Suitable approximations for the self-accelerating parameters in iterative methods with memory

F. I. Chicharro ♭[1], N. Garrido ♭, A. Cordero ♮ and J.R. Torregrosa ♮

(♭) Escuela Superior de Ingeniería y Tecnología, Universidad Internacional de La Rioja
Av. La Paz 137, 26006-Logroño, Spain,
(♮) Instituto de Matemática Multidisciplinar, Universitat Politècnica de València,
Camí de Vera s/n, 46022-València, Spain.

1 Introduction

Solving the nonlinear equation $f(x) = 0$, $f : D \subseteq \mathbb{R} \to \mathbb{R}$ is a common problem in several areas of Science and Engineering. Since exact solutions of the nonlinear equation are hardly available, scientists best rely on numerical solutions, such as those given by iterative methods.

Iterative procedures for finding the root α of f can be classified according to different criteria: the presence or absence of derivatives in the iterative expression, single- or multiple-root finding methods, or the number of previous iterations to obtain the current one, among others. For instance, we can find iterative methods without memory for finding single roots with derivatives [1] and derivative-free [2], or multiple-root finding with derivatives [3] and derivative-free [4]. In the methods with memory counterpart for single roots, we can also find [5] with derivatives, or [6] without derivatives.

Focusing on the last type of schemes, they are known as iterative methods with memory. These methods can be expressed by

$$x_{k+1} = M(x_k, x_{k-1}, \ldots, x_{k-p}),$$

where M describes the iterative method. Therefore, p previous values of the sequence are necessary to obtain the following one. The major advantage of this sort of methods is to enhance the order of convergence of the original method without introducing new evaluations of the function f [7].

One technique to obtain iterative schemes with memory starts with an iterative method without memory that includes a parameter. Depending on the error equation of the method, the parameter can be replaced by an expression that includes the previous iterations. A basic example of this technique can be found at [8]. Starting from Traub's method [9], the parameter δ is included in the denominator of the first step. Its order of convergence is

$$e_{k+1} = 2c_2(c_2 + \delta)e_k^3 + \mathcal{O}(e_k^4),$$

[1] e-mail: francisco.chicharro@unir.net

where $e_k = x_k - \alpha$ and $c_j = \frac{f^{(j)}(\alpha)}{j!f'(\alpha)}$, $j \geq 2$. Replacing δ by a suitable approximation that includes the previous iterates, the method becomes into an iterative procedure with memory.

The strategy to obtain these expressions is essential. In this work we propose suitable approximations that are present in the literature for different iterative methods with memory. On the one hand, the polynomials approximation, such as Newton's interpolation polynomial. On the other hand, the non-polynomials approximation, such as Padé's approximants. Finally, we discuss on the most suitable choice in terms of convergence and stability.

2 Inclusion of memory

Generally, the error equation is of the form

$$e_{k+1} = K(k_0 + k_1\delta + k_2c_j)e_k^p + \mathcal{O}(e_k^{p+1}),$$

where K, k_0, k_1 and k_2 are constants. In order to cancel the error of order p, the parameter must satisfy

$$\delta = -\frac{k_0 + k_2c_j}{k_1},$$

but c_j includes $f^{(j)}(\alpha)$ and α is unknown. Therefore, an approximation of $f^{(j)}(\alpha)$ is performed. There are two main trends in this situation: Newton's interpolation polynomial and Padé's approximants.

2.1 Newton's interpolation polynomial

Newton's interpolation polynomial has been applied in [10]. In this case, the iterative expression is

$$\begin{array}{rcl} y_k & = & x_k - \frac{f(x_k)}{f[x_k,w_k]}, \\ x_{k+1} & = & y_k - \left(1 + \frac{f(y_k)}{f(x_k)} + \lambda\frac{f^2(y_k)}{f^2(x_k)}\right)\frac{f(y_k)}{f[y_k,w_k]}, \quad k = 1, 2, \ldots, \end{array} \tag{1}$$

where $w_k = x_k + \delta f(x_k)$, being δ and λ free parameters. The error equation of (1) is

$$e_{k+1} = -K\frac{c_2}{2}\left(1 + \delta f'(\alpha)\right)^2 e_k^4 + \mathcal{O}(e_k^5).$$

Therefore, for increasing the order of convergence, we need to approximate $\delta \approx -\frac{1}{f'(\alpha)}$. Using Newton's polynomial of first degree $N(t) = f(x_k) + f[x_k, x_{k-1}](t - x_k)$, we have

$$f'(\alpha) \approx N'(x_k) = f[x_k, x_{k-1}],$$

and the R-order of the method [11] is 4.45.

