Numerical Methods ITGS219

Lecture Bisection Method

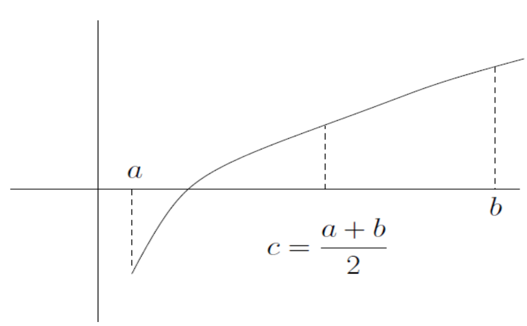
Root Finding

By: Zahra A. Elashaal

Root Finding by using Bisection Method

In the fixed point iteration method we used an initial estimate for a root: here we shall use a bracketing **interval**.

- Our initial assumption is that this **interval** contains a **single root**. In order to check this hypothesis we shall build in checks at each stage.
- If the initial interval is between *x = a* and *x = b* we know that *f(a)* and *f(b)* are *of different signs (for the interval to contain a single root): in which* case their product must be negative.
- As the name of the method suggests we bisect the interval and define the point *c = (b+a)/2.*
- *We can now evaluate the* function to obtain *f(c).*
- *This must either be positive or negative (or if we are very lucky zero).*
- *In which case f(c) will have the same sign as* either *f(a)* or *f(b).*
- *The new interval can then be defined as c and whichever end of the previous interval represents a change in sign of the function.*



Root Finding (Bisection Method

- A simple version of a code to use this method in an interval [a, b] would be

```
a = 1; b = 5;
for j = 1:10
   c = (a+b)/2;
   if f(c)*f(a) > 0
       a  = c;
    else
       b = c;
  end
end
```

If we wish to be slightly more careful we would use the codes

```
% bisect.m
function [answer, iflag] = bisect(fun, a, b)
global tolerance maxits
iflag = 0;
iterations = 0 ;
f_a = feval(fun,a);
f_b = feval(fun,b);
  while ((f_a*f_b<0) & iterations < maxits) & (b-a)>tolerance
       iterations = iterations + 1 ;
       c = (b+a)/2;
       f_c = feval(fun,c);
       if f_c*f_a<0
              b=c; f_b = f_c;
       elseif f_b*f_c < 0
              a=c; f_a = f_c;
       else
             iflag = 1; answer = c;
      end
   end
switch iterations
    case maxits
         iflag = -1; answer = NaN;
    case 0
        iflag = -2; answer = NaN;
    otherwise
        iflag = iterations; answer = c;
end
```

- where in this case *a = 1, b = 5* and the function *f(x)* is defined in a *routine f.m.*
- This method is perfectly adequate if you know that the method will ***converge*** in 10 steps, the interval [*a,* b] *actually contains a root and* ***no future iterations*** actually coincide with a root

Root Finding (Bisection Method Cont.)

```
% func.m
function [f] = func(x)
f = x-2*sin(x.^2);
```

```
% mbisect.m
global tolerance maxits
tolerance = 1e-4;
maxits = 30;
xlower = 0.4;
xupper = 0.6;
[root,iflag] = bisect('func', xlower, xupper);
switch iflag
  case -1
     disp('Root finding failed')
  case -2
     disp('Initial range does not only contain one root')
  otherwise
     disp([' Root = ' num2str(root) ...' found in '
     num2str(iflag) ' iterations'])
end
```

- This method is guaranteed to work provided only one root is in the **relevant interval** and the function is ***continuous***.
- It may work if there are three roots but this is not recommended: in fact it appears to work provided there are an odd number of roots, because each iteration may remove a number of roots which is guaranteed to be even.
- We note that the length of the interval after *n* ***iterations*** is $(b - a)/2^n$. Hence if we wish to know the *root* to within a ***given tolerance*** we can work out **how many iterations we need** to perform.
- So that, if the required ***tolerance*** is ***e***, we find that:

Number of iterations we need is: $$n > \frac{1}{\ln 2} \ln\left(\frac{b-a}{\epsilon}\right).$$

Example 4.2 To determine a root of a continuous function *f(x)* between [0,1], given that $f(0)f(1) < 0$ to within 1×10^{-4} requires $n > \ln(10^4)/\ln 2 \approx 13.28$: in other words ***14*** *iterations.*

Root Finding(Bisection Method Cont ...)

