Nonlinear Equations

Lecturer:

T. Basaruddin & Heru Suhartanto

Motivation

- Most real-world problems are non-linear in nature;
  - Exponential growth or decay of nature
  - Periodic/trigonometric behavior of wave/signal
  - Dynamic problems in physical phenomenon such as
- More complicated than linear ones
  - Cannot be solved in a finite number of steps (generally)

Non Linear Problems

The general form of a non-linear problem can be represented as: Find *x* satisfying

$$F(x) = 0; \quad x \in R^n, F : R^n \to R^n$$

The solution *x** is called **root**. If *n*=1 we have a scalar problem.

<u>Possible cases</u>

1. No solution
2. Finite # solution
3. Infinite # solution

Examples

1. $x^2 - 3x - 4 = 0$; easy

2. $e^{-x} - 2x = 0$; not easy, but you can still figure it out

3. $\begin{cases} 2xy - x\cos(y) = 1 \\ x\ln(y) + ye^{x} = 0 \end{cases}$; very hard even to imagine

Existence of solutions

- In general, the existence of solution is more difficult to established than that the linear problem
- Continuity and sign-change is sufficient condition for the existence of solution in scalar problem
- No simple analogy for *n*-dimensional problems

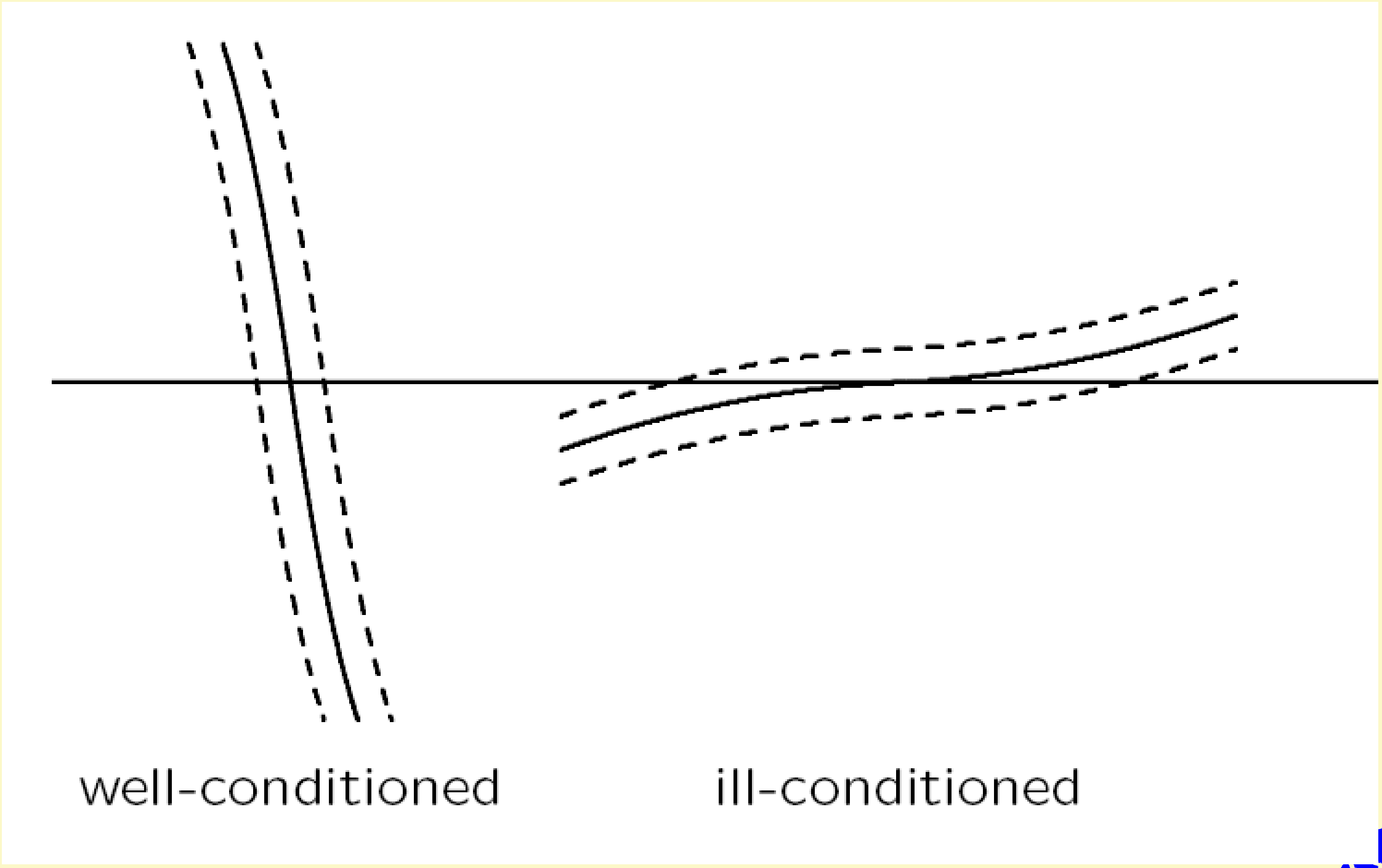
Examples

- $\exp(x) + 1 = 0$ has no solution
- $\exp(-x) - x = 0$ has one solution
