CMMSE: Jacobian-free vectorial iterative scheme to find several solutions simultaneously

Paula Triguero-Navarro[1], Alicia Cordero[2], and Juan Ramon Torregrosa[2]

[1]Centro Universitario Escuela de Empresarios
[2]Universitat Politecnica de Valencia Institut de Matematica Multidisciplinar

September 15, 2023

Abstract

This manuscript is devoted to a derivative-free parametric iterative step to obtain roots simultaneously for both nonlinear systems and equations. We prove that when it is added to an arbitrary scheme, it doubles the convergence order of the original procedure and defines a new algorithm that obtains several solutions simultaneously. Different numerical experiments are carried out to check the behaviour of the iterative methods by changing the value of the parameter and the initial guesses. Also, it is perform a numerical example where the dynamical planes are carried out to see and compare the basins of attraction.

ORIGINAL ARTICLE

Journal Section

CMMSE: Jacobian-free vectorial iterative scheme to find several solutions simultaneously

Alicia Cordero [2] | Juan R. Torregrosa[2] | Paula Triguero-Navarro*[1]

[1]Centro Universitario EDEM- Escuela de Empresarios, València, Spain

[2]Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, València, Spain

Correspondence
Email: ptrinav@doctor.upv.es

Funding information
This research was partially supported by Centro Universitario EDEM - Escuela de Empresarios, Valencia, Spain and by Universitat Politècnica de València Contrato Predoctoral PAID-01-20-17 (UPV).

This manuscript is devoted to a derivative-free parametric iterative step to obtain roots simultaneously for both nonlinear systems and equations. We prove that when it is added to an arbitrary scheme, it doubles the convergence order of the original procedure and defines a new algorithm that obtains several solutions simultaneously. Different numerical experiments are carried out to check the behaviour of the iterative methods by changing the value of the parameter and the initial guesses. Also, it is perform a numerical example where the dynamical planes are carried out to see and compare the basins of attraction.

KEYWORDS
Iterative procedures; Nonlinear equations and systems; Simultaneously; Derivative-free; Jacobian-free

Introduction and design of the iterative step

Iterative methods have become very important in recent years because it is not always possible to solve a nonlinear equation or system in an exact way. These iterative processes make it possible to obtain a approximations that define a convergent sequence (under certain conditions) to the solution of the problem.

In general, iterative methods only converge to one solution each time, which is why iterative methods that obtain several solutions simultaneously have also become increasingly popular recently.

One of them is the Abert-Ehrlich method, see [8], with the following expression:

$$x_i^{(k+1)} = x_i^{(k)} - \frac{f(x_i^{(k)})}{f'(x_i^{(k)}) - f(x_i^{(k)}) \sum_{j=1, j\neq i}^{n} \frac{1}{x_i^{(k)} - x_j^{(k)}}}, \ k = 0, 1, \ldots,$$

which is a well known method for solving polynomial equations simultaneously.

In [8], it is only studied the case in which the functions to be solved are polynomials, with simple roots, and not any kind of arbitrary equation. Moreover, in most of the literature it is assumed that the equations to be solved are polynomials, see for instance [12, 17].

For this reason, we highlight the relevance of the study we carried out in [6], where we study how to add the Ehrlich step to any other iterative scheme, thus obtaining a new method that obtains solutions simultaneously. In this paper, the order of convergence is analysed for arbitrary equations, not only polynomials.

In [5], the particular case in which the functions are polynomials is analysed, as well as a case in which the solutions of arbitrary nonlinear equations have different multiplicities.

As can be seen in the expression of the iterative method, the iterates are evaluated in $f'(x)$, so the problem to solve must be derivable in order to be able to apply this method. For that reason, we propose to replace the derivative by a divided difference operator and thus obtain the following iterative method that we propose for n simultaneous roots of an scalar equation $f(x) = 0$, that we denote by DFS_ϕ:

$$\begin{cases} y_i^{(k)} = \phi(x_i^{(k)}), \text{ for } i = 1, \ldots, n \\ x_i^{(k+1)} = y_i^{(k)} - \dfrac{f(y_i^{(k)})}{f[y_i^{(k)}, y_i^{(k)} + \beta_i f(y_i^{(k)})] - f(y_i^{(k)}) \sum_{j=1, j\neq i}^{n} \frac{1}{y_i^{(k)} - y_j^{(k)}}}, \text{ for } i, j = 1, \ldots, n, \end{cases} \quad (1)$$

where ϕ is the fixed point function that describes any iterative method for equations.

In addition, in [4], the iterative step studied in [6] is modified in order to obtain a compatible iterative process for systems, a subject that, to our knowledge, has not been done for systems of arbitrary nonlinear equations.

This iterative method uses the Jacobian matrix evaluated in the iterations, so, as in the case of equations, the function describing the system must be derivable. For this reason, an idea arises to extend the step (1) to systems in order to obtain a derivative-free scheme. We will first discuss the notation used and then show the iterative step.

Suppose that the system $F(x) = 0$, where $F : \mathbb{C}^n \to \mathbb{C}^n$, has n solutions $\alpha_i = \left(\alpha_{i_1}, \alpha_{i_2}, \ldots, \alpha_{i_m}\right)$, for $i = 1, \ldots, n$.

Our aim now is to design an iterative step, that is able to estimate all the solutions in a simultaneous way. So, we consider a set of n seeds denoted by $x_i^{(0)} = \left(x_{i_1}^{(0)}, x_{i_2}^{(0)}, \ldots, x_{i_m}^{(0)}\right)$, for $i = 1, \ldots, n$.

We define $S_i(x^{(k)}) := \left(S_{i,1}(x^{(k)}), S_{i,2}(x^{(k)}), \ldots, S_{i,m}(x^{(k)})\right)$, where

$$S_{i,r}(x^{(k)}) = \sum_{j\neq i} \frac{1}{x_{i_r}^{(k)} - x_{j_r}^{(k)}}.$$

We then extend the step (1) to systems and obtain the following iterative step:

$$x_i^{(k+1)} = x_i^{(k)} - \left([x_i^{(k)}, x_i^{(k)} + \beta_i F(x_i^{(k)}); F] - F(x_i^{(k)}) S_i(x^{(k)})\right)^{-1} F(x_i^{(k)}). \quad (2)$$

It is clear that the size of operator $[x_i^{(k)}, x_i^{(k)} + \beta_i F(x_i^{(k)}); F]$ is $d \times m$ which coincides with the size of the product of column vector of size $d \times 1$ $F\left(x_i^{(k)}\right)$ and the row vector of size $1 \times m$ $S_i(x^{(k)})$. Let us denote this scheme by JFS.

The rest of this manuscript is structured as follows. In Section 2, we design the iterative steps analyzing the order of convergence of the resulting methods for nonlinear equations and systems. In Section 3, we carry out several numerical experiments to see the behaviour of the new methods and we finish the work in Section 4 with conclusions derived from the study.

1 | CONVERGENCE ANALYSIS

Now we demonstrate that scheme DFS_ϕ has order $2p$ whenever ϕ is a scheme with order p, for any parameters β_i, $i = 1, \ldots, n$, with $\beta_i \in \mathbb{R}\backslash\{0\}$.

Theorem 1 *Let us consider a sufficiently differentiable function $f : D \subseteq \mathbb{C} \longrightarrow \mathbb{C}$ defined in a neighbourhood D of α_i, for $i = 1, \ldots, n$, satisfying $f(\alpha_i) = 0$, for $i = 1, \ldots, n$. Let us also assume that $f'(\alpha_i) \neq 0$ for $i = 1, \ldots, n$. If ϕ is an iterative scheme with convergence order p, then, given an estimate $x^{(0)} \in \mathbb{C}^n$ close enough to $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, the sequence of iterates $\{x^{(k)}\}$ generated by the method DFS_ϕ (1) converges to α with order $2p$.*

Proof We denote by $e_{i,k} = x_i^{(k)} - \alpha_i$. Let us notice that ϕ denotes an iterative procedure with convergence order p, so $e_{y,i,k} \sim e_{i,k}^p$.

