MA 668-Numerical Analysis I
Nonlinear Equations in One Variable

Jaman Mohebujjaman

Department of Mathematics
University of Alabama at Birmingham (UAB)

February 5, 2024

Finding Roots

We will focus on finding solutions to scalar, nonlinear equation

$$f(x) = 0, \quad x \in [a, b],$$

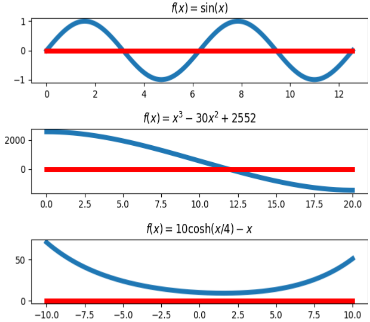
under the assumption $f \in C[a, b]$. We will denote a solution by x^*.

Example: $f(x) = x - 1$, on the interval $[a, b] = [0, 2]$, has the only one solution $x^* = 1$.

Finding Roots

Examples:

1. $f(x) = \sin(x)$, the solutions are $x^* = n\pi, \quad n \in \mathbb{Z}$. It has five roots in the interval $[a, b] = [0, 4\pi]$.
2. $f(x) = x^3 - 30x^2 + 2552$, $0 \leq x \leq 20$, has one root in the given interval.
3. $f(x) = 10\cosh\left(\frac{x}{4}\right) - x, \quad x \in [-10, 10]$. No roots.



Python code

```
import numpy as np
import matplotlib.pyplot as plt

def my_pol(x):
  return x**3-30*x**2+2552

def my_cosh(x):
  return 10*np.cosh(x/4.0)-x

t1=np.linspace(0,4*np.pi,1000)
t2=np.linspace(0,20,1000)
t3=np.linspace(-10,10,1000)
```

Python code

```
fig, axs = plt.subplots(3)
axs[0].plot(t1,np.sin(t1),t1,0*t1,'r-',linewidth=5)
axs[1].plot(t2,my_pol(t2),t2,0*t2,'r-',linewidth=5)
axs[2].plot(t3,my_cosh(t3),t3,0*t3,'r-',linewidth=5)
axs[0].set_title('$f(x)=\sin(x)$')
axs[1].set_title('$f(x)=x^3-30x^2+2552$')
axs[2].set_title('$f(x)=10\cosh(x/4)-x$')
plt.tight_layout()
plt.show()
```

Iterative Methods for Finding Roots

- Initial guess x_0:
  - Plot $f(x)$.
  - Look for locations where $f(x)$ changes sign: If $f(a) \cdot f(b) < 0$ then $f(x)$ has a root in $[a, b]$.
- Generates a sequence of iterates $\{x_1, x_2, \cdots, x_k, \cdots\}$
- Stopping criteria: $|x_n - x_{n-1}| < atol$, and/or $\frac{|x_n - x_{n-1}|}{|x_n|} < rtol$, and/or $|f(x_n)| < ftol$, where *atol*, *rtol*, and *ftol* are user-specified constants.
  - Often the relative criterion is more robust than the absolute one.
  - a favorite combination uses $\frac{|x_n - x_{n-1}|}{1+|x_n|} < tol$
- Desirable properties of root finding algorithms:
  - Efficient- requires a small number of function evaluations.
  - Robust-fails rarely, if fails it announce.
  - Require a minimal amount of additional data such as the function's derivative.
  - Requires f to satisfy only minimal smoothness properties.
  - Scalable.

Algorithm 1: Bisection Method

Given $f(x) \in C[a, b]$, with $f(a) \cdot f(b) < 0$.
Step 1: Evaluate $f(p)$ where $p = \frac{a+b}{2}$. If $|x^* - p| \leq atol$ then $x^* \approx p$, STOP.
Step 2: If $f(a) \cdot f(p) < 0$:
$b \leftarrow p$. Go to Step 1.
else if $f(a) \cdot f(p) > 0$:
$a \leftarrow p$. Go to Step 1.
else $x^* = p$ is the root, STOP.

