Comments on An Improvement to the Brent's Method

Steven A. Stage *steve.stage@iem.com*
IEM
8550 United Plaza Boulevard, Suite 501
Baton Rouge, Louisiana
70808-0200, United States of America

Abstract

Zhang (2011)[1] presented improvements to Brent's method for finding roots of a function of a single variable. Zhang's improvements make the algorithm simpler and much more understandable. He shows one test example and finds for that case that his method converges more rapidly than Brent's method. There are a few easily-correctible flaws in the algorithm as presented by Zhang which must be corrected in order to implement it. This paper shows these corrections.

We then proceed to compare the performance of several well-known root finding methods on a number of test functions. Methods tested are Zhang's method, Bisection, Regula Falsi with the Illinois algorithm, Ridder's method, and Brent's method. The results show that Brent's method and Regula Falsi generally give relatively slow initial convergence followed by very rapid final convergence and that Regula Falsi converges nearly as rapidly as Brent's method. Zhang's method and Ridder's method show similar convergence with both having faster initial convergence than Brent and Regula Falsi but slower final convergence. In many situations, the more rapid initial convergence of the Zhang method and Ridder's method leads to obtaining solutions with fewer total function evaluations than needed for Brent or Regula Falsi. Selection of the best method depends on the function being evaluated, the size of the initial interval, and the amount of accuracy required for the solution. Large initial intervals and low accuracy favor the Zhang and Ridder methods, while smaller intervals and high accuracy requirements favor Brent and Regula Falsi methods.

Guidance is presented to help the reader determine which root-finding method may be most efficient in a particular situation.

Keywords: Brent's Method, Zhang's Method, Ridder's Method, Regula Falsi Method, Bisection Method, Root Finding, Simplification, Improvement

1. INTRODUCTION

A common problem in numerical analysis is to the find the root of a function. In other words, for some given function F(x), to find the value of x such that F(x) = 0. The original purpose of this paper was to examine the behavior of a new root-finding algorithm proposed by Zhang (2011)[1], however, the work expanded to include comparative evaluation of several other root-finding methods.

Through the years, this author has needed a root finding algorithm on several occasions, most often selecting Regula Falsi with the Illinois algorithm as dependable and sufficiently fast and rejecting Brent's method as too complicated to bother with. Zhang's paper promised a simple but powerful method, however, it was necessary to the correct problems mentioned below in Section 2 before Zhang's method could be implemented or tested. By comparing the convergence of Zhang's method with several well-known root finding methods, we provide some guidance that may help users select the method that is best suited for their particular application. Texts discussing root finding methods emphasize the order of convergence, but generally do not discuss the speed of initial convergence, which, as shown below, can be a major factor in how rapidly a satisfactory root can be found.

Brent (1973)[2] presented a method for finding roots of functions of single variables that is both reliable and has better than linear convergence. This method is widely presented in textbooks [3][4], however it involves a complicated set of rules which make the method difficult to understand and requires complex computer code. Zhang (2011)[1] proposed revisions to Brent's method that make it much simpler and therefore easy to follow. Zhang tested this method for finding the root of an example function and showed that The Zhang method gave more rapid convergence than the traditional Brent method for that example.

While implementing and testing the Zhang method, this author found a couple of flaws in the algorithm as presented by Zhang. Fortunately, these flaws can be readily corrected as shown below. The corrected Zhang algorithm was tested for finding roots of several functions and compared with well-known root finding algorithms. The results are shown below and demonstrate that the Zhang compares favorably with the Brent method and other traditional methods in many cases, is superior in some situations, and is inferior in other situations..

2. CORRECTED ZHANG METHOD

This section first identifies the flaws found in the Zhang algorithm as presented by Zhang and then shows a corrected version of the method.

2.1 Flaws in the Algorithm

The following flaws have been identified in the algorithm:

- In the text, Zhang states that his method assumes $a<b$, however, the logic in his algorithm for selecting which two points to retain for use in the next iteration only works for $b<a$.

- Sometimes the algorithm can lead to $fc=0$ or $fs=0$. The logic does not always result in these values being identified as roots.

- The value of s found using inverse quadratic interpolation can sometimes be outside of the interval (a,b) leading to a function evaluation that cannot be the desired root. Furthermore, the logic for selecting the interval for the next iteration does not correctly exclude use of s in some of these cases.

For cases when s is not between b and a, there are three obvious options:

- Do not do the second function evaluation. Select the next interval based on a, c, and b. This can be accomplished by setting $s:=c$.

- Find s by bisecting (a,c) or (c,b), depending on which contains the root and evaluate f(s).

- Find s using the secant method on (a,c) or (c,b), depending on which contains the root and evaluate f(s).

The algorithm shown in the next section allows use of any of these three options. The performance of the three methods is compared below. This situation does not occur in many of the tests shown here, so the three options produce identical results for most tests. In the tests done here, the choice of option does not appear to have significant impact on convergence.

2.2 Corrected Algorithm

The above-mentioned flaws are corrected in the following version of the Zhang algorithm. Comments have been added to clarify logic.

- **input** a, b, and a pointer to a subroutine for f
- **if b<a then swap(a,b) end if** (Logic below requires a<b)
- calculate f(a)
- calculate f(b)
- **if** f(a)*f(b) >= 0 **then** error-exit **end if**
- **repeat until** f(a or b) = 0 or |b – a| is small enough (convergence)
    - c =(a+b)/2
    - calculate f(c)
    - **if** f(a) ≠ f(c) and f(b) ≠ f(c) **then**
        - calculate s (inverse quadratic interpolation)
        - **If** a<s and s<b **then**
            - calculate f(s)
        - **else**
            - s is not in (a,b). Use logic shown in section 2.1.
        - **End if**
    - **else**
        - calculate s (secant rule) (Use the interval (a,c) or (c,b) that contains the root.)
        - calculate f(s)
    - **end if**
    - **if** c> s **then** swap(s,c) (Ensures that a<=c<=s<=b as required for the logic below to work.)
    - **if** f(c)*f(s) <= 0 **then** (The equal sign here ensures that points with f(c or s)=0 will be used.)
    - a: = c
    - b: = s
    - **else**
    - **if** f(a)*f(c) < 0 **then** b: = c **else** a: = s **end if** (Corrected from Zhang.)
    - **end** if
- **end repeat**
- **output** a or b (Return root.)

