The "Gauss-Seidelization" of iterative methods for solving nonlinear equations in the complex plane*

José Manuel Gutiérrez, Ángel Alberto Magreñán and Juan Luis Varona[†]

Dpto. de Matemáticas y Computación, Universidad de La Rioja, 26004 Logroño, Spain
`jmguti@unirioja.es, angel-alberto.magrenan@unirioja.es, jvarona@unirioja.es`

Dedicated to the memory of Professor Sergio Plaza

Abstract

In this paper we introduce a process we have called "Gauss-Seidelization" for solving nonlinear equations. We have used this name because the process is inspired by the well-known Gauss-Seidel method to numerically solve a system of linear equations. Together with some convergence results, we present several numerical experiments in order to emphasize how the Gauss-Seidelization process influences on the dynamical behavior of an iterative method for solving nonlinear equations.

1 Introduction

Let us suppose that we want to solve the linear system

$$\begin{cases} ax + by = \lambda, \\ cx + dy = \mu, \end{cases} \tag{1}$$

where a, b, c, d, λ, μ are fixed real constants. Starting in a point $(x_0, y_0) \in \mathbb{R}^2$, the well-known Jacobi method for solving the linear system (1) is defined by the following iterative scheme:

$$\begin{cases} x_{n+1} = \dfrac{\lambda - by_n}{a}, \\ y_{n+1} = \dfrac{\mu - cx_n}{d}. \end{cases} \tag{2}$$

Under appropriate hypothesis, this iterative method generates a sequence $\{(x_n, y_n)\}_{n=0}^{\infty}$ that converges to (x, y), the solution of (1).

Let us suppose that $\{(x_n, y_n)\}_{n=0}^{\infty}$ is approaching the solution (x, y). To compute (x_{n+1}, y_{n+1}) in (2), we first compute x_{n+1}; heuristically speaking, we can thing that x_{n+1} is "closer to the solution" than x_n. So

---

the idea is to use x_{n+1} instead of x_n to evaluate y_{n+1} in (2). The corresponding algorithm,

$$\begin{cases} x_{n+1} = \dfrac{\lambda - by_n}{a}, \\ y_{n+1} = \dfrac{\mu - cx_{n+1}}{d}, \end{cases} \qquad (3)$$

is the well-known Gauss-Seidel method for solving (1).

Notice that we can also define the following variant of the Gauss-Seidel method, just by changing the roles of x and y:

$$\begin{cases} y_{n+1} = \dfrac{\mu - cx_n}{d}, \\ x_{n+1} = \dfrac{\lambda - by_{n+1}}{a}. \end{cases} \qquad (4)$$

At a first glance, one can think that when both methods (2) and (3) converge to the solution (x, y), the Gauss-Seidel method converges "faster" than the Jacobi method. Of course, there are rigorous results dealing with the convergence of both Jacobi and Gauss-Seidel iterative methods to solve linear systems (and not only in $\mathbb{R}^2$, but in $\mathbb{R}^d$). They can be found in many books devoted to numerical analysis. But the aim of this paper is not to study linear systems.

Instead, we are going to consider a complex function $\phi : \mathbb{C} \to \mathbb{C}$ and an iterative sequence

$$z_{n+1} = \phi(z_n), \qquad z_0 \in \mathbb{C}. \qquad (5)$$

If we take $U = \operatorname{Re} \phi$, $V = \operatorname{Im} \phi$, $z_n = x_n + y_n i$ we can write (5) as a system of recurrences

$$\begin{cases} x_{n+1} = U(x_n, y_n), \\ y_{n+1} = V(x_n, y_n), \end{cases} \qquad (x_0, y_0) \in \mathbb{R}^2. \qquad (6)$$

Although (6) is, in general, a non-linear recurrence, we can consider the same ideas used to construct the Gauss-Seidel iterative methods (3) or (4). In fact, we can define

$$\begin{cases} x_{n+1} = U(x_n, y_n), \\ y_{n+1} = V(x_{n+1}, y_n), \end{cases} \qquad (x_0, y_0) \in \mathbb{R}^2, \qquad (7)$$

or

$$\begin{cases} y_{n+1} = V(x_n, y_n), \\ x_{n+1} = U(x_n, y_{n+1}), \end{cases} \qquad (x_0, y_0) \in \mathbb{R}^2. \qquad (8)$$

We say that (7) (or (8)) are the *Gauss-Seidelization* (or the *yx-Gauss-Seidelization*) of an iterative method (5). In general, the theoretical study of the dynamics for (7) or (8) can be much more difficult than the study of dynamics for (5), because complex analysis can no longer be used in the mathematical reasoning.

For instance, let us consider the famous Mandelbrot set, defined as the set of points $c \in \mathbb{C}$ for which the orbit of 0 under iteration of the complex quadratic polynomial $z_{n+1} = z_n^2 + c$ remains bounded. In [4] it is introduced a variation of the Mandelbrot set, called as Chicho set, that is mainly a Gauss-Seidelization of the Mandelbrot set. In fact, if we write $c = a + bi$ (with $a, b \in \mathbb{R}$) and $z_n = x_n + iy_n$, the previous complex iterative sequence can be written as

$$(x_{n+1}, y_{n+1}) = M_c(x_n, y_n),$$

where

$$M_c(x, y) = (x^2 - y^2 + a, 2xy + b).$$

The Gauss-Seidelization process applied to the function $M_c(x, y)$ gives raise to the iteration of the function
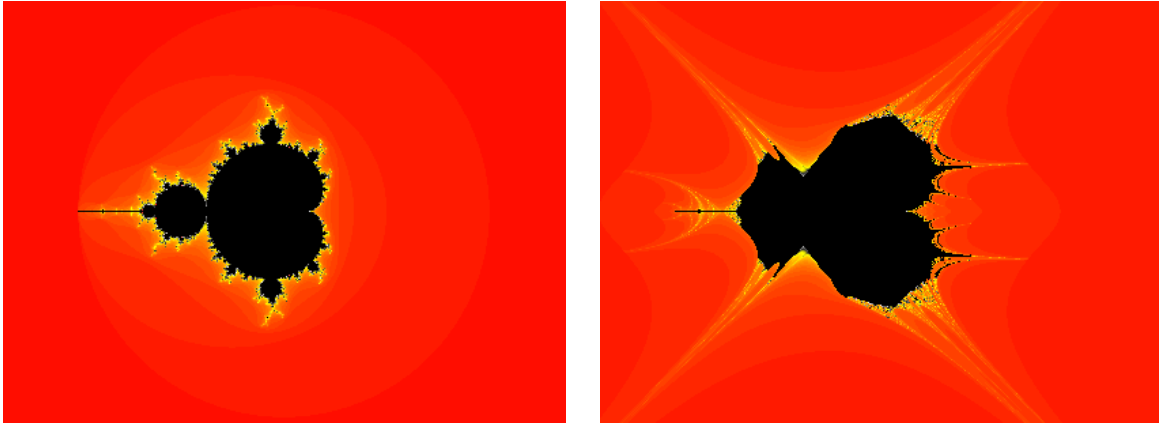
$$T_c(x, y) = (x^2 - y^2 + a, 2(x^2 - y^2 + a)y + b).$$

2

Figure 1: The Mandelbrot set and its Gauss-Seidelization, the Chicho set.

In this context, we obtain the Chicho set as the set of the parameters c such that $T_c^n(0,0)$ is bounded when $n \to \infty$.

Some basic properties and a computer picture of the Chicho set are presented in [4]. Perhaps, it could be surprising to see that both sets, Mandelbrot and Chicho, have a completely different dynamical behavior. For instance, it is well known (and easy to check) that if one of the steps of the sequence $z_{n+1} = z_n^2 + c$ has a modulus greater than 2, then $|z_n| \to \infty$. Consequently, the corresponding c does not belong to the Mandelbrot set. But, as far as we know, there is not a similar result for the Chicho set, and thus it is very difficult to build an algorithm to ensure that $c = a + bi$ is not in the Chicho set. We can compare the aspect of Mandelbrot and Chicho sets in Figure 1.

