PHYSICS 410

SOLVING BVPs WITH THE ode45 INTEGRATOR AND SHOOTING

Example: Deuteron Model

- Differential equation
$$\frac{d^2u}{dx^2} + (E - V)u = 0$$
where the potential $V(x)$ is given by
$$V(x) = \begin{cases} -1 & 0 \leq x < x_0 \\ 0 & x > x_0 \end{cases}$$
and the boundary conditions at $x = 0$ are
$$u(0) = 0 \qquad \frac{du}{dr}(0) = \psi(0) = 1$$

- E is an eigenvalue and, in the solution of the ODE, must be adjusted using bisection, for example, until the true boundary condition
$$\lim_{x \to \infty} u(x) = 0$$
is achieved

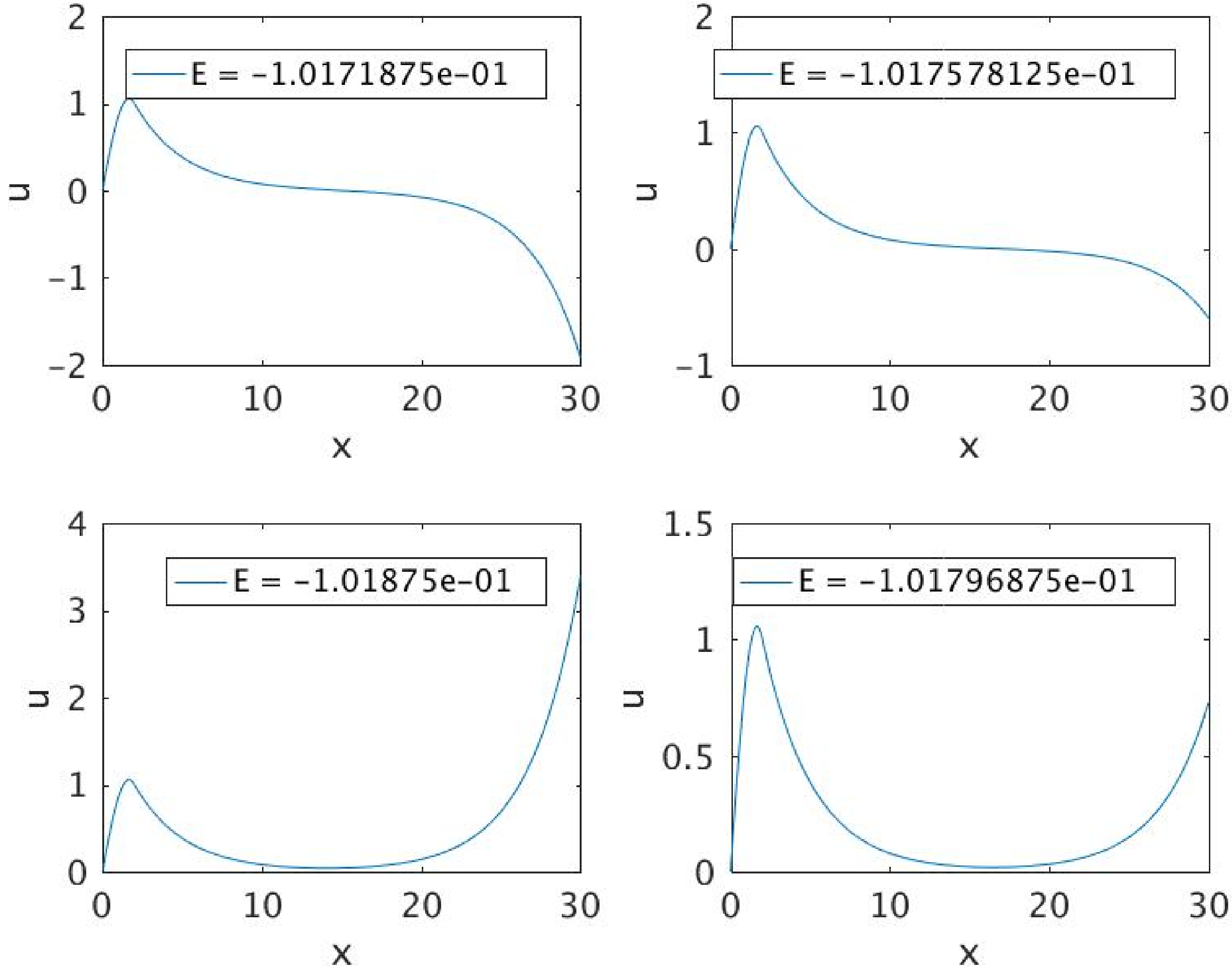
Script deut.m

```
% deut: Solves ODE for toy deuteron problem.
global x0  E;
% Domain outer boundary ...
xmax = 60.0;
% Tolerance parameters ...
abstol = 1.0e-8;
reltol = 1.0e-8;
options = odeset('AbsTol', abstol * [1 1]', 'RelTol', reltol);
% Parameters ...
x0 = 6.0;
E = -0.80067
% Integrate ODE ...
[xout yout] = ode45(@fcn_deut, [0.0 xmax], [0.0 1.0]', options);
% Make plot and output as JPEG ...
figure(1);
clf;
hold on;
axis square;
xlabel('x');
ylabel('u');
plot(xout, yout(:, 1));
print('deut.jpg','-djpeg');
```

