Applied Mathematical Sciences, Vol. 4, 2010, no. 15, 709 - 718

Bisected Direct Quadratic Regula Falsi

Robert G. Gottlieb[1] **and Blair F. Thompson**

Odyssey Space Research
1120 NASA Parkway, Suite 505
Houston, Texas 77058, USA

Abstract

A new and powerful root finding algorithm that uses direct quadratic interpolation rather than inverse quadratic interpolation has been developed. The new method rapidly converges to the correct root without any need for sign checks or divide-by-zero checks. Numerical tests show the new direct method converges faster than other regula falsi root finding algorithms including Brent's inverse quadratic approach.

1 Introduction

Various numerical techniques exist for finding a root (i.e., zero) of a nonlinear function. If the function is real, continuous, and changes sign over a known interval, the method of regula falsi (or false position) can be used[2]. Depending on the characteristic "shape" of the function and the span of the initial search interval, it may take many iterations for regula falsi to converge to the root. To speed up convergence, a bisection method can be used to divide the search interval in half with each iteration[2]. The speed of convergence of the bisection method can be improved by fitting a quadratic polynomial to the function at the endpoints and midpoint of the search interval. The root of this bisected quadratic then becomes one of the endpoints of the new search interval in the next iteration. This process is repeated until convergence. As the search interval quickly decreases with each iteration, the function and the associated quadratic will become more linear over the search interval, thus improving convergence speed even more. This method of root finding, called *bisected direct quadratic regula falsi* (BDQRF), can be used in lieu of standard regula falsi to significantly reduce the number of iterations (and function calls) required for convergence. The BDQRF method is illustrated in Fig.1. BDQRF is fast, simple to implement, and will always find the desired root without the need to check for sign ambiguities or other pitfalls. Numerical tests on a variety

[1]rgottlieb@odysseysr.com

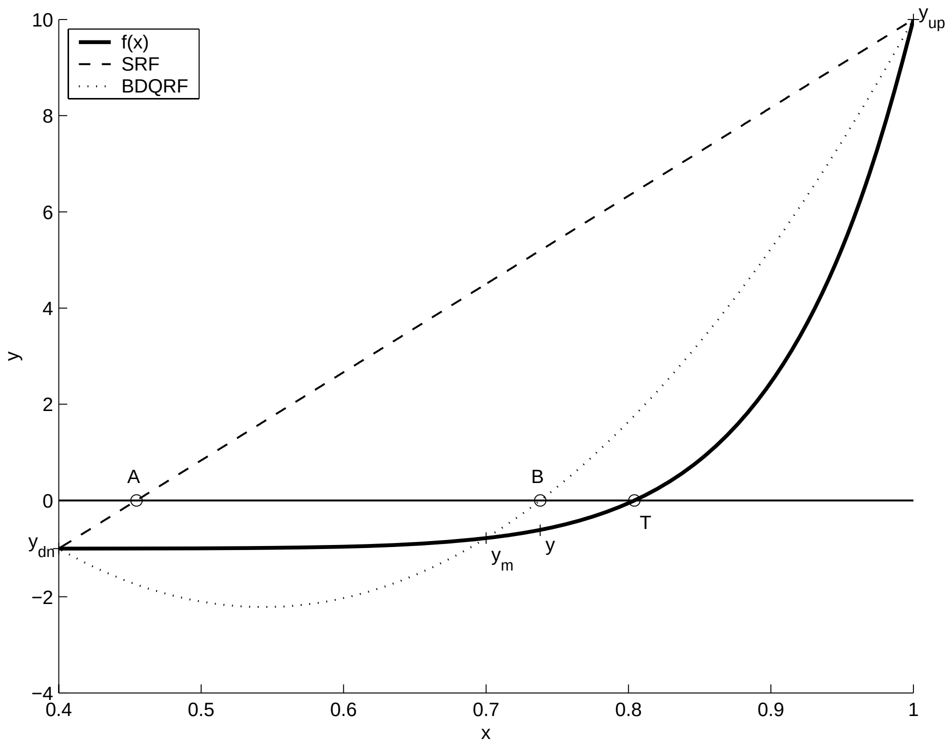

Figure 1: Root finding by regula falsi. Standard regula falsi (SRF) will select A as the root of function $f(x)$. Bisected direct quadratic regula falsi (BDQRF) will select B as the root (which is closer to the true root at T) by fitting a quadratic polynomial through function points y_{dn}, y_m and y_{up}. In this example, point y will become y_{dn} in the next iteration, significantly reducing the search interval.

of functions show that BDQRF requires fewer iterations than other regula falsi or bisection methods.

