This code calculates QSH state using "the counting" of edge state

```
        This is works for finite spin-mixing as well zero spin mixing and any
        \alpha= p/q valuve

clear all;
clc;

disp('Band structure for edge state counting......')
%plot 1
Nk=100;

% Initialization of the Model parameter
lambda =1.0;
count=1;
tx=1.0;
ty=1.0;
gama=0.25;
n=101; % number of site along open boundary condition choose it to be
 N*q-1 where N= some integer.
nn=0.0;
ntot=1;
U=0.0;
mu=0.0;
mu=mu-U*ntot/2;
% initialise the matrix
sigmaup=zeros(n,n);
sigmadw=zeros(n,n);
sigmaud=zeros(n,n);
sigmadu=zeros(n,n);
%full matrix
comp=zeros(2*n,2*n);
alpha=6;

for ky =0.0-0.2:pi/Nk:2.0*pi+0.2

%for ky =-pi:pi/Nk:pi



 %spin up terms

 %diagonal entries

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

```
 sigmaup(q,p)=-ty*(exp(1i*ky)*exp(1i*2.0*(x)*pi/
alpha)+exp(-1i*ky)*exp(-1i*2.0*(x)*pi/alpha))+(-1)^x*lambda+mu
+(-1)^y*U*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

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


 sigmadw(q,p)=-ty*(exp(1i*ky)*exp(-1i*2.0*(x)*pi/
alpha)+exp(-1i*ky)*exp(1i*2.0*(x)*pi/alpha))+(-1)^x*lambda+mu
+(-1)^y*U*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
 %sigmadw=conj(sigmaup);
%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
 %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


 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;
 %Ekk=[sigmaup sigmaud; sigmadu sigmadw];
 %Ekk=sigmaup;
  vkk=eig(Ekk);

  % vkk=eig(sigmaup);
   %gkk=eig(sigmadw);

  %h=figure(1);
  %xlabel('ky');
  %ylabel('E')
  Ef=0;
  %axis([0 2*pi -1.0 1.0])
  plot(ky,-vkk, 'or');
  %plot(ky,gkk,'g');
  plot(ky,Ef,'ob');



   hold on




end

 %saveas(h,'dispersion_1.0_gamma_0.25.png')
```

Published with MATLAB® R2015b