Numerical Analysis

Project. Polynomial Roots Finder.

102061125 Kuan-Chun Chen

1. Objective

Given a polynomial of degree n as the following

$$P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x^1 + a_0$$

with all coefficients real, $a_i \in R, i = 0, \ldots, n$.

In this project, I want to find a method that can find all roots given a polynomial. There are three different methods will be discussed in this project.

1. Newton-Horner Method
2. Lin's Quadratic Method
3. Bairstow's Method

However, each of these three methods has their constraints, so I will go through some polynomials and see the results to compare them.

2. Approach

2.1. Newton-Horner Method

$$P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdot\cdot\cdot + a_1 x + a_0$$

$$P_{n-1}(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdot\cdot\cdot + b_1 x + b_0$$

$$\rightarrow P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdot\cdot\cdot + a_1 x + a_0$$

$$= (\mathrm{x} - z_1)(b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdot\cdot\cdot + b_1 x + b_0)$$

And

$$a_n = b_{n-1}, \qquad\qquad b_{n-1} = a_n,$$

$$a_{n-1} = b_{n-2} - b_{n-1}z_1, \qquad\qquad b_{n-2} = a_{n-1} + b_{n-1}z_1,$$

$$a_{n-2} = b_{n-3} - b_{n-2}z_1, \qquad\qquad b_{n-3} = a_{n-2} + b_{n-2}z_1,$$

$$\dots \qquad\qquad \dots$$

$$a_1 = b_0 - b_1z_1, \qquad\qquad b_1 = a_2 - b_2z_1,$$

$$a_0 = -b_0z_1. \qquad\qquad b_0 = a_1 - b_1z_1.$$

This process is called deflation, and the coefficients can be rewrite as recursive form

$$b_{n-1} = a_n,$$

$$b_j = a_{j+1} + z_1b_{j+1}, \qquad \mathrm{j = n - 2, \dots, 0}$$

Now if we define

$$\begin{aligned} b_{-1} &= a_0 + z_1b_0 \\ &= a_0 + z_1(a_1 + z_1b_1) \\ &= a_0 + z_1(a_1 + z_1(a_2 + z_1b_2)) \\ &= a_0 + z_1(a_1 + z_1(a_2 + z_1(a_3 + z_1(\dots z_1(a_{n-1} + z_1a_n))) \\ &= P_n(z_1) \end{aligned}$$

And when z_1 is a root of $P_n(x)$ then $b_{-1} = 0$.

Next, we need the derivative, $P'_n(x)$, when using Newton's method to find a root.

$$P_n(x) = (x - z_1)P_{n-1}(x)$$

Then

$$\frac{dP_n(x)}{dx} = P_{n-1}(x) + (\mathrm{x} - z_1)\frac{dP_{n-1}(x)}{dx}$$

$$\begin{aligned} P'_n(z_1) &= P_{n-1}(z_1) \\ &= b_{n-1}z_1{}^{n-1} + b_{n-2}z_1{}^{n-2} + \dots + b_1z_1 + b_0 \end{aligned}$$

Now we do deflation again, and we can get $c_{n-2}, c_{n-3}, \dots, c_0, c_{-1}$,
where c_{-1} is the value of the derivative.
And the solution can be solved in iterative form

$$z_1{}^{(k+1)} = z_1{}^{(k)} - b_{-1}/c_{-1}$$

Algorithm. Newton-Horner Method

Given an n-degree polynomial with coefficients $a_0, a_1, \dots, a_n$, and initial guess $x^{(0)}$, a small number ε and an integer *maxiter*,

while $(n \geq 1)$ do
  $err = 1 + \varepsilon;\ \ k = 0;$
  while $\big((n \geq \varepsilon)$ and $(k < maxiter)\big)$ do
    $b_{n-1} = a_n;\ \ c_{n-2} = b_{n-1};$
    for$(j = n-2; j \geq -1; j--)\ \ b_j = a_{j+1} + x^{(k)} b_{j+1};$
    for$(j = n-3; j \geq -1; j--)\ \ c_j = b_{j+1} + x^{(k)} c_{j+1};$
    $f = b_{-1};\ \ f' = c_{-1};$
    $x^{(k+1)} = x^{(k)} - \frac{f}{f'};$
    $err = |f|;\ \ k = k + 1;$
  end while
  $z_n = x^{(k)};$
  for$(j = 0; j < n; j++)\ \ a_j = b_j;$
  $x^{(0)} = z_n; n = n - 1;$
end while

