# ROOTS OF POLYNOMIALS

The Java program `SolvePolynomial.java` solves polynomial equations of orders 2 through 8 (inclusive). The methodologies used by the program are outlined herein.

## Order 2 (Quadratic)

The equation to solve is

$$x^2 + a_1 x + a_0 = 0. \tag{1}$$

Thus, the Quadratic Formula is used, *viz.*,

$$x = -\frac{1}{2}a_1 \pm \frac{1}{2}\sqrt{a_1^2 - 4a_0}\,, \tag{2}$$

which is implemented in class `O2.java` of the program. Test cases are:

$$\begin{array}{ccc} x^2 - 2x - 15 = 0 & \Rightarrow & x = -3\,,5 \\ x^2 + 16x + 64 = 0 & \Rightarrow & x = -8 \text{ (twice)} \\ x^2 - 2x + 5 = 0 & \Rightarrow & x = 1 \pm 2i\,. \end{array}$$

## Order 3 (Cubic)

The cubic polynomial equation

$$f(x) = x^3 + a_2 x^2 + a_1 x + a_0 = 0 \tag{3}$$

has at least one real root $z$. This real root can be found via bisection, which method is described here.

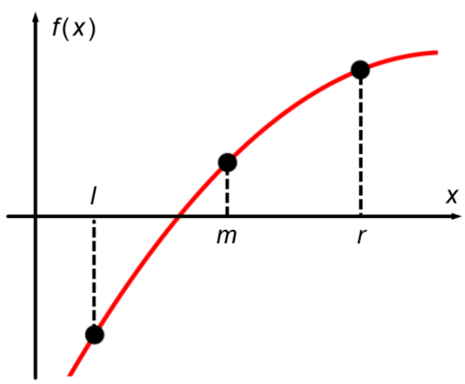


Figure 1. $f(m)$ and $f(r)$ of the same sign.

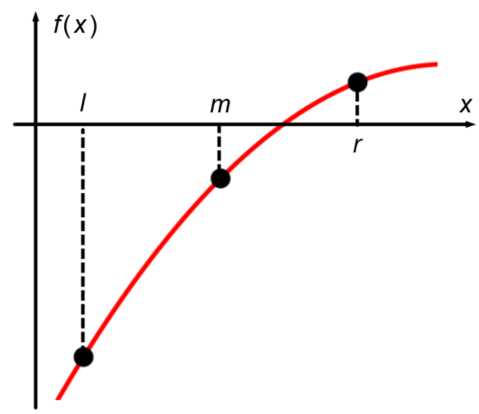


Figure 2. $f(l)$ and $f(m)$ of the same sign.

Start with two $x$–values $x = l$ and $x = r$ for which $f(l)$ and $f(r)$ differ in sign, as depicted in Figs. 1 and 2 above (the program assumes initially that $l =$ `X_MIN = -100` and $r =$ `X_MAX = 100`, see class `Polynomial.java`). The $x$–value $m$ is the average of $l$ and $r$, *i.e.*, $m = 0.5(l + r)$. In Fig. 1, defining $r \leftarrow m$ bounds the solution $f(x) = 0$ more closely; while in Fig. 2, $l \leftarrow m$ bounds the solution more closely. Continuing this procedure, the real root $z$ can be found to a specified tolerance. The program uses $r - l = 1 \times 10^{-7}$ as this tolerance (see the variable `TOL` in class `Polynomial.java`), and once this tolerance is reached, the real root is $z = m$.

Having found $z$, the other two roots of eqn. (3) are found via polynomial division. Namely, solving

$$x^2 + b_1 x + b_0 = 0 \tag{4}$$

via the Quadratic Formula (2) gives the other two roots of eqn. (3), with

$$b_1 = z + a_2 \quad , \quad b_0 = z^2 + a_2 z + a_1\,.$$

This procedure is implemented in class `O3.java` of the program. A test case is

$$x^3 - 6x^2 + 13x - 20 = 0 \qquad \Rightarrow \qquad x = 4\,, 1 \pm 2i\,.$$

## Order 4 (Quartic)

We want to solve

$$f(x) = x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = 0, \tag{5}$$

which can be accomplished by factoring $f(x)$ into two quadratics, *viz.*,

$$f(x) = (x^2 + c_0x + c_1)(x^2 + c_2x + c_3). \tag{6}$$

Expanding eqn. (6) and comparing it to eqn. (5) gives that the coefficients $c_i$ satisfy

$$\begin{aligned} c_0 + c_2 &= a_3 \\ c_1 + c_3 + c_0c_2 &= a_2 \\ c_0c_3 + c_1c_2 &= a_1 \\ c_1c_3 &= a_0\,, \end{aligned} \tag{7}$$

which is a nonlinear system of four equations for the coefficients of the quadratics.