Applying Taylor's expansion to $f(y_i^{(k)})$ around α_i, we obtain

$$f(y_i^{(k)}) = f'(\alpha_i)\left(e_{y,i,k} + C_{2,i} e_{y,i,k}^2 + O(e_{y,i,k}^3)\right),$$

being $C_{2,i} = \dfrac{f''(\alpha_i)}{2f'(\alpha_i)}$.

$$f[y_i^{(k)}, y_i^{(k)} + \beta_i f(y_i^{(k)})] = f'(\alpha_i)(1 + C_{2,i}(2 + \beta_i f'(\alpha_i))e_{y,i,k}) + O(e_{y,i,k}^2).$$

From the above equalities, the following identity is obtained

$$f[y_i^{(k)}, y_i^{(k)} + \beta_i f(y_i^{(k)})] - f(y_i^{(k)}) \sum_{j=1}^{n} \frac{1}{y_i^{(k)} - y_j^{(k)}} = f'(\alpha_i)\left(1 + \left(C_{2,i}(2 + \beta_i f'(\alpha_i)) - \sum_{j=1}^{n} \frac{1}{y_i^{(k)} - y_j^{(k)}}\right) e_{y,i,k} + O(e_{y,i,k}^2)\right).$$

Then,

$$
\begin{aligned}
x_i^{(k+1)} - \alpha_i &= y_i^{(k)} - \alpha_i - \frac{f(y_i^k)}{f[y_i^{(k)}, y_i^{(k)} + \beta_i f(y_i^{(k)})] - f(y_i^{(k)}) \sum_{j=1, j\neq i}^{n} \dfrac{1}{y_i^{(k)} - y_j^{(k)}}} \\
&= e_{y,i,k} - \frac{e_{y,i,k} + C_{2,i} e_{y,i,k}^2 + O(e_{y,i,k}^3)}{1 + \left(C_{2,i}(2 + \beta_i f'(\alpha_i)) - \sum_{j=1}^{n} \dfrac{1}{y_i^{(k)} - y_j^{(k)}} \right) e_{y,i,k} + O(e_{y,i,k}^2)} \\
&= \frac{\left(C_{2,i}(2 + \beta_i f'(\alpha_i)) - \sum_{j=1}^{n} \dfrac{1}{y_i^{(k)} - y_j^{(k)}} - C_{2,i} \right) e_{y,i,k}^2 + O(e_{y,i,k}^3)}{1 + \left(C_{2,i}(2 + \beta_i f'(\alpha_i)) - \sum_{j=1}^{n} \dfrac{1}{y_i^{(k)} - y_j^{(k)}} \right) e_{y,i,k} + O(e_{y,i,k}^2)} \\
&= \frac{\left(C_{2,i}(1 + \beta_i f'(\alpha_i)) - \sum_{j=1}^{n} \dfrac{1}{y_i^{(k)} - y_j^{(k)}} \right) e_{y,i,k}^2 + O(e_{y,i,k}^3)}{1 + \left(C_{2,i}(2 + \beta_i f'(\alpha_i)) - \sum_{j=1}^{n} \dfrac{1}{y_i^{(k)} - y_j^{(k)}} \right) e_{y,i,k} + O(e_{y,i,k}^2)}.
\end{aligned}
$$

Thus, by the previous relation and given that ϕ has order p we obtain that

$$
e_{i,k+1} \sim e_{y,i,k}^2 \sim (e_{i,k}^p)^2 \sim e_{i,k}^{2p}.
$$

Thus, it is proven that the method DFS_ϕ has order of convergence $2p$.

In the following, we prove that JFS has order of convergence 2 for any value of parameters β_i, $\beta_i \in \mathbb{R}\backslash\{0\}$, for $i = 1, \ldots, n$.

Theorem 2 *Let us consider a sufficiently differentiable function $F : \mathbb{R}^m \longrightarrow \mathbb{R}^d$ defined in a convex neighbourhood of α_i,denoted by $D_i \subset \mathbb{R}^m$, satisfying $F(\alpha_i) = 0$, $i = 1, ..., n$. Also, let us assume that, for $i = 1, ..., n$, $F'(\alpha_i)$ is nonsingular. Then, using a seed $x_i^{(0)} \in \mathbb{R}^m$ close enough to α_i, for $i = 1, ..., n$, the sequences $\{x_i^{(k)}\}_{k\geq 0}$ generated by the iterations of method JFS converge to α_i with order 2.*

Proof Let us denote $F = (F_1, F_2, \ldots, F_d)$, where $F_p : \mathbb{R}^m \to \mathbb{R}$ are the coordinate functions of F, $p = 1, 2, \ldots, d$.

Consider now the Taylor development of $F_p\left(x_i^{(k)}\right)$ around α_i, for $p = 1, 2, \ldots, d$:

$$
F_p\left(x_i^{(k)}\right) = \sum_{j_1=1}^{m} \frac{\partial F_p(\alpha_i)}{\partial x_{j_1}} e_{i,k_{j_1}} + \sum_{j_1=1}^{m}\sum_{j_2=1}^{m} \frac{\partial^2 F_p(\alpha_i)}{\partial x_{j_1}\partial x_{j_2}} e_{i,k_{j_1}} e_{i,k_{j_2}} + O_3\left(e_{i,k}\right),
$$

where $e_{i,k_{j_1}} = x_{i_{j_1}}^{(k)} - \alpha_{i_{j_1}}$ for $i \in \{1, \ldots, n\}$ and $j_1 \in \{1, 2, \ldots, m\}$. The residual $O_3\left(e_{i,k}\right)$ contains the elements of the Taylor expansion where the sums of the exponents of $e_{i,k_{j_1}}^q$, $j_1 \in \{1, 2, \ldots, m\}$, satisfy $q \geq 3$.

Since

$$
F_p(x_i^{(k)}) S_{i,r}(x^{(k)}) = \left(\sum_{j_1=1}^{m} \frac{\partial F_p(\alpha_i)}{\partial x_{j_1}} e_{i,k_{j_1}} \right) S_{i,r}(x^{(k)}) + O_2(e_{i,k}), \tag{3}
$$

then, we can rewrite the relation as $F_p(x_i^{(k)})S_{i,r}(x^{(k)}) = A_{p,r}e_{i,k} + O(e_{i,k})$, for $p = 1, \dots, d$ and $r = 1, \dots, m$.

Therefore, we can rewrite $F(x_i^{(k)})S_i(x^{(k)})$ as $Ae_{i,k} + O_2(e_{i,k})$.

On the other hand,

$$[x_i^{(k)}, x_i^{(k)} + \beta_i F(x_i^{(k)}); F] = F'(x_i^{(k)}) + \frac{1}{2}F''(x_i^{(k)})h + O_2(h),$$

where $h = \beta_i F(x_i^{(k)})$ by using Genocchi-Hermite [10].