3. PERFORMANCE TESTS

This section compares the performance the Zhang method with several other classic root finding methods.

3.1 Methods Compared

The following root-finding methods are included in these tests:

- Bisection
- Regula Falsi (including the Illinois algorithm)
- Ridder
- Brent
- Zhang
- Zhang mid
- Zhang sec

With the exception of the Zhang method, these algorithms are well known and can be found in standard texts (for instance, Press, et. al. (1995) [4]) and at numerous locations on the internet.

The performance of the basic Regula Falsi method is reduced by the tendency for one end of the evaluation interval to remain stationary while the other improves. In the Regula Falsi method used here, this tendency is reduced by using the Illinois algorithm [5] described as follows. Use of this algorithm greatly improves final convergence of Regula Falsi.

Before iterations begin:

- **Set** iEnd=-1

Within the iteration loop, when selecting the interval to use for the next point:

- **If** fa * fc <= 0 **then**
    - b:=c
    - **If** iEnd =0 **then**
        - fa=fa/2
    - **end if**
    - iEnd =0
- **else**
    - a:=c
    - **If** iEnd =1 **then**
        - fb=fb/2
    - **end if**
    - iEnd =1
- **end if**

The three Zhang methods correspond to the three options described in section 2.1 for handling cases when s is not in the interval (a, b). For the majority of the tests run here, this situation does not arise and the three options give the identical same results. Except as noted, all three Zhang options produce a single curve on the plots shown below.

3.2 Initial Interval

Except as noted, the tests shown in this section were run using -10 and 10 as the initial interval. This interval was used because it is large enough to capture some of the behavior of the algorithms when they are not in the final stages of convergence. Tests were also done using other intervals to determine the sensitivity of the results to initial interval and the results were considered in reaching the conclusions stated in this paper.

All functions tested have roots between 0 and 1 with locations of roots chosen to not be the exact result of bisection of the initial intervals used.

3.3 Convergence Conditions

For these tests, the convergence condition was that any one of the following conditions be true:

- $|a-b| \le 10^{-15}$
- $|\mathrm{f}(a)| \le 10^{-15}$
- $|\mathrm{f}(b)| \le 10^{-15}$
- $|a-b| \le \frac{1}{2}|a+b| 10^{-15}$

The last of these conditions was never encountered in these tests, because the roots all had absolute values less than 1. However, in general, having a condition like this is important to account for machine accuracy for cases with roots having large absolute values.

3.4 Testing Function From Zhang

Function tested

$$f(x) = \cos(x) - x^3$$

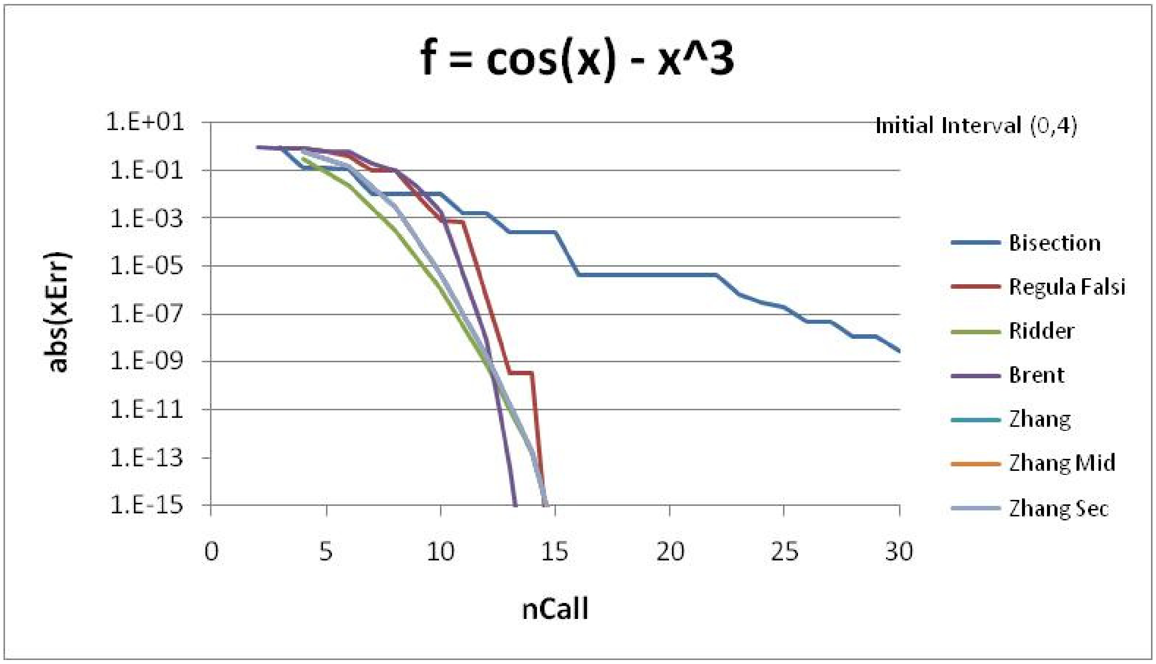

FIGURE 1: Convergence for Zhang's Test Function. Initial Interval (0,4).

This is the function tested by Zhang and uses the initial interval (0,4), which corresponds to Zhang. The code used in this paper exactly reproduces the values shown in Zhang Table 1 verifying the code implementation.

The plot shows the absolute value of error in *x* as a function of the number of function calls, where the error in *x* is the difference between the current estimate of *x* and the value at convergence.

In Figure 1, the Zhang and Ridder methods give the fastest initial convergence, but Brent and Regula Falsi have faster final convergence and catch up. Bisection is slow. All three options for the Zhang method produce identical results for this test.

Figure 2, shows convergence when the initial interval is (-10,10). In this case, Brent and Regula Falsi give the fastest convergence. Ridder and Zhang (all three options) are next and Bisection is slowest. Tests using other intervals show order of the convergence curves is quite dependent on initial interval. However all tests indicate that final convergence is fastest (the curves drop mostly steeply) for Brent and Regula Falsi and that final convergence of Ridder and Zhang is slower and about equal to each other.