- $x^2 - 4\sin(x) = 0$ has two solutions
- $x^3 + 6x^2 + 11x - 6 = 0$ has three solutions
- $\sin(x) = 0$ has infinitely many solutions

Sensitivity and Conditioning

- Conditioning of root finding problem opposite to that for evaluating function
- Absolute condition number of root finding problem for root x^* of $f : R \to R$ is $1 / | f'(x^*) |$
- Root ill-conditioned if tangent line nearly horizontal

Sensitivity and Conditioning



Method of solving

- No algorithm can solve NLP exactly in a finite number of steps
- Use iterative approach
  - Start with an initial guess
  - Re-iterate till convergence occur
- Issues
  - Convergence may not occur
  - If it does, it may be slow or too slow
  - In case of multiple roots, which one ?

More formal representation

Take a scalar NLP: Find x satisfying $f(x)=0$

Let x^0 be the initial guess, based on which we generate the sequence $\{x^{(k)}\}$

It is desirable if the sequence converges to some x^* with $f(x^*)=0$, i.e.

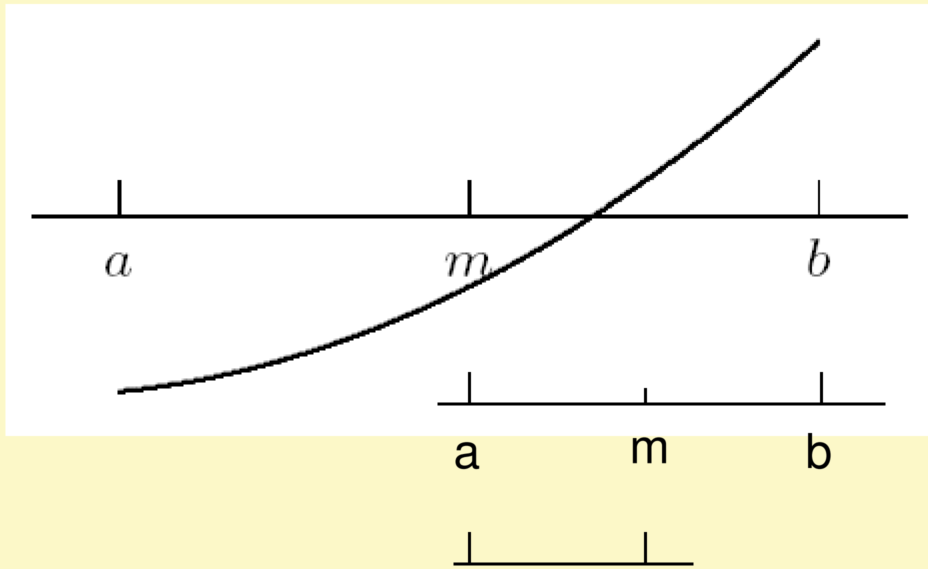
$$\lim_{k\to\infty} x^k = x^* \ \& \quad f(x^*)=0$$

Define $e_k = x^k - x^*$

We want $$\lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k|^p} = C$$

For some C independent of k and $p \geq 1$. p is referred to as the rate of convergent