Now, we consider Taylor's development of $F''(x_i^{(k)})$, $F'(x_i^{(k)})$ and $F(x_i^{(k)})$ around α_i,

$$\begin{aligned}
F(x_i^{(k)}) &= F'(\alpha_i)(e_{i,k} + C_{2,i}e_{i,k}^2 + O_3(e_{i,k})),\\
F'(x_i^{(k)}) &= F'(\alpha_i)(I + 2C_{2,i}e_{i,k} + O_2(e_{i,k})),\\
F''(x_i^{(k)}) &= F'(\alpha_i)(2C_{2,i} + O(e_{i,k})).
\end{aligned}$$

Therefore,

$$\begin{aligned}
[x_i^{(k)}, x_i^{(k)} + \beta_i F(x_i^{(k)}); F] &= F'(\alpha_i)\left(I + 2C_{2,i}e_{i,k} + \frac{1}{2}2C_{2,i}(\beta_i F'(\alpha_i)e_{i,k})\right) + O_2(e_{i,k})\\
&= F'(\alpha_i)\left(I + C_{2,i}\left(2I + \beta_i F'(\alpha_i)\right)e_{i,k}\right) + O_2(e_{i,k}).
\end{aligned} \tag{4}$$

Then, by using (3) and (4), we obtain

$$\begin{aligned}
\left([x_i^{(k)}, x_i^{(k)} + \beta_i F(x_i^{(k)}); F] - F(x_i^{(k)})S_i(x^{(k)})\right) &= F'(\alpha_i)\left(I + C_{2,i}\left(2I + \beta_i F'(\alpha_i)\right)e_{i,k}\right) - Ae_{i,k} + O_2\left(e_{i,k}\right)\\
&= F'(\alpha_i)\left(I + \left(C_{2,i}\left(2I + \beta_i F'(\alpha_i)\right) - F'(\alpha_i)^{-1}A\right)e_{i,k}\right) + O_2\left(e_{i,k}\right).
\end{aligned} \tag{5}$$

From (5), it follows

$$\left([x_i^{(k)}, x_i^{(k)} + \beta_i F(x_i^{(k)}); F] - F(x_i^{(k)})S_i(x^{(k)})\right)^{-1} = \left(I - \left(C_{2,i}\left(2I + \beta_i F'(\alpha_i)\right) - F'(\alpha_i)^{-1}A\right)e_{i,k}\right)F'(\alpha_i)^{-1} + O_2\left(e_{i,k}\right). \tag{6}$$

Then, by using (6),

$$\begin{aligned}
e_{i,k+1} &= e_{i,k} - \left(I - \left(C_{2,i}\left(2I + \beta_i F'(\alpha_i)\right) - F'(\alpha_i)^{-1}A\right)e_{i,k}\right)\left(e_{i,k} + C_{2,i}e_{i,k}^2\right) + O_3\left(e_{i,k}\right)\\
&= e_{i,k} - e_{i,k} - C_{2,i}e_{i,k}^2 + \left(C_{2,i}\left(2I + \beta_i F'(\alpha_i)\right) - F'(\alpha_i)^{-1}A\right)e_{i,k}^2 + O_3\left(e_{i,k}\right)\\
&= \left(C_{2,i}\left(I + \beta_i F'(\alpha_i)\right) - F'(\alpha_i)^{-1}A\right)e_{i,k}^2 + O_3\left(e_{i,k}\right).
\end{aligned}$$

Therefore, it is proven that JFS holds order of convergence 2.

It can be demonstrated in a simple way, as in [6], that if we combine any iterative method for systems with the iterative step (2), we will obtain a new method that obtains several solutions simultaneously with duplicated order regarding the original scheme for systems.

2 | NUMERICAL RESULTS

For the computational calculations, we use the software Matlab R2020b with variable precision arithmetic of 5000 digits. The stopping criterium used is the norm of the function F evaluated at the last iteration $x^{(k+1)} = \left(x_1^{(k+1)}, x_2^{(k+1)}, \ldots, x_n^{(k+1)}\right)$ is lower than the tolerance a, which we modify according to the problem. When we search n solutions,

$$\frac{1}{n}\sum_{i=1}^{n}\left\|F\left(x_i^{(k+1)}\right)\right\| < a.$$

Let us remark that

$$\left\|F\left(x^{(k+1)}\right)\right\| := \frac{1}{n}\sum_{i=1}^{n}\left\|F\left(x_i^{(k+1)}\right)\right\|.$$

We set the maximum of iterations at 100 as a stopping criterion.

We are going to compare the performance of methods in the different examples by using different aspects, as the approximate solution found, the mean norm of the residual $\frac{1}{n}\sum_{i=1}^{n}\left\|F\left(x_i^{(k+1)}\right)\right\| < a$, the norm of the difference between the last two approximations, the amount of iterations needed to reach the required tolerance, the computational time and the Approximate Computational Order of Convergence (ACOC), defined in [7] by Cordero and Torregrosa,

$$p \approx ACOC = \frac{\ln\left(\|x^{(k+1)} - x^{(k)}\| / \|x^{(k)} - x^{(k-1)}\|\right)}{\ln\left(\|x^{(k)} - x^{(k-1)}\| / \|x^{(k-1)} - x^{(k-2)}\|\right)}.$$

2.1 | Nonlinear equation

First, we start by solving a nonlinear equation, which is not a polynomial, to check that our iterative method obtains good results for non-polynomial equations. In this case, the tolerance chosen is 10^{-200}. The equation to solve is $f(x) = e^{x^2} - x = 0$, where the solutions are $s_1 \approx 0.61435 + 0.68106i$ and $s_2 \approx 0.61435 - 0.68106i$. As seed, we have chosen $x^{(0)} = (-i, i)$.

In Table 1, the results obtained by Ehrlich method for arbitrary functions and by the DFS method are shown by changing the value of β, that in this case is equal for all the components of the initial guess. It is illustrated in the table that all iterative methods require 11 iterations to satisfy the stopping criterion as well as all methods obtain the same $ACOC$ which coincides with the expected theoretical convergence order.

TABLE 1 Numerical results for $f(x) = e^{x^2} - x = 0$

Method	$\|x^{(k+1)} - x^{(k)}\|_2$	$\|f(x^{(k+1)})\|_2$	Iteration	ACOC
Ehrlich	6.1897×10^{-186}	3.8288×10^{-371}	11	2.0
DFS with $\beta = 0.1$	6.0534×10^{-199}	3.8458×10^{-397}	11	2.0
DFS with $\beta = -0.1$	6.2936×10^{-157}	3.8755×10^{-313}	11	2.0
DFS with $\beta = 0.5$	8.8698×10^{-135}	1.1467×10^{-268}	11	2.0

2.2 | Freudenstein-Roth function

Next, we will solve the Freudenstein-Roth function, see [9], denoted as $FR(x) = 0$, which is determined as follows:

$$(-13 + x_1 + (5x_2 - x_2^2 - 2)x_2)^2 + (-29 + x_1 + (x_2^2 + x_2 - 14)x_2)^2 = 0.$$

This problem has 3 solutions, which are:

- $(5, 4)$,
- $(13 + 14i, -1 + i)$,
- $(13 - 14i, -1 - i)$.

In this case, the stopping criterion is 10^{-100}, and as initial guess, we choose

- $x_1^{(0)} = (6, 6)$,
- $x_2^{(0)} = (13 + 13i, i)$,
- $x_2^{(0)} = (13 - 13i, -i)$.

The results obtain for the Freudenstein-Roth function are shown in Table 2. The methods employed are PS, from [6], and JFS by changing the values of the parameter β, which in this case is equal for all the components of the initial guess. Here it is shown that depending on how we change the values of the parameter, the results obtained also vary. Some of the elements of the JFS family obtain similar results to those obtained by the PS method, but on other occasions they need a greater number of iterations or do not converge under these circumstances.