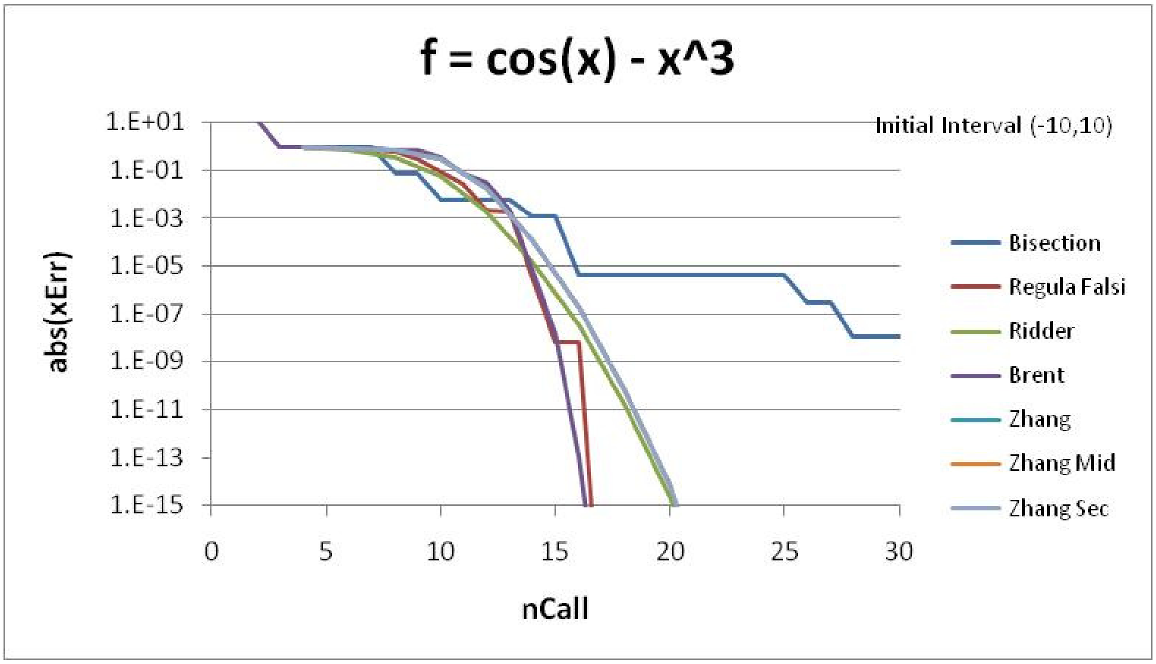

FIGURE 2: Convergence for Zhang's Test Function. Initial Interval (-10,10).

3.5 Testing Another Cosine Function

Function tested

$$\mathrm{f}(x) = \cos(x) - x$$

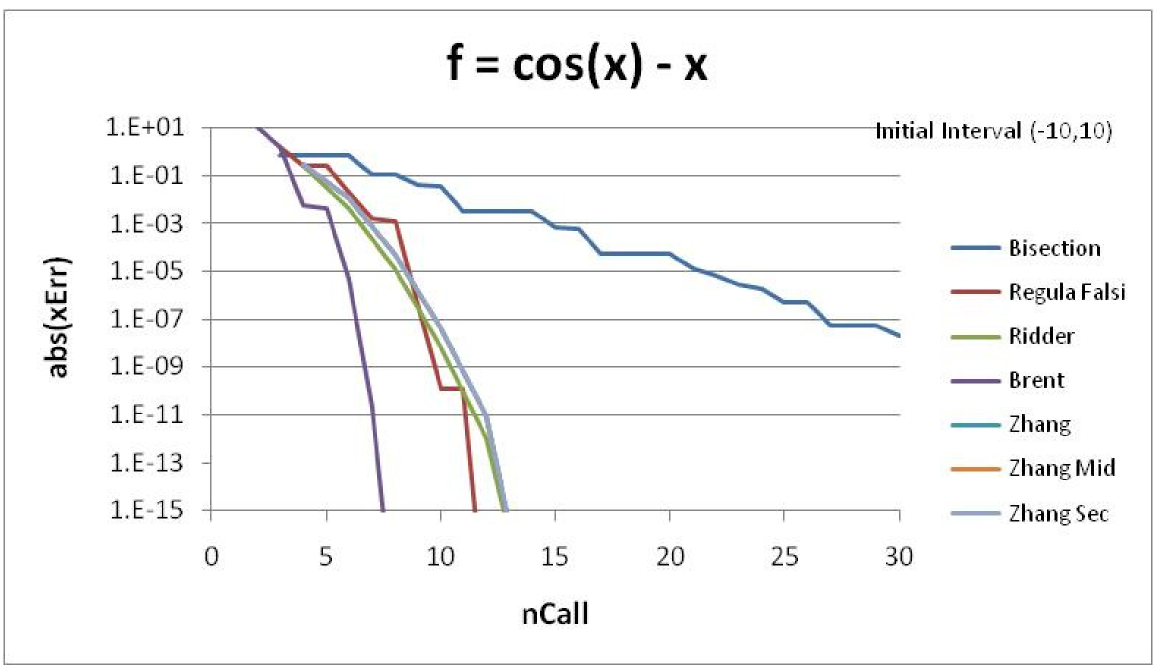

FIGURE 3: Convergence for a Cosine Function.

This is another function similar to the first. In this case, Brent is fastest. Regula Falsi takes more steps to converge, but has very steep final convergence. Tests indicate that although Regula Falsi happens to be a little slow here, it is nearly as fast as Brent for most initial intervals. Ridder and Zhang have very similar convergence and Bisection is slow.

