VII. FUTURE WORKS

I have tried to obtain roots for above determinant via symbolic computation tools of MATLAB. But an error occured and running out of time problem iss still unsolved. Though I included this simple MATLAB file as the last file in the appendix. Some topics in the proposed project abstract is not completed. This is due to lack of time and tiring process of coding and debugging codes which takes times up to a month. Introduction of magnetic field to ring problem and making the system time dependent is left for future study. We will continue to complete this study in 2008-2009 spring semester. Progress can be tracked from project website[4].

ACKNOWLEDGEMENTS

I would like to acknowledge two people: R. Onur Umucalilar for useful discussions on numerical methods and tunneling phenomenon and my advisor Dr. Hakioglu for his patience and helpful guidance. Despite all the rush, he devoted great amount of time(which is priceless) to me and this study.

* Electronic address: `ooncel@ug.bilkent.edu.tr`

[1] Introduction to Quantum Mechanics, Prentice-Hall, NJ; 2nd ed - D.J. Griffiths (2004).

[2] Introduction to Scanning Tunneling Microscopy, Oxford University Press, New York - C.J. Chen (1993).

[3] Gravitational Effects On And Of Vacuum Decay - S Coleman, F De Luccia - Physical Review D, (1980).

[4] `http://www.ug.bcc.bilkent.edu.tr/%7Eooncel/seniorf08`

[5] Principles of Quantum Mechanics, Springer; 2nd edition - R. Shankar (1994).

[6] Mechanics, Addison Wesley; 3rd ed - K.R. Symon (1971).

[7] Computer Generated Motion Pictures of One Dimensional Quantum Mechanical Reflection and Transmission - Goldberg, Schey, Schwartz - Am.J. Physics (1966)

[8] Time Dependent Tunelling In One Dimensional Quantum Mechanics, PHYS 491, Senior Project Report I - O.O.Oncel (Fall 2008)

[9] `http://amo.physik.hu-berlin.de/mater/q2_06/au.pdf`

[10] Scientific Computation An Introductory Survey, McGraw-Hill; 2nd ed - Michael T. Heath

[11] `http://ktchu.serendipityresearch.org/download/education/educational_material/quantum_plots/scattering/scattering.m`


```
clear all;
clc

% Propagation of a Gaussian Wavepacket
% Author: Omer Ogul Oncel
% Dept. of Physics, Faculty of Science, Bilkent University, Ankara, Turkey
% Questions and Comments: ooncel@ug.bilkent.edu.tr
% v1.1 - 17.Jan.2009

x_start= -15.005; % x lower limit
x_finish=15.005; % x upper limit

t_start= -6.505; % t lower limit
t_finish=6.500; % t upper limit

k_start= -10.005; % k lower limit
k_finish=10.005; % k upper limit

dk=0.1; % increment step of k
dt=0.1; % increment step of t
dx=0.1; % increment step of x

x=[x_start:dx:x_finish]; % matrix containing x values
t=[t_start:dt:t_finish]; % matrix containing x values
k=[k_start:dk:k_finish]; % matrix containing x values

h_bar=1; % h bar
m=1; % electron mass
```

