Bisection Method Algorithm

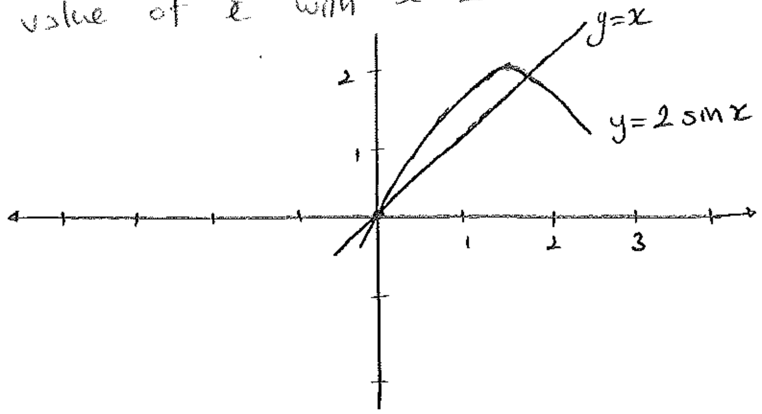
```
function root=bisection(funname,interval,tolerance,maxiter)
% Call:
% root=bisection('functionname',[lowerbound upperbound],tolerance,maxiteration)
% Input:
%       funname  : function name
%       interval : 2-dimensional array such as [lowerbound upperbound]
%       tolerance: desired accuracy
%       maxiter  : # of maximum iteration to avoid infinite loop structure
% Output:
%       root     : scalar, which is th root of given function

a=min(interval); % lowerbound
b=max(interval); % upperbound

fa=feval(funname,a); % function value at lowerbound
fb=feval(funname,b); % function value at upperbound

% Check if it satisfy the bisection method constraint %%%%%%%%%%
if fa*fb>0
    error('bisection method can not be applied to given interval');
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
i=1; % # of step
while (i<=maxiter)

    p=(a+b)/2; % p is the middle point of an and bn

    root=p; % approximated root at nth step

    fprintf('iter=%d \t a=%.5f \t b=%.5f\t p=%.5f\n',i,a,b,p);

    fp=feval(funname,p); % function value at middle point

    % control if the approx. root is close enough to the root of gives function
    if (abs(fp)<10^-12 || (b-a)/2<tolerance)% 10^-12 is random small number
        fprintf('\nthe root of the given function is:\t %.12f \n',root);
        break; % it goes out of while loop, (observe that i<maxiter)
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % if fa and fp has the same sign then change the lowerbound
    if (fa*fp>0)
        a=p;
        fa=fp;
    else % else change the upperbound
        b=p;
        fb=fp;
    end

    i=i+1; % update the number of step
end

% if the algorithm can not find the app. root in maxiter steps
% the algorithm gives the warning below
if (i>=maxiter)
    warning('method failed after given maximum iteration');
end

return
```

Exercise Set 2.1

Q.7) a) Sketch the graph of $y=x$ and $y=2\sin x$

b) Use Bisection method to find an approx. to within 10^{-5} to the first positive value of x with $x=2\sin x$.

Solution: a)



b) First define $f(x) = x - 2\sin x$, so clearly we want to find the root of $f(x)$. Write the function below in matlab

```
function fval = q7(x)
    fval = x - 2*sin(x);
return.
```

Call bisection method algorithm with accuracy: 10^{-5}, funname = 'q7', maxiter = 50, interval = [0.01 2] (since we want the first positive root)

>> root = bisection('q7', [0.01 2], 10^{-5}, 50)

(...)

at 18th iteration we will get root = 1.89549

Q.11) Let $f(x) = (x+2)(x+1)x(x-1)^3(x-2)$. To which zero of f does the following intervals Bisection method algorithm converge when applied on the following

a) $[-3, 2.5]$ c) $(-1.75, 1.5]$

Solution: First observe that f has roots as: $-2, -1, 0, 1, 2$.

a) Between -3 and 2.5 f has 5 roots: Bisection Method may not converge, there is no guarantee for that!...

