ภาคผนวก 1

โปรแกรมกลุ่มคลื่น

```
function wpacket(Command)
%
%     WPACKET is a GUI-INTERFACE for MATLAB 5
%     it solves the 1-dimensional Schr๖dinger equation for various
%     Gaussian wave
%     packets and potential profiles. The routine calls the %
%     following files:
%
%     GSSPCKT.M, which generates a Gaussian wave-packet
%     PTNTLS.M, this provides a collection af various energy
%     potential profiles
%     SCHROEQ.DLL, this is the core routine that performs the time
%     evolution of any
%     given wave amplitude
%     SCHROEQ.M, see this for information on how the routine
%     SCHROEQ.DLL works
%
%     Input "help <filename> from MATLAB command line for
%     explanations on
%     the usage of the above functions.
%
%     The program can be enriched by adding other potential
%     profiles in PTNTLS.M
%     and corresponding items in the string potString in this
%     routine.


global Psi x xWidth runFlag Comment V xo Dx

if nargin < 1,
       Command = 'Initialise';
      figure(1)
      clf
   set(1, 'Name', ['Schr๖dinger wave packets in 1dimension',...
             '  - ']);
      set(1, 'Userdata', zeros(size(1:30)));
 end
```

```
if ~strcmp(Command, 'Initialise'),

   handles=get(1, 'Userdata'); % retrieve data saved at the end of
          %the 'Initialize' call; used by all other calls:
   H_AX=handles(1); % save data in the Userdata buffer of the window
   H_LV=handles(2);
   H_LP=handles(3);
   H_LR=handles(4);
   H_LI=handles(5);
   H_EkTxt=handles(6);
   H_LEav=handles(7);
   H_TXT=handles(8);
   H_TXT0=handles(9);
      H_POP=handles(10);
   H_FIG=handles(11);
   H_TEav=handles(12);
      H_EavTxt=handles(13);
      relMass=handles(14);
   potFlag=handles(15);
      Ko=handles(16);
   Dx=handles(17);
   H_SL=handles(18);
   H_ED=handles(19);
      Eav=handles(20);
      Ek=handles(21);
   lambda=handles(22);
   H_RUN=handles(23);
   H_STOP=handles(24);
   H_REST=handles(25);
   H_LamTxt=handles(26);
   H_DxTxt=handles(27);
   H_WED = handles(28);
   H_WSL = handles(29);


end

if strcmp(Command, 'Initialise'),

      %Default initialisation:
 %handles=zeros(1,36);
```

```
      relMass=1; % as default value in gsspckt.m
      xWidth=40;  % as default value in gsspckt.m
      %pNum=1001;  % as default value in gsspckt.m
   pNum=801;
   Ko=0;    %initially the particle is at rest

      N=(pNum-1)/2;
      x=xWidth*(-N:N)/pNum;
      x=x(:);

      [V, xo, Dx, Comment]=ptntls(x,0);

      [Psi,x,Ek,lambda]= gsspckt(xo, Dx, Ko, relMass, xWidth, pNum);


      Eav = Ek+sum(V.*(abs(Psi).^2));  % averaged state energy

   potFlag=0;
   runFlag=0;

%----------------USER INTERFACE CONTROLS ------------------------------
            H_FIG=1;

            H_RUN=uicontrol(1, 'style', 'pushbutton',...
         'Units', 'normalized',...
         'Position', [0.82,0.6,0.08,0.04],... % run button
                  'string', 'Run',...
                  'callback', 'wpacket(''Run'');');

            H_STOP=uicontrol(1, 'style', 'Pushbutton',...
         'Units', 'normalized',...
         'Position', [0.91,0.6,0.08,0.04],... % stop button
         'string', 'Stop',...
         'Enable', 'off', ...
         'callback','wpacket(''Stop'');');

         H_REST=uicontrol(1,'Style','PushButton',...
%        restart button
         'Units', 'normalized',...
         'Position', [0.82,0.55,0.08,0.04],...
         'string', 'Restart',...
         'Callback','wpacket(''Rest'');',...
         'Enable', 'off');

      txtEdPos=[0.0,0.0,0.3,0.04];
      H_TXT = uicontrol(1,...
                         'Style', 'text',...
             'Units', 'normalized',...
             'Position', txtEdPos,...
             'FontSize', 10,...
             'String', ['Mean wave-number [1/',197,'] = '],...
             'Backgroundcolor', [0.8,0.8,0.8],...
             'HorizontalAlignment', 'right');

       Ext=get(H_TXT, 'Extent');

       edPos=[0.01+Ext(3),0.0,  0.05,0.04];

       H_ED =uicontrol(1,...
%      Edit window for wave-number width control
             'Style', 'edit',...
             'Units', 'normalized',...
             'Position', edPos,...
             'String', '0',...
             'HorizontalAlignment', 'center',...
             'Callback','wpacket(''WNumberEditChanged'');',...
             'Backgroundcolor', [1,1,1]);

            slPos=[Ext(3)+0.06,  0.0,  0.15,  0.04];

      H_SL=uicontrol(1,'Style', 'slide',...
%     Slider for wave-number control
             'Min', 0,...
             'Max', 10,...
             'Value', 0,...
             'Callback', 'wpacket(''WNumberSliderMoved'');',...
             'Units', 'normalized',...
             'Position', slPos);

      txtWPos=[0.51,0.0,0.3,0.04];

      H_WTXT = uicontrol(1,...
                         'Style', 'text',...
             'Units', 'normalized',...
             'Position', txtWPos,...
             'FontSize', 10,...
             'String', ['Gaussian semi-width [',197,'] = '],...
             'Backgroundcolor', [0.8,0.8,0.8],...
             'HorizontalAlignment', 'right');
       WExt=get(H_TXT, 'Extent');

       edWPos=[0.51+Ext(3),0.0, 0.07,0.04];

       H_WED =uicontrol(1,...
%      Edit window for Gaussian width control
             'Style', 'edit',...
             'Units', 'normalized',...
             'Position', edWPos,...
             'String', '0.900',...
             'HorizontalAlignment', 'center',...
             'Callback','wpacket(''GWidthEditChanged'');',...