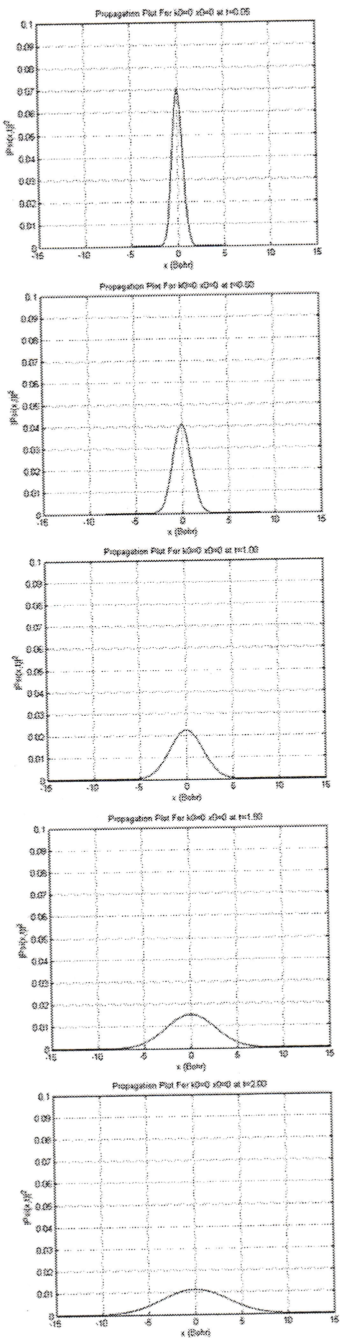

FIG. 5: Gaussian wavefunction propagation with time. x_0=0, k_0=0 Because of zero average momentum, center of gaussian does not change with time. Slides correspond to t=0.05, 0.5, 1, 1.5 and 2 respectively.

```
x0=-7; % initial mean of x
k0=5; % initial mean of k
t0=t_start; % zero time

Dk=0.8; % sigma of k
```

```
E=k.^2/2; % energy of k

for p=1:length(k);
    phikk(p)=( ((pi*Dk^2)^(1/4)) / (sqrt(2*pi)) )*exp( (( -((k(p)-k0).^2)*Dk^2 )/2) -i*k(p)*x0 ); %%shar
end

%normalization
normpsi=1;
psiIA=abs(phikk).^2;
aapp=trapz(psiIA);
normpsi=aapp(1);
phikk=( 1/sqrt(normpsi) )*phikk;
psiIA=abs(phikk).^2;
aapp=trapz(psiIA);

x=[x_start:dx:x_finish]; % matrix containing x values
t=[t_start:dt:t_finish]; % matrix containing x values
k=[k_start:dk:k_finish]; % matrix containing x

psiI=0;

for s=1:length(t)
  for r=1:length(x)
      for p=1:length(k)
        psiI(r,s,p) = (1/sqrt(2*pi))*exp(i*k(p)*x(r)-i*E(p)*(t(s)-t0))*phikk(p)*dk;
      end
      psiI(r,s)=sum(psiI(r,s,:));
  end
end

psiIAA=abs(psiI(r,s)).^2;
aa=trapz(psiIAA);

for s = 1:length(t)
  hold off;
  plot(x,abs(psiI(:,s)).^2,'b');
  hold on;
  axis([x_start x_finish 0 .1]);
  grid on;
  xlabel('x (Bohr)');
  ylabel('|Psi(x,t)|^2');
  title(sprintf('Propagation Plot For k0=%d x0=%d at t=%4.2f ',k0,x0,s*0.05));
  M1(s) = getframe;
end

% Note: For details please check the Report II
% (http://www.ug.bcc.bilkent.edu.tr/%7Eooncel/seniorf08)


clear all;
clc

% Transmission and Reflection of a Gaussian Wavepacket
% 1D Tunneling Simulation
% Author: Omer Ogul Oncel
% Dept. of Physics, Faculty of Science, Bilkent University, Ankara, Turkey
```

```
% Questions and Comments: ooncel@ug.bilkent.edu.tr
% v1.1 - 17.Jan.2009

cpu1=cputime; % start CPU counting

x_start= -15.005; % x lower limit
x_finish=15.005; % x upper limit

t_start= -2.505; % t lower limit
t_finish=2.500; % t upper limit

k_start= 0.005; % k lower limit
k_finish=10.005; % k upper limit

dk=0.1; % increment step of k
dt=0.1; % increment step of t
dx=0.1; % increment step of x
%%
x=[x_start:dx:x_finish]; % matrix containing x values
t=[t_start:dt:t_finish]; % matrix containing x values
k=[k_start:dk:k_finish]; % matrix containing x values

d=1; % barrier width
a=d/2;
h_bar=1; % h bar
m=1; % electron mass
V0=1.3; % barrier potential
x0=-7; % initial mean of x
k0=5; % initial mean of k
t0=t_start; % zero time

Dk=0.8; % sigma of k
E=k.^2/2; % energy of k
for p=1:length(k);
    phik(p)=( ((pi*Dk^2)^(1/4)) / (sqrt(2*pi)) )*exp( (( -((k(p)-k0).^2)*Dk^2 )/2) -i*k(p)*x0 ); %%shank
end
normpsi=1;
psiIA=abs(phik).^2;
aap=trapz(psiIA);
normpsi=aap(1);
phik=( 1/sqrt(normpsi) )*phik;
psiIA=abs(phik).^2;
aap=trapz(psiIA);

x=[x_start:dx:x_finish]; % matrix containing x values
t=[t_start:dt:t_finish]; % matrix containing x values
k=[k_start:dk:k_finish]; % matrix containing x values


%%%%%%%%%%%%%%%%%%%%%%%%%%


k1 = sqrt(2.*m.*E)/h_bar;
K1 = sqrt(2.*m.*(V0-E))/h_bar;