TABLE 2 Numerical results for $FR(x) = 0$

Method	$\|x^{(k+1)} - x^{(k)}\|$	$\|FR(x^{(k+1)})\|$	Iteration	ACOC
PS	2.3739×10^{-73}	2.058×10^{-147}	10	2.0001
JFS with $\beta = 0.1$	8.1354×10^{-88}	4.5384×10^{-175}	14	2.0639
JFS with $\beta = -0.1$	n.c.	n.c.	n.c.	n.c.
JFS with $\beta = 0.5$	n.c.	n.c.	n.c.	n.c.
JFS with $\beta = 0.01$	3.4005×10^{-91}	3.144×10^{-183}	11	2.0
JFS with $\beta = -0.01$	1.9307×10^{-52}	1.3047×10^{-105}	10	2.0001
JFS with $\beta = 0.001$	4.7502×10^{-76}	8.2636×10^{-153}	10	2.0001
JFS with $\beta = -0.001$	5.1394×10^{-87}	2.3928×10^{-173}	11	2.0121

2.3 | No differentiable system

In the following, a non-differentiable problem is solved in order to see the results obtained by the elements of the JFS family, which in this case the value of the parameter is equal for all the components of the initial guess. The problem to be solved in this case is the system $D(x) = 0$, illustrated as

$$D(x) = \begin{cases} x_1 x_2 - |x_1| = 0, \\ x_1 x_2 - |x_2| = 0. \end{cases} \tag{7}$$

The solutions of this system are $(-1, -1)$ and $(1, 1)$. The stopping criterion determined in this case is 10^{-100}. As initial vector we employ $x_1^{(0)} = (-2, -2)$ and $x_2^{(0)} = (2, 2)$.

Table 3 shows the results obtained for certain elements of the JFS family by changing the values of the parameter. In the data collected in this table, it is shown that the ACOC is similar to that expected in all cases, and as before, depending on the value used as parameter, the elements of the family needs a greater or fewer number of iterations.

TABLE 3 Numerical results for $D(x) = 0$

Method	$\|x^{(k+1)} - x^{(k)}\|$	$\|D(x^{(k+1)})\|$	Iteration	ACOC
JFS with $\beta = 0.1$	2.2427×10^{-75}	1.7783×10^{-151}	7	2.0
JFS with $\beta = -0.1$	2.2427×10^{-75}	1.7783×10^{-151}	7	2.0
JFS with $\beta = 0.01$	2.4455×10^{-78}	2.1145×10^{-158}	6	1.9986
JFS with $\beta = -0.01$	2.4455×10^{-78}	2.1145×10^{-158}	6	1.9986
JFS with $\beta = 0.005$	3.8449×10^{-90}	2.6133×10^{-182}	6	1.9996
JFS with $\beta = -0.005$	3.8449×10^{-90}	2.6133×10^{-182}	6	1.9996
JFS with $\beta = 0.5$	3.4588×10^{-58}	2.1149×10^{-116}	8	2.0
JFS with $\beta = -0.5$	3.4588×10^{-58}	2.1149×10^{-116}	8	2.0

2.4 | Solving a system with different initial guesses

Below, a problem from [1] is solved. The system to be solved is $B(x) = 0$, defined as follows

$$B(x) = \begin{cases} 2\arctan(x_1 + 1) + x_2 - 3 = 0, \\ \arctan(x_1 + 1)x_2 - 1 = 0. \end{cases} \tag{8}$$

The solutions of this systems are approximately $(-0.4537, 2)$ and $(0.55741, 1)$. The stopping criterion determined in this case is 10^{-100}. In Table 4, we use $2x_1^{(0)} = (-1, 1.5)$ and $x_2^{(0)} = (0, 0.5)$ as the initial vector. As in the previous cases, depending on the values of the parameter for the elements of JFS, different results are obtained. In this case, for several of the parameter values the results obtained for the JFS elements are obtained in a smaller number of iterations than for the PS method.

TABLE 4 Numerical results for $B(x) = 0$

Method	$\left\|x^{(k+1)} - x^{(k)}\right\|$	$\left\|B(x^{(k+1)})\right\|$	Iteration	ACOC
PS	8.4954×10^{-87}	5.1034×10^{-173}	15	2.0
JFS with $\beta = 0.1$	n.c.	n.c.	n.c.	n.c.
JFS with $\beta = -0.1$	2.6927×10^{-60}	5.0×10^{-120}	10	1.9992
JFS with $\beta = 0.5$	n.c.	n.c.	n.c.	n.c.
JFS with $\beta = -0.5$	7.2215×10^{-82}	5.085×10^{-163}	10	2.0001
JFS with $\beta = 0.01$	1.1022×10^{-93}	8.6361×10^{-187}	13	2.0
JFS with $\beta = -0.01$	4.0003×10^{-81}	1.1251×10^{-161}	13	2.0
JFS with $\beta = 0.001$	4.8349×10^{-93}	1.6539×10^{-185}	16	2.0
JFS with $\beta = -0.001$	1.4701×10^{-59}	1.5274×10^{-118}	14	2.0

In Table 5, as initial vector we employ $x_1^{(0)} = (-1, 1.5)$ and $x_2^{(0)} = (0, 0.6)$, to illustrate what happens when one of the initial vectors is slightly different. We can see that by suddenly changing one of the values of $x^{(0)}$, the number of iterations needed is much smaller than in the previous case.

TABLE 5 Numerical results for $B(x) = 0$

Method	$\left\|x^{(k+1)} - x^{(k)}\right\|$	$\left\|B(x^{(k+1)})\right\|$	Iteration	ACOC
PS	1.4761×10^{-61}	1.3439×10^{-122}	11	2.0
JFS with $\beta = 0.1$	1.1503×10^{-54}	9.7224×10^{-109}	12	1.9997
JFS with $\beta = -0.1$	7.9837×10^{-70}	4.3954×10^{-139}	10	2.0
JFS with $\beta = 0.5$	n.c.	n.c.	n.c.	n.c.
JFS with $\beta = -0.5$	1.8466×10^{-83}	3.3212×10^{-166}	10	2.0001
JFS with $\beta = 0.01$	1.6569×10^{-53}	1.6744×10^{-106}	11	2.0
JFS with $\beta = -0.01$	1.8241×10^{-59}	2.0766×10^{-138}	11	2.0
JFS with $\beta = 0.001$	9.2722×10^{-61}	5.2963×10^{-121}	11	2.0
JFS with $\beta = -0.001$	2.3577×10^{-62}	3.4322×10^{-124}	11	2.0

2.5 | Searching equilibrium solutions in N-body problem

In this case, we solve the same problem as the one studied in the article [2]. We denote the problem to be solved as $A(x, y) = 0$:

$$A(x,y) = \begin{cases} \left(\sqrt{3}x - y\right)\left(1 - \dfrac{1}{\left(x^2+y^2\right)^{3/2}}\right) + \mu_1\left(\sqrt{3}(x-1)+y\right)\left(1 - \dfrac{1}{\left((x-1)^2+y^2\right)^{3/2}}\right) = 0, \\ 2y\left(1 - \dfrac{1}{\left(x^2+y^2\right)^{3/2}}\right) + \mu_2\left(\sqrt{3}(x-1)+y\right)\left(1 - \dfrac{1}{\left(\left(x-\frac{1}{2}\right)^2 + \left(y - \frac{\sqrt{3}}{2}\right)^2\right)^{3/2}}\right) = 0. \end{cases} \tag{9}$$

This problem has between 8 and 10 roots depending on the values of μ_1 and μ_2. In this case we will solve for what happens when $\mu_1 = 0.65$ and $\mu_2 = 0.65$. For this particular case, we have a total of 8 solutions, in the range $[-1, 2] \times [-1, 2]$.

Since the problem has another level of complexity, the chosen stopping criterion is 10^{-5} and the number of digits of precision is 500.

As initial estimates the following set of vectors are selected

- $x_1^{(0)} = (-0.6, -0.3)$,
- $x_2^{(0)} = (-0.3, 0.8)$,
- $x_3^{(0)} = (0.3, 0.4)$,
- $x_4^{(0)} = (0.54, 0)$,
- $x_5^{(0)} = (0.55, -0.7)$,
- $x_6^{(0)} = (0.58, 1.4)$,
- $x_7^{(0)} = (1.2, 0.7)$,
- $x_8^{(0)} = (1.5, -0.2)$.

