Computational Science:
Computational Methods in Engineering

The Bisection Method

Outline

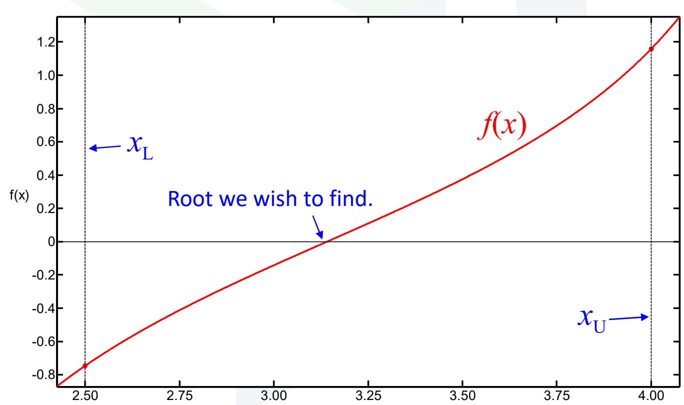
- Description of the Method
- Notes on Implementation

Description of the Method

Slide 3

Step 1

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

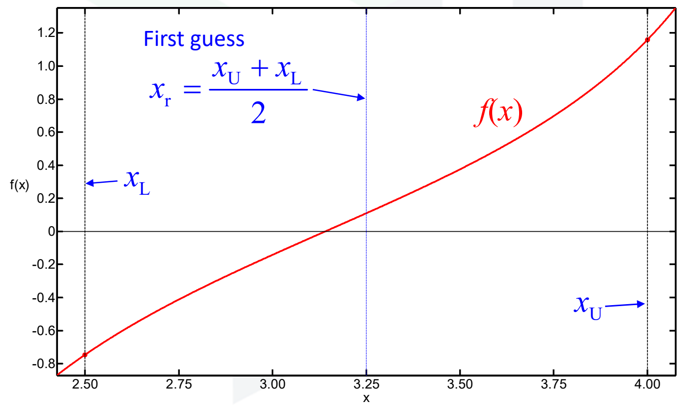


EMPossible

4

Step 2

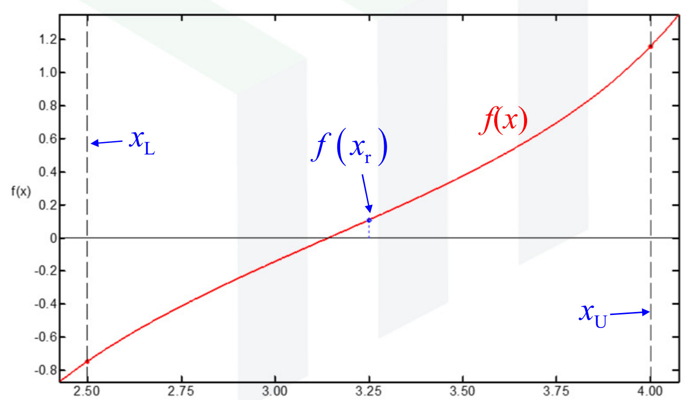
Calculate the midpoint between x_L and x_U as the first guess for the root.



EMPossible

Step 3

Calculate the function at the midpoint x_r.

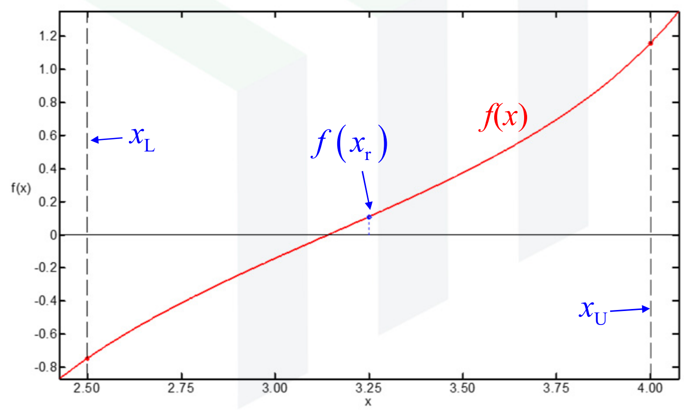


EMPossible

Step 4

Adjust the bounds.

