```
function[root,fx,ea,iter] = falsePosition(func,xl,xu,esmax,itermax)
%
%-------------------
% Harvinder Singh Virk, MECH-105, Last Edited - 19-FEB-2018, Time:
 10:24 AM.
% { It locates the root by joining f(xl) and f(xu) with a straight
 line. The intersection of this line with x-axis represents an
 improved estimate of the root }
%
% Input:
% func - the function being evaluated
% xl - the lower guess
% xu - the upper guess
% esmax - the desired relative error (should default to 0.0001%)
% itermax - the number of iterations desired (should default to 200)
%
% Outputs:
% root - the estimated root location
% fx - the function evaluated at the root location
% ea - the approximate relative error (%)
% iter - how many iterations were performed
if nargin == 3 % Number of the arguement.
    esmax = 0.0001; % The maximum error percentage.
    itermax = 200; % The maximum number of iteration that can run.
elseif nargin == 4 % Number of arugements to be 4.
    itermax = 200; % The maximum number of iteration. (Limiting the
 iteration).
   else
    nargin < 5; % Number of arguement 5.
    disp('time to evaluate the function') % Displaying that it's ready
 to evaluate the function to find zeros.
end

% Checking inputs for possible error:
if (func(xl)*func(xu)) >= 0 % Checking for sign change in the section.
    error('Function should have one positive and one negative
 bracket') % Throwing error if it's the same sign.
elseif xl == xu
    error('same value entered for both xl and xu'); % The value of xu
 and xl shouldn't be equal. This will throw an error if same values
 are entered.
elseif (0 > esmax) || (esmax > 100) % Error should be between 0% and
 100%.
    error('Error is outside the limit')
elseif (0 > itermax)  % The maximum iteration shouldn't be less then
 zero.
    error('Iteration is outside the limits')
end
% Running the first iteration outside the loop to attain an error
 which will be used for error estimation:
root = xu - (func(xu)*(xl-xu))/(func(xl)-func(xu)); % formula to find
 the roots.
```

```
iter = 1; % Running the first iteration.
ea = 2*abs(xl - xu)/abs(xl+xu)*100; % Formula to find the first
 approximate relative error.
fx = func(root); % The function evaluated at root location.

% Creating for loop to evaluate the function:
for iter = 2:itermax %Running itertion from 2 to desired
 iterations(itermax).
       root = xu - (func(xu)*(xl-xu))/(func(xl)-func(xu)); % Formula
 to find the roots.
fx = func(root); % The output value is the function of the root.
       ea = abs(root - xu)/abs(root)*100; % Computing for the relative
 error.
       if ea < esmax % Relative error should be less then desired
 relative error.
           break % If that's right then break the loop.
       elseif (func(root)*func(xu)) < 0 % If that condition is wrong
 then the root of the function should be less then zero.
           xl = root; % Make xl as the new root value if the value is
 positive.
       else
           xu = root; % If that value is positive then make xu as the
 new limit.
        end
end
end
```

Published with MATLAB® R2018a