2.2. Lin's Quadratic Method

Given the polynomial as before

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

We assume

$$P_n(x) = (x^2 + px + q)(b_{n-2} x^{n-2} + b_{n-3} x^{n-3} + \cdots + b_1 x + b_0) + Rx + S$$

In case that $x^2 + px + q$ is a factor of $P(x)$, then

$$R = 0, \qquad S = 0$$

Comparing the coefficients, we have

$$a_n = b_{n-2} \qquad\qquad b_{n-2} = a_n$$

$$a_{n-1} = b_{n-3} + p b_{n-2} \qquad\qquad b_{n-3} = a_{n-1} - p b_{n-2}$$

$$a_{n-2} = b_{n-4} + p b_{n-3} + q b_{n-2} \qquad\qquad b_{n-4} = a_{n-2} - p b_{n-3} - q b_{n-2}$$

$$\dots \qquad\qquad \dots$$

$$a_2 = b_0 + p b_1 + q b_2 \qquad\qquad b_0 = a_2 - p b_1 - q b_2$$

$$a_1 = p b_0 + q b_1 + R \qquad\qquad R = a_1 - p b_0 - q b_1$$

$$a_0 = q b_0 + S \qquad\qquad S = a_0 - q b_0$$

Or in recursive form b_i can be found by

$$b_n = 0,$$

$$b_{n-1} = 0,$$

$$b_j = a_{j+2} - pb_{j+1} - qb_{j+2}, \quad j = n-2, \dots, 0.$$

If $x^2 + px + q$ is a factor of $P_n(x)$

$$\mathrm{R} = a_1 - pb_0 - qb_1 = 0$$

$$\mathrm{S} = a_0 - qb_0 = 0$$

$$\rightarrow q^{(k+1)} = \frac{a_0}{b_0}, \qquad p^{(k+1)} = \frac{a_1 - q^{(k)}b_1}{b_0}.$$

$$\rightarrow q^{(k+1)} = \frac{a_0 - b_0 q^{(k)}}{b_0} + q^{(k)}, \quad p^{(k+1)} = \frac{a_1 - b_0 p^{(k)} q^{(k)} b_1}{b_0} + p^{(k)}.$$

$$\rightarrow q^{(k+1)} = \frac{\mathrm{S}}{b_0} + q^{(k)}, \qquad p^{(k+1)} = \frac{\mathrm{R}}{b_0} + p^{(k)}.$$

Once the quadratic factor, $x^2 + px + q$, is found then the real roots or the complex conjugates can be calculated quickly.
$P_n(x)$ can e deflated again to the $n-2$ polynomial. The same process can be carried out for the next factors.

2.3. Bairstow's Method

Given the polynomial as before

$$P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdot\cdot\cdot + a_1 x + a_0$$

We assume

$$P_n(x) = (x^2 + px + q)(b_{n-2}x^{n-2} + b_{n-3}x^{n-3} + \cdots + b_1 x + b_0) + Rx + S$$

$$b_{n-2} = a_n$$

$$b_{n-3} = a_{n-1} - pb_{n-2}$$

$$b_{n-4} = a_{n-2} - pb_{n-3} - qb_{n-2}$$

$$\dots$$

$$b_0 = a_2 - pb_1 - qb_2$$

$$R = a_1 - pb_0 - qb_1$$

$$S = a_0 - qb_0$$

To apply Newton's method, we need to find $\frac{\partial R}{\partial p}, \frac{\partial R}{\partial q}, \frac{\partial S}{\partial p}, \frac{\partial S}{\partial q}$, to form the iterations.

