# 2072U Computational Science I
# Winter 2022

| Week | Topic |
|---|---|
| 1 | Introduction |
| 1–2 | Solving nonlinear equations in one variable |
| 3–4 | Solving systems of (non)linear equations |
| 5–6 | Computational complexity |
| 6–8 | Interpolation and least squares |
| 8–10 | Integration & differentiation |
| 10–12 | Additional Topics |

1. Error plotting
2. Recursion
3. Newton-Raphson iteration

## Bisection

## Bisection

- Evaluate function only.

## Bisection

- Evaluate function only.
- Need to find $a$ and $b$ such that $f$ is continuous and has unique zero on $[a, b]$.

## Bisection

- Evaluate function only.
- Need to find *a* and *b* such that *f* is continuous and has unique zero on $[a, b]$.
- Upper bound for error: $|x^{(k)} - x^*| \leq \epsilon^{(k)} = |b^{(k)} - a^{(k)}|$

## Bisection

- Evaluate function only.
- Need to find $a$ and $b$ such that $f$ is continuous and has unique zero on $[a, b]$.
- Upper bound for error: $|x^{(k)} - x^*| \leq \epsilon^{(k)} = |b^{(k)} - a^{(k)}|$
- Decreases as $\epsilon^{(k+1)} = \epsilon^{(k)}/2 \ldots$

## Bisection

- Evaluate function only.
- Need to find $a$ and $b$ such that $f$ is continuous and has unique zero on $[a, b]$.
- Upper bound for error: $|x^{(k)} - x^*| \leq \epsilon^{(k)} = |b^{(k)} - a^{(k)}|$
- Decreases as $\epsilon^{(k+1)} = \epsilon^{(k)}/2 \ldots$
- ...so that $\epsilon^{(k)} = \epsilon^{(0)}/2^k$. aka **linear** convergence.

## Bisection

- Evaluate function only.
- Need to find $a$ and $b$ such that $f$ is continuous and has unique zero on $[a, b]$.
- Upper bound for error: $|x^{(k)} - x^*| \leq \epsilon^{(k)} = |b^{(k)} - a^{(k)}|$
- Decreases as $\epsilon^{(k+1)} = \epsilon^{(k)}/2 \ldots$
- $\ldots$ so that $\epsilon^{(k)} = \epsilon^{(0)}/2^k$. aka **linear** convergence.
- Straight line on semilog plot.

## Bisection

- Evaluate function only.
- Need to find *a* and *b* such that *f* is continuous and has unique zero on $[a, b]$.
- Upper bound for error: $|x^{(k)} - x^*| \leq \epsilon^{(k)} = |b^{(k)} - a^{(k)}|$
- Decreases as $\epsilon^{(k+1)} = \epsilon^{(k)}/2 \ldots$
- $\ldots$ so that $\epsilon^{(k)} = \epsilon^{(0)}/2^k$. aka **linear** convergence.
- Straight line on semilog plot.
- Works only for one unknown.

Newton/secant

Bisection

- Evaluate function only.
- Need to find $a$ and $b$ such that $f$ is continuous and has unique zero on $[a, b]$.
- Upper bound for error: $|x^{(k)} - x^*| \leq \epsilon^{(k)} = |b^{(k)} - a^{(k)}|$
- Decreases as $\epsilon^{(k+1)} = \epsilon^{(k)}/2 \ldots$
- $\ldots$ so that $\epsilon^{(k)} = \epsilon^{(0)}/2^k$. aka **linear** convergence.
- Straight line on semilog plot.
- Works only for one unknown.

## Bisection

- Evaluate function only.
- Need to find *a* and *b* such that *f* is continuous and has unique zero on $[a, b]$.
- Upper bound for error: $|x^{(k)} - x^*| \leq \epsilon^{(k)} = |b^{(k)} - a^{(k)}|$
- Decreases as $\epsilon^{(k+1)} = \epsilon^{(k)}/2 \ldots$
- …so that $\epsilon^{(k)} = \epsilon^{(0)}/2^k$. aka **linear** convergence.
- Straight line on semilog plot.
- Works only for one unknown.

## Newton/secant

- Uses function *and derivative*/two initial points.