3.6 Testing a Linear Function

Function tested

$$f(x) = 1 - \tfrac{3}{4}x$$

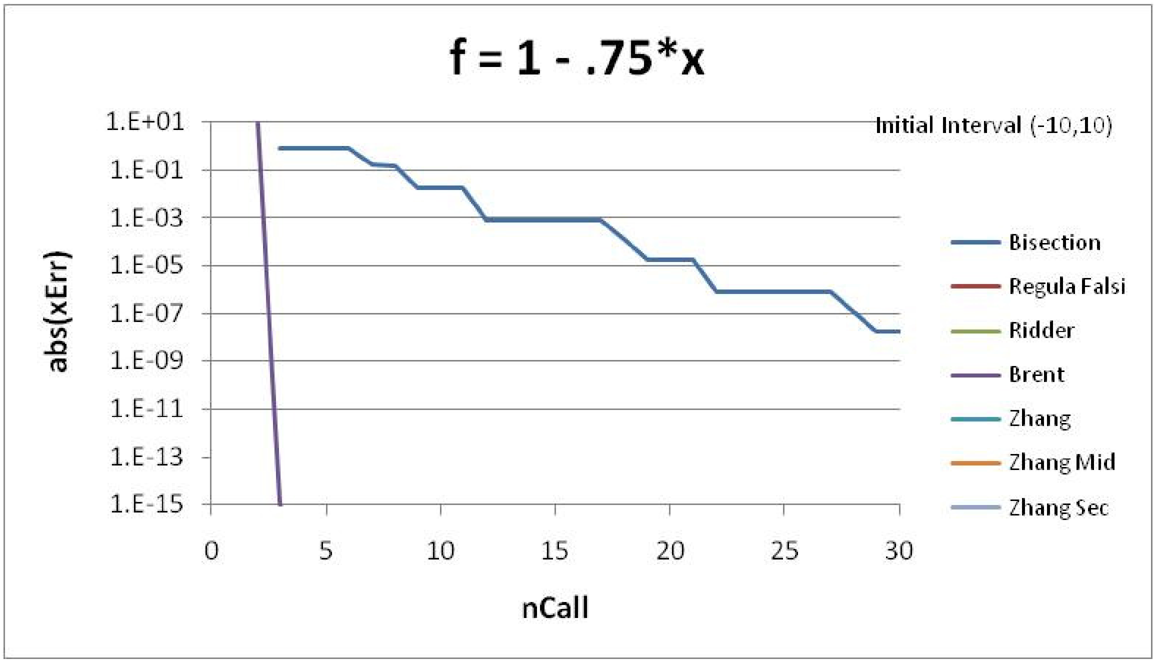

FIGURE 4: Convergence for a Linear Function.

This function is linear. As expected, the Regula Falsi, Brent, and Zhang (all options) methods are able to exactly find the root after only 3 or 4 function evaluations. Bisection converges slowly.

3.7 Testing An Inverse Quadratic Function

Function tested

$$\mathrm{f}\left(x\right)=\begin{Bmatrix} \left|x-\tfrac{2}{3}\right|^{0.5}-0.1 & \text{if } x\le \tfrac{2}{3} \\ -\left|x-\tfrac{2}{3}\right|^{0.5}-0.1 & \text{otherwise} \end{Bmatrix}$$

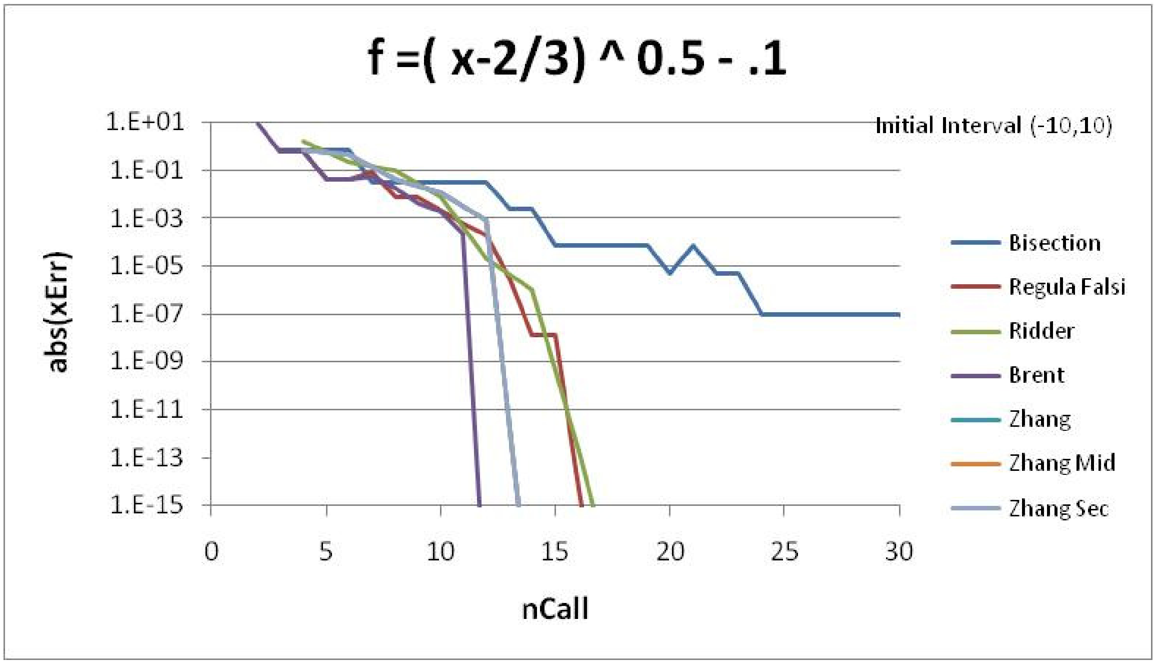

FIGURE 5: Convergence for an Inverse Quadratic Function.

The two parts of this function are inverse quadratic equations. The Brent and Zhang methods use inverse quadratic interpolation and immediately converge to the exact solution once the evaluation interval is small enough that it does not include x=2/3, where the equation has its transition point. Regula Falsi and Ridder also do very well for this function. Tests using various initial intervals indicate that Brent converges first, followed by Zhang, Regula Falsi, Ridder and Bisection, in that order.