Pros

- Most simple method
- Requires minimal assumptions on $f(x)$

Cons

- Slow in convergence
- Difficult to generalize to higher dimensions.

Bisection Method

After a total of n iterations the algorithm is guaranteed to converge if

$$|x^* - x_n| \leq \frac{b-a}{2} 2^{-n} \leq atol,$$

which gives

$$n = \left\lceil \log_2 \left(\frac{b-a}{2\ atol} \right) \right\rceil.$$

Bisection Method

Examples:

1. Apply the bisection routine, to find the root of the function

$$f(x) = \sqrt{x} - 1.1$$

   starting from the interval $[0, 2]$ with $atol = 0.1$.

2. Write a Python code of the Bisection algorithm and find the root of $f(x) = \sqrt{x} - 1.1$ with $[a, b] = [0, 2]$ and $atol = 1.e - 8$. Show the results in a table with columns as: Iteration number, p value, and $|f(p)|$ value.
3. Write a Python code of the Bisection algorithm using the "while" loop and test it in Problem 2.
4. Write a Python code of the Bisection algorithm using the "Recursive function" and test it in Problem 2.

Fixed Point Iteration Method (FPIM)

- Unlike the Bisection method, the Fixed Point Iteration method is scalable.
- Let $f(x) = 0$ be rewritten as $x = g(x)$, e.g.,

$$x(x-1) = 0 \Leftrightarrow x = x^2.$$

- Starting from an initial guess x_0, generate a sequence $\{x_1, x_2, \cdots, x_k, \cdots\}$ as

$$x_{k+1} = g(x_k)$$

so that a **fixed point** x^* can be found such that

$$x^* = g(x^*).$$

- The convergence of the above sequence depends on the choice of $g(x)$ (not all choices are good).

Fixed Point Iteration Method

Some choice of $g(x)$ can be

- $g(x) = x - f(x)$,
- $g(x) = x + 10f(x)$,
- $g(x) = x - \frac{f(x)}{f'(x)}$, assuming $f'(x)$ exists and $f'(x) \neq 0$.

Algorithm 2: Fixed Point Iteration Method

Given $f(x) \in C[a, b]$, $?g(x) \ni f(x) = 0 \Leftrightarrow x = g(x)$.
Step 1: Initial guess x_0.
Step 2: For $k = 0, 1, 2, \cdots$

$$x_{k+1} = g(x_k)$$

continue until the termination criterion is satisfied.

Fixed Point Iteration Method

Existence and uniqueness: Let $g(x) \in C[a, b]$

1. Is there a fixed point $x^* \in [a, b]$?
2. If yes, is it unique?

Fixed Point Iteration Method

Theorem (Fixed Point)

If $g \in C[a,b]$, and $g(x) \in [a,b] \;\; \forall x \in [a,b]$, then there exists a fixed point $x^* \in [a,b]$.
If, in addition, g' exists and $\exists \underbrace{\rho}_{\in \mathbb{R}^+} < 1 \ni |g'(x)| \leq \rho \;\; \forall x \in (a,b)$, then the fixed point x^* is unique in the interval.

Fixed Point Iteration Method

Convergence:

- Does the sequence $\{x_1, x_2, \cdots, x_k, \cdots\}$ with $x_{k+1} = g(x_k)$ converges to a fixed point x^*?

Fixed Point Iteration Method

Convergence:

- Does the sequence $\{x_1, x_2, \cdots, x_k, \cdots\}$ with $x_{k+1} = g(x_k)$ converges to a fixed point x^*?

Proof.

