Design and multidimensional extension of iterative methods for solving nonlinear problems ☆

S. Artidiello[a], Alicia Cordero[b], Juan R. Torregrosa[b,*], M. P. Vassileva[a]

[a]*Instituto Tecnológico de Santo Domingo (INTEC), Santo Domingo, Dominican Republic*
[b]*Instituto de Matemáticas Multidisciplinar, Universitat Politècnica de València, València, Spain*

Abstract

In this paper, a three-step iterative method with sixth-order local convergence for approximating the solution of a nonlinear system is presented. From Ostrowski's scheme adding one step of Newton with 'frozen' derivative and by using a divided difference operator we construct an iterative scheme of order six for solving nonlinear systems. The computational efficiency of the new method is compared with some known ones, obtaining good conclusions. Numerical comparisons are made with other existing methods, on standard nonlinear systems and the classical 1D-Bratu problem by transforming it in a nonlinear system by using finite differences. From this numerical examples, we confirm the theoretical results and show the performance of the presented scheme.

Keywords: Nonlinear systems, iterative method, convergence, efficiency index, Bratu's problem.

1. Introduction

The design of fixed point iterative methods for solving equations and systems of nonlinear equations is an important and challenging task in the field of numerical analysis. In 1990, Moré [18] proposed a collection of nonlinear problems and most of them are phrased in terms of $F(x) = 0$. On the other hand, Grosan and Abraham [10] also discussed the applicability of the system of nonlinear equations in neurophysiology, kinematics syntheses problem, chemical equilibrium problem, combustion problem and economics modeling problem. In addition, the reactor and steering problems are solved in [2, 23] by phrasing these problems in the form of $F(x) = 0$. Moreover, Lin et al. [15] also discussed the applicability of the systems of nonlinear equations in transport theory. These and other more examples allow us to affirm that finding the solution ξ of a nonlinear system $F(x) = 0$ is a classical and difficult problem with many applications in science and engineering, wherein $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ is a sufficiently Frechet differentiable function in an open convex set D. Since 1980s', many iterative methods have been constructed for solving nonlinear systems, see, for example, [2, 6, 10, 12, 17, 22, 25] and the references therein. The best known method for finding a solution $\xi \in D$ is Newton's scheme,

$$x^{(k+1)} = x^{(k)} - [F'(x^{(k)})]^{-1}F(x^{(k)}),\ k = 0, 1, 2, \dots,$$

where $F'(x^{(k)})$ is the Jacobian matrix of function F evaluated at the kth iteration.

The proliferation of iterative methods for solving nonlinear equations has been spectacular in the last years (in [21] we can see a good overview). Some of these methods can be transferred easily to the context of nonlinear systems, keeping the order of convergence, but others, at least apparently, cannot be extended to multidimensional case. In this paper, we use a procedure that allows this extension for many known multi-point iterative methods designed for solving nonlinear equations, making use

☆This research was partially supported by Ministerio de Economía y Competitividad MTM2014-52016-C2-2-P and FONDOCYT 2014-1C1-088 República Dominicana.

*Corresponding author

Email addresses: santiago.artidiello@intec.edu.do (S. Artidiello), acordero@mat.upv.es (Alicia Cordero), jrtorre@mat.upv.es (Juan R. Torregrosa), maria.penkova@intec.edu.do (M. P. Vassileva)

of tools such as the divided difference operator. Of course, this translation only has interest if the order of convergence is preserved, as it is the case.

Based on Newton's or Newton-like iterations, some higher order methods for computing a solution of nonlinear system $F(x) = 0$ have been proposed in the literature. The aim of these new schemes is to accelerate the convergence or to improve the computational efficiency. For example, among other authors, Montazeri et al. [17] and Hueso et al. [12], developed sixth-order iterative methods requiring two evaluations of function F and two of Jacobian F' per iteration, Sharma and Arora [22] designed a sixth-order method which requires three functional and two Jacobian evaluations per iteration. On the other hand, Wang et al. [25] have constructed a seventh-order derivative free iterative method by using the first order divided difference operator $[x, y; F]$ evaluated three times per iteration.

Specifically, the sixth-order scheme by Montazeri et al. [17], that we denote by MSSM, has three-step and its iterative expression is

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{2}{3}[F'(x^{(k)})]^{-1}F(x^{(k)}), \\ z^{(k)} &= x^{(k)} - \left[\frac{23}{8}I - 3[F'(x^{(k)})]^{-1}F'(y^{(k)}) + \frac{9}{8}\left([F'(x^{(k)})]^{-1}F'(y^{(k)})\right)^2\right][F'(x^{(k)})]^{-1}F(x^{(k)}), \\ x^{(k+1)} &= z^{(k)} - \left[\frac{5}{2}I - \frac{3}{2}[F'(x^{(k)})]^{-1}F'(y^{(k)})\right][F'(x^{(k)})]^{-1}F(z^{(k)}), \end{aligned} \tag{1}$$

where I is the identity matrix of size $n \times n$.

Hueso et al. in [12] developed several iterative schemes of order six; we use some of them in the numerical section for comparing with our proposed scheme on different test problems. In particular, in [12] the authors present the following method that we denote by HMT1

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{2}{3}[F'(x^{(k)})]^{-1}F(x^{(k)}), \\ z^{(k)} &= x^{(k)} - \left[-\frac{1}{2}I + \frac{9}{8}[F'(y^{(k)})]^{-1}F'(x^{(k)}) + \frac{3}{8}[F'(x^{(k)})]^{-1}F'(y^{(k)})\right][F'(x^{(k)})]^{-1}F(x^{(k)}), \\ x^{(k+1)} &= z^{(k)} - \left[-\frac{9}{4}I + \frac{15}{8}[F'(y^{(k)})]^{-1}F'(x^{(k)}) + \frac{11}{8}[F'(x^{(k)})]^{-1}F'(y^{(k)})\right][F'(y^{(k)})]^{-1}F(z^{(k)}) \end{aligned} \tag{2}$$

and the three-step scheme, denoted by HMT2

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{2}{3}[F'(x^{(k)})]^{-1}F(x^{(k)}), \\ z^{(k)} &= x^{(k)} - \left[\frac{5}{8}I + \frac{3}{8}\left([F'(y^{(k)})]^{-1}F'(x^{(k)})\right)^2\right][F'(x^{(k)})]^{-1}F(x^{(k)}), \\ x^{(k+1)} &= z^{(k)} - \left[-\frac{9}{4}I + \frac{15}{8}[F'(y^{(k)})]^{-1}F'(x^{(k)}) + \frac{11}{8}[F'(x^{(k)})]^{-1}F'(y^{(k)})\right][F'(y^{(k)})]^{-1}F(z^{(k)}). \end{aligned} \tag{3}$$