This kind of ideas can be also applied to define the Gauss-Seidelization of a Julia set on the complex plane. In particular, in this paper we are interested in studying the Gauss-Seidelization of some iterative methods that are used for solving nonlinear equations on the complex plane.

Although in this paper we are mainly concerned in some dynamical aspects about the Gauss-Seidelization process, there are other questions that eventually could be taked into account. For instance, as the computer time to calculate every step in (7) (or (8)) is equivalent to the time used for every step in (6), from a computational point of view, the Gauss-Seidelizations of a convergent iterative method could serve to increase the speed of convergence. In the same way, we can wonder ourselves about the influence of the Gauss-Seidelization process in other computational indexes such as the computational order of convergence [9] or the efficiency index [6, 13].

2   Iterative methods

The most famous iterative method for solving nonlinear equations is Newton's method (also known as Newton-Raphson's method). In the real line, if we have a differentiable function $f : \mathbb{R} \to \mathbb{R}$, and we want to find a solution $x^* \in \mathbb{R}$ we can take $x_0 \in \mathbb{R}$ and find the tangent line to the curve $y = f(x)$ on $(x_0, f(x_0))$. This tangent line intersects the real axis at a point x_1 given by $x_1 = x_0 - f(x_0)/f'(x_0)$. Usually, x_1 is a better approximation to x^* than x_0 and, of course, we can iterate and calculate x_2, x_3, and so on. Under adequate conditions for the iteration function f and the starting point x_0, the sequence $\{x_n\}_{n=0}^\infty$ tends to the root x^*.

This iterative method has a natural generalization for finding the roots of complex functions

$$f : \mathbb{C} \to \mathbb{C}.$$

In fact, Newton's method to solve nonlinear equations in the complex plane is given by

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}, \quad z_0 \in \mathbb{C}. \tag{9}$$

If we have z^* such that $f(z^*) = 0$ and we start with z_0 close enough to z^*, the iterative method (9) will converge to z^* when $n \to \infty$. In the literature there are many works that study this method, showing many different kind of sufficient conditions that guarantee its convergence (see, for instance, [2, 13] or [15]).

Given an iterative method and a root z^*, the *attraction basin* of z^* is the set of all the starting points $z_0 \in \mathbb{C}$ such that the iterative method converges to z^*. Every root has its own attraction basin and it is well known that, in general, the frontier between the attraction basins is not a simple line, but a intricate fractal, a Julia set whose Hausdorff dimension is greater than 1. This happens, in particular, when the function f is a polynomial of degree greater than 2 (see, for instance, [5] or [11, Section 13.9], although there are hundreds of papers and books that could be cited).

Clearly, (9) is a particular case of (5), so we can apply to this equation the Gauss-Seidelization processes seen in (7) or (8). To do that, let us denote $z = x + yi$ (and the same with subindexes) and $f(z) = u(x, y) + iv(x, y)$. Taking into account the Cauchy-Riemann equations, we can write $f'(z) = u_x(x, y) + iv_x(x, y)$ and then (9) becomes

$$x_{n+1} + iy_{n+1} = x_n + iy_n - \frac{u(x_n, y_n) + iv(x_n, y_n)}{u_x(x_n, y_n) + iv_x(x_n, y_n)}.$$

Now, after multiplying the numerator and the denominator by $u_x(x_n, y_n) - iv_x(x_n, y_n)$, we separate the real and imaginary parts to obtain

$$\begin{cases} x_{n+1} = x_n - \dfrac{u(x_n, y_n)u_x(x_n, y_n) + v(x_n, y_n)v_x(x_n, y_n)}{u_x(x_n, y_n)^2 + v_x(x_n, y_n)^2}, \\ y_{n+1} = y_n - \dfrac{v(x_n, y_n)u_x(x_n, y_n) - u(x_n, y_n)v_x(x_n, y_n)}{u_x(x_n, y_n)^2 + v_x(x_n, y_n)^2}, \end{cases}$$

that is equivalent to (9) and has the form (6). In this way we can easily obtain the corresponding Gauss-Seidelization (7) (or the yx-Gauss-Seidelization (8)) for the Newton method.

Let us illustrate these processes with a particular example. We apply Newton's method (9) to the function $f(z) = z^3 - 1$ to obtain

$$z_{n+1} = z_n - \frac{z_n^3 - 1}{3z_n^2}.$$

Then, taking $z_n = x_n + iy_n$ and separating into real and imaginary parts we have

$$\begin{cases} x_{n+1} = x_n - \dfrac{x_n^5 + 2x_n^3 y_n^2 - x_n^2 + x_n y_n^4 + y_n^2}{3(x_n^2 + y_n^2)^2}, \\ y_{n+1} = y_n - \dfrac{x_n^4 y_n + 2x_n^2 y_n^3 + 2x_n y_n + y_n^5}{3(x_n^2 + y_n^2)^2}. \end{cases}$$

Consequently, the Gauss-Seidelization (7) of this process is

$$\begin{cases} x_{n+1} = x_n - \dfrac{x_n^5 + 2x_n^3 y_n^2 - x_n^2 + x_n y_n^4 + y_n^2}{3(x_n^2 + y_n^2)^2}, \\ y_{n+1} = y_n - \dfrac{x_{n+1}^4 y_n + 2x_{n+1}^2 y_n^3 + 2x_{n+1} y_n + y_n^5}{3(x_{n+1}^2 + y_n^2)^2}, \end{cases}$$

and the yx-Gauss-Seidelization (8) is

$$\begin{cases} y_{n+1} = y_n - \dfrac{x_n^4 y_n + 2x_n^2 y_n^3 + 2x_n y_n + y_n^5}{3(x_n^2 + y_n)^2}, \\ x_{n+1} = x_n - \dfrac{x_n^5 + 2x_n^3 y_{n+1}^2 - x_n^2 + x_n y_{n+1}^4 + y_{n+1}^2}{3(x_n^2 + y_{n+1}^2)^2}. \end{cases}$$

What happens with the attraction basins when we make the Gauss-Seidelization process? As we have mention above, the conditions for the convergence of an iterative method and its Gauss-Seidelizations are different, and (at least for solving a linear system) often the Gauss-Seidelizations converge "faster" than the original method. Of course, this will affect to the frontier of the attraction basins. We can expect to get a simple frontier when the convergence improves. To measure how such a frontier is more or less intricate we can use its Hausdorff dimension, that will be greater if the frontier is more complicate.

We cannot exactly know the Hausdorff dimension of a fractal (the frontier that separate the attraction basins, in our case), but we can numerically compute some estimations. One of the most common algorithms for this purpose is the box-counting method (see, for instance, [11, Section 4.4]). In brief, given a rectangle of the complex plane with a fractal in it, we cover the rectangle by mean of boxes or length ℓ. Then, the fractal dimension is

$$d = \lim_{\ell \to 0} \frac{\log(N(\ell))}{-\log(\ell)},$$

where $N(\ell)$ is the number of boxes needed to completely cover the fractal. Graphically speaking, d corresponds with the slope of the plot of $\log(N(\ell))$ versus $-\log(\ell)$. Of course, we cannot do $\ell \to 0$, so the usual is to take ℓ small enough. In practice, let us assume that the rectangle is a square, and divide the square in $2^m \times 2^m$ boxes, for $m = 2, 3, \ldots$. For the "level" m, the estimated fractal dimension is

$$d_m = \frac{\log(N_m)}{\log(2^m)},$$

where N_m is number of boxes that intersect the fractal. The behavior for a m big enough can give us a suitable estimation for such fractal dimension.

