,nky));
                g42=dot(gap(:,10,nth,nky),gap(:,8,nthh,nky));
                g43=dot(gap(:,10,nth,nky),gap(:,9,nthh,nky));
                g44=dot(gap(:,10,nth,nky),gap(:,10,nthh,nky));
                g45=dot(gap(:,10,nth,nky),gap(:,11,nthh,nky));
```

```
                g46=dot(gap(:,10,nth,nky),gap(:,12,nthh,nky));

                g51=dot(gap(:,11,nth,nky),gap(:,7,nthh,nky));
                g52=dot(gap(:,11,nth,nky),gap(:,8,nthh,nky));
                g53=dot(gap(:,11,nth,nky),gap(:,9,nthh,nky));
                g54=dot(gap(:,11,nth,nky),gap(:,10,nthh,nky));
                g55=dot(gap(:,11,nth,nky),gap(:,11,nthh,nky));
                g56=dot(gap(:,11,nth,nky),gap(:,12,nthh,nky));

                g61=dot(gap(:,12,nth,nky),gap(:,7,nthh,nky));
                g62=dot(gap(:,12,nth,nky),gap(:,8,nthh,nky));
                g63=dot(gap(:,12,nth,nky),gap(:,9,nthh,nky));
                g64=dot(gap(:,12,nth,nky),gap(:,10,nthh,nky));
                g65=dot(gap(:,12,nth,nky),gap(:,11,nthh,nky));
                g66=dot(gap(:,12,nth,nky),gap(:,12,nthh,nky));

                S1=det([g11 g12 g13 g14 g15 g16;
                        g21 g22 g23 g24 g25 g26;
                        g31 g32 g33 g34 g35 g36;
                        g41 g42 g43 g44 g45 g46;
                        g51 g52 g53 g54 g55 g56;
                        g61 g62 g63 g64 g65 g66]);


                   U1(nth,nky)=S1/abs(S1);

 end
end

%initialising U2
U2=zeros(2*Ntheta+1,2*Nk+1);

for nth=1:2*Ntheta+1
 for nky=1:2*Nk+1
%-----------------------------------------------------------------
        %for nth terms
        %---------------------------------------------------------
        % for U_2 from the Draft
        %---------------------------------------------------------


   if (nky < 2*Nk+1)
       nkyy=nky+1;
   else
       nkyy=1;
   end

                i11=dot(gap(:,7,nth,nky),gap(:,7,nth,nkyy));
                i12=dot(gap(:,7,nth,nky),gap(:,8,nth,nkyy));
                i13=dot(gap(:,7,nth,nky),gap(:,9,nth,nkyy));
                i14=dot(gap(:,7,nth,nky),gap(:,10,nth,nkyy));
                i15=dot(gap(:,7,nth,nky),gap(:,11,nth,nkyy));
                i16=dot(gap(:,7,nth,nky),gap(:,12,nth,nkyy));
```

```
                        i21=dot(gap(:,8,nth,nky),gap(:,7,nth,nkyy));
                        i22=dot(gap(:,8,nth,nky),gap(:,8,nth,nkyy));
                        i23=dot(gap(:,8,nth,nky),gap(:,9,nth,nkyy));
                        i24=dot(gap(:,8,nth,nky),gap(:,10,nth,nkyy));
                        i25=dot(gap(:,8,nth,nky),gap(:,11,nth,nkyy));
                        i26=dot(gap(:,8,nth,nky),gap(:,12,nth,nkyy));

                        i31=dot(gap(:,9,nth,nky),gap(:,7,nth,nkyy));
                        i32=dot(gap(:,9,nth,nky),gap(:,8,nth,nkyy));
                        i33=dot(gap(:,9,nth,nky),gap(:,9,nth,nkyy));
                        i34=dot(gap(:,9,nth,nky),gap(:,10,nth,nkyy));
                        i35=dot(gap(:,9,nth,nky),gap(:,11,nth,nkyy));
                        i36=dot(gap(:,9,nth,nky),gap(:,12,nth,nkyy));

                        i41=dot(gap(:,10,nth,nky),gap(:,7,nth,nkyy));
                        i42=dot(gap(:,10,nth,nky),gap(:,8,nth,nkyy));
                        i43=dot(gap(:,10,nth,nky),gap(:,9,nth,nkyy));
                        i44=dot(gap(:,10,nth,nky),gap(:,10,nth,nkyy));
                        i45=dot(gap(:,10,nth,nky),gap(:,11,nth,nkyy));
                        i46=dot(gap(:,10,nth,nky),gap(:,12,nth,nkyy));

                        i51=dot(gap(:,11,nth,nky),gap(:,7,nth,nkyy));
                        i52=dot(gap(:,11,nth,nky),gap(:,8,nth,nkyy));
                        i53=dot(gap(:,11,nth,nky),gap(:,9,nth,nkyy));
                        i54=dot(gap(:,11,nth,nky),gap(:,10,nth,nkyy));
                        i55=dot(gap(:,11,nth,nky),gap(:,11,nth,nkyy));
                        i56=dot(gap(:,11,nth,nky),gap(:,12,nth,nkyy));

                        i61=dot(gap(:,12,nth,nky),gap(:,7,nth,nkyy));
                        i62=dot(gap(:,12,nth,nky),gap(:,8,nth,nkyy));
                        i63=dot(gap(:,12,nth,nky),gap(:,9,nth,nkyy));
                        i64=dot(gap(:,12,nth,nky),gap(:,10,nth,nkyy));
                        i65=dot(gap(:,12,nth,nky),gap(:,11,nth,nkyy));
                        i66=dot(gap(:,12,nth,nky),gap(:,12,nth,nkyy));

                        S2=det([i11 i12 i13 i14 i15 i16;
                                i21 i22 i23 i24 i25 i26;
                                i31 i32 i33 i34 i35 i36;
                                i41 i42 i43 i44 i45 i46;
                                i51 i52 i53 i54 i55 i56;
                                i61 i62 i63 i64 i65 i66]);

                            U2(nth,nky)=S2/abs(S2);
 end
end
size(gap);

% Initialising the Field strength
F=zeros(2*Ntheta+1,2*Nk+1);
%
 for nth=1:2*Ntheta+1
  for nky=1:2*Nk+1

      if (nth < 2*Ntheta+1)
```

```
          nthh=nth+1;
        else
          nthh=1;
         end

         if(nky < 2*Nk+1)
          nkyy=nky+1;
       else
          nkyy=1;
        end

        % F(nth,nky)=U1(nth,nkyy)*U2(nth,nky)
        F(nth,nky)=log(U1(nth,nky)*U2(nthh,nky)/
(U1(nth,nkyy)*U2(nth,nky)));
      end
    end

    mu=sum(sum(F)); %summing the  stregnth

    mu=real(mu/(4.0*pi*1j));
% We calculate the Z_2 invariant only when the system is gapped
if (gapabs > 0.01)
        mu=mu;
 else
        mu=0.0;
end
```

Printing the Z_2 invariant and the gap

```
fprintf('The value of Z_2 invariant is %d\n', mu);
fprintf('The value of gap is %d\n', gapabs);
```

Published with MATLAB® R2015b