10.4.4.2.1

EE24BTECH11049 - Patnam Shariq Faraz Muhammed

Question:

Find the values of k for which the quadratic equation $2x^2 + kx + 5 = 0$. So they've two equal roots.

Solution:

Given Equation:

$$2x^2 + kx + 5 = 0 \tag{0.1}$$

Numerical Solution:

- If the roots are equal then the value of Discriminant is to 0

$$b^2 - 4ac = 0 \tag{0.2}$$

$$k^2 - 4 \times 2 \times 5 = 0 \tag{0.3}$$

$$k = \pm 2\sqrt{10} \tag{0.4}$$

We get two equations since there are two values for k.

$$2x^2 + 2\sqrt{10}x + 5 = 0 \tag{0.5}$$

$$2x^2 - 2\sqrt{10}x + 5 = 0 \tag{0.6}$$

- They've a single root that is $\frac{-b}{2a}$.
  - root of (0.5) $= -\sqrt{\frac{5}{2}}$
  - root of (0.6) $= +\sqrt{\frac{5}{2}}$

Computational Solutions:

1) **Fixed Point iteration**
- The fixed point iteration method in numerical analysis is used to find an approximate solution to algebraic and transcendental equations.
- **How it works?**
  - Rewrite into $X = g(x)$
  - Idea the root is $r = g(r)$, $r =$ fixed Point
  - initial guess $= x_0$, compute $g(x_0)$
  - Hopefully $x_1 = g(x_0)$ is closer to r (Occurs when it's convergence)
  - Do iteration until stop criteria

$$x_{n+1} = g(x_n) \tag{1.1}$$

  - end
- **Stop Criteria:**

$$|x_{n+1} - x_n| \leq tolerence \tag{1.2}$$
$$|g(x_n) - x_n| \leq tolerence \tag{1.3}$$

- This algorithm succeeds with proper choice of $g(x)$ and must be convergent
- **Error Analysis:**
  - Let r be the root, that is, $r = g(r)$
  - Iteration:

$$x_{n+1} = g(x_n) \tag{1.4}$$

  - Error e:

$$e_n = |x_n - r| \tag{1.5}$$
$$e_{n+1} = |x_{n+1} - r| \tag{1.6}$$
$$= |g(x_n) - g(r)| \tag{1.7}$$
$$= \left|g'(c)\right| |x_n - r| \text{ where, } x_n < c < r \tag{1.8}$$
$$= \left|g'(c)\right| e_n \tag{1.9}$$

  - Observation:
    * If $|g'(c)| < 1 \implies e_{n+1} < e_n$: convergence.
    * If $|g'(c)| > 1 \implies e_{n+1} > e_n$: Divergence.
- **Convergence Condition:** There exist an interval $I = [r - c, r + c]$ for some $c > 0$ such that $|g'(x)| < 1$ on I and $x_0 \in I$

Quadratic 1: $2x^2 + 2\sqrt{10}x + 5 = 0$	**Quadratic 2:** $2x^2 - 2\sqrt{10}x + 5 = 0$
$g_1(x) = -\left(\sqrt{10}x + \frac{5}{2x}\right)$	$g_2(x) = -\left(-\sqrt{10}x + \frac{5}{2x}\right)$
$g_3(x) = -\frac{(2x^2+5)}{2\sqrt{10}}$	$g_4(x) = \frac{(2x^2+5)}{2\sqrt{10}}$

TABLE 1: Fixed point iteration

- For the given Question there are 2 choices for each Quadratic
- For all $g(x)$, fixed point algo diverges.
- Initial guess
  Quad1: x0 = -5, x1 = -3, x2 = 0, x3 = 2
  Quad2: x0 = -2, x1 = 0, x2 = 3, x3 = 5
- **Program Output:**

Quad1:	Quad2:
g_1	g_2
Guess 1 failed for root	Guess 1 failed for root
Guess 2 failed for root	Guess 2 failed for root
Failed to detect root.	Failed to detect root.
Guess 1 failed for root	Guess 1 failed for root
Guess 2 failed for root	Guess 2 failed for root
Failed to detect root.	Failed to detect root.
Fixed-point iteration failed to converge.	Fixed-point iteration failed to converge.
g_3	g_4
Guess 1 failed for root	Guess 1 failed for root
Guess 2 failed for root	Guess 2 failed for root
Failed to detect root.	Failed to detect root.
Guess 1 failed for root	Guess 1 failed for root
Guess 2 failed for root	Guess 2 failed for root
Failed to detect root.	Failed to detect root.
Fixed-point iteration failed to converge.	Fixed-point iteration failed to converge.

2) **Newton's Method**
- Newton's Method is an iterative numerical technique used to approximate the roots of a real-valued function. It's particularly effective when you have a good initial guess for the root.
- Newton's Method uses the idea of approximating a function by its tangent line at a given point:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2.1}$$

- It is iterated until convergence.
- We can view newton's method as "optimal" fixed point because $g'(r) = 0$

$$x_{n+1} = g(x_n) = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2.2}$$

$$g'(x) = \frac{f(x) f''(x)}{(f'(x))^2} \tag{2.3}$$

$$f(r) = 0 \implies g'(r) = 0 \tag{2.4}$$

- **Error Analysis:**

- Error $e_k = |x_n - r|$ From Taylor's expansion, Final result

$$e_{n+1} \leq \frac{1}{2} max \left| g''(c) \right| e_n^2 \tag{2.5}$$

- It exhibits quadratic convergence near the root, provided the initial guess is close enough, the function is sufficiently smooth, and the derivative at the root is not zero.
- This means the error roughly squares with each iteration, making Newton's method much faster once it's near the solution.

