

PERTURBATIVE DERIVATION AND COMPARISONS OF ROOT-FINDING ALGORITHMS WITH FOURTH ORDER DERIVATIVES

M. Pakdemirli*, H. Boyacı* and H. A. Yurtsever**

*Department of Mechanical Engineering, Celal Bayar University
Muradiye, Manisa, Turkey. mpak@bayar.edu.tr, hakan.boyaci@bayar.edu.tr

**Department of Mathematics, Georgetown University, Washington DC

Abstract – Perturbation theory is systematically used to generate root finding algorithms with fourth order derivatives. Depending on the number of correction terms in the perturbation expansion and the number of Taylor expansion terms, different root finding formulas can be generated. Expanding Taylor series up to fourth order derivatives and taking two, three and four correction terms in the perturbation expansions, three different root finding algorithms are derived. The algorithms are contrasted numerically with each other as well as with the Newton-Raphson algorithm. It is found that the quadruple-correction-term algorithm performs better than the others.

Keywords – Root finding algorithms, perturbation theory, Newton-Raphson method

1. INTRODUCTION

Perturbation theory is well established and used in search of approximate solutions of algebraic equations, differential equations, integro-differential equations, difference equations etc. In finding the roots of a function, perturbation methods can be used. Many examples of algebraic equations with small parameters were treated in the book by Nayfeh [1]. By expanding the root in a perturbation series, each correction term was calculated in order to find the approximate root. As is usual in perturbation methods, the correction terms were calculated once and no iterations over the corrections were made. In the book by Hinch [2], the perturbation method and iteration method were treated as separate methods. A combination of perturbations with iterations or the so-called “perturbation-iteration method” would be a better choice. In fact, the well known formulas such as Newton Raphson and its second and higher order corrections, namely the Householder’s iteration and Schroder family [3] can be derived from perturbations. Depending on the number of terms taken in the perturbation expansion, on the number of terms in the Taylor expansion and the way the resulting equations are separated, different iteration formulas which may or may not belong to the mentioned class of iteration formulas can be generated.

The link between perturbations and root finding algorithms was exploited in a recent work [4]. Root finding formulas consisting of up to third order derivatives were derived in that work. With referral to the number of terms taken in the perturbation expansions, formulas were classified as single-correction-term algorithms, double-correction-term algorithms, triple-correction-term algorithms. Taylor expansions were taken up to third order derivatives in that work. The root finding algorithms were contrasted with those derived by Abbasbandy [5] using modified Adomian decomposition method.

In this work, Taylor series expansions are carried up to fourth order derivatives. In the perturbation expansions, two correction, three correction and four correction terms are taken each leading to a different root finding algorithm (double-correction-

term algorithm, triple-correction-term algorithm, and quadruple-correction-term algorithm in our referral). Numerical comparisons of the three formulas derived and Newton-Raphson formula yield that the higher order algorithms perform better than the Newton-Raphson method with the best being quadruple-correction-term algorithm. Based on this work as well as on the previous work [4], one may conclude that the best algorithms can be derived by taking the same number of correction terms both in Taylor expansions and perturbation expansions.

2. PERTURBATION-ITERATION METHOD

In this section, three single point iteration formulas consisting fourth order derivatives will be derived by using perturbation theory. The formulas are classified with respect to the number of correction terms in the perturbation expansion.

2.1. Double-Correction-Term Algorithm

To find the roots of the nonlinear equation

$$f(x) = 0 \quad (1)$$

one may assume a perturbation expansion of the below form with two correction terms

$$x = x_0 + \varepsilon x_1 + \varepsilon^2 x_2 \quad (2)$$

Inserting (2) into (1) and expanding in a Taylor series up to fourth order derivative terms yields

$$f(x_0 + \varepsilon x_1 + \varepsilon^2 x_2) \cong f(x_0) + f'(x_0)(\varepsilon x_1 + \varepsilon^2 x_2) + \frac{f''(x_0)}{2!}(\varepsilon x_1 + \varepsilon^2 x_2)^2 + \frac{f'''(x_0)}{3!}(\varepsilon x_1 + \varepsilon^2 x_2)^3 + \frac{f^{iv}(x_0)}{4!}(\varepsilon x_1 + \varepsilon^2 x_2)^4 = 0 \quad (3)$$