Under the above-stated assumption, we have

$$|x_{k+1} - x^*| = |g(x_k) - g(x^*)| \overset{\text{MVT}}{=} |g'(\xi)(x_k - x^*)| \leq \underbrace{\rho}_{<1} |x_k - x^*|,$$

which is a *contraction* by the factor ρ. Thus,

$$|x_{k+1} - x^*| \leq \rho|x_k - x^*| \leq \rho^2|x_{k-1} - x^*| \leq \cdots \leq \rho^{k+1}|x_0 - x^*|.$$

Now

$$\lim_{k\to\infty} |x_{k+1} - x^*| \leq \lim_{k\to\infty} \rho^{k+1}|x_0 - x^*| \implies \lim_{k\to\infty} x_{k+1} = x^*.$$

Fixed Point Iteration Method

Example: For the function $g(x) = e^{-x}$ on $[0.2, 1.2]$, find a fixed point x^* satisfying $x^* = e^{-x^*}$ with an initial guess $x_0 = 1$.

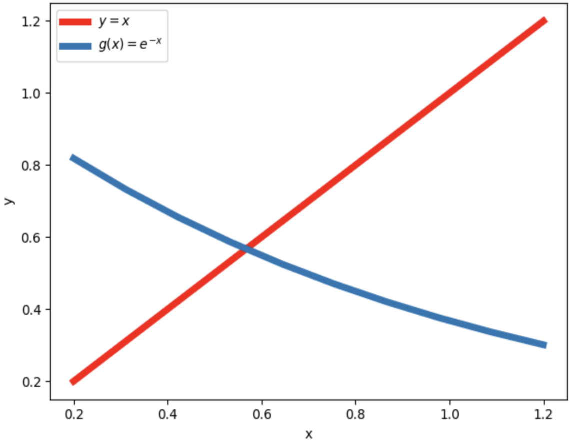


Figure: Graphs of the functions $y = x$, and $g(x) = e^{-x}$.

Fixed Point Iteration Method

Example: For the function $g(x) = e^{-x}$ on $[0.2, 1.2]$, find a fixed point x^* satisfying $x^* = e^{-x^*}$ with an initial guess $x_0 = 1$.
Solution: $g \in C[0.2, 1.2]$, monotonically decreasing, $g(0.2) \approx 0.82$, and $g(1.2) \approx 0.30$, thus $g(x) \in [0.2, 1.2] \implies \exists$ a fixed point $x^* \in [0.2, 1.2]$.

$$
\begin{aligned}
x_1 &= g(x_0) = g(1) = e^{-1} \approx 0.37 \\
x_2 &= g(x_1) = g(0.37) = e^{-0.37} \approx 0.69 \\
x_3 &= g(x_2) = g(0.69) = e^{-0.69} \approx 0.50 \\
x_4 &= g(x_3) = g(0.50) = e^{-0.50} \approx 0.61 \\
x_5 &= g(x_4) = g(0.61) = e^{-0.61} \approx 0.54 \\
x_6 &= g(x_5) = g(0.54) = e^{-0.54} \approx 0.58 \\
x_7 &= g(x_6) = g(0.58) = e^{-0.58} \approx 0.56 \\
x_8 &= g(x_7) = g(0.56) = e^{-0.56} \approx 0.57 \\
x_9 &= g(x_8) = g(0.57) = e^{-0.57} \approx 0.57.
\end{aligned}
$$

Consider $f(x) = 2\cosh(\frac{x}{4}) - x$, which has two roots at x_1^*, and x_2^*, and we can bracket them as $2 \leq x_1^* \leq 4,\ \ 8 \leq x_2^* \leq 10$.

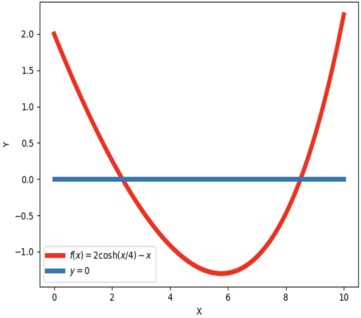


Figure: Roots of $f(x) = 2\cosh(\frac{x}{4}) - x$.