Bisection Method Example

Determine the root of the given equation $x^2-3 = 0$ for x ∈ [1,2]

- Given: $x^2-3 = 0$ => Let $f(x) = x^2-3$
- Now, find the value of $f(x)$ at a=1 and b=2.

$f(x=1) = 1^2-3 = 1 - 3 = -2 < 0$

$f(x=2) = 2^2-3 = 4 - 3 = 1 > 0$

- The given function is continuous, and the root lies in the interval [1, 2].
- Let "c" be the midpoint of the interval. => $c = (a+b)/2$

c = (1+2)/2 , c = 3 / 2 , c = 1.5

- Therefore, the value of the function at "c" is

$f(c) = f(1.5) = (1.5)^2-3 = 2.25 - 3 = -0.75 < 0$

- If f(c) < 0, let a = c. And If f(c) > 0, let b = c.
- f(c) is negative, so a is replaced with c = 1.5 for the next iterations.

The iterations for the given function are:

Iteration	a	b	c	f(a)	f(b)	f(c)
1	1	2	1.5	-2	1	-0.75
2	1.5	2	1.75	-0.75	1	0.062
3	1.5	1.75	1.625	-0.75	0.0625	-0.359
4	1.625	1.75	1.6875	-0.3594	0.0625	-0.1523
5	1.6875	1.75	1.7188	-01523	0.0625	-0.0457
6	1.7188	1.75	1.7344	-0.0457	0.0625	0.0081
7	1.7188	1.7344	1.7266	-0.0457	0.0081	-0.0189

- So, at the seventh iteration, we get the final interval [1.7266, 1.7344]
- Hence, 1.7344 is the approximated solution.

Root Finding(Bisection Method Cont ...)

Algorithm

Bisection method Steps (Rule)	
Step-1:	Find points a and b such that $a < b$ and $f(a) \cdot f(b) < 0$.
Step-2:	Take the interval $[a, b]$ and find next value $x_0 = \frac{a+b}{2}$
Step-3:	If $f(x_0) = 0$ then x_0 is an exact root, else if $f(a) \cdot f(x_0) < 0$ then $b = x_0$, else if $f(x_0) \cdot f(b) < 0$ then $a = x_0$.
Step-4:	Repeat steps 2 & 3 until $f(x_i) = 0$ or $\lvert f(x_i) \rvert \le$ Accuracy

Example-2 By using Bisection method find the root for $f(x)=x^3-x-1$ in the interval [1,2] and with acceptance error =0.0005

Solution:

Here $x^3-x-1=0$ => Let $f(x)=x^3-x-1$

Here $f(1) = -1 < 0$ and $f(2) = 5 > 0$ so the root lies between 1 and 2

as $e=0.0005$ then number of iterations is:

$$n > \frac{1}{\ln 2} \ln\left(\frac{b-a}{\epsilon}\right).$$

n > 1/ln2 * ln (1/.0005)
> 1.44*7.6 > 10.94 ∴ n = 11

$c = (a+b)/2$ = (1+2)/2 = 1.5 > 0

	a	b	c
x	0	2	1.5
$f(x)$	-1	5	0.875

Root Finding(Bisection Method Cont …)

Example-2 By using Bisection method find the root for **$f(x)=x^3-x-1$** in the interval [1,2]

n	a	$f(a)$	b	$f(b)$	$c=\frac{a+b}{2}$	$f(c)$	Update
1	1	-1	2	5	1.5	0.875	$b=c$
2	1	-1	1.5	0.875	1.25	-0.29688	$a=c$
3	1.25	-0.29688	1.5	0.875	1.375	0.22461	$b=c$
4	1.25	-0.29688	1.375	0.22461	1.3125	-0.05151	$a=c$
5	1.3125	-0.05151	1.375	0.22461	1.34375	0.08261	$b=c$
6	1.3125	-0.05151	1.34375	0.08261	1.32812	0.01458	$b=c$
7	1.3125	-0.05151	1.32812	0.01458	1.32031	-0.01871	$a=c$
8	1.32031	-0.01871	1.32812	0.01458	1.32422	-0.00213	$a=c$
9	1.32422	-0.00213	1.32812	0.01458	1.32617	0.00621	$b=c$
10	1.32422	-0.00213	1.32617	0.00621	1.3252	0.00204	$b=c$
11	1.32422	-0.00213	1.3252	0.00204	1.32471	-0.00005	$a=c$

Approximate root of the equation **$x^3-x-1=0$** using Bisection method is 1.32471