Re-arranging the terms with respect to similar powers of ε , one has

$$f(x_0 + \varepsilon x_1 + \varepsilon^2 x_2) \cong f(x_0) + \varepsilon(x_1 f'(x_0)) + \varepsilon^2 \left[x_2 f'(x_0) + \frac{1}{2} x_1^2 f''(x_0) \right] + \varepsilon^3 \left[x_1 x_2 f''(x_0) + \frac{1}{6} x_1^3 f'''(x_0) \right] + \varepsilon^4 \left[\frac{1}{2} x_2^2 f''(x_0) + \frac{1}{2} x_1^2 x_2 f'''(x_0) + \frac{1}{24} x_1^4 f^{iv}(x_0) \right] = 0 \quad (4)$$

Equation (4) contains two unknowns x_1 and x_2 which requires two equations to be solved. Hence equation (4) is separated into two blocks as follows

$$f(x_0) + \varepsilon(x_1 f'(x_0)) = 0 \quad (5)$$

$$\varepsilon^2 \left[x_2 f'(x_0) + \frac{1}{2} x_1^2 f''(x_0) \right] + \varepsilon^3 \left[x_1 x_2 f''(x_0) + \frac{1}{6} x_1^3 f'''(x_0) \right] + \varepsilon^4 \left[\frac{1}{2} x_2^2 f''(x_0) + \frac{1}{2} x_1^2 x_2 f'''(x_0) + \frac{1}{24} x_1^4 f^{iv}(x_0) \right] = 0 \quad (6)$$

Solving (5), one obtains

$$\varepsilon x_1 = -\frac{f(x_0)}{f'(x_0)} \quad (7)$$

Equation (6) is solved for $\varepsilon^2 x_2$ with equation (7) substituted when necessary yielding finally

$$\varepsilon^2 x_2 = \frac{2f f' f'' - 2f'^3 - f^2 f'''}{2f'^2 f''} \mp \frac{1}{2\sqrt{3}f'^2 f''} \sqrt{12f'^6 + 3f^4 f'''^2 - 24f f'^4 f'' + 12f^2 f'^3 f'' - 8f^3 f' f'' f'' - f^4 f'' f^{iv}} \quad (8)$$

Here, for the sake of brevity, $f = f(x_0)$ is taken. The iterative scheme is then constructed by inserting the correction terms to the perturbation expansion

$$\begin{aligned} x_{n+1} = & x_n - f(x_n)/f'(x_n) \\ & + \left(2f(x_n)f'(x_n)f''(x_n) - 2f'^3(x_n) - f^2(x_n)f'''(x_n)\right)/(2f'^2(x_n)f''(x_n)) \\ & \mp \left(12f'^6(x_n) + 3f^4(x_n)f'''^2(x_n) - 24f(x_n)f'^4(x_n)f''(x_n) + 12f^2(x_n)f'^3(x_n)f'''(x_n) \right. \\ & \left. - 8f^3(x_n)f'(x_n)f''(x_n)f'''(x_n) - f^4(x_n)f''(x_n)f^{iv}(x_n)\right)^{1/2} / \left(2\sqrt{3} f'^2(x_n)f''(x_n)\right) \end{aligned} \quad (9)$$

Note that the sum of the last two terms is the second correction term. When $f'(x_n) \cong 0$, Newton-Raphson algorithm has convergence problems. Equation (9) in addition, will have convergence problems for $f''(x_n) \cong 0$.

2.2. Triple-Correction-Term Algorithm

One may now take three correction terms in the perturbation expansion as follows

$$x = x_0 + \varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3 \quad (10)$$

Inserting (10) into (1) and expanding in Taylor Series up to fourth order derivatives one has

$$\begin{aligned} f(x_0 + \varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3) \cong & f(x_0) + f'(x_0)(\varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3) + \frac{f''(x_0)}{2!}(\varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3)^2 \\ & + \frac{f'''(x_0)}{3!}(\varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3)^3 + \frac{f^{iv}(x_0)}{4!}(\varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3)^4 = 0 \end{aligned} \quad (11)$$

