Acceleration of the Robust Newton Method by the use of the S-iteration

Krzysztof Gdawiec[1(✉)][0000−0001−9434−9307], Wiesław Kotarski[2][0000−0002−5920−5161], and Agnieszka Lisowska[1][0000−0003−0492−772X]

[1] Institute of Computer Science, University of Silesia, Będzińska 39, 41–200 Sosnowiec, Poland
{krzysztof.gdawiec,agnieszka.lisowska}@us.edu.pl
[2] Independent Researcher
kotarski@ux2.math.us.edu.pl

Abstract. In this paper, we propose an improvement of the Robust Newton's Method (RNM). The RNM is a generalisation of the known Newton's root finding method restricted to polynomials. Unfortunately, the RNM is slow. Thus, in this paper, we propose the acceleration of this method by replacing the standard Picard iteration in the RNM by the S-iteration. This leads to an essential acceleration of the modified method. We present the advantages of the proposed algorithm over the RNM using polynomiagraphs and some numerical measures. Moreover, we present its possible application to the generation of artistic patterns.

Keywords: robust Newton · S-iteration · polynomiography.

1 Introduction

Newton's method is used for finding solutions of nonlinear equations [3] and in optimisation techniques [11], but it is not robust. An improvement of this method with respect to its robustness can be achieved, for instance, by the use of a damping factor in Newton's algorithm. This method called the Damped Newton Method, decreases or even eliminates the fractal boundaries between basins [4].

Recently, the Robust Newton Method (RNM) was proposed [10], which radically smooths the boundaries between basins of attraction. That result was obtained by precise controlling of the step's length in Newton's method. Unfortunately, the RNM usually needs a huge number of iterations [10]. Thus, even for not so accurate computations for some points, we need to perform a large number of iterations. This is a drawback of the RNM.

In this paper, we propose a modification of the RNM by introducing the S-iteration instead of the Picard one. This modification allows decreasing the average number of iterations needed to find the solution. We will also analyse, numerically, different aspects of the proposed modification. Moreover, by using various sequences of parameters in the S-iteration, we will generate very complex and intriguing patterns from the dynamics of the modified method.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_28

The paper is organised as follows. In Section 2 the RNM is described. In Section 3 the proposed modification of the RNM is presented. Section 4 presents the experimental results. And Section 5 concludes this paper.

2 The Robust Newton's Method

Newton's method has one drawback – the lack of definition at critical points. To overcome this, the Robust Newton's Method (RNM) was proposed [10]. This method is defined by applying the Geometric Modulus Principle [9] and the Modulus Reduction Theorem [10] to the Newton's method. The RNM guarantees the reduction of the polynomial's modulus in successive iterations. Additionally, the RNM has a few differences in comparison to the Newton's method, i.e. it converges globally, and it finds both the roots and the critical points of polynomials, unlike the Newton's method. Let us follow the definitions from [10].

Let us consider a complex polynomial p of degree $n \in \mathbb{N}$. Assume that $p(z) \neq 0$, $z \in \mathbb{C}$ and let us define [10]: $k := k(z) = \min\{j \geq 1 : p^{(j)}(z) \neq 0\}$, $A(z) = \max\left\{\frac{|p^{(j)}(z)|}{j!} : j = 0, \ldots, n\right\}$, and $u_k := u_k(z) = \frac{1}{k!}p(z)\overline{p^{(k)}(z)}$.

Moreover, let us define $\gamma_k = 2 \cdot Re(u_k^{k-1})$, $\delta_k = -2 \cdot Im(u_k^{k-1})$, and $c_k = \max\{|\gamma_k|, |\delta_k|\}$. Additionally, let θ_k be the angle given by the following formula:

$$\theta_k = \begin{cases} 0, & \text{if } c_k = |\gamma_k| \wedge \gamma_k < 0, \\ \pi/k, & \text{if } c_k = |\gamma_k| \wedge \gamma_k > 0, \\ \pi/(2k), & \text{if } c_k = |\delta_k| \wedge \delta_k < 0, \\ 3\pi/(2k), & \text{if } c_k = |\delta_k| \wedge \delta_k > 0. \end{cases} \tag{1}$$

Then, the RNM for the starting point $z_0 \in \mathbb{C}$ is defined as

$$z_{i+1} = N_p(z_i) := z_i + \frac{C_k(z_i)}{3}\frac{u_k}{|u_k|}e^{\mathbf{i}\theta_k}, \quad i = 0, 1, 2, \ldots, \tag{2}$$

where $\mathbf{i}$ denotes the imaginary unit, i.e., $\mathbf{i} = \sqrt{-1}$, and $C_k(z_i) = \frac{c_k|u_k|^{2-k}}{6A^2(z_i)}$. The term $(u_k/|u_k|)e^{\mathbf{i}\theta_k}$ is called the normalised robust Newton direction at z_i and $C_k(z_i)/3$ is called the step-size [10].