Speed of Convergence

Bisection Method: The speed of convergence with $\rho = \frac{1}{2}$. Here $a = 2$, $b = 4$, and $atol = 1e-8$.

```
Iter=  1    p= 3.0000000000  |f(p)|= 0.4106334306
Iter=  2    p= 2.5000000000  |f(p)|= 0.0964926140
Iter=  3    p= 2.2500000000  |f(p)|= 0.0748374817
Iter=  4    p= 2.3750000000  |f(p)|= 0.0119814777
.
.
.
Iter=  23   p= 2.3575508595  |f(p)|= 0.0000001338
Iter=  24   p= 2.3575509787  |f(p)|= 0.0000000517
Iter=  25   p= 2.3575510383  |f(p)|= 0.0000000107
Iter=  26   p= 2.3575510681  |f(p)|= 0.0000000098
Iter=  27   p= 2.3575510532  |f(p)|= 0.0000000005
```

We found $x_1^* = 2.35755106061697$

Speed of Convergence

Bisection Method: The speed of convergence with $\rho = \frac{1}{2}$. Here $a = 8$, $b = 10$, and $atol = 1e - 8$.

```
Iter=  1    p= 9.0000000000  |f(p)|= 0.5931350609
Iter=  2    p= 8.5000000000  |f(p)|= 0.0076695436
Iter=  3    p= 8.7500000000  |f(p)|= 0.2750998717
Iter=  4    p= 8.6250000000  |f(p)|= 0.1294401872
.
.
.
Iter=  23   p= 8.5071995258  |f(p)|= 0.0000000479
Iter=  24   p= 8.5071996450  |f(p)|= 0.0000000793
Iter=  25   p= 8.5071995854  |f(p)|= 0.0000000157
Iter=  26   p= 8.5071995556  |f(p)|= 0.0000000161
Iter=  27   p= 8.5071995705  |f(p)|= 0.0000000002
```

We found $x_2^* = 8.507199577987194$

Fixed Point Iteration Method: $g(x) = 2\cosh\left(\frac{x}{4}\right)$, $g'(x) = \frac{1}{2}\sinh(\frac{x}{4})$ near x_1^*, $|g'(x)| < 0.4$, and near x_2^*, $|g'(x)| > 1$.

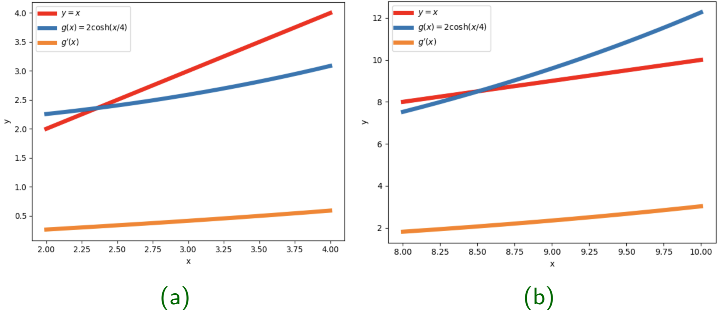


Figure: Roots of $f(x) = 2\cosh\frac{x}{4} - x$ (a) $2 \le x_1^* \le 4$, and (b) $8 \le x_2^* \le 10$.

Speed of Convergence

Fixed Point Iteration Method: Initial guess $x_0 = 2$, $atol = 1e-8$

```
n= 1    p= 2.2552519304   g(p)= 2.3263957274   |f(p)|= 0.0711437969
n= 2    p= 2.3263957274   g(p)= 2.3479003113   |f(p)|= 0.0215045840
n= 3    p= 2.3479003113   g(p)= 2.3545463562   |f(p)|= 0.0066460449
. . .
n= 9    p= 2.3575421720   g(p)= 2.3575482822   |f(p)|= 0.0000061103
n= 10   p= 2.3575482822   g(p)= 2.3575501890   |f(p)|= 0.0000019067
n= 11   p= 2.3575501890   g(p)= 2.3575507840   |f(p)|= 0.0000005950
n= 12   p= 2.3575507840   g(p)= 2.3575509697   |f(p)|= 0.0000001857
n= 13   p= 2.3575509697   g(p)= 2.3575510276   |f(p)|= 0.0000000579
n= 14   p= 2.3575510276   g(p)= 2.3575510457   |f(p)|= 0.0000000181
n= 15   p= 2.3575510457   g(p)= 2.3575510513   |f(p)|= 0.0000000056
n= 16   p= 2.3575510513   g(p)= 2.3575510531   |f(p)|= 0.0000000018
```

