PEC 1

Pablo Riutort Grande

April 16, 2023

M0.534 - Chaotic Dynamical Systems

Master's Degree in Computational and Mathematical Engineering

1

Consider the polynomial $p(x) = x^5 + x^2 - x + 1$. Newton's method applied to the polynomial p is given by

$$\begin{aligned} &x_0 && \text{initial seed} \\ &x_{n+1} = x_n - \frac{p(x_n)}{p'(x_n)} \end{aligned}$$

(a) Write the expression of the Newton's method applied to the polynomial p.

$$\begin{aligned} p(x) &= x^5 + x^2 - x + 1 \\ p'(x) &= 5x^4 + 2x - 1 \end{aligned}$$

$$\begin{aligned} &x_0 && \text{initial seed} \\ &x_{n+1} = x_n - \frac{x_n^5 + x_n^2 - x_n + 1}{5x_n^4 + 2x_n - 1} \end{aligned}$$

where:

- x_n is the nth approximation of the root of $p(x)$.
- x_{n+1} is the $(n+1)$th approximation of the root of $p(x)$.
- $p(x_n)$ is the value of $p(x)$ evaluated at x_n.
- $p'(x_n)$ is the first derivative of $p(x)$ evaluated at x_n.

(b) Using different initial seeds x_0 find an approximation of the five roots of p.
You must use real and complex initial seeds, since the polynomial p has only one real root.

- $x_0 \in \mathbb{R}$

$$x_0 = -2$$
$$x_1 = -2 - \frac{-2^5 - 2^2 + 2 + 1}{5(-2)^4 + 2(-2) - 1}$$

$$x_1 = -2 - \frac{-25}{75}$$
$$x_1 = -2 - 0.333$$
$$x_1 = -1.666$$

$$x_2 = -1.666 - \frac{-7.4156}{34.2469}$$
$$x_2 = -1.666 + 0.2165$$
$$x_2 = -1.4501$$

$$x_3 = -1.4501 - \frac{-1.8596}{18.2103}$$
$$x_3 = -1.4501 + 0.1021$$
$$x_3 = -1.3480$$

$$x_4 = -1.3480 - \frac{-0.2859}{12.8139}$$
$$x_4 = -1.3480 + 0.0223$$
$$x_4 = -1.3256$$

$$x_5 = -1.3256 - \frac{-0.011}{11.7920}$$
$$x_5 = -1.3256 + 0.0009$$
$$x_5 = -1.3247$$

There is little difference between x_4 and x_5 (0.001) and successive approximations do not change significantly.

- $x_0 \in \mathbb{C}$
  - $x_0 = 1 - 1i$

$$x_1 = 1.0000 - 1.0000i - \frac{-4.0000 + 3.0000i}{-19.0000 - 2.0000i}$$
$$x_1 = 1.0000 - 1.0000i - 0.1918 - 0.1781i$$
$$x_1 = 0.8082 - 0.8219i$$

$$x_2 = 0.8082 - 0.8219i - \frac{-1.2080 + 0.9917i}{-8.2067 - 1.3472i}$$
$$x_2 = 0.8082 - 0.8219i - 0.1240 - 0.1412i$$
$$x_2 = 0.6842 - 0.6807i$$

$$x_3 = 0.6842 - 0.6807i - \frac{-0.2791 + 0.3338i}{-3.9700 - 1.4057i}$$
$$x_3 = 0.6842 - 0.6807i - 0.0360 - 0.0968i$$
$$x_3 = 0.6482 - 0.5839i$$

$$x_4 = 0.6482 - 0.5839i - \frac{-0.0063 + 0.0801i}{-2.5369 - 1.7674i}$$
$$x_4 = 0.6482 - 0.5839i - -0.0131 - 0.0224i$$
$$x_4 = 0.6613 - 0.5615i$$

$$x_5 = 0.6613 - 0.5615i - \frac{0.0041 + 0.0002i}{-2.3602 - 2.0297i}$$
$$x_5 = 0.6613 - 0.5615i - -0.0010 + 0.0008i$$
$$x_5 = 0.6624 - 0.5623i$$

  - $x_0 = -1 + 1i$

$$x_0 = -1 + 1i$$
$$x_1 = -1.0000 + 1.0000i - \frac{6.0000 - 7.0000i}{-23.0000 + 2.0000i}$$
$$x_1 = -1.0000 + 1.0000i - -0.2852 + 0.2795i$$
$$x_1 = -0.7148 + 0.7205i$$

$$x_2 = -0.7148 + 0.7205i - \frac{2.4530 - 2.5266i}{-7.7337 + 1.5241i}$$
$$x_2 = -0.7148 + 0.7205i - -0.3673 + 0.2543i$$
$$x_2 = -0.3475 + 0.4661i$$

$$x_3 = -0.3475 + 0.4661i - \frac{1.2551 - 0.8564i}{-2.1733 + 1.2450i}$$
$$x_3 = -0.3475 + 0.4661i - -0.6048 + 0.0476i$$
$$x_3 = 0.2573 + 0.4185i$$

$$x_4 = 0.2573 + 0.4185i - \frac{0.6445 - 0.2297i}{-0.6580 + 0.6024i}$$
$$x_4 = 0.2573 + 0.4185i - -0.7068 - 0.2980i$$
$$x_4 = 0.9640 + 0.7165i$$

$$x_5 = 0.9640 + 0.7165i - \frac{-2.0446 + 0.5296i}{-7.7490 + 7.1801i}$$
$$x_5 = 0.9640 + 0.7165i - 0.1760 + 0.0948i$$
$$x_5 = 0.7880 + 0.6217i$$