In the next section, we compute the box-counting dimension for Newton's method and its Gauss-Seidelizations. In addition, we also consider other iterative methods to solve nonlinear equations, together with their corresponding Gauss-Seidelizations. To be precise, we are going to experiment with the same methods showed in [14] (where we can find precise references for these methods and many comments); a dynamical study of some of these root-finding algorithms con be found, for instance in [1, 8, 10] or [12]. In addition to Newton's method (9) we will consider the following one-point methods. In the foregoing we use the notations

$$u(z) = \frac{f(z)}{f'(z)}, \quad L_f(z) = \frac{f(z)f''(z)}{f'(z)^2}.$$

- Newton's method for multiple roots (also known as Schröder's method):

$$z_{n+1} = z_n - \frac{1}{1 - L_f(z_n)} \frac{f(z_n)}{f'(z_n)}.$$

- Convex acceleration of Whittaker's method:

$$z_{n+1} = z_n - \frac{f(z_n)}{2f'(z_n)}(2 - L_f(z_n)).$$

- Double convex acceleration of Whittaker's method:

$$z_{n+1} = z_n - \frac{f(z_n)}{4f'(z_n)}\left(2 - L_f(z_n) + \frac{4 + 2L_f(z_n)}{2 - L_f(z_n)(2 - L_f(z_n))}\right).$$

- Halley's method (also known as the method of tangent hyperbolas):

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)} \frac{2}{2 - L_f(z_n)}.$$

5

- Chebyshev's method (also known as Euler-Chebyshev's method or method of tangent parabolas):

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}\left(1 + \frac{L_f(z_n)}{2}\right).$$

- Convex acceleration of Newton's method or the super-Halley method (also known as Halley-Werner's method):

$$z_{n+1} = z_n - \frac{f(z_n)}{2f'(z_n)}\frac{2 - L_f(z_n)}{1 - L_f(z_n)}.$$

A method that is a adaptation of a fixed point method (taking into account that to solve $f(z) = 0$ is the same than solving $z - f(z) = 0$):

- (Shifted) Stirling's method:

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n - f(z_n))}.$$

Finally, we also consider the following multipoint iterative methods:

- Steffensen's method:

$$z_{n+1} = z_n - \frac{f(z_n)}{g(z_n)}$$

  with $g(z) = (f(z + f(z)) - f(z))/f(z)$.

- Midpoint method:

$$z_{n+1} = z_n - \frac{f(z_n)}{f'\left(z_n - u(z_n)/2\right)}.$$

- Traub-Ostrowski's method:

$$z_{n+1} = z_n - u(z_n)\frac{f(z_n - u(z_n)) - f(z_n)}{2f(z_n - u(z_n)) - f(z_n)}.$$

- Jarratt's method:

$$z_{n+1} = z_n - \tfrac{1}{2}u(z_n) + \frac{f(z_n)}{f'(z_n) - 3f'(z_n - \tfrac{2}{3}u(z_n))}.$$

- Inverse-free Jarratt's method:

$$z_{n+1} = z_n - u(z_n) + \frac{3}{4}u(z_n)h(z_n)\left(1 - \frac{3}{2}h(z_n)\right),$$

with $h(z) = \frac{f'\left(z - \frac{2}{3}u(z)\right) - f'(z)}{f'(z)}$.

3 Numerical experiments

We can make thousands of numerical experiments just by considering different functions f and next by applying the abovementioned iterative methods together with their corresponding Gauss-Seidelizations (7) and (8). In particular, we are interested in calculating the fractal dimension of the frontier of the attraction basins of the roots, that is the fractal dimension of the involved Julia sets. Our goal is to check the influence of the Gauss-Seidelization process in such fractal dimensions. To do that, we consider as a test function the following one:

$$f(z) = z^3 - 1,$$

defined in the rectangle $[-2.5, 2.5] \times [-2.5, 2.5]$. For this particular choice, we will graphic the Julia sets for all the methods considered in Section 2 and their corresponding Gauss-Seidelizations. In all these cases, we will compute the corresponding box-counting dimension.

The roots of $f(z) = z^3 - 1$ are 1, $e^{2\pi i/3}$ and $e^{4\pi i/3}$. Their basins of attraction are colored in cyan, magenta and yellow respectively. To be more precise, we assign the color to a point z_0 in the rectangle $[-2.5, 2.5] \times [-2.5, 2.5]$ if the iterative method starting from z_0 converges with a fixed precision $|z_n - \text{root}| < 10^{-3}$ in a maximum of 25 iterations of the method. We mark the point in black if the method does not converge to any of the roots with these criteria. In addition, we make the color lighter or darker according to the number of iterations needed to reach the root with the required precision.

In Figures 2, 3, 4 and 5 we show the pictures corresponding to the attraction basins of the three roots for all the methods detailed in the previous sections, as well as their Gauss-Seidelizations (7) and xy-Gauss-Seidelizations (8). Note that Traub-Ostrowski's method and Jarratt's method coincide when they are applied to cubic polynomials, so they have the same pictures.

In all these cases we have included the box-counting dimension d of the frontier between the attraction basins. To numerically compute these dimensions we have proceeded as follows:

1. For each iterative method, we have decomposed the square $[-2.5, 2.5] \times [-2.5, 2.5]$ on $2^m \times 2^m$ in small boxes, with $m = 5, 6, 7, 8, 9$ and 10 (successively). Next, we try to calculate an estimation N_m of the number of small boxes that contains points in the Julia set.

2. In the first case, for $m = 5$, we take 10 points in every side of the corresponding small box; then we say that the small box contains points in the Julia set if the dynamic of one of these points is different to the dynamic of the center of the box.

3. In the successive cases $m = 6, 7, 8, 9, 10$ we use the same method, but taking $20, 40, 80, 160, 320$ points in every side of the corresponding small boxes. Notice that with this scheme we make a more accurate analysis in the regions that are more intricate.

4. With the data obtained in the previous steps, we calculate the box-counting dimension d as the slope of the linear regression of the points

$$\{(\log(2^m), \log(N_m)) : m = 5, 6, \ldots, 10\}.$$

Note that the box-counting dimension depends essentially on the procedure used to decide if a small box contains points in the Julia set or not. So different procedures can produce small variations in the obtained dimension; for instance, the estimation given in [7] for the Julia set associated to Newton's method for $z^3 - 1$ is 1.42, that is slightly slower than the estimate 1.45 we have obtained. Nevertheless our goal in this paper is not to compare the dimensions obtained by using different methods. What we would like to highlight here is the procedure itself to calculate fractal dimensions and the fact that all the fractal dimensions obtained in this paper have been obtained by following the aforementioned procedure: all the dimensions in this paper haven calculated by the same procedure so we can compare them.
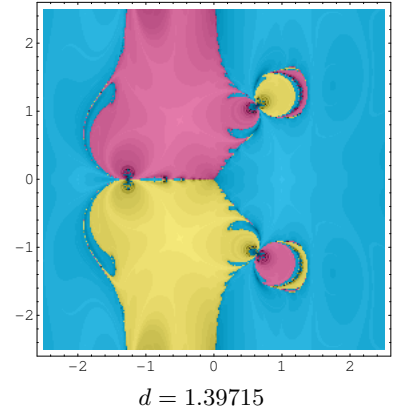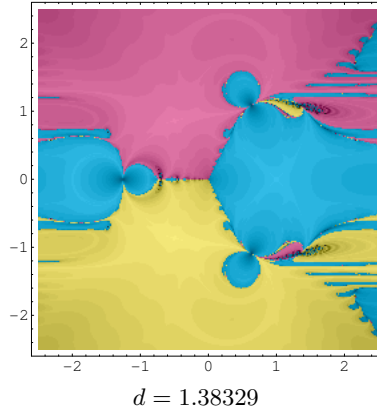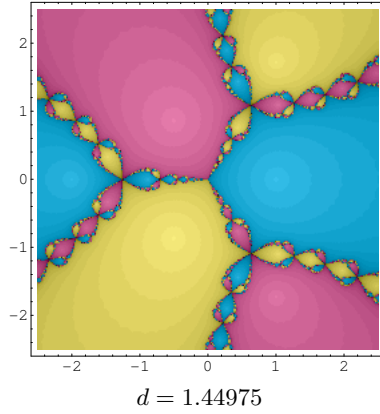
The pictures shown in Figures 2 to 5 allow us to see how the basin of attraction, the Julia sets and the fractal dimensions of an iterative method changes with a Gauss-Seidelization process. Even more, we can appreciate how demanding the method is regarding to the starting point of the iterative process. In fact, a first graphical inspection shows that a method requires more conditions on the initial point when the associated fractal becomes more complicated. In all the cases of our particular experiment, we can see that the two Gauss-Seidelizations generate a fractal with lower dimension than the original method. This fact corroborate empirically the idea underlying these notes: the Gauss-Seidelization process produces "less intricate" Julia sets. As far as we know, there is not theoretical results supporting this idea and we are aware that we have not included any kind of justification in these notes. But the experiments done with other functions usually produce very similar results to the particular case $f(z) = z^3 - 1$ considered in this paper.