Speed of Convergence

Fixed Point Iteration Method: Initial guess $x_0 = 4$, $atol = 1e - 8$

```
n= 1    p= 3.0861612696   g(p)= 2.6253959730   |f(p)|= 0.4607652967
n= 2    p= 2.6253959730   g(p)= 2.4464830853   |f(p)|= 0.1789128877
n= 3    p= 2.4464830853   g(p)= 2.3858876707   |f(p)|= 0.0605954146
...
n= 10   p= 2.3575774751   g(p)= 2.3575592988   |f(p)|= 0.0000181763
n= 11   p= 2.3575592988   g(p)= 2.3575536267   |f(p)|= 0.0000056720
n= 12   p= 2.3575536267   g(p)= 2.3575518568   |f(p)|= 0.0000017700
n= 13   p= 2.3575518568   g(p)= 2.3575513044   |f(p)|= 0.0000005523
n= 14   p= 2.3575513044   g(p)= 2.3575511321   |f(p)|= 0.0000001724
n= 15   p= 2.3575511321   g(p)= 2.3575510783   |f(p)|= 0.0000000538
n= 16   p= 2.3575510783   g(p)= 2.3575510615   |f(p)|= 0.0000000168
n= 17   p= 2.3575510615   g(p)= 2.3575510563   |f(p)|= 0.0000000052
n= 18   p= 2.3575510563   g(p)= 2.3575510546   |f(p)|= 0.0000000016
```

Fixed Point Iteration Method: Initial guess $x_0 = 9$ $atol = 1e-8$

```
n= 1  p= 9.59314    g(p)= 11.09515  |f(p)|= 1.50201
n= 2  p= 11.09515   g(p)= 16.08161  |f(p)|= 4.98646
n= 3  p= 16.08161   g(p)= 55.74146  |f(p)|= 39.65986
n= 4  p= 55.74146   g(p)= 1127334.04079  |f(p)|= 1127278.29933
---------------------------------------------------------------------------
OverflowError                             Traceback (most recent call last)
```

Speed of Convergence

Fixed Point Iteration Method: Initial guess $x_0 = 8$ $atol = 1e - 8$

```
n= 1    p= 7.52439138   g(p)= 6.71312623  |f(p)|= 0.81126516
n= 2    p= 6.71312623   g(p)= 5.54303801  |f(p)|= 1.17008822
n= 3    p= 5.54303801   g(p)= 4.24799489  |f(p)|= 1.29504312
...
n= 17   p= 2.35755208   g(p)= 2.35755137  |f(p)|= 0.00000071
n= 18   p= 2.35755137   g(p)= 2.35755115  |f(p)|= 0.00000022
n= 19   p= 2.35755115   g(p)= 2.35755109  |f(p)|= 0.00000007
n= 20   p= 2.35755109   g(p)= 2.35755106  |f(p)|= 0.00000002
n= 21   p= 2.35755106   g(p)= 2.35755106  |f(p)|= 0.00000001
n= 22   p= 2.35755106   g(p)= 2.35755105  |f(p)|= 0.00000000
```

Rate of Convergence

In a fixed point iteration:

- Assume x^* is a given root and $\rho = |g'(x^*)|$ with $0 < \rho < 1$.
- x_0 is an initial guess, sufficiently close to x^*.