$$\begin{bmatrix} p^{(k+1)} \\ q^{(k+1)} \end{bmatrix} = \begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix} - \begin{bmatrix} \frac{\partial R}{\partial p} & \frac{\partial R}{\partial q} \\ \frac{\partial S}{\partial p} & \frac{\partial S}{\partial q} \end{bmatrix}^{-1} \begin{bmatrix} R(p^{(k)}, q^{(k)}) \\ S(p^{(k)}, q^{(k)}) \end{bmatrix}$$

Thus we have

$$\frac{\partial b_{n-2}}{\partial p} = 0 \qquad \frac{\partial b_{n-2}}{\partial q} = 0$$

$$\frac{\partial b_{n-3}}{\partial p} = -b_{n-2} - p\frac{\partial b_{n-2}}{\partial p} \qquad \frac{\partial b_{n-3}}{\partial q} = -p\frac{\partial b_{n-2}}{\partial q}$$

$$\frac{\partial b_{n-4}}{\partial p} = -b_{n-3} - p\frac{\partial b_{n-3}}{\partial p} - q\frac{\partial b_{n-2}}{\partial p} \qquad \frac{\partial b_{n-4}}{\partial q} = -p\frac{\partial b_{n-3}}{\partial q} - b_{n-2} - q\frac{\partial b_{n-2}}{\partial q}$$

$$\cdots \qquad \cdots$$

$$\frac{\partial b_0}{\partial p} = -b_1 - p\frac{\partial b_1}{\partial p} - q\frac{\partial b_2}{\partial p} \qquad \frac{\partial b_0}{\partial q} = -p\frac{\partial b_1}{\partial q} - b_2 - q\frac{\partial b_2}{\partial q}$$

$$\frac{\partial \mathrm{R}}{\partial p} = -b_0 - p\frac{\partial b_0}{\partial p} - q\frac{\partial b_1}{\partial p} \qquad \frac{\partial \mathrm{R}}{\partial q} = -p\frac{\partial b_0}{\partial q} - b_1 - q\frac{\partial b_1}{\partial q}$$

$$\frac{\partial \mathrm{S}}{\partial p} = -q\frac{\partial b_0}{\partial p} \qquad \frac{\partial \mathrm{S}}{\partial q} = -b_0 - q\frac{\partial b_0}{\partial q}$$

Let $c_j = \frac{\partial b_j}{\partial p}$ and $d_j = \frac{\partial b_j}{\partial q}$, then we have

$$c_{n-2} = 0 \qquad d_{n-2} = 0$$

$$c_{n-3} = -b_{n-2} - pc_{n-2} \qquad d_{n-3} = -pd_{n-2}$$

$$c_{n-4} = -b_{n-3} - pc_{n-3} - qc_{n-2} \qquad d_{n-4} = -pd_{n-3} - b_{n-2} - qd_{n-2}$$

$$\cdots \qquad \cdots$$

$$c_0 = -b_1 - pc_1 - qc_2 \qquad d_0 = -pd_1 - b_2 - qd_2$$

$$\frac{\partial \mathrm{R}}{\partial p} = -b_0 - pc_0 - qc_1 \qquad \frac{\partial \mathrm{R}}{\partial q} = -pc_0 - b_1 - qd_1$$

$$\frac{\partial \mathrm{S}}{\partial p} = -qc_0 \qquad \frac{\partial \mathrm{S}}{\partial q} = -b_0 - qd_0$$

Thus, to find a quadratic factor $x^2 + px + q$ of an n degree polynomial, we have the following algorithm.

Algorithm. Bairstow's Method

Given $p^{(0)}, q^{(0)}$, and integer maxiter and a small number ε,

Let $err = 1 + \varepsilon, k = 0$

while $(err \geq \varepsilon)$ do

$b_{n-2} = a_n, b_{n-3} = a_{n-1} - p^{(k)} b_{n-2}$

$b_j = a_{j+2} - p^{(k)} b_{j+1} - q^{(k)} b_{j+2},\ j = n-4, \dots, 0$

$R = a_1 - p^{(k)} b_0 - q^{(k)} b_1,\ j = n-4, \dots, 0$

$c_{n-2} = 0,\ c_{n-3} = -b_{n-2} - p^{(k)} c_{n-2}$

$c_j = b_{j+1} - p^{(k)} c_{j+1} - q^{(k)} c_{j+2},\ j = n-4, \dots, 0$

$\frac{\partial R}{\partial p} = -b_0 - p^{(k)} c_0 - q^{(k)} c_1,\ \frac{\partial S}{\partial p} = -q^{(k)} c_0$