In the examples included in Figures 2 to 5, the graphics related with the yx-Gauss-Seidelization process (8) that are shown in the right column have in general a lower fractal dimension than the graphics related

- Newton's method:
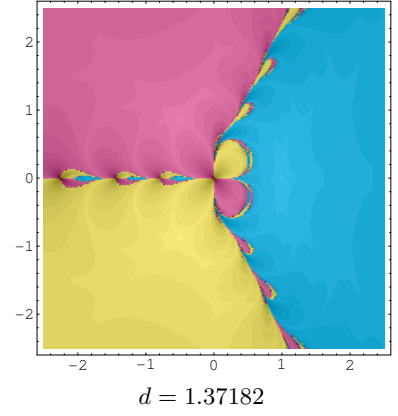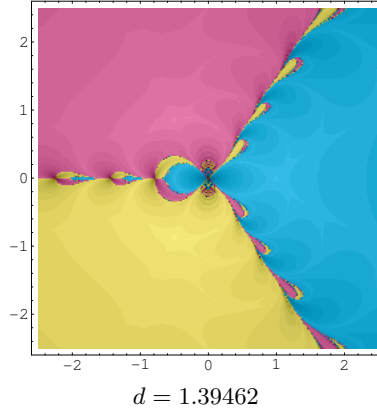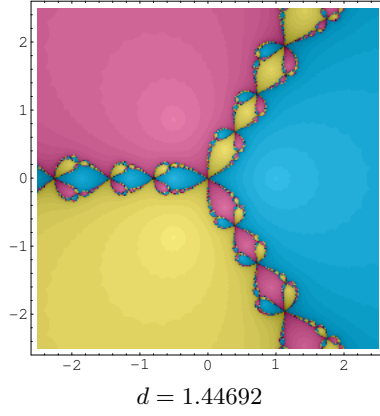
$$d = 1.44692 \qquad d = 1.39462 \qquad d = 1.37182$$

- Newton's method for multiple roots:

$$d = 1.44975 \qquad d = 1.38329 \qquad d = 1.39715$$

- Convex acceleration of Whittaker's method:

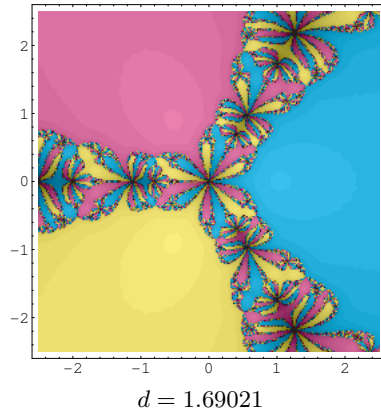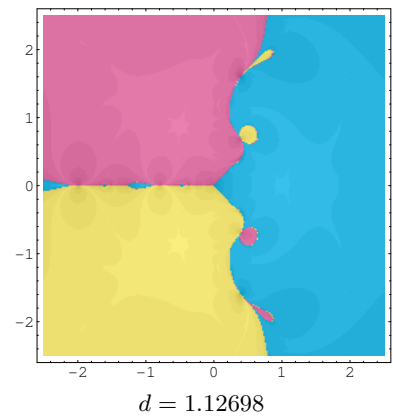$$d = 1.73137 \qquad d = 1.71972 \qquad d = 1.71701$$
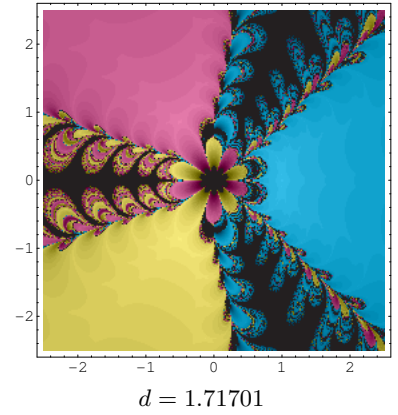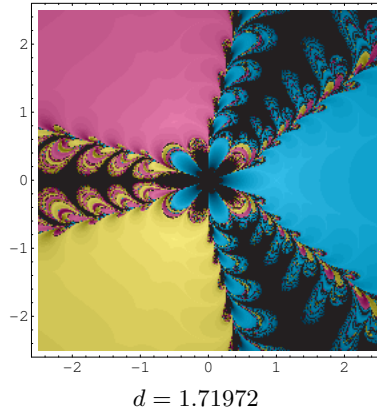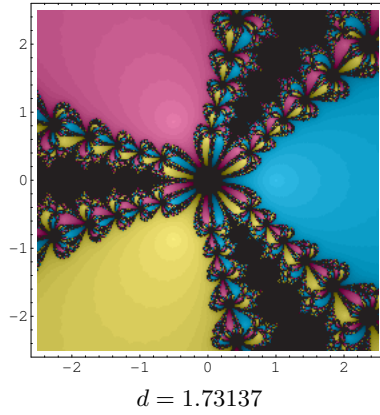
Figure 2: Attraction basins of some iterative methods and their Gauss-Seidelizations, with the corresponding box-counting dimensions d.

8

- Double convex acceleration of Whittaker's method:

$d = 1.69021$

$d = 1.64686$

$d = 1.6081$

- Halley's method:

$d = 1.21446$

$d = 1.16534$

$d = 1.12698$

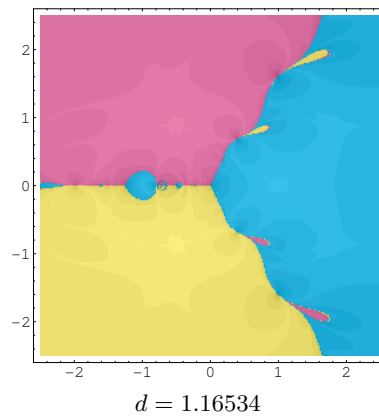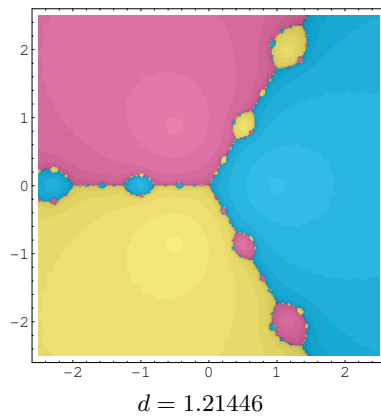- Chebyshev's method:

$d = 1.56943$

$d = 1.55426$

$d = 1.49167$
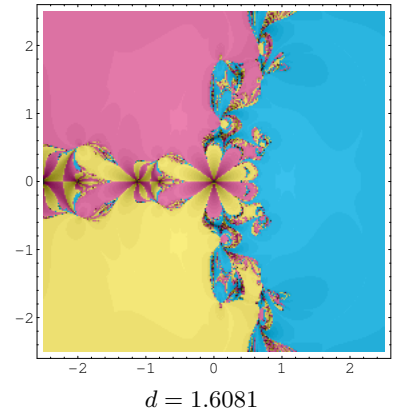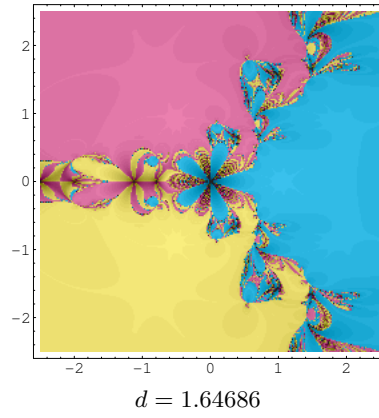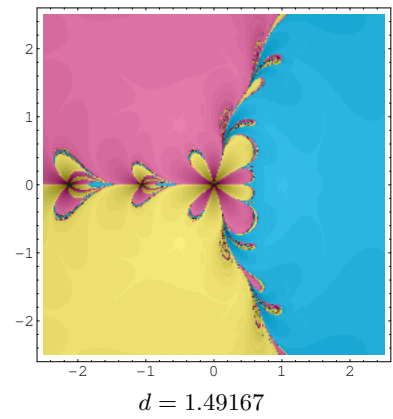
Figure 3: Attraction basins of some iterative methods and their Gauss-Seidelizations, with the corresponding box-counting dimensions d.

