Course Instructor
Dr. Raymond C. Rumpf
Office: A-337
Phone: (915) 747-6958
E-Mail: rcrumpf@utep.edu

Computational Methods in EE

Topic 4a – Introduction to Root Finding & Bracketing Methods

EE 4386/5301 Computational Methods in EE

Outline

Computational Methods in EE

- Introduction
- Bracketing Methods
  - The Bisection Method
  - False-Position Method
- Multiple Roots

Computational Methods in EE

Introduction

What is Root Finding?

Computational Methods in EE

What values of x does $f(x) = 0$?

$$\text{Let } f(x) = ax^2 + bx + c = 0$$

We can figure this out algebraically

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

But what if…

$$f(x) = e^x - x = 0$$

This cannot be solved analytically.
We need to use a numerical method.

Root Finding Methods

Computational Methods in EE

- Bracketing Methods
  - Finding a single root that falls within a known range.
  - Very robust
  - Must know something ahead of time.
- Open Methods
  - Trial-and-error iterative methods
  - Do not need bounds, only an initial guess.
  - More efficient than bracketing methods
  - Can be unstable and not find a solution
- Roots of Polynomials
  - Algorithm specific to polynomials
  - Physics of your problem must be fit to a polynomial
  - Able to find all roots.

Requires initial bounds

$$x_L \le x_r \le x_U$$

Requires an initial guess

$$x_r \approx x_1$$

Requires multiple points

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$$

Generalizing Root Finding Algorithms

Computational Methods in EE

Root finding algorithms find all values of x such that $f(x) = 0$.

What if we wish to find all values of x such that $f(x) = a$?

Generalization

$$f(x) = a$$

$$f(x) - a = 0$$

$$\text{Let } g(x) = f(x) - a$$

Now perform standard root finding on $g(x)$.

Preliminary Root Location

Computational Methods in EE

The basic root finding algorithms all require that a root be roughly located. The root finding algorithm only refines the rough location.

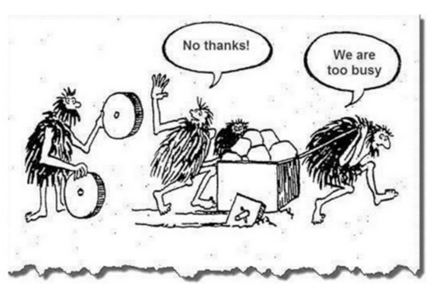
This means we must determine at least approximate locations of the roots. This information can come from one of several things.

1. Something is known about the physics of the problem being solved that gives you information about where the roots should be.
2. If nothing else is known, plot the function to identify the location of the roots and feed each of those rough locations into a root-finding algorithm. You may have to generate a number of plots and play with axis scaling to find the roots.

Need for Optimized Algorithms

Computational Methods in EE

- In many circumstances, a single computation of $f(x)$ may take hours, days, or weeks! In these cases it is highly desired to minimize the total number of computations of $f(x)$.
- It is often worth the investment of a few hours to write an awesome code that runs in an hour than waiting days or weeks for the answer to come from a simple code you wrote in a few minutes.



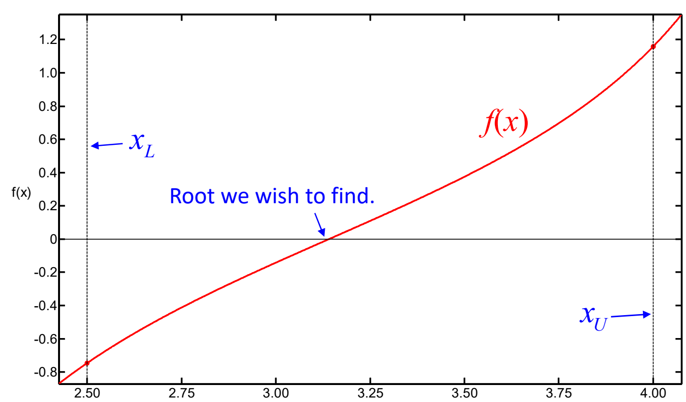
Computational Methods in EE

The Bisection Method

Step 1

Computational Methods in EE

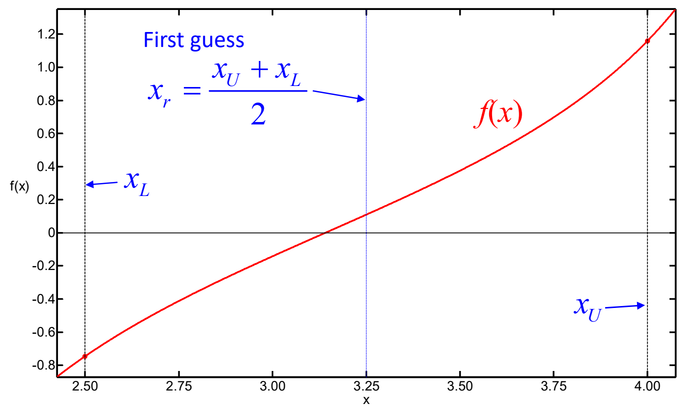
Pick a lower and upper bound, x_L and x_U that is known to contain a single root between them.