On the other hand, Wang et al. in [25] describe the following derivative-free seventh-order scheme, that we denote by WZQT

$$\begin{aligned} y^{(k)} &= x^{(k)} - B^{-1}F(x^{(k)}), \\ z^{(k)} &= y^{(k)} - \left[3I - 2B^{-1}[y^{(k)}, x^{(k)}; F]\right]B^{-1}F(y^{(k)}), \\ x^{(k+1)} &= z^{(k)} - \left[\frac{13}{4}I - B^{-1}[z^{(k)}, y^{(k)}; F]\left(\frac{7}{2}I - \frac{5}{4}B^{-1}[z^{(k)}, y^{(k)}; F]\right)\right]B^{-1}F(z^{(k)}), \end{aligned} \tag{4}$$

where $B = \left[x^{(k)} + F(x^{(k)}), x^{(k)} - F(x^{(k)}); F\right]$ and $[x, y; F] : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ is the first order divided difference on D.

In order to compare the different methods under the point of view of the computational cost, we recall the computational efficiency index, CI, introduced by the authors in [6], which combine the efficiency index defined by Ostrowski [20] and the number of products-quotients required per iteration. We define this index as $CI = p^{1/(d+op)}$, where p is the order of convergence, d is the number of functional evaluations and op is the number of products-quotients per iteration. Let us remark that for evaluating function F we need n scalar functional evaluations (the coordinate functions of F), whilst for evaluating Jacobian F' it is necessary to evaluate n^2 functions (all the entries of matrix F'). On the other hand, all the iterative methods for solving nonlinear systems require one or more matrix inversion, that is, one or more linear systems must be solved. So, the number of operations needed for solving a linear system plays in this context an important role.

We recall that the number of products and quotients required for solving a linear system by Gaussian elimination is $\frac{1}{3}n^3 + n^2 - \frac{1}{3}n$, where n is the size of the system. In addition, for solving q linear systems, with the same matrix of coefficients, by using LU decomposition we need $\frac{1}{3}n^3 + qn^2 - \frac{1}{3}n$ products-quotients. By using this information, in Section 3 we compare the computational efficiency index of the different methods used in this manuscript.

The main objective of this paper is to develop a high-order iterative method in such a way that it utilizes as lower number of functional evaluations as possible, as well as it has good stability properties. From the known Ostrowski's method, generalized to nonlinear systems, by adding a new step that keeps the number of matrix inversions, we design a three-step sixth-order iterative method which combines the Jacobian matrix with the first order divided difference operator.

We summarize the contents of this paper. In Section 2, we describe a procedure to extend many iterative methods designed for scalar equations to nonlinear systems. This procedure allows us to construct a three-step sixth-order iterative method. The computational efficiency index of our method and of other known ones is analyzed in Section 3. Section 4 is devoted to solve some academic problems and the nonlinear system obtained from the discretization of the planar 1D-Bratu problem, by using the finite difference method. We compare the numerical results that our scheme gives with those obtained by other known methods with the same or higher order of convergence. With the references used, the paper is finished.

2. Development and convergence of the method

For extending scalar iterative schemes to the multidimensional case, we follow the procedure described in [1, 5, 8] for constructing our proposal, which has the following iterative expression

$$\begin{aligned} y^{(k)} &= x^{(k)} - [F'(x^{(k)})]^{-1}F(x^{(k)}), \\ z^{(k)} &= y^{(k)} - \left[2[x^{(k)}, y^{(k)}; F] - F'(x^{(k)})\right]^{-1} F(y^{(k)}), \\ x^{(k+1)} &= z^{(k)} - \left(3I - 2[F'(x^{(k)})]^{-1}[x^{(k)}, y^{(k)}; F]\right) [F'(x^{(k)})]^{-1}F(z^{(k)}), \end{aligned} \tag{5}$$

where I denotes the identity matrix of size $n \times n$. The firsts two steps correspond to Ostrowski's method for nonlinear systems. With the third step we increase the order of convergence without introducing new matrix inversions. The divided difference $[x, y; F]$ is the map $[\cdot, \cdot; F] : D \times D \subset \mathbb{R}^n \times \mathbb{R}^n \to \mathcal{L}(\mathbb{R}^n)$ defined by Ortega and Rheinbold in [19] satisfying $[x, y; F](x - y) = F(x) - F(y), \forall x, y \in D$.

This method has been obtained as a particular case of the following family of parametric schemes

$$x^{(k+1)} = z^{(k)} - (\alpha + \beta u + \gamma v)[F'(x^{(k)})]^{-1}F(z^{(k)}),$$

where $u = I - [F'(x^{(k)})]^{-1}[x^{(k)}, y^{(k)}; F]$, $v = [x^{(k)}, y^{(k)}; F]^{-1}F'(x^{(k)})$ and $y^{(k)}$ and $z^{(k)}$ are those defined in (5). It can be proved that this family has order of convergence six for every value of the parameters. For obtaining method (5) we have chosen $\alpha = 1$, $\beta = 2$ and $\gamma = 0$.

