```
function [root,fx,ea,iter] = bisect(func,xl,xu,es,maxit,varargin)
%bisect Find root of function using the bisection method
%   [root,fx,ea,iter] = bisect(func,xl,xu,es,maxit,p1*,p2*....)
% Inputs:
%   func - the function being evaluated
%   xl, xu - guesses, lower and upper
%   es - desired relative error (default = .0001%)
%   maxit - maximum allowable iterations (default = 50)
%   p1*,p2*... - additional parameters used by function
% Outputs:
%   root - real root
%   fx - function evaluated at root
%   ea - approximate relative error (%)
%   iter - the number of iterations performed
if nargin < 3
error('At least 3 arguments required')
end
% Var test checks if sign change detected
test = func(xl,varargin{:})*func(xu,varargin{:});
% If test > 0
if test > 0
error('No sign change over interval specified')
end
% Default Variable tests
if nargin < 4 || isempty(es)
es = 0.0001;
end
if nargin < 5 || isempty(maxit)
maxit = 50;
end
% Initialize variables
iter = 0; xr = xl; ea = 100;
while iter < maxit && ea >= es
xrold = xr;
xr = (xl + xu) / 2; % BISECTION OF INTERVAL! MEAT AND POTATOES OF FUN
iter = iter + 1;
% Think, why do you need this?
if xr ~= 0
ea = abs((xr-xrold)/xr)*100;
end
% Bisection method, old guess becomes new bracket
if func(xl,varargin{:})*func(xr,varargin{:}) < 0
xu = xr;
elseif func(xu,varargin{:})*func(xr,varargin{:}) < 0
xl = xr;
else
ea = 0;
end
end
% Set output variables
root = double(xr);
fx = double(func(xr, varargin{:}));
```

```
ea = double(ea);
iter = double(iter);
end

Error using bisect (line 16)
At least 3 arguments required
```

Published with MATLAB® R2017a