- Convex acceleration of Newton's method (or super-Halley):

$d = 1.31023$ $d = 1.25284$ $d = 1.23593$

- Stirling's method:

$d = 1.37075$ $d = 1.35342$ $d = 1.35355$

- Steffensen's method:

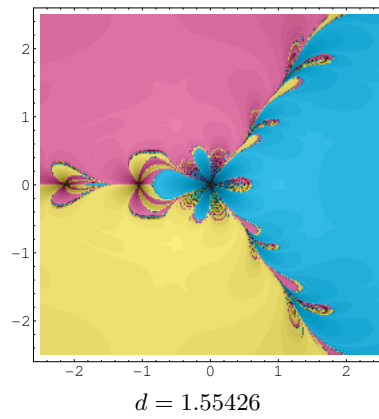$d = 1.52055$ $d = 1.42731$ $d = 1.46027$

Figure 4: Attraction basins of some iterative methods and their Gauss-Seidelizations, with the corresponding box-counting dimensions d.

- Midpoint method:

$$d = 1.45978 \qquad d = 1.42257 \qquad d = 1.39210$$

- Traub-Ostrowski's method and Jarratt's method:
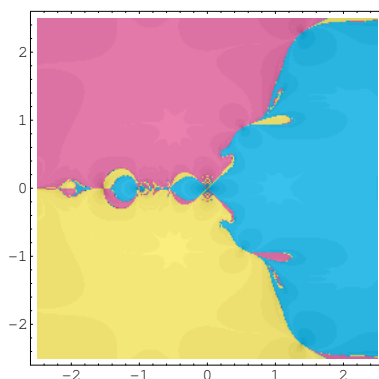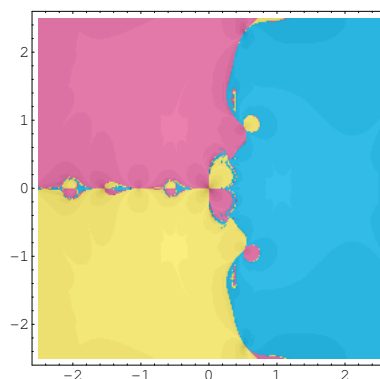
$$d = 1.36387 \qquad d = 1.35137 \qquad d = 1.32805$$

- Inverse-free Jarratt's method:
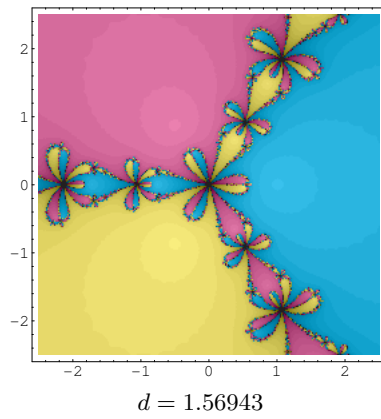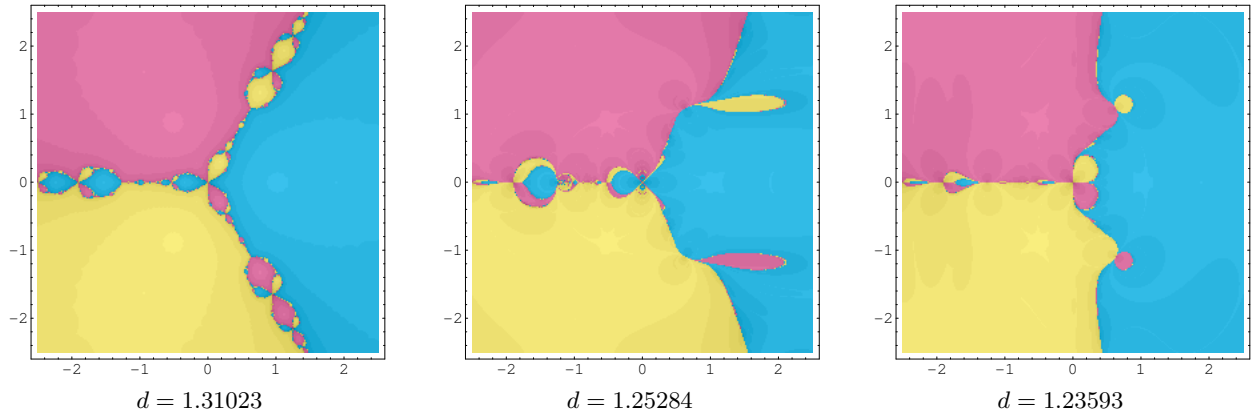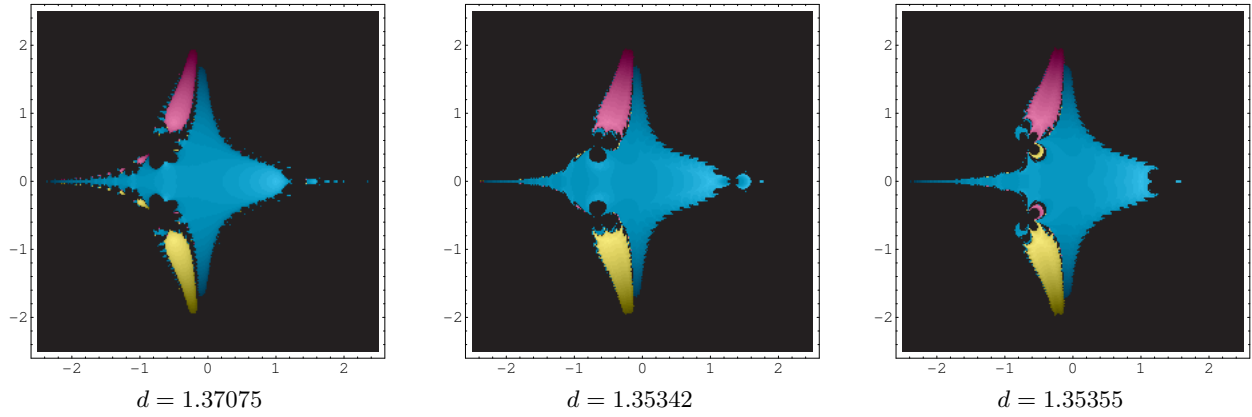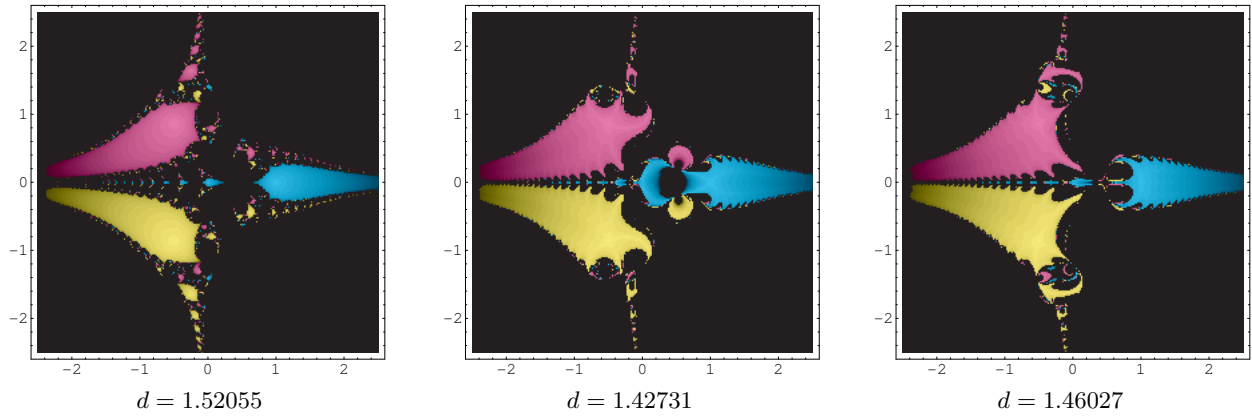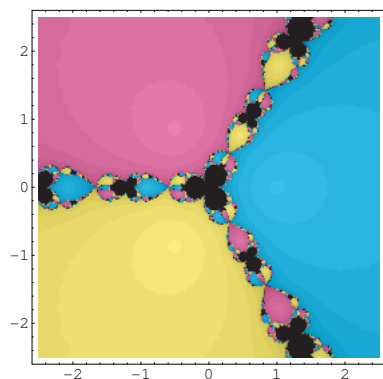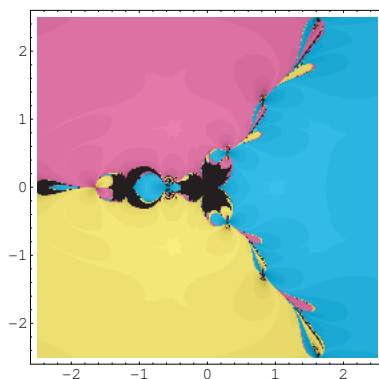