  $|x_k - x^*| = |g'(\xi)||x_{k-1} - x^*| \approx \rho|x_{k-1} - x^*| \approx \cdots \approx \rho^k|x_0 - x^*|$

$$\rho^{-k} \approx \frac{|x_0 - x^*|}{|x_k - x^*|}$$

$$-k\log_{10}\rho \approx \log_{10}\left(\left|\frac{x_0 - x^*}{x_k - x^*}\right|\right)$$

The rate of convergence is defined by

$$Rate = -\log_{10}\rho$$

Then, the number of iteration takes

$$k \approx \frac{\log_{10}\left(\left|\frac{x_0 - x^*}{x_k - x^*}\right|\right)}{Rate}$$

Rate of Convergence

Bisection Method: (though it is not an iterative method) $\rho = \frac{1}{2}$.

$$Rate = -\log_{10}\rho = -\log_{10}\left(\frac{1}{2}\right) \approx 0.301$$

If we want $\left|\frac{x_0 - x^*}{x_k - x^*}\right| \approx 10$, then $k = 4$. After $k = 4$, iteration, the error reduction in bisection method is a factor of $16 > 10$. For Fixed Point Iteration Method:

- For $f(x) = 2\cosh(\frac{x}{4}) - x$, $g(x) = 2\cosh(x/4)$, and $g'(x) = 0.5\sinh(x/4)$, at $x_1^* = 2.3575510513$, $\rho = g'(x_1^*) \approx 0.312$
- $Rate = -\log_{10}(0.312) \approx 0.506$
- $k = 2$.

If $\rho > 1 \implies$ Negative rate! No convergence.
If $\rho = 0 \implies$ Infinite rate $\implies$ Error reduction is faster than a constant factor.

- Given $f \in C^2[a, b]$
- Taylor series: For ξ between x, and x_k

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(\xi)(x - x_k)^2$$

$$0 = f(x_k) + f'(x_k)(x^* - x_k) + \underbrace{f''(\xi)}_{=0,\text{ if } f \text{ linear}} \frac{1}{2}(x^* - x_k)^2$$

$$x^* = x_k - \frac{f(x_k)}{f'(x_k)}$$

- If f is non-linear, and x_k is close to x^*, then

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \cdots$$

Newton's Method (NM)

Algorithm 3: Newton's Method

Given $f(x) \in C^2[a, b]$.
Step 1: Start from an initial guess x_0.
Step 2: For $k = 0, 1, 2, \cdots$, set

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

until x_{k+1} satisfies termination criteria.

Newton's Method

Example: For $f(x) = 2\cosh(\frac{x}{4}) - x$, the Newton iteration is

$$x_{k+1} = x_k - \frac{2\cosh\left(\frac{x_k}{4}\right) - x_k}{0.5\sinh\left(\frac{x_k}{4}\right) - 1}$$

```
Iter= 0    p= 8.000000000000000    |f(p)|= 0.475608617832737
Iter= 1    p= 8.584695055013547    |f(p)|= 0.084309114681618
Iter= 2    p= 8.508657714758835    |f(p)|= 0.001556683216439
Iter= 3    p= 8.507200100111358    |f(p)|= 0.000000564969103
Iter= 4    p= 8.507199570713095    |f(p)|= 0.000000000000073
Iter= 5    p= 8.507199570713027    |f(p)|= 0.000000000000002
```

Comparison

- $atol = 1e - 8$
- $x_1^* = 2.35755105$
- $x_2^* = 8.50719957$

		Number of Iterations		
x_0	Root	NM	FPIM	BSM
2	x_1^*	4	16	27
4	x_1^*	5	18	
8	x_2^*	5	DNC	27
10	x_2^*	5	DNC	

Example:

- Apply the Newton's Method, to find the root of the function

$$f(x) = \sqrt{x} - 1.1$$

starting from an initial guess $x_0 = 2$ with $atol = 0.001$.

Speed of Convergence

Assume the method converges and

$$\lim_{k \to \infty} x_k = x^*$$

then

$$\lim_{k \to \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^q} = \rho$$

- **Linear convergence:** $0 < \rho < 1$, and $q = 1$
- **Quadratic convergence:** $0 < \rho < \infty$, and $q = 2$
- **Cubic convergence:** $0 < \rho < \infty$, and $q = 3$
- **Superlinear convergence:** $\rho = 0$, and $q = 1$

Quadratic and cubic convergence are both superlinear convergence.