p=i.*K1;
```

```
%%% Compute coefficient for transmitted (E>V0)
T =exp(-2.*i.*k1.*a).*((2.*k1)./p)./( ((2.*k1)./p).*(cos(2.*p.*a))...
-(1+(k1.^2./p.^2)).*i.*sin(2.*p.*a) );

% Compute coefficient for reflected wave (E>V0)
R =exp(-2.*i.*k1.*a).*(i.*sin(2.*p.*a)).*(1-(k1.^2./p.^2))./( ((2.*k1)./p).*(cos(2.*p.*a))...
-(1+(k1.^2./p.^2)).*i.*sin(2.*p.*a) );

%%% Compute coefficient for transmitted (E<V0)
TT =exp(-2.*i.*k1.*a).*((2.*i.*k1)./K1)./( ((2.*i.*k1)./K1).*(cosh(2.*K1.*a))...
-(1-(k1.^2./K1.^2)).*sinh(2.*K1.*a) );

% Compute coefficient for reflected wave (E<V0)
RR =exp(-2.*i.*k1.*a).*(sinh(2.*K1.*a)).*(1+(k1.^2./K1.^2))./(... ((2.*i.*k1)./K1).*(cosh(2.*K1.*a))-(1-

%% Compute coefficient for transmitted wave (E=V0)
TTT= (exp(-2.*i.*k1.*a))./(1-i.*k1.*a);

%% Compute coefficient for reflected wave (E=V0)
RRR= exp(-2.*i.*k1.*a).*(-i.*k1.*a)./( 1-i.*a.*k1 );


%% Below Method calculates general T and R for selecting true transmission
%% and reflection coefficients for each k, according to E & V0 relation:

Trans=0;
Ref=0;

Tt=0;
tT=0;
f=0;

for k=k_start:dk:k_finish

f=f+1;

EE=(((h_bar).^2).*(k.^2))./2;

if EE > V0
    Trans(1,f)=T(1,f);
    Ref(1,f)=R(1,f);
Tt=Tt+1;
end

if EE < V0
    Trans(1,f)=TT(1,f);
    Ref(1,f)=RR(1,f);
tT=tT+1;
end

if EE==V0
    Trans(1,f)=TTT(1,f);
    Ref(1,f)=RRR(1,f);
end
```

```
end

%******


%******

k=[k_start:dk:k_finish];

phikR= phik.*Ref;
phikT= phik.*Trans;

disp('STATUS(1/4): all variables initialized. evaluating psiI(x,t) integral');
for s=1:length(t)
  for r=1:length(x)
    for p=1:length(k)
        psiI(r,s,p) = (1/sqrt(2*pi))*exp(i*k(p)*x(r)-i*E(p)*(t(s)-t0))*phik(p)*dk;
    end
    psiI(r,s)=sum(psiI(r,s,:));
  end
end

disp('STATUS(2/4): evaluating psiR(x,t) integral');
x_before= [x_start:dx:0]; %[1]
psiR = zeros(length(x_before),length(t));
for s=1:length(t)
  for r=1:length(x_before)
      for p=1:length(k)
        psiR(r,s,p) = exp(-i*k(p)*x_before(r)-i*E(p)*(t(s)-t0))*phikR(p)*dk;
      end
      psiR(r,s)=sum(psiR(r,s,:));
  end
end

psiB = psiI(1:(length(psiR)-(length(psiR)-size(psiR))),:,:)+psiR; %[1]

x_after= [1:dx:x_finish]; %[1]
disp('STATUS(3/4): evaluating psiT(x,t) integral');
psiT = zeros(length(x_after),length(t));
for s=1:length(t)
  for r=1:length(x_after)
      for p=1:length(k)
        psiT(r,s,p) = exp(i*k(p)*x_after(r)-i*E(p)*(t(s)-t0))*phikT(p)*dk;
      end
      psiT(r,s)=sum(psiT(r,s,:));
  end
end
cpu2=cputime; % finish CPU count
cpu=cpu2-cpu1; % net CPU time elapsed
disp(sprintf('Net CPU Time Elapsed %6.6f:',cpu));
disp('STATUS(4/4): Plotting..');
hparameter=.2; % maximum y value of plot
for s = 1:length(t)
  hold off;
  plot(x_before,abs(psiB(:,s)).^2,'k'); %[1]
```

```
  hold on;
  plot(x_after,abs(psiT(:,s)).^2,'r'); %[1]
  axis([x_start x_finish 0 hparameter]);
  rectangle('Position',[0,0,2*a,hparameter/2]);
  grid on;
  xlabel('x (Bohr)');
  ylabel('|Psi(x,t)|^2');
  title(sprintf('Tunneling Plot For k0=%d x0=%d V0=%4.2f at t=%4.2f ',k0,x0,V0,s*0.05));