Step 2

Computational Methods in EE

Calculate the midpoint between x_L and x_U as the first guess for the root.

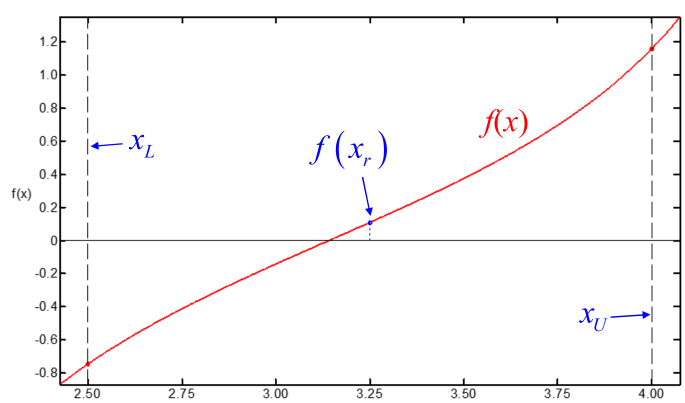


Topic 4a -- Intro & Bracketing Methods 11

Step 3

Computational Methods in EE

Calculate the function at the midpoint x_r.

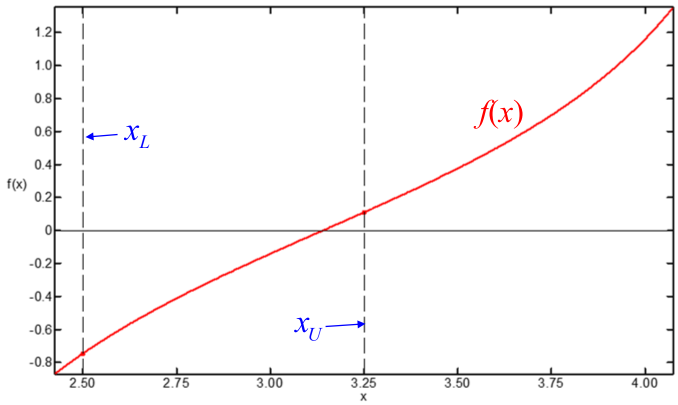


Topic 4a -- Intro & Bracketing Methods 12

Step 4

Computational Methods in EE

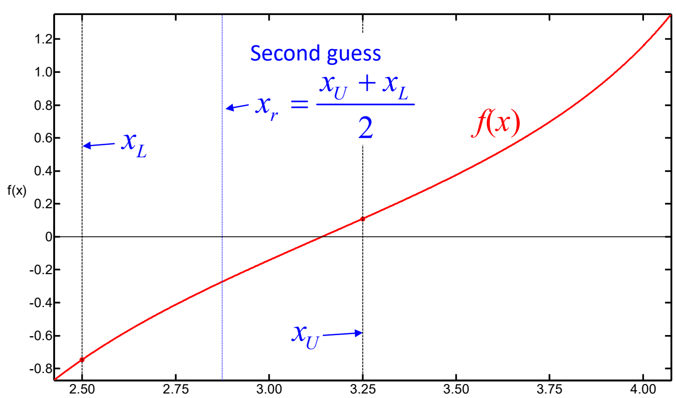
Adjust the bounds.



Step 5

Computational Methods in EE

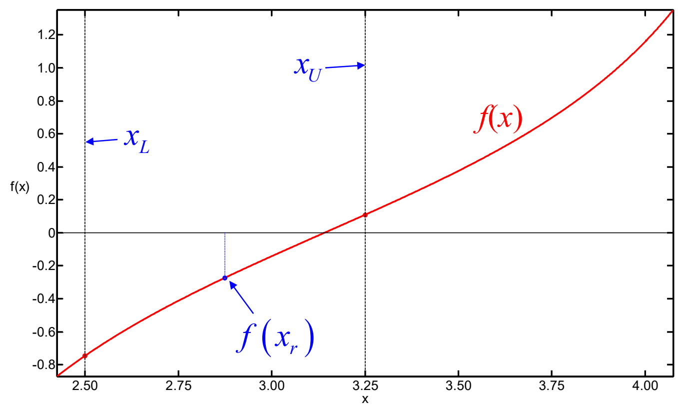
Calculate the new midpoint.