             'Backgroundcolor', [1,1,1]);

            slWPos=[0.51+WExt(3)+0.07, 0.0, 0.12, 0.04];

      H_WSL=uicontrol(1,'Style', 'slide',...
%     Slider for Gaussian width control
             'Min', 0.20,...
             'Max', 10.0,...
             'Value', 0.90,...
             'Callback', 'wpacket(''GWidthSliderMoved'');',...
             'Units', 'normalized',...
             'Position', slWPos);

%-------------DATA PLOTTING ----------------------------
            uicontrol(1, 'style', 'text',...
         'string','Potential type', ...
         'Units', 'normalized',...
         'Position', [0.8, 0.94, 0.18, 0.03],...
         'Background', [0.8, 0.8, 0.8]);

      potString = ['Zero potential well|High narrow barrier|Small
barrier|',...
         'Medium barrier|Wide barrier|Large barrier|',...
         'Deep well|Small well|'...
         'Medium well|Wide well|Large well|',...
         'Positive linear|Negative linear|',...
         'Positive periodic|Negative periodic|',...
         'Harmonic oscillator|Harm. narrow packet|',...
         'Harm. wide packet|Quadric symmetric|',...
         'Quadric tilted|Quadric strongly tilted|Special',...
         'Gaussian random'];
      H_POP = uicontrol(1, 'style', 'pop','String',potString,...
      'Callback', 'wpacket(''Choose_pot'')',...
         'Value', 1,...
      'Units', 'normalized',...
         'Position', [0.8, 0.9, 0.2, 0.03],...
         'Background', 'w',...
         'Enable', 'on');

      H_AX = axes('Units', 'normalized',...
      'Position', [0.075, 0.15, 0.725, 0.78], ...
      'XLim',[-xWidth/2,xWidth/2], 'XLimMode','manual',...
      'YLim',[-200,400],'YLimMode','manual', 'Box', 'on');

      text(0.75, 0.15, '- Probability density',...
         'Unit', 'Normalized',...
         'Color','k', 'FontName', 'Arial',...
         'FontSize', 8);


      text(0.75, 0.1, '- Real part',...
         'Unit', 'Normalized',...
      'Color',[1,0.5,1], 'FontName', 'Arial',...
         'FontSize', 8);

      text(0.75, 0.05, '- Imaginary part',...
            'Unit', 'Normalized',...
      'Color',[0, 0.8, 0.8], 'FontName', 'Arial',...
         'FontSize', 8);

   xlabel('X-axis position (Angstrom)');
      title('Potential Energy Profile in eV (thick blue line) &
Wave Forms ');

      H_LV =line('XData', x, 'Ydata', V, 'Color',...
      'b', 'LineWidth', 3, 'EraseMode', 'xor');


      H_TEav=text(-19, Eav+16,'<E>',...
      'EraseMode', 'background','Color', 'r',...
      'FontName', 'Arial','FontSize', 9);

      H_LP=line('XData', x,'YData',zeros(size(x)), 'Color',
'k','EraseMode', 'xor');

      H_LR=line('XData', x, 'YData', zeros(size(x)),...
            'Color', 'c','EraseMode', 'xor');
      H_LI=line('XData', x, 'YData', zeros(size(x)),...
            'Color', [1, 0.75, 1],'EraseMode', 'xor');


      H_LEav=line('XData', [-20,20], 'YData', [Eav,Eav],...
            'Color', 'r','EraseMode', 'xor');


   H_TXT0=text(0.02,0.05,'Time [fsec] = ','Units', 'normalized',...
                  'Horizontalalignment', 'left',...
                  'EraseMode', 'xor',...
                  'FontName', 'Arial',...
                  'FontSize', 8);

      Extent=get(H_TXT0, 'Extent');
      H_TXT =     text(0.02+Extent(3),0.05,'0.00','EraseMode', 'xor',...
            'Horizontalalignment', 'left',...
                  'Units', 'normalized',...
            'FontName', 'Arial',...
            'FontSize', 8);

   H_ASMALL = axes('Units', 'normalized',...
         'Position', [0.805, 0.15, 0.195, 0.2], ...
         'Box', 'on',...
         'XLim',[0,1], ...
         'YLim',[0,1],...
         'FontName', 'Arial',...
         'FontSize', 9,...
         'YTickMode', 'manual',...
         'XTickMode', 'manual');

  title('Parameters at t=0');

  H_EavTxt= text(0.02, 0.85, ['<E> = ',num2str(Eav),' eV'], ...
         'FontName', 'Arial',...
         'FontSize', 9);


  H_EkTxt= text(0.02, 0.65, ['<E_{kin}> = ',num2str(Ek),' eV'], ...
         'FontName', 'Arial',...
         'FontSize', 9);

  H_LamTxt= text(0.02, 0.45, ['<\lambda> = ', num2str(lambda),' ', 197],
...
         'FontName', 'Arial',...
         'FontSize', 9);
```

```
H_DxTxt= text(0.02, 0.25, ['<\Delta x> = ', num2str(Dx),' ', 197], ...
       'FontName', 'Arial',...
       'FontSize', 9);


    %---------------END DATA PLOTTING --------------------------------

    %---------SAVING DATA IN THE FIGURE-WINDOW USERDATA STACK -------
    handles=[H_AX, H_LV, H_LP, H_LR, H_LI,...
            H_EkTxt, H_LEav, H_TXT, H_TXT0,...
            H_POP, H_FIG, H_TEav, H_EavTxt, relMass,...
      potFlag, Ko, Dx, H_SL, H_ED, Eav, Ek, lambda,...
      H_RUN, H_STOP, H_REST, H_LamTxt, H_DxTxt, H_WED, H_WSL];

  set(1, 'Userdata', handles); % save data in Fig. stack

 wpacket('again');

    elseif strcmp(Command, 'WNumberSliderMoved'),

    if runFlag > 0,
          set(H_SL,'value',Ko);
       return
    end
    Ko = get(H_SL,'value');
    set(H_ED, 'String', num2str(Ko));,...
    handles(16)=Ko;

    set(1,'Userdata', handles);
       wpacket('again');


 elseif strcmp(Command, 'WNumberEditChanged'),

      if runFlag >0, ...