Collecting like powers of ε yields

$$\begin{aligned} f(x_0 + \varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3) \cong & f(x_0) + \varepsilon x_1 f'(x_0) + \varepsilon^2 \left[x_2 f'(x_0) + \frac{x_1^2}{2} f''(x_0) \right] \\ & + \varepsilon^3 \left[x_3 f'(x_0) + x_1 x_2 f''(x_0) + \frac{x_1^3}{6} f'''(x_0) \right] \\ & + \varepsilon^4 \left[x_1 x_3 f''(x_0) + \frac{x_2^2}{2} f''(x_0) + \frac{x_1^2 x_2}{2} f'''(x_0) + \frac{x_1^4}{24} f^{iv}(x_0) \right] = 0 \end{aligned} \quad (12)$$

The equations are then separated into three equations for solving the three unknowns as follows

$$f(x_0) + \varepsilon x_1 f'(x_0) = 0 \quad (13)$$

$$\varepsilon^2 \left[x_2 f'(x_0) + \frac{x_1^2}{2} f''(x_0) \right] = 0 \quad (14)$$

$$\begin{aligned} & \varepsilon^3 \left[x_3 f'(x_0) + x_1 x_2 f''(x_0) + \frac{x_1^3}{6} f'''(x_0) \right] \\ & + \varepsilon^4 \left[x_1 x_3 f''(x_0) + \frac{x_2^2}{2} f''(x_0) + \frac{x_1^2 x_2}{2} f'''(x_0) + \frac{x_1^4}{24} f^{iv}(x_0) \right] = 0 \end{aligned} \quad (15)$$

Equation (13) yields the same solution as given in equation (7). From (14) the second unknown is solved as follows

$$\varepsilon^2 x_2 = -\frac{1}{2} \frac{f^2(x_0) f''(x_0)}{f'^3(x_0)} \quad (16)$$

Using (7) and (16), finally equation (15) is solved

$$\varepsilon^3 x_3 = f^3 \frac{4f'^3 f''' - 12f'^2 f''^2 - 3ff''^3 + 6ff' f'' f''' - ff'^2 f^{iv}}{24f'^5 (f'^2 - ff'')} \quad (17)$$

where $f=f(x_0)$ for brevity. Substituting all results into the perturbation expansion yields the recursion algorithm

$$\begin{aligned} x_{n+1} = & x_n - f(x_n)/f'(x_n) - f^2(x_n)f''(x_n)/(2f'^3(x_n)) \\ & + f^3(x_n)(4f'^3(x_n)f'''(x_n) - 12f'^2(x_n)f''^2(x_n) - 3f(x_n)f''^3(x_n) \\ & + 6f(x_n)f'(x_n)f''(x_n)f'''(x_n) - f(x_n)f'^2(x_n)f^{iv}(x_n)) \\ & / (24f'^5(x_n)[f'^2(x_n) - f(x_n)f''(x_n)]) \end{aligned} \quad (18)$$

2.3. Quadruple-Correction-Term Algorithm

Four correction terms are now considered in the perturbation expansion as follows

$$x = x_0 + \varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3 + \varepsilon^4 x_4 \quad (19)$$

Inserting the expansion into equation (1) and expanding in a Taylor series up to fourth order derivative

$$\begin{aligned} f(x_0 + \varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3 + \varepsilon^4 x_4) & \cong f(x_0) + f'(x_0)(\varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3 + \varepsilon^4 x_4) \\ & + \frac{f''(x_0)}{2!} (\varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3 + \varepsilon^4 x_4)^2 + \frac{f'''(x_0)}{3!} (\varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3 + \varepsilon^4 x_4)^3 \\ & + \frac{f^{iv}(x_0)}{4!} (\varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3 + \varepsilon^4 x_4)^4 = 0 \end{aligned} \quad (20)$$

and grouping the similar order terms yields

$$\begin{aligned} & f(x_0 + \varepsilon x_1 + \varepsilon^2 x_2 + \varepsilon^3 x_3 + \varepsilon^4 x_4) \cong f(x_0) + \varepsilon x_1 f'(x_0) \\ & + \varepsilon^2 \left[x_2 f'(x_0) + \frac{x_1^2}{2} f''(x_0) \right] + \varepsilon^3 \left[x_3 f'(x_0) + x_1 x_2 f''(x_0) + \frac{x_1^3}{6} f'''(x_0) \right] \\ & + \varepsilon^4 \left[x_4 f'(x_0) + \frac{x_2^2}{2} f''(x_0) + x_1 x_3 f''(x_0) + \frac{x_1^2 x_2}{2} f'''(x_0) + \frac{x_1^4}{24} f^{iv}(x_0) \right] = 0 \end{aligned} \quad (21)$$