Step 6

Computational Methods in EE

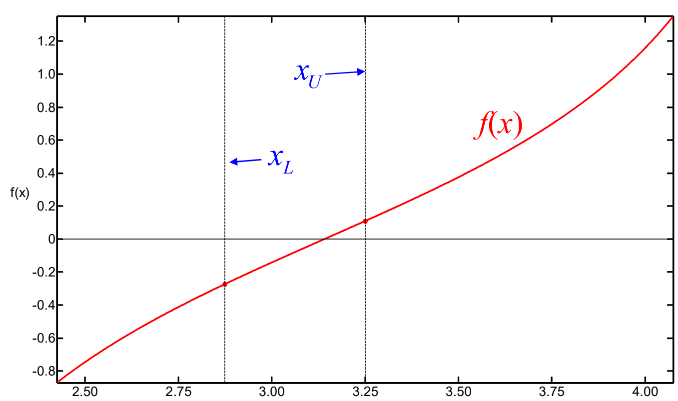
Calculate the function at the new midpoint x_r.



Step 7

Computational Methods in EE

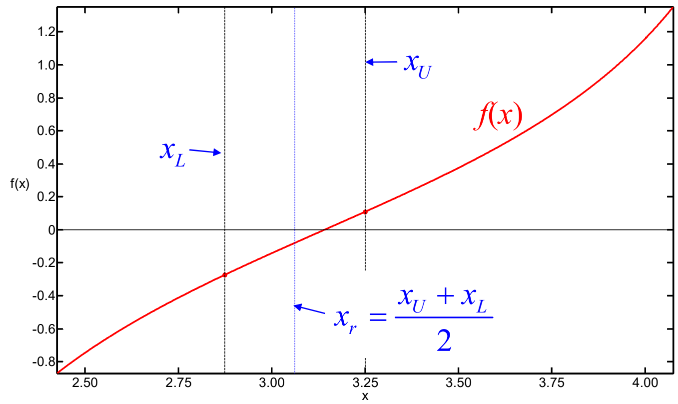
Adjust the bounds.



Step 8

Computational Methods in EE

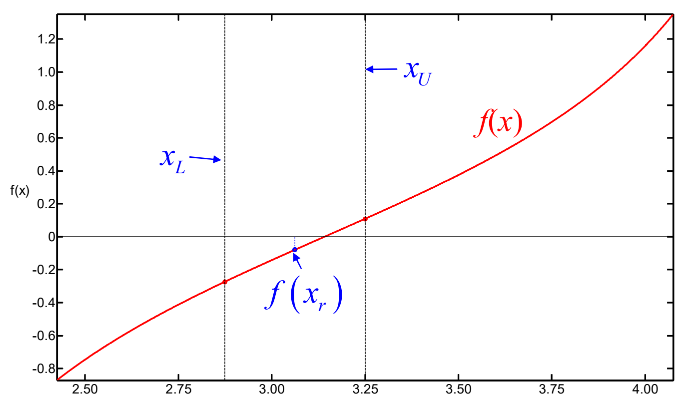
Calculate a new midpoint x_r.



Step 9

Computational Methods in EE

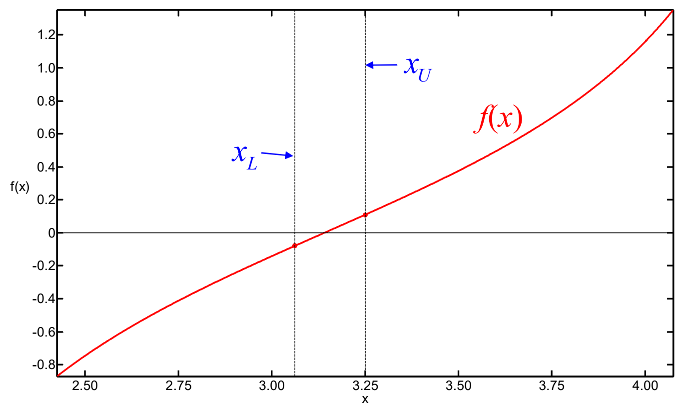
Calculate the function at the new midpoint x_r.



Step 10

Computational Methods in EE

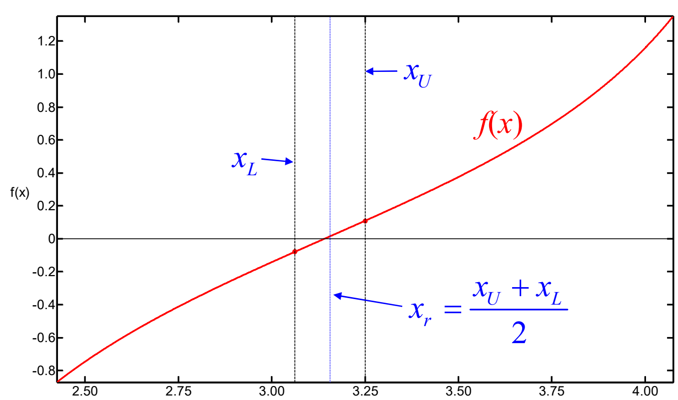
Adjust the bounds.