  mov(s) = getframe;
end

% References
% [1] Kevin Chu's Code: http://ktchu.serendipityresearch.org/download/education
%                       /educational_material/quantum_plots/scattering/scattering.m
%
% Note: For details please check the Report II (http://www.ug.bcc.bilkent.edu.tr/%7Eooncel/seniorf08)


clear all;

% Graphical Root Finding for Determinant
% Author: Omer Ogul Oncel
% Dept. of Physics, Faculty of Science, Bilkent University, Ankara, Turkey
% Questions and Comments: ooncel@ug.bilkent.edu.tr
% v1.0 - 17.Jan.2009
% E>V0

%%%%%%

d=1; % barrier width
a=d/2;
h_bar=1; % h bar
m=1; % electron mass
V0=1.3; % barrier potential


%% Incident From Left

L=100; % circumference
syms k1;
K1=sqrt(2*m*((k1^2/2)-V0))/h_bar;
p=i.*K1;
%%% Compute coefficient for transmitted (E>V0)
T_A_L =exp(-2.*i.*k1.*a).*((2.*k1)./p)./( ((2.*k1)./p).*(cos(2.*p.*a))...
-(1+(k1.^2./p.^2)).*i.*sin(2.*p.*a) );

% Compute coefficient for reflected wave (E>V0)
R_A_L =exp(-2.*i.*k1.*a).*(i.*sin(2.*p.*a)).*(1-(k1.^2./p.^2))./(... ((2.*k1)./p).*(cos(2.*p.*a))-(1+(k1

%%% Compute coefficient for transmitted (E<V0)
T_B_L =exp(-2.*i.*k1.*a).*((2.*i.*k1)./K1)./( ((2.*i.*k1)./K1).*(cosh(2.*K1.*a))...
-(1-(k1.^2./K1.^2)).*sinh(2.*K1.*a) );

% Compute coefficient for reflected wave (E<V0)
R_B_L =exp(-2.*i.*k1.*a).*(sinh(2.*K1.*a)).*(1+(k1.^2./K1.^2))./(... ((2.*i.*k1)./K1).*(cosh(2.*K1.*a))-
```

```
%% Compute coefficient for transmitted wave (E=V0)
T_E_L= (exp(-2.*i.*k1.*a))./(1-i.*k1.*a);

%% Compute coefficient for reflected wave (E=V0)
R_E_L= (exp(-2.*i.*k1.*a)).*(-i.*k1.*a)./( 1-i.*a.*k1 );



%% Incident From Right (a --> -a)


%%% Compute coefficient for transmitted (E>V0)
T_A_R =exp(2.*i.*k1.*a).*((2.*k1)./p)./( ((2.*k1)./p).*(cos(-2.*p.*a))...
-(1+(k1.^2./p.^2)).*i.*sin(-2.*p.*a) );

% Compute coefficient for reflected wave (E>V0)
R_A_R =exp(2.*i.*k1.*a).*(i.*sin(-2.*p.*a)).*(1-(k1.^2./p.^2))./(... ((2.*k1)./p).*(cos(-2.*p.*a))-(1+(k

%%% Compute coefficient for transmitted (E<V0)
T_B_R =exp(2.*i.*k1.*a).*((2.*i.*k1)./K1)./( ((2.*i.*k1)./K1).*(cosh(-2.*K1.*a))...
-(1-(k1.^2./K1.^2)).*sinh(-2.*K1.*a) );

% Compute coefficient for reflected wave (E<V0)
R_B_R =exp(2.*i.*k1.*a).*(sinh(-2.*K1.*a)).*(1+(k1.^2./K1.^2))./(... ((2.*i.*k1)./K1).*(cosh(-2.*K1.*a))

%% Compute coefficient for transmitted wave (E=V0)
T_E_R= (exp(2.*i.*k1.*a))./(1-i.*k1.*(-a));

%% Compute coefficient for reflected wave (E=V0)
R_E_R= exp(2.*i.*k1.*a).*(i.*k1.*a)./( 1-i.*(-a).*k1 );

S=sym(zeros(2,2)); % S Matrix
M=sym(zeros(2,2)); % T Matrix

%%%% FOR E>V0

S(1,1)=R_A_L;
S(1,2)=T_A_R;
S(2,1)=T_A_L;
S(2,2)=R_A_R;


M(1,1)= S(2,1)-(S(1,1)*S(2,2))/S(1,2);
M(1,2)= S(2,2)/S(1,2);
M(2,1)= -S(1,1)/S(1,2);
M(2,2)= 1/S(1,2);

U(1,1)=exp(-i*k1*(L-a));
U(1,2)=0;
U(2,1)=0;
U(2,2)=exp(i*k1*(L-a));
A=det(M-U);
for c=1:.01:10

    B=abs(subs(A,'k1',c));
    plot(c,B)
```

```
    hold on
end
for n=0:1:5
ktheoretical=2*pi*n/L
end
```

