45

Modifying Kurchatov's method to find multiple roots

A. Cordero[♭], N. Garrido[♭], J.R. Torregrosa[♭] and P. Triguero-Navarro[♭,1]

(♭) Instituto de Matemática Multidisciplinar, Universitat Politècnica de València,
Camino de Vera, s/n, 46022-Valencia, Spain,
acordero@mat.upv.es, neugarsa@mat.upv.es, jrtorre@mat.upv.es, ptrinav@doctor.upv.es

1 Introduction

In many problems in engineering or applied mathematics, it is necessary to solve nonlinear equations $f(x) = 0$. They cannot always be solved exactly, which is why iterative methods appear to solve them. A well-known one is Newton's method, which has the following expression:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \text{ for } k = 0, 1, \ldots$$

To ensure the convergence of Newton's method, the derivative of the function evaluated in the solution must be non-zero, that is, the solution must be a simple root of $f(x) = 0$. This is not always the case. For this reason, iterative methods appear that allow us to obtain solutions with a multiplicity greater than 1.

One of them is the following modification of Newton's method, that can be find in [6], where m is the multiplicity of the solution of the equation.

$$x_{k+1} = x_k - m\frac{f(x_k)}{f'(x_k)}, \text{ for } k = 1, 2, 3 \ldots$$

In order to be able to apply this method, we must know the multiplicity of the solution in a priori.

To avoid the need to know the multiplicity in advance, iterative methods for multiple roots are designed that do not use this multiplicity in their iterative expression, see [2].

In this way, we propose the following iteratives methods based on Kurchatov's method.

To estimate the roots of $f(x) = 0$, we define the following method, denoted by KM,

$$x_{k+1} = x_k - \frac{g(x_k)}{g[2x_k - x_{k-1}, x_{k-1}]}, \qquad k = 0, 1, 2, \ldots$$

where $g(x) = \frac{f(x)}{f'(x)}$ and $g[y, z](y - z) = g(y) - g(z)$.

To calculate the expression of $g(x)$ in the previous method we use the derivative of the function to be solved. We can replace this derivative by a divided difference operator, so that to estimate the roots of $f(x) = 0$, we define the following method, denoted by KMD,

$$x_{k+1} = x_k - \frac{g(x_k)}{g[2x_k - x_{k-1}, x_{k-1}]}, \qquad k = 0, 1, 2, \ldots$$

[1]ptrinav@doctor.upv.es

where $g(x) = \frac{f(x)}{f[x+f(x),x]}$.

In this paper, we will analyse the order of convergence of the proposed methods and we will also perform numerical experiments to illustrate the behaviour of them.

2 Convergence analysis

Theorem 20. *Let $f : \mathbb{C} \longrightarrow \mathbb{C}$ be a sufficiently differentiable function in an neighbourhood of α, which we denote by $D \subset \mathbb{C}$, such that α is a multiple zero of $f(x) = 0$ with unknown multiplicity $m \in \mathbb{N}-\{1\}$. Then, taking an estimate x_0 close enough to α, the sequence of iterates $\{x_k\}$ generated by method KM converges to α with order 2.*

Proof. We first obtain the Taylor expansion of $f(x_k)$ around α where $e_k = x_k - \alpha$:

$$f(x_k) = \frac{f^{(m)}(\alpha)}{m!}\left(e_k^m + C_1 e_k^{m+1} + C_2 e_k^{m+2} + C_3 e_k^{m+3}\right) + O(e_k^{m+4}),$$

being $C_j = \frac{m!}{(m+j)!}\frac{f^{(m+j)}(\alpha)}{f^{(m)}(\alpha)}$ for $j = 1, 2, \ldots$

Calculating the derivative of the above expression we obtain

$$f'(x_k) = \frac{f^{(m)}(\alpha)}{m!}\left(m e_k^{m-1} + (m+1)C_1 e_k^m + (m+2)C_2 e_k^{m+1} + (m+3)C_3 e_k^{m+2}\right) + O(e_k^{m+3}).$$

