Muller's Method
A Powerful User Friendly Tool For Finding Roots

By Joseph Pousada
December 15, 2013

There are various techniques available for estimating the root(s) of a problem. One popular and powerful method is the Newton-Raphson Method "$p_{n+1} = p_n - \frac{f(p_n)}{f^{(1)}(p_n)}$" (Faires, J. D., & Burden, R. L.,pg 52, (1998)). This method while powerful requires that we have the derivative of the function and it has the following shortcomings:

"Our approximation is far away from the actual root,
The 2nd derivative is very large, or
The derivative at x_n is close to zero." (Wilhelm Harder, D. (n.d.))

We could pursue the secant method "$p_{n+1} = p_n - \frac{f(p_n) \cdot (p_n - p_{n-1})}{f(p_n) - f(p_{n-1})}$ " (Faires, J. D., & Burden, R. L.,pg 52,pg 44, (1998)), which does not require us to have the first derivative. But, this methods uses successive secant lines to approximate the root (which is not always a disadvantage) and assumes the user is aware ahead of time that the root would be real or complex. An alternate method that does not require the user have the first derivative and does not require the user to be aware of the root is complex ahead of time is Muller's Method. This method works well with real or complex numbers since it uses the quadratic equation in its' method. Muller's Method is defined as:

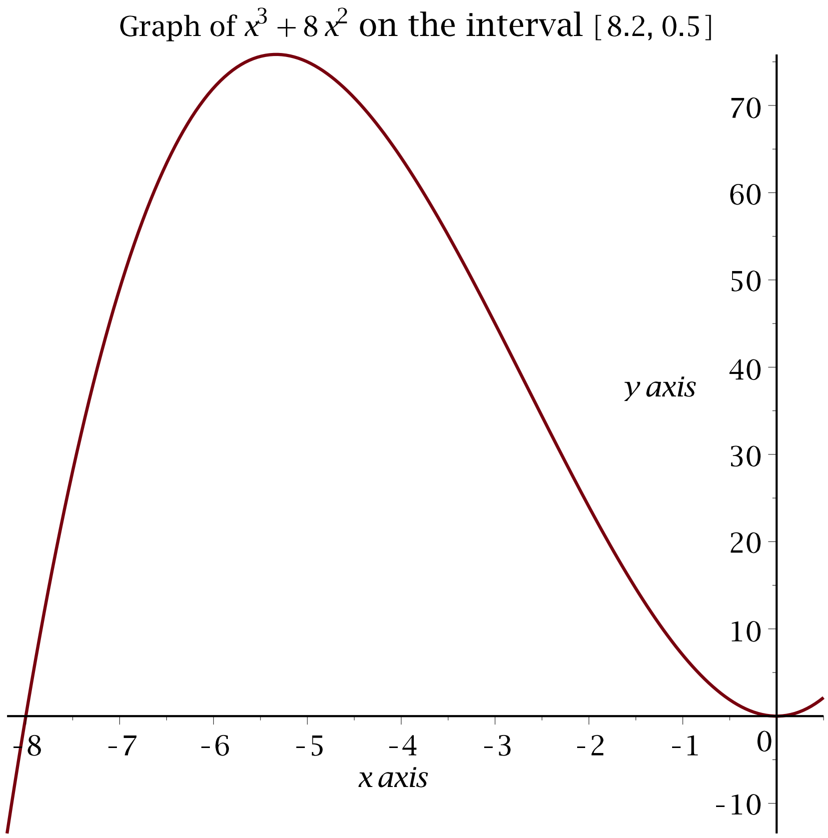
"Given initial approximations p_0, p_1 & p_2 generates $p_3 = p_2 - \frac{2 \cdot c}{b + sgn(b)\sqrt{(b)^2 - 4 \cdot a \cdot c}}$

where, $c = f(p_2)$, $b = \frac{((p_0 - p_2)^2 \cdot [f(p_1) - f(p_2)] - (p_1 - p_2)^2 \cdot [f(p_0) - f(p_2)])}{(p_0 - p_2) \cdot (p_1 - p_2) \cdot (p_0 - p_1)}$ &

$a = \frac{((p_1 - p_2) \cdot [f(p_0) - f(p_2)] - (p_0 - p_2) \cdot [f(p_1) - f(p_2)])}{(p_0 - p_2) \cdot (p_1 - p_2) \cdot (p_0 - p_1)}$ " (Faires, J. D., & Burden, R. L.,pg 66, (1998))