Some heuristic approaches



Bi-Section

while $((b-a) > tol)$ **do**

$m = a + (b-a)/2$

if $\text{sign}(f(a)) = \text{sign}(f(m))$ **then**

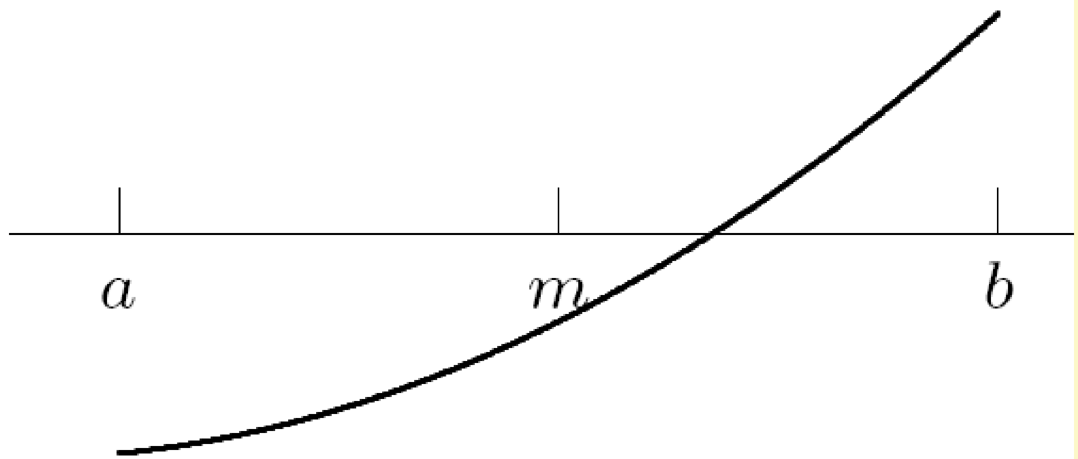
$a = m$

else

$b = m$

end

end



Convergence very slow: require $n > {}^{2}\log \frac{(b-a)}{Tol}$

Fixed Point

- Fixed point of given function $g : R \to R$ is value x such that

$$x = g(x)$$

- Many iterative methods for solving nonlinear equations use iteration scheme of form

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

where g is function whose fixed points are solutions for $f(x)=0$

- Scheme called *fixed-point iteration of functional iteration,* since function g applied repeatedly to initial starting value x_0
- For given equation $f(x) = 0$, may be many equivalent fixed-point problems $x = g(x)$ with different choices for g

Fixed Point (cont)