$$d = 1.65063 \qquad d = 1.65084 \qquad d = 1.58334$$

Figure 5: Attraction basins of some iterative methods and their Gauss-Seidelizations, with the corresponding box-counting dimensions d.
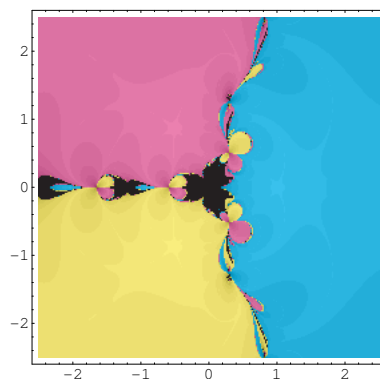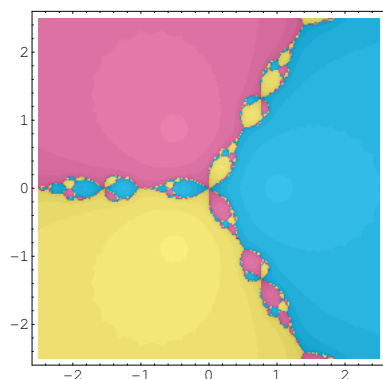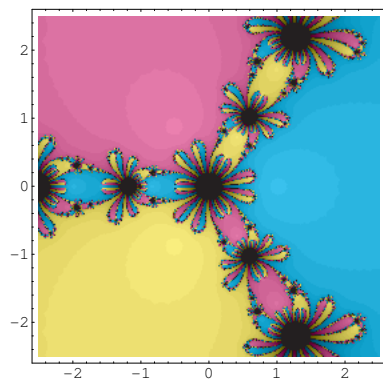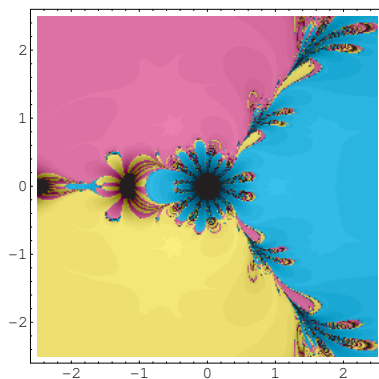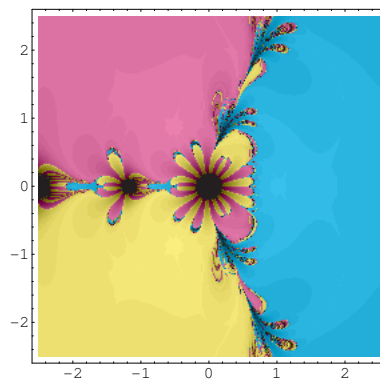
Method	Equation $z^3 - 1 = 0$			Equation $z^3 - i = 0$		
	Standard	GS	yx-GS	Standard	GS	yx-GS
Nw	1.44692	1.39462	1.37182	1.44125	1.36616	1.39103
NwM	1.44975	1.38329	1.39715	1.45122	1.36523	1.35362
CaWh	1.73137	1.71972	1.71701	1.73775	1.72504	1.72456
DcaWh	1.69021	1.64686	1.60810	1.69871	1.61497	1.65612
Ha	1.21446	1.16534	1.12698	1.20688	1.10896	1.14811
Ch	1.56943	1.55426	1.49167	1.56019	1.48333	1.56511
CaN/sH	1.31023	1.25284	1.23593	1.30455	1.19240	1.25126
Stir	1.37075	1.35342	1.35355	1.33015	1.35963	1.34954
Steff	1.52055	1.42731	1.46027	1.45247	1.44377	1.42029
Mid	1.45978	1.42257	1.39210	1.45199	1.39655	1.43551
Tr-Os & Ja	1.36387	1.35137	1.32805	1.35691	1.32447	1.35199
IfJa	1.65063	1.65084	1.58334	1.65063	1.58538	1.66333

Table 1: Box-counting dimensions corresponding to the iterative methods to solve $z^3 - 1 = 0$ and their Gauss-Seidelizations, and to solve $z^3 - i = 0$ and their Gauss-Seidelizations (the methods, denoted with short labels, follow the same order than in Section 2).

with the ordinary Gauss-Seidelization process (7). This fact could be a consequence of the symmetry of the roots of the equation $z^3 - 1 = 0$ respect to the x-axis. But this is not a general rule. For instance, let us consider the equation $z^3 - i = 0$ instead of $z^3 - 1 = 0$. In this case, the roots are symmetric respect to the y-axis. In this situation the Gauss-Seidelization process (7) usually provide the lowest fractal dimensions. We do not reproduce the corresponding pictures in this paper, but we show in Table 1 a comparative between the fractal dimensions of both cases, $z^3 - 1 = 0$ and $z^3 - i = 0$ (of course, for $z^3 - i = 0$ we compute the box-counting dimension in the same way than for $z^3 - 1 = 0$).

Another experiment that can be done to compare the original methods with their Gauss-Seidelizations is to shown the proportion of divergent points. With this aim, we are going to use a grid of 1024×1024 points in the square $[-2.5, 2.5] \times [-2.5, 2.5]$. Then, for the methods considered in Section 2 and their Gauss-Seidelizations, we count how many of these points generate divergent sequences when the iterative method starts on them; in any case, we assume that the sequence diverges if has not reached a root with precision 10^{-6} when the method is iterated 25 times. We have done these numerical experiments to solve $f(z) = 0$ both for $f(z) = z^3 - 1$ and $f(z) = z^3 - i$. We show the results in Table 2. As we can expect according the aforementioned behaviors and the pictures, the Gauss-Seidelization processes always provide a meaningful reduction of the quantity of divergent points. (Except for Stirling and Steffensen methods, that are rather peculiar, observe in the table the symmetry between GS and yx-GS when we change $z^3 - 1$ by $z^3 - i$.)

4   Convergence results

Let us assume that the iterative method $z_{n+1} = \phi(z_n)$ is used to numerically solve an equation $f(z) = 0$, where $f : \mathbb{C} \to \mathbb{C}$. That is, the limit of the sequence $\{z_n\}$, a fixed point of ϕ, is precisely a solution of the previous equation. All the methods showed in Section 2 can be written in this form for a suitable iteration function $\phi(z)$.