2.2 Padé's approximant

Padé's approximant has been applied in [12]. The starting iterative expression is

$$\begin{array}{rcl} y_k & = & x_k - \frac{f(x_k)}{f'(x_k)+\delta f(x_k)}, \\ x_{k+1} & = & y_k - \frac{1+\beta u_k+u_k^2}{1+(\beta-2)u_k+P_2(\beta)u_k^2}\frac{f(y_k)}{f'(x_k)+2\delta f(x_k)}, \quad k = 1, 2, ldots, \end{array} \tag{2}$$

where $u_k = f(y_k)/f(x_k)$, $P_2(\beta) = 0.17\beta^2 - 0.8075\beta + 2.9166$, and both δ and λ are free parameters. The error equation of (2) is

$$e_{k+1} = K(\delta + c_2)e_k^4 + \mathcal{O}(e_k^5).$$

Again, for increasing the order of convergence, $\delta \approx c_2 = -\frac{f''(\alpha)}{2f'(\alpha)}$. Using Padé's approximant

$$m(t) = \frac{a_1 + a_2(t - x_k)}{1 + a_3(t - x_k)},$$

and constraining

$$\begin{cases} m(x_k) &= f(x_k), \\ m(x_{k-1}) &= f(x_{k-1}), \\ m'(x_k) &= f(x_k), \end{cases}$$

the approximation can be performed as

$$\delta \approx -\frac{f''(\alpha)}{2f'(\alpha)} = \frac{f(x_k) - f(x_{k-1}) + f'(x_k)(x_{k-1} - x_k)}{(f(x_k) - f(x_{k-1}))(x_k - x_{k-1})}.$$

In this case, the *R*-order is 4.24.

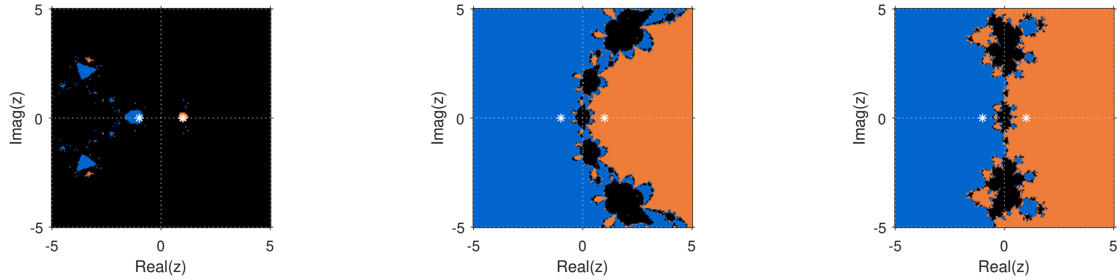
3 Effects on dynamical behavior

The dynamical behavior of the iterative methods highlights the stability of the methods in the sense of the amount of initial estimations that converge to the desired root. Many papers collect the basics of the dynamical analysis [13, 14] in general, and [15, 16] in the particular case of iterative methods with memory.

For the sake of comparison, we are showing different dynamical planes of the original methods and after the inclusion of memory. The dynamical planes have been generated following the guidelines of [17]. The specific values are