               Ko=handles(16);
               set(H_ED,'String',num2str(Ko));
          return;
      end;
      vStr = get(H_ED, 'String');
      Ko = str2num(vStr);

      if(Ko<0),
         Ko=0;
      end
```

```
     if(Ko>10),
        Ko=10;
     end

     set(H_ED, 'String', num2str(Ko));
     set(H_SL, 'Value', Ko);
     handles(16)=Ko;
     set(1,'Userdata', handles);
     wpacket('again');

   elseif strcmp(Command, 'GWidthSliderMoved'),

    if runFlag > 0,
         set(H_WSL,'value',Dx);
      return
    end
    Dx = get(H_WSL,'value');
    set(H_WED, 'String', num2str(Dx));,...
    handles(17)=Dx;
    set(1,'Userdata', handles);
       wpacket('again');

elseif strcmp(Command, 'GWidthEditChanged'),

     if runFlag > 0, ...
              Dx=handles(17);
              set(H_WED,'String',num2str(Dx));
         return;
     end;
     vStr = get(H_WED, 'String');
     Dx = str2num(vStr);

     if(Dx<0.2),
        Dx=0.2;
     end

     if(Dx>10),
        Dx=10;
     end

     set(H_WED, 'String', num2str(Dx));
     set(H_WSL, 'Value', Dx);
     handles(17)=Dx;
     set(1,'Userdata', handles);
```

```
        wpacket('again');


   elseif strcmp(Command, 'Run'),

            runFlag=1;
            set(H_RUN, 'Enable', 'off');
      set(H_STOP, 'Enable', 'on');
      set(H_POP, 'Enable', 'off');

      wpacket('again');

      elseif strcmp(Command, 'Stop'),

            runFlag=2;
            set(H_STOP, 'Enable', 'off');
      set(H_RUN, 'Enable', 'on');
      set(H_REST, 'Enable', 'on');
      set(H_POP, 'Enable', 'off');
      wpacket('again');

   elseif strcmp(Command, 'Rest'),

      runFlag=0;
      set(H_STOP, 'Enable', 'off');
      set(H_RUN, 'Enable', 'on');
      set(H_POP, 'Enable', 'on');
      set(H_REST, 'Enable', 'off');
      wpacket('again');

      elseif strcmp(Command, 'Choose_pot'),
      if runFlag,
         return;
      end


      potFlag = get(H_POP, 'Value')-1;
      string = get(H_POP, 'String');
            [V, xo, Dx, Comment]=ptntls(x, potFlag);

         set(H_WED, 'String', num2str(Dx));
      set(H_WSL, 'Value', Dx);
      handles(17)=Dx;
      set(1,'Userdata', handles);
      wpacket('again');

   elseif strcmp(Command, 'again'),

            if runFlag==0,
                  [Psi, x, Ek, lambda] =  gsspckt(xo, Dx, Ko);
                  Eav = Ek+sum((abs(Psi).^2).*V);
%     averaged state energy
                  handles(20)=Eav;
         handles(21)=Ek;
         handles(22)=lambda;
                  set(1, 'Userdata', handles);
         set(H_LV, 'YData', V , 'Color', 'b', 'LineWidth',
3,'EraseMode', 'xor');

      end

            figure(1)
      axes(H_AX);


            if abs(lambda)> 1e9;,
                  lambdaStr='\infty';
            else
                  lambdaStr=[num2str(lambda),' ',197];
            end


            %--------------WAVE_PACKET ANIMATION -------------------------
------
            if Eav<50,
                  dt=0.01*4.13/50;
            else
                  dt = 0.01*4.13/real(Eav); % elementary time-interval
in femtoseconds
%       it must be E*dt/h_bar << 2*pi, i.e. dt << 4.13/E femtosec
            end

      set(H_TEav, 'Position', [-19, Eav+16], 'String', '<E>','Color',
'r');
      set(H_LEav,'YData', [Eav, Eav]);
      set(H_EkTxt, 'String', ['<E_{kin}> = ',num2str(Ek),' eV']);
      set(H_EavTxt, 'String', ['<E> = ',num2str(Eav),' eV']);
      set(H_LamTxt, 'String', ['<\lambda> = ',lambdaStr]);
      set(H_DxTxt, 'String', ['<\Delta x> = ', num2str(Dx),' ', 197]);
      CycleNum=10; % number of times the equation is incrementally
solved by the routine WVPCKT.MEX
            PAR=[dt, xWidth, relMass, CycleNum];
            j=0;
            firstPlot=1;

      while runFlag==1 | firstPlot==1,


                  set(H_LR, 'YData', 400*imag(Psi));
         set(H_LI, 'YData', 400*real(Psi));
                  set(H_TXT, 'String', num2str(j*dt*CycleNum));
         set(H_LP, 'YData',(0.05*400*400)*abs(Psi).^2);

         drawnow;
                  firstPlot=0;

         Psi=schroeq(Psi, V, PAR);
                  j=j+1;
            end
end
```

ภาคผนวก 2

โปรแกรมไอเกนฟังก์ชันและค่าไอเกน

```
      function potwells(Command, radioButt)
%
%     POTWELLS.M is a MATLAB-5 GUI-INTERFACE for the visualisation
%     of eigenfunctions and eigenvalues of a 1-dimensional electron
%     in various potential wells. The programme calls the routines
%     SCHROSC.DLL to perform the time evolution of an electron
%     eigenfunction within an oscillating well and SCHRSTP.DLL the
%     time
%     evolution of an electron eigenfunction within a well evolving
%     to another well (see SCHROSC.M and SCHRSTP.M).
%
%     The program can be enriched by adding other potential
%     profiles names in the UICONTROL with handle H_POP and
%     corresponding potential profile
%     definitions in the list of the called-back function:
%     potwells('Choose_Pot').
%
%     The eigenvalue equation solved by the routine is:
%
%      -K*[d^2u(x)]/dx^2 + V(x)*u(x) = u(x)*E
%
%     with K = (h_bar)^2/(2*Me) =  3.8099827 eV*A^2 (A=Angstrom),
%     where
%     h_bar is the Planck constant divided by 2*pi and Me is the
%     electron mass.