- If $f(x_\mathrm{r}) < 0$ move lower bound x_L to x_r.
- If $f(x_\mathrm{r}) > 0$ move upper bound x_U to x_r.

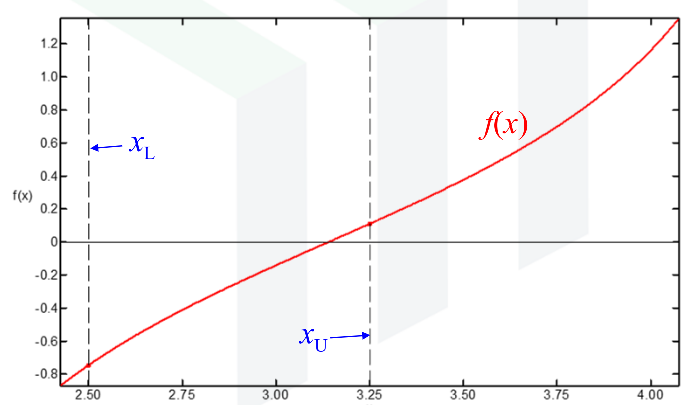


EMPossible

Step 4

Adjust the bounds.

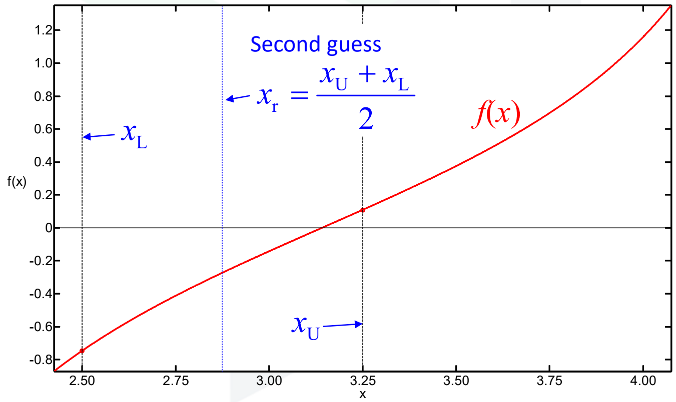
- If $f(x_\mathrm{r}) < 0$ move lower bound x_L to x_r.
- If $f(x_\mathrm{r}) > 0$ move upper bound x_U to x_r.



EMPossible

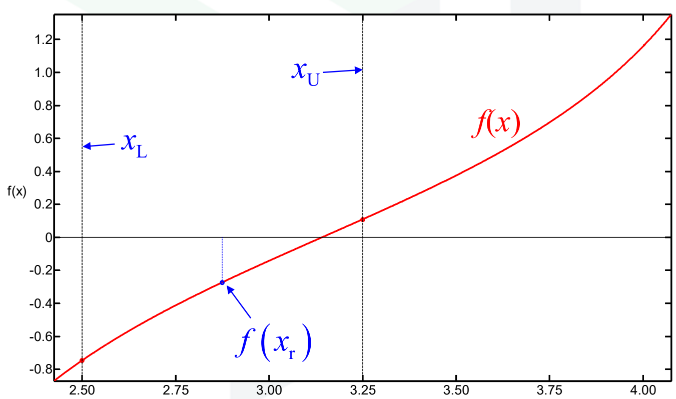
Step 5

Calculate the new midpoint.



Step 6

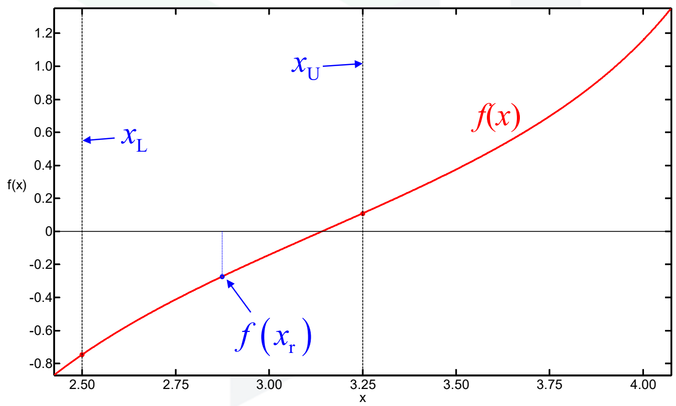
Calculate the function at the new midpoint x_r.



Step 7

Adjust the bounds.