3.8 Testing a More Difficult Inverse Quadratic Function

Function tested

$$f(x)=\begin{cases} \left|x-\tfrac{2}{3}\right|^{0.5} & \text{if } x\le \tfrac{2}{3} \\ -\left|x-\tfrac{2}{3}\right|^{0.5} & \text{otherwise} \end{cases}$$

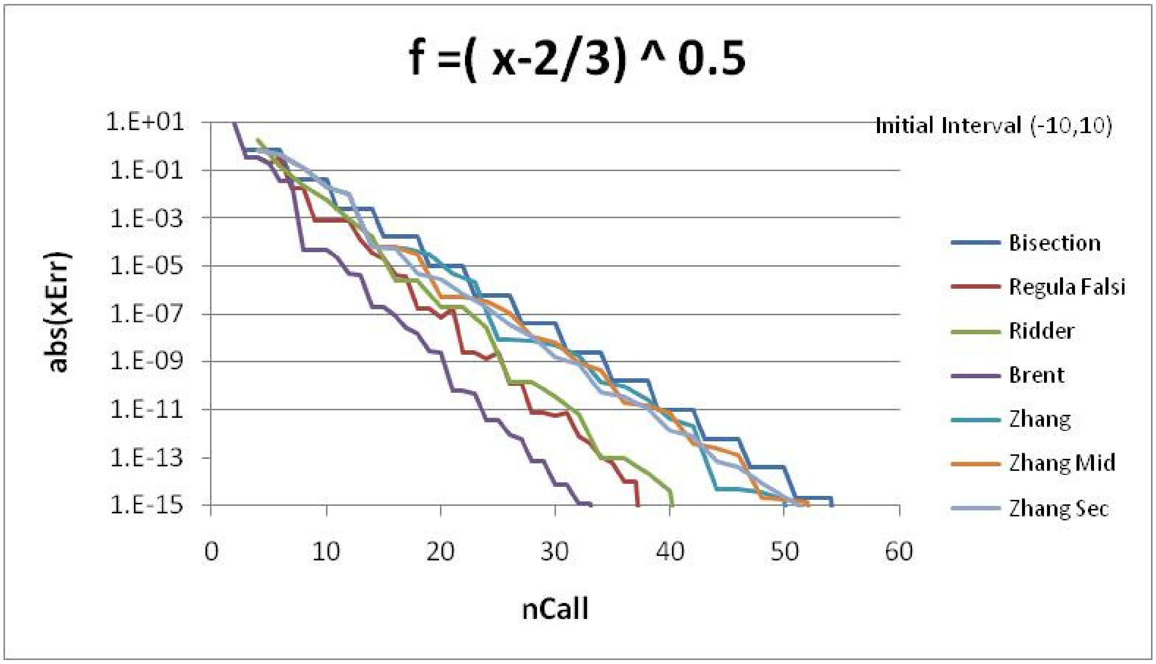

FIGURE 6: Convergence for an Inverse Quadratic Function.

The two parts of this function are inverse quadratic equations; however, the root of this function is at the point where the two sections of the curve join and the derivative is infinite there. As a result, the inverse quadratic interpolations in the Brent and Zhang methods are never able to find the exact root in a single step.

For all initial intervals tested, Brent, Regula Falsi, and Ridder gave the fastest convergence, in that order. In this case, the Zhang method produces a step when s is not in the interval (a,b), thus the three options for Zhang do not give identical results, however, the three options have indistinguishable rates of convergence and are as slow as bisection. The slope of the Brent, Regula Falsi, and Ridder methods are equal to each other and only very slightly steeper than the slopes of the Zhang curves and Bisection. The steepness of this function at the root and the discontinuous derivatives there frustrate final convergence of all of methods and cause them to be nearly as slow as bisection.

3.9 Testing an Inverse 5th Order Polynomial Function

Function tested

$$f(x)=\begin{Bmatrix} \left|x-\frac{2}{3}\right|^{0.2} & \text{if } x\le\frac{2}{3} \\ -\left|x-\frac{2}{3}\right|^{0.2} & \text{otherwise} \end{Bmatrix}$$

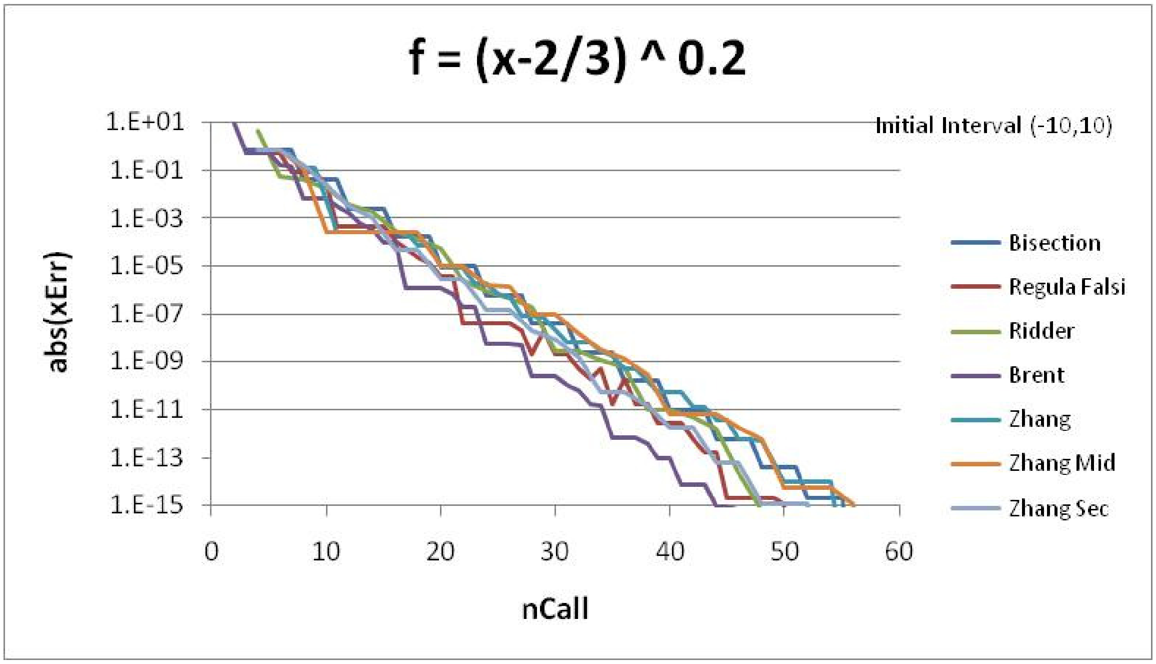

FIGURE 7: Convergence for an Inverse 5th Order Polynomial.

The two parts of this function are inverse 5th order polynomials. As with the previous function, the root of this function is at the point where the two sections of the curve join and the derivative is infinite there making root finding challenging. As in the previous example, all methods have slow convergence similar to Bisection.

