A remarkable variant of the Secant method to find a root

Soubhadra Sen [a], N. Mohankumar [b]

a. Radiological Safety Division, IGCAR, Kalpakkam, India 603102, Email: ssen@igcar.gov.in

b. 40, D J Nagar, Coimbatore, India 641004, Email: kovainmk@gmail.com

Abstract

An accurate and efficient approach based on the conventional Secant method is proposed to find a root of an algebraic or transcendental equation. This root finder needs two function evaluations in each iteration and it has the term 2.414 as the order of convergence. The root can be bracketed by adopting a Regula-Falsi like technique. The superiority of this method over the existing methods is demonstrated with a set of test problems.

Key words: Non-linear equation, roots, iterative algorithm, Secant and Predictor-Corrector methods, Regula-Falsi algorithm.

MSC: 65H04

1. Introduction

The precise estimation of the roots of algebraic and transcendental equations is a practical requirement in many problems of physical sciences for which the classical Newton-Raphson (NR) method is one of the most preferred tools. In this method in order to find the root of the equation $f(x) = 0$, one starts with a trial root x_0 that is improved by the following iteration.

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \tag{1}$$

In the Secant method which is a variant of the NR method, the derivative term in the above equation $f'(x_i)$ is replaced by its approximant $\left[\frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}\right]$ and this gives the modified iteration indicated below.

$$x_{i+1} = x_i - \left[\frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})}\right] f(x_i) \tag{2}$$

The NR method has a quadratic convergence while the Secant approach converges super-linearly with an order term 1.618 [1]. There exist several algorithms with a higher rate of convergence but in these cases one needs to evaluate the higher-order derivatives additionally [2, 3]. Another popular alternative way to increase the convergence rate is by adopting a multistep predictor-corrector method [1, 4, 5]. Here, we report a new scheme where the rate of convergence of the conventional Secant method is improved significantly by a two step variation.

2. The new method

The new algorithm uses a two step, predictor-corrector Secant method for the evaluation of the root. The calculation involves the following steps.

$$x_i^* = x_i - \left[\frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})}\right] f(x_i) \tag{3}$$

$$x_{i+1} = x_i - \left[\frac{x_i - x_i^*}{f(x_i) - f(x_i^*)}\right] f(x_i) \tag{4}$$

One may note that the intermediate value x_i^* is used in the second step for the derivative evaluation that improves the rate of convergence. It is pertinent to note that we need $f(x_i^*)$ in addition to $f(x_i)$ in each loop while $f(x_{i-1})$ is evaluated in the previous iteration.

3. Brackting of the root

The NR and the Secant methods are open ended ones where the root is not bracketed by two approximate values. For the Secant method, one normally resorts to the Regula-Falsi (RF) technique to bound an isolated root. For the present scheme too, we follow the same bracketing method with the following variations.

Case I: During a particular iteration, x_{i+1} may fall outside the interval $[x_i, x_{i-1}]$. In that case, we replace x_{i+1} with x_i^* and then choose either $[x_i, x_{i+1}]$ or $[x_{i+1}, x_{i-1}]$ where the actual root lies by the RF technique.

Case II: On the otherhand, x_{i+1} may lie between x_i and x_{i-1}. In this case, the required root lies in one of the three intervals, i.e. $[x_i, x_{i+1}]$, $[x_{i+1}, x_{i-1}]$ and $[x_{i+1}, x_i^*]$ and the interval with the root is obtained once again by the RF method.

4. Convergence

Let us write $x_i^* = r + \epsilon_i^*$ and $x_i = r + \epsilon_i$ where r is the actual root and ϵ is the error. Then from Eq. (3) we get

$$r + \epsilon_i^* = r + \epsilon_i - \left[\frac{\epsilon_i - \epsilon_{i-1}}{f(r + \epsilon_i) - f(r + \epsilon_{i-1})}\right] f(r + \epsilon_i) \tag{5}$$

Now expanding $f(r+\epsilon_i)$ in a Taylor series and keeping terms up to $f''(r)$ one gets

$$\epsilon_i^* \simeq \alpha\epsilon_i\epsilon_{i-1} \quad ; \quad \alpha = \frac{f''(r)}{2f'(r)} \tag{6}$$

This is the expression for the conventional (single step) Secant method [6]. Substituting ϵ_i^* in Eq. (4) we get

$$r+\epsilon_{i+1} = r+\epsilon_i - \left[\frac{\epsilon_i - \alpha\epsilon_i\epsilon_{i-1}}{f(r+\epsilon_i) - f(r+\alpha\epsilon_i\epsilon_{i-1})}\right] f(r+\epsilon_i) \tag{7}$$

After simplification this reduces to

$$\epsilon_{i+1} \simeq \alpha^2\epsilon_i^2\epsilon_{i-1} \tag{8}$$

Now, let ϵ_i and ϵ_{i+1} be connected by the following relation.