- If $f(x_r) < 0$ move lower bound x_L to x_r.
- If $f(x_r) > 0$ move upper bound x_U to x_r.

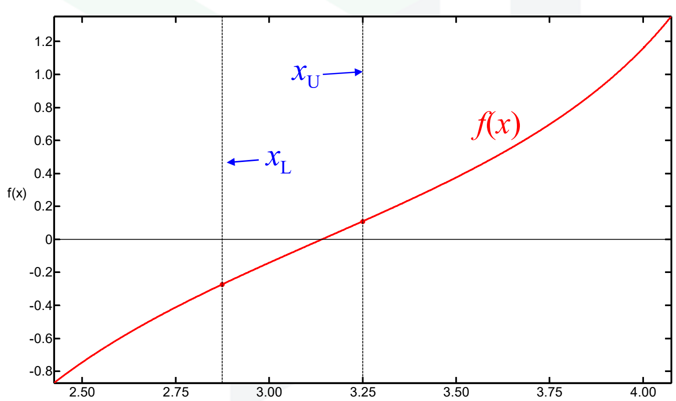


11

Step 7

Adjust the bounds.

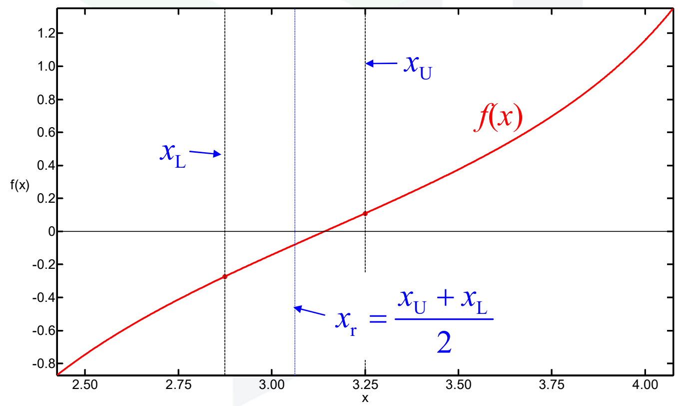
- If $f(x_r) < 0$ move lower bound x_L to x_r.
- If $f(x_r) > 0$ move upper bound x_U to x_r.



12

Step 8

Calculate a new midpoint x_r.

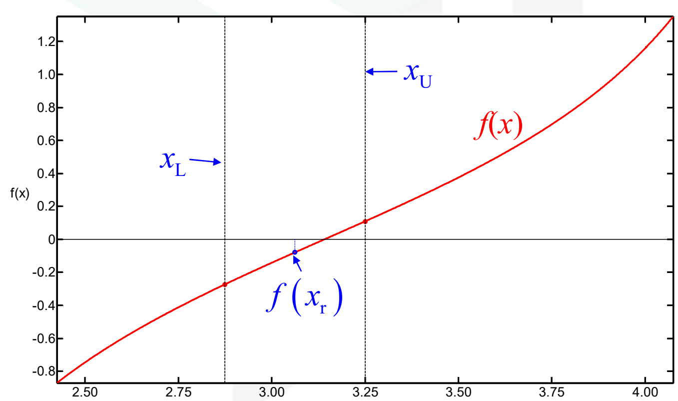


EMPossible

13

Step 9

Calculate the function at the new midpoint x_r.



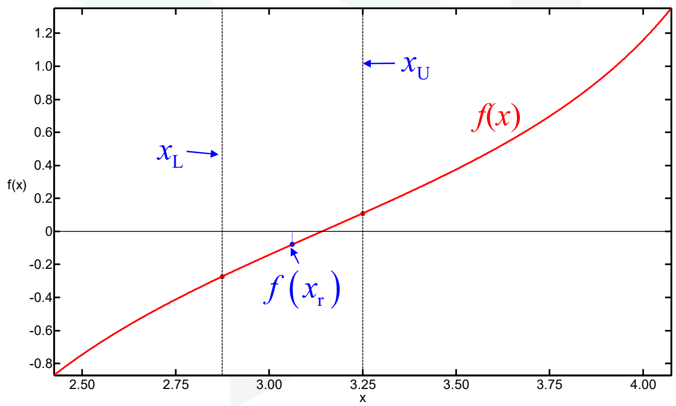
EMPossible

14

Step 10

Adjust the bounds.