* $x_0 = -1 - 1i$

$$x_1 = -1.0000 - 1.0000i - \frac{6.0000 + 7.0000i}{-23.0000 - 2.0000i}$$
$$x_1 = -1.0000 - 1.0000i - -0.2852 - 0.2795i$$
$$x_1 = -0.7148 - 0.7205i$$

$$x_2 = -0.7148 - 0.7205i - \frac{2.4530 + 2.5266i}{-7.7337 - 1.5241i}$$
$$x_2 = -0.7148 - 0.7205i - -0.3673 - 0.2543i$$
$$x_2 = -0.3475 - 0.4661i$$

$$x_3 = -0.3475 - 0.4661i - \frac{1.2551 + 0.8564i}{-2.1733 - 1.2450i}$$
$$x_3 = -0.3475 - 0.4661i - -0.6048 - 0.0476i$$
$$x_3 = 0.2573 - 0.4185i$$

$$x_4 = 0.2573 - 0.4185i - \frac{0.6445 + 0.2297i}{-0.6580 - 0.6024i}$$
$$x_4 = 0.2573 - 0.4185i - -0.7068 + 0.2980i$$
$$x_4 = 0.9640 - 0.7165i$$

$$x_5 = 0.9640 - 0.7165i - \frac{-2.0446 - 0.5296i}{-7.7490 - 7.1801i}$$
$$x_5 = 0.9640 - 0.7165i - 0.1760 - 0.0948i$$
$$x_5 = 0.7880 - 0.6217i$$

(c) Divide the complex plane in a fine grid and using the approximation of the five roots and five different colors plot each point in the grid depending on the root which it converges.

For this section we will make use of Wolfram Alpha software. We can input $p(x)$ and the complex seed $1 + 1i$ to follow the algorithm and generate a colored plot.