In the study of convergence of this scheme, we are going to use the tools and procedure introduced in [6]. Since the local convergence analysis that we are going to carry out in our method is based on the Taylor series expansion around the solution, we need to obtain the corresponding development of the divided difference operator. To achieve this, we use the Genocchi-Hermite formula (see [11])

$$[x, x + h; F] = \int_0^1 F'(x + th)dt$$

and, by developing $F'(x+th)$ in Taylor series around x, we obtain

$$\int_0^1 F'(x+th)dt = F'(x) + \frac{1}{2}F''(x)h + \frac{1}{6}F'''(x)h^2 + \mathcal{O}(h^3). \tag{6}$$

Denoting by $e = x - \xi$ and assuming that $F'(\xi)$ is nonsingular, we have:

$$\begin{aligned} F(x) &= F'(\xi)(e + C_2e^2 + C_3e^3 + C_4e^4 + C_5e^5) + O(e^6), \\ F'(x) &= F'(\xi)(I + 2C_2e + 3C_3e^2 + 4C_4e^3 + 5C_5e^4 +) + O(e^5), \\ F''(x) &= F'(\xi)(2C_2 + 6C_3e + 12C_4e^2) + O(e^3), \\ F'''(x) &= F'(\xi)(6C_3 + 24C_4e) + O(e^2), \end{aligned} \tag{7}$$

where $C_q = \dfrac{1}{q!}[F'(\xi)]^{-1}F^{(q)}(\xi)$, $q \geq 2$. Replacing these developments in the formula of Genocchi-Hermite and denoting the second point of the divided difference by $y = x + h$ and the error at the first step by $e_y = y - \xi$, we have

$$[x, y; F] = F'(\xi)[I + C_2(e_y + e) + C_3e^2] + O(e^3).$$

In particular, if y is an approximation of the solution provided by the Newton's method, i.e. $h = x - y = [F'(x)]^{-1}F(x)$, we obtain

$$[x, y; F] = F'(\xi)[I + C_2e + (C_2^2 + C_3)e^2] + O(e^3).$$

We denote by $e^{(k)} = x^{(k)} - \xi$ the error in the kth iteration and we call the relation

$$e^{(k+1)} = L(e^{(k)})^p + O(e^{(k)^{p+1}})$$

error equation. Here, p is the order of convergence, L is a p-linear function, that is, $L \in \mathcal{L}(\mathbb{R}^n \times \mathbb{R}^n \cdots \times \mathbb{R}^n, \mathbb{R}^n)$, $\mathcal{L}$ denotes the set of bounded linear functions.

Theorem 1. *Let $F : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ be a sufficiently differentiable function in an open convex set D and $\xi \in D$ a solution of the nonlinear system $F(x) = 0$. Let us suppose that $F'(x)$ is continuous and nonsingular at ξ and $x^{(0)}$ is an initial guess close enough to ξ. Then, sequence $\{x^{(k)}\}_{k\geq 0}$ obtained by using expression (5) converges to ξ with order of convergence six, being its error equation*

$$e^{(k+1)} = \left[6C_2^5 - 7C_2^3C_3 + C_2C_3^2\right] e^{(k)^6} + O(e^{(k)^7}),$$

where $C_k = \dfrac{1}{k!}[F'(\xi)]^{-1}F^{(k)}(\xi)$, $k = 2, 3, \ldots$

Proof. By using Taylor expansion of $F(x^{(k)})$ and $F'(x^{(k)})$ around ξ, we obtain

$$\begin{aligned} F(x^{(k)}) &= F'(\xi)\left[e^{(k)} + C_2e^{(k)^2} + C_3e^{(k)^3} + C_4e^{(k)^4} + C_5e^{(k)^5} + C_6e^{(k)^6}\right] + O(e^{(k)^7}), \\ F'(x^{(k)}) &= F'(\xi)\left[I + 2C_2e^{(k)} + 3C_3e^{(k)^2} + 4C_4e^{(k)^3} + 5C_5e^{(k)^4} + 6C_6e^{(k)^5}\right] + O(e^{(k)^6}), \end{aligned}$$

and

$$[F'(x^{(k)})]^{-1} = \left[I + X_2e^{(k)} + X_3e^{(k)^2} + X_4e^{(k)^3} + X_5e^{(k)^4} + X_6e^{(k)^5}\right][F'(\xi)]^{-1} + O(e^{(k)^6}), \tag{8}$$

where $X_1 = I$ and $X_m = -\displaystyle\sum_{j=2}^{m} jX_{m-j+1}C_j$, $m = 2, 3, \ldots$. The values of X_m have been obtained from the identity $[F'(x^{(k)})]^{-1}F(x^{(k)}) = I$. Substituting these results in the first step of (5) we get

$$y^{(k)} = \xi - \left[A_2e^{(k)^2} + A_3e^{(k)^3} + A_4e^{(k)^4} + A_5e^{(k)^5}\right] + O(e^{(k)^6}),$$

with $A_2 = -C_2$, $A_s = C_s + \sum_{j=3}^{s} X_{s-j+2}C_{j-1} + X_s$, $s = 3, 4, ...$

Using again Taylor series for developing $F(y^{(k)})$ around ξ, we have

$$F(y^{(k)}) = F'(\xi)\left[B_2 e^{(k)^2} + B_3 e^{(k)^3} + B_4 e^{(k)^4} + B_5 e^{(k)^5}\right] + O(e^{(k)^6}),$$

where $B_2 = -A_2$, $B_3 = -A_3$, $B_4 = -A_4 + C_2A_2^2$, $B_5 = -A_5 + C_2(A_2A_4 + A_3A_2)$ and $B_6 = -A_6 + C_2(A_2A_4 + A_3^2 + A_4A_2) - C_3A_2^3$.

For analyzing the order of the second step of (5), we need the Taylor series expansion around ξ of the divided difference operator $[x^{(k)}, y^{(k)}; F]$. By using expressions (6) and (7), we obtain

$$\begin{aligned}[x^{(k)}, y^{(k)}; F] &= F'(x^{(k)}) + \frac{1}{2!}F''(x^{(k)})h + \frac{1}{3!}F'''(x^{(k)})h^2 + O(h^3) \\ &= F'(\xi)[I + P_1e^{(k)} + P_2e^{(k)^2} + P_3e^{(k)^3} + P_4e^{(k)^4} + P_5e^{(k)^5} + P_6e^{(k)^6}] + O(e^{(k)^7}),\end{aligned}$$

where $h = x^{(k)} - y^{(k)} = [F'(x^{(k)})]^{-1}F(x^{(k)})$, $P_1 = C_2$, $P_2 = -C_2A_2 + C_3$, $P_3 = 2C_4 - 2C_2A_3 - 3C_3A_2 + C_3D_3$, $P_4 = 5C_5 - C_2A_4 - 3C_3A_3 - 6C_4A_2 + C_3D_4 + 4C_4D_3$, $P_5 = 11C_6 - C_2A_5 - 3C_3A_4 - 6C_4A_3 - 10C_5A_2 + C_3D_5 + 4C_4D_4 + 10C_5D_3$, with $D_3 = 2A_2$, $D_4 = A_2^2 + 2A_3$, $D_5 = A_2A_3 + A_3A_2 + 2A_4$ and $D_6 = A_2A_4 + A_3^2 + A_4A_2 + 2A_5$.