Step 11

Computational Methods in EE

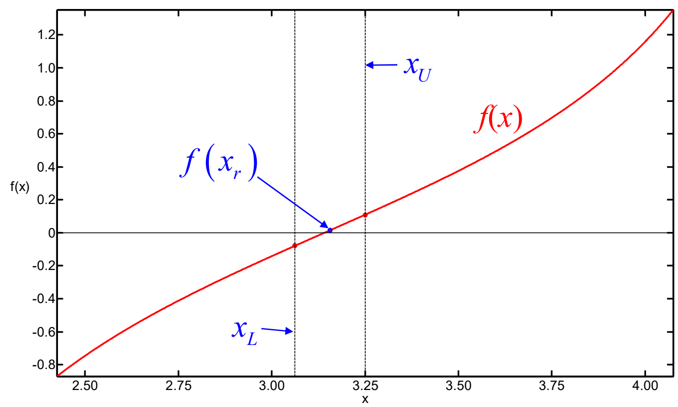
Calculate a new midpoint x_r.



Step 12

Computational Methods in EE

Calculate the function as the new midpoint.

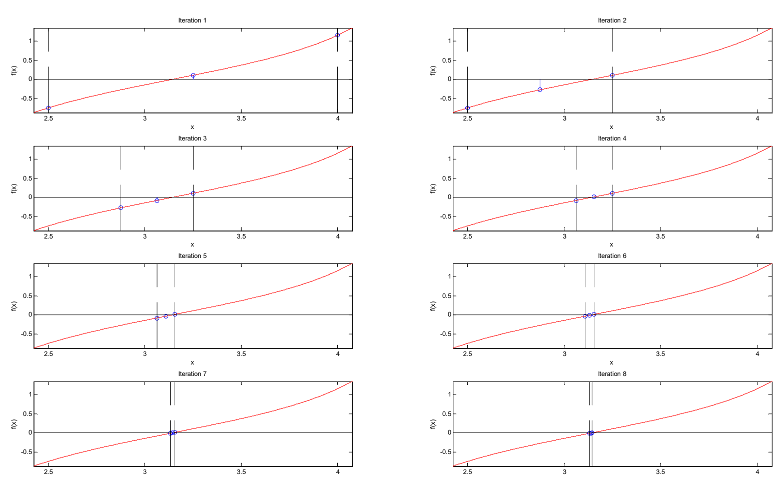


Topic 4a -- Intro & Bracketing Methods 21

Step 13 and Beyond…

Computational Methods in EE

And so on…

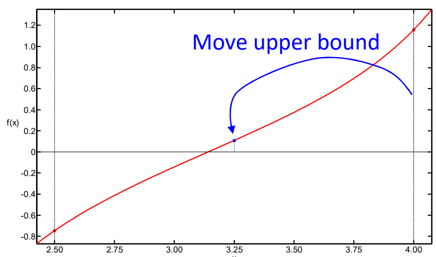


Topic 4a -- Intro & Bracketing Methods 22

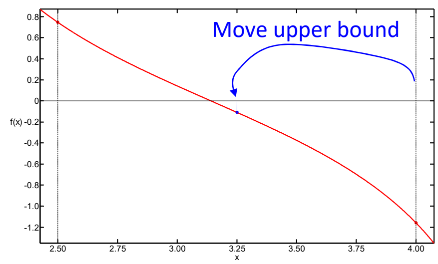
Adjusting the Bounds (1 of 2)

Computational Methods in EE

Be careful about signs when adjusting the bounds.



Here the function is positive and we move the upper bound.



Here the function is negative and we still move the upper bound.

Adjusting the Bounds (2 of 2)

Computational Methods in EE

If $f(x_L)f(x_r)<0$ then there is a sign change between x_L and x_r. This means the root is closer to x_L than x_U. Move x_U to x_r.

If $f(x_L)f(x_r)>0$ then the sign change must be between x_U and x_r. This means the root is closer to x_U than x_L. Move x_L to x_r.

If $f(x_L)f(x_r)=0$ then we are done because we have the root exactly.

```
% Adjust the Bounds
if fL*fr<0           %root toward fL
    xU = xr;
    fU = fr;
elseif fL*fr>0       %root toward fU
    xL = xr;
    fL = fr;
else
    break;
end
```

When is the Algorithm Finished?

Computational Methods in EE

i. Calculate the amount x_r has moved from one iteration to the next.

$$\delta x_r = \left| \frac{x_r^{\text{new}} - x_r^{\text{old}}}{x_r^{\text{old}}} \right|$$

ii. At the end of each iteration, check if δx_r is less than some threshold.

