```
function y = eulerMethodSolver(x,y, func)
%Takes in the function, x, and y arrays and solves using euler method, then
%plots. It returns the y array updated with the solutions
%   inputs: -array of x values (x and y array can be obtained from
%            eulerMethodsetup)
%           -array of y values (initial y value must already be correct)
%           -the function to solve (as an anonymous function)
%   outputs: y array now filled in with solutions

%to avoid needing another input for this function, we just deduce what h is
h = x(end)/(length(x)-1);

%now we make h, x, and y for different stepsizes
hsmaller = h/10;
xsmaller = [x(1):hsmaller:x(end)];
ysmaller = zeros(size(xsmaller));
ysmaller(1) = y(1);

hbigger = h*10;
xbigger = [x(1):hbigger:x(end)];
ybigger = zeros(size(xbigger));
ybigger(1) = y(1);

%solves ode for problem using what user input
for i=2:numel(y)-1 %starting at the second value
   y(i) = y(i-1) + h * func(x(i-1),y(i-1)); %current value is based on this formula
end
%solve ode for 10x smaller stepsize
for i=2:numel(ysmaller)-1
   ysmaller(i) = ysmaller(i-1) + hsmaller * func(xsmaller(i-1),ysmaller(i-1));
end
%solve ode for 10x larger stepsize
for i=2:numel(ybigger)-1
   ybigger(i) = ybigger(i-1) + hbigger * func(xbigger(i-1),ybigger(i-1));
end
%solve using matlab's ide solver
[odex, odey] = ode45(func,[0 x(end)], 1);
%solving all of this is just for the sake of comparison later


figure;
plot(odex,odey, 'b');%we first plot the ode45 result to get correct axis
hold on
%plot the other solutions
plot(x,y,'r');
plot(xsmaller,ysmaller, 'c');
plot(xbigger,ybigger, 'g');
title('All approximations');
xlabel('Select a point to compare approximations at that x-coordinate');
%make legend and turn autoupdate off so user input doesn't show up in legend
legend('ode45','user input','smaller stepsize', 'larger stepsize', 'AutoUpdate', 'off');

[xi,yi] = ginput(1);%store point that user selects
plot(xi,yi, 'm*');%plot the point and label it
text(xi+.1, yi, 'User Selection', 'color', 'r');

%because each line has different x and y arrays, we have to find the
%closest x value to the selected x value in each array, then find the y
%value at that x value
[xclosest, xidx] = min(abs(x-xi));%finds the closest x value to user input
xclosest = x(xidx);%reset closest to be the x value at that index
[xsmallerclosest, xsmalleridx] = min(abs(xsmaller-xi));%do the same for the rest
xsmallerclosest = xsmaller(xsmalleridx);
[xbiggerclosest, xbiggeridx] = min(abs(xbigger-xi));
xbiggerclosest = xbigger(xbiggeridx);
[xodeclosest, xodeidx] = min(abs(odex-xi));
xodeclosest = odex(xodeidx);
%display each closest x value and the y value that solution returned there
fprintf('Approximations  at selected x value:\n')
fprintf('User input: (%f,%f)\nSmaller: (%f,%f)\nbigger: (%f,%f)\node45: (%f,%f)\n', xclosest, y(xidx), xsmallerclosest, ysmaller(xsmalleridx), xbiggerclosest, ybigg
%new figure window with a title
figure('Name', 'Approximations Reparated with Relected Point Shown');
%we again plot the ode45 solution first to get proper axis scaling
subplot(2,2,4);
plot(odex,odey, 'b');
hold on
grid on
v = axis;%store the axis limits in v
title('Graph using Matlab''s ode45 function');
plot(xodeclosest, odey(xodeidx), 'kp');%we also plot the user selected point
hold off
%We now plot the other solutions in each subplot
subplot(2,2,1);
plot(x,y, 'r');
hold on
plot(xclosest, y(xidx), 'gx');
hold off
grid on
axis([v]);%make sure each axis is the same
```

```
title('Graph using user input stepsize');

subplot(2,2,2);
plot(xsmaller,ysmaller, 'c');
grid on
axis([v]);
title('Graph using stepsize 10x smaller');
hold on
plot(xsmallerclosest, ysmaller(xsmalleridx), 'ms');
hold off

subplot(2,2,3);
plot(xbigger,ybigger, 'g');
grid on
axis([v]);
title('Graph using stepsize 10x bigger');
hold on
plot(xbiggerclosest, ybigger(xbiggeridx), 'r+');
hold off

end
```

Published with MATLAB® R2021a