From a theoretical point of view, the Gauss-Seidelization process of the method $\phi(z)$ can be written in the following form:

$$z_{n+1/3} = \phi(z_n), \tag{10}$$

$$z_{n+2/3} = \operatorname{Re}(z_{n+1/3}) + i \operatorname{Im}(z_n), \tag{11}$$

$$z_{n+1} = \phi(z_{n+2/3}). \tag{12}$$

	Equation $z^3 - 1 = 0$			Equation $z^3 - i = 0$		
Method	Standard	GS	yx-GS	Standard	GS	yx-GS
Nw	0.125885	0.016022	0.009918	0.125885	0.009918	0.016022
NwM	0.133705	0.057793	0.049782	0.133705	0.049782	0.057793
CaWh	29.6438	21.3057	18.6897	29.6438	18.6897	21.3057
DcaWh	1.13640	0.417519	0.253105	1.13640	0.253105	0.417519
Ha	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Ch	0.578499	0.244331	0.113106	0.578499	0.113106	0.244331
CaN/sH	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Stir	87.3001	86.1042	86.1088	94.9564	93.3659	93.0178
Steff	85.3380	81.5908	82.6757	78.3817	75.3695	75.8236
Mid	4.63600	2.85053	2.43340	4.63600	2.43340	2.85053
Tr-Os & Ja	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
IfJa	4.70905	2.21519	1.49555	4.70905	1.49555	2.21519

Table 2: Percentage of divergent points corresponding to the iterative methods to solve $z^3 - 1 = 0$ and their Gauss-Seidelizations, and to solve $z^3 - i = 0$ and their Gauss-Seidelizations (the methods, denoted with short labels, follow the same order than in Section 2).

Prior to continue, let us note that the above decomposition is not, in general, advisable for practical implementation in a computer if we are interested in fast computations. To explain it, let us assume that a step of (5) is equivalent to a step of (6), both using a time T, and that a step of (7) requires approximately the same time than a step of (6). Then, a step of the Gauss-Seidelization (7) uses a time T, whereas (10)–(12) (although mathematically serve to obtain the same result) uses a time $2T$.

However, and this is our interest here, the decomposition (10)–(12) is useful to establish theoretical results about the convergence of a method after a Gauss-Seidelization process. Actually, we can state the following results. From now on we use the notation $z_k = x_k + iy_k$ to indicate the real and imaginary parts of an iterate z_k defined in the previous process.

Theorem 1. *Let $\xi = \xi_x + i\xi_y$ be a fixed point of ϕ. Let us assume that ϕ satisfies a center-Lipschitz condition in the form*

$$\|\phi(z) - \xi\|_\infty \leq C\|z - \xi\|_\infty$$

with $0 < C < 1$ on a certain domain

$$\Omega = \left\{ z = x + yi : \|z - \xi\|_\infty = \max\{|x - \xi_x|, |y - \xi_y|\} < R \right\}.$$

Then the sequence $\{z_n\}$ defined by the Gauss-Seidelization process (7) (or equivalently by (10)–(12)) and starting at $z_0 \in \Omega$ converges to ξ.

Proof. Let us start at $z_0 \in \Omega$ and let us assume that $z_n \in \Omega$ for a given $n \in \mathbb{N}$, that is, $\|z_n - \xi\|_\infty < R$. Then $z_{n+1/3}$ defined in (10) belongs to Ω:

$$\|z_{n+1/3} - \xi\|_\infty = \|\phi(z_n) - \xi\|_\infty \leq C\|z_n - \xi\|_\infty < R.$$

Now we have that $z_{n+2/3}$ defined in (11) is also inside Ω:

$$\begin{aligned}
\|z_{n+2/3} - \xi\|_\infty &= \max\{|x_{n+2/3} - \xi_x|, |y_{n+2/3} - \xi_y|\} \\
&= \max\{|x_{n+1/3} - \xi_x|, |y_n - \xi_y|\} \\
&\leq \max\{\|z_{n+1/3} - \xi\|_\infty, \|z_n - \xi\|_\infty\} < R.
\end{aligned}$$

13

Consequently, we can define z_{n+1} and, in addition,

$$\begin{aligned}
\|z_{n+1} - \xi\|_\infty = \|\phi(z_{n+2/3}) - \xi\|_\infty &\leq C\|z_{n+2/3} - \xi\|_\infty \\
&= C \max\{|x_{n+2/3} - \xi_x|, |y_{n+2/3} - \xi_y|\} \\
&= C \max\{|x_{n+1/3} - \xi_x|, |y_n - \xi_y|\} \\
&\leq C \max\{\|z_{n+1/3} - \xi\|_\infty, \|z_n - \xi\|_\infty\} \\
&\leq C \max\{C\|z_n - \xi\|_\infty, \|z_n - \xi\|_\infty\} = C\|z_n - \xi\|_\infty.
\end{aligned}$$

Then we have

$$\|z_n - \xi\|_\infty \leq C^n\|z_0 - \xi\|_\infty$$

with $0 < C < 1$. This inequality guarantees the convergence of the sequence $\{z_n\}$ defined by (10)–(12) to the limit ξ. $\qquad\square$

The previous result depends clearly of the chosen norm, the ∞-norm (otherwise, we cannot guarantee that $z_{n+2/3} \in \Omega$ and the proof is not valid). But, on the other hand, we are aware that the usual norm for complex numbers is the euclidean norm defined by

$$\|z\|_2 = \sqrt{x^2 + y^2}, \quad z = x + iy.$$

Therein, as a consequence of Theorem 1 and by taking into account that $\|\cdot\|_\infty \leq \|\cdot\|_2 \leq \sqrt{2}\|\cdot\|_\infty$, we can give the following result.

Corollary 2. *Let* $\xi = \xi_x + i\xi_y$ *be a fixed point of* ϕ. *Let us assume that* ϕ *satisfies a center-Lipschitz condition*

$$\|\phi(z) - \xi\|_2 \leq C\|z - \xi\|_2$$

with $0 < C < 1/\sqrt{2}$ *on a certain domain*

$$\Omega_1 = \left\{z = x + yi : \|z - \xi\|_2 < R_1\right\}.$$

In addition, let us denote

$$\Omega_0 = \left\{z = x + yi : \|z - \xi\|_\infty < \frac{\sqrt{2}R_1}{2}\right\}.$$

Then, if $z_k \in \Omega_0$ *for some* $k \in \mathbb{N}$ *(for instance, if we start at* $z_0 \in \Omega_0$*), the sequence* $\{z_n\}$ *defined by the Gauss-Seidelization process* (7) *(or, equivalently, by* (10)–(12)*) converges to* ξ.

Proof. Firstly notice that $\Omega_0 \subseteq \Omega_1$. Secondly, we have that ϕ also satisfies a center-Lipschitz condition with $\|\cdot\|_\infty$ on Ω_0. Actually, for $z \in \Omega_0 \subseteq \Omega_1$ we have

$$\|\phi(z) - \xi\|_\infty \leq \|\phi(z) - \xi\|_2 \leq C\|z - \xi\|_2 \leq C\sqrt{2}\|\phi(z) - \xi\|_\infty$$

with $\sqrt{2}C < 1$.

Then, if $z_k \in \Omega_0$ for some $k \in \mathbb{N}$, by Theorem 1 we have that $z_{k+j} \in \Omega_0$ for all $j \geq 1$ and

$$\|z_{k+j} - \xi\|_\infty \leq \left(\sqrt{2}C\right)^j \|z_k - \xi\|_\infty.$$

Consequently $z_{k+j} \in \Omega_1$ for all $j \geq 1$ and

$$\|z_{k+j} - \xi\|_2 \leq \sqrt{2}\|z_{k+j} - \xi\|_\infty \leq \sqrt{2}\left(\sqrt{2}C\right)^j \|z_k - \xi\|_\infty \leq \sqrt{2}\left(\sqrt{2}C\right)^j \|z_k - \xi\|_2,$$

and this implies the convergence of the sequence $\{z_n\}$ defined by (10)–(12) to the limit ξ. $\qquad\square$

All the methods ϕ considered in Section 2 for solving a nonlinear equation $f(z) = 0$ are convergent with an order of convergence ranging from 2 to 4. Consequently these methods and, in general, all the methods with order of convergence p bigger than 1, satisfy an error equation in the form

$$\|z_{n+1} - \xi\| \leq C\|z_n - \xi\|^p,$$

where ξ is the root of $f(z) = 0$, p is the order of convergence of each method and C is the asymptotic error constant (see [3] for an exhaustive study of p and C in the most usual higher order iterative methods for solving nonlinear equations). We would like to emphasize that an inequality of this kind exists for any chosen norm.

We can state the following result that guarantees that if we have an iterative method with order of convergence $p > 1$, then its Gauss-Seidelization has order of convergence at least p. We would like to highlight that this result can be applied to all the methods considered in Section 2 because all of them have at least quadratic convergence, that is $p \geq 2$.