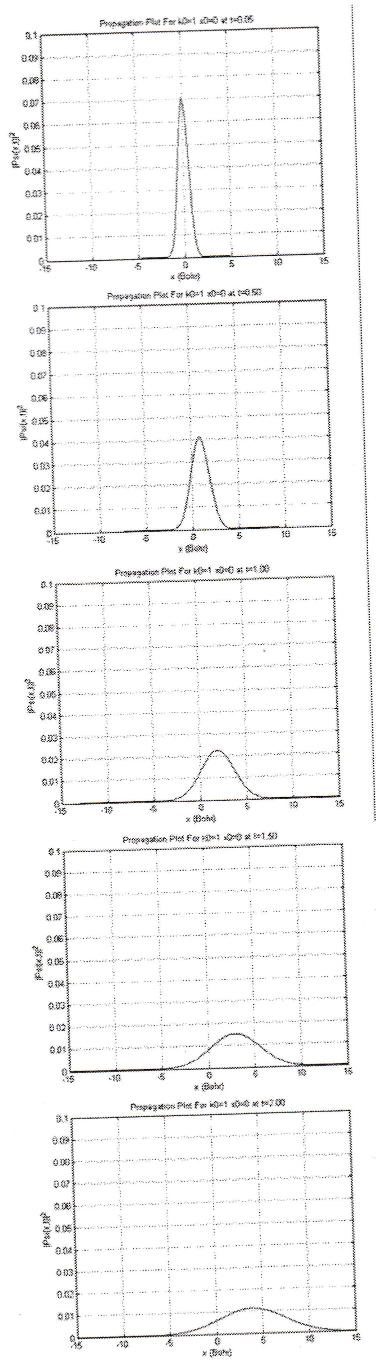

FIG. 6: Gaussian wavefunction propagation with time. x_0=0, k_0=1 This time average momentum is not zero, center of gaussian travels with proceeding time. Slides correspond to t=0.05, 0.5, 1, 1.5 and 2 respectively.

