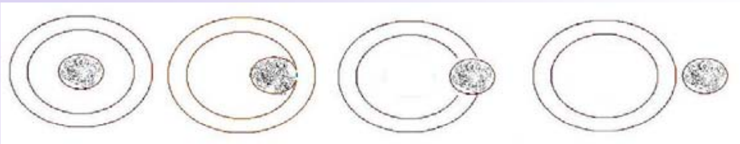
XIIth-IIC-EMTCCM

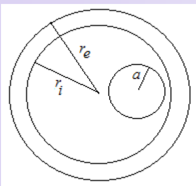
European Master in Theoretical Chemistry and Computational Modelling

Case study: Aharonov-Bom effect

J. Planelles

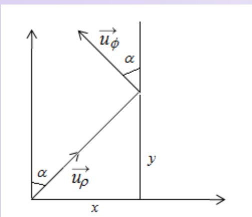
Case study: effects on the energy spectrum of a beam of magnetic flux restricted to a small area inside, outside and piercing a quantum ring.





$$\begin{cases} A_1 = A(0<\rho<a) = \frac{1}{2} B_0\, \rho\, \vec{u}_\phi \\ A_2 = A(a<\rho<\infty) = \frac{1}{2\rho} B_0\, a^2\, \vec{u}_\phi \end{cases}$$

$$A_1(\rho = a) = A_2(\rho = a)$$



$$\vec{u}_\phi = \vec{\imath}\,(-\sin\alpha) + \vec{\jmath}\,(\cos\alpha) = \frac{1}{\rho}(-y, x, 0)$$

$$\begin{cases} A_1 = \frac{B_0}{2}(-y,x,0) \;\rightarrow\; \vec{B} = \nabla \times A = B_0\, \vec{k} \\ A_2 = \frac{B_0\, a^2}{2\, \rho^2}(-y,x,0) \;\rightarrow\; \vec{B} = \nabla \times A = 0 \end{cases}$$

Off-centering $x \rightarrow (x-x_0)$

$$\begin{cases} A_1 = \dfrac{B_0}{2}(-y,(x-x_0),0) \\ A_2 = \dfrac{B_0}{2}\dfrac{a^2}{\rho^2}(-y,(x-x_0),0) \end{cases}$$

$$\rho = \sqrt{(x-x_0)^2+y^2}$$

The Hamiltonian

$$H = \frac{1}{2m}(p-eA)^2 + V$$

a.u. and Coulomb gauge ($\nabla A = 0$)

$$H = \frac{\hat{p}^2}{2m} + \frac{A\cdot\hat{p}}{m} + \frac{A^2}{2m} + V$$

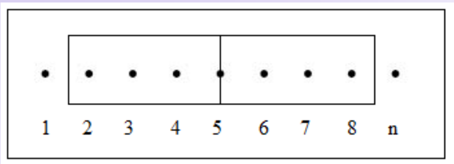
$$A\cdot p = -i\,\frac{B_0}{2}\,f\,(-y\,\partial_x + (x-x_0)\,\partial_y) \quad \begin{cases} f_1 = 1 \\ f_2 = \dfrac{a^2}{(x-x_0)^2+y^2} \end{cases}$$

$$A^2 = \frac{B_0^2}{4}\,sf \quad \begin{cases} sf_1 = (x-x_0)^2+y^2 \\ sf_2 = \dfrac{a^4}{(x-x_0)^2+y^2} \end{cases}$$

$$H = -\frac{1}{2m}(\partial_x^2+\partial_y^2) - i\,\frac{B_0}{2m}\,f\,(-y\,\partial_x + (x-x_0)\,\partial_y) + \frac{B_0^2}{4}\,sf \; + V(x,y)$$

$$H\,F(x,y) = E\,F(x,y)$$

Grid centering



n nodes (n odd)

(n-1) intervals (even)

$$p\,\frac{d^2F}{dx^2} + q\,\frac{dF}{dx} + rF = E\,F \;\leftrightarrow\; \alpha\,F_{i-1} + d\,F_i + a\,F_{i+1} = E\,F_i$$

BCs: $F_1 = F_n = 0$ $\rightarrow$ *i = 2 … (n-1)* $\rightarrow$ *do-loops i = 1 … (n-2)*