- **Result:**
  - By taking the initial guesses -3 and 0 (assuming it has two roots)for the first quadratic (0.5) we get the result in 24 iterations.

$$x = -1.5811389 \tag{2.6}$$

  - By taking the initial guesses 3 and 0 (assuming it has two roots)for the second quadratic (0.6) we get the result in 24 iterations.

$$x = 1.5811389 \tag{2.7}$$

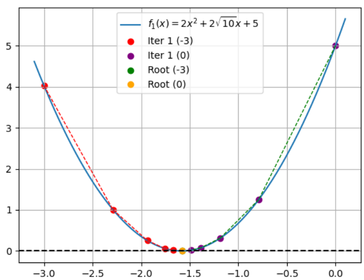

Fig. 2.1: Root of the function1

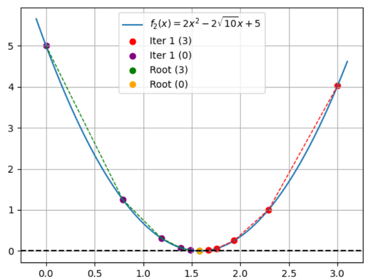

Fig. 2.2: Root of the function2

3) **Secant Method**

- It's a Hybrid idea(variant) of newton's and more robust.
- We need to obtain a good initial Guess x_0 And apply newton's to it.
- In this method we can avoid computing $f'(x)$
- In newton's we use an tangent line, here we use secant for approximation
- secant's iteration:

$$x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \tag{3.1}$$

- **Advantages:**
  - No computation of $f'(x)$
  - One function per iteration is calculated
  - Very rapid convergence

- **Error**

$$e_{k+1} \leq M.e_k^{\alpha} \text{ Where } \alpha = \frac{1}{2}\left(1 + \sqrt{5}\right) \tag{3.2}$$

It is super linear convergence since $1 < \alpha < 2$

- **Failure**
  - For the both the quadratic equations the secant's algo fails to find the root. Output: Difference of the functions is too small.
  - This shows us that both the Quadratics have roots as their minimum values.
  - The secant method fails at minima or maxima of that function
    * The slope between two points is almost zero.
    * The secant line becomes nearly horizontal, and the intersection with the x-axis is poorly defined.

4) **Matrix Method**

If we consider the polynomial equation as the characteristic equation of a matrix, then by finding the eigen-values of that matrix, we can find the roots of the equation. The matrix whose eigen-values are the roots of polynomial equation is called the companion matrix of the said equation. If the given polynomial is,

$$P(x) = c_0 + c_1\, x + c_2\, x^2 + \cdots + c_{n-1}\, x^{n-1} + c_n x^n \tag{4.1}$$

The companion matrix of this polynomial can be written as

$$c = \begin{pmatrix} 0 & 0 & \dots & 0 & -\frac{c_0}{c_n} \\ 1 & 0 & \dots & 0 & -\frac{c_1}{c_n} \\ 0 & 1 & \dots & 0 & -\frac{c_2}{c_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -\frac{c_{n-1}}{c_n} \end{pmatrix} \tag{4.2}$$

Power Iteration

- The Power iteration finds the largest eigen value λ_{max} of a companion matrix
- now divide the polynomial with $x - \lambda_{max}$ by performing synthetic division. This results in a new polynomial of degree one less.
- Repeat this process the degree becomes one.

The power Iteration follows the following steps iteratively:

$$\tilde{\mathbf{v_n}} = C\mathbf{v_{n-1}} \tag{4.3}$$

$$\mathbf{v_n} = \frac{\tilde{\mathbf{v_n}}}{||\tilde{\mathbf{v_n}}||} \tag{4.4}$$

This iteration stops when

$$\left|\lambda_{max}^{(n)} - \lambda_{max}^{(n-1)}\right| < \epsilon \tag{4.5}$$

After ϵ is tolerance, and

$$\lambda_n = \frac{\mathbf{x_n}^\top C \mathbf{x_n}}{\mathbf{x_n}^\top \mathbf{x_n}} \tag{4.6}$$

Once $\lambda_{\max}$ is found, synthetic division is performed to reduce the polynomial:

$$P(x) = P(x) \div (x - \lambda_{\max}), \tag{4.7}$$

$$P(x) = c_{1,0} + c_{1,1}x + c_{1,2}x^2 + \cdots + c_{1,(n-1)}x^{n-1}. \tag{4.8}$$

Now, the new companion matrix will be evaluated.
This process is repeated iteratively until the polynomial is reduced to a degree-1 equation:

$$P_1(x) = c_{n-1,0} + c_{n-1,1}x. \tag{4.9}$$

Thus, the effective update equation would be:

$$P_{n-1}(x) = P_n(x) \div (x - \lambda_n), \tag{4.10}$$

5) **Time complexity**
   - The construction of the companion matrix: $O(n^2)$.
   - The power iteration process: $O(k \cdot n^2)$, where k is the number of iterations required for convergence.
   - Synthetic division: $O(n)$.