$$|\epsilon_{i+1}| = C|\epsilon_i|^P \tag{9}$$

From Eqs. $(8,9)$ we get

$$|\epsilon_i| = \left(\frac{|\alpha^2|}{C}|\epsilon_{i-1}|\right)^{\frac{1}{P-2}} \tag{10}$$

Upon comparing the order expressions from Eq. (9) and Eq. (10), we get $P = \dfrac{1}{P-2}$. Since $P > 0$, one obtains finally

$$P = \left(1+\sqrt{2}\right) = 2.414 > 2 \tag{11}$$

Thus the rate of convergence of the present method is more than quadratic. One must note that this is the estimation of the convergence rate for the open algorithm. To bracket a root, for this new algorithm one may resort to a RF like method. In that scenario, the following cases may arise.

Case A: As discussed earlier in Case I, in a particular loop the root x_{i+1} may fall outside the interval $[x_i, x_{i-1}]$. Then, we replace x_{i+1} with x_i^* and the present method converges like the conventional RF method. For the new method, this sets the lower bound for the convergence rate.

Case B: In a particular loops, x_{i+1} may fall between x_i and x_{i-1} but the actual root does not lie between x_{i+1} and x_i^*. In this case, the rate of convergence will be the same for the open and the bracketed versions.

Case C: In a loop, it may so happen that x_{i+1} lies between x_i and x_{i-1} and the actual root is bracketed by x_{i+1} and x_i^*. This is similar to Case B but with the corresponding intervals being much shorter. Hence the convergence will be relatively faster (even better than that of the open version).

During a root evaluation, the iterations involve a combination of the above mentioned cases and the overall rate of convergence of the bracketed version will fall between those of cases C and A.

5. Numerical examples

We solve a set of standard test problems using this new method to demonstrate its efficiency. For comparison, these problems are also evaluated by other well known schemes like the NR, the Secant, the Secant method with RF and the Ridders' method [7]. The results are tabulated in table 1. The following test problems are tried.

Prob. 1:

$$f(x) = sin^2(x) - x^2 + 1 \quad ; \quad x_0 = 1 \quad , \quad x_1 = 3 \tag{12}$$

Prob. 2:

$$f(x) = sin^2(x) - x^2 + 1 \quad ; \quad x_0 = 3 \quad , \quad x_1 = 1 \tag{13}$$

Prob. 3:

$$f(x) = x^2 - e^x - 3x + 2 \quad ; \quad x_0 = -50000000 \quad , \quad x_1 = 3 \tag{14}$$

Prob. 4:

$$f(x) = x^2 - e^x - 3x + 2 \quad ; \quad x_0 = 3 \quad , \quad x_1 = -50000000 \tag{15}$$

Prob. 5:

$$f(x) = xe^x - 10 \quad ; \quad x_0 = 0 \quad , \quad x_1 = 2 \tag{16}$$

Prob. 6:

$$f(x) = cos(x) \quad ; \quad x_0 = 100^o \quad , \quad x_1 = 280^o \tag{17}$$

Prob. 7:

$$f(x) = sin(x) \quad ; \quad x_0 = 10^o \quad , \quad x_1 = 280^o \tag{18}$$

Prob. 8:

$$f(x) = x^3 - 2x - 5 \quad ; \quad x_0 = 2.5 \quad , \quad x_1 = 0.01 \tag{19}$$

All the calculations are performed in double precision. If convergence of the root to the correct value is not obtained after around 10000 iterations, then it is termed a 'failure' in the tabulation. Barring problem 6, the open version of this new algorithm is better than the other two open methods (the NR and the conventional Secant). If the comparison is made among the bounded methods (between the Secant with RF, the Ridders' method and the bracketed version of this new algorithm), with the exception of problem 5, the new method has an edge over the other methods in this category. It is to be noted that the NR, the Ridders' and both the versions of this new scheme require two function evaluations per iteration while the Secant method with and without RF requires only one function evaluation per loop.

Table 1: Comparison of the number of iterations needed for convergence for different methods

Test Problem	Root	Newton Raphson	Secant method	Secant with Regula-Falsi	Ridders' method	New method (open ended)	New method with bracket
Prob. 1	1.4044916482153	6	8	46	6	5	5
Prob. 2	1.4044916482153	6	8	46	6	4	4
Prob. 3	0.2575302854399	6	8	Failure	27	5	5
Prob. 4	0.2575302854399	29	9	Failure	27	6	6
Prob. 5	1.7455280027407	5	8	19	4	4	4
Prob. 6	270.00000000000	3	6	5	7	3	3
Prob. 7	180.00000000000	Failure	Failure	7	7	4	4
Prob. 8	2.0945514815423	22	13	21	6	10	5

6. Conclusions

In this paper, we have reported a two-step algorithm based on the conventional Secant method for finding a root that has a convergence rate better than quadratic. The performance of this algorithm (in both open and bounded forms) is compared with that of the existing classical and new methods for a set of standard test problems. It is obvious that the new method performs reliably and more efficiently than the other standard methods.