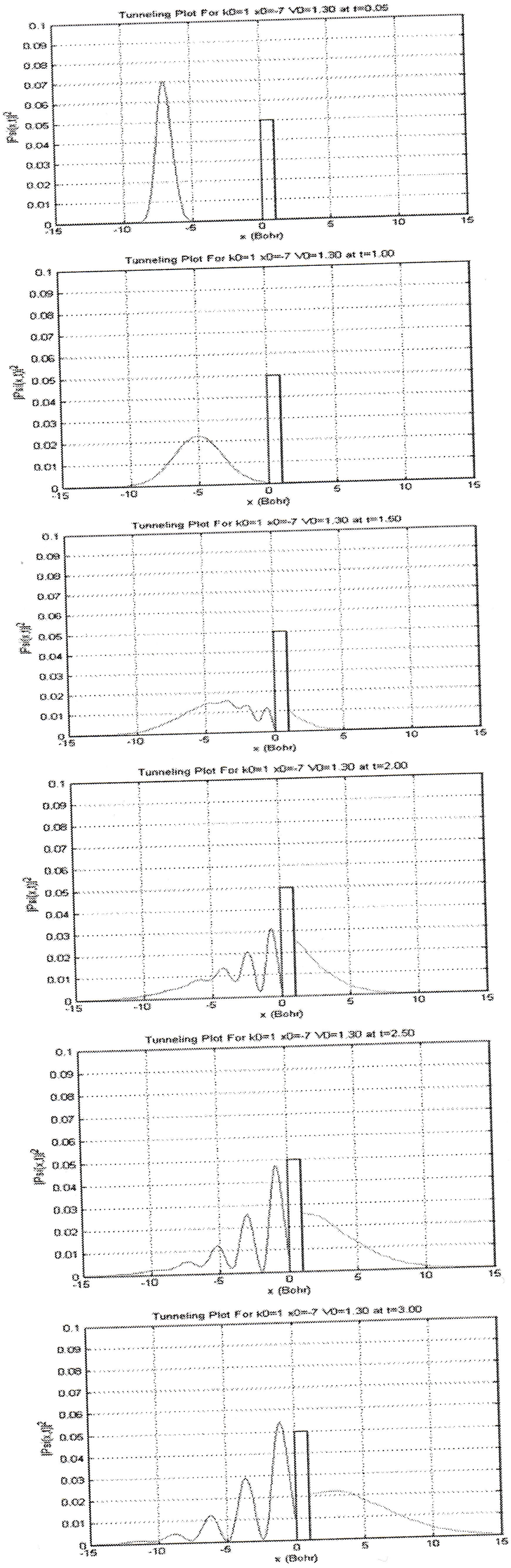

FIG. 7: Gaussian wavefunction tunneling with $x_0 = -7$ and $k_0 = 1$. Slides correspond to t=0.05, 0.5, 1, 1.5, 2, 2.5 and 3 respectively.

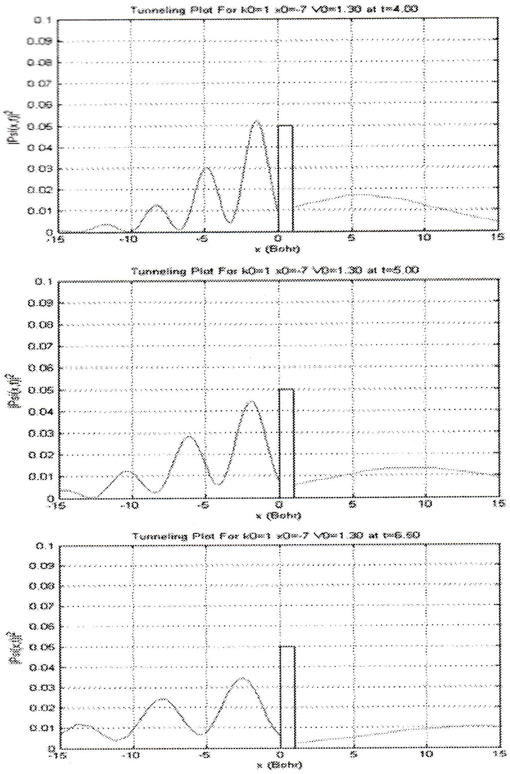

FIG. 8: Continuation of previous plots. Slides correspond to t=4, 5 and 6,5 respectively.