The above block is separated into four equations for four unknowns

$$f(x_0) + \varepsilon x_1 f'(x_0) = 0 \quad (22)$$

$$\varepsilon^2 \left[x_2 f'(x_0) + \frac{x_1^2}{2} f''(x_0) \right] = 0 \tag{23}$$

$$\varepsilon^3 \left[x_3 f'(x_0) + x_1 x_2 f''(x_0) + \frac{x_1^3}{6} f'''(x_0) \right] = 0 \tag{24}$$

$$\varepsilon^4 \left[x_4 f'(x_0) + \frac{x_2^2}{2} f''(x_0) + x_1 x_3 f''(x_0) + \frac{x_1^2 x_2}{2} f'''(x_0) + \frac{x_1^4}{24} f^{iv}(x_0) \right] = 0 \tag{25}$$

Solution of (22) is given in (7) and that of (23) in (16). Solving (24) yields

$$\varepsilon^3 x_3 = f^3(x_0) \frac{f'(x_0)f'''(x_0) - 3f''^2(x_0)}{6f'^5(x_0)} \tag{26}$$

Finally, from (25), one has

$$\varepsilon^4 x_4 = -f^4(x_0) \frac{15f'^3(x_0) - 10f'(x_0)f''(x_0)f'''(x_0) + f'^2(x_0)f^{iv}(x_0)}{24f'^7(x_0)} \tag{27}$$

Inserting all into the perturbation expansion finally yields the recursion formula

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n)f''(x_n)}{2f'^3(x_n)} + f^3(x_n) \frac{f'(x_n)f'''(x_n) - 3f''^2(x_n)}{6f'^5(x_n)} - f^4(x_n) \frac{15f'^3(x_n) - 10f'(x_n)f''(x_n)f'''(x_n) + f'^2(x_n)f^{iv}(x_n)}{24f'^7(x_n)} \tag{28}$$

Numerical comparisons of the three different schemes will be presented next.

3. NUMERICAL COMPARISONS

Numerical comparisons of different methods are given in this section. In Table 1, the iteration numbers required to find the root of $f(x) = e^{-x} - x = 0$ is tested. Newton-Raphson, and fourth order derivative schemes with double, triple, and quadruple correction terms are compared. Double-correction-term algorithm works like Newton-Raphson, but triple and quadruple-correction-term algorithms are better because they require fewer iteration to converge to the root. It is hard to decide whether triple-correction or quadruple-correction algorithm performs better with this example so another equation is selected. The equation is $f(x) = \tan(x) - \tanh(x) = 0$ and the results are presented in Table 2. From Table 2, in addition to the conclusions retrieved from Table 1, one can conclude that quadruple-correction-term algorithm performs better than the triple-correction-term algorithm.

4. CONCLUDING REMARKS

In the outlined calculations, one may take n correction terms in the perturbation expansion and m additional terms in the Taylor expansion. Obviously $m \geq n$ for all unknowns to be solved. From this paper and the previous paper [4], one may conclude that the performance becomes better as n increases with an optimum selection of $m=n$. In this paper $m=n=4$ is the best algorithm selected compared to the $m=4, n=3$ and $m=4, n=2$ algorithms.