The system (7) can be solved using Newton-Raphson iteration, which procedure is described here. First, define the residual vector $r_i$ , *i.e.*,

$$\begin{aligned} r_0 &= c_0 + c_2 - a_3 \\ r_1 &= c_1 + c_3 + c_0c_2 - a_2 \\ r_2 &= c_0c_3 + c_1c_2 - a_1 \\ r_3 &= c_1c_3 - a_0 \end{aligned} \tag{8}$$

so that the desired solution is given by $r_i = 0$. Now, let $c_j^I$ be the currently best guess for the solution. An improved guess $c_j^{I+1}$ can be obtained by looking at Fig. 3. Defining $\Delta c_i = c_i^{I+1} - c_i^I$, the figure shows that

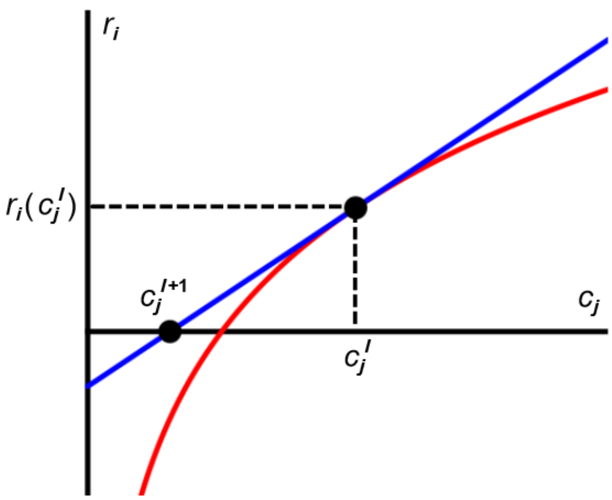


Figure 3. Schematic for Newton-Raphson iteration.

$$\left.\frac{\partial r_i}{\partial c_j}\right|_{\mathbf{c}^I} \equiv s_{ij}(\mathbf{c}^I) = \frac{r_i(\mathbf{c}^I)}{c_j^I - c_j^{I+1}} = \frac{-r_i(\mathbf{c}^I)}{c_j^{I+1} - c_j^I}$$

or

$$s_{ij}(\mathbf{c}^I) = \frac{-r_i(\mathbf{c}^I)}{\Delta c_j}\,. \tag{9}$$

Rearranging eqn. (9), the improved guess for the solution $c_i^{I+1}$ is obtained by solving the four-by-four system

$$\sum_{j=0}^{3} s_{ij}(\mathbf{c}^I)\Delta c_j = -r_i(\mathbf{c}^I) \tag{10}$$

for $\Delta c_j$ , and then by using $c_i^{I+1} = c_i^I + \Delta c_i$ .

Differentiating eqns. (8), the entries of the matrix $s_{ij}$ are

$$s_{ij} \equiv \frac{\partial r_i}{\partial c_j} \qquad \Rightarrow \qquad \mathbf{s} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ c_2 & 1 & c_0 & 1 \\ c_3 & c_2 & c_1 & c_0 \\ 0 & c_3 & 0 & c_1 \end{bmatrix}. \tag{11}$$

In any case, this procedure can be continued until a specified tolerance is achieved. The program uses

$$|\Delta \mathbf{c}| = \sqrt{\sum_{i=0}^{3} (\Delta c_i)^2} < 1 \times 10^{-7}.$$

Once the coefficients $c_i$ have been calculated, the four solutions to eqn. (5) are obtained by solving the two quadratics $x^2 + c_0 x + c_1 = 0$ and $x^2 + c_2 x + c_3 = 0$ with the Quadratic Formula. This methodology is implemented in class `O4` of the program.