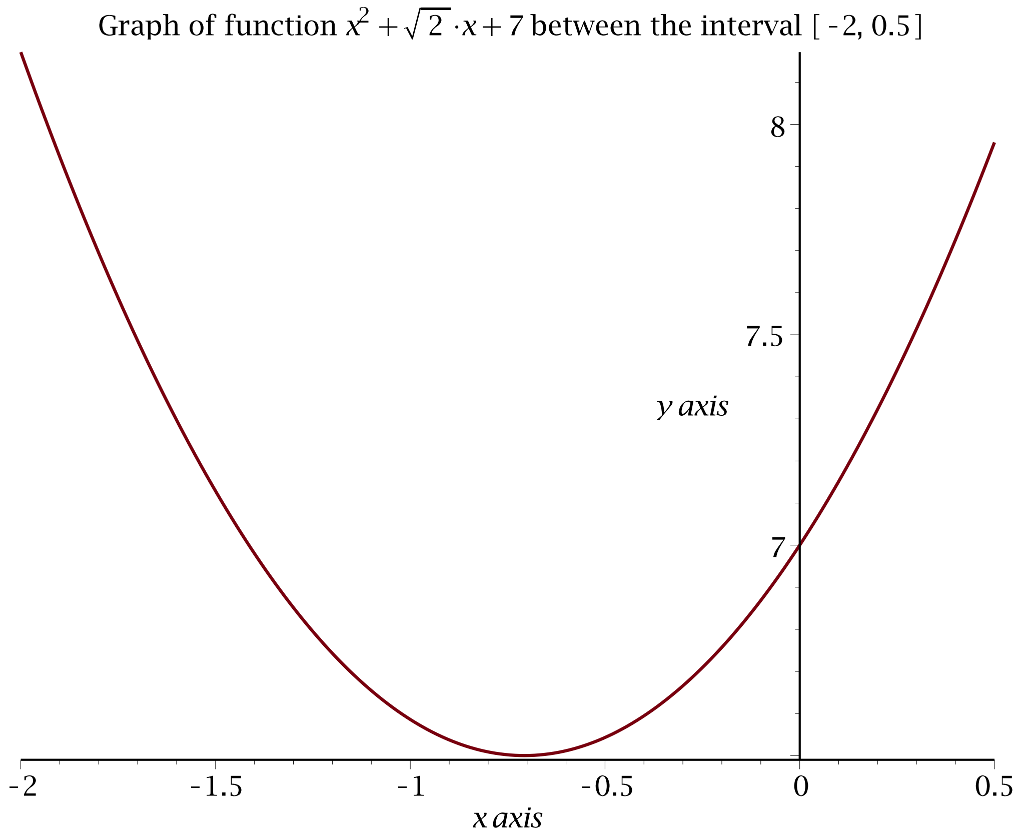
The fundamental approach behind Muller's Method is to take three points relatively near the root for the equation, use interpolation techniques to derive a polynomial and then apply the quadratic equation to the polynomial to derive the estimated root.

If we consider the equation $y = x^3 + 8x^2$ and graph the function (see graph below), we can see that there is a root somewhere around the x value of -8. We will use this to compare the secant method to Muller's method.



If we use the secant method picking points -8.2 and -7.5 which are close to the root we get the estimated value of -7.973564573, with just one iteration. With Muller's method picking the points -8.2 , -7.9, -7.5 we get the estimated value of -8.177515479. It turns out in this example that the actual answer is a real root of -8. The absolute error (for the x value) for the secant method is .026435427 and the absolute error for Muller's method (for the x value) is .177515479. In this example if we look at the interval on the graph of the function between -8.2 and -7.5 we see that the graph looks almost like a straight line. This is why the secant method performed slightly better than Muller's method. But, even with conditions favourable to the secant method, Muller's method performed quite well.

We will consider another equation $y = x^2 + \sqrt{2} \cdot x + 7$ and graph the function (see graph below), we can see that the equation does not cross the x axis. We will use this to compare the secant method to Muller's method one again.



The fact that this equation does not cross the x axis means that the roots will be imaginary. However, this assumes that the user is aware of this. If the user is not aware of this and tries to use the secant method as if trying to find real roots then the first four iterations using the secant method will produce the following estimates (with points -0.9 and -0.6 selected): first iteration=75.30327931,second iteration= -0.685545003, third iteration = -0.7710414961, fourth iteration = 152.7252049. We can see that the solutions do not converge at all.

If the user is aware that the roots are imaginary and uses the secant method to find complex roots then the first four iterations produce the following results (using the same selected points): first iteration = $-2.661176470-2.508981237\,\mathrm{I}$, second iteration = $-2.508362099+0.380440928\,\mathrm{I}$, third iteration = $-2.915476764-0.6187031827\,\mathrm{I}$ and the fourth iteration =$-2.544984497-0.6334153469\,\mathrm{I}$.

If we use Muller's method (with points -0.9, -0.6 & -0.4 selected) we get the estimated root of $-0.7071067810+3.705956360\,\mathrm{I}$. It turns out that the actual root is

$-\frac{1}{2}\sqrt{2}+\frac{1}{2}\,\mathrm{I}\sqrt{26}$ and if we estimate these values to 10 digits we get $-0.7071067810 + 2.549509757\,\mathrm{I}$. Not only did the user not need to know ahead of time that the roots would be complex but, this method produced significantly better results than the secant method did after four iterations.

References

Faires, J. D., & Burden, R. L. (1998). Numerical methods. Pacific Grove, CA: Brooks/Cole Pub. Co.

NumericalMethodsGuy (2003, March 23). Derivation of Secant Method: Approach 1 [Video file]. Retrieved from http://www.youtube.com/watch?v=jXIUi7zxWIU

Wilhelm Harder, D. (n.d.). 10.3 Newton's Method | Department of Electrical and Computer Engineering | University of Waterloo. Retrieved from https://ece.uwaterloo.ca/~dwharder/NumericalAnalysis/10RootFinding/newton/

Wilhelm Harder, D. (n.d.). 10.6 Müller's Method | Department of Electrical and Computer Engineering | University of Waterloo. Retrieved from https://ece.uwaterloo.ca/~dwharder/NumericalAnalysis/10RootFinding/mueller/