3.10 Testing an Cubic Polynomial Function

Function tested

$$f(x) = \left(x - \tfrac{7}{9}\right)^3 + 0.001\left(x - \tfrac{7}{9}\right)$$

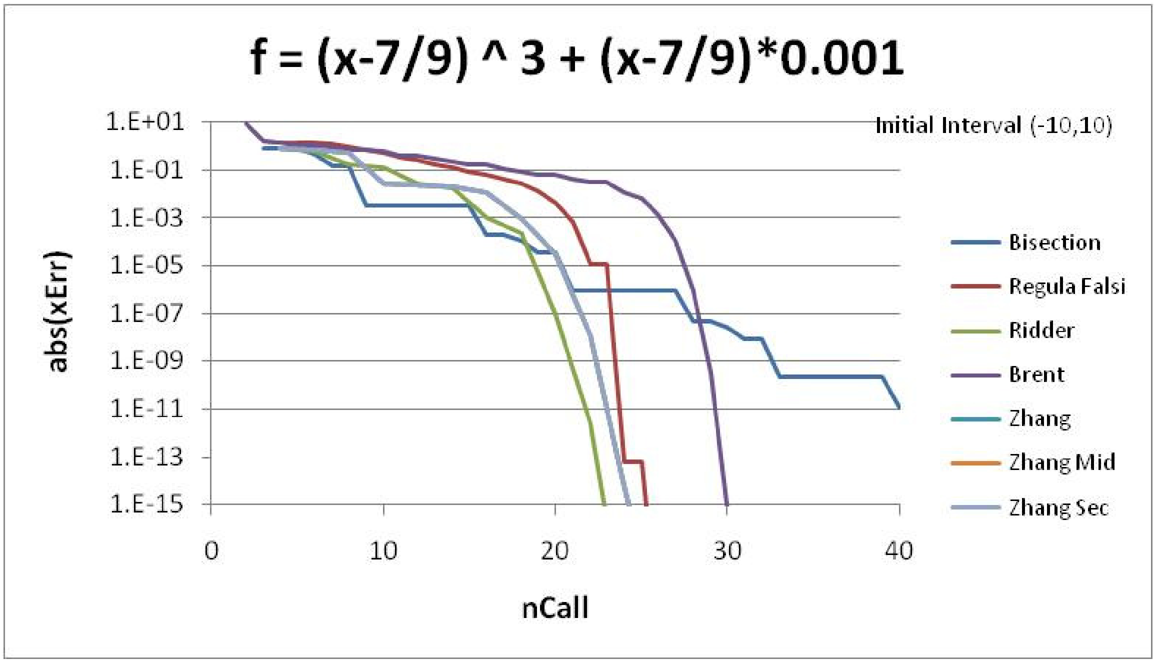

FIGURE 8: Convergence for a Cubic Polynomial.

This function is a cubic polynomial with root at 7/9. This case demonstrates the behavior of the methods during two distinct phases of convergence. Away from the root, the cubic term dominates causing slow initial convergence for all methods. As the root is approached, the cubic term becomes less important, the linear term dominates, and the root-finding methods are able to rapidly converge.

Bisection has the same, slow, steady rate throughout the convergence, but actually beats the other methods initially. Among the remaining methods, initial convergence is fastest for Ridder, followed by Zhang, Regula Falsi, and Brent in that order. The speed of final convergence is the reverse with Brent fastest, followed by Regula Falsi, then Zhang and Ridder. In this example, initial interval is relatively large and the overall convergence is determined by the initial speed. If the initial interval is made larger, the initial phase is even more important and he convergence curves spread out. If the initial interval is made smaller, the curves bunch more tightly together and may change relative position.

The importance of the initial and final phases of convergence can be modified by changing the coefficient of the linear term, the larger the coefficient, the sooner the linear term takes over and the sooner final convergence sets in.

3.11 Testing a Heaviside Function

Function tested

$$f(x) = \begin{Bmatrix} -0.5 & \text{if } x \le \frac{1}{3} \\ 0.5 & \text{otherwise} \end{Bmatrix}$$

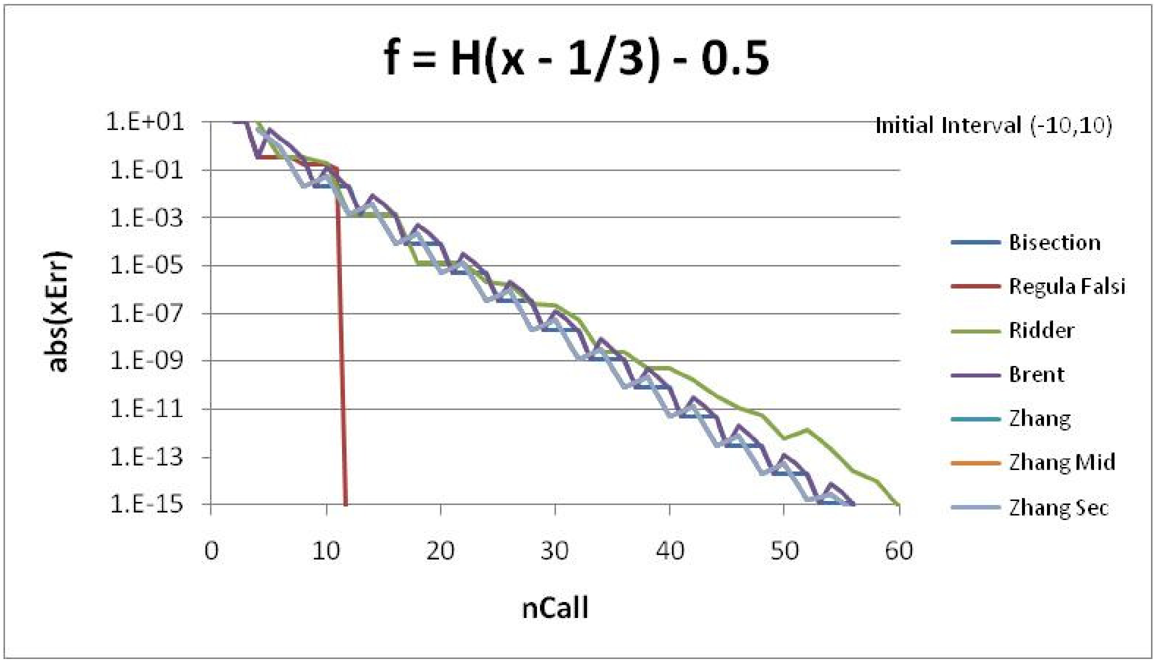

FIGURE 9: Convergence for a Heaviside Function.