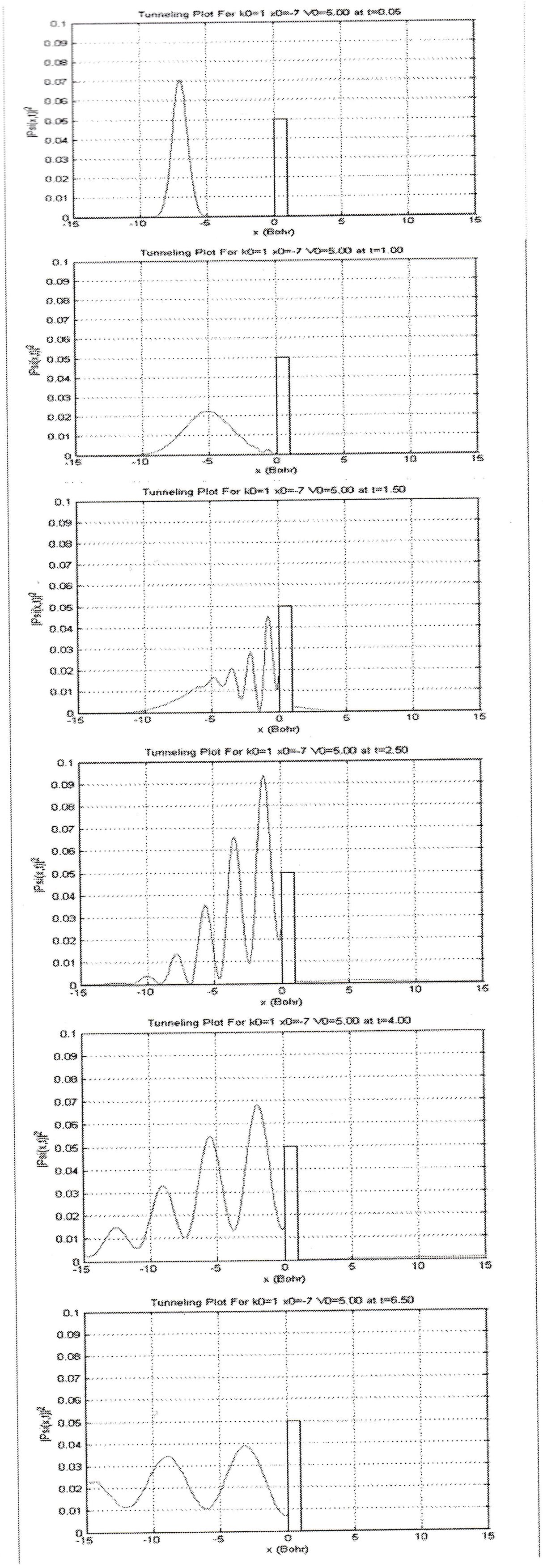

FIG. 9: Gaussian tunneling with $x_0 = -7$ and $k_0 = 1$ but $V_0 = 5$. Slides correspond to t=0.05, 1, 2.5, 4, 6.5 respectively

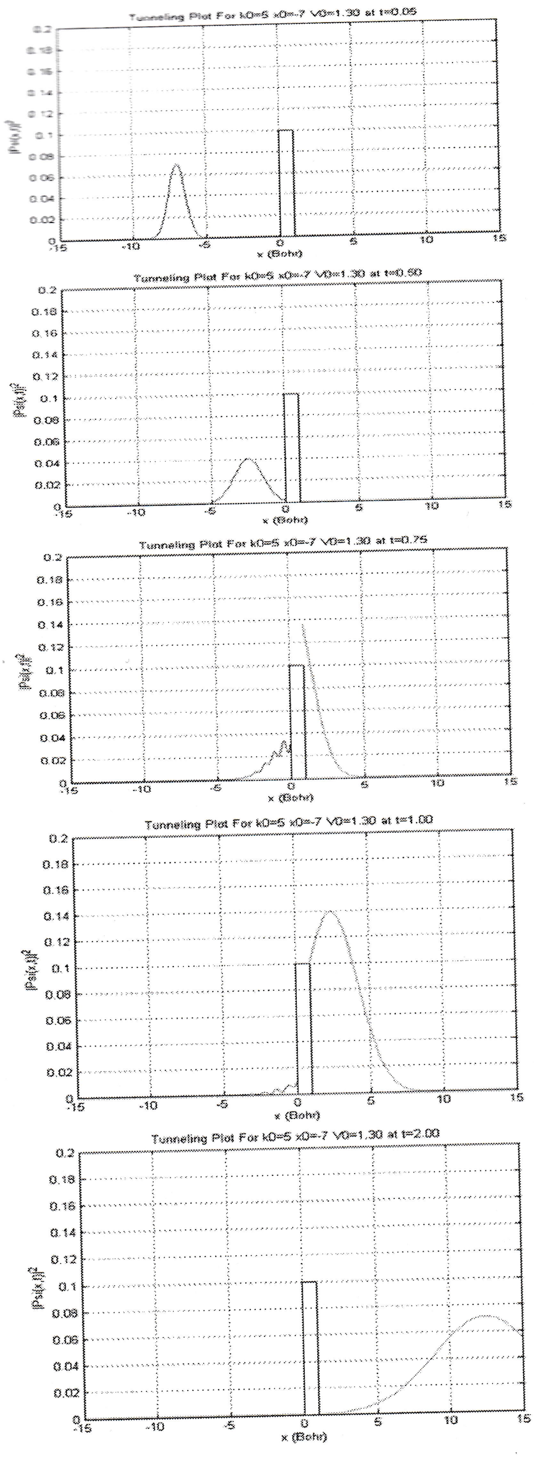

FIG. 10: Gaussian tunneling with $x_0 = -7$ but $k_0 = 5$ with $V_0 = 1.3$. Slides correspond to 0.05, 05, 0.75, 1, 2