$d_{n-2} = 0,\ d_{n-3} = -p^{(k)} d_{n-2}$

$d_j = -p^{(k)} d_{j+1} - b_{j+2} - q^{(k)} d_{j+2},\ j = n-4, \dots, 0$

$\frac{\partial R}{\partial q} = -p^{(k)} d_0 - b_1 - q^{(k)} d_1,\ \frac{\partial S}{\partial q} = -b_0 - q^{(k)} d_0$

$$\begin{bmatrix} p^{(k+1)} \\ q^{(k+1)} \end{bmatrix} = \begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix} - \begin{bmatrix} \frac{\partial R}{\partial p} & \frac{\partial R}{\partial q} \\ \frac{\partial S}{\partial p} & \frac{\partial S}{\partial q} \end{bmatrix}^{-1} \begin{bmatrix} R \\ S \end{bmatrix}$$

$k = k + 1,\ err = max(|R|, |S|)$

end while

3. Results

3.1. Order of Convergence

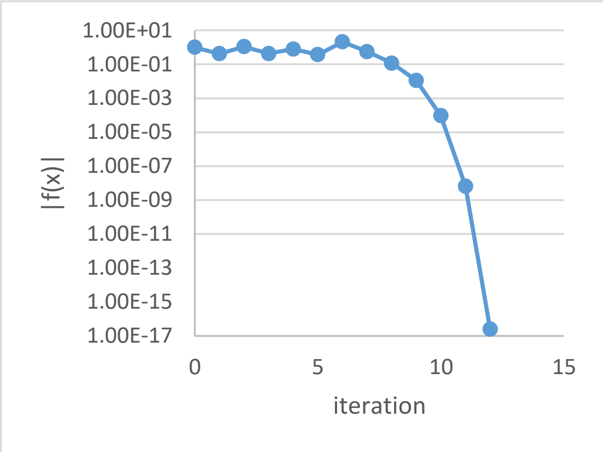
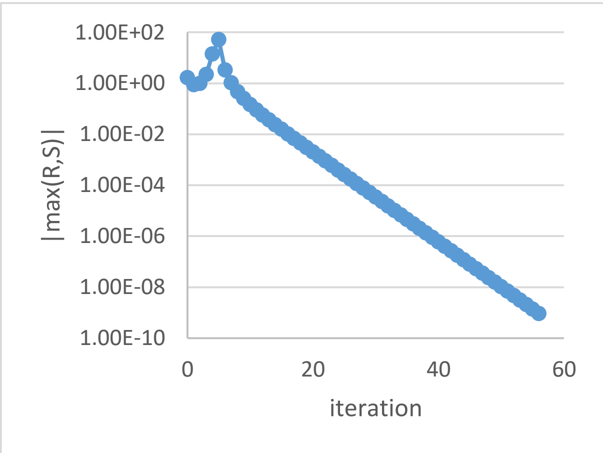
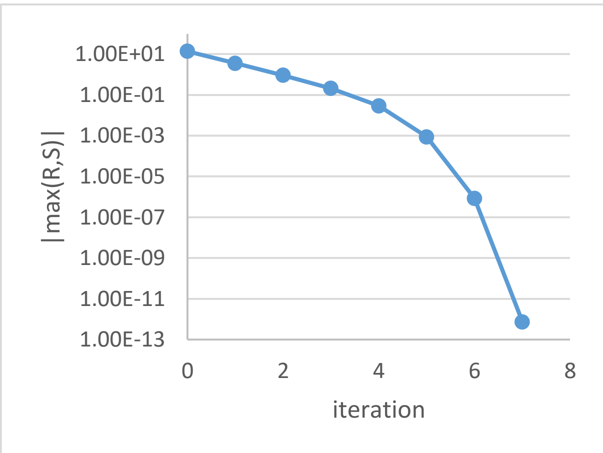
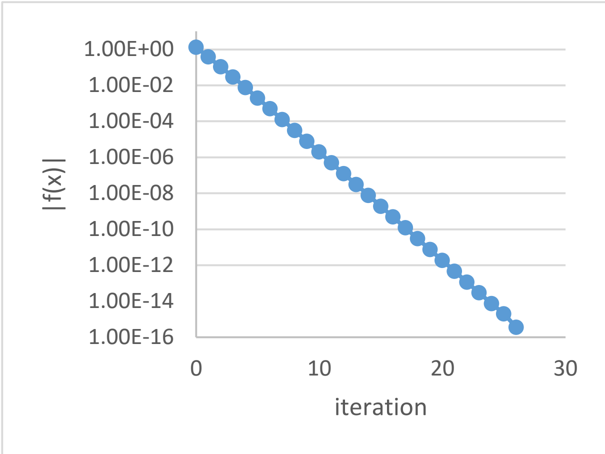
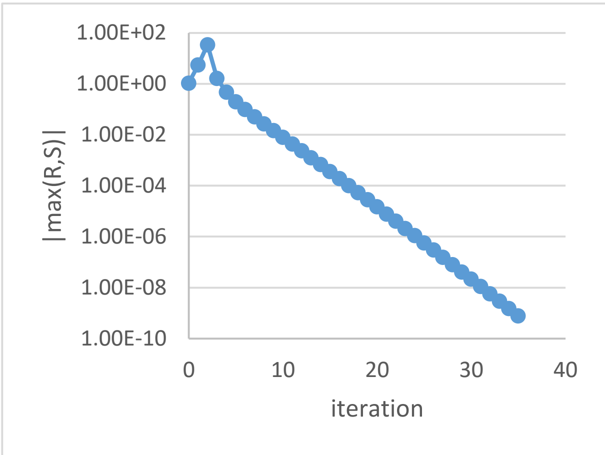
	$f(x) = x^3 - 6x^2 + 11x - 6$ Simple roots: 1, 2, 3.	
	Plot	Order of convergence
Newton-Horner Method initial guess: $x^{(0)} = i$		2
Lin's Quadratic Method initial guess: $p^{(0)} = \frac{a_{n-1}}{a_n},\ q^{(0)} = \frac{a_{n-2}}{a_n}$		1
Bairstow's Method initial guess: $p^{(0)} = 0.5,\ q^{(0)} = 0.5$		2