%
%     Useful data:
%     h_bar= 0.65821 eV*fsec, the Planck's constant
%    (fsec=femtosecond = 10^(-12) sec)
%     c = 2998 A/fsec, the velocity of light (A = Angstrom = 10^(-
%     11) m)
%     m_e = 5.11e5 eV, the electron mass
%
%     The X-axis interval is discretised as a set of N=201 points.
%     Initially,
%     the x-axisscale ranges from -10 A to 10 A (A=Angstrom)

global Psi Phi Pden E x u Elevel N V DV Vo Vfig Vold runFlag
stopFlag to
```

```
if nargin < 1,
   Command = 'Initialise';
      figure(1)
      clf
   set(1, 'Name', ['Potential wells ']);
      set(1, 'Userdata', zeros(size(1:36)));
end

if ~strcmp(Command, 'Initialise'),
%   retrieving data saved at the end of the 'Initialise' call;
      handle=get(1, 'Userdata');

%   retrieving data from the Userdata buffer of the window
   H_RAD=zeros(1,4);

   H_LSPEC=zeros(1:5);

   H_FIG= handle(1);
   H_AX = handle(2);
   H_LV = handle(3);
   H_LP = handle(4);
   H_LE = handle(5);
   H_ED = handle(6);
   H_ETXT= handle(7);
   H_POP = handle(8);
   H_SLSCALE = handle(9);
   H_RAD(1)=handle(10);
      H_RAD(2)=handle(11);
      H_RAD(3)=handle(12);
      H_RAD(4)=handle(13);
   H_EDFREQ=handle(14);
   H_STOP = handle(15);
   H_CONT = handle(16);
      H_REST = handle(17);
      H_LFR = handle(18);
      H_LFI = handle(19);
   H_ETXT0 = handle(20);
   H_EDSCALE = handle(21);
   H_SLFREQ = handle(22);
      H_SPEC = handle(23);

   dT = handle(24);
   Vtop = handle(25);
   Vbottom=handle(26);
   scale = handle(27);
   oldRadioButt=handle(28);
      N = handle(29);
      K = handle(30);
   M = handle(31);
      oscFlag=handle(32);
   freq=handle(33);
   Yr = handle(34);
   level = handle(35);
      H_LSPEC(1) = handle(36);
   H_LSPEC(2) = handle(37);
   H_LSPEC(3) = handle(38);
   H_LSPEC(4) = handle(39);
   H_LSPEC(5) = handle(40);
      freqFlag = handle(41);
   H_FCBUTT = handle(42);
   H_FREQTXT = handle(43);

end

if strcmp(Command, 'Initialise'),

% -------------- SYSTEM SETTING ------------------------------

      handle=zeros(1,46);

      K = 3.8099827;
      M=80;
      level=1;
      scale = 0.1;
      x=scale*(-100:100);
      N=length(x);
      V=zeros(1,N);
      Vold=V;
      Vfig=V;
      runFlag=0;
      stopFlag=0;
      freqFlag=1; % frequency change enabled
      oscFlag=0;
      to=0;
      dT=0.001;
      freq=10;
      Phi = zeros(1,N);
      Psi = zeros(N, N);
      Pden =zeros(N, N);

      Vbottom=-2;
      Vtop=4;
      Yr=0;

      H_AX = axes('Units', 'normalized',...
         'Position', [0.075, 0.15, 0.725, 0.78], ...
                  'XLim',[x(1),x(N)], ...
         'YLim',[Vbottom, Vtop],...
         'Box', 'on');
      title('Energy eigenfunctions and eigenvalues of an
electron');
      xlabel(['Electron position (Angstr',246,'m)']);
            ylabel('Energy level (eV)');


      H_LV = line('XData', x, 'Ydata', V, 'Color',...
                        'b', 'LineWidth', 2, 'EraseMode', 'xor');

      H_LP = line('XData', x, 'Color','k', 'EraseMode', 'xor');

      H_LFR = line('XData', x, 'Color',[1,0.75, 1],'EraseMode',
'xor');

            H_LFI = line('XData', x, 'Color','c','EraseMode',
'xor');

      H_LE = line('XData', [x(1), x(N)], 'Color','r', 'EraseMode',
'xor');

            line('XData', [100*x(1), 100*x(N)],...
%       baseline of probability density
              'YData', [0, 0],...
              'LineStyle', ':',...
                    'Color',[0.8, 0.8, 0.8], 'EraseMode', 'xor');

      if Yr ~=0,
         line('XData', [100*x(1), 100*x(N)],...
%       baseline of complex and real parts
              'YData', [Yr, Yr],...
              'LineStyle', ':',...
                    'Color', [0.8, 0.8, 0.8], 'EraseMode', 'xor');
            end

% ---------------- UICONTROLS -------------

            H_FIG=1;
      uicontrol(H_FIG, 'style', 'text',...
         'string', 'Energy level No:', ...
         'Units', 'normalized',...
         'Position', [0.7, 0.015, 0.175, 0.04],...
         'Background', [0.8, 0.8, 0.8]);

      text(0.75, 0.15, '- Probability density',...
         'Unit', 'Normalized',...
         'Color','k', 'FontName', 'Arial',...
         'FontSize', 8);
```

```
text(0.75, 0.1, '- Real part',...
   'Unit', 'Normalized',...
      'Color','m', 'FontName', 'Arial',...
   'FontSize', 8);

      text(0.75, 0.05, '- Imaginary part',...
      'Unit', 'Normalized',...
      'Color',[0, 0.8, 0.8], 'FontName', 'Arial',...
   'FontSize', 8);


H_ED =  uicontrol(H_FIG, 'style', 'edit',...
   'Units', 'normalized',...
   'Position', [0.875, 0.02, 0.05, 0.04],...
   'String', '1',...
      'Background', [0.8, 0.8, 0.8]);

H_BUP =  uicontrol(H_FIG, 'Style', 'pushbutton',...
   'String', 'up',...
   'Callback', 'potwells(''UpButt'');',...
   'Units', 'normalized',...