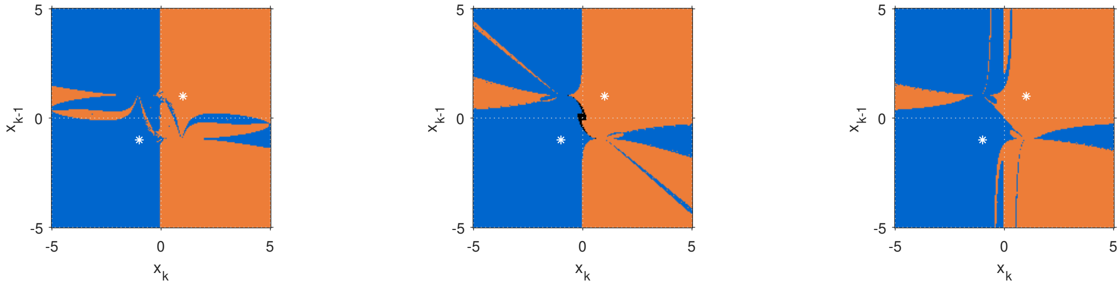
- 200 points of grid in the square $\Re\{z\} \in [-5, 5], \Im\{z\} \in [-5, 5]$ for the method without memory, and $x_k \in [-5, 5], x_{k-1} \in [-5, 5]$.
- 50 maximum of iterations.
- The method interprets convergence when the difference between the iterate and α is lower than 10^{-6}.
- The nonlinear function is $f(z) = z^2 - 1, z \in \hat{\mathbb{C}}$, for the methods without memory, and $f(x) = x^2 - 1, x \in \mathbb{R}$, for methods with memory.
- The $\{\text{blue}, \text{orange}, \text{black}\}$ colors represent the convergence to $\{-1, 1, \text{other}\}$.

Figure 1 represents the dynamical planes of (1) without memory, using $\delta = \frac{1}{10}$ and $\lambda \in \{-550, -\frac{9}{2}, 10\}$, while Figure 2 is the version of (1) with memory, for the same set of values of λ.



(a) $\lambda = -550$ (b) $\lambda = -\frac{9}{2}$ (c) $\lambda = 10$

Figure 1: Dynamical planes of method (1) without memory, $\delta = \frac{1}{10}$.

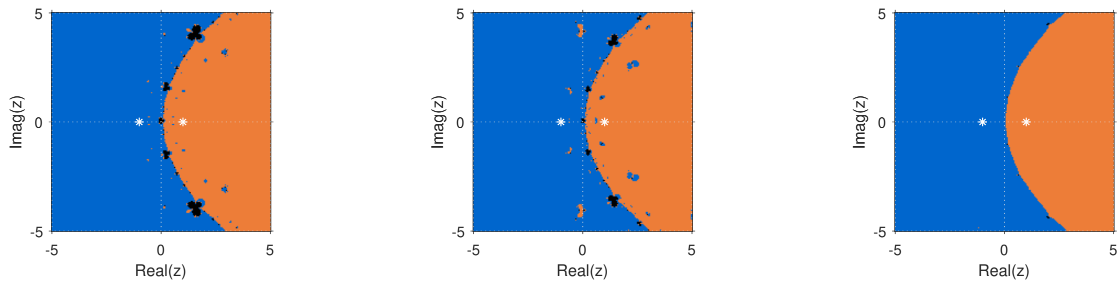


(a) $\lambda = -550$ (b) $\lambda = -\frac{9}{2}$ (c) $\lambda = 10$

Figure 2: Dynamical planes of method (1) with memory.

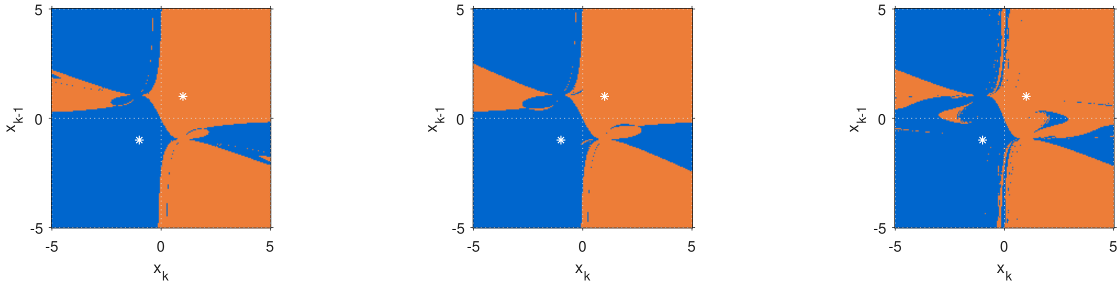
There is presence of black regions in Figure 1, that show the convergence to a point that does not match with the roots. With the inclusion of memory, we can see in Figure 2 that regions in black are trifling, and almost every initial guess converges to the roots of the polynomial.

Figure 3 represents the dynamical planes of (2) without memory, using $\delta = \frac{1}{10}$ and $\lambda \in \{4, \frac{32}{5}, 67\}$, while Figure 4 is the version of (2) with memory, for the same set of values of λ.