Now, we denote $M = 2[x^{(k)}, y^{(k)}; F] - F'(x^{(k)})$ and

$$M = F'(\xi)\left[I + M_1e^{(k)} + M_2e^{(k)^2} + M_3e^{(k)^3} + M_4e^{(k)^4} + M_5e^{(k)^5} + M_6e^{(k)^6}\right] + O(e^{(k)^7}), \tag{9}$$

where $M_1 = 2P_1 - 2C_2$, $M_2 = 2P_2 - 3C_3$, $M_3 = 2P_3 - 4C_4$, $M_4 = 2P_4 - 5C_5$, $M_5 = 2P_5 - 6C_6$ and $M_6 = 2P_6 - 7C_7$.

For obtaining the expression of the inverse of M, we force it to satisfy

$$\begin{aligned}I = M^{-1}M = &\left[I + Y_1e^{(k)} + Y_2e^{(k)^2} + Y_3e^{(k)^3} + Y_4e^{(k)^4} + Y_5e^{(k)^5} + Y_6e^{(k)^6}\right][F'(\xi)]^{-1}F'(\xi) \\ &\left[I + M_1e^{(k)} + M_2e^{(k)^2} + M_3e^{(k)^3} + M_4e^{(k)^4} + M_5e^{(k)^5} + M_6e^{(k)^6}\right] + O(e^{(k)^7})\end{aligned}$$

obtaining $Y_1 = -M_1$, and $Y_s = -M_s - \sum_{j=1}^{s-1} Y_jM_{s-j}$, $s = 2, 3, \ldots$

Then,

$$L = M^{-1}F(y^{(k)}) = L_2e^{(k)^2} + L_3e^{(k)^3} + L_4e^{(k)^4} + L_5e^{(k)^5} + L_6e^{(k)^6} + O(e^{(k)^7}),$$

where $L_2 = B_2$, $L_3 = B_3 + Y_1B_2$, $L_4 = B_4 + Y_1B_3 + Y_2B_2$, $L_5 = B_5 + Y_1B_4 + Y_2B_3 + Y_3B_2$ and $L_6 = B_6 + Y_1B_5 + Y_2B_4 + Y_3B_3 + Y_4B_2$.

Therefore,

$$z^{(k)} = y^{(k)} - L = \xi - \left[K_2e^{(k)^2} + K_3e^{(k)^3} + K_4e^{(k)^4} + K_5e^{(k)^5} + K_6e^{(k)^6}\right] + O(e^{(k)^7}), \tag{10}$$

where $K_2 = A_2 + L_2$, $K_3 = A_3 + L_3$, $K_4 = A_4 + L_4$, $K_5 = A_5 + L_5$ and $K_6 = A_6 + L_6$.

Analyzing the values of K_i, we obtain

$$\begin{aligned}K_2 &= A_2 + L_2 = -C_2 + B_2 = -C_2 + C_2 = 0, \\ K_3 &= A_3 + L_3 = A_3 - A_3 = 0, \\ K_4 &= A_4 + L_4 = A_4 - A_4 + C_2^3 - C_3C_2 = C_2^3 - C_3C_2,\end{aligned}$$

so

$$z^{(k)} = \xi - \left[K_4e^{(k)^4} + K_5e^{(k)^5} + K_6e^{(k)^6}\right] + O(e^{(k)^7}).$$

In a similar way as before, we have

$$F(z^{(k)}) = -F'(\xi)\left[K_4 e^{(k)^4} + K_5 e^{(k)^5} + K_6 e^{(k)^6}\right] + O(e^{(k)^7}). \tag{11}$$

By using (8) and (11) we get,

$$N = [F'(x^{(k)})]^{-1}F(z^{(k)}) = \left[N_4 e^{(k)^4} + N_5 e^{(k)^5} + N_6 e^{(k)^6}\right] + O(e^{(k)^7}), \tag{12}$$

being $N_4 = K_4$, $N_5 = X_2K_4 + K_5$ and $N_6 = X_2K_5 + X_3K_4 + K_6$.

Now, by using (8) and (9) we obtain

$$R = [F'(x^{(k)})]^{-1}[x^{(k)}, y^{(k)}; F] = \left[I + R_1 e^{(k)} + R_2 e^{(k)^2} + R_3 e^{(k)^3} + R_4 e^{(k)^4} + R_5 e^{(k)^5} + R_6 e^{(k)^6}\right] + O(e^{(k)^7}), \tag{13}$$

with $R_1 = P_1 + X_2$, $R_m = P_m + \displaystyle\sum_{j=2}^{m} X_j M_{m-j+1} + X_{m+1}$, $m = 2, 3, \ldots$

By using (12) and (13),

$$T = (3I - 2R)N = \left[T_4 e^{(k)^4} + T_5 e^{(k)^5} + T_6 e^{(k)^6}\right] + O(e^{(k)^7}), \tag{14}$$

where $T_4 = N_4$, $T_5 = N_5 + 2R_1N_4$ and $T_6 = N_6 - 2R_1N_5 - 2R_2N_4$.

Finally,

$$\begin{aligned} x^{(k+1)} &= z^{(k)} - T \\ &= \xi + \left[K_4 e^{(k)^4} + K_5 e^{(k)^5} + K_6 e^{(k)^6} - (T_4 e^{(k)^4} + T_5 e^{(k)^5} + T_6 e^{(k)^6})\right] + O(e^{(k)^7}) \\ &= \xi + (K_4 - T_4)e^{(k)^4} + (K_5 - T_5)e^{(k)^5} + (K_6 - T_6)e^{(k)^6} + O(e^{(k)^7}). \end{aligned}$$

Algebraic manipulations allow us to affirm that $K_4 - T_4 = 0$, $K_5 - T_5 = 0$ and the error equation has the form

$$e^{(k+1)} = \left[6C_2^5 - 7C_2^3C_3 + C_2C_3^2\right] e^{(k)^6} + O(e^{(k)^7})$$

and the proof is finished. ■

3. Computational efficiency

We are going to use of the definition of computational efficiency index CI, presented in Section 1, to compare the efficiency of the different methods. Let us recall that $CI = p^{1/(d+op)}$ where p is the order of convergence, d is the number of functional evaluations per iteration and op is the number of products-quotients per iteration. To compute F in any iterative method we need to calculate n scalar functions. The number of scalar functional evaluations is n^2 for any new derivative F'. When we compute a first-order divided difference $[x, y; F]$ we need $n(n-1)$ scalar functional evaluations, where $F(x)$ and $F(y)$ are computed separately.