   'Position', [0.93, 0.04, 0.06, 0.04]);

H_BDOWN =  uicontrol(H_FIG, 'Style', 'pushbutton',...
   'String', 'down',...
   'Callback', 'potwells(''DownButt'');',...
   'Units', 'normalized',...
   'Position', [0.93, 0.0, 0.06, 0.04]);

      H_ETXT0= uicontrol(H_FIG, 'style', 'text',...
   'string', 'Energy (eV) = ', ...
   'HorizontalAlignment', 'right',...
   'Units', 'normalized',...
   'Position', [0.12, 0.155, 0.15, 0.03],...
   'Background', 'w'); %[0.8, 0.8, 0.8]);

      H_ETXT = uicontrol(H_FIG, 'style', 'text',...
   'string', '0.0', ...
   'HorizontalAlignment', 'left',...
   'Units', 'normalized',...
   'Position', [0.28, 0.155, 0.15, 0.03],...
   'Background', 'w'); %[0.8, 0.8, 0.8]);

      Yo=0.105;

uicontrol(H_FIG, 'style', 'text',...
   'string','Potential type', ...
   'Units', 'normalized',...
   'Position', [0.8, 0.865+Yo, 0.18, 0.03],...
   'Background', [0.8, 0.8, 0.8]);
```

```
           H_POP = uicontrol(H_FIG, 'style', 'pop',...
        'string',['Flat well|Step well|Double well|Triple
well|'...
        'Flat well tilted|Step well tilted|Double well tilted|'...
        'Staircase|Staircase tilted|Multibarrier|'...
        'Parabolic|Quartic|Quartic tilted'],...
           'Callback', 'potwells(''Choose_pot'')',...
        'Value', 1,...
           'Units', 'normalized',...
        'Position', [0.8, 0.84+Yo, 0.2, 0.02],...
        'Background', 'w');


     H_SLSCALE = uicontrol(H_FIG, 'style', 'slider',...
        'Min', 0.01, 'Max', 0.8, 'Value', scale, ...
        'Callback','potwells(''ScaleSlider'')', ...
                 'Units', 'normalized',...
        'Position', [0.41, 0.01, 0.14, 0.04]);

           H_EDSCALE = uicontrol(H_FIG, 'style', 'edit',...
        'Callback','potwells(''EdScale'')', ...
                 'Units', 'normalized',...
        'Position', [0.35, 0.01, 0.06, 0.04],...
        'String', num2str((N-1)*scale),...
        'Background', 'w');

     uicontrol(H_FIG, 'style', 'text',...
        'string','X-axis width (Angstrom):', ...
        'Units', 'normalized',...
        'Position', [0.14, 0.01, 0.21, 0.04],...
        'Background', [0.8, 0.8, 0.8]);


% ---------- DYNAMIC SECTION ---------------------
        uicontrol(H_FIG, 'style', 'frame',...
        'Units', 'normalized',...
        'Position', [0.805, 0.31+Yo, 0.195, 0.495],...
        'Background', [0.8, 0.8, 0.8]);



           uicontrol(H_FIG, 'style', 'text',...
        'string','DYNAMIC SECTION', ...
        'Units', 'normalized',...
        'Position', [0.81, 0.773+Yo, 0.18, 0.03],...
        'Background', [0.8, 0.8, 0.8]);

           H_RAD=zeros(1,4);
```

```
      H_RAD(1) = uicontrol(H_FIG, 'style', 'radio',...
   'string','Fast change', ... %         'Value', 1, ...
      'Callback','potwells(''Radio'', 1)', ...
      'Units', 'normalized',...
      'Position', [0.81, 0.73+Yo, 0.18, 0.03],...
   'Background', [0.8, 0.8, 0.8]);

H_RAD(2) = uicontrol(H_FIG, 'style', 'radio',...
   'string','Moderate change', ...
   'Callback','potwells(''Radio'', 2)', ...
   'Units', 'normalized',...
      'Position', [0.81, 0.68+Yo, 0.185, 0.03],...
   'Background', [0.8, 0.8, 0.8]);

H_RAD(3) = uicontrol(H_FIG, 'style', 'radio',...
   'string','Slow change', ...
   'Callback','potwells(''Radio'', 3)', ...
      'Units', 'normalized',...
      'Position', [0.81, 0.63+Yo, 0.185, 0.03],...
   'Background', [0.8, 0.8, 0.8]);

   H_RAD(4) = uicontrol(H_FIG, 'style', 'radio',...
   'string','10 oscillations', ...
   'Callback','potwells(''Radio'', 4)', ...
   'Units', 'normalized',...
   'Position', [0.81, 0.58+Yo, 0.18, 0.03],...
   'Background', [0.8, 0.8, 0.8]);

 H_SLFREQ = uicontrol(H_FIG, 'style', 'slider',...
   'Min', 0.01, 'Max', 10, 'Value', 0.05, ...
   'Callback','potwells(''FreqSlider'')', ...
            'Units', 'normalized',...
   'Position', [0.87, 0.5+Yo, 0.125, 0.04]);

H_EDFREQ = uicontrol(H_FIG, 'style', 'edit',...
   'Callback','potwells(''EdFreq'')', ...
            'Units', 'normalized',...
   'Position', [0.81, 0.5+Yo, 0.06, 0.04],...
   'String', ' ',...
   'Background', 'w');

H_FREQTXT=uicontrol(H_FIG, 'style', 'text',...
   'Units', 'normalized',...
   'Position', [0.81, 0.45+Yo, 0.185, 0.04],...
   'String', ' ',...
   'Background', [0.8, 0.8, 0.8]);
```

```
H_FCBUTT = uicontrol(H_FIG, 'Style', 'pushbutton',...
   'String', 'Enable freq. change',...
   'Callback', 'potwells(''FreqChange'');',...
   'Units', 'normalized',...
   'Position', [0.81, 0.41+Yo, 0.185, 0.04]);

uicontrol(H_FIG, 'style', 'text',...
   'string','Frequency (1/fsec)', ...
   'Units', 'normalized',...
   'Position', [0.81, 0.545+Yo, 0.185, 0.03],...
   'HorizontalAlignment', 'left',...
      'Background', [0.8, 0.8, 0.8]);



H_STOP =  uicontrol(H_FIG, 'Style', 'pushbutton',...