The stopping criterion is given by the following condition: $|p(z_i)| < \varepsilon \ \vee \ |p'(z_i)| < \varepsilon$, where $\varepsilon > 0$ is the accuracy. This is a different criterion than in the classical method since the RNM finds both the roots and the critical points.

3 The Robust Newton's Method with the S-iteration

The RNM presented in the previous section uses the Picard iteration for finding fixed points (see eq. (2)). However, we can use any method from the fixed point theory. One of them is the S-iteration [1] defined in the following way:

$$\begin{cases} z_{i+1} = (1 - \alpha_i)T(z_i) + \alpha_i T(v_i), \\ v_i = (1 - \beta_i)z_i + \beta_i T(z_i), \quad i = 0, 1, 2, \ldots, \end{cases} \tag{3}$$

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_28

where $\alpha_i, \beta_i \in [0,1]$ and T is a mapping. Depending on the type of the mapping and space, the conditions on the sequences α_i and β_i may be expanded to assure the convergence to a fixed point. Let us note that when $\alpha_i = 0$ or $\beta_i = 0$ for all i, the S-iteration reduces to the Picard iteration.

Now, we combine the RNM with the S-iteration, obtaining a new method, called shortly as S-RNM. This combination is done by using the N_p as T in (3) and extending the $[0,1]$ interval for the parameters to $\mathbb{R}$, obtaining:

$$\begin{cases} z_{i+1} = (1-\alpha_i)N_p(z_i) + \alpha_i N_p(v_i), \\ v_i = (1-\beta_k)z_i + \beta_i N_p(z_i), \quad i = 0,1,2,\ldots, \end{cases} \tag{4}$$

where $\alpha_i, \beta_i \in \mathbb{R}$.

Let us see the first step of (4) after including the formula for N_p:

$$v_i = (1-\beta_i)z_i + \beta_i \left(z_i + \frac{C_k(z_i)}{3} \frac{u_k}{|u_k|} e^{\mathbf{i}\theta_k} \right) = z_i + \beta_i \frac{C_k(z_i)}{3} \frac{u_k}{|u_k|} e^{\mathbf{i}\theta_k}. \tag{5}$$

We can see that if we join β_i with the term representing the step size, i.e., $C_k(z_i)/3$, then we can change the step size in the original RNM. We can look at this step as a predictor step. In the second step of the S-iteration, we combine the values of the original point and the predictor one to obtain the final point in a given iteration. The second step can be treated as a corrector step. Therefore, the S-RNM uses the predictor–corrector strategy – a well-known strategy in numerical methods [11].

4 Experiments

In this section, we present two categories of numerical results: polynomiographs [8] of two types (basins of attraction and polynomiographs showing the speed of convergence and dynamics) and the plots presenting various numerical measures: average number of iterations (ANI) [2], convergence area index (CAI) [2] and polynomiograph's generation time [5]. The experiments were performed for $\alpha_i = const = \alpha$ and $\beta_i = const = \beta$.

We have performed the experiments for different polynomials, but due to the lack of space, we present the complete results only for the polynomial $p_3(z) = z^3 - 1$. This polynomial has three roots: $-\frac{1}{2} - \frac{\sqrt{3}}{2}\mathbf{i}$, $-\frac{1}{2} + \frac{\sqrt{3}}{2}\mathbf{i}$, 1, and only one critical point, namely 0.