Finally, a test case is

$$x^4 - 8x^3 + 42x^2 - 80x + 125 = 0 \qquad \Rightarrow \qquad x = 1 \pm 2i\,, 3 \pm 4i.$$

## Order 5 (Quintic)

The quintic polynomial equation

$$f(x) = x^5 + a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 = 0 \tag{12}$$

has at least one real root $z$, which is found via bisection, as described above for the cubic polynomial equation. Having the a real root, polynomial division gives that the other four solutions to eqn. (12) are found by solving the quartic

$$x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0 = 0\,,$$

where

$$b_3 = z + a_4\,, \quad b_2 = z^2 + a_4 z + a_3\,, \quad b_1 = z^3 + a_4 z^2 + a_3 z + a_2\,, \quad b_0 = z^4 + a_4 z^3 + a_3 z^2 + a_2 z + a_1\,.$$

Finally, a test case is

$$x^5 - x^4 - 14x^3 + 214x^2 - 435x + 875 = 0 \qquad \Rightarrow \qquad x = -7\,, 1 \pm 2i\,, 3 \pm 4i\,.$$

This procedure is implemented by class `O5` of the program.

## Order 6 (Sextic)

To solve the sextic polynomial equation

$$f(x) = x^6 + a_5 x^5 + a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 = 0\,, \tag{13}$$

factor it into three quadratics, *i.e.*,

$$f(x) = (x^2 + c_0 x + c_1)(x^2 + c_2 x + c_3)(x^2 + c_4 x + c_5)\,. \tag{14}$$

Expanding eqn. (14) and comparing the result to eqn. (13), one sees

$$\begin{aligned} c_0 + c_2 + c_4 &= a_5 \\ c_1 + c_3 + c_5 + c_0 c_2 + c_0 c_4 + c_2 c_4 &= a_4 \\ c_0 c_3 + c_0 c_5 + c_1 c_2 + c_1 c_4 + c_2 c_5 + c_3 c_4 + c_0 c_2 c_4 &= a_3 \\ c_1 c_3 + c_1 c_5 + c_3 c_5 + c_0 c_2 c_5 + c_0 c_3 c_4 + c_1 c_2 c_4 &= a_2 \\ c_0 c_3 c_5 + c_1 c_2 c_5 + c_1 c_3 c_4 &= a_1 \\ c_1 c_3 c_5 &= a_0\,, \end{aligned} \tag{15}$$

which are six nonlinear equations in the six unknowns $c_i$ . Equations (15) are then solved via Newton-Raphson iteration, as explained above for the quartic polynomial equation. Having the coefficients $c_i$ , the six solutions to eqn. (13) are found by solving the quadratics $x^2 + c_0 x + c_1 = 0$, $x^2 + c_2 x + c_3 = 0$ and $x^2 + c_4 x + c_5 = 0$. This method is implemented by class `O6` of the `SolvePolynomial` program. A test case is

$$x^6 - 18x^5 + 183x^4 - 988x^3 + 3487x^2 - 6130x + 7625 = 0 \qquad \Rightarrow \qquad x = 1 \pm 2i\,, 3 \pm 4i\,, 5 \pm 6i\,.$$

## Order 7 (Septic)

To solve the septic polynomial equation

$$f(x) = x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = 0\,, \qquad (16)$$

first find a real root $z$ via bisection, and then solve the sextic polynomial equation

$$f(x) = x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = 0$$

for the other six solutions to eqn. (16), where

$$\begin{aligned} b_5 &= z + a_6\,, & b_4 &= z^2 + a_6z + a_5\,, \\ b_3 &= z^3 + a_6z^2 + a_5z + a_4\,, & b_2 &= z^4 + a_6z^3 + a_5z^2 + a_4z + a_3\,, \\ b_1 &= z^5 + a_6z^4 + a_5z^3 + a_4z^2 + a_3z + a_2\,, & b_0 &= z^6 + a_6z^5 + a_5z^4 + a_4z^3 + a_3z^2 + a_2z + a_1\,. \end{aligned}$$