   'String', 'Stop',...
   'Enable', 'off',...
      'Callback', 'potwells(''StopButt'');',...
   'Units', 'normalized',...
   'Position', [0.81, 0.36+Yo, 0.09, 0.04]);

H_CONT =  uicontrol(H_FIG, 'Style', 'pushbutton',...
   'String', 'Continue',...
   'Enable', 'off',...
   'Callback', 'potwells(''ContButt'');',...
   'Units', 'normalized',...
   'Position', [0.905, 0.36+Yo, 0.09, 0.04]);

H_REST =  uicontrol(H_FIG, 'Style', 'pushbutton',...
   'String', 'Restore',...
   'Enable', 'off',...
   'Callback', 'potwells(''RestoreButt'');',...
   'Units', 'normalized',...
   'Position', [0.81, 0.315+Yo, 0.09, 0.04]);

H_SPEC =  uicontrol(H_FIG, 'Style', 'pushbutton',...
   'String', 'Spectrum',...
   'Enable', 'off',...
   'Callback', 'potwells(''SpecButt'');',...
   'Units', 'normalized',...
   'Position', [0.905, 0.315+Yo, 0.09, 0.04]);


H_ASPEC = axes('Units', 'normalized',...
   'Position', [0.805, 0.15+Yo, 0.195, 0.162], ...
         'XLim',[0.5,5.5], ...
   'YLim',[0,1],...
   'Box', 'on',...
```

```
         'YTickMode', 'manual',...
         'FontName', 'Arial',...
         'FontSize', 7,...
         'XTickLabel',['E1|E2|E3|E4|E5']);

            uicontrol(H_FIG, 'style', 'text',...
         'string','Energy spectrum of the system in the final
potential configuration.', ...
         'Units', 'normalized',...
         'Position', [0.82, 0.037+Yo, 0.19, 0.084],...
         'HorizontalAlignment', 'left',...
         'Background', [0.8, 0.8, 0.8],...
         'FontName', 'Arial',...
         'FontSize', 7);

            for n=1:5,
         H_LSPEC(n)= line('XData', [n,n], ...
            'YData', [0,0], ...
                        'Color', 'r');
            end


      oldRadioButt=0;
      time=0;
      handle(1) = H_FIG;
            handle(2) = H_AX;
      handle(3) = H_LV;
      handle(4) = H_LP;
      handle(5) = H_LE;
      handle(6) = H_ED;
      handle(7) = H_ETXT;
      handle(8)= H_POP;
      handle(9) = H_SLSCALE;
      handle(10) = H_RAD(1);
            handle(11) = H_RAD(2);
            handle(12) = H_RAD(3);
      handle(13) = H_RAD(4);
      handle(14) = H_EDFREQ;
            handle(15) = H_STOP;
            handle(16) = H_CONT;
            handle(17) = H_REST;
            handle(18) = H_LFR;
      handle(19) = H_LFI;
            handle(20) = H_ETXT0;
      handle(21) = H_EDSCALE;
      handle(22) = H_SLFREQ;
            handle(23) = H_SPEC;
      handle(24) = dT;
      handle(25) = Vtop;
      handle(26) = Vbottom;
      handle(27) = scale;
      handle(28) = oldRadioButt;
      handle(29) = N;
      handle(30)= K;
      handle(31) = M;
            handle(32) = oscFlag;
            handle(33) = freq;
      handle(34)=Yr;
      handle(35) = level;
      handle(36) = H_LSPEC(1);
      handle(37) = H_LSPEC(2);
      handle(38) = H_LSPEC(3);
      handle(39) = H_LSPEC(4);
      handle(40) = H_LSPEC(5);
      handle(41) = freqFlag;
            handle(42) = H_FCBUTT;
      handle(43) = H_FREQTXT;


      set(H_FIG, 'Userdata', handle);
%     save data in Userdata buffer

      potwells('Choose_pot');

   elseif strcmp(Command, 'EdScale'),
      if runFlag,
         set(H_EDSCALE, 'String', num2str(scale));
         return
      end
      scaleString = get(H_EDSCALE, 'String');
      scale = str2num(scaleString);
      maxVal=get(H_SLSCALE, 'Max');
      minVal=get(H_SLSCALE, 'Min');
      if scale > maxVal,
         scale = maxVal;
         set(H_EDSCALE, 'String', maxVal);
      elseif scale < minVal,
                  scale = minVal;
         set(H_EDSCALE, 'String', minVal);
            end

      set(H_SLSCALE, 'Value', scale);
      handle(27)=scale;
      set(H_FIG, 'Userdata', handle);
      potwells('ScaleSlider');
```

```
elseif strcmp(Command, 'ScaleSlider'),
   if runFlag,
      set(H_SLSCALE, 'Value', scale);
      return
   end
   scale = get(H_SLSCALE, 'Value');
   set(H_EDSCALE, 'String', num2str((N-1)*scale));

   handle(27)=scale;

   set(1, 'Userdata', handle);

         x = scale*(-100:100);

   set(H_AX, 'XLim', [x(1), x(N)]);
   set(H_LV, 'Xdata', x);
   set(H_LP, 'Xdata', x);
   set(H_LFR,'Xdata', x);
   set(H_LFI,'Xdata', x);
   set(H_LE, 'Xdata', [x(1), x(N)]);

   potwells('Choose_pot');


elseif strcmp(Command, 'EdFreq'),
   if runFlag,
      set(H_EDFREQ, 'String', num2str(freq));
      return
   end
   freqString = get(H_EDFREQ, 'String');
         freq = str2num(freqString);
   maxVal=get(H_SLFREQ, 'Max');
   minVal=get(H_SLFREQ, 'Min');
   if freq > maxVal,
      freq = maxVal;
      set(H_EDFREQ, 'String', maxVal);
   elseif freq < minVal,
               freq = minVal;
      set(H_EDFREQ, 'String', minVal);
         end

   set(H_SLFREQ, 'Value', freq);
   handle(33)=freq;
   set(H_FIG, 'Userdata', handle);

   elseif strcmp(Command, 'FreqSlider'),
   if runFlag,
      set(H_SLFREQ, 'Value', freq);
      return
   end
   freq = get(H_SLFREQ, 'Value');
   set(H_EDFREQ, 'String', num2str(freq));

   handle(33)=freq;

   set(1, 'Userdata', handle);

   elseif strcmp(Command, 'FreqChange'),
   if runFlag,
         return
      end
   set(H_FREQTXT, 'String','');
   if freqFlag==1,
      set(H_FCBUTT, 'String', 'Disable freq. change');
      freqFlag = 0;
   else,
      set(H_FCBUTT, 'String', 'Enable freq. change');
      freqFlag = 1;
   end
   handle(41)=freqFlag;
   set(1, 'Userdata', handle);

elseif strcmp(Command, 'UpButt'),
   if runFlag + stopFlag,
      return
   end
   level=level+1;
   if level > M,
      level = M;
   end
   handle(35)=level;
   set(1, 'Userdata', handle);
   potwells('Draw');

elseif strcmp(Command, 'DownButt'),
   if runFlag + stopFlag,
      return
   end

   level=level-1;
   if level < 1,
      level = 1;
   end

   handle(35)=level;
   set(1, 'Userdata', handle);
   potwells('Draw');
   elseif strcmp(Command, 'Choose_pot'),
      if runFlag,
         return
      end

      choice = get(H_POP, 'Value');
      string = get(H_POP, 'String');

      Vo=zeros(1,N);
      V=Vo;

% 'Flat well|Step well|Double well|Triple well|'...