Relation between Newton's Method and general FPIM

Note: x^* is a root of $f(x)$.

1. If the assumptions of the Fixed Point Theorem holds, also $g'(x^*) \neq 0$, then the method converges **linearly**.
2. In this case, the size of $|g'(x^*)|$ quantifies the *rate of convergence.*
3. If $g'(x^*) = 0$, then the method may converge faster than linearly:
   - which is the case of the Newton's method with

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

   - $g'(x) =$
   - $g'(x^*) =$

Disadvantages of Newton's Method

For $k = 0, 1, 2, \cdots$, set

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

until x_{k+1} satisfies termination criteria.

1. The derivative f' need to exist, and need to evaluate it at each iteration.
2. The local nature of the method's convergence.

Secant Method

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

Algorithm 4: Secant Method

Given a scalar differential function $f(x)$:

1. Start from two initial guess x_0, and x_1.
2. For $k = 1, 2, \cdots$, set

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$$

until x_{k+1} satisfies termination criteria.

Secant Method

- $f(x) = 2\cosh(\frac{x}{4}) - x$
- $atol = 1e - 8$
- $x_1^* = 2.35755105$
- $x_2^* = 8.50719957$
- $x_0 = 10,\ x_1 = 8$

```
n= 1   p= 8.3471358106   |f(p)|= 0.1640947903
n= 2   p= 8.5299950386   |f(p)|= 0.0244655135
n= 3   p= 8.5062692341   |f(p)|= 0.0009926166
n= 4   p= 8.5071943070   |f(p)|= 0.0000056173
n= 5   p= 8.5071995719   |f(p)|= 0.0000000013
n= 6   p= 8.5071995707   |f(p)|= 0.0000000000
```

Comparison

- $f(x) = 2\cosh(\frac{x}{4}) - x$
- $atol = 1e - 8$
- $x_1^* = 2.35755105$
- $x_2^* = 8.50719957$
- $x_0 = 10,\ x_1 = 8$

			Number of Iterations		
x_0	Root	NM	FPIM	BSM	Secant
2	x_1^*	4	16	27	6
4	x_1^*	5	18		
8	x_2^*	5	DNC	27	7
10	x_2^*	5	DNC		

Theorem

Convergence of the Newton and Secant Methods If $f \in C^2[a, b]$ *and* $\exists\, x^* \in [a, b] \ni f(x^*) = 0 \wedge f'(x^*) \neq 0$, *then* $\exists\, \delta \ni$ *starting from with* x_0 *(also* x_1 *in the case of Secant Method) from anywhere in the neighborhood* $[x^* - \delta, x^* + \delta]$, *Newton's Method converges quadratically and Secant Method converges superlinearly.*

Multiple Root

$f(x^*) = 0 \wedge f'(x^*) \neq 0$ are the assumptions of Newton's Method and Secant Method.

- What if $f'(x^*) = 0$? This is the case of a **multiple root**.
- If $f(x) = (x - x^*)^m q(x)$ with $q(x^*) \neq 0$, then x^* is a root of multiplicity m.
- If $m > 1$, then $f'(x^*) = 0$.

Example: $f(x) = x^m,\ m > 1.\ x^* = 0.\ f'(x) =$ $\frac{f(x)}{f'(x)} =$
Newton's method: $x_{k+1} = x_k -$

$$|x_{k+1} - x^*| = \underbrace{\frac{m-1}{m}}_{\rho} |x_k - x^*| \ \text{:Linearly convergent}$$

Rate$= -\log_{10}\left(\frac{m-1}{m}\right)$, as $m \to \infty$, Rate$\to 0$ For $f(x) = x^2$, $m = 2$, then $\rho = \frac{1}{2}$.

- Globalizing Newton's method
- Convergence and roundoff errors