7. The numerical codes

Here, we provide the numerical codes for the two versions (open and bracketed) of the new method. One may run these programs using Octave 4.0 or Matlab 5.3. First, the code for the open algorithm is listed. It needs x1 (initial first guess of the root), x0 (initial second guess of the root) and limit (accuracy of convergence) as the input parameters. The equation under consideration is to be given in a separate function called func(). The following is the said code.

```
***************************************************************
function rootval=senmohan(x1, x0, limit)
% A variant of the Secant method by Soubhadra Sen and N. Mohankumar.
% This is the open alogrithm that means a root is not bracketed by the
% bounds.If one needs to evaluate a root of the equation f(x)=0 with two
% initial guesses x1 and x0, the following inputs are to be entered.
% Input: x1 (initial first guess), x0 (initial second guess),
% limit (accuracy of convergence).
% The equation is to be provided in the function named func().
%
diff=10;
i=0;
term2=func(x0);
%
while diff > limit
    i=i+1;
    term1=func(x1);
    term3=(x1-x0)*term1;
    term4=term1-term2;
    hhstar=term3/term4;
    xstar=x1-hhstar;
    chkk=abs(x1-xstar);
    %
    if chkk < limit
       break
    end
    %
    term5=func(xstar);
    hh=(x1-xstar)*term1/(term1-term5);
    xnew=x1-hh;
    diff=abs(x1-xnew);
    x0=x1;
```

```
    x1=xnew;
    term2=term1;
end
rootval=x1;
iteration=i-1;
%
rootval
iteration
```

****

Next we provide the program for the bracketed algorithm. Here also one has to enter the similar inputs along with the equation in the function named func(). The said code is given below.

****

```
function rootval=senmohanrf(x1, x0, limit)
% A variant of the Secant method by Soubhadra Sen and N. Mohankumar.
% This is the bracketed alogrithm that means a root is bounded by a
% Regula-Falsi like technique.
% If one needs to evaluate a root of the equation f(x)=0 falling
% between x1 and x0, the following input
% parameters are to be entered.
% Input: x1 (bound at one end), x0 (bound at other end),
% limit (accuracy of convergence).
% The equation is to be provided in the function func().
%
diff=10;
i=0;
term1=func(x1);
term2=func(x0);
%
while diff > limit
    i=i+1;
    %
    term3=(x1-x0)*term1;
    term4=term1-term2;
```

```
hhstar=term3/term4;
xstar=x1-hhstar;
chkk=abs(x1-xstar);
%
if chkk < limit
   break
end
%
term5=func(xstar);
hh=(x1-xstar)*term1/(term1-term5);
xnew=x1-hh;
diff=abs(x1-xnew);
%
term7=func(xnew);
tt1=term1*term7;
tt2=term1*term5;
tt3=term5*term7;
tt4=term2*term7;
%
ad1=xnew-x1;
ad2=x0-xnew;
pdt=ad1*ad2;
if pdt < 0
   if tt2 < 0
      x0=x1;
      term2=term1;
   end
x1=xstar;
term1=term5;
%
elseif pdt > 0
     if tt3 < 0
        x0=xstar;
        term2=term5;
     elseif tt1 < 0
```

```
            x0=x1;
            term2=term1;
        end
        x1=xnew;
        term1=term7;
    end
    %
end
rootval=x1;
iteration=i-1;
%
rootval
iteration
```

****

8. References

[1] T. J. McDougalla, S. J. Wotherspoon, A simple modification of Newtons method to achieve convergence of order $1+\sqrt{2}$, Appl. Math. Lett. 29 (2014) 20-25.

[2] T. R. Scavo, J. B. Thoo, On the Geometry of Halleys Method, Amer. Math. Monthly 102 (5) (1995) 417-426.

[3] A. S. Householder, The Numerical Treatment of a Single Nonlinear Equation, McGraw-Hill, New York, 1970.

[4] S. Weerakoon, T. G. I. Fernando, A variant of Newtons method with accelerated third-order convergence, Appl. Math. Lett. 13 (2000) 87-93.

[5] H. Homeier, A modified Newton method for root finding with cubic convergence, J. Comput. Appl. Math. 157 (1) (2003) 227-230.

[6] P. Diez, A Note on the Convergence of the Secant Method for Simple and Multiple Roots, Appl. Math. Lett. 16 (2003) 1211-1215.

[7] C. J. F. Ridders, A New Algorithm for Computing a Single Root of a Real Continuous Function, IEEE Trans. Circuits Syst. CAS-26 (1979) 979-980.