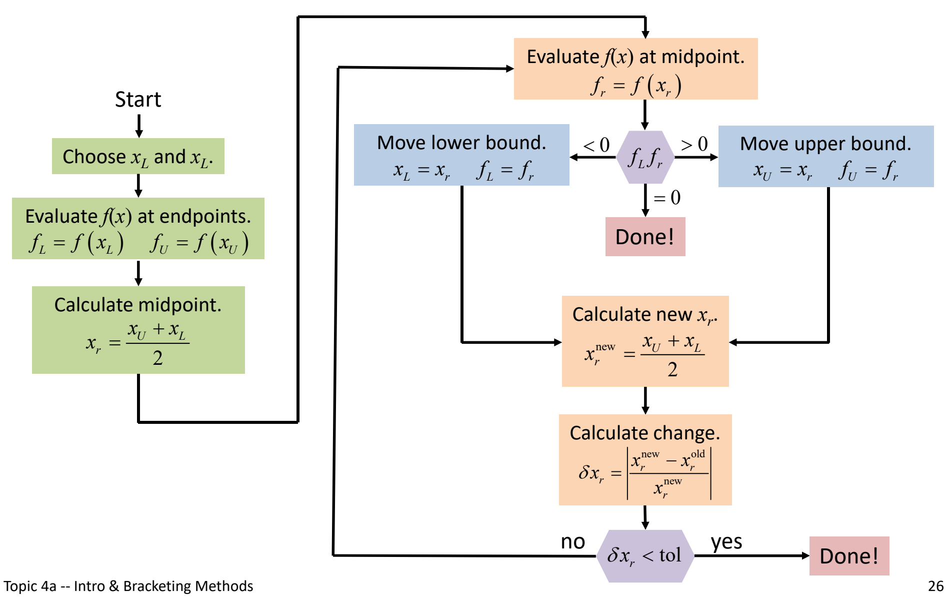
$$\delta x_r \leq \text{tolerance}$$

Rule of thumb:
If you want some number of digits of precision, ensure δx_r is at least an order of magnitude less than the desired precision.

WARNING! Do not set your convergence condition to $|x_U - x_L| < \text{tolerance}$ because this will fail when the same boundary is being adjusted each iteration.

Block Diagram of Bisection Method

Computational Methods in EE



Algorithm for Bisection Method

Computational Methods in EE

1. Choose lower and upper bounds, x_L and x_U so that they surround a root.
2. Evaluate the function at the endpoints, $f(x_L)$ and $f(x_U)$.
3. Calculate midpoint x_r.
$$x_r = \frac{x_U + x_L}{2}$$
4. Iterate until converged
   a) Evaluate the function at the midpoint $f(x_r)$.
   b) Adjust the bounds.
   $$\text{If } f(x_L) f(x_r) < 0, \text{ then } x_U = x_r$$
   $$\text{If } f(x_L) f(x_r) > 0, \text{ then } x_L = x_r$$
   $$\text{If } f(x_r) = 0, \text{ then DONE!}$$
   c) Update the midpoint.
   $$x_r = \frac{x_U + x_L}{2}$$
   d) Determine if converged
      i. Calculate step size
      $$\delta x_r = \left| \frac{x_r^{\text{new}} - x_r^{\text{old}}}{x_r^{\text{new}}} \right|$$
      ii. Algorithm is converged if $\delta x_r <$ tolerance.

Notes on Bisection Method

Computational Methods in EE

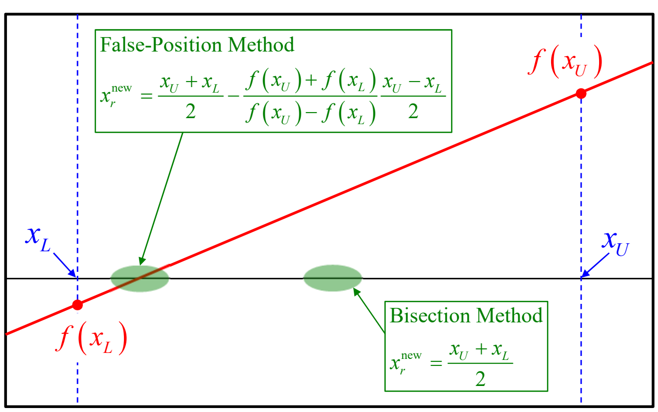
- Most robust root finder
- Least efficient root finder
- Guaranteed to find a root as long as the bounds span a crossing
- Sometimes good to check sign change of bounds
  – No sign change – 0 or even number of roots.
  – Sign change – odd number of roots.