2 Derivation

Let x_{dn} and x_{up} bound the search interval in which a single root of a known function exists.[2] That is, the function is continuous and changes sign between x_{dn} and x_{up}. Define Δ as one-half the interval between x_{dn} and x_{up}.

$$\Delta \equiv \frac{(x_{up} - x_{dn})}{2} \tag{1}$$

[2]It is *not* necessary that $x_{dn} < x_{up}$, only that $x_{dn} \neq x_{up}$.

Note that Δ will always be non-zero ($\Delta \neq 0$). Define x_m as the midpoint between x_{dn} and x_{up}.

$$x_m \equiv \frac{(x_{up} + x_{dn})}{2} \tag{2}$$

The standard form of the quadratic equation is

$$y = ax^2 + bx + c \tag{3}$$

where x is the independent "input" variable. Introduce a variable, τ, which is defined to be zero at the midpoint of the $[x_{dn}, x_{up}]$ interval. Mapping from τ to x is done using

$$x = x_m + \tau \tag{4}$$

Because Δ is one-half the search interval and $\tau_m = 0$, the search interval for τ is $[\tau_{dn}, \tau_{up}] = [-\Delta, +\Delta]$. This interval for τ simplifies the solution of the quadratic equation. Once the root τ of the quadratic is found, the root x in the original search interval $[x_{dn}, x_{up}]$ can be found using equation (4).

Substituting these values

$$\tau_{dn} = -\Delta, \qquad \tau_m = 0, \qquad \tau_{up} = \Delta \tag{5}$$

for x in equation (3) results in

$$\begin{aligned} y_{dn} &= a\Delta^2 - b\Delta + c \\ y_m &= c \\ y_{up} &= a\Delta^2 + b\Delta + c \end{aligned} \tag{6}$$

These equations can be combined to form

$$\begin{aligned} s_{dn} &\equiv \frac{y_{dn} - y_m}{\Delta} = a\Delta - b \\ s_{up} &\equiv \frac{y_{up} - y_m}{\Delta} = a\Delta + b \end{aligned} \tag{7}$$

Consequently,

$$\begin{aligned} a &= \frac{s_{up} + s_{dn}}{2\Delta} = \frac{y_{up} + y_{dn} - 2y_m}{2\Delta^2} \\ b &= \frac{s_{up} - s_{dn}}{2} = \frac{y_{up} - y_{dn}}{2\Delta} \\ c &= y_m \end{aligned} \tag{8}$$

The equation for c is from equation (6). Because y_{dn} and y_{up} have opposite signs and Δ is always a non-zero number, it follows that b must always be a non-zero, real number.

With the coefficients a, b, and c of the quadratic equation known, the two roots can be computed using the well known quadratic formula.

$$\tau_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \qquad \tau_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{9}$$

Noting that

$$\tau_1 \tau_2 = \frac{b^2 - b^2 + 4ac}{4a^2} = \frac{c}{a} \tag{10}$$

the two roots of the quadratic equation can be expressed as

$$\begin{aligned} \tau_1 &= \frac{c}{a\tau_2} = \frac{2c}{-b - \sqrt{b^2 - 4ac}} = \frac{-2c}{b + \sqrt{b^2 - 4ac}} \\ \tau_2 &= \frac{c}{a\tau_1} = \frac{2c}{-b + \sqrt{b^2 - 4ac}} = \frac{-2c}{b - \sqrt{b^2 - 4ac}} \end{aligned} \tag{11}$$

The problem now is to choose the correct root, τ_1 or τ_2, that lies in the search interval $[\tau_{dn}, \tau_{up}]$. Note that if equation (3) is linear (i.e., if $a = 0$) there will be only one root.

$$0 = b\tau + c \Rightarrow \tau = -\frac{c}{b} \tag{12}$$

As the search interval becomes successively smaller with each iteration of BDQRF, the function will become more linear ($a \to 0$) over the search interval. The formulation of the root we desire is the one that reduces to $\tau = -c/b$ as $a \to 0$. An unambiguous way of forcing this to happen is to write equations (11) in the following form

$$\tau = \frac{-2c}{b\left(1 + \sqrt{1 - \frac{4ac}{b^2}}\right)} \tag{13}$$

Since b will never be zero, equation (13) will always be the correct root regardless of the sign of b. Additionally, as long as we ensure that y_{dn} and y_{up} have opposite signs, we can use the midpoint conditions from one iteration to replace the conditions at either τ_{dn} or τ_{up} in the subsequent iteration. In this way, the midpoint is not only used to compute a solution closer to the actual root, it may also replace one of the end points, thereby decreasing the search interval.