Table 1- Roots of $f(x) = e^{-x} - x = 0$ by different methods

	Newton-Raphson	Double- Correction - Term (Eq. 9)	Triple- Correction- Term (Eq. 18)	Quadruple- Correction -Term (Eq. 28)
Initial Value	0.1	0.1	0.1	0.1
First Iteration	0.52252	0.60952	0.56717	0.56723
Second Iteration	0.56678	0.56746	0.56714	0.56714
Third Iteration	0.56714	0.56714		
Initial Value	0.2	0.2	0.2	0.2
First Iteration	0.54020	0.59318	0.56715	0.56717
Second Iteration	0.56701	0.56727	0.56714	0.56714
Third Iteration	0.56714	0.56714		
Initial Value	0.3	0.3	0.3	0.3
First Iteration	0.55322	0.58078	0.56714	0.56715
Second Iteration	0.56711	0.56718		0.56714
Third Iteration	0.56714	0.56714		
Initial Value	0.4	0.4	0.4	0.4
First Iteration	0.56184	0.57240	0.56714	0.56714
Second Iteration	0.56714	0.56715		
Third Iteration		0.56714		
Initial Value	0.5	0.5	0.5	0.5
First Iteration	0.56631	0.56797	0.56714	0.56714
Second Iteration	0.56714	0.56714		
Initial Value	0.6	0.6	0.6	0.6
First Iteration	0.56695	0.56734	0.56714	0.56714
Second Iteration	0.56714	0.56714		
Initial Value	0.7	0.7	0.7	0.7
First Iteration	0.56408	0.57021	0.56714	0.56714
Second Iteration	0.56714	0.56714		
Initial Value	0.8	0.8	0.8	0.8
First Iteration	0.55805	0.56722	0.56714	0.56714
Second Iteration	0.56713	0.56716		
Third Iteration	0.56714	0.56714		
Initial Value	0.9	0.9	0.9	0.9
First Iteration	0.54920	0.58494	0.56715	0.56714
Second Iteration	0.56708	0.56720	0.56714	
Third Iteration	0.56714	0.56714		

Table 2- Roots of $f(x) = \tan(x) - \tanh(x) = 0$ by different methods

	Newton-Raphson	Double Correction Term (Eq. 9)	Triple Correction Term (Eq. 18)	Quadruple Correction Term (Eq. 28)
Initial Value	3.6	3.6	3.6	3.6
First Iteration	4.0071	3.9662	3.9264	3.9293
Second Iteration	3.9334	3.9265	3.9266	3.9266
Third Iteration	3.9266	3.9266		
Initial Value	3.7	3.7	3.7	3.7
First Iteration	3.9695	3.8809	3.9267	3.9271
Second Iteration	3.9285	3.9245	3.9266	3.9266
Third Iteration	3.9266	3.9266		
Initial Value	3.8	3.8	3.8	3.8
First Iteration	3.9412	3.9111	3.9266	3.9266
Second Iteration	3.9268	3.9264		
Third Iteration	3.9266	3.9266		
Initial Value	3.9	3.9	3.9	3.9
First Iteration	3.9273	3.9259	3.9266	3.9266
Second Iteration	3.9266	3.9266		
Initial Value	4.0	4.0	4.0	4.0
First Iteration	3.9322	3.9213	3.9266	3.9266
Second Iteration	3.9266	3.9266		
Initial Value	4.1	4.1	4.1	4.1
First Iteration	3.9598	3.8987	3.9262	3.9266
Second Iteration	3.9277	3.9258	3.9266	
Third Iteration	3.9266	3.9266		
Initial Value	4.2	4.2	4.2	4.2
First Iteration	4.0129	3.8633	3.8633	3.9274
Second Iteration	3.9345	3.9226	3.9226	3.9266
Third Iteration	3.9267	3.9266	3.9266	
Fourth Iteration	3.9266			
Initial Value	4.3	4.3	4.3	4.3
First Iteration	4.0934	3.8279	3.6327	3.9349
Second Iteration	3.9572	3.9204	3.9266	3.9266
Third Iteration	3.9276	3.9267		
Fourth Iteration	3.9266	3.9266		

REFERENCES

1. A. H. Nayfeh, *Introduction to Perturbation Techniques*, John Wiley and Sons, New York, 1981.
2. E. J. Hinch, *Perturbation Methods*, Cambridge University Press, New York, 1991.
3. A. S. Householder, *The Numerical Treatment of a Single Nonlinear Equation*, McGraw-Hill, New York, 1970.
4. M. Pakdemirli, and H. Boyacı, Generation of root finding algorithms via perturbation theory and some formulas, *Applied Mathematics and Computation* (in press).
5. S. Abbasbandy, Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method, *Applied Mathematics and Computation* **145**, 887-893, 2003.