Table 6 shows the numerical results got by the methods illustrated in the first column of the table. In this case it is shown that for the parameter value -0.1, not all the solutions are found, only 6 of the solutions are found, while for the other values and PS we obtain all the solutions simultaneously. The approximate convergence orders are slightly higher than 2 in some of the cases. As before, most of the methods obtain similar results, and by changing the value of the parameter, we can perform one more or less iteration depending on the one chosen.

TABLE 6 Numerical results for $A(x) = 0$

Method	$\left\|x^{(k+1)} - x^{(k)}\right\|$	$\left\|A(x^{(k+1)})\right\|$	Iteration	ACOC
PS	8.8472-$\times 10^{-5}$	9.4542×10^{-8}	5	2.33
JFS with $\beta = 0.1$	4.3383×10^{-4}	6.9396×10^{-6}	6	2.1205
JFS with $\beta = -0.1$	Only found 6 solutions			
JFS with $\beta = 0.01$	3.0777×10^{-5}	8.3186×10^{-9}	5	2.1905
JFS with $\beta = -0.01$	2.2256×10^{-4}	7.6916×10^{-7}	5	2.4565

2.6 | Himmelblau system

The convergence of an iterative method is dependent on the initial guess chosen, so what we will do next is to generate random vectors as initial guesses to observe the number of roots obtained.

In this case, the Himmelblau system is solved, see [9], the structure of which is:

$$H(x) = \begin{cases} 4x_1x_2 + 4x_1^3 + 2x_2^2 - (42x_1 + 14) = 0, \\ 4x_1x_2 + 4x_2^3 - 26x_2 + 2x_1^2 - 22 = 0. \end{cases} \tag{10}$$

The approximate solutions of this system are

- $r_1 \approx (3, 2)$
- $r_2 \approx (3.58, -1.85)$
- $r_3 \approx (-3.78, -3.3)$
- $r_4 \approx (3.39, 0.0739)$
- $r_5 \approx (-2.81, 3.13)$
- $r_6 \approx (-0.271, -0.923)$
- $r_7 \approx (0.0867, 2.88)$
- $r_8 \approx (-3.07, -0.0814)$
- $r_9 \approx (-0.128, -1.95)$

Given 9 solutions, we generate 9 vectors of 2 components uniformly randomly between $[-5.5] \times [-5.5]$.

In this case, the chosen tolerance is 10^{-10} and the maximum allowed iterations are also reduced to 50.

In order to obtain an average of how many times we converge to each root or how many roots we converge to, we will repeat the random vector generation process ten times, and we will denote each root generation process by Pi, where i denotes the number of trial.

In Table 7, we illustrate the iterative method and the number of times it has converged to each of root, while in Table 8 we show the number of roots obtained in each random vector generation test and an average of the number of roots obtained.

TABLE 7 Number of times the solutions are found by each method

Method	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9
PS	5	3	2	5	4	3	3	4	7
JFS with $\beta = 0.01$	8	7	5	3	6	8	4	6	3
JFS with $\beta = 0.1$	6	9	8	0	8	2	2	3	9
JFS with $\beta = -0.1$	9	8	6	3	6	1	3	4	1
JFS with $\beta = -0.01$	8	1	4	6	5	3	3	2	2
JFS with $\beta = -0.5$	4	6	1	0	5	0	1	1	0
JFS with $\beta = 0.5$	3	3	2	1	2	0	0	2	0

TABLE 8 Number of solutions found in each proof

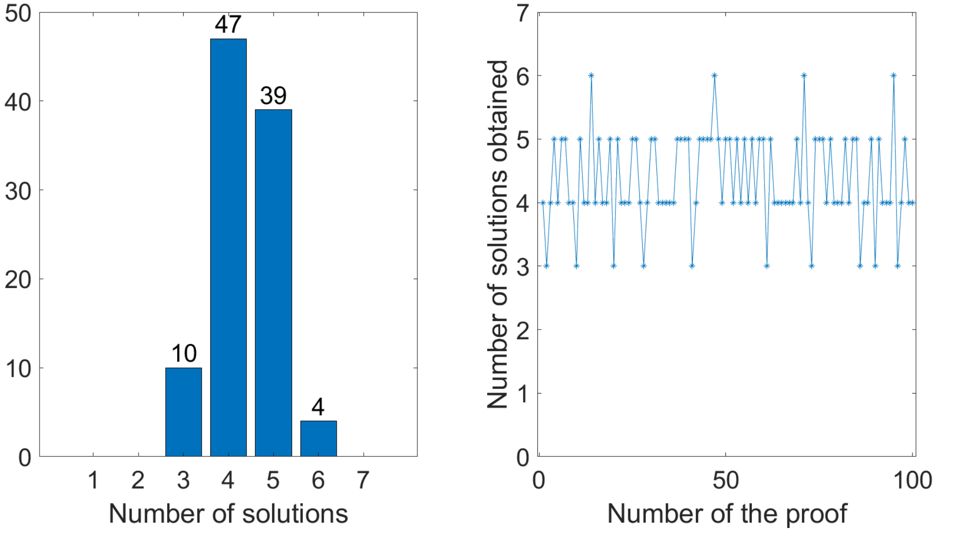
Method	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Total	Media
PS	4	5	5	3	3	4	3	3	3	3	36	3.6
JFS with $\beta = 0.01$	4	6	4	6	4	6	4	6	4	6	50	5
JFS with $\beta = 0.1$	5	6	4	4	5	5	5	4	5	4	47	4.7
JFS with $\beta = -0.1$	4	4	3	4	4	6	4	4	4	4	41	4.1
JFS with $\beta = -0.01$	3	4	4	3	3	5	3	3	3	3	34	3.4
JFS with $\beta = -0.5$	1	3	3	2	3	2	1	2	0	1	18	1.8
JFS with $\beta = 0.5$	3	3	1	0	0	2	3	1	0	0	13	1.3

As is illustrated in both tables, 7 and 8, for the parameter values $\beta = 0.01$ and $\beta = 0.1$ a larger number of roots have been found than for the PS method, while for the methods where the parameter satisfies $|\beta| = 0.5$ the number of mean roots to converge is less than 2. This confirms again that depending on the parameter the convergence zones are larger or smaller, and that for certain parameter values the derivative-free methods converge on a larger set of initial estimates.

The above data correspond to a total of 10 tests where random vectors are generated as initial estimates. In the following, we illustrate what happens for a total of 100 trials where the same process is followed, that is, in each of trial 9 vectors with random components are generated. The data illustrated are the number of solutions reached by each method in each trial.

In Figure 1, we have two images, the first one is a histogram with the number of times that the iterative method PS finds the respective number of solutions, while the second image is a graph where the abscissa axis represents the test or proof we are representing and the ordinate axis represents the number of solutions found for this test with this iterative method.

FIGURE 1 Method PS



In Figures 2, 3, 4, 5, 6 and 7, is represented the same as in Figure 1 but for the iterative methods JFS where we change the value of the parameter to be $\beta = 0.1$, $\beta = 0.01$, $\beta = 0.5$, $\beta = -0.1$, $\beta = -0.01$ and $\beta = -0.5$, respectively.