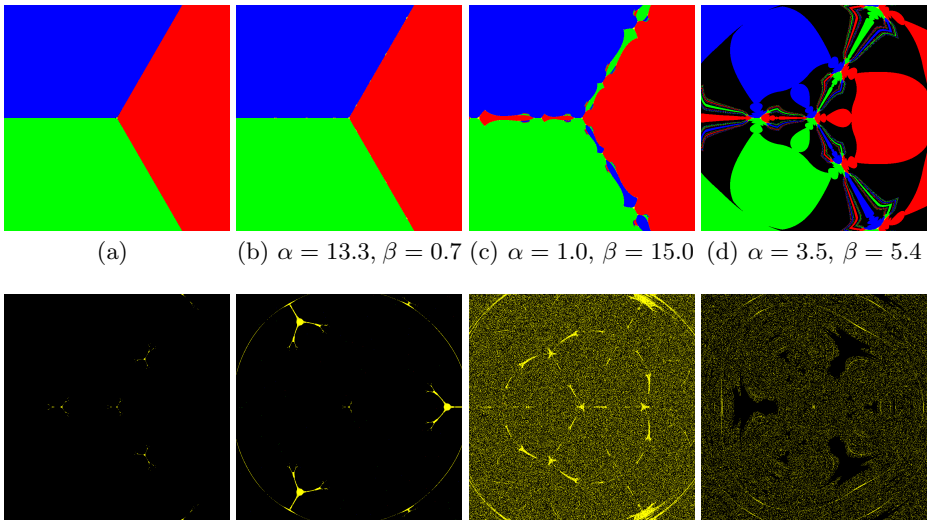
To generate the polynomiographs we used the following parameters: area $[-3,3]^2$, the maximal number of iterations equals to 250, accuracy $\varepsilon = 0.001$, and image resolution 800×800 pixels. In the basins of attraction, we used red, green and blue colours for the roots, yellow one for the critical points, and black for non-convergent points. The values of α and β, used in the S-iteration, were taken from $[0,20]$ with the step equal to 0.1, which gives 40 401 polynomiographs.

The experiments were performed on the computer with: Intel i5-9600K (@ 3.70 GHz) processor, 32 GB DDR4 RAM, NVIDIA GeForce GTX 1660 Ti with 6

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_28

GB GDDR6 SDRAM and Windows 10 (64-bit). The software was implemented in C++, OpenGL and GLSL. The computations were performed on the graphics card with GLSL shaders.

We analyse, first, the basins of attraction of the proposed method for p_3, presented in Fig. 1 for the S-RNM. From these images one can observe that when we increase the values of α or/and β, the borders of the basins are changing. Indeed, a third basin appears between the two ones (Fig. 1(c)). Going further causes that more points are not converging (Fig. 1(d)). This tendency continues up to nearly all non-convergent points. Only some convergence to critical points appears (yellow dots, Figs. 1(e)–(f)). Finally, we can observe something like a chaos with the play of critical points (Figs. 1(g)–(h)). When we look at Fig. 1(b) and Fig. 1(a) (the Picard iteration case), we can observe that the basins of attraction have similar shapes.



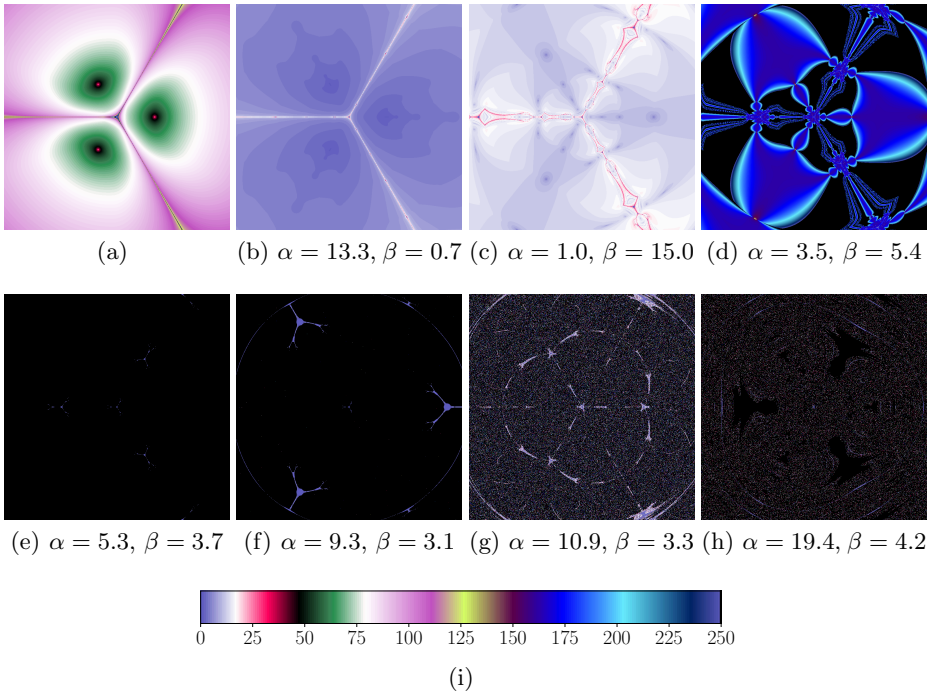
(a) (b) $\alpha = 13.3$, $\beta = 0.7$ (c) $\alpha = 1.0$, $\beta = 15.0$ (d) $\alpha = 3.5$, $\beta = 5.4$

(e) $\alpha = 5.3$, $\beta = 3.7$ (f) $\alpha = 9.3$, $\beta = 3.1$ (g) $\alpha = 10.9$, $\beta = 3.3$ (h) $\alpha = 19.4$, $\beta = 4.2$

Fig. 1. The examples of basins of attraction for $p_3(z) = z^3 - 1$ for (a) Picard iteration, (b)–(h) S-iteration with various values of α and β.