This Heaviside function is discontinuous at x=1/3 and has no actual root. However it is a useful test, because all of the methods tested here use intervals that trap the root and algorithms that are guaranteed to converge to either a root or a discontinuity (see, for instance, Press, *et. al* [4]). Thus, this is a test of how quickly the methods can find the discontinuity.

The function evaluations do not contain any information about the location of the root except which subinterval the root is in. Linear (secant) interpolation of this function results in bisection of the interval thus all methods converge by bisection.

Regula Falsi finds the root to within double precision accuracy after 12 function evaluations. If the initial interval is changed the convergence curve for Regula Falsi joins the other curves. This illustrates that root finding algorithms can stumble into the root for some initial intervals, but that this is a matter of luck rather than skill. In this case, the use of the Illinois method results in a value of 1/3.

3.12 Testing a More Difficult Heaviside Function

Function tested

$$\mathrm{f}\left(x\right)=\left\{\begin{matrix} -10^{-3} & \text{if } x\le \frac{1}{3} \\ 1-10^{-3} & \text{otherwise} \end{matrix}\right\}$$

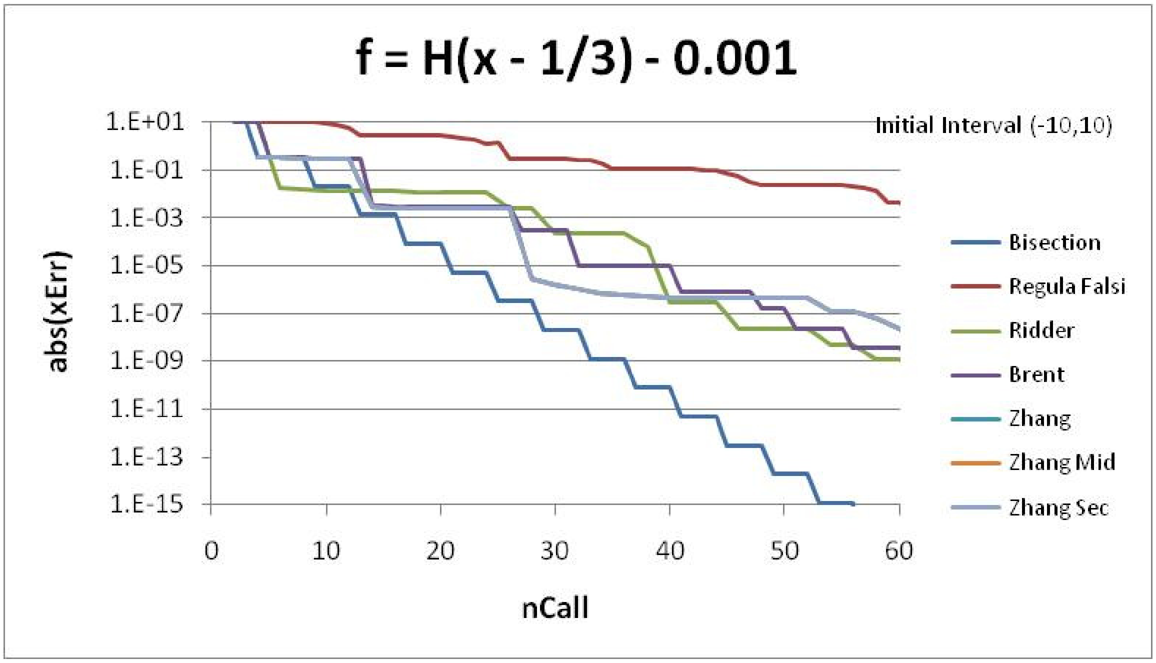

FIGURE 10: Convergence for a More Difficult Heaviside Function.

This function is a Heaviside function with a different vertical offset than the previous test. For the previous test function, use of linear (secant) interpolation resulted in bisection of the interval. For the function in this test, linear interpolation leads to a point much closer to one end of the interval than to the other end. This makes Regula Falsi very slow. Ridder, Zhang, and Brent methods use curved functions to interpolate which do better than linear interpolation but cannot find this root as rapidly as Bisection. Other initial intervals show similar results.

This test is relevant to a set of continuous functions that occur in real-world problems, namely functions that are more or less flat at one value, rapidly change over a narrow transition region, and then are relatively flat again. In these cases, the flat parts of the function do not provide much information about the location of the transition. In fact, the information actually misleads the root-finding algorithms and causes them to be slower than bisection until an evaluation point is found that is in the transition region.

3.13 Testing a Discontinuous Function

Function tested

$$\mathrm{f}\left(x\right)=\begin{Bmatrix} 0 & \text{if } x=0 \\ \left(1-\frac{2}{3}\right)^{-1} & \text{otherwise} \end{Bmatrix}$$

