**Numerical Analysis**

**Phase 1 – Root Finding**

**Algorithms**

**Dahlia Chehata Mahmoud 28**

**Moamen Raafat El-Baroudy 52**

**Mohamed Ayman Farrag 61**

**Mahmoud Mohamed Hussein 70**

**Moustafa Mahmoud Reda El-Deen 75**

1. **Algorithms & Pseudo Code:**
   1. **Bisection:**
      1. **Description:**

The Bisection method is an algorithm the repeatedly bisects an interval into two halves and then select a half in which a root must lie in for further processing.

* + 1. **Pros:**
       1. The Basic idea is easy and robust.
       2. It'll always converge to a root if the interval is valid.
       3. Number of the iterations required to attain an absolute error can be computed theoretically.
    2. **Cons:**
       1. The Method may appear to be fast due to its logarithmic time complexity but It's slow compared to other methods like Newton Raphson or Regular Falsi.
       2. It's required to find a valid interval which contains an odd number of roots to begin the algorithm.
       3. It doesn't use the fact that which side has the minimum absolute value is closer to the root.
    3. **Pseudo Code:**



* 1. **Regular Falsi:**
     1. **Description:**

Regula Falsi assumes that f(x) is linear even though these methods are needed only when f(x) is not linear and usually work well anyway.

The ratio of the change in x, to the result change in y is:

Then we try to interpolate the position of the root on the x-axis but since the function doesn't represent a linear function, in most cases it won't be the root so we choose a new interval with the new point and one of the previous two points which makes the interval valid.

* + 1. **Pros:**
       1. It'll always converge to a root if the interval is.
       2. The method converges quickly and fast compared to the Bisection in most cases.
    2. **Cons:**
       1. The Method may become slow or get stuck at all in case of convex and concave curves near the x-axis if they aren’t handled separately.
       2. It's required to find a valid interval which contains an odd number of roots to begin the algorithm.
    3. **Pseudo Code:**



* 1. **Fixed Point:** 
     1. **Description:**

a fixed point (sometimes shortened to fixpoint, also known as an invariant point) of a function is an element of the function's domain that is mapped to itself by the function. That is to say, c is a fixed point of the function f(x) if and only if f(c) = c.

Not all functions have fixed points: for example, if f is a function defined on the real numbers as f(x) = x + 1, then it has no fixed points, since x is never equal to x + 1 for any real number. In graphical terms, a fixed point means the point (x, f(x)) is on the line y = x, or in other words the graph of f has a point in common with that line.

* + 1. **Pros:**
       1. There can be multiple valid g(x) for a function f(x) where they can give multiple options and some converge faster.
    2. **Cons:**
       1. The Method doesn’t guarantee convergence where upon the g(x) it may sometimes diverge.
    3. **Pseudo Code:**



* 1. **Newton Raphson:** 
     1. **Description:**

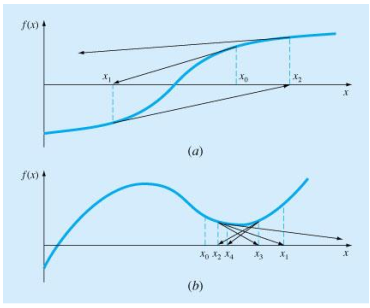
The Newton–Raphson method in one variable is implemented as follows:

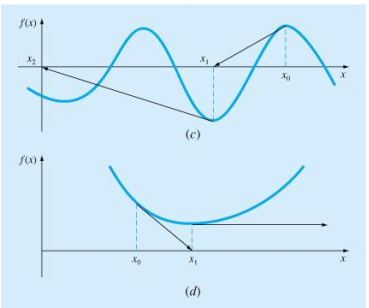
The method starts with a function f defined over the real numbers x, the function's derivative f ′, and an initial guess x0 for a root of the function f. If the function satisfies the assumptions made in the derivation of the formula and the initial guess is close, then a better approximation x1 is

Geometrically, (x1, 0) is the intersection of the x-axis and the tangent of the graph of f at (x0, f (x0)).

* + 1. **Pros:**
       1. When the initial guess is near the root, the method usually converges.
       2. The Method converges Quadratically when it doesn’t diverge.
       3. Knowledge of the multiplicity of the roots can speed the convergence for a non-1 value multiplicity.
       4. A good algorithm can identify if the iterative process is diverging or not.
    2. **Cons:**
       1. The calculation of the derivative of the function isn’t always possible or straight forward.
       2. There’s no general convergence criteria, It depends on the nature and accuracy of the initial guess.
       3. If the first derivative isn’t well behaved near the initial guess the method may overshoot and diverge.
       4. The Method may misbehave by going into a cycle or jump to another loop.
       5. Slow convergence for roots of multiplicity other than 1.
       6. Zero slope causes division by zero because of the derivative which needs to be handled separately in the code.
    3. **Pseudo Code:**







* 1. **Secant:** 
     1. **Description:**

is a root-finding algorithm that uses a succession of roots of secant lines to better approximate a root of a function f. The secant method can be thought of as a finite difference approximation of Newton's method. However, the method was developed independently of Newton's method, and predates it by over 3,000 years.

* + 1. **Pros:**
       1. Knowledge of the multiplicity of the roots can speed the convergence for a non-1 value multiplicity.
       2. It’s very fast compared to bisection and other slow algorithms.
    2. **Cons:**
       1. There’s no general convergence criteria, It depends on the nature and accuracy of the initial guess.
       2. It requires two initial guesses instead of one in Newton Raphson.
       3. It’s slower than Newton Raphson.
    3. **Pseudo Code:**



* 1. **Birge Vieta:** 
     1. **Description:**

This is an iterative method to find a real root of the **n**th degree polynomial equation.

* + 1. **Pseudo Code:**



* 1. **General Algorithm(Illnois):**