The BDQRF approach is part bisection and part quadratic regula falsi. It is simple, robust, and fast. In the case of a nearly linear function ($a \approx 0$), BDQRF may require a few more function calls than standard regula falsi. However, BDQRF will converge much more quickly in pathological situations where standard regula falsi could fail to converge within a pre-determined number of iterations.

2.1 Proof That $(b^2 - 4ac)$ Is Always Positive

In order to have full confidence in equations (9) through (13), we must ensure that $(b^2 - 4ac) > 0$ in all cases. We show this to be true by substituting equations (8) for a, b, and c in $(b^2 - 4ac)$.

$$\begin{aligned} b^2 - 4ac &= \frac{y_{up}^2 - 2y_{up}y_{dn} + y_{dn}^2}{4\Delta^2} - 8\frac{y_{up}y_m + y_{dn}y_m - 2y_m^2}{4\Delta^2} \\ &= \frac{1}{4\Delta^2}\left[y_{up}^2 + y_{dn}^2 + 16y_m^2 - 2y_{up}y_{dn} - 8y_m\left(y_{up} + y_{dn}\right)\right] \end{aligned} \tag{14}$$

Since y_m is non-zero[3], either y_{up} or y_{dn} must have the same sign as y_m. Let us assume for a moment that y_m has the same sign as y_{up} (positive by definition). Then $y_m y_{dn}$ is a negative product. Equation (14) can be written as

$$b^2 - 4ac = \frac{1}{4\Delta^2}\left[\left(y_{up} - 4y_m\right)^2 + y_{dn}^2 - 2y_{up}y_{dn} - 8y_m y_{dn}\right] \tag{15}$$

We know the product $y_{up}y_{dn}$ is negative because y_{up} and y_{dn} always have opposite sign. As stated above, $y_m y_{dn}$ is negative when $y_m > 0$. By examination of equation (15), it follows that $(b^2 - 4ac) > 0$ for any positive value of y_m.

In the case $y_m < 0$, the product $y_m y_{up}$ is negative. Equation (14) can be written as

$$b^2 - 4ac = \frac{1}{4\Delta^2}\left[\left(y_{dn} - 4y_m\right)^2 + y_{up}^2 - 2y_{up}y_{dn} - 8y_m y_{up}\right] \tag{16}$$

Again, by examination, it follows that $(b^2 - 4ac) > 0$ for any negative value of y_m. Furthermore,

$$b^2 - 4ac = b^2\left(1 - \frac{4ac}{b^2}\right) > 0 \tag{17}$$

Since b is non-zero and real, b^2 must be a positive number. Therefore, the quantity in parentheses must also be positive.

$$\left(1 - \frac{4ac}{b^2}\right) > 0 \tag{18}$$

This allows equation (13) to be used without concern about taking the square root of a negative number.

[3] If y_m (or any y) is zero then we have found the root we are seeking, and there is no need to continue.

2.2 BDQRF Algorithm Pseudocode

This section presents pseudocode of the BDQRF algorithm suitable for programming. It assumes an initial search interval $[x_1, x_2]$, the function $f(x)$, and convergence tolerance *tol*. The pseudocode is based on the derivation in the previous section, along with two additional features to make the routine more robust. First, the code assigns x_{dn} and x_{up} according to the sign of the associated y. Second, iteration is terminated if two successive estimates of x are equal, indicating the limit of computing precision has been reached.

$y = f(x_1)$
if $(y < 0)$ **then**
  $x_{dn} = x_1;\ y_{dn} = y;$
  $x_{up} = x_2;\ y_{up} = f(x_{up});$
else
  $x_{up} = x_1;\ y_{up} = y;$
  $x_{dn} = x_2;\ y_{dn} = f(x_{dn});$
end if
$x_{last} = 1.0 \times 10^{20}$ (large initial value)
while $(|y| > tol)$ **do**
  $D = (x_{up} - x_{dn})/2;$
  $x_m = (x_{up} + x_{dn})/2;$
  $y_m = f(x_m);$
  $a = (y_{up} + y_{dn} - 2y_m)/\left(2D^2\right);$
  $b = (y_{up} - y_{dn})/(2D);$
  $x = x_m - 2y_m/\left[b\left(1 + \sqrt{1 - 4ay_m/b^2}\right)\right];$
  if $(x == x_{last})$ **then**
    break;
  end if
  $x_{last} = x;$
  $y = f(x);$
  if $(y > 0)$ **then**
    $y_{up} = y;\ x_{up} = x;$
    if $(y_m < 0)$ **then**
      $y_{dn} = y_m;\ x_{dn} = x_m;$
    end if
  else
    $y_{dn} = y;\ x_{dn} = x;$
    if $(y_m > 0)$ **then**
      $y_{up} = y_m;\ x_{up} = x_m;$
    end if
  end if
end while