Now. we calculate $g(x_k)$

$$g(x_k) = \frac{f(x_k)}{f'(x_k)} = \frac{1}{m}\left(e_k - \frac{1}{m}C_1 e_k^2 + \frac{(m+1)C_1^2 - 2mC_2}{m^2}e_k^3\right) + O(e_k^4).$$

In an equivalent way, we obtain the following expressions for $g(x_{k-1})$ and $g(2x_k - x_{k-1})$

$$g(x_{k_1}) = \frac{f(x_{k-1})}{f'(x_{k-1})} = \frac{1}{m}\left(e_{k-1} - \frac{1}{m}C_1 e_{k-1}^2 + \frac{(m+1)C_1^2 - 2mC_2}{m^2}e_{k-1}^3\right) + O(e_{k-1}^4),$$

$$\begin{aligned} g(2x_k - x_{k-1}) &= \frac{f(x_k)}{f'(2x_k - x_{k-1})} = \\ &= \frac{1}{m}\left(2e_k - e_{k-1} - \frac{1}{m}C_1(2e_k - e_{k-1})^2 + \frac{(m+1)C_1^2 - 2mC_2}{m^2}(2e_k - e_{k-1})^3\right) + O_4(e_k, e_{k-1}), \end{aligned}$$

with $e_{k-1} = x_{k-1} - \alpha$.

From the above relations, we obtain

$$\begin{aligned} g[2x_k - x_{k-1}, x_{k-1}] &= \frac{g\left(2x_k - x_{k-1}\right) - g\left(x_{k-1}\right)}{2(x_k - x_{k-1})} \\ &= \frac{1}{m}\left(1 - \frac{2}{m}C_1 e_k + \frac{(m+1)C_1^2 - 2mC_2}{m^2}\left(4e_k^2 - 2e_j e_{k-1} + e_{k-1}^2\right)\right) + O_3(e_k, e_{k-1}). \end{aligned}$$

Thus, applying the above relationship, the following error equation is obtained:

$$\begin{aligned} x_{k+1} - \alpha &= x_k - \alpha - \frac{g(x_k)}{g[2x_k - x_{k-1}, x_{k-1}]} \\ &= \frac{-1}{m}C_1 e_k^2 + \frac{(m+1)C_1^2 - 2mC_2}{m^2}\left(-5e_k^3 + 2e_k^2 e_{k-1} - e_k e_{k-1}^2\right) + O_4(e_k, e_{k-1}). \end{aligned}$$

We have some different possibilities for the behaviour of e_{k+1} respect to e_k and e_{k-1}.

By the expression, we only are going to take into account if the behaviour is like e_k^2 or $e_k e_{k-1}^2$, because e_k^3 and $e_k^2 e_{k-1}$ converge faster to 0 than e_k^2.

Then,

$$e_{k+1} \sim \frac{-1}{m} C_1 e_k^2 - \frac{(m+1)C_1^2 - 2mC_2}{m^2} e_k e_{k-1}^2.$$

- If $e_{k+1} \sim e_k^2$, then the order of convergence is 2.
- Now, we suppose that $e_{k+1} \sim e_k e_{k-1}^2$. We assume that the method has R-order p, that means,

  $$e_{k+1} \sim e_k^p.$$

  In the same way, $e_k \sim e_{k-1}^p$. From the above relations, we get

  $$e_{k+1} \sim e_{k-1}^{p^2}.$$

  Then, the error equation is

  $$e_{k+1} \sim e_k e_{k-1}^2 \sim e_{k-1}^{p+2}.$$

  By equating the exponents of e_{k-1} of the above relations, we obtain the following polynomial $p^2 - p - 2 = 0$, whose only positive root is $p = 2$, then the order of convergence of the method is 2.