On the other hand, in order to compute an inverse linear operator we solve a $n \times n$ linear system where we have to do $\dfrac{1}{3}n^3 + n^2 - \dfrac{1}{3}n$ products-quotients for obtaining LU decomposition and to solve two triangular linear systems. In addition, we need n^2 divisions for computing a first order divided difference and n^2 products for matrix-vector multiplication.

Taking into account the previous considerations, we give the index CI of method ACTV. For each iteration, we need to evaluate function F three times, once the Jacobian F' and once the divided difference operator, so $2n^2 + 2n$ functional evaluations are needed. In addition, we must to solve three linear systems with $F'(x^{(k)})$ as coefficients matrix (that is $(1/3)n^3 + 3n^2 - (1/3)n$ products-quotients), one linear system with $2[x^{(k)}, y^{(k)}, F] - F'(x^{(k)})$ as coefficients matrix (that is $(1/3)n^3 + n^2 - (1/3)n$ products-quotients), one product matrix-vector (n^2 products-quotients) and n^2 divisions in the construction of the divided difference operator. Therefore, the value of index CI for method ACTV on a nonlinear system of size $n \times n$ is

$$CI_{ACTV} = 6^{\frac{1}{\frac{2}{3}n^3+8n^2+\frac{4}{3}n}}.$$

In Table 1 we show index CI of the schemes ACTV, HMT1, HMT2, MSSM and WZQT. In it, NFE is the number of functional evaluations, $NLS1$ denotes the number of linear systems solved with the same matrix of coefficients, $NLS2$ is the number of linear systems solved with other matrix of coefficient, $M \times V$ denotes the number of products matrix-vector and DD the number of divided difference operators.

Let us observe that, for example, method WZQT needs to solve, per iteration, six linear systems with coefficients matrix B, three divided differences, that is $3n^2$ quotients, three products matrix-vector and $3n^2$ functional evaluations corresponding to three evaluations of F and three evaluations of the divided difference $[x, y; F]$.

In Figure 1, we can observe the value of index CI for the used methods and different sizes of the nonlinear system. Let us remark that, for $2 < n < 19$, the best index CI corresponds to method ACTV. From $n \geq 20$, the best CI belong to WZQT as it has seventh-order of convergence; the following best one is our proposed scheme.

4. Numerical results

To illustrate the convergence behavior of the proposed method ACTV we consider some numerical examples and compare the performance with the schemes described in the introduction and denoted by MSSM, HMT1, HMT2 and WZQT. All computations are performed in the programming package Matlab $R2013b$ using variable precision arithmetics with 200 digits of mantissa. The divided differences performed are of order 1. For every method, we analyze the number of iterations (iter) needed to converge to the solution such that $\|x^{(k+1)} - x^{(k)}\| < 10^{-100}$ or $\|F(x^{(k+1)})\| < 10^{-100}$ is satisfied, where $\|\cdot\|$ denotes the Euclidean norm. To verify the theoretical order of convergence (p), we calculate the computational order of convergence (ACOC) introduced in [7] as

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

Let us remark that, if the entries of vector ACOC do not stabilize their values along the iterative process, it is marked as '-'.

In Table 2, we show the numerical results obtained applying the described methods on seven academic nonlinear systems and in Tables from 3 to 7 the results given by these schemes on 1D Bratu's problem are presented. For the first ones, the involved functions and their corresponding roots are:

$$F_1(x_1, x_2) = (e^{x_1}e^{x_2} + x_1\cos(x_2), x_1 + x_2 - 1), \ \ \xi \approx (3.47063096, -2.47063096)^T,$$

$$F_2(x_1, x_2) = \left(\ln(x_1^2) - 2\ln(\cos(x_2)), x_1\tan\left(\frac{x_1}{\sqrt{2}} + x_2\right) - \sqrt{2}\right), \ \ \xi \approx (0.9548, 6.5850)^T,$$

$$F_3(x_1, x_2) = (x_1 + e^{x_2} - \cos(x_2), 3x_1 - x_2 - \sin(x_2)), \ \ \xi = (0, 0)^T,$$

$$F_4(x_1, x_2, x_3) = \left(\cos(x_2) - \sin(x_1), x_3^{x_1} - \frac{1}{x_2}, e^{x_1} - x_3^2\right), \ \ \xi \approx (0.9096, 0.6612, 1.5758)^T,$$

$F_5(x_1, x_2, x_3, x_4) = (x_2x_3 + x_4(x_2 + x_3), x_1x_3 + x_4(x_1 + x_3), x_1x_2 + x_4(x_1 + x_2), x_1x_2 + x_1x_3 + x_2x_3 - 1)$, whose solution is $\xi \approx (0.5774, 0.5774, 0.5774, -0.2847)^T$,

$F_6(x_1, x_2, \ldots, x_n)$ whose coordinate functions are $f_6^i = x_i^2x_{i+1} - 1$, for $i = 1, 2, \ldots, n-1$ and $f_6^n = x_n^2x_1 - 1$, being the root $\xi \approx (1, 1, \ldots, 1, 1)^T, n = 49$,

$F_7(x_1, x_2, x_3, x_4)$ whose coordinate functions are $f_7^i = x_i - \cos\left(2x_i - \sum_{i=1}^{4} x_i\right), i = 1, 2, 3, 4,$ being in this case $\xi \approx (0.5149, 0.5149, 0.5149, 0.5149)^T$.

If any of the methods used does not reach convergence in a maximum of 50 iterations, it is marked in Table 2 as 'nc'.

Proposed method ACTV shows a good performance, even compared with the seventh-order scheme WZQT. It is quite stable when the initial estimation is close to the root, but also in other cases. It is also interesting to be noticed that, being near the theoretical value, the calculated ACOC is not exactly 6 because of the low number of iterations needed to converge (3 or 4 in the most of cases).