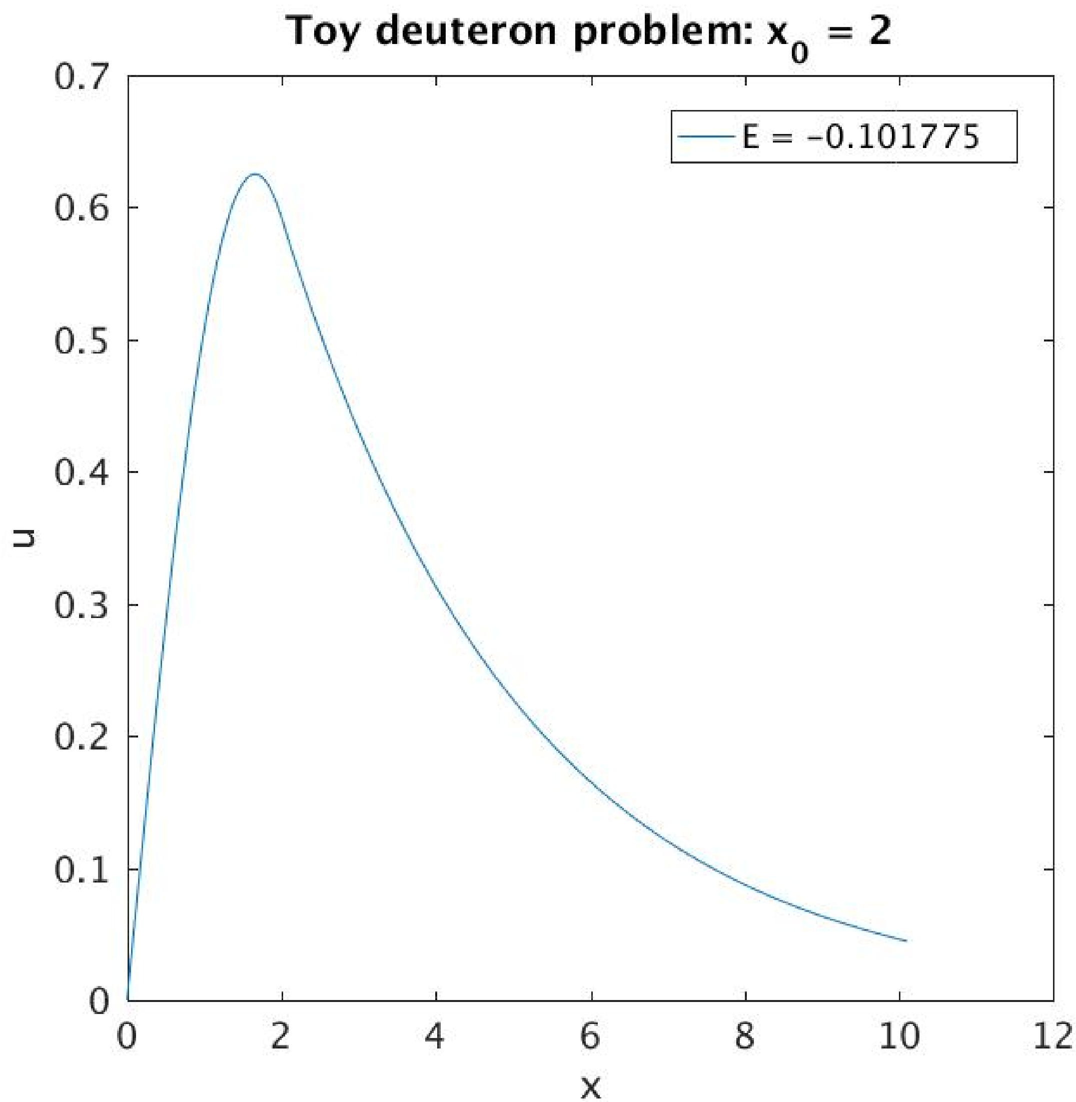
Function `fcn_deut`

```
function dydx = fcn_deut(x, y)
% Function fcn_deut evaluates derivatives for toy deuteron problem.
   global x0  E;

   dydx = ones(2,1);

   dydx(1,1) = y(2);
   if x <= x0
      dydx(2,1) = (-1 - E) * y(1);
   else
      dydx(2,1) = -E * y(1);
   end
end
```