FIGURE 2 Method JFS with $\beta = 0.1$

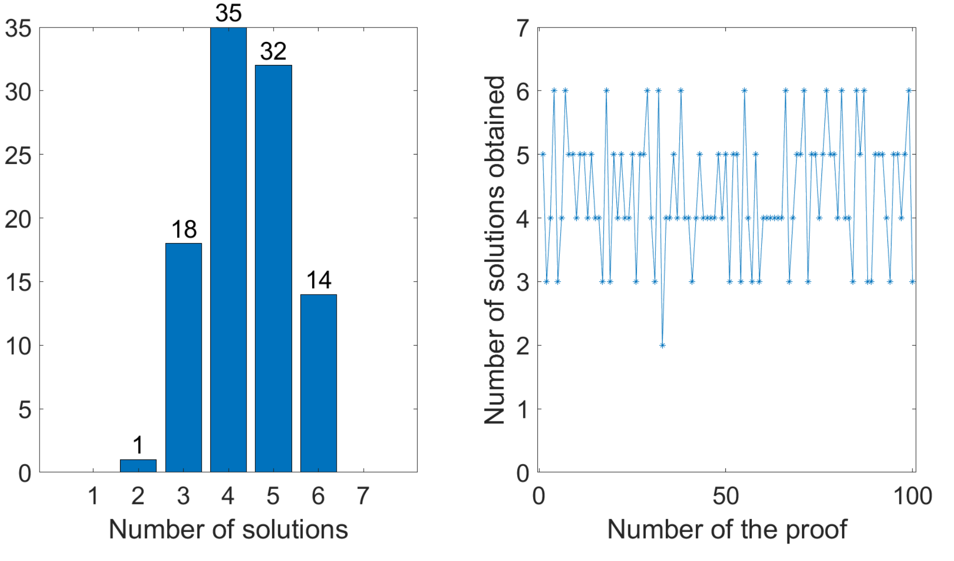


FIGURE 3 Method JFS with $\beta = 0.01$

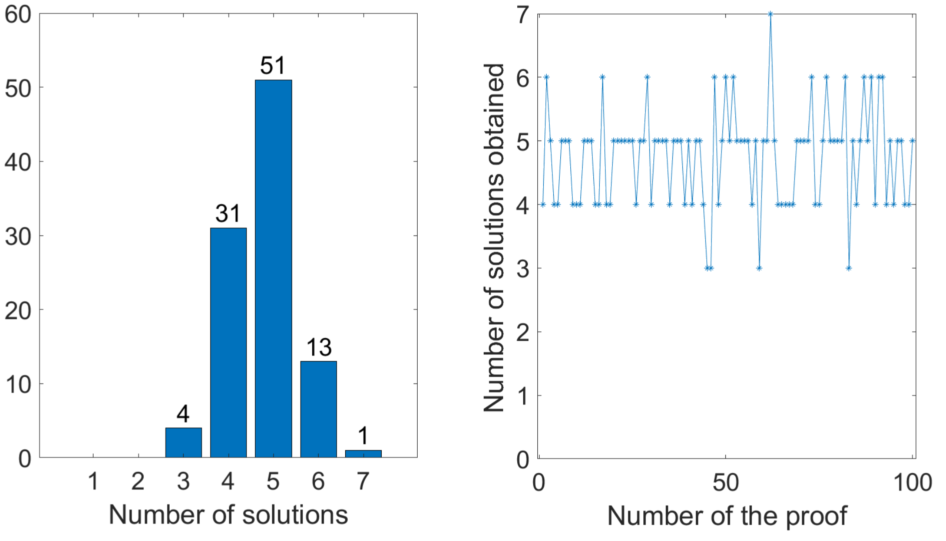


FIGURE 4 MethodJFS with $\beta = 0.5$

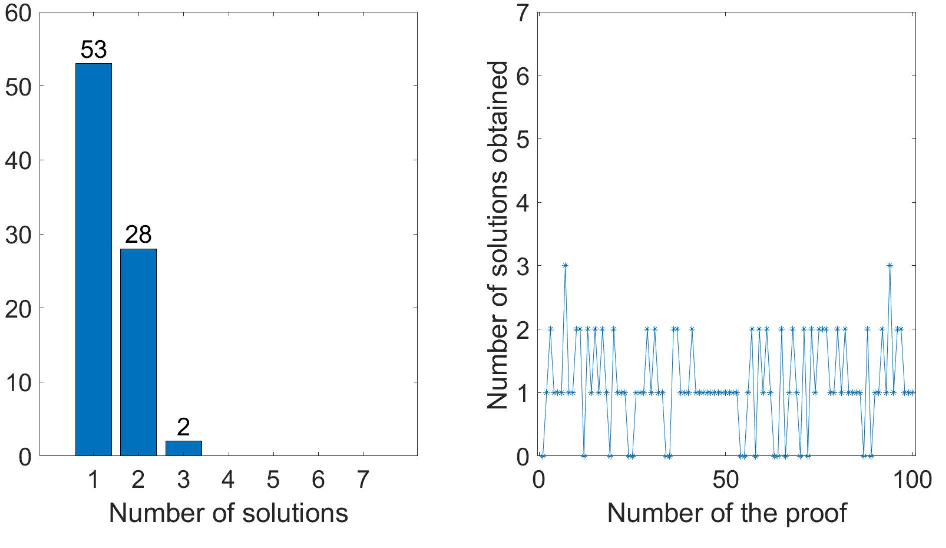


FIGURE 5 Method JFS with $\beta = -0.1$

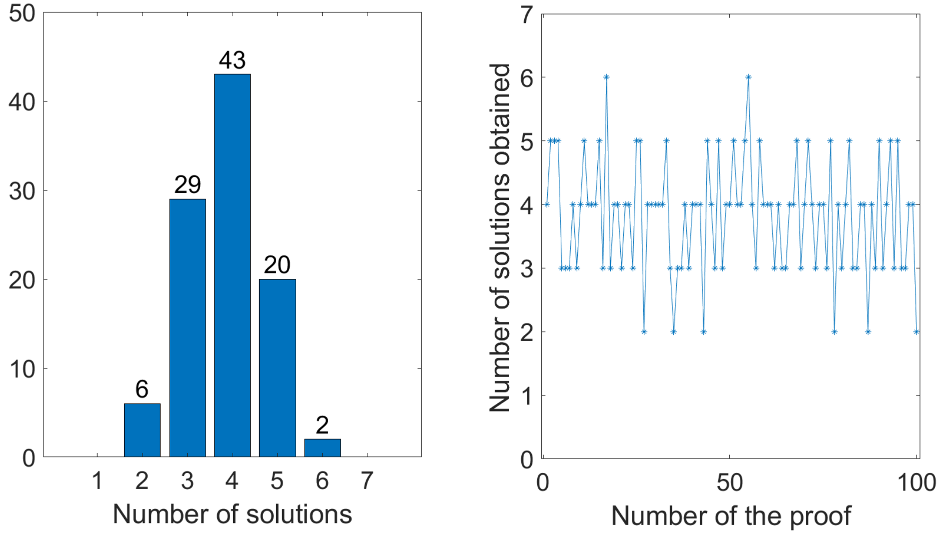


FIGURE 6 Method JFS with $\beta = -0.01$

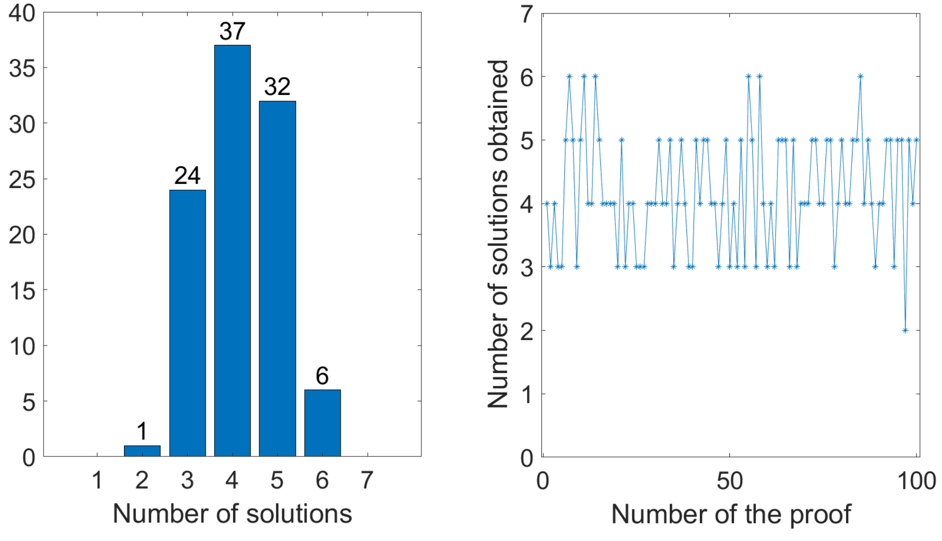
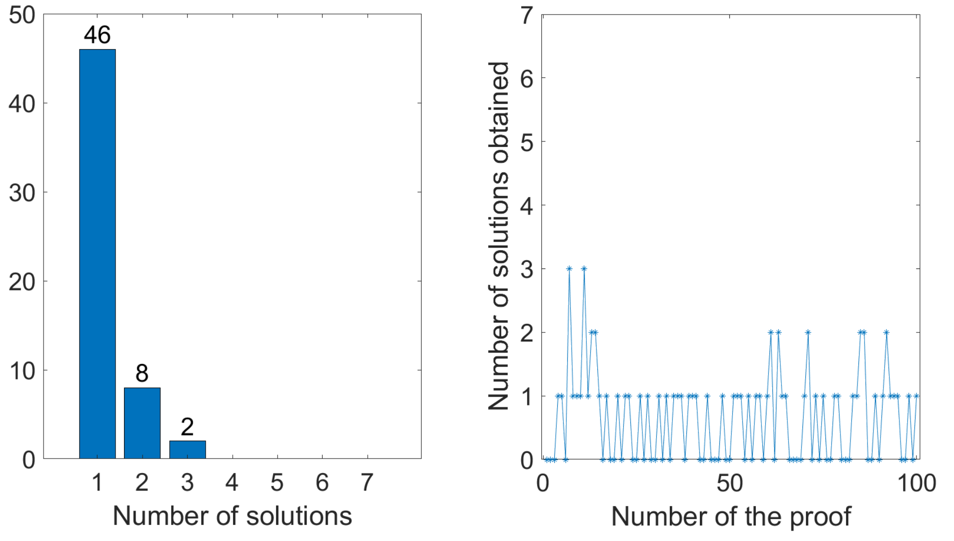


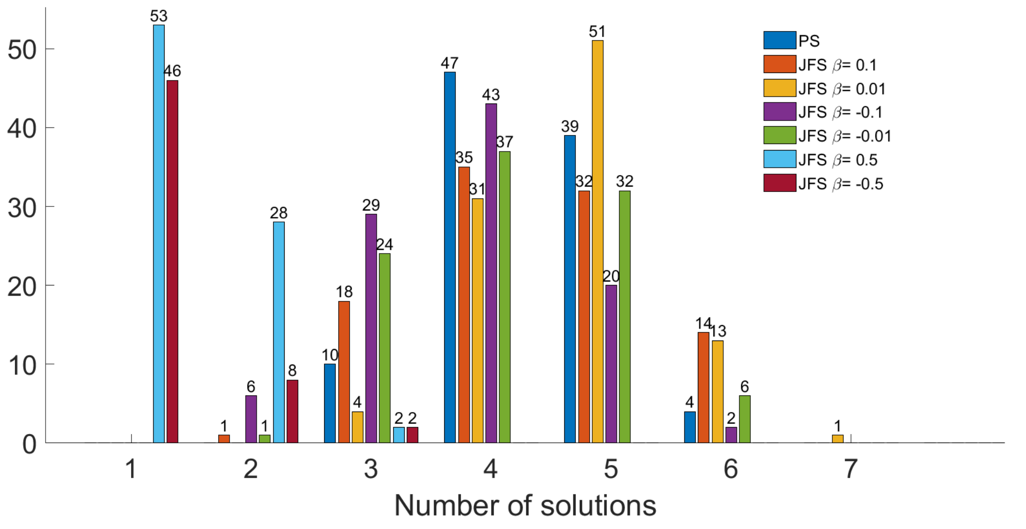
FIGURE 7 Method JFS with $\beta = -0.5$



In Figures 1, 2, 3, 4, 5, 6 and 7, are illustrated similar results to those obtained when 10 tests were performed, showing the need to change the parameter to obtain better results.

Finally, in Figure 8, we represent a histogram with the number of times such a number of roots found is reached for the different methods represented in the image legend. This figure shows that smaller values of the parameter obtain better solutions than the method with Jacobian matrices and the methods with larger values of the parameter. We highlight as methods that obtain good results those that use the parameters 0.01 and 0.1.

FIGURE 8 Histogram number of solutions found



2.7 | Dynamical behaviour example

Next, in order to compare the dynamics of the iterative DFS method with the Ehrlich method, we compare the dynamical planes obtained for the problem $z^2 - 1 = 0$ by both methods.