Center: $n_0 = (n-1)/2$; *step: h* ; *position* $x_i = (i - n_0)*h$

position x_0 : $n_1 = ROUND\,(n - x_0/L)$; *L = grid length* $n_1 \in N$

Linear System of equations

$$H = -\frac{1}{2m}\left(\partial_x^2 + \partial_y^2\right) - i\,\frac{B_0}{2m}\, f\left(-y\,\partial_x + (x - x_0)\,\partial_y\right) + \frac{B_0^2}{4}\, sf \;\; + V(x,y)$$

$$\partial_x^2\, F = \frac{1}{h^2}\left(F_{i+1,j} + F_{i-1,j} - 2F_{i,j}\right) \qquad \partial_y\, F = \frac{1}{2h}\left(F_{i,j+1} - F_{i,j-1}\right)$$

$$\alpha\, F_{i-1,j} + d\, F_{i,j} + a\, F_{i+1,j} + \beta F_{i,j-1} + bF_{i,j+1} = E\, F_{i,j}$$

$$\alpha \;= -\frac{1}{2\,m\,h^2} - \frac{i\,B_0}{4\,m\,h}\, f_{i-1,j}\, y_j \qquad \beta \;= -\frac{1}{2\,m\,h^2} + \frac{i\,B_0}{4\,m\,h}\, f_{i,j-1}\,(x_i - x_0)$$

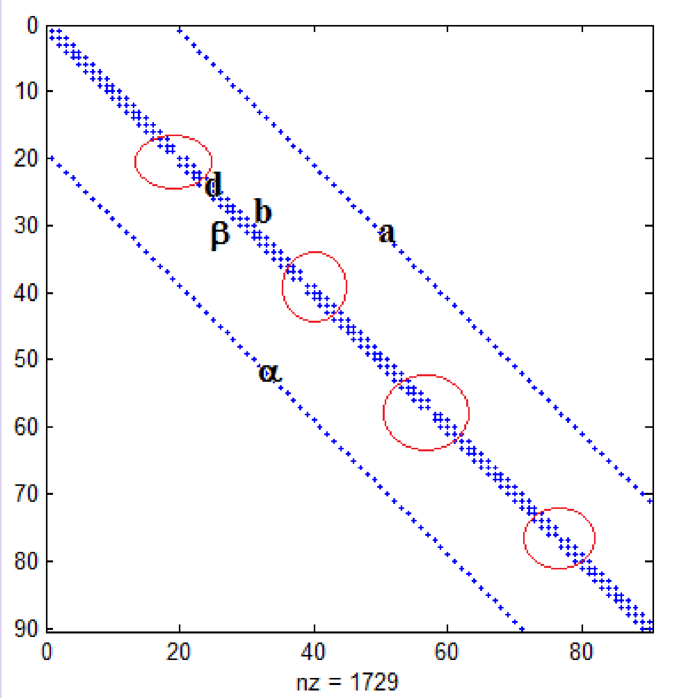
$$a \;= -\frac{1}{2\,m\,h^2} + \frac{i\,B_0}{4\,m\,h}\, f_{i+1,j}\, y_j \qquad b \;= -\frac{1}{2\,m\,h^2} - \frac{i\,B_0}{4\,m\,h}\, f_{i,j+1}\,(x_i - x_0)$$

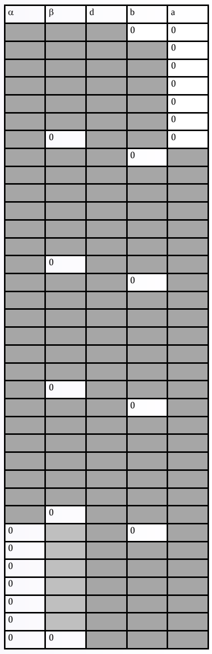
$$d \;= \frac{2}{m\,h^2} + \frac{B_0^2}{8\,m}\, sf_{i,j} + V_{i,j}$$

BCs: $F_{1,j} = F_{n,j} = F_{i,1} = F_{i,n} = 0$

$$\alpha\, F_{i-1,j} + d\, F_{i,j} + a\, F_{i+1,j} + \beta F_{i,j-1} + bF_{i,j+1} = E\, F_{i,j}$$