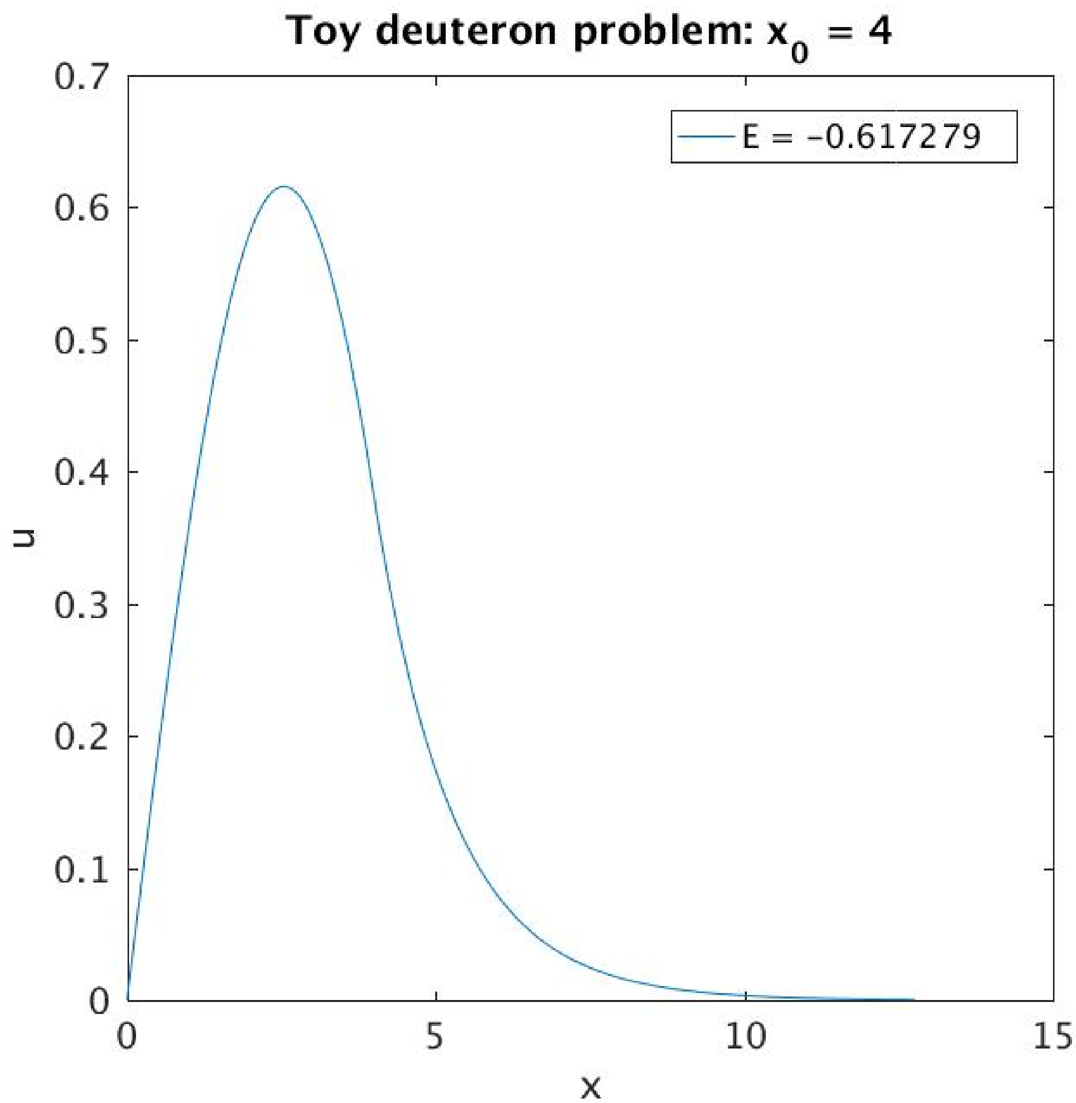
- Sample output during bisection procedure (shooting) for $x_0 = 2.0$