if $f(x) = x^2 - x - 2$

then fixed points of each functions

$$g(x) = x^2 - 2$$

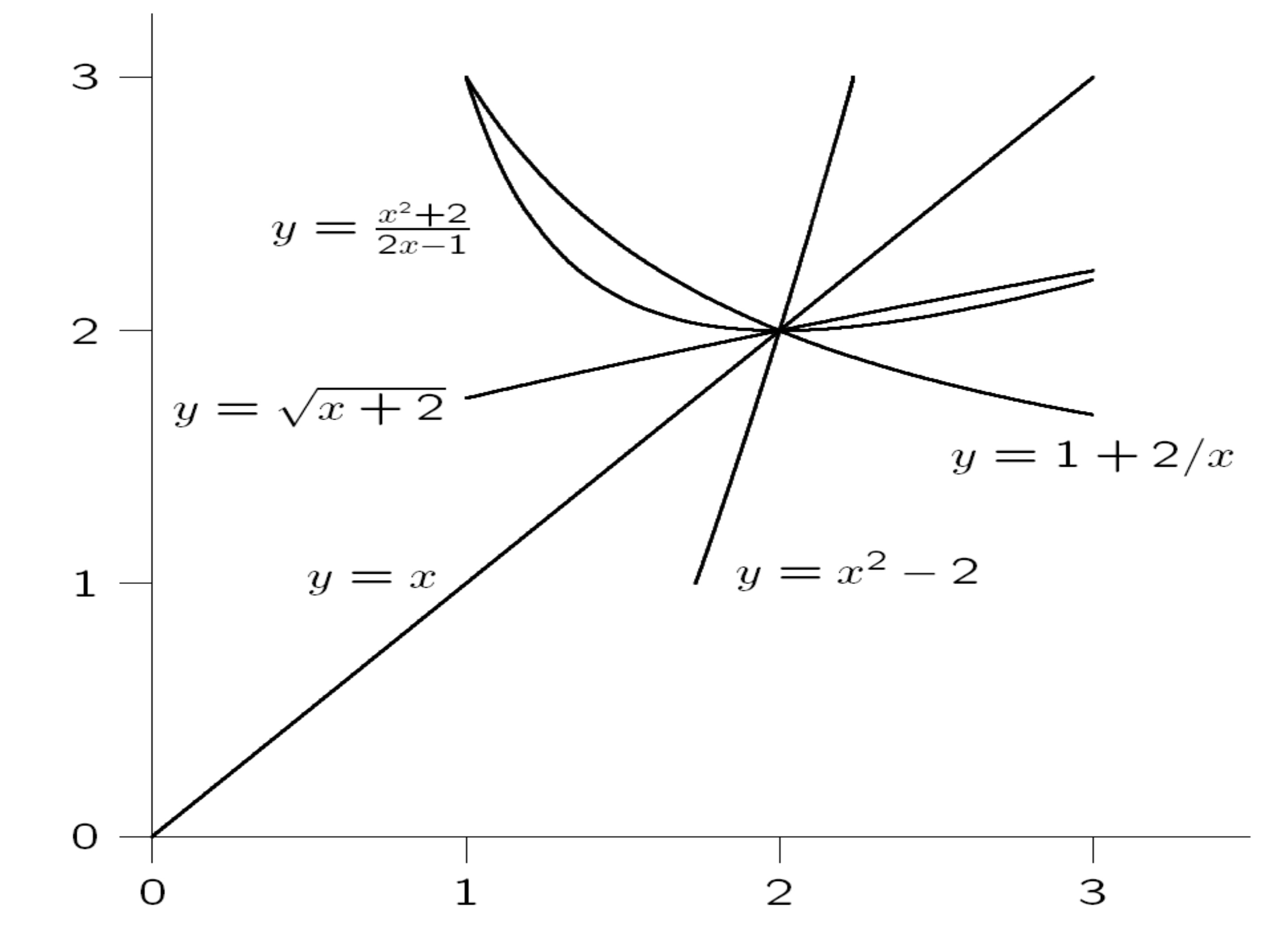
$$g(x) = \sqrt{x+2}$$

$$g(x) = 1 + 2/x$$

$$g(x) = \frac{x^2 + 2}{2x - 1}$$

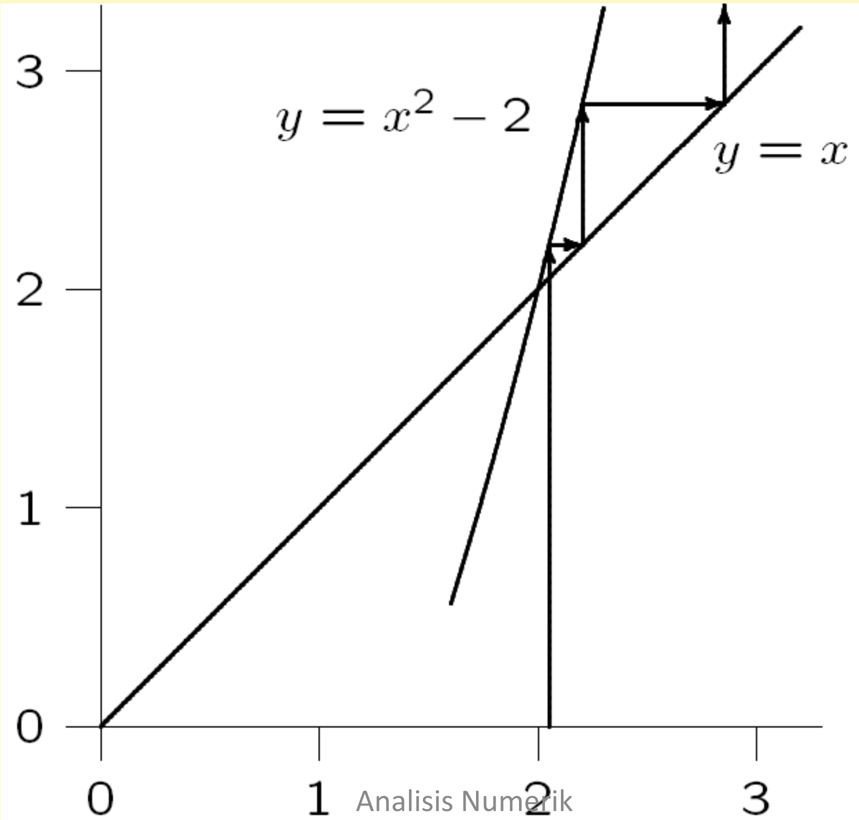
are solutions to equation $f(x) = 0$

Fixed Point (cont)