(a) $\lambda = 4$ (b) $\lambda = \frac{32}{5}$ (c) $\lambda = 67$

Figure 3: Dynamical planes of method (2) without memory, $\delta = \frac{1}{10}$.



(a) $\lambda = 4$ (b) $\lambda = \frac{32}{5}$ (c) $\lambda = 67$

Figure 4: Dynamical planes of method (2) with memory.

In Figure 3 we can observer that the higher value of λ, the wider stability. The inclusion of memory removes the presence of black regions, as Figure 4 evidences.

4 Conclusions

The inclusion of memory in iterative methods enhances the order of convergence of the original method with no additional cost in function evaluations. Two techniques have presented: Newton's interpolation polynomial and Padé's approximant. In the first case, the original method has significant basins of attraction that do not own to the roots. The inclusion of memory has softened these basins, and almost every initial iterate converges to the expected solution. In the second case, a similar behavior can be observed.

Acknowledgements

This research was supported by PGC2018-095896-B-C22 (MCIU/AEI/FEDER, UE).

References

[1] Amorós, C., Argyros, I.K., González, R., Magreñán, Á.A., Orcos, L., Sarría, Í., Study of a high order family: local convergence and dynamics, *Mathematics* 7(225), 2019.

[2] Chicharro, F.I., Cordero, A., Torregrosa, J.R., Dynamics and fractal dimension of Steffensen-type methods, *Algorithms* 8(2):271–279, 2015.

[3] Akram, S., Zafar, F., Yasmin, N., An optimal eighth-order family of iterative methods for multiple roots, *Mathematics* 7(672), 2019.

[4] Yun, B.I., A derivative free iterative method for finding multiple roots of nonlinear equations, *Applied Mathematics Letters* 22(12): 1859–1863, 2009.

[5] Lee, M.Y., Kim, Y.I., Magreñán, Á.A., On the dynamics of tri-parametric family of optimal fourth-order multiple-zero finders with a weight function of the principal mth root of a function-to-function ratio, *Applied Mathematics and Computation* 315: 564–590, 2017.

[6] Džunić, J., Petković, M.S., On generalized biparametric multipoint root finding methods with memory, *Journal of Computational and Applied Mathematics*, 255: 362–375, 2014.

[7] Petković, M.S., Neta, B., Petković, L.D., Džunić, J., Multipoint Methods for Solving Nonlinear Equations. Amsterdam, Academic Press, 2013.

[8] Chicharro, F.I., Cordero, A., Garrido, N., Torregrosa, J.R., Impact on stability by the use of memory in Traub-type schemes, *Mathematics* 8(274), 2020.

[9] Traub, J.F., Iterative Methods for the Solution of Equations. New York, Prentice-Hall, 1964.

[10] Chicharro, F.I., Cordero, A., Torregrosa, J.R., Dynamics of iterative families with memory based on weight functions procedure, *Journal of Computational and Applied Mathematics* 354:286–298, 2019.

[11] Ortega, J.M., Rheinboldt, W.G., Iterative Solutions of Nonlinear Equations in Several Variables. New York, Academic Press, 1970.

[12] Chicharro, F.I., Cordero, A., Garrido, N., Torregrosa, J.R., On the choice of the best members of the Kim family and the improvement of its convergence, *Mathematical Methods in the Applied Sciences* 43(14):8051–8066, 2020.

[13] Blanchard, P., The dynamics of Newton's method, *Proceedings of the Symposia in Applied Mathematics* 49:139–154, 1994.

[14] Revaney, R.L., An Introduction to Chaotic Dynamical Systems. Colorado, Addison-Wesley, 1989.

[15] Campos, B., Cordero, A., Torregrosa, J.R., Vindel, P., A multidimensional dynamical approach to iterative methods with memory, *Applied Mathematics and Computation* 271:701–715, 2015.

[16] Campos, B., Cordero, A., Torregrosa, J.R., Vindel, P., Stability of King's family of iterative methods with memory, *Journal of Computational and Applied Mathematics* 318:504–514, 2017.

[17] Chicharro, F.I., Cordero, A., Torregrosa, J.R., Drawing dynamical and parameters planes of iterative families and methods, *The Scientific World Journal* ID 780153, 2013.