		α	d	a	β	b
$i = 2$	j=2	$F_{1,2} = 0$	$F_{2,2}$	$F_{3,2}$	$F_{2,1} = 0$	$F_{2,3}$
	j=3	$F_{1,3} = 0$	$F_{2,3}$	$F_{3,3}$	$F_{2,2}$	$F_{2,4}$
	…	…	…	…	…	…
	j=n-1	$F_{1,(n-1)} = 0$	$F_{2,(n-1)}$	$F_{3,(n-1)}$	$F_{2,(n-2)}$	$F_{2,n} = 0$
$i = 3$	j=2	$F_{2,2}$	$F_{3,2}$	$F_{4,2}$	$F_{3,1} = 0$	$F_{3,3}$
	j=3	$F_{2,3}$	$F_{3,3}$	$F_{4,3}$	$F_{3,2}$	$F_{3,4}$
	…	…	…	…	…	…
	j=n-1	$F_{2,(n-1)}$	$F_{3,(n-1)}$	$F_{4,(n-1)}$	$F_{3,(n-2)}$	$F_{3,n} = 0$
	…	…	…	…	…	…
$i = n-1$	j=2	$F_{(n-2),2}$	$F_{(n-1),2}$	$F_{n,2} = 0$	$F_{(n-1),1} = 0$	$F_{(n-1),3}$
	j=3	$F_{(n-2),3}$	$F_{(n-1),3}$	$F_{n,3} = 0$	$F_{(n-1),2}$	$F_{(n-1),4}$
	…	…	…	…	…	…
	j=n-1	$F_{(n-2),(n-1)}$	$F_{(n-1),(n-1)}$	$F_{n,(n-1)} = 0$	$F_{(n-1),(n-2)}$	$F_{(n-1),n} = 0$






```
[di dc ds] =

     1     1     1
     2     2     2
     3     3     3
     4     4     4
     5     5     5
     6     6     6
     7     7     7
     8     8     8
     9     9     9
```

matrix = spdiags ([di,dc,ds] , [-1,0,1], n, n); `matrix =`

```
1  2  0  0  0  0  0  0  0
1  2  3  0  0  0  0  0  0
0  2  3  4  0  0  0  0  0
0  0  3  4  5  0  0  0  0
0  0  0  4  5  6  0  0  0
0  0  0  0  5  6  7  0  0
0  0  0  0  0  6  7  8  0
0  0  0  0  0  0  7  8  9
0  0  0  0  0  0  0  8  9
```

```
[B, d]  =spdiags(matrix)

B =   1     1     0          d =  -1
      2     2     2                0
      3     3     3                1
      4     4     4
      5     5     5
      6     6     6
      7     7     7
      8     8     8
      0     9     9
```

Code matlab/octave

```
% main.m
clear all;
out=fopen('out.tex','a')                            % opening files to write
ban=0.52917706d0;elv=27.21161d0;tesla=235054;       % parameters

inputdata;                                          % input dates file

rin=rin/ban;rad=rad/ban;lon=lon/ban;
ve=ve/elv;vq=vq/elv;                                % convering to a.u.
x0=x0/ban;radbf=radbf/ban;

xx=[];yy=[];

for bf=bfini:bfpas:bfend
  bf
  bf=bf/tesla;
  xx=[xx;bf*tesla];                                 % list of magnetic field

%discretization of the eigenvalue equation

i=sqrt(-1);
h=lon/(n-1);

cons=-1/(2*m*h^2);cons2=i*bf/(4*m*h);
cons3=bf^2/(8*m);cons2e=cons2*radbf^2;              % defining some constants

%building up the sparse matrix

potential_cent;                                     % building up the potential

diags_spm;                                          % building the diagonals of the sparse matrix

% Defining the sparse matrix from its diagonals

M=spdiags([alp bet d b a],[-(n-2),-1,0,1,(n-2)],(n-2)^2,(n-2)^2);

[evec,eval,flag]=eigs(M,neig,'sm');                 % diagonalization, 'sm' small in magnitude

fprintf(out,'%5.3f  ',bf*tesla,sort(diag(real(eval)))*elv*1000);
fprintf(out,' \n ');

yy=[yy;1000*elv*sort(diag(real(eval))')];           % list of eigenvalues for a given magnetic field

end

plot(xx,yy,'k.-')                                   % ploting energies vs. magnetic field

%print -dps fig.ps
fclose(out);
```


```
% main.m

clear all;
out=fopen('out.tex','a')                            % opening files to write
ban=0.52917706d0;elv=27.21161d0;tesla=235054;       % parameters

inputdata;                                          % input dates file

rin=rin/ban;rad=rad/ban;lon=lon/ban;
ve=ve/elv;vq=vq/elv;                                % convering to a.u.
x0=x0/ban;radbf=radbf/ban;

xx=[];yy=[];

for bf=bfini:bfpas:bfend
  bf
  bf=bf/tesla;
  xx=[xx;bf*tesla];                                 % list of magnetic field

%discretization of the eigenvalue equation

i=sqrt(-1);
h=lon/(n-1);

cons=-1/(2*m*h^2);cons2=i*bf/(4*m*h);
cons3=bf^2/(8*m);cons2e=cons2*radbf^2;              % defining some constants
```

```
%building up the sparse matrix

potential_cent;                               % building up the potential

diags_spm;                                    % building the diagonals of the sparse matrix

% Defining the sparse matrix out of its diagonals

M=spdiags([alp bet d b a],[-(n-2),-1,0,1,(n-2)],(n-2)^2,(n-2)^2);

neig=12;                                      % selected number of eigenvalues

[evec,eval,flag]=eigs(M,neig,'sm');           % diagonalization, 'sm' small in magnitud

fprintf(out,'%5.3f  ',bf*tesla,sort(diag(real(eval)))*elv*1000);
fprintf(out,' \n ');

yy=[yy;1000*elv*sort(diag(real(eval))')];     % list of eigenvalues for a given magnetic field

end

plot(xx,yy,'k.-')                             % ploting energies vs. magnetic field

%print -dps fig.ps
fclose(out);
```