Theorem 3. *Let $\xi = \xi_x + i\xi_y$ be a fixed point of ϕ. Let us assume that ϕ satisfies*

$$\|\phi(z) - \xi\|_\infty \leq C\|z - \xi\|_\infty^p \tag{13}$$

with $p > 1$ and $C > 0$ on a certain domain

$$\Omega = \left\{ z \in \mathbb{C} : \|z - \xi\|_\infty < R \right\}.$$

Moreover, for a given $\lambda \in (0, 1)$, let us take $R_2 \leq R$ small enough to ensure that

$$C\|z - \xi\|_\infty^{p-1} \leq \lambda$$

for each $z \in \Omega_2$, where

$$\Omega_2 = \left\{ z \in \mathbb{C} : \|z - \xi\|_\infty < R_2 \right\}.$$

Then the sequence $\{z_n\}$ defined by the Gauss-Seidelization process (7) (or, equivalently, by (10)–(12)) and starting at $z_0 \in \Omega_2$ converges to ξ with order of convergence at least p.

Proof. Firstly, notice that condition (13) guarantees that the sequence generated by the iteration function ϕ locally converges to ξ with order p. Now, let us consider the sequence $\{z_n\}$ defined by the Gauss-Seidelization procedure (7). We want to show that $\{z_n\}$ is also convergent to ξ with order p, that is,

$$\|z_{n+1} - \xi\|_\infty \leq C\|z_n - \xi\|_\infty^p.$$

Let us start at $z_0 \in \Omega_2$, and assume that $z_n \in \Omega_2$ for a given $n \in \mathbb{N}$. Then

$$\|z_{n+1/3} - \xi\|_\infty = \|\phi(z_n) - \xi\|_\infty \leq C\|z_n - \xi\|_\infty^p$$
$$= C\|z_n - \xi\|_\infty^{p-1}\|z_n - \xi\|_\infty \leq \lambda\|z_n - \xi\|_\infty,$$

and this implies that $z_{n+1/3}$ defined in (10) belongs to Ω_2. Now, as in the proof of Theorem 1, we obtain the inequality

$$\|z_{n+2/3} - \xi\|_\infty \leq \max\{\|z_{n+1/3} - \xi\|_\infty, \|z_n - \xi\|_\infty\},$$

so $z_{n+2/3}$ defined in (11) is also in Ω_2. Finally we have

$$\|z_{n+1} - \xi\|_\infty = \|\phi(z_{n+2/3}) - \xi\|_\infty \leq C\|z_{n+2/3} - \xi\|_\infty^p$$
$$\leq C\max\{\|z_{n+1/3} - \xi\|_\infty^p, \|z_n - \xi\|_\infty^p\}$$
$$\leq C\|z_n - \xi\|_\infty^p,$$

and consequently $z_{n+1} \in \Omega_2$. In addition, the previous inequality shows that the Gauss-Seidelization procedure defined in (7) has order of convergence at least p. $\qquad\square$

Let us finish this section by noticing that, although in Theorems 1 and 3 and Corollary 2 we have established convergence results for the Gauss-Seidelization process defined in (7), we can give twin results on the convergence of the yx-Gauss-Seidelization process defined in (8). In this case we must define the sequences

$$\begin{aligned}
z'_{n+1/3} &= \phi(z'_n), \\
z'_{n+2/3} &= \operatorname{Re}(z'_n) + i\operatorname{Im}(z'_{n+1/3}), \\
z'_{n+1} &= \phi(z'_{n+2/3}),
\end{aligned}$$

instead of the sequences (10)–(12) and follow the procedures detailed in the proofs of Theorems 1 and 3 and Corollary 2.

5   Conclusions

In a similar way than Jacobi iterative method to solve linear systems can be transformed into the Gauss-Seidel method, we can do the same with iterative methods for solving nonlinear equations in the complex plane. In this way, we get the so-called "Gauss-Seidelization" of the method.

As the Gauss-Seidel method to solve linear systems usually produces better results than the Jacobi method, we can expect a similar behavior for the Gauss-Seidelization process in the nonlinear case. Eventually, this could be used to increase the speed of convergence of the original iterative method.

When we want to solve a complex equation $f(z) = 0$ by mean of an iterative method, the attraction basins of the roots are separated, in general, by a intricate frontier of fractal nature. The fractal dimension of the frontier is a measure of how intricate is a frontier and, in some way, it serves to indicate how demanding the method is regarding the starting point to find a solution. We have experimentally computed the box-counting dimension of this fractal for many iterative methods and its Gauss-Seidelizations, and we have concluded that, apparently, the dimension in the Gauss-Seidelizations are lower than in the original methods; this suggests that the Gauss-Seidelizations are less demanding with respect to the starting point. In addition, we have measured the quantity of divergent points. These numerical experiments show that he Gauss-Seidelization processes usually produce an important decrease of the number of such points.

We have stated some theoretical results in order to ensure that the Gauss-Seidelization of an iterative method converges and, moreover, it preserves the order of convergence. In other words, if we have a method with order of convergence p, then its Gauss-Seidelization has order of convergence at least p.

Although the theoretical result given in this paper does not provide any advantage between using an iterative method or its corresponding Gauss-Seidelization, the numerical experiments seem to show that the Gauss-Seidelization process allow us to obtain, at least in some sense, better iterative methods for solving nonlinear equations.

References

[1] S. Amat, S. Busquier and S. Plaza, Review of some iterative root-finding methods from a dynamical point of view, *Sci. Ser. A Math. Sci. (N.S.)* **10** (2004), 3–35.

[2] I. K. Argyros, *Convergence and applications of Newton-type iterations*, Springer, 2008.

[3] D. K. R. Babajee, *Analysis of higher order variants of Newton's method and their applications to differential and integral equations and in ocean acidification*, PhD Thesis, University of Mauritius, 2010.

[4] M. Benito, J. M. Gutiérrez and V. Lanchares, Chicho's fractal (Spanish), *Margarita mathematica en memoria de José Javier (Chicho) Guadalupe Hernández*, 247–254, Univ. La Rioja, Logroño, 2001.

[5] P. Blanchard, The dynamics of Newton's method. Complex dynamical systems (Cincinnati, OH, 1994), 139–154, *Proc. Sympos. Appl. Math.*, 49, Amer. Math. Soc., Providence, RI, 1994.

[6] A. Cordero, J. L. Hueso, E. Martínez and J. R. Torregrosa, Efficient high-order methods based on golden ratio for nonlinear systems, *Appl. Math. Comput.* **217** (2011), no. 9, 4548–4556.

[7] B. I. Epureanu and H. S. Greenside, Fractal basins of attraction associated with a damped Newton's method, *SIAM Rev.* **40** (1998), no. 1, 102–109.

[8] W. J. Gilbert, Generalizations of Newton's method, *Fractals* **9** (2001), no. 3, 251–262.

[9] M. Grau-Sánchez, M. Noguera and J. M. Gutiérrez, On some computational orders of convergence, *Appl. Math. Lett.* **23** (2010), no. 4, 472–478.

[10] K. Kneisl, Julia sets for the super-Newton method, Cauchy's method, and Halley's method, *Chaos* **11** (2001), no. 2, 359–370.

[11] H.-O. Peitgen, H. Jürgens and D. Saupe, *Chaos and fractals: New frontiers of science*, 2nd ed., Springer-Verlag, 2004.

[12] G. E. Roberts and J. Horgan-Kobelski, Newton's versus Halley's method: a dynamical systems approach, *Internat. J. Bifur. Chaos Appl. Sci. Engrg.* **14** (2004), no. 10, 3459–3475.

[13] J. F. Traub, *Iterative methods for the solution of equations*, Prentice-Hall, 1964.

[14] J. L. Varona, Graphic and numerical comparison between iterative methods, *Math. Intelligencer* **24** (2002), no. 1, 37–46.

[15] L. Yau and A. Ben-Israel, The Newton and Halley methods for complex roots, *Amer. Math. Monthly* **105** (1998), no. 9, 806–818.