Computational Methods in EE

False-Position Method

More Intelligent "Midpoint"

Computational Methods in EE

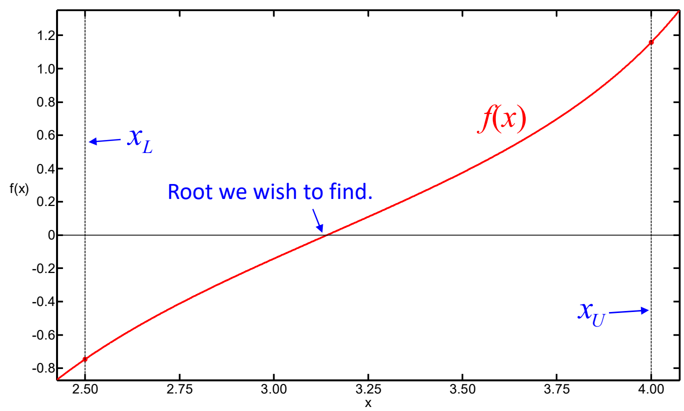


Since $|f(x_L)| << |f(x_H)|$, it is a good assumption that the root is closer to x_L than it is to x_H.

Step 1

Computational Methods in EE

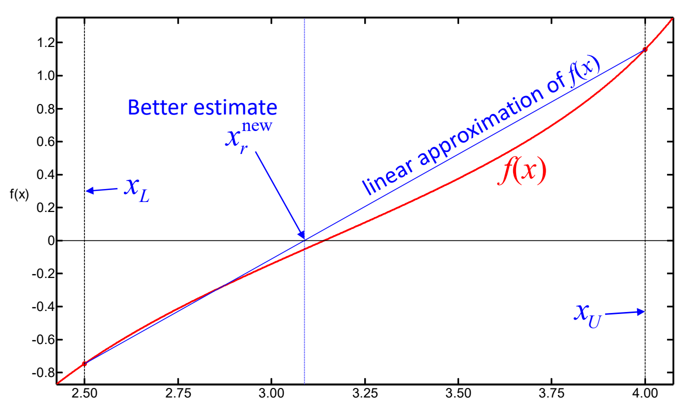
Pick a lower and upper bound, x_L and x_U that is known to contain a single root.



Step 2

Computational Methods in EE

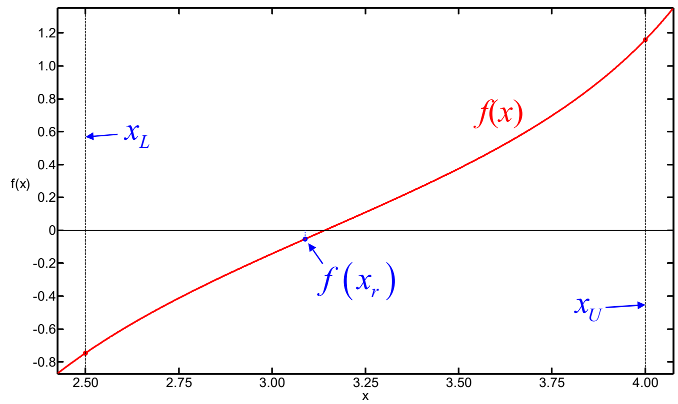
Calculate a better estimate of the position of the root using a linear approximation.



Step 3

Computational Methods in EE

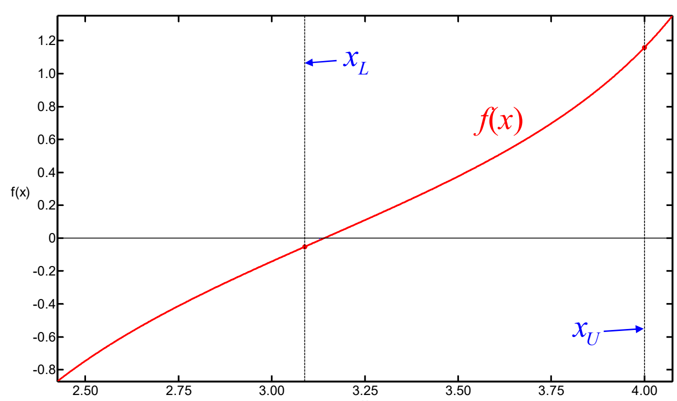
Calculate the function at the new estimate for x_r.



Step 4

Computational Methods in EE

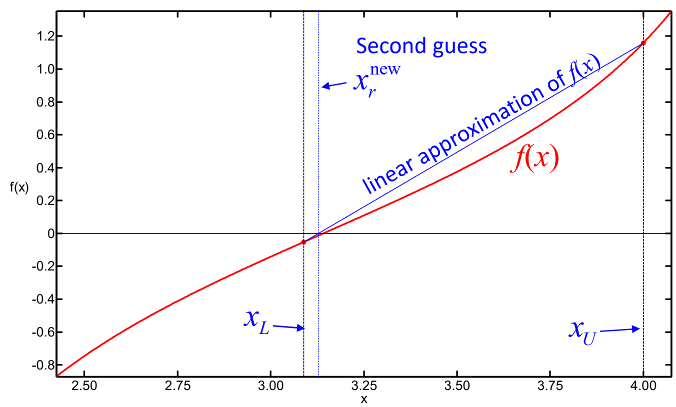
Adjust the bounds.