4.1. The planar 1D-Bratu problem

We consider the classical Bratu problem [3] which is an elliptic nonlinear partial differential equation with homogeneous Dirichlet boundary conditions. In the planar 1D-case, the problem can be expressed as

$$\begin{gathered} u_{xx} + Ce^{u} = 0, \quad 0 \leq x \leq 1, \\ u(0) = u(1) = 0. \end{gathered} \tag{15}$$

This problem appears in a large variety of application areas such as the fuel ignition model of thermal combustion, radiative heat transfer, thermal reaction, the Chandrasekhar model of the expansion of the universe, chemical reactor theory and nanotechnology [9, 24, 13, 14]. Parameter C involved in Bratu problem has a critical turning point at $C_c \approx 3.513830719$. So, Bratu problem has two solutions for $C < C_c$, leading to two branches; there is only one solution for $C = C_c$ and no solution for $C > C_c$.

The exact solution to the planar 1D-Bratu problem is given by:

$$u(x) = 2ln\left[\frac{\cosh(\alpha)}{\cosh(\alpha(1-2x))}\right], \tag{16}$$

where α satisfies the transcendental equation

$$\cosh(\alpha) = \frac{4}{\sqrt{2C}}\alpha. \tag{17}$$

The solution of this equation may be performed using the classical Newton method, which results can be observed in Table 3, for two values of C, one of them stable $C = 3$, and also $C = 3.5$ (close to the critical turning value); both of them will be used in the numerical experiments. Let's notice that depending on the initial estimation of the iterative method, either the value corresponding to the lower or upper branch is obtained.

The solution of the problem using the finite-difference method involves discretizing the partial differential equation (15), including the boundary conditions. The method transforms the problem into a system of simultaneous nonlinear equations which are then solved using different iterative methods. Using a uniform grid with separation $h = 1/(n+1)$ for integer n, the range $0 \leq x \leq 1$ is discretized as $x_j = 0 + jh$, with $j = 0, 1, \ldots, n+1$.

By using a standard second order finite-difference scheme, the discrete version of the planar Bratu problem will be

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + Ce^{u_i} = 0, \quad i = 1, 2, \ldots, n, \tag{18}$$

with $u_0 = u_{n+1} = 0$. In the tests, we take $n = 10$.

For treating the resulting nonlinear system, it is well-known that the solution is much more reliable and obtained much more quickly if the initial guess is good. Then, a simple sinusoidal starting function (see [4] and [16]), $u(x) = a\sin(\pi x)$, having appropriate amplitude satisfying the boundary conditions, is used. In particular, for the lower branch, a should be taken as $a < u_c$, while the upper branch requires $a > u_c$ (otherwise, it could happen that due to a wrong initial guess, our method would converge to the opposite solution), where u_c is the unique solution of (16) when $C = C_c$. We have chosen $a = 1$ and $a = 3$ respectively. So, for each value of C and a, we calculate the estimated solution of the nonlinear system (18) by using the same schemes as in academic cases. The stopping criteria used is $\|u^{(k+1)} - u^{(k)}\| < 10^{-25}$ or $\|F(u^{(k+1)})\| < 10^{-25}$ and the maximum of iterations considered is 50. The information displayed at Table 4 corresponds to the estimations of the error of the iterative process $\|u^{(k+1)} - u^{(k)}\|$ and $\|F(x^{(k+1)})\|$ and the number of iterations, for $C = 3$, and values $a = 1$ and $a = 3$, in order to see the performance of the methods approximating each one of the branches of the solution. The approximated computational order of convergence $ACOC$ is not displayed, as it has a very unstable behavior and gives no relevant information, in this case.

Observed values in Table 4 show a stable behavior in all the methods. All the estimations are close enough to the solution but adding one more iteration (that is the case of MSSM method for $a = 3$) does not improve it: only the difference between the two last iterations has lower error, meanwhile the norm of the system at the last iteration is in the same range as the rest of the methods, that have one iteration less. Moreover, in Table 5, as the exact solution u is known, the norm of the exact error $\|u - u^{(k)}\|$ and the maximum error $\max\limits_{i\in\{1,\ldots,n\}} |u_i - u_i^{(k)}|$ are shown, both for lower (LB) and upper branch (UB). Let us remark that all the displayed errors coincide for the methods under analysis. Results presented in Table 5 coincide with the expected: when $a = 1$, all the methods converge to the solution at the lower branch, as the exact error at both branches shows. If $a = 3$, the lowest error appears at the upper branch. Let us remark that the exact error is much bigger than the tolerance used in the iterative method, this follows from the order of the discretization of the original differential equation.

In Figure 2, we observe the exact errors of the proposed method ACTV for $C = 3$ and $C = 3.5$, when $a = 3$, that is, the solution at upper branch is searched.

However, if $C = 3.5$ is considered (see Tables 6 and 7), the proximity to the critical value C_c yields different performances of the methods. In order to visualize this behavior, higher tolerance of the error has been used, being the stopping criteria used in this case $\|u^{(k+1)} - u^{(k)}\| < 10^{-4}$ or $\|F(u^{(k+1)})\| < 10^{-4}$. Let us notice that, depending on the branch, some procedures fail, diverging for the initial estimation used. Although the difference between the two last iterates is quite big, the value of the nonlinear system at the last iterate satisfies the stopping criterium. Moreover, the processes are, in general, more stable to find the solution at the upper branch than the one at the lower one, as can be noticed by the divergence of some methods and the diversity in the number of iterations needed in case of convergence. Indeed, the exact error of the iterative schemes are different, as can be observed in Table 7. The performance of the proposed methods is good in all convergent cases. Moreover, results in Table 7 show that, even in almost singular cases, the proposed methods and some of the known ones yield good enough results, in terms of proximity to the exact solution. In this case, lower and upper branches are close and the respective exact errors are similar simultaneously in both branches, for both values of parameter a.

[1] M. Abad, A. Cordero, J.R. Torregrosa, A family of seventh-order schemes for solving nonlinear systems. Bull. Math. Soc. Sci. Math. Roumanie 57(105) (2014) 133–145.

[2] F. Awawdeh, On new iterative method for solving systems of nonlinear equations, Numer. Algor. 54 (2010) 395–409.

[3] G. Bratu, Sur les equation integrals non-lineaires, Bull. Math. Soc. France 42 (1914) 113–142.

[4] J.P. Boyd, One-point pseudospectral collocation for the one-dimensional Bratu equation, Appl. Math. Comput. 217 (2011) 5553–5565.