To create these dynamical planes, the first step is to perform a mesh of 500×500 points, where each point of the mesh corresponds to an initial vector, in which we represent on the abscissa axis the first component of the vector, while on the ordinate axis we represent the second component of the vector of initial guesses.

We represent in yellow the initial guess vector if it converges to $(1, -1)$, in purple when it converges to $(-1, 1)$, in blue when there is no convergence in less than 100 iterations and in black when one of the components is higher than 10^3.

We define the vector of initial estimates to converge whenever the distance of each of the components is less than the prefixed tolerance, which in this case is 10^{-3}.

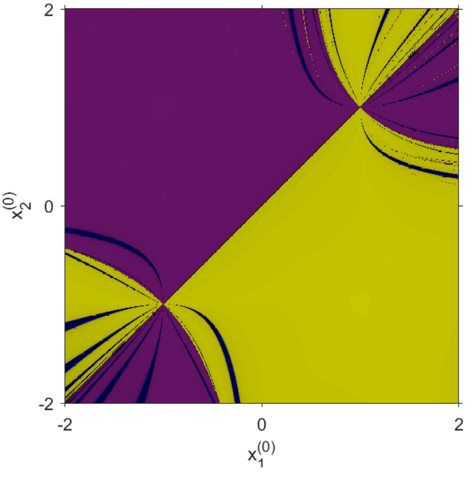


FIGURE 9 DFS with $\beta = 0.1$

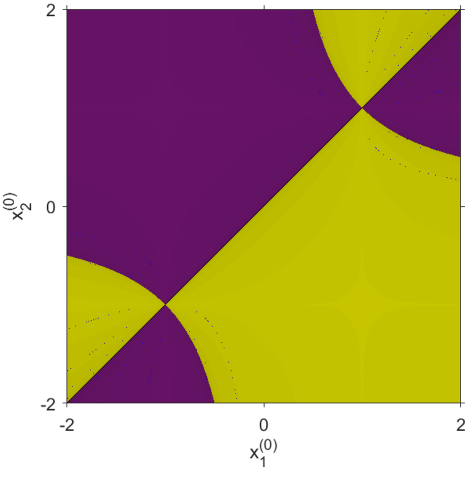


FIGURE 10 Ehrlich

As can be seen in Figures 9, 10, the difference between both dynamical planes is that the plane obtained by the DFS method has a greater number of points that do not converge in 100 iterations, but if we increase the number of iterations that can be performed we would obtain the same planes.

If we increase the area in which the meshing takes place, we can find more differences as we will see below.