Then, we analyse the examples of dynamics polynomiographs of p_3, presented in Fig. 2, for the S-RNM. From these images, we can observe that by increasing the values of α or/and β, the dynamics changes similarly as in the case of the basins of attraction but in a more subtle way. However, the same idea remains – these changes go to the vanishing of convergent points and then to making chaos with the critical points. By comparing the polynomiographs from Figs. 1(b)–(c) and Fig. 1(a) (the Picard case) we see that the use of the S-iteration has lowered the number of iterations needed to find the solution.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_28



(a) (b) $\alpha = 13.3, \beta = 0.7$ (c) $\alpha = 1.0, \beta = 15.0$ (d) $\alpha = 3.5, \beta = 5.4$

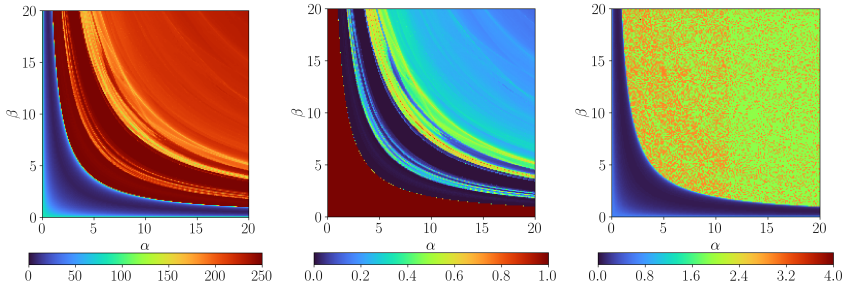
(e) $\alpha = 5.3, \beta = 3.7$ (f) $\alpha = 9.3, \beta = 3.1$ (g) $\alpha = 10.9, \beta = 3.3$ (h) $\alpha = 19.4, \beta = 4.2$

(i)

Fig. 2. The examples of dynamics polynomiographs for $p_3(z) = z^3 - 1$ for (a) Picard iteration, (b)–(h) S-iteration with various values of α and β. (i) the colour map.

Next, in Fig. 3 the heatmaps on (α, β)-plane for p_3 are shown, presenting the values of ANI, CAI and time generation, coded in colour. From all these plots one can see that the results are nearly symmetrical along $\beta = \alpha$ line. This tendency is also present in the resulting polynomiographs. In these plots it is possible to point out the values of the parameters α and β for which the S-RNM behaves well, i.e. the method is convergent to the roots or the critical point and has low ANI and time generation parameters (blue areas in Fig. 3(a) and (c), respectively) and CAI parameter is close to 1.0 (the dark red area in Fig. 3(b)). Outside these areas, the algorithm needs much more iterations to find the solution or might be non-convergent. In the mentioned areas there are located α and β parameters for which the S-RNM is essentially better than the RNM. The following values were obtained for the S-RNM in the case of $z^3 - 1$: simultaneously minimal ANI = 5.18 and time = 0.047s with CAI = 1.0, for $\alpha = 13.3$ and $\beta = 0.7$ (for which the basins of attraction and dynamics polynomiograph are presented in Figs. 1(b) and 2(b), respectively), whereas for the RNM: ANI = 84.89, CAI = 0.99, and time = 0.667s (the basins of attraction and dynamics polynomiograph are presented in Fig. 1(a) and 2(a)). It means that in this case, the S-RNM needs nearly 17 times iterations less in comparison to the RNM to achieve the same goal in considerably lower computation time.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_28



(a) ANI (b) CAI (c) time

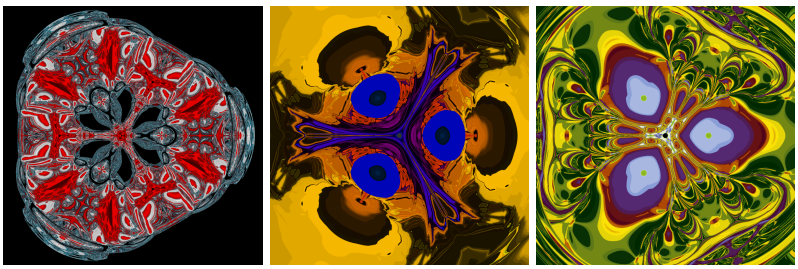
Fig. 3. The plots of: (a) ANI, (b) CAI, and (c) time of generation for $p_3(z) = z^3 - 1$.