- If $f(x_r) < 0$ move lower bound x_L to x_r.
- If $f(x_r) > 0$ move upper bound x_U to x_r.

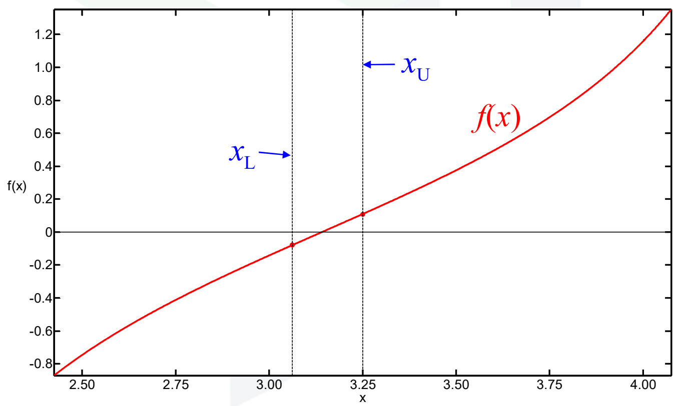


EMPossible 15

Step 10

Adjust the bounds.

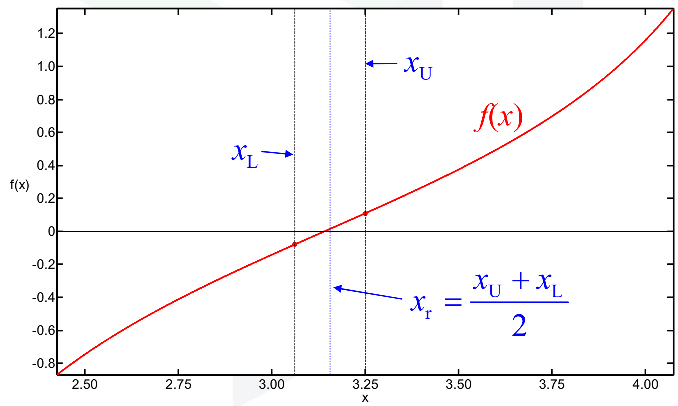
- If $f(x_r) < 0$ move lower bound x_L to x_r.
- If $f(x_r) > 0$ move upper bound x_U to x_r.



EMPossible 16

Step 11

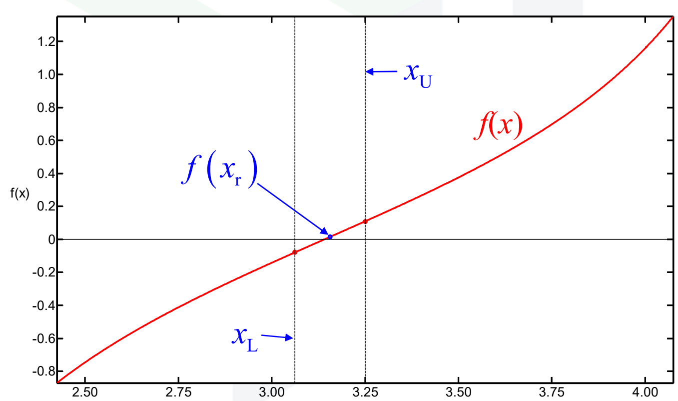
Calculate a new midpoint x_r.



EMPossible

Step 12

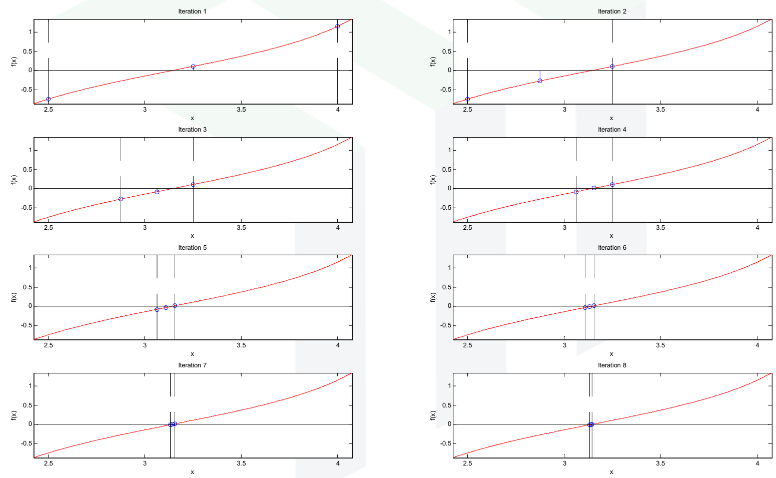
Calculate the function as the new midpoint.