Call: root = bisection('q11', [-3, 2.5], 10^{-1}, 50)

where

```
function fval = q11(x)
    fval = (x+2)*(x+1)*x*(x-1)^3*(x-2)
return
```

$\Rightarrow f(-3) < 0$, $f(2.5) > 0$; $p_1 = -0.25$

$f(p_1) < 0$: update the lower bound

Between $[-0.25, 2.5]$, f has multiple root there is no guarantee for convergence.

$f(-0.25) < 0$, $f(2.5) > 0$; $p_2 = 1.125$, $f(p_2) < 0$

since f has exactly 1-root on $[1.125, 2.5]$

so bisection method will converge to that one, which is 2

Solution: 11.c) $f(-1.75)>0$, $f(1.5)<0$, $P_1=-0.125$, $f(P_1)<0$

After first step you will get the feasible interval $[-1.75, -0.125]$ where f has exactly 1-root in, therefore bisection method will converge to -1

Q.13) Find an approx. to $\sqrt[3]{25}$ correct to within 10^{-4} using the Bisection Algorithm

Solution: Define $f(x)=x^3-25$ which has the root $\sqrt[3]{25}$. Therefore call

root = bisection ('q13', $\underbrace{[2\ \ 3]}$, 10^{-4}, 50)

naive upper and lower bound for $\sqrt[3]{25}$

where function fval = q13(x)

fval = x^3 - 25;

return

then at 14th step the approximated root is found as

root $\approx$ 2.9240

Q.14) Use Theorem 2.1 to find the bound the number of iterations needed to achieve an approximation with accuracy 10^{-3} to the solution of $x^3-x-1=0$. lying in the interval $[1,2]$. Find an approx. to the root ~~with~~ with this degree of accuracy.

Solution: First revise Theorem 2.1:

"Suppose that $f\in C[a,b]$ and $f(a)\cdot f(b)<0$. The Bisection method generates a sequence $\{p_n\}_{n=1}^{\infty}$ approximating a zero p of f with

$$|p_n-p|\le \frac{b-a}{2^n} \quad \text{when } n\geq 1."$$

Now assume at Nth iteration we have the desired accuracy $\Rightarrow$

$$|p_N-p|\le \underline{\frac{b-a}{2^N}\le 10^{-3}} \quad \text{where } a=1,\ b=2$$

if we solve the right part of the inequality we will get

$N\geq 11.5507 \Rightarrow N\approx 12.$ (observe that N is function independent!...)

Call root = bisection ('q14', [1 2], 10^{-3}, 12)

$\Rightarrow$ at 10^{th} iteration $(10\le N\approx 12)$ the approx. root = 1.325

Q.17) Let $\{p_n\}$ be the sequence defined by $p_n = \sum_{k=1}^{n} \frac{1}{k}$. Show that $\{p_n\}$ diverges even though $\lim_{n\to\infty} (p_n - p_{n-1}) = 0$

Solution: $\lim_{n\to\infty} p_n = (1) + (1/2) + \left(1/3 + 1/4\right) + \left(1/5 + 1/6 + 1/7 + 1/8\right) + \cdots$

$$> (1) + (1/2) + \left(1/4 + 1/4\right) + \left(1/8 + 1/8 + 1/8 + 1/8\right) + \cdots$$

$$= 1 + 1/2 + 1/2 + 1/2 + \cdots$$

$$= 1 + \sum_{k=1}^{\infty} 1/2 = \infty$$

$\Rightarrow \lim_{n\to\infty} p_n > \infty \iff p_n$ diverges

Fixed-Point iteration Method Algorithm

```
function fixedpoint=fixpointiter(funname,p0,tol,maxiter)
% Call:
%  fixedpoint=fixpointiter('functionname',initialvale,tolerance,maxiteration)
% Input:
%     funname     : name of the function
%     p0          : scalar, initial value
%     tol         : desired accuracy
%     maxiter     : # of maximum iteration to avoid infinite loop structure
% Output:
%     fixedpoint  : approximated fixed-point of the given function

i=1; % # of step

while i<maxiter

    p=feval(funname,p0); % p is the function value at p0
    % observe above that, in while loop, p acts as p(n) and p0 acts as p(n-1)
    fprintf('iter=%d \t p=%.12f\n',i,p);
    % control if the consecutive value of p(n) are close enough w.r.t.
    % given tolerance
    if abs(p-p0)<tol
        fixedpoint=p; % if so, we found the fixedpoint
        return;       % it stops the algorithm
    end

    i=i+1; % increment of number of step
    % if the control above is not satisfied, use p (which acts as p(n)) as the
initial point (p0) to
    % find the next p value.
    p0=p;
end
% if the algorithm can not find the app. fixed-point in maxiter steps
% the algorithm gives the message below
fprintf('\nthe method is failed after given maximum iteration\n\n');
return
```