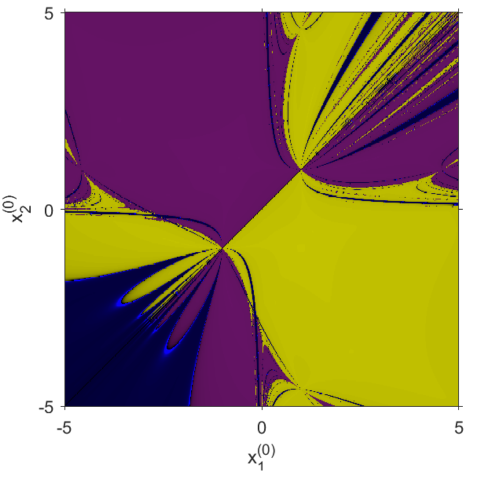


FIGURE 11 DFS with $\beta = 0.1$

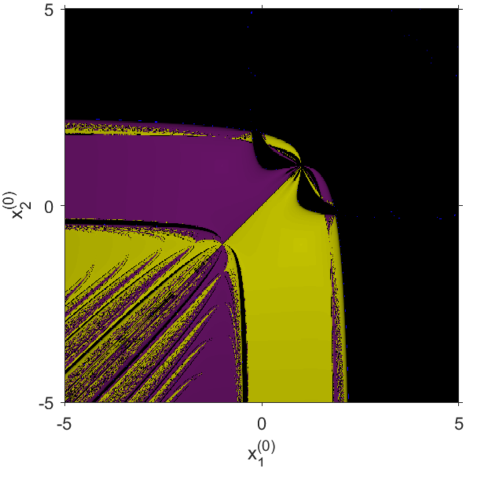


FIGURE 12 DFS with $\beta = -1$

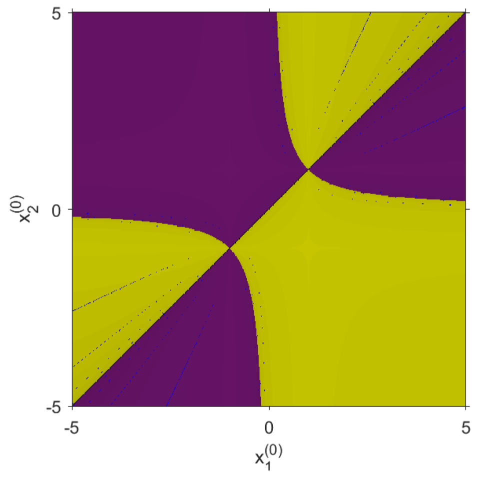


FIGURE 13 Ehrlich

In Figures 11, 12, 13, we can see that the non-Ehrlich methods have fewer convergence points. In addition, it can be seen that with higher parameter values, there is a larger non convergence zone, so that, as was deduced in the other numerical experiments, smaller parameter values tend to give better results.

3 | OUTCOMES AND CONCLUSIONS

In this manuscript, an algorithm for obtaining simultaneous solutions which is Jacobian-free has been defined. This scheme has order 2 and can obtain several solutions simultaneously for nonlinear equations and systems. Also, this step can be added to any procedure for solving systems of nonlinear equations such that the resulting method obtains several roots simultaneously and has duplicates the order of convergence of the original scheme.

This iterative step defined is a modification of the iterative method proposed in [4] which is not Jacobian-free. In this article, several experiments were carried out to compare both methods in order to see the properties and behaviour of those iterative procedures proposed.

Conflict of interest

The authors declare no potential conflict of interests.

References

[1] D. Budzko, A. Cordero, and J. R. Torregrosa. "A new family of iterative methods widening areas of convergence". In: *Applied Mathematics and Computation* 252 (2015), pp. 405–417. doi: https://doi.org/10.1016/j.amc.2014.12.028.

[2] D. Budzko, J. L. Hueso, E. Martínez, and C. Teruel. “Dynamical study while searching equilibrium solutions in N-body problem”. In: *Journal of Computational and Applied Mathematics* 297 (2016), pp. 26–40. doi: `https://doi.org/10.1016/j.cam.2015.11.010`.

[3] F.I. Chicharro, A. Cordero, N. Garrido, and J.R. Torregrosa. “On the improvement of the order of convergence of iterative methods for solving nonlinear systems by means of memory”. In: *Appl. Math. Letters* 104 (2020), p. 106277. doi: `https://doi.org/10.1016/j.aml.2020.106277`.

[4] F. Chinesta, A. Cordero, N. Garrido, J.R. Torregrosa, and P. Triguero-Navarro. “Simultaneous roots for vectorial problems”. In: *Computational and Applied Mathematics* 42 (2023). doi: `https://doi.org/10.1007/s40314-023-02366-y5`.

[5] A. Cordero, N. Garrido, J.R. Torregrosa, and P. Triguero-Navarro. “An iterative scheme to obtain multiple solutions simultaneously”. In: *Applied Mathematics Letters* 145 (2023), p. 108738. issn: 0893-9659. doi: `https://doi.org/10.1016/j.aml.2023.108738`.

[6] A. Cordero, N. Garrido, J.R. Torregrosa, and P. Triguero-Navarro. “Iterative schemes for finding all roots simultaneously of nonlinear equations”. In: *Applied Mathematics Letters* 134 (2022), p. 108325. doi: `https://doi.org/10.1016/j.aml.2022.108325`.

[7] A. Cordero and J. R. Torregrosa. “Variants of Newton’s method using fifth-order quadrature formulas.” In: *Appl. Math. Comput.* 190 (2007), pp. 686–698. doi: `10.1016/j.amc.2007.01.062`.

[8] L.W. Ehrlich. “A modified Newton method for polinomials”. In: *Commum. ACM* 10 (1967), pp. 107–108.

[9] C. Grosan and A. Abraham. “Multiple Solutions for a System of Nonlinear Equations”. In: *International Journal of Innovative Computing, Information and Control* 4 (Sept. 2007).

[10] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Society for Industrial and Applied Mathematics, 1970.

[11] I. Petković, S. Ilić, and S. Tričković. “A family of simultaneous methods zero-finding methods”. In: *Computers & Mathematics with Applications* 34 (1997), pp. 49–59. doi: `https://doi.org/10.1016/S0898-1221(97)00206-X`.

[12] M. Petković, L.D. Petković, and J. Dzunic. “On an efficient simultaneous method for finding polynomial zeros”. In: *Appl. Math. Letters* 28 (Feb. 2014), pp. 60–65. doi: `10.1016/j.aml.2013.09.011`.

[13] P. Proinov. “On the local convergence of Ehrlich method for numerical computation of polynomial zeros”. In: *Calcolo* 53 (Sept. 2015). doi: `10.1007/s10092-015-0155-y`.

[14] P. Proinov and S. Ivanov. “Convergence analysis of Sakurai-Torii-Sugiura iterative method for simultaneous approximation of polynomial zeros”. In: *Comput. Appl. Math.* 357 (2019), pp. 56–70. doi: `https://doi.org/10.1016/j.cam.2019.02.021`.

[15] P. Proinov and M. Vassileva. “On a family of Weirstrass-type root-finding methods with accelerated convergence”. In: *Appl. Math. Comput.* 273 (2016), pp. 957–968. doi: `https://doi.org/10.1016/j.amc.2015.10.048`.

[16] P. Proinov and M. Vassileva. “On the convergence of high-order Gargantini-Farmer-Loizon type iterative methods for simultaneous approximation of polynomial zeros”. In: *Appl. Math. Comput.* 361 (2019), pp. 202–214. doi: `https://doi.org/10.1016/j.amc.2019.05.026`.

[17] M. Shams et al. "On iterative techniques for estimating all roots of nonlinear equation and its system with application in differential equation". In: *Advances in Difference Equations* 2021 (Nov. 2021), pp. 480–498. doi: 10.1186/s13662-021-03636-x.