■

Theorem 21. *Let $f : \mathbb{C} \longrightarrow \mathbb{C}$ be a sufficiently differentiable function in an neighbourhood of α, which we denote by $D \subset \mathbb{C}$, such that α is a multiple zero of $f(x) = 0$ with unknown multiplicity $m \in \mathbb{N} - \{1\}$. Then, taking an estimate x_0 close enough to α, the sequence of iterates $\{x_k\}$ generated by method KMD converges to α with order 2.*

Proof. We first obtain the Taylor expansion of $f(x_k)$ around α where $e_k = x_k - \alpha$:

$$f(x_k) = \frac{f^{(m)}(\alpha)}{m!}\left(e_k^m + C_1 e_k^{m+1}\right) + O(e_k^{m+2}).$$

being $C_j = \frac{m!}{(m+j)!}\frac{f^{(m+j)}(\alpha)}{f^{(m)}(\alpha)}$ for $j = 1, 2, \ldots$

In the same way,

$$f(x_k + f(x_k)) = \frac{f^{(m)}(\alpha)}{m!}\left((e_k + f(x_k))^m + C_1 (e_k + f(x_k))^{m+1}\right) + O(e_k^{m+2}).$$

Then,

$$f(x_k + f(x_k)) - f(x_k) = \frac{f^{(m)}(\alpha)}{m!}\left((e_k + f(x_k))^m - e_k^m + C_1\left((e_k + f(x_k))^{m+1} - e_k^{m+1}\right)\right) + O(e_k^{m+2}).$$

Using Newton's binomial and the Taylor expansion of $f(x_k)$ around α we obtain that

$$\frac{f(x_k + f(x_k)) - f(x_k)}{x_k + f(x_k) - x_k} = \frac{f^{(m)}(\alpha)}{m!}\left(m e_k^{m-1} + (m+1)C_1 e_k^m\right) + O(e_k^{m+1}).$$

We then calculate $g(x_k)$ from the above expressions.

$$g(x_k) = \frac{f(x_k)}{f[x_k + f(x_k), x_k]} = \frac{e_k^m + C_1 e_k^{m+1} + O(e_k^{m+2})}{m e_k^{m-1} + (m+1)C_1 e_k^m + O(e_k^{m+1})}$$

$$= \frac{1}{m}\left(e_k - \frac{1}{m}C_1 e_k^2\right) + O(e_k^3).$$

In an equivalent way, we obtain the following expressions for $g(x_{k-1})$ and $g(2x_k - x_{k-1})$

$$g(x_{k-1}) = \frac{1}{m}\left(e_{k-1} - \frac{1}{m}C_1 e_{k-1}^2\right) + O(e_{k-1}^3),$$

$$g(2x_k - x_{k-1}) = \frac{1}{m}\left(2e_k - e_{k-1} - \frac{1}{m}C_1(2e_k - e_{k-1})^2\right) + O_3(e_k, e_{k-1}),$$

with $e_{k-1} = x_{k-1} - \alpha$.

Then, appyling the above relations, we obtain that

$$\begin{aligned} g[2x_k - x_{k-1}, x_{k-1}] &= \frac{g(2x_k - x_{k-1}) - g(x_{k-1})}{2(x_k - x_{k-1})} \\ &= \frac{1}{m}\left(1 - \frac{2}{m}C_1 e_k\right) + O_2(e_k, e_{k-1}). \end{aligned}$$

Thus, the following error equation is obtained

$$\begin{aligned} x_{k+1} - \alpha &= x_k - \alpha - \frac{g(x_k)}{g[2x_k - x_{k-1}, x_{k-1}]} \\ &= -\frac{1}{m}C_1 e_k^2 + e_k O_2(e_k, e_{k-1}) + O(e_k^3). \end{aligned}$$

We have some different possibilities for the behaviour of e_{k+1} respect to e_k and e_{k-1}.

By the expression, we only are going to take into account if the behaviour is like e_k^2 or $e_k e_{k-1}^2$, because e_k^3 and $e_k^2 e_{k-1}$ converge faster to 0 than e_k^2.