Q.7) Use Theorem 2.3 to show that $g(x) = \pi + \frac{1}{2} \sin\left(\frac{x}{2}\right)$ has a unique fixed point on $[0, 2\pi]$. Use fixed-point iteration method to find an approx. to the fixed point that is accurate to within 10^{-2}. Use Corollary 2.5 to estimate the # of iterations required to achieve 10^{-2} accuracy, and compare this estimate to number actually needed.

Solution: First revise the thm 2.3:

" i) if $g \in C[a,b]$ and $g(x) \in [a,b]$ for all $x \in [a,b]$, then there exist at least 1-fixed point

ii) if, in addition to i), $\exists$ $k < 1$ s.t $\max_{x \in (a,b)} |g'(x)| < k$, then the fixed point is unique."

Solution 7-) Since $-1 < \sin\left(\frac{x}{2}\right) < 1$, $x \in [0, 2\pi]$

$\Leftrightarrow \quad -1/2 < 1/2 \sin\left(\frac{x}{2}\right) < 1/2$, $x \in [0, 2\pi]$

$\Leftrightarrow \quad \pi - 1/2 < \pi + 1/2 \sin\left(\frac{x}{2}\right) < \pi + 1/2$, $x \in [0, 2\pi]$

$\Leftrightarrow \quad g(x) \in [\pi - 1/2, \pi + 1/2] \quad \forall x \in [0, 2\pi]$

since $[\pi - 1/2, \pi + 1/2] \subset [0, 2\pi]$

then $g(x) \in [0, 2\pi] \quad \forall x \in [0, 2\pi]$. And clearly $g(x)$ is continuous for all x, then thm 2.3, i) is satisfied.

$|g'(x)| = \left|1/4 \cos(x/2)\right| \leq |1/4| < 1 \Rightarrow \max_{x \in [0, 2\pi]} |g'(x)| < \underset{= 1/4}{k}$ is satisfied $\begin{pmatrix}\text{thm 2.3} \\ ii)\end{pmatrix}$

$\Rightarrow$ According to thm2.3 $g(x)$ has unique fixed point in $[0, 2\pi]$. Now revise corollary 2.5: "If thm 2.4 is satisfied (fixed-point iteration converges to unique fixed point), then at nth step $|p_n - p| \leq \frac{k^n}{1-k} |p_1 - p_0|$ where k satisfies:

$\max_{x \in [a,b]} |g'(x)| < k < 1$

Now assume at Nth step, fixed point method converged

$\Leftrightarrow \quad |p_N - p| \leq \frac{k^N}{1-k} |p_1 - p_0| \leq 10^{-2}$, since we found $k = 1/4$

and for random initial point (say $p_0 = \pi/2$) then

$\frac{(1/4)^N}{1 - 1/4} \cdot |g(\pi/2) - \pi/2| \leq 10^{-2}$ gives us $N \geq 4.4\ldots$

$N \approx 5$

$\Leftrightarrow$ for $p_0 = \pi/2$, at 5 step will guarantee the convergence.

Call fixedpoint = fixedpointiter('q7b', pi/2, 10^{-2}, 5)

where function fval = q7b(x)

fval = $\pi - 1/2 \sin(x/2)$;

return

at 3rd step, the approx. fixed point is 2.65

Summary

Bisection method:

- If, for a given interval, f has more than 1 root then, Bisection method may not converge!
- Slow convergence (linear convergence)
- At every step, the feasible interval is halved.

Fixed-point method:

- Convergence speed is dependend on k which satisfies for given interval [a,b]

$$|g'(x)| < k < 1 \quad \text{for all } x \in (a,b)$$

$\Rightarrow$ smaller k increases the rate of convergence!...

- Only if such k is found, the convergence is guaranteed. But generally it is hard to find an upper bound for $|g'(x)|$ for given interval.