## Bisection

- Evaluate function only.
- Need to find $a$ and $b$ such that $f$ is continuous and has unique zero on $[a, b]$.
- Upper bound for error: $|x^{(k)} - x^*| \leq \epsilon^{(k)} = |b^{(k)} - a^{(k)}|$
- Decreases as $\epsilon^{(k+1)} = \epsilon^{(k)}/2 \ldots$
- ...so that $\epsilon^{(k)} = \epsilon^{(0)}/2^k$. aka **linear** convergence.
- Straight line on semilog plot.
- Works only for one unknown.

## Newton/secant

- Uses function *and derivative*/two initial points.
- Function must be *continuously differentiable* /continuous around $x^*$.

## Bisection

- Evaluate function only.
- Need to find $a$ and $b$ such that $f$ is continuous and has unique zero on $[a, b]$.
- Upper bound for error: $|x^{(k)} - x^*| \leq \epsilon^{(k)} = |b^{(k)} - a^{(k)}|$
- Decreases as $\epsilon^{(k+1)} = \epsilon^{(k)}/2 \ldots$
- $\ldots$ so that $\epsilon^{(k)} = \epsilon^{(0)}/2^k$. aka **linear** convergence.
- Straight line on semilog plot.
- Works only for one unknown.

## Newton/secant

- Uses function *and derivative*/two initial points.
- Function must be *continuously differentiable* /continuous around $x^*$.
- Need one/two initial guesses.

## Bisection

- Evaluate function only.
- Need to find $a$ and $b$ such that $f$ is continuous and has unique zero on $[a, b]$.
- Upper bound for error: $|x^{(k)} - x^*| \leq \epsilon^{(k)} = |b^{(k)} - a^{(k)}|$
- Decreases as $\epsilon^{(k+1)} = \epsilon^{(k)}/2 \ldots$
- …so that $\epsilon^{(k)} = \epsilon^{(0)}/2^k$. aka **linear** convergence.
- Straight line on semilog plot.
- Works only for one unknown.

## Newton/secant

- Uses function *and derivative*/two initial points.
- Function must be *continuously differentiable* /continuous around $x^*$.
- Need one/two initial guesses.
- Error estimate: $\epsilon^{(k)} \approx |\delta x^{(k)}|$.

## Bisection

- Evaluate function only.
- Need to find $a$ and $b$ such that $f$ is continuous and has unique zero on $[a, b]$.
- Upper bound for error: $|x^{(k)} - x^*| \leq \epsilon^{(k)} = |b^{(k)} - a^{(k)}|$
- Decreases as $\epsilon^{(k+1)} = \epsilon^{(k)}/2 \ldots$
- ...so that $\epsilon^{(k)} = \epsilon^{(0)}/2^k$. aka **linear** convergence.
- Straight line on semilog plot.
- Works only for one unknown.

## Newton/secant

- Uses function *and derivative*/two initial points.
- Function must be *continuously differentiable* /continuous around $x^*$.
- Need one/two initial guesses.
- Error estimate: $\epsilon^{(k)} \approx |\delta x^{(k)}|$.
- Decreases as $\epsilon^{(k+1)} \approx \left(\epsilon^{(k)}\right)^2 \ldots$

## Bisection

- Evaluate function only.
- Need to find $a$ and $b$ such that $f$ is continuous and has unique zero on $[a, b]$.
- Upper bound for error: $|x^{(k)} - x^*| \leq \epsilon^{(k)} = |b^{(k)} - a^{(k)}|$
- Decreases as $\epsilon^{(k+1)} = \epsilon^{(k)}/2 \ldots$
- $\ldots$so that $\epsilon^{(k)} = \epsilon^{(0)}/2^k$. aka **linear** convergence.
- Straight line on semilog plot.
- Works only for one unknown.