inputdata.m

```
%input.m

% 'm' electron mass
% 'rin', 'rad' (A) inner and outer ring radii
% 'lon' (A) length of the discretization grid
% 've', 'vq' (eV) outer and inner ring potential
% 'bfini', 'bfpas', 'bfend' magnetic field loop
% 'x0' (A) center of the flux disk
% 'radbf' (A) radius of the flux disk
% 'neig' number of eigenvalues to be calculated

m=0.067;rin=120;rad=160;lon=400;ve=10.0;vq=0;n=101;
bfini=0;bfpas=5;bfend=250;
x0=140;radbf=50;
neig=12;
```

potential_centered.m

```
%potential_cent.m

pote=ve*ones(n-2,n-2);                   % matrix potential -> external potential
n0=round((n-1)/2);                       % origin position -- n must be odd, then round is not needed
for ii=1:(n-2)
   for jj=1:(n-2)
      aux=sqrt(((ii-n0)^2+(jj-n0)^2))*h;
      if and(aux >= rin,aux < rad)
         pote(ii,jj)=vq;                 % inserting vq potential in the ring region
      end
   end
end
```

diagonals of the sparse matrix diags_spm.m

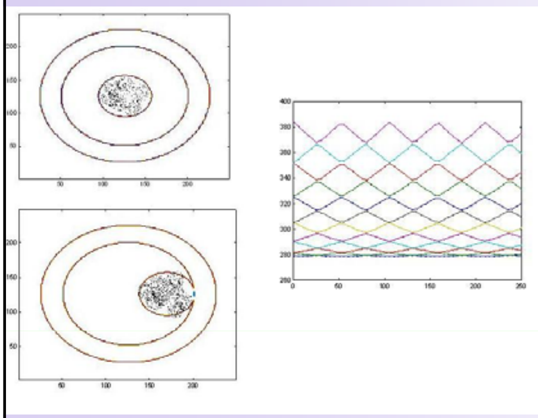
```matlab
%diags_spm.m

% defined in main.m ->  cons=-1/(2*m*h^2);cons2=i*bf/(4*m*h);cons3=bf^2/(8*m);

d=zeros((n-2)^2,1);                                     % filling diagonals with zeros
b=d;bet=d;a=d;alp=d;

n0=round((n-1)/2);                                      % center of the system
n1=round(n*(x0/lon));                                   % center of the magnetic disk

for ii=1:(n-2)
   for jj=1:(n-2)
      kk=(ii-1)*(n-2)+jj;
      aux=sqrt(((ii-n0-n1)^2+(jj-n0)^2))*h;             % distance node - magnetic disk center

      if aux < radbf                                    % filling diagonals region B neq 0
        aux1=cons3*aux^2;

        d(kk)=-4*cons+aux1+pote(ii,jj);
        b(kk)=cons-cons2*(ii-n0-n1)*h;
        bet(kk)=cons+cons2*(ii-n0-n1)*h;
        a(kk)=cons+cons2*(jj-n0)*h;
        alp(kk)=cons-cons2*(jj-n0)*h;

        else                                            % filling diagonals region B = 0
        aux1=cons3*(radbf^4/aux^2);

        d(kk)=-4*cons+aux1+pote(ii,jj);
        b(kk)=cons-cons2e*(ii-n0-n1)*h/aux^2;
        bet(kk)=cons+cons2e*(ii-n0-n1)*h/aux^2;
        a(kk)=cons+cons2e*(jj-n0)*h/aux^2;
        alp(kk)=cons-cons2e*(jj-n0)*h/aux^2;
      end
   end
end

for k=1:(n-2):(n-2)^2                                   % filling with zeros
   b(k)=0;                                              % upper diagonal [0,values]
end

for k=(n-2):(n-2):(n-2)^2                               % filling with zeros
   bet(k)=0;                                            % lower diagonal [values,0]
end

for k=1:(n-2)                                           % filling with zeros
   a(k)=0;                                              % upper diagonal [0,values]
end

for k=(n-2)*(n-2)+1:(n-2)^2                             % filling with zeros
   alp(k)=0;                                            % lower diagonal [values,0]
end

%BCs : F = 0 at the grid borders
```