Fixed Point Iteration

$$x^{n+1} = g(x^n)$$



Convergence of FPI

- Let $e_k = x_k - x^*$ be the error at step-*k* and let *x** be the solution of *f*(*x*)=0, then

$$
\begin{aligned}
e_{k+1} &= x_{k+1} - x^* \\
&= g(x_k) - g(x^*);\ \text{note that x*=g(x*)} \\
&= g'(\widehat{x})\, e_k,\ \text{by the virtue of the MVT}
\end{aligned}
$$

- Thus, for convergence we need

$$|g'(x)| < 1$$

for all *x* close to *x**

Examples: $f(x)=x^2-x-2$, $x^*=-1$ & 2

$$g(x) = x^2 - 2$$

$$g(x) = \sqrt{x+2}$$

$$g(x) = 1 + 2/x$$

$$g(x) = \frac{x^2 + 2}{2x - 1}$$

$|g'(x)|>1$ for $|x|>0.5$ → no convg

$|g'(x)|<1$ for $x>= -1$ → convg

More on convergence of FPI

Recall that

$$e_{k+1} = x^{k+1} - x^*$$
$$= g(x^k) - x^*$$
$$= g(x^* + e_k) - x^*$$

By Taylor expansion we have

$$g(x^* + e_k) = g(x^*) + g'(x^*)e_k + \frac{g''(x^*)}{2}e_k^2 + \cdots + \frac{g^{(n)}(x^*)}{n!}e_k^n + \cdots$$

Since $g(x^*)=x^*$, we then have

$$e_{k+1} = g'(x^*)e_k + \frac{g''(x^*)}{2}e_k^2 + \cdots + \frac{g^{(n)}(x^*)}{n!}e_k^n + \cdots$$