Step 5

Computational Methods in EE

Estimate the position of the root by linear approximation.



Derivation of Estimate (1 of 3)

Computational Methods in EE

The equation of a line given a point (x_0, y_0) and slope m is

$$(y - y_0) = m(x - x_0)$$

Assuming the function connecting the bounds is close to linear, the slope is approximately

$$m \approx \frac{f(x_U) - f(x_L)}{x_U - x_L}$$

Derivation of Estimate (2 of 3)

Computational Methods in EE

Let's choose the lower bound to be the point in our line equation.

$$x_0 = x_L \qquad y_0 = f(x_L)$$

$$y - f(x_L) = \underbrace{\frac{f(x_U) - f(x_L)}{x_U - x_L}}_{m}(x - x_L)$$

We now estimate the position of the root by setting $y = 0$ and solving for x.

$$0 - f(x_L) = \frac{f(x_U) - f(x_L)}{x_U - x_L}\left(x_r^{\text{new}} - x_L\right)$$

$$\boxed{x_r^{\text{new}} = x_L - \frac{f(x_L)}{f(x_U) - f(x_L)}(x_U - x_L)}$$

Derivation of Estimate (3 of 3)

Computational Methods in EE

Let's choose the upper bound to be the point in our line equation.

$$x_0 = x_U \qquad y_0 = f(x_U)$$

$$y - f(x_U) = \underbrace{\frac{f(x_U) - f(x_L)}{x_U - x_L}}_{m}(x - x_U)$$

We now estimate the position of the root by setting $y = 0$ and solving for x.

$$0 - f(x_U) = \frac{f(x_U) - f(x_L)}{x_U - x_L}\left(x_r^{\text{new}} - x_U\right)$$

$$\boxed{x_r^{\text{new}} = x_U - \frac{f(x_U)}{f(x_U) - f(x_L)}(x_U - x_L)}$$

Interpretation of Estimate (1 of 2)

Computational Methods in EE

We have two possible equations to estimate the position of the root.

$$x_r^{\text{new}} = x_L - \frac{f(x_L)}{f(x_U) - f(x_L)}(x_U - x_L) \qquad x_r^{\text{new}} = x_U - \frac{f(x_U)}{f(x_U) - f(x_L)}(x_U - x_L)$$

Let's average these equations.

$$x_r^{\text{new}} = \frac{\left[x_L - \frac{f(x_L)}{f(x_U) - f(x_L)}(x_U - x_L)\right] + \left[x_U - \frac{f(x_U)}{f(x_U) - f(x_L)}(x_U - x_L)\right]}{2}$$

$$= \frac{x_U + x_L}{2} - \frac{f(x_U) + f(x_L)}{f(x_U) - f(x_L)} \frac{x_U - x_L}{2}$$

Interpretation of Estimate (2 of 2)

Computational Methods in EE

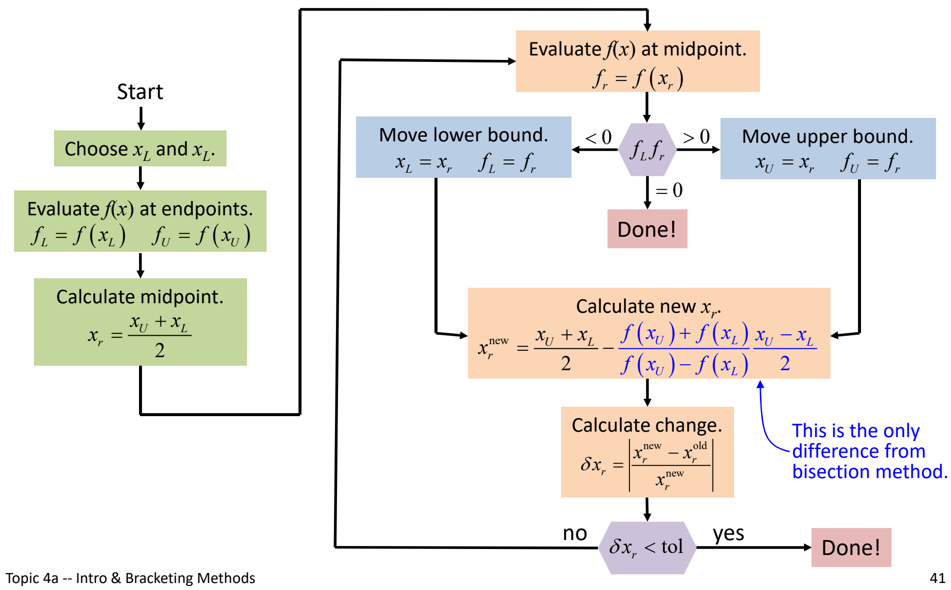
$$x_r^{\text{new}} = \underbrace{\frac{x_U + x_L}{2}}_{\text{Typical bisection method equation}} - \underbrace{\frac{f(x_U) + f(x_L)}{f(x_U) - f(x_L)} \frac{x_U - x_L}{2}}_{\text{A correction term to give a better estimate.}}$$

Typical bisection method equation

A correction term to give a better estimate.