2.3 Inverse Quadratic Fit - Brent's Method

During numerical evaluation and testing (cf. section 3) we compare BDQRF to different regula falsi methods including the inverse quadratic fit which is the basis of Brent's method[1]. The method fits x as a function of y instead of y as a function of x. Assume we have three points (y_1, x_1), (y_2, x_2), (y_3, x_3). We want to fit the function $x = ay^2 + by + c$ to these points. This leads to

$$\begin{aligned} x_1 = ay_1^2 + by_1 + c \Rightarrow \frac{x_1}{y_1} \equiv s_1 = ay_1 + b + \frac{c}{y_1} \\ x_2 = ay_2^2 + by_2 + c \Rightarrow \frac{x_2}{y_2} \equiv s_2 = ay_2 + b + \frac{c}{y_2} \\ x_3 = ay_3^2 + by_3 + c \Rightarrow \frac{x_3}{y_3} \equiv s_3 = ay_3 + b + \frac{c}{y_3} \end{aligned} \tag{19}$$

Since we only want c (the value of x when y is zero), we can construct

$$\begin{aligned} s_1 - s_2 = a(y_1 - y_2) + c\left(\frac{1}{y_1} - \frac{1}{y_2}\right) = a(y_1 - y_2) - c\left(\frac{y_1 - y_2}{y_1 y_2}\right) \\ s_3 - s_2 = a(y_3 - y_2) + c\left(\frac{1}{y_3} - \frac{1}{y_2}\right) = a(y_3 - y_2) - c\left(\frac{y_3 - y_2}{y_3 y_2}\right) \end{aligned} \tag{20}$$

We can now construct

$$\begin{aligned} \frac{s_1 - s_2}{y_1 - y_2} \equiv d_{12} = a - c\frac{1}{y_1 y_2} \\ \frac{s_3 - s_2}{y_3 - y_2} \equiv d_{32} = a - c\frac{1}{y_3 y_2} \end{aligned} \tag{21}$$

But this lead to

$$d_{12} - d_{32} = -c\frac{1}{y_1 y_2} + c\frac{1}{y_3 y_2} = \frac{c}{y_2}\left(\frac{1}{y_3} - \frac{1}{y_1}\right) \Rightarrow \tag{22}$$

$$c = \frac{y_1 y_2 y_3 (d_{12} - d_{32})}{y_1 - y_3} \tag{23}$$

We want to express c in terms of the initial variables.

$$\begin{aligned} d_{12} - d_{32} &= \frac{s_1 - s_2}{y_1 - y_2} - \frac{s_3 - s_2}{y_3 - y_2} \\ &= \frac{s_1 y_3 - s_2 y_3 - s_1 y_2 - s_3 y_1 + s_2 y_1 + s_3 y_2}{(y_1 - y_2)(y_3 - y_2)} \\ &= \frac{s_1 (y_3 - y_2) + s_2 (y_1 - y_3) - s_3 (y_1 - y_2)}{(y_1 - y_2)(y_3 - y_2)} \end{aligned} \tag{24}$$

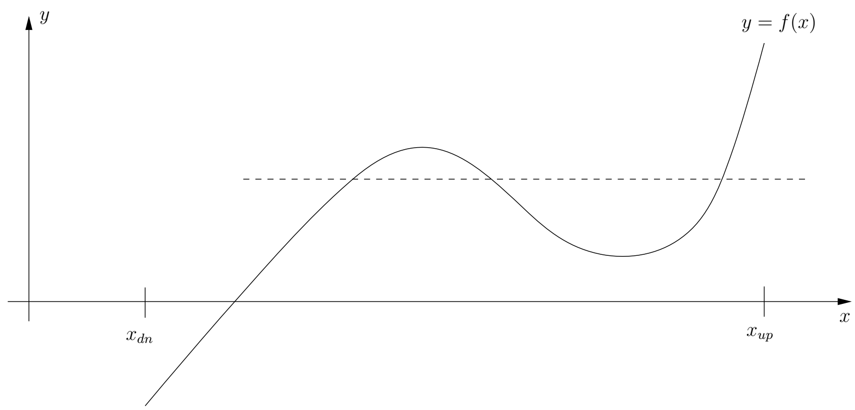

Figure 2: Example of three equal values of $y = f(x)$ in one root search interval.