Table 1.

<table>
<tr><th rowspan="2"></th><th colspan="2">$f(x) = x^3 - 4x^2 + 5x - 2$
Double roots: 1, 1. Simple root: 2.</th></tr>
<tr><th>Plot</th><th>Order of convergence</th></tr>
<tr><td>Newton-Horner Method

initial guess:
$x^{(0)} = i$</td><td>
</td><td>1</td></tr>
<tr><td>Lin's Quadratic Method

initial guess:
$p^{(0)} = \frac{a_{n-1}}{a_n},\ q^{(0)} = \frac{a_{n-2}}{a_n}$</td><td>
</td><td>1</td></tr>
<tr><td>Bairstow's Method

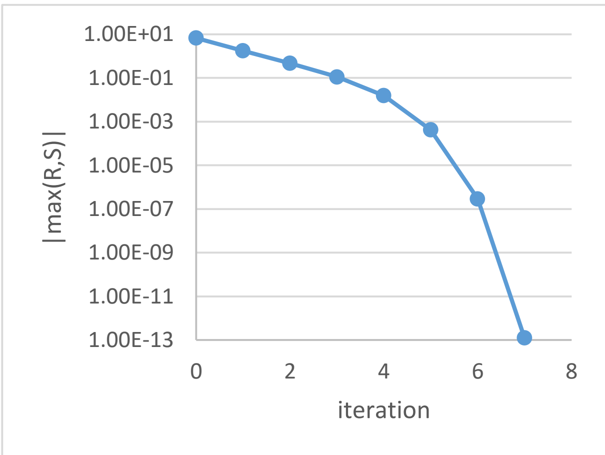
initial guess:
$p^{(0)} = 0.5,\ q^{(0)} = 0.5$</td><td>
</td><td>2</td></tr>
</table>

Table 2.