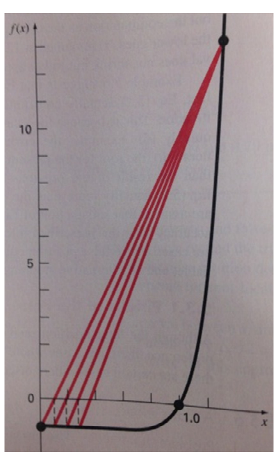
Block Diagram of False-Position Method

Computational Methods in EE



When False-Position Fails

Computational Methods in EE



The false-position method can fail or exhibit extremely slow convergence when the function is highly nonlinear between the bounds.

This happens because the estimated root is a very poor guess.

Notes on False-Position Method

Computational Methods in EE

- Very similar to bisection method
- Calculates a more intelligent "midpoint."
- Converges much faster for near linear functions.
- Converges slower for functions with abrupt curves

Computational Methods in EE

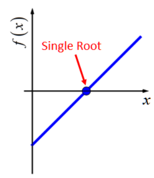
Multiple Roots

Recognizing Number of Roots

Computational Methods in EE

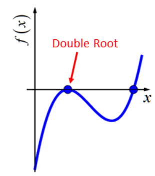
Single Root

- Sign changes on either side of root.
  $f(x_r-\delta)f(x_r+\delta)<0$
- Slope at root is not zero.
  $f'(x_r)\neq 0$



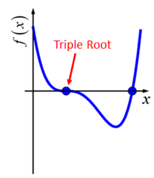
Double Root

- Sign is same on either side of root.
  $f(x_r-\delta)f(x_r+\delta)>0$
- Slope at root is zero.
  $f'(x_r)=0$
- Curvature is same on either side of root.
  $f''(x_r-\delta)f''(x_r+\delta)>0$



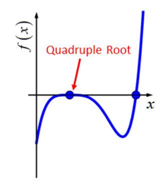
Triple Root

- Sign changes on either side of root.
  $f(x_r-\delta)f(x_r+\delta)<0$
- Slope at root is zero.
  $f'(x_r)=0$
- Curvature changes on either side of root.
  $f''(x_r-\delta)f''(x_r+\delta)<0$



Quadruple Root

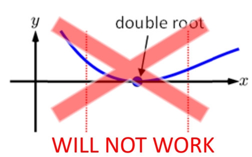
- Sign is same on either side of root.
  $f(x_r-\delta)f(x_r+\delta)>0$
- Slope at root is zero.
  $f'(x_r)=0$
- Curvature is same on either side of root.
  $f''(x_r-\delta)f''(x_r+\delta)>0$
- Curvature is broad and flat.

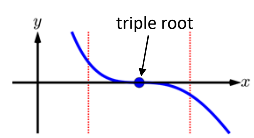


Problem with Multiple Roots

Computational Methods in EE

Bracketing methods require a sign change on either side of the root. This means they only work for odd multiple roots.

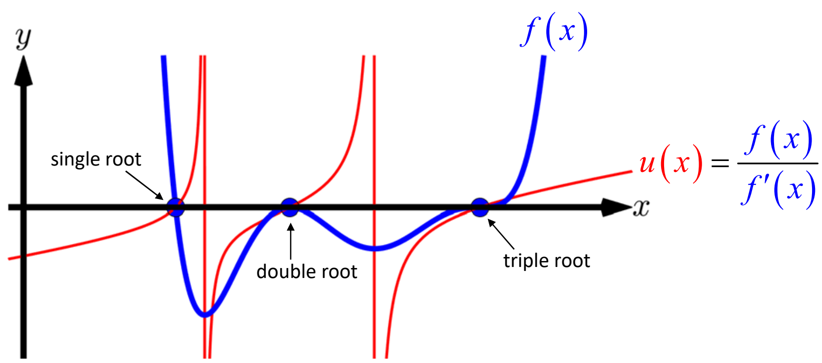




A Useful Property

Computational Methods in EE

We can define an auxiliary function $u(x)$ that will have the same roots as $f(x)$ but will always change sign on either side of any multiple root.



The Fix

Computational Methods in EE

If $f(x)$ has even multiple roots, perform root finding on the auxiliary function $u(x)$ instead.