Dynamics of a discrete dynamical system can give rise to fascinating patterns, see for instance [5,7,12]. The introduction of the S-iteration in the RNM can give birth to very complicated patterns of possible artistic applications. In Fig. 4 we present three patterns obtained with the S-RNM for p_3. The other parameters used to generate these patterns are the following:

(a) $\varepsilon = 0.001$, the maximal number of iterations equal to 250, the area $[-3.2, 3.8] \times [-3.5, 3.5]$ and $\alpha_i = 0.5\sin(i)\tan(7i) + 0.5$, $\beta_i = 20\sin(0.25i)\cos(10i) - \cos(0.5i)\tan(7.4i) + 1.25$,

(b) $\varepsilon = 0.01$, the maximal number of iterations equal to 200, the area $[-3, 3]^2$ and $\alpha_i = 25\sin(0.25i)\cos(0.3i) - \cos(0.5i)\tan(7.4i) + 1.25$, $\beta_i = 0.5$,

(c) $\varepsilon = 0.01$, the maximal number of iterations equal to 50, the area $[-3, 3]^2$ and $\alpha_i = 1.1\sin(1.1i)\tan(2.5i) + 0.9$, $\beta_i = 1.1\sin(2i)\tan(1.8i) + 10$.



(a) (b) (c)

Fig. 4. The examples of artistic patterns obtained by the use of the S-RNM.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_28

5 Conclusions

In this paper, we modified the RNM by replacing the Picard iteration with the S-iteration. The proposed S-RNM significantly accelerates the RNM, but the optimal α and β values are polynomial dependent. The S-RNM does not lose the good properties of the RNM such as global convergence, stability and robustness.

In our future study, we would like to concentrate on two aspects regarding the RNM. The first aspect will be the study on variable α_i and β_i sequences, which might further accelerate the S-RNM. In the second aspect, we will try to replace the S-iteration with other iterations known in literature and collected, for instance, in [6]. This could also lead to the acceleration of the RNM. Moreover, we will investigate the artistic applications of the patterns generated with the new methods.

References

1. Agarwal, R., O'Regan, D., Sahu, D.: Iterative construction of fixed points of nearly asymptotically nonexpansive mappings. Journal of Nonlinear and Convex Analysis **8**(1), 61–79 (2007)
2. Ardelean, G., Balog, L.: A qualitative study of Agarwal et al. iteration procedure for fixed points approximation. Creative Mathematics and Informatics **25**(2), 135–139 (2016)
3. Deuflhard, P.: Newton Methods for Nonlinear Problems. Springer-Verlag, Berlin (2011). https://doi.org/10.1007/978-3-642-23899-4
4. Epureanu, B., Greenside, H.: Fractal basins of attraction associated with a damped Newton's method. SIAM Review **40**(1), 102–109 (1998). https://doi.org/10.1137/S0036144596310033
5. Gdawiec, K.: Fractal patterns from the dynamics of combined polynomial root finding methods. Nonlinear Dynamics **90**(4), 2457–2479 (2017). https://doi.org/10.1007/s11071-017-3813-6
6. Gdawiec, K., Kotarski, W.: Polynomiography for the polynomial infinity norm via Kalantari's formula and nonstandard iterations. Applied Mathematics and Computation **307**, 17–30 (2017). https://doi.org/10.1016/j.amc.2017.02.038
7. Gdawiec, K., Kotarski, W., Lisowska, A.: On the robust Newton's method with the Mann iteration and the artistic patterns from its dynamics. Nonlinear Dynamics (in press). https://doi.org/10.1007/s11071-021-06306-5
8. Kalantari, B.: Polynomial Root-Finding and Polynomiography. World Scientific, Singapore (2009). https://doi.org/10.1142/9789812811837
9. Kalantari, B.: A geometric modulus principle for polynomials. The American Mathematical Monthly **118**(10), 931–935 (2011). https://doi.org/10.4169/amer.math.monthly.118.10.931
10. Kalantari, B.: A globally convergent Newton method for polynomials. https://arxiv.org/abs/2003.00372 (2020)
11. Nocedal, J., Wright, S.: Numerical Optimization, 2nd Edition. Springer, New York (2006). https://doi.org/10.1007/978-0-387-40065-5
12. Ouyang, P., Tang, X., Chung, K., Yu, T.: Spiral patterns of color symmetry from dynamics. Nonlinear Dynamics **94**(1), 261–272 (2018). https://doi.org/10.1007/s11071-018-4357-0

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_28