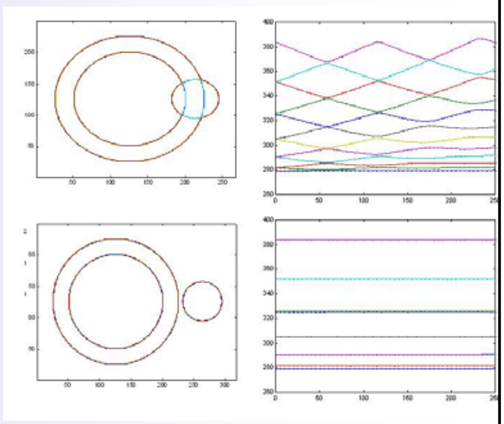
```matlab
%diags_spm.m

% defined in main.m -> cons=-1/(2*m*h^2);cons2=i*bf/(4*m*h);
                     % cons3=bf^2/(8*m);cons2e=cons2*radbf^2;

d=zeros((n-2)^2,1);                             % filling diagonals with zeros
b=d;bet=d;a=d;alp=d;

n0=round((n-1)/2);                              % center of the system
n1=round(n*(x0/lon));                           % center of the magnetic disk
```

```
for ii=1:(n-2)
    for jj=1:(n-2)
        kk=(ii-1)*(n-2)+jj;
        aux=sqrt(((ii-n0-n1)^2+(jj-n0)^2))*h;           % distance node - magnetic disk center

        if aux < radbf                                  % filling diagonals region B neg 0
          aux1=cons3*aux^2;

          d(kk)=-4*cons+aux1+pote(ii,jj);
          b(kk)=cons-cons2*(ii-n0-n1)*h;
          bet(kk)=cons+cons2*(ii-n0-n1)*h;
          a(kk)=cons+cons2*(jj-n0)*h;
          alp(kk)=cons-cons2*(jj-n0)*h;

          else                                          % filling diagonals region B = 0
          aux1=cons3*(radbf^4/aux^2);

          d(kk)=-4*cons+aux1+pote(ii,jj);
          b(kk)=cons-cons2e*(ii-n0-n1)*h/aux^2;
          bet(kk)=cons+cons2e*(ii-n0-n1)*h/aux^2;
          a(kk)=cons+cons2e*(jj-n0)*h/aux^2;
          alp(kk)=cons-cons2e*(jj-n0)*h/aux^2;
        end
    end
end
```

```
for k=1:(n-2):(n-2)^2                              % filling with zeros
    b(k)=0;                                        % upper diagonal [0,values]
end

for k=(n-2):(n-2):(n-2)^2                          % filling with zeros
    bet(k)=0;                                      % lower diagonal [values,0]
end

for k=1:(n-2)                                      % filling with zeros
    a(k)=0;                                        % upper diagonal [0,values]
end

for k=(n-2)*(n-2)+1:(n-2)^2                        % filling with zeros
    alp(k)=0;                                      % lower diagonal [values,0]
end

%BCs : F = 0 at the grid borders
```

The energy spectrum of a single or a many-electron system in a QR (complex topology) can be affected by a magnetic field despite the field strength is null in the region where the electrons are confined

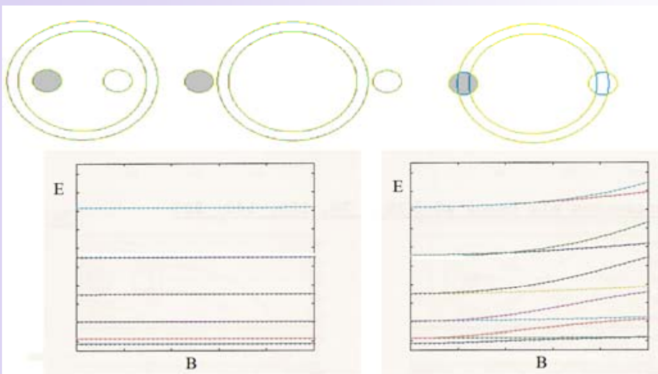




It is not the case for a QD (simple connected topology confining potential)

HomeWork

1. Consider two beams of magnetic field having the same strength but opposite sign. Show that only when the field pierces the ring it has an effect on the ring energy spectrum.



2. Consider now different strength and sign. Show that it is the total magnetic flux what matters