% 'Flat well tilted|Step well tilted|Double well tilted|'...
% 'Staircase|Staircase tilted|Multibarrier|'...
% 'Parabolic|Quartic|Quartic tilted|'...

      if choice==1, % FLAT WELL

                  DV=(0.01/scale)*x;

      elseif choice==2, % STEP WELL
         Na = floor(N/4);
            V(Na:N)=ones(size(Vo(Na:N)));
                  DV=(0.01/scale)*x;

      elseif choice==3, % DOUBLE WELL

         Na = floor(N/3);
         Nb = N-Na;
         V(Na:Nb)= 2*ones(size(Vo(Na:Nb)));
         V(Nb:N)=ones(size(Vo(Nb:N)));
            DV=(0.01/scale)*x;

      elseif choice==4, % TRIPLE WELL

         Na = floor(N/4);
         Nb = floor(N/2);
         Nc = N-Na;
         V(Na:Nb)= 2*ones(size(Vo(Na:Nb)));
         V(Nb:Nc)=-1*ones(size(Vo(Nb:Nc)));
         V(Nc:N)= ones(size(Vo(Nc:N)));
                  DV=(0.01/scale)*x;

      elseif choice==5, % FLAT WELL TILTED
         V = V - (0.01/scale)*x;
                  DV=(0.02/scale)*x;

      elseif choice==6, % STEP WELL TILTED
         Na = floor(N/4);
                  V(Na:N)=ones(size(Vo(Na:N)));
            V = V - (0.01/scale)*x;
         DV=(0.02/scale)*x;

       elseif choice==7, % DOUBLE WELL TILTED

         Na = floor(N/3);
         Nb = N-Na;
         V(Na:Nb)= 2*ones(size(Vo(Na:Nb)));
         V(Nb:N)=ones(size(Vo(Nb:N)));
         V = V - (0.01/scale)*x;
                  DV=(0.02/scale)*x;

       elseif choice==8, % STAIRCASE
         Na = floor(N/6);
                  Nb = floor(2*N/6);
                  Nc = floor(3*N/6);
         Nd = floor(4*N/6);
         V(Na:Nb)= 0.5*ones(size(Vo(Na:Nb)));
         V(Nb:Nc)= 1.5*ones(size(Vo(Nb:Nc)));
                  V(Nc:Nd)= 2*ones(size(Vo(Nc:Nd)));
                  V(Nd:N)= 2.5*ones(size(Vo(Nd:N)));
            DV=(0.01/scale)*x;

      elseif choice==9, % STAIRCASE TILTED

         Na = floor(N/6);
                  Nb = floor(2*N/6);
                  Nc = floor(3*N/6);
         Nd = floor(4*N/6);
         V(Na:Nb)= 0.5*ones(size(Vo(Na:Nb)));
         V(Nb:Nc)= 1.5*ones(size(Vo(Nb:Nc)));
                  V(Nc:Nd)= 2*ones(size(Vo(Nc:Nd)));
                  V(Nd:N)= 2.5*ones(size(Vo(Nd:N)));
                  V = V - (0.01/scale)*x;
            DV=(0.02/scale)*x;

         elseif choice==10, % MULTIBARRIER
            V=Vo;

            W1 =floor(N/21);
                        W2=floor(N/33);
            Nz=zeros(1,26);

            Nz(1)=W1-1;
            Nz(2)=Nz(1)+W2;
            V(Nz(1):Nz(2))= ones(size(Vo(Nz(1):Nz(2))));
```

```
            for r=3:2:25,
                  Nz(r)=Nz(r-1)+W1;
                  Nz(r+1)=Nz(r)+W2;
                  V(Nz(r):Nz(r+1))= ones(size(Vo(Nz(r):Nz(r+1))));
            end

                  DV=(0.01/scale)*x;

        elseif choice==11, % PARABOLIC

                  V = x.^2;
            V=(4/V(1))*V;
                  DV=(0.01/scale)*x;

        elseif choice==12, % QUARTIC

         const = x(1)^4;
         fact= 0.8*x(1)*x(1);
         V = x.^4 - fact*x.^2 + 0.1*const;
         V=(4/V(1))*V;
                  DV=(0.01/scale)*x;


       elseif choice==13, % QUARTIC TILTED

         const = x(1)^4;
         fact= 0.8*x(1)*x(1);
         V = x.^4 - fact*x.^2 + 0.1*const;
         V=(4/V(1))*V;
         V = V - (0.01/scale)*x;
                  V=(4/V(1))*V;
                  DV=(0.02/scale)*x;

       end

      V(1)=1e6;
      V(N)=1e6;


      set(H_LV, 'Ydata', V);

      potwells('Solve_equation');
      potwells('Draw');

   elseif strcmp(Command, 'Solve_equation'),

      if runFlag,
         return
      end
      Laplacian = diag(ones(1,N-1),1)+diag(ones(1,N-1),-1)-
diag(2*ones(1,N));

      H = -K/(scale*scale)*Laplacian+diag(V);

            [Psi,D]=eig(H);
            [E, Ind]=sort(diag(D));

            Psi=Psi(:,Ind)+i*eps*zeros(size(Psi));

      D=D(:,Ind);

      M=40;

            for n=1:M,
            Psi(:,n)=Psi(:,n)/norm(Psi(:,n), 'fro');
            a=sum(Psi(1:floor(N/2), n));
            if a < 0,
                  Psi(:, n)=-Psi(:,n);
            end