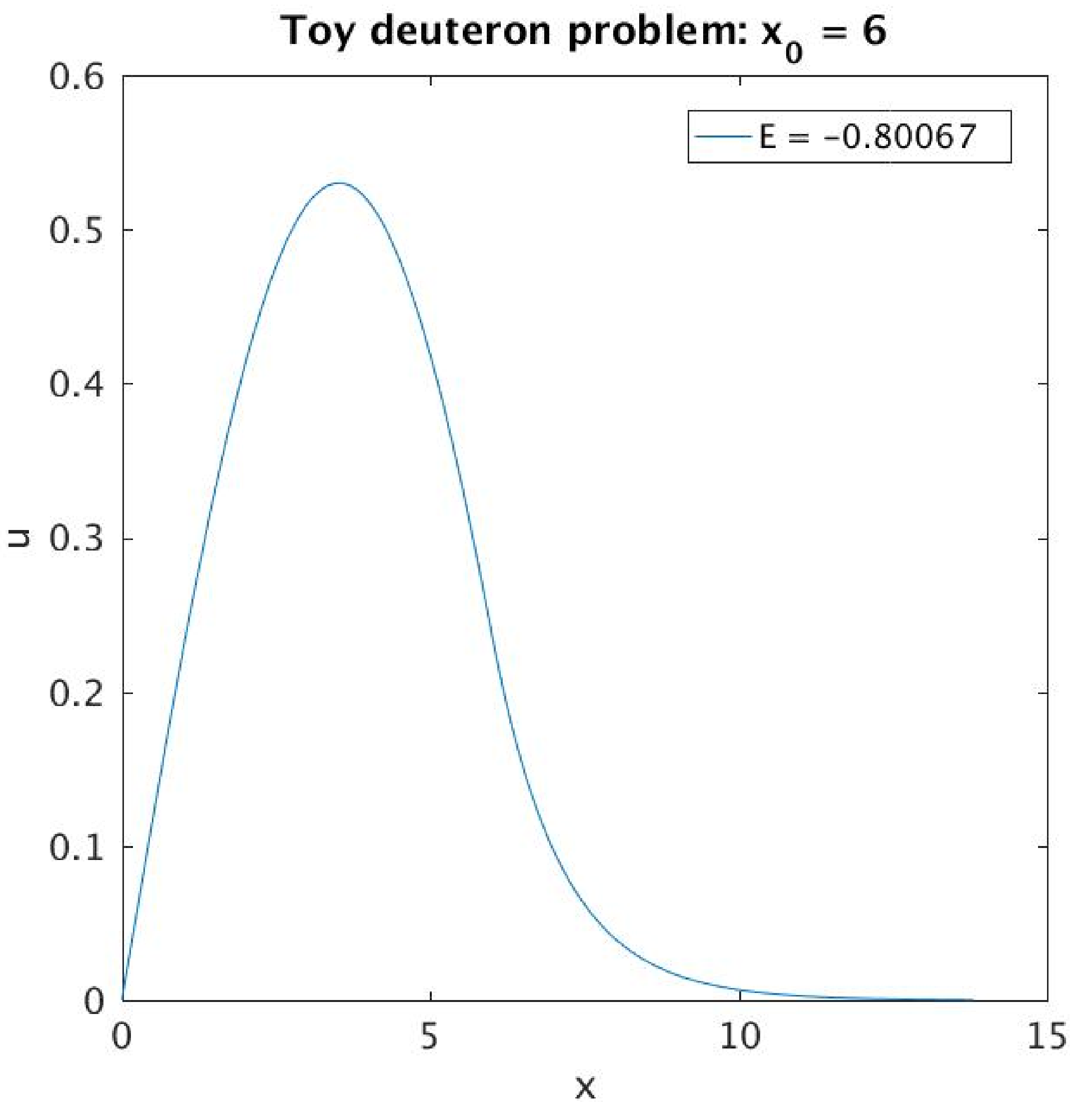
- Normalized wave function for $x_0 = 2.0$



- Normalized wave function for $x_0 = 4.0$



- Normalized wave function for $x_0 = 6.0$



- Normalized wave function for $x_0 = 8.0$