[5] D. Budzko, A. Cordero, J.R. Torregrosa, A new family of iterative methods widening areas of convergence, Appl. Math. Comput. 252 (2015) 405–417.

[6] A. Cordero, J.L. Hueso, E. Martínez, J.R. Torregrosa, A modified Newton-Jarratt's composition, Numer. Algor. 55 (2010) 87–99.

[7] A. Cordero, J.R. Torregrosa, Variants of Newton's method using fifth-order quadrature formulas, Appl. Math. Comput. 190 (2007) 686–698.

[8] A. Cordero, J.G. Maimó, J.R. Torregrosa, M.P. Vassileva, Solving nonlinear problems by Ostrowski–Chun type parametric families, Math. Chem. 53 (2015) 430–449.

[9] I.M. Gelfand, Some problems in the theory of quasi-linear equations, Trans. Amer. Math. Soc. Ser. 2 (1963) 295–381.

[10] C. Grosan, A. Abraham, A new approach for solving nonlinear equations systems, IEEE Trans. Syst. Man Cybernet Part A: Syst. Humans 38 (2008) 698–714.

[11] C. Hermite, Sur la formule d'interpolation de Lagrange, Reine Angew. Math. 84 (1878) 70–79.

[12] J.L. Hueso, E. Martínez, C. Teruel, Convergence, effiency and dinamimics of new fourth and sixth order families of iterative methods for nonlinear systems, Comput. Appl. Math. 275 (2015) 412–420.

[13] J. Jacobsen, K. Schmitt, The Liouville-Bratu-Gelfand problem for radial operators, Differ. Equ. 184 (2002) 283–298.

[14] R. Jalilian, Non-polynomial spline method for solving Bratu's problem, Comput. Phys. Comm. 181 (2010) 1868–1872.

[15] Y. Lin, L. Bao, X. Jia, Convergence analysis of a variant of the Newton method for solving nonlinear equations, Comput. Math. Appl. 59 (2010) 2121–2127.

[16] A. Mohsen, A simple solution of the Bratu problem, Comput. Math. Appl. 67 (2014) 26–33.

[17] H. Montazeri, F. Soleymani, S. Shateyi, S.S. Motsa, On a new method for computing the numerical solution of systems of nonlinear equations, Appl. Math. vol. 2012 (2012) ID. 751975, 15 pages.

[18] J.J. Moré, A collection of nonlinear model problems, in:E.L. Allgower, K. Georg(Eds.), Computational Solution of Nonlinear Systems of Equations, Lectures in Applied Mathematics, vol.26, Amer. Math. Soc. Providence, RI, 1990 723–762.

[19] J.M. Ortega, W. C. Rheinbolt, Iterative Solutions of Nonlinears Equations in Several Variables, Academic Press, Inc., 1970.

[20] A.M. Ostrowski, Solution of equations and systems of equations, Prentice-Hall, Englewood Cliffs, New York, 1964.

[21] M. Petković, B. Neta, L. Petković, J. Džunić, Multipoint methods for solving nonlinear equations, Academic Press, Amsterdam, 2013.

[22] J.R. Sharma, H. Arora, Efficient Jarratt-like methods for solving systems of nonlinear equations, Calcolo 51 (2014) 193–210.

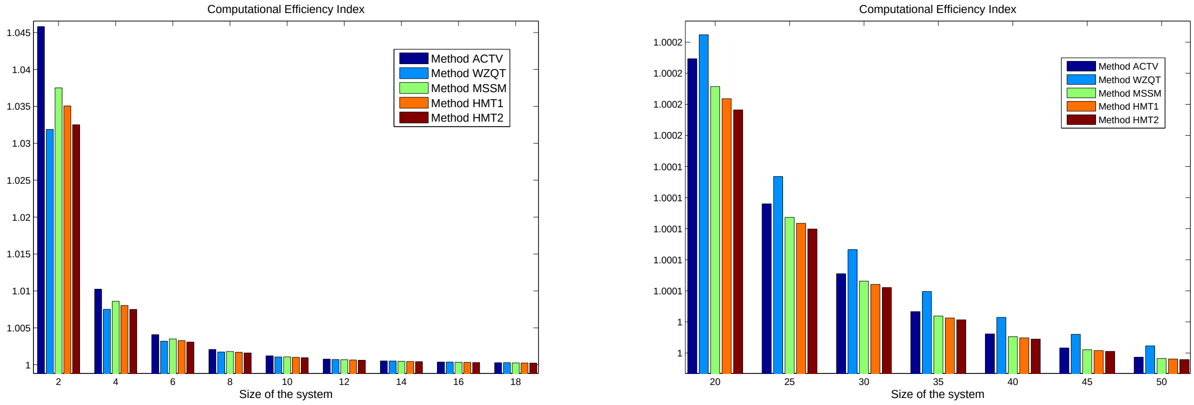
[23] I.G. Tsoulos, A. Stavrakoudis, On locating all roots of systems of nonlinear equations inside bounded domain using global optimization methods, Nonlinear Anal. Real World Appl. 11 (2010) 2465–2471.

[24] Y.Q. Wan, Q. Guo, N. Pan, Thermo-electro-hydrodynamic model for electrospinning process, Int. Nonlinear Sci. Numer. Simul. 5 (2004) 5–8.

[25] X. Wang, T. Zhang, W. Qian, M. Teng, Seventh-order derivative-free iterative method for solving nonlinear systems, Numer. Algor. 70 (2015) 545–558.