            Pden(:, n) = abs(Psi(:,n)).^2;
      end


     elseif strcmp(Command, 'Draw'),
        if runFlag,
         return
      end

      set(H_LP, 'Ydata', 60*Pden(:,level),'EraseMode', 'xor');

      set(H_LFR,'Ydata', Yr+4*real(Psi(:, level)));

      set(H_LFI,'Ydata', Yr+4*imag(Psi(:, level)));


      set(H_LE, 'Ydata', [E(level), E(level)], 'Color','r',
'EraseMode', 'xor');

      set(H_ED, 'String', num2str(level));

      set(H_ETXT, 'String', num2str(E(level)));

   elseif strcmp(Command, 'Radio'),
      if runFlag,
            potwells('RestoreButt');
      end
      h_bar= 0.65821; % eV*fsec
      set(H_FREQTXT, 'String','');
      if freqFlag,
            if radioButt==1,
            freq = 1e-2/dT;  % help schrstp.m
         elseif radioButt==2,
            freq=1e-4/dT;
            elseif radioButt==3,
            freq=1e-5/dT;
            elseif radioButt==4,
            freq=(E(level+1)-E(level))/(2*pi*h_bar);
% help schrosc.m
            set(H_FREQTXT, 'String',['Freq.=[E(',
num2str(level+1),')-E(',num2str(level),')]/h'])

         end
            end

      set(H_RAD, 'Value', 0);
            set(H_RAD(radioButt), 'Value', 1);
         handle(28)=radioButt;
      handle(33)=freq;

      set(H_FIG, 'Userdata', handle);

      set(H_SLFREQ, 'Value', freq);
      set(H_EDFREQ, 'String', num2str(freq));


      runFlag=1;
      stopFlag=0;
      time=0;

      set(H_LE, 'Ydata', [10, 10]);
            set(H_ETXT0, 'String', 'Time (fsec) = ');

            set(H_STOP, 'Enable', 'on');
      set(H_FCBUTT, 'Enable', 'off');

      potwells('Solve_equation');
         potwells('InitDynamics', radioButt);

      potwells('Evolve', radioButt);

   elseif strcmp(Command, 'StopButt'),

      set(H_RAD, 'Value', 0);
      runFlag=0;
      stopFlag=1;
      set(H_CONT, 'Enable', 'on');
      set(H_REST, 'Enable', 'on');
      set(H_SPEC, 'Enable', 'on');
      set(H_STOP, 'Enable', 'off');

      for n=1:5,
         set(H_LSPEC(n), 'YData', [-1,-1]);
      end

   elseif strcmp(Command, 'ContButt'),

      for n=1:5,
         set(H_LSPEC(n), 'YData', [-1,-1]);
      end

            set(H_SPEC, 'Enable', 'off');
      set(H_STOP, 'Enable', 'on');
      set(H_REST, 'Enable', 'off');
            set(H_SPEC, 'Enable', 'off');

      set(H_RAD(oldRadioButt), 'Value', 1);
      runFlag=1;

      potwells('Evolve', oldRadioButt);

   elseif strcmp(Command, 'RestoreButt'),

      runFlag=0;
      stopFlag=0;
      to=0;
      set(H_STOP, 'Enable', 'off');
      set(H_CONT, 'Enable', 'off');
            set(H_REST, 'Enable', 'off');
      set(H_SPEC, 'Enable', 'off');
            set(H_FCBUTT, 'Enable', 'on');

      for n=1:5,
         set(H_LSPEC(n), 'YData', [-1,-1]);
      end

      set(H_ETXT0, 'String', 'Energy (eV) = ');
      set(H_ETXT, 'String','0.00');

      set(H_RAD(oldRadioButt), 'Value', 0);

      set(H_LV, 'Ydata', V);

      potwells('Solve_equation');
      potwells('Draw');
```

```
elseif strcmp(Command, 'SpecButt'),
   Vold=V;
   V=Vfig;
   potwells('Solve_equation');

   W=zeros(1:5);


   for n=1:5,
      W(n)=abs(Psi(:,n)'*conj(Phi))^2;
         set(H_LSPEC(n),'YData',[0, W(n)]);
   end

         V=Vold;
         potwells('Solve_equation');

   elseif strcmp(Command, 'InitDynamics'),


   Phi = Psi(:, level);

elseif strcmp(Command, 'Evolve'),


   if radioButt==4,
      cycleNum=100;
   else
      cycleNum=100;
   end

   omega =2*pi*freq;
   tLim=10/freq;

   PAR=[dT, omega, cycleNum, x(N)-x(1), 1];
   Zr=zeros(size(DV));

   while runFlag,

   if radioButt==4,
      if to<tLim,
         [Phi, to, Vfig] = Schrosc(Phi, to, V, DV, PAR);
      else
         [Phi, to, Vfig] = schrosc(Phi, to, V, Zr, PAR);
      end
   else
      [Phi, to, Vfig] =schrstp(Phi, to, V, DV, PAR);
         end
```

```
        set(H_LV, 'Ydata', Vfig);
        set(H_LFI,'Ydata', Yr + 4*imag(Phi));
        set(H_LFR,'Ydata', Yr + 4*real(Phi));
        set(H_LP,'Ydata', 60*abs(Phi).^2);
              set(H_ETXT, 'String',num2Str(to));

        drawnow;
    end
end
```