|  |
| --- |
| [**DOING PHYSICS WITH MATLAB**](http://www.physics.usyd.edu.au/teach_res/mp/mphome.htm)  **TIME DEPENDENT QUANTUM MECHANICAL SCATTERING IN TWO DIMENSIONS**  Ian Cooper  Any comments, suggestions or corrections, please email me at  [matlabvisualphysics@gmail.com](mailto:matlabvisualphysics@gmail.com)  [VISUAL PHYSICS ONLINE](http://www.physics.usyd.edu.au/teach_res/hsp/sp/spHome.htm) |

|  |
| --- |
| **MATLAB**  [DOWNLOAD DIRECTORY FOR SCRIPTS](http://www.physics.usyd.edu.au/teach_res/mp/mscripts/)  **qm2DB.m**  A finite difference time development method (FDTD) is used to solve the two dimensional time dependent Schrodinger Equation. This method is applied to the free propagation of a Gaussian pulse and the scattering of the pulse from different potential energy functions: wall, cliff, single slit, double slit and Coulomb (Rutherford scattering). The variable flagU is used to select the potential. The results of the computations are presented as color graphs portraying the probability density function. The amplitude of the probability for the scattered wave is often very small, so the probability density is scaled to better display the scattering. Arbitrary units are used for all quantities. Values of the wavefunction and potential energy are calculated on a [2D] mesh for a square of length 1. The centre of the square has Cartesian coordinates (0.5, 0.5). The initial wavefunction is a plane with a [2D] Gaussian envelope. The wavefunction is zero at the boundaries of the square. So, when the wave packet strikes the boundaries, reflections occur. This results in interference patterns developing because of the superposition of the incident and reflected waves. *I do not know how to apply absorbing boundary conditions to eliminate the reflections (someone might be able to help me)*. The motion of the pulse can be saved as an animated gif file (flagG). |

|  |
| --- |
| **THE 2-DIMENSIONAL SCHRODINGER EQUATION**  In 2 dimensions and Cartesian coordinates (*x*, *y*) the Schrodinger equation can be expressed as  (1)  The wavefunction  is a complex function  (2)  where the real part of the wavefunction is  and its imaginary part is .  Using equations 1 and 2 and equating the real and imaginary parts, the Schrodinger equation becomes  (3A)  (3B)  We can use the finite difference time development method to solve the 2D Schrodinger equation. Time and position are defined at a set of discrete points  (4A)  (4B)  (4C)  The finite difference approximation assumes that the derivates of a function can be expanded in a Taylor series around every point of the mesh up to a desired order of accuracy.  The first derivative can be approximated as  (5A)  forward difference  (5B)  central difference  The second derivative is approximated as  (5C)  The finite difference approximation from equations 1 to 5 at time step for the  mesh point becomes  (6A) time steps    (6B) time steps    **MATLAB** **qm2DB.m**    The Script **qm2DB.m** uses arbitrary units  for all quantities. The initial pulse is a plane wave with a [2D] envelope. The number 1 attached to a variable represents the current time and a 2 represents the value of the variable at the next time step.  % [2D] GAUSSIAN PULSE (WAVE PACKET) ==================================  % Initial centre of pulse  x0 = 0.20; y0 = 0.5;  if flagU == 6  x0 = 0.40; y0 = 0.75; % y0 impact parameter  end  % Initial amplitude of pulse  A = 10;  % Pulse width: sigma squared [5e-3]  s = 1e-3;  % Wavenumber [50]  k0 = 20;    % Envelope  psiE = A\*exp(-(x-x0).^2/s).\*exp(-(y-y0).^2/s);  % Plane wave propagation in +X direction  psiP = exp(1i\*k0\*x);  % Wavefunction  psi1 = psiE.\*psiP;  % Probability Density  prob1 = conj(psi1).\*psi1;  % Extract Real and Imaginary parts  R1 = real(psi1); I1 = imag(psi1);    Fig. 1. Initial wave packet (pulse)  The wavefunction is called at each successive time step and half time step by calling the functions shown below. The number of time steps is given by the variable nT.  % FUNCTIONS ==========================================================  % psi1 (current value at time t)--> psi2 (next value at time t+dt/2 & t+dt  % functioin pisI at times t+dt/2, t+3dt/2, ....  % function psiR at times t+dt, t+2dt,    function I2 = psiI(N, I1, R1, dt, U, f)  I2 = zeros(N,N);  x = 2:N-1;  y = 2:N-1;    I2(x,y) = I1(x,y) + f\*(R1(x+1,y)-2\*R1(x,y)+R1(x-1,y)+R1(x,y+1)-2\*R1(x,y)+R1(x,y-1))...  - dt\*U(x,y).\*R1(x,y);  end    function R2 = psiR(N, R1, I1, dt, U, f)  R2 = zeros(N,N);  x = 2:N-1;  y = 2:N-1;    R2(x,y) = R1(x,y) - f\*(I1(x+1,y)-2\*I1(x,y)+I1(x-1,y)+I1(x,y+1)-2\*I1(x,y)+I1(x,y-1))...  + dt\*U(x,y).\*I1(x,y);  end  Then the current value of the probability is displayed in a Figure Window.  figure(1)  set(gcf,'units','normalized');  set(gcf,'position',[0.05 0.1 0.30 0.70]);  set(gcf,'color','w');    for c = 1:nT  % Update real prt of wavefunction  R2 = psiR(N, R1, I1, dt, U, f);  R1 = R2;    % Update imaginary part of wavefunction  I2 = psiI(N, I1, R2, dt, U, f);    % Probability Density Function  prob2 = R2.^2 + I1.\*I2;    I1 = I2;    subplot(2,1,1)  . . .  **Simulation 1: Free propagation of the pulse**  We can model the free propagation of the wave packet moving in the X direction. The potential energy function is set to zero at all mesh points.    Fig. 2. Zero potential energy field.  Figure 3 shows an animated view for the motion of the wave packet. The wave number is k0 = 100 and there are 500 time steps. As the wave packet propagates, it spreads in all directions.    Fig.3. Animated view for the motion of the wave packet (k0 =100) for 500 time steps. |