Finally, we get the following equation for c

$$\begin{aligned} c &= y_1 y_2 y_3 \frac{s_1 (y_3 - y_2) + s_2 (y_1 - y_3) - s_3 (y_1 - y_2)}{(y_1 - y_2)(y_3 - y_2)(y_1 - y_3)} \\ &= \frac{x_1 y_2 y_3}{(y_1 - y_2)(y_1 - y_3)} + \frac{y_1 x_2 y_3}{(y_2 - y_1)(y_2 - y_3)} + \frac{y_1 y_2 x_3}{(y_3 - y_1)(y_3 - y_2)} \end{aligned} \tag{25}$$

This is the estimate of x when y is zero. This form of the equation reveals a pitfall of the inverse method. If the function is double- or triple-valued in y, we run the risk of a bad iteration due to divide-by-zero problems (see Fig.2). The BDQRF method does not suffer from this limitation.

3 Numerical Testing

To numerically evaluate the BDQRF method, the functions and initial intervals shown in Table 1 were solved (i.e., the root found) by various regula falsi methods: standard regula falsi (SRF), the improved regula falsi method of Naghipoor, Ahmadian, and Soheili[3] (NIRF)[4], bisected inverse quadratic regula falsi - Brent's approach (BIQRF), and bisected direct quadratic regula falsi (BDQRF). These test functions were taken from Naghipoor[3] plus an additional test function of our own (Table 1). The numerical test results (i.e., number of function calls) are summarized in Table 2.

The test results show that BDQRF rapidly converged on the root in each case. BDQRF required fewer function calls than standard regula falsi or NIRF, and required the same of fewer function calls than BIQRF which uses the inverse fit approach of Brent's method.

[4]The NIRF method did not work as published. After correcting the algorithm, we found it performed slightly better (i.e., fewer iterations) than the published results[3].

Table 1: Test case functions and initial search intervals.

Test No.	Function	Initial Interval
1	$3\sin x - 2 = 0$	$[0, 1]$
2	$xe^x - 1 = 0$	$[-1, 1]$
3	$11x^{11} - 1 = 0$	$[0.1, 0.9]$
4	$e^{x^2+7x-30} - 1 = 0$	$[2.8, 3.1]$
5	$1/x - \sin x + 1 = 0$	$[-1.3, -0.5]$
6	$x^3 - 2x - 5 = 0$	$[2, 3]$
7	$1/x - 1 = 0$	$[0.5, 2.0]$

Table 2: Number of function calls required for convergence for various regula falsi methods: standard regula falsi (SRF), Naghipoor *et al.* improved regula falsi (NIRF), bisected inverse quadratic regula falsi (BIQRF), and bisected direct quadratic regula falsi (BDQRF). The tolerance was 10^{-10}.

Test No.	Fn Calls: SRF	NIRF	BIQRF	BDQRF	x
1	16	9	10	8	0.7297276562
2	24	13	12	8	0.5671432904
3	38	17	26	10	0.8041330975
4	40	15	10	10	3.0000000000
5	16	11	10	8	−0.6294464841
6	26	11	10	6	2.0945514815
7	35	14	10	10	1.0000000000

4 Conclusions

The bisected direct quadratic regula falsi (BDQRF) method is a powerful root finding technique. It converges quickly to the correct root without any need for sign checks or divide-by-zero checks. The BDQRF method was shown to be superior to the inverse quadratic fit approach which is the basis of Brent's method.

References

[1] Brent, R.P., *Algorithms for Minimization Without Derivatives*, Prentice Hall, Englewood Cliffs, New Jersey, 1973, Ch. 3-4.

[2] Gerald, C.F., and Wheatley, P.O., *Applied Numerical Analysis*, 6th ed., Addison-Wesley, Reading, Massachusetts, 1999.

[3] Naghipoor, J., Ahmadian S.A., and Soheili, A.R., "An Improved Regula Falsi Method for Finding Simple Zeros of Nonlinear Equations," *Applied Mathematical Sciences*, Vol. 2, No. 8, 2008, pp.381-386.

Received: August, 2009