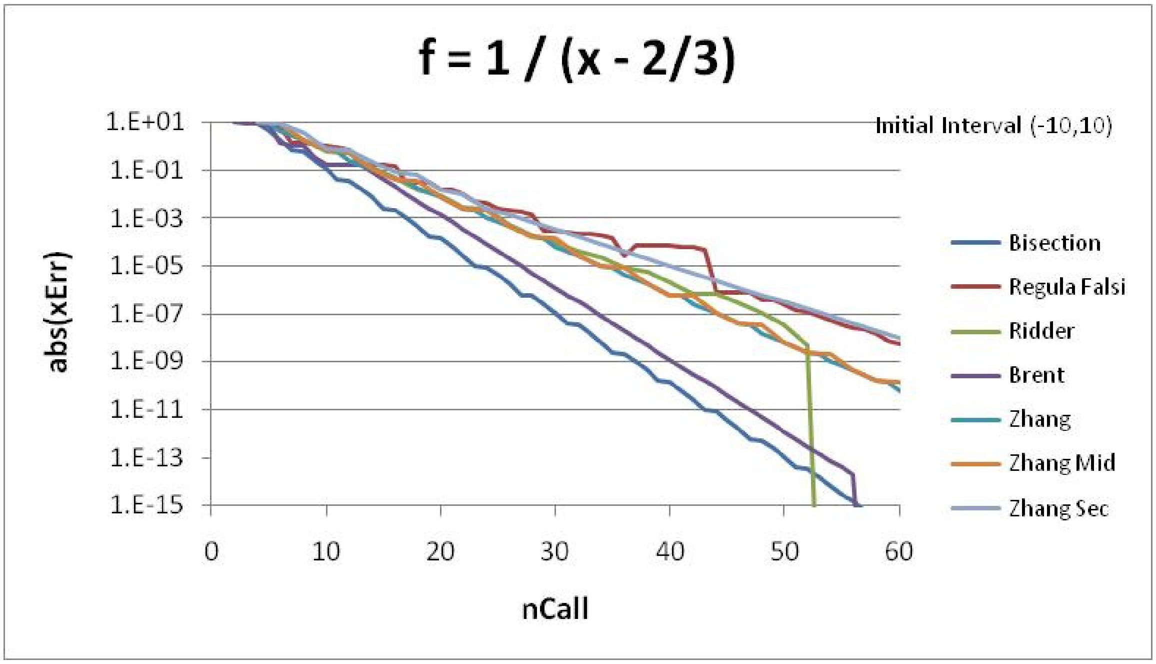

FIGURE 11: Convergence for a Discontinuous Function.

This is a particularly challenging function. It has an infinite-magnitude discontinuity rather than a root. Except for Bisection, all of the root-finding algorithms make assumptions that the function can be fit by linear or other continuous functions, which distinctly do not apply to this function.

This case produces situations when inverse quadratic interpolation produces values outside the interval, thus the three options for the Zhang method produce different convergence.

For this function, Bisection gives fastest convergence, followed by Brent. Regula Falsi gives the slowest convergence. Ridder and the three options of Zhang are intermediate with locations that change substantially depending on the initial interval. When a variety of initial intervals is considered, the three options for Zhang perform about equally.

Most real-world applications do not involve functions that are this poorly behaved; however, this test indicates how well the various methods perform when such cases do arise.

For the initial interval shown here, Ridder stumbles into the root. For an initial interval of (0,7) Brent stumbles into the root.

4. DISCUSSION AND CONCLUSIONS

The tests shown here indicate that the corrected Zhang method can compete with other well-known and commonly used root-finding methods. The methods tested here are:

- Bisection
- Regula Falsi (including the Illinois algorithm)
- Ridder
- Brent
- Zhang
- Zhang mid
- Zhang sec

The relative performance of the methods depends on:

- The function being evaluated
- The initial interval
- The tolerances desired for the root.

These tests were selected to include a range of functions from those that are smooth and obey the assumptions made by the root-finding methods to functions selected to deliberately challenge the algorithms. Root finding usually involves a slow initial phase when the interval is large followed by final convergence when the interval is small and the function can be approximated by simple interpolating functions (linear, inverse quadratic or exponential for these methods) giving much faster convergence.

Most of the literature evaluates root finding methods by the speed of final convergence, however, the results shown here demonstrate that a significant fraction of the computation can be spent during initial convergence and that this may determine which method converges first. When the initial evaluation interval is large, relatively more time is spent in the initial search. If tolerances for the root are large, an adequate root may be found without many final-convergence steps. Conversely, requiring high-accuracy roots increases the importance of using a method with rapid final convergence.

According to these tests, the Brent method tends to have relatively slow initial convergence followed by very rapid final convergence. The Regula Falsi method also tends to have slow initial convergence followed by final convergence that compares favorably with the Brent method. In these tests, Regula Falsi typically came in a close second to Brent. Regula Falsi is very easy to understand and program, whereas Brent is much more complex.

Ridder's method and Zhang's method are structurally simple and very similar to each other with the primary difference being the interpolation function used. According to these tests the two methods have similar convergence with initial convergence that is generally faster than achieved by Brent and Regula Falsi and final convergence that is generally somewhat slower. In problems having large initial intervals and/or not requiring very high accuracy roots, Ridder or Zhang can be good choices.

Any implementation of Zhang must use one of the options described above to handle cases when the inverse quadratic interpolation is outside the interval. The tests done here are not sufficient to recommend which option is best. Fortunately, this situation is rarely encountered. Furthermore, the situation occurs during initial convergence rather than during final convergence. This author favors the Zhang Secant option on purely heuristic grounds. The Zhang method with no second evaluation is simply bisection. The Zhang Mid option is equivalent to two bisections. The Zhang Secant option would seem to offer the highest order convergence for the step. At any rate, it does not appear that choice of this option has much impact on convergence. Further experience with Zhang's method in practical problems may reveal situations in which one of these options is superior to the others

Hopefully, this paper will help the readers to better appreciate the factors that contribute to performance of root finders and to help them determine which method will best meet their needs.

Most texts discuss the order of convergence and suggest higher order methods are superior. The present research indicates that this is an overly simplistic view. In many cases, initial convergence takes up a substantial fraction of the computation time. Higher-order methods are also more prone to failure when the function does not have the shape assumed when deriving the root-finding method.

It is also hoped that analysis similar to that shown here will be included with future research on root-finding algorithms and that textbooks and instruction will help students to understand these factors.

5. ACKNOWLEDGEMENTS

This work was partially funded under contract number IEM/W911SR-09-D-09-TO7-05 from the US Army Chemical Materials Agency.

6. REFERENCES

[1] Z. Zhang, An Improvement to the Brent's Method, IJEA, vol. 2, pp. 21-26, May 31, 2011.

[2] R.P. Brent. Algorithms for Minimization without Derivatives. Chapter 4. Prentice- Hall, Englewood Cliffs, NJ.

[3] H.M. Antia. *Numerical Methods for Scientists and Engineers*, Birkhäuser, 2002, pp.362-365, 2 ed.

[4] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing* Second Edition, Cambridge University Press, November 27, 1992, pp. 358–362.

[5] G. Dahlguist and A. Bjorck. *Numerical Methods*. Dover Publications, 2003, p 232.