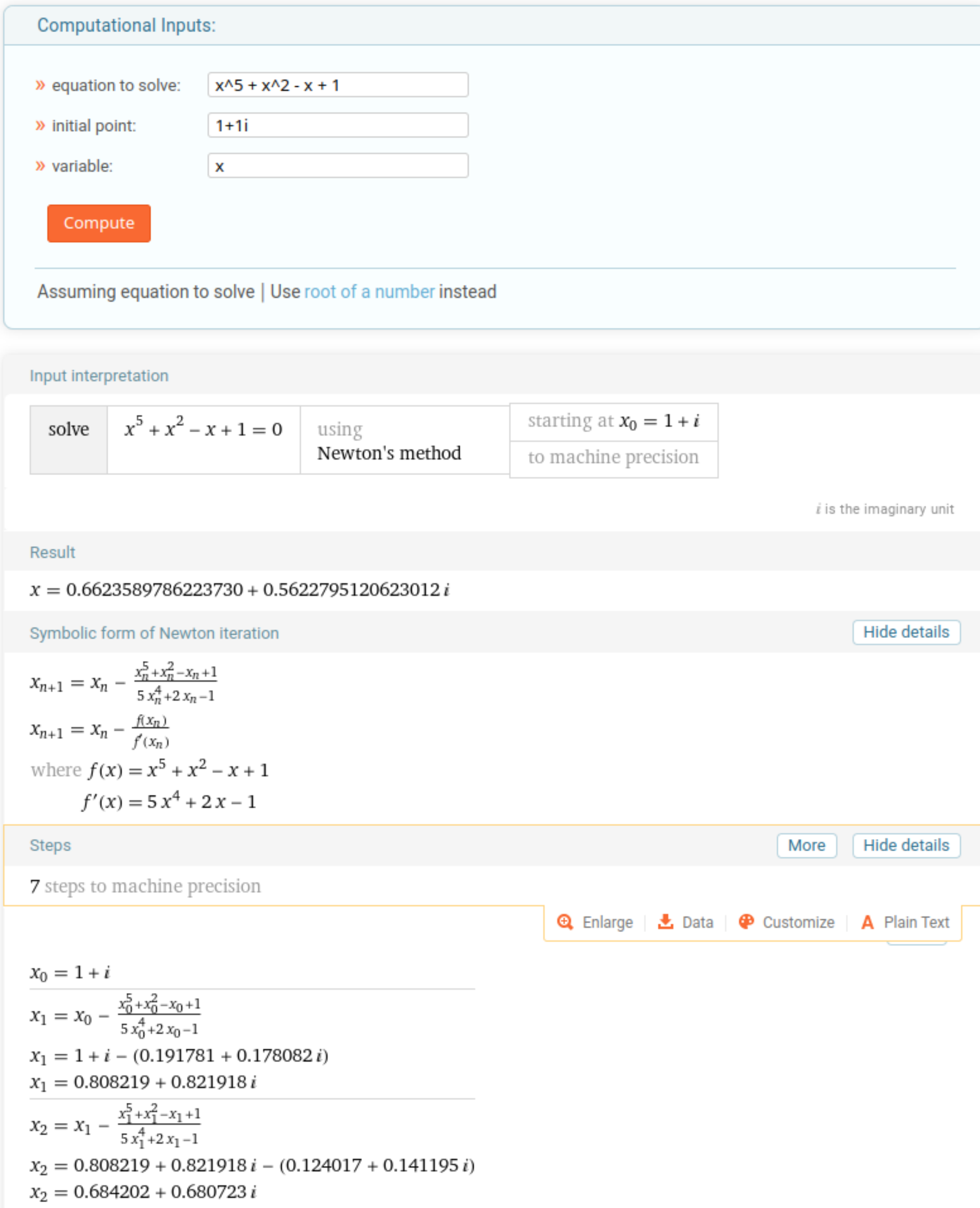

Figure 1: Newton Raphson solver for $p(x)$ with $x_0 = 1 + 1i$

As shown in Figure 2 we can see convergence to 5 different roots:

1. $-1.32472 + 0i$
2. $-i$
3. i
4. $0.662359 - 0.56228i$
5. $0.662359 + 0.56225i$

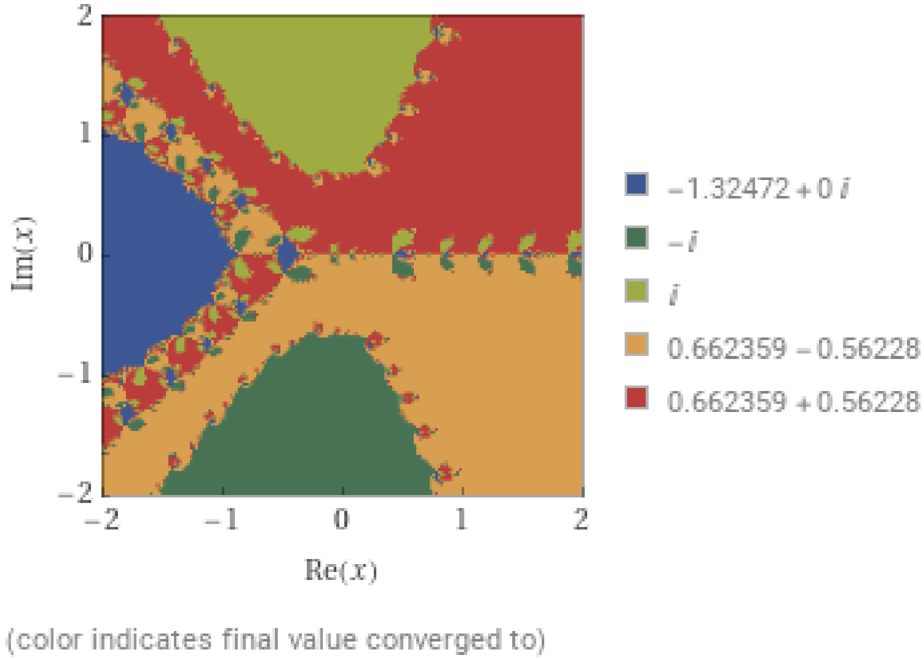

Figure 2: Plot showing convergence to 5 different roots

A Script

Helper script for Newton's method for finding the roots of a polynomial function.

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*
"""
Newton's Method for finding the roots of a polynomial function.
"""
from textwrap import dedent


def polynomial(x):
    return x**5 + x**2 - x + 1


def derivative_p(x):
    return 5 * x**4 + 2 * x - 1


def newton(x_n, step):
    """
    Computes the next approximation of the root of the polynomial function using Newton's Method.

    Args:
        x_n (float): The initial guess for the root of the polynomial function.
        step (int): The step number of the iteration. Defaults to 0.

    Returns:
        float: The next approximation of the root of the polynomial function.
    """
    p = polynomial(x_n)
    p_prime = derivative_p(x_n)
    f = p / p_prime
    x_n1 = x_n - f
    print(
        dedent(
            f"""
            x_{step} = {x_n} - ({p}/{p_prime})
            x_{step} = {x_n} - ({f})
            x_{step} = {x_n1}
            """
        )
    )
    return x_n1


steps = 6
seeds = [-2, 1 - 1j, -1 + 1j, -1 - 1j]
for seed in seeds:
    print(f"x_0: {seed}")
    for i in range(1, steps + 1):
        seed = newton(seed, i)
```