Which means that the iteration scheme converges at rate p if

$$g'(x^*) = g''(x^*) = \cdots = g^{(p-1)}(x^*) = 0, \ \& \ g^p(x^*) \neq 0$$

Newton Method

- Based on the truncated Taylor expansion

$$f(x_k + h) = f(x_k) + hf'(x_k)$$

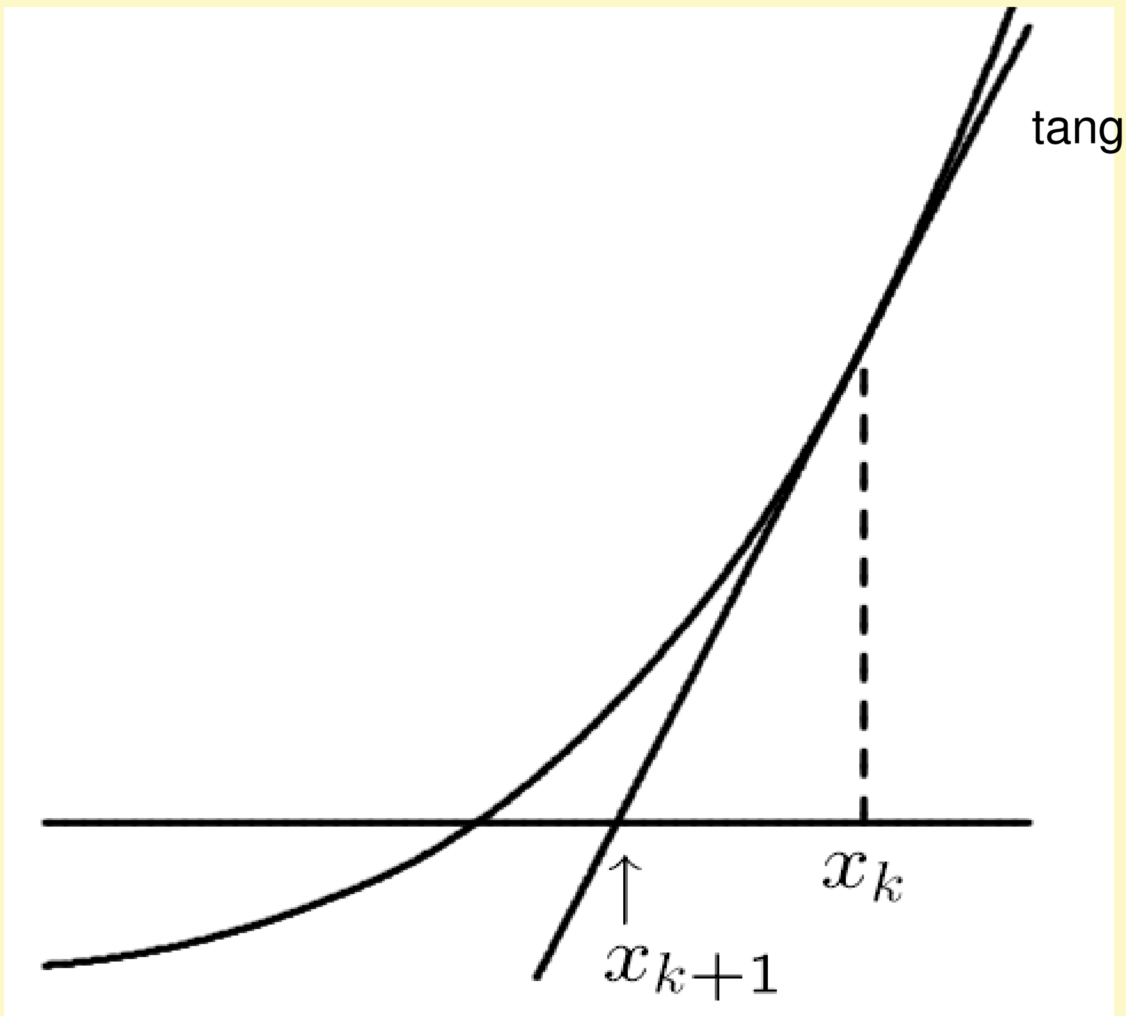
which gives rise to assuming that

$$h = {-f(x_k)}/{f'(x_k)}$$

- Thus we can take

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

Pictorially



Example

- Use Newton's method to find root of

$$f(x) = x^2 - 4\sin(x) = 0$$

Derivate is

$$f'(x) = 2x - 4\cos(x)$$

so iteration scheme is

$$x_{k+1} = x_k - \frac{x_k^2 - 4\sin(x_k)}{2x_k - 4\cos(x_k)}$$

Things that can go wrong with Newton Method

- Flat gradient: $f'(x_k)=0$ for some k
- Cyclical iteration: $x_{k+n} = x_k$ for some k & n
- Run away: horizontal asymptote

Convergence of Newton

The Newton method can be seen as a special case of the Fixed-Point, i.e.

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

In this regard, we have

$$g'(x) = 1 - \frac{[f'(x)]^2 - f(x) f''(x)}{[f'(x)]^2} = \frac{f(x) f''(x)}{[f'(x)]^2}$$

and

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

Thus, the Newton method is quadratically convergent if $f'(x^*) \neq 0$

Example: Newton vs. FPI

$f(x) = x^2 - x - 2;$ x*=2.0000

1. FPI : $x_{k+1} = \sqrt{x_k + 2}$

2. Newton: $x_{k+1} = \dfrac{x_k^2 + 2}{2x_k - 1}$

k	***x*(*k*)-FPI**	***x*(*k*)-Newton**	**(*x*(*k*)-*x**)-FPI**	**(*x*(*k*)-*x**)-Newton**
1	1.0	1.0	-1.0	-1.0
2	1.7321	3.0	-0.2679	1.0
3	1.9319	2.2	-0.0681	0.2
4	1.9829	2.0118	-0.0171	0.0118
5	1.9957	2.0000	-0.0043	0.0000

Modified Newton

- In case of multiple roots (with multiplicity m), we can use

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

  which is again quadratically convergent
- We can also approximate the first derivative using

$$(f(x_k) - f(x_{k-1}))/(x_k - x_{k-1})$$

  so that we have the so-called secant method

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

- The convergence rate is now only 1.6

Finding more than 1 root

- Use deflation method to ensure that the iteration will converge to the root different from the one already found
- If for instance x_1^* is the root of $f(x)$ that has been computed, then define

$$g(x) = f(x)/(x - x_1^*)$$

- We then compute the root of $g(x)$.

System of NLEs

- Recall a system of nonlinear equations

$$F(x) = 0;$$

$$x \in R^n,\ F : R^n \to R^n$$

- Let $J(i,k) = \dfrac{\partial F_i}{\partial x_k}$ be the Jacobian of *F*, then the Newton

iteration for solving the above problem can be written as

1. Set x_0
2. Compute $F(x_0)$ and $J(x_0)$
3. Solve $J(x_0)d = -F(x_0)$
4. Set $x_1 = x_0 + d$
5. Iterate