This is implemented in class `O7` of the program. A test case is

$$x^7 - 16x^6 + 147x^5 - 622x^4 + 1511x^3 + 844x^2 - 4635x + 15{,}250 = 0 \quad \Rightarrow$$
$$x = -2\,, 1 \pm 2i\,, 3 \pm 4i\,, 5 \pm 6i\,.$$

## Order 8 (Octic)

To solve the octic polynomial equation

$$f(x) = x^8 + a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = 0\,, \qquad (17)$$

factor it into four quadratics, *i.e.*,

$$f(x) = (x^2 + c_0x + c_1)(x^2 + c_2x + c_3)(x^2 + c_4x + c_5)(x^2 + c_6x + c_7)\,. \qquad (18)$$

Expanding eqn. (18) and comparing the result to eqn. (17), one sees that the coefficients $c_i$ obey

$$\begin{aligned} c_0 + c_2 + c_4 + c_6 &= a_7 \\ c_1 + c_3 + c_5 + c_7 + c_0c_2 + c_0c_4 + c_0c_6 + c_2c_4 + c_2c_6 + c_4c_6 &= a_6 \\ c_0c_3 + c_0c_5 + c_0c_7 + c_1c_2 + c_1c_4 + c_1c_6 + c_2c_5 + c_2c_7 + c_3c_4 + c_3c_6 + c_4c_7 + c_5c_6 + {} & \\ + c_0c_2c_4 + c_0c_2c_6 + c_0c_4c_6 + c_2c_4c_6 &= a_5 \\ c_1c_3 + c_1c_5 + c_1c_7 + c_3c_5 + c_3c_7 + c_5c_7 + c_0c_2c_5 + c_0c_2c_7 + c_0c_3c_4 + c_0c_3c_6 + c_0c_4c_7 + c_0c_5c_6 + {} & \\ + c_1c_2c_4 + c_1c_2c_6 + c_1c_4c_6 + c_2c_4c_7 + c_2c_5c_6 + c_3c_4c_6 + c_0c_2c_4c_6 &= a_4 \\ c_0c_3c_5 + c_0c_3c_7 + c_0c_5c_7 + c_1c_2c_5 + c_1c_2c_7 + c_1c_3c_4 + {} & \\ + c_1c_3c_6 + c_1c_4c_7 + c_1c_5c_6 + c_2c_5c_7 + c_3c_4c_7 + c_3c_5c_6 + c_0c_2c_4c_7 + c_0c_2c_5c_6 + c_0c_3c_4c_6 + c_1c_2c_4c_6 &= a_3 \\ c_1c_3c_5 + c_1c_3c_7 + c_1c_5c_7 + c_3c_5c_7 + {} & \\ + c_0c_2c_5c_7 + c_0c_3c_4c_7 + c_0c_3c_5c_6 + c_1c_2c_4c_7 + c_1c_2c_5c_6 + c_1c_3c_4c_6 &= a_2 \\ c_0c_3c_5c_7 + c_1c_2c_5c_7 + c_1c_3c_4c_7 + c_1c_3c_5c_6 &= a_1 \\ c_1c_3c_5c_7 &= a_0\,, \end{aligned} \qquad (19)$$

which are eight nonlinear equations in the eight unknowns $c_i$ . Equations (19) then are solved via Newton-Raphson iteration as has been described above. Knowing the coefficients $c_i$ then, the eight solutions to eqn. (17) are obtained by solving the four quadratics $x^2 + c_0x + c_1 = 0$ , $x^2 + c_2x + c_3 = 0$ , $x^2 + c_4x + c_5 = 0$ and $x^2 + c_6x + c_7 = 0$. This procedure is implemented in the class `O8` of the program. Finally, a test case is

$$x^8 - 32x^7 + 548x^6 - 5584x^5 + 37{,}998x^4 - 166{,}592x^3 + 487{,}476x^2 - 799{,}440x + 861{,}625 = 0 \quad \Rightarrow$$
$$x = 1 \pm 2i\,, 3 \pm 4i\,, 5 \pm 6i\,, 7 \pm 8i\,.$$

## Ad Infinitum

The above procedures can be continued indefinitely. Namely, any even-order polynomial equation can be factored into quadratics via Newton-Raphson iteration. Additionally, any odd-order polynomial equation can be solved by using bisection to find a real root, and the remaining roots found by solving an even-order polynomial equation.