Then

- If $e_{k+1} \sim e_k^2$, then the order of convergence is 2.
- If we assume that $e_{k+1} \sim e_k e_{k-1}^2$. Then, we assume that the method has R-order p, that means,

  $$e_{k+1} \sim D_{k,p} e_k^p.$$

  At the same time, $e_k \sim e_{k-1}^p$, then we obtain that

  $$e_{k+1} \sim e_{k-1}^{p^2}.$$

  From the error equation and the last relation, we obtain that

  $$e_{k+1} \sim e_k e_{k-1}^2 \sim e_{k-1}^{p+2}.$$

  By equating the exponents of e_{k-1} of the last two equation, we obtain the following polynomial $p^2 - p - 2 = 0$, whose only positive root is $p = 2$, then the order of convergence of the method is 2.

■

3 Numerical experiments

Now, we perform a numerical experiment to see the behaviour of the two proposed iterative methods. For the computational calculations, we use Matlab R2020b with an arithmetic precision of 500 digits iterating from an initial estimate x_0 until it is verified that the absolute value of the function evaluated in the iteration is less than 10^{-50}, that is,

$$|f(x_k)| < 10^{-50}.$$

The numerical results we are going to compare the methods in the different examples are:

- the approximation obtained,
- the norm of the equation evaluated in that approximation,
- the norm of the distance between the last two approximations,
- the number of iterations necessary to satisfy the required tolerance,
- the computational time and the approximate computational convergence order (ACOC), defined by Cordero and Torregrosa in [3], which has the following expression

$$p \approx ACOC = \frac{\ln(|x_{k+1} - x_k|/|x_k - x_{k-1}|)}{\ln(|x_k - x_{k-1}|/|x_{k-1} - x_{k-2}|)}.$$

The equation we try to solve is $f(x) = (x^2 - 1)^3$, which has two roots with multiplicity 3.

Table 1: Results for $(x^2 - 1)^3 = 0$.

	x_0	x_{-1}	$\|x_{k+1} - x_k\|$	$\|g(x_{k+1})\|$	Iter	ACOC
KM	0.5	0.1	3.3307e-16	0	7	2.0058
KMD	0.5	0.1	9.7478e-14	0	9	1.7006

The results obtained for each of the methods for the function to be solved are shown in Table 1. We can see from the Tables that in all cases the ACOC is close to the theoretical convergence order shown above. It can be seen that the best results for this numerical experiment are obtained with the KM method. Both methods give good approximations to the solution, although the *KM* method performs less iterations to verify the stopping criterion.

4 Conclusions

In this work, we have studied two iterative methods for multiple roots with memory, obtaining that the order of convergence of them is 2. These iterative methods do not use the multiplicity of the root in their iterative expression, so it is not necessary to know this multiplicity before applying the iterative method. In the numerical experiments, we have verified the theoretical results concerning the order of convergence of the methods.

Financial disclosure

This research was partially supported by Spanish Ministerio de Ciencia, Innovación y Universidades PGC2018-095896-B-C22, by the internal research project ADMIREN of Universidad Internacional de La Rioja (UNIR) and was partially supported by Universitat Politécnica de Valéncia Contrato Predoctoral PAID-01-20-17 (UPV).

References

[1] F.I. Chicharro, R. Contreras & N. Garrido. A Family of Multiple-Root Finding Mathematics, 8, 2020.

[2] A. Cordero, B. Neta & J.R. Torregrosa. Memorizing Schröder's Method as an Efficient Strategy for Estimating Roots of Unknown Multiplicity. Mathematics, 9, 2021.

[3] A. Cordero, A. & J.R. Torregrosa. Variants of Newton's method using fifth-order quadrature formulas. Appl. Math. Comput., 190. 2007.

[4] B. Neta, C. Chun & M. Scott. On the development of iterative methods for multiple roots. Appl. Math. Comput., 224. 2013.

[5] S.M. Shakhno. On a Kurchatov's method of linear interpolation for solving nonlinear equations. PAMM Proc. Appl. Math. Mech. 4, 650–651, 2004.

[6] E. Schröder, Über unendlich viele Algorithmen zur Auflösung der Gleichungen, Math. Ann. 2, 317–365, 1870.

[7] F. Zafar, A. Cordero, J.R. Torregrosa & M. Penkova. A family of optimal fourth-order methods for multiple roots of nonlinear equations. Math Meth Appl Sci, 43. 2020.begindocument