EMPossible

Step 13 and Beyond…

And so on…



EMPossible

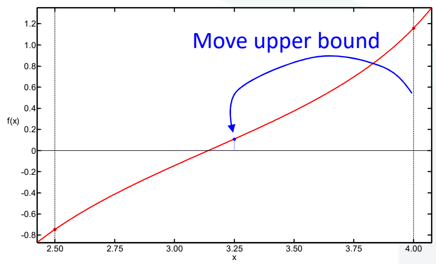
19

Notes on Implementation

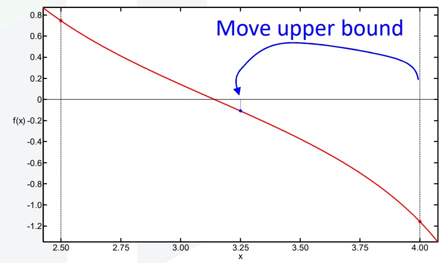
Slide 20

Adjusting the Bounds (1 of 2)

Be careful about signs when adjusting the bounds.



Here the function is positive and it is the upper bound that is moved.



Here the function is negative and it is still the upper bound that is moved.

Adjusting the Bounds (2 of 2)

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 the root is found 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?

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.

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

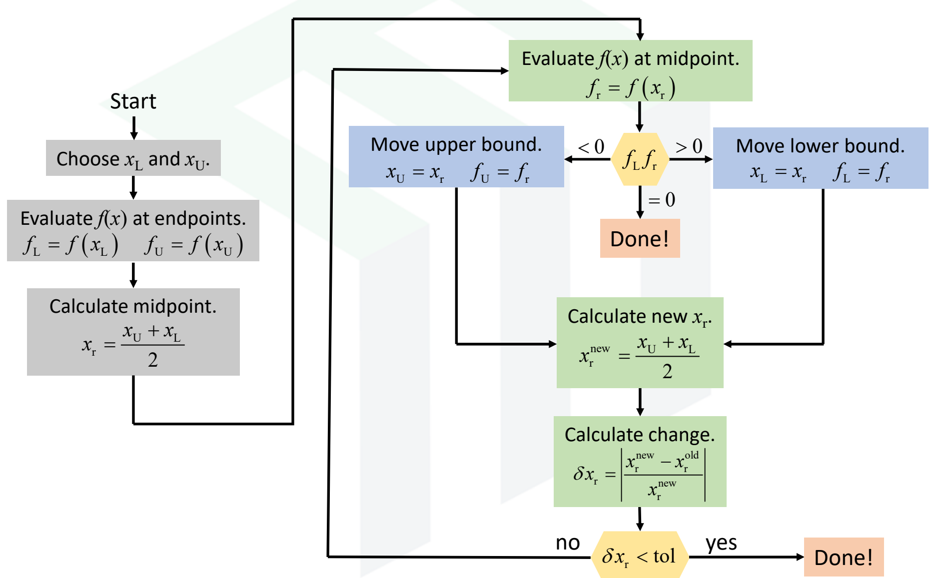
Rule of thumb:
If you want some number of digits of precision, ensure δx_r is at least an order of magnitude less than the desired precision.

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

EMPossible

23

Block Diagram of Bisection Method



EMPossible

24

Algorithm for Bisection Method

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.
      If $f(x_L)f(x_r) < 0$, then $x_U = x_r$ and $f_U = f_r$
      If $f(x_L)f(x_r) > 0$, then $x_L = x_r$ and $f_L = f_r$
      If $f(x_L)f(x_r) = 0$, then DONE!
   c) Update the midpoint.
      $$x_r = \frac{x_U + x_L}{2}$$
   d) Determine if converged
      i. Calculate step size
         $$\delta x_r = \left| \frac{x_r^{new} - x_r^{old}}{x_r^{new}} \right|$$
      ii. Algorithm is converged if $\delta x_r <$ tolerance.

EMPossible

25

Notes on Bisection Method

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

EMPossible

26