## Newton/secant

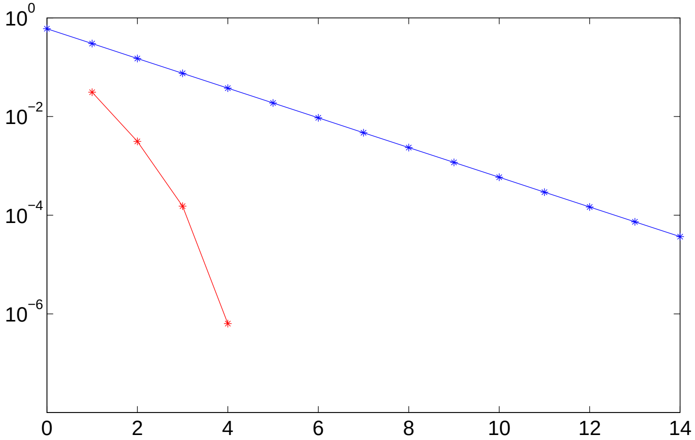
- Uses function *and derivative*/two initial points.
- Function must be *continuously differentiable* /continuous around $x^*$.
- Need one/two initial guesses.
- Error estimate: $\epsilon^{(k)} \approx |\delta x^{(k)}|$.
- Decreases as $\epsilon^{(k+1)} \approx \left(\epsilon^{(k)}\right)^2 \ldots$
- $\ldots$so that $\epsilon^{(k)} \approx (\epsilon^{(0)})^{2^k}$. aka **quadratic** convergence.

## Bisection

- Evaluate function only.
- Need to find $a$ and $b$ such that $f$ is continuous and has unique zero on $[a, b]$.
- Upper bound for error: $|x^{(k)} - x^*| \leq \epsilon^{(k)} = |b^{(k)} - a^{(k)}|$
- Decreases as $\epsilon^{(k+1)} = \epsilon^{(k)}/2 \dots$
- $\dots$ so that $\epsilon^{(k)} = \epsilon^{(0)}/2^k$. aka **linear** convergence.
- Straight line on semilog plot.
- Works only for one unknown.

## Newton/secant

- Uses function *and derivative*/two initial points.
- Function must be *continuously differentiable* /continuous around $x^*$.
- Need one/two initial guesses.
- Error estimate: $\epsilon^{(k)} \approx |\delta x^{(k)}|$.
- Decreases as $\epsilon^{(k+1)} \approx \left(\epsilon^{(k)}\right)^2 \dots$
- $\dots$ so that $\epsilon^{(k)} \approx (\epsilon^{(0)})^{2^k}$. aka **quadratic** convergence.
- Steeper than linear on semilog plot.



$10^{0}$
$10^{-2}$
$10^{-4}$
$10^{-6}$
0
2
4
6
8
10
12
14

Simple example of *recursive* programming:

```
def bisection(a, b, kMax, epsX, epsF):
    mid = (a+b)/2 ; x = mid
    err = abs(a-b)/2
    fm = func(mid)
    res = abs(fmid)
    if (kMax > 0):
        if err > epsX or res > epsF:
            fa = func(a)
            if (fm*fa > 0):
                a = mid
            else:
                b = mid
            x,err,res = bisection(a, b, kMax-1, epsX
                                  ,epsF)
    else:
        print("Warning: no convergence...")
    return x,err,res
```

Pros:

Pros:

- clean, concise, elegant code;
- conceptually clear.

Cons:

Pros:

- clean, concise, elegant code;
- conceptually clear.

Cons:

- code harder to follow;
- uses more memory;
- small mistake can give bad crash!

Newton iteration can be generalized to $n$ equations with $n$ unknowns.
Alternative derivation in 1D:

$$f(x+\delta x) \approx f(x) + f'(x)\delta x = 0 \quad \Rightarrow \quad \delta x = -\frac{f(x)}{f'(x)}$$

Now in 2D. We want to find $x_1$ and $x_2$ such that

$$f_1(x_1, x_2) = 0$$
$$f_1(x_1, x_2) = 0$$

Note that, in general, we need the same number of equations and unknowns to find (isolated) solutions...

$$f_1(x_1+\delta x_1, x_2+\delta x_2) \approx f_1(x_1,x_2) + \frac{\partial f_1}{\partial x_1}(x_1,x_2)\delta x_1 + \frac{\partial f_1}{\partial x_2}(x_1,x_2)\delta x_2$$
$$f_2(x_1+\delta x_1, x_2+\delta x_2) \approx f_2(x_1,x_2) + \frac{\partial f_2}{\partial x_1}(x_1,x_2)\delta x_1 + \frac{\partial f_2}{\partial x_2}(x_1,x_2)\delta x_2$$

which gives:

$$\begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x_1,x_2) & \frac{\partial f_1}{\partial x_2}(x_1,x_2) \\ \frac{\partial f_2}{\partial x_1}(x_1,x_2) & \frac{\partial f_2}{\partial x_2}(x_1,x_2) \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = -\begin{pmatrix} f_1(x_1,x_2) \\ f_2(x_1,x_2) \end{pmatrix}$$