Order of convergence	$x^3 - 6x^2 + 11x - 6$ Simple roots: 1, 2, 3.	$x^3 - 4x^2 + 5x - 2$ Double roots: 1, 1. Simple root: 2.
Newton-Horner Method	2	1
Lin's Quadratic Method	1	1
Bairstow's Method	2	2

Table 3.

3.2. Roots and Error

	$x^3 - 6x^2 + 11x - 6$ Simple roots: 1, 2, 3.		$x^3 - 4x^2 + 5x - 2$ Double roots: 1, 1. Simple root: 2.	
	roots	error	roots	error
Newton-Horner Method	1-1.45948e-23i	2.22E-16	1+2.48856e-06i	7.44E-12
initial guess:	2+3.04679e-12i	7.08E-12	1-4.97708e-06i	2.98E-11
$x^{(0)} = i$	3-1.52339e-12i	7.07E-12	2-2.78168e-11i	3.72E-11
Lin's Quadratic Method	1	2.22E-16	1	1.82E-11
initial guess:	2	9.41E-10	0.999996	1.82E-11
$p^{(0)} = \frac{a_{n-1}}{a_n}$, $q^{(0)} = \frac{a_{n-2}}{a_n}$	3	1.88E-09	2	6.08E-10
Bairstow's Method	1	0	1	0
initial guess:	2	0	1	0
$p^{(0)} = 0.5$, $q^{(0)} = 0.5$	3	1.47E-12	2	1.49E-13

Table 4.

4. Conclusion

These three methods, Newton-Horner Method, Lin's Quadratic Method and Bairstow's Method, all need initial guess, but guess for different thing.

- Newton-Horner Method needs initial guess for $x^{(0)}$.
- Lin's Quadratic Method needs initial guess for $p^{(0)}, q^{(0)}$.
- Bairstow's Method needs initial guess for $p^{(0)}, q^{(0)}$.

With good initial guess, each method can find all roots of polynomials. However, when we go through the algorithm, there may have some problem.

For example, in Lin's Quadratic Method algorithm

$$\rightarrow q^{(k+1)} = \frac{\mathrm{S}}{b_0} + q^{(k)} \ , \qquad p^{(k+1)} = \frac{\mathrm{R}}{b_0} + p^{(k)} \ ,$$

when b_0 becomes 0, $q^{(k+1)}$ and $p^{(k+1)}$ will go to infinity (or negative infinity).

Anothoer example, in Bairstow's Method algorithm

$$\begin{bmatrix} p^{(k+1)} \\ q^{(k+1)} \end{bmatrix} = \begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix} - \begin{bmatrix} \frac{\partial R}{\partial p} & \frac{\partial R}{\partial q} \\ \frac{\partial S}{\partial p} & \frac{\partial S}{\partial q} \end{bmatrix}^{-1} \begin{bmatrix} R \\ S \end{bmatrix},$$

if the determinant of $\begin{bmatrix} \frac{\partial R}{\partial p} & \frac{\partial R}{\partial q} \\ \frac{\partial S}{\partial p} & \frac{\partial S}{\partial q} \end{bmatrix}$ is 0, $q^{(k+1)}$ and $p^{(k+1)}$ will go to infinity (or negative infinity).

As the result, when we find that the roots do not make sense, maybe we can fine-tune the initial guess to solve this problem.

Let's look at table 2, table 3 and table 4:

- The **accuracy** of these three methods:

  Bairstow's Method > Newton-Horner Method > Lin's Quadratic Method

- The **order of convergence** when polynomial contains only simple roots:

  Bairstow's Method = Newton-Horner Method > Lin's Quadratic Method

- The **order of convergence** when polynomial contains double roots:

  Bairstow's Method > Newton-Horner Method = Lin's Quadratic Method

By the results above, I think Bairstow's Method is the better method to find all roots of polynomials.