Method	NFE	$NLS1$	$NLS2$	DD	$M \times V$	CI
ACTV	$2n^2+2n$	3	1	1	1	$6^{\frac{1}{\frac{2}{3}n^3+8n^2+\frac{4}{3}n}}$
HMT1	$2n^2+2n$	3	3	0	3	$6^{\frac{1}{\frac{2}{3}n^3+11n^2+\frac{4}{3}n}}$
HMT2	$2n^2+2n$	3	3	0	4	$6^{\frac{1}{\frac{2}{3}n^3+12n^2+\frac{4}{3}n}}$
MSSM	$2n^2+2n$	5	0	0	3	$6^{\frac{1}{\frac{1}{3}n^3+10n^2+\frac{5}{3}n}}$
WZQT	$3n^2$	6	0	3	3	$7^{\frac{1}{\frac{1}{3}n^3+15n^2-\frac{1}{3}n}}$

Table 1: Functional evaluations and products-quotients of the methods



(a) Size from 2 to 18

(b) Size from 20 to 50

Figure 1: Index CI for different sizes of the system

	Methods	$\|x^{(k+1)} - x^{(k)}\|$	$\|F(x^{(k+1)})\|$	iter	$ACOC$
F_1	ACTV	4.65e-49	1.40e-206	5	6.0344
$x_0 = (2, -1)^T$	HMT1	1.43e-25	1.17e-152	6	6.5230
	HMT2	3.62e-33	1.72e-197	6	6.1459
	MSSM	6.89e-66	1.09e-198	5	-
	WZQT	nc	nc	nc	nc
F_2	ACTV	2.67e-38	4.94e-159	5	-
$x_0 = (1.5, 5.5)^T$	HMT1	4.16e-65	5.51e-208	4	5.9373
	HMT2	8.43e-76	7.80e-208	4	5.9948
	MSSM	5.05e-34	3.15e-201	4	5.9474
	WZQT	nc	nc	nc	nc
F_3	ACTV	2.25e-19	2.79e-113	3	5.5763
$x_0 = (0.5, 0.5)^T$	HMT1	4.00e-22	1.27e-131	3	5.7071
	HMT2	3.92e-23	6.67e-138	3	5.7516
	MSSM	5.52e-90	2.18e-209	4	5.9948
	WZQT	2.31e-20	4.88e-138	3	6.3732
F_4	ACTV	2.35e-78	3.48e-207	5	-
$x_0 = (1.5, 0.5, 1)^T$	HMT1	nc	nc	nc	nc
	HMT2	nc	nc	nc	nc
	MSSM	nc	nc	nc	nc
	WZQT	1.87e-34	4.96e-170	10	-
F_5	ACTV	1.67e-100	1.95e-208	4	5.9997
$x_0 = (1, 1, 1, -0.5)^T$	HMT1	2.47e-21	2.30e-145	3	6.5058
	HMT2	2.93e-22	4.76e-152	3	5.5603
	MSSM	1.74e-81	1.95e-208	4	5.9987
	WZQT	nc	nc	nc	nc
F_6	ACTV	5.01e-21	1.07e-125	3	5.8071
$x_0 = (1.25, \ldots, 1.25)^T$	HMT1	3.54e-24	2.24e-145	3	5.9329
$n = 49$	HMT2	2.43e-25	1.35e-152	3	5.9646
	MSSM	2.13e-99	0	4	5.9997
	WZQT	1.40e-19	5.72e-136	3	6.9500
F_7	ACTV	1.97e-12	1.36e-72	4	5.6879
$x_0 = (0, 0, 0, 0)^T$	HMT1	nc	nc	nc	nc
	HMT2	2.18e-33	8.69e-199	4	5.9570
	MSSM	nc	nc	nc	nc
	WZQT	2.75e-26	7.05e-181	15	-

Table 2: Numerical performance of proposed and comparing methods on academic nonlinear systems

	Value of α	
C	Lower branch	Upper branch
3	0.8434	1.6441
3.5	1.1380	1.2636

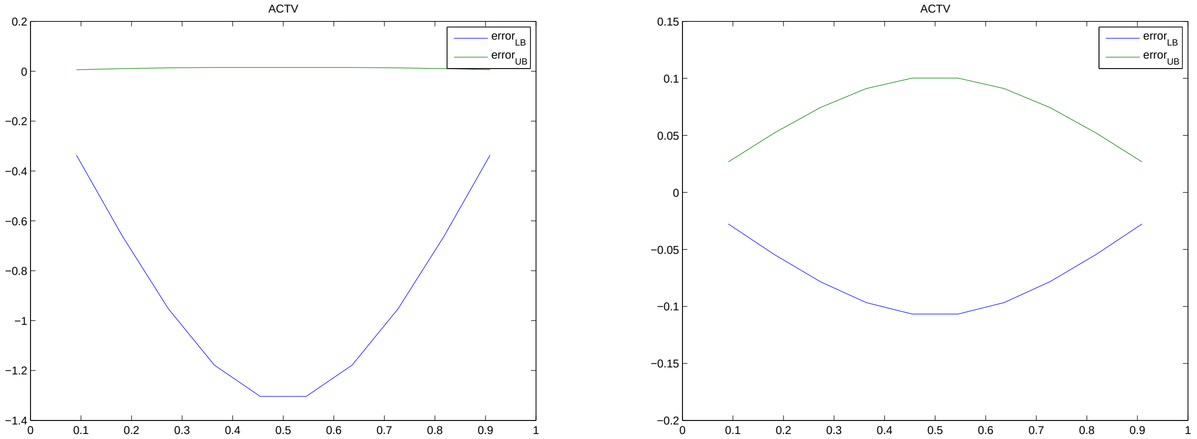
Table 3: Both solutions, lower and upper, of equation (17) for different values of C

	Methods	$\|u^{(k+1)} - u^{(k)}\|$	$\|F(u^{(k+1)})\|$	iter
$C = 3$	ACTV	9.93e-13	3.69e-40	3
$a = 1$	HMT1	1.70e-14	2.10e-40	3
	HMT2	8.81e-17	4.37e-41	3
	MSSM	2.98e-6	4.76e-37	3
	WZQT	5.72e-6	2.42e-40	3
$C = 3$	ACTV	1.58e-6	9.43e-38	3
$a = 3$	HMT1	5.18e-8	6.93e-40	3
	HMT2	1.06e-8	6.84e-40	3
	MSSM	5.46e-23	1.18e-39	4
	WZQT	1.49e-5	3.16e-35	3

Table 4: Approximating results of new and known methods on Bratu problem for $C = 3$

	Methods	$\|u-u^{(k)}\|_{LB}$	$\max\limits_{i\in 1,\ldots,n}\|u_i-u_i^{(k)}\|_{LB}$	$\|u-u^{(k)}\|_{UB}$	$\max\limits_{i\in 1,\ldots,n}\|u_i-u_i^{(k)}\|_{UB}$
$C=3$	ACTV	0.0091	0.0039	3.0444	1.3148
$a=1$	HMT1	0.0091	0.0039	3.0444	1.3148
	HMT2	0.0091	0.0039	3.0444	1.3148
	MSSM	0.0091	0.0039	3.0444	1.3148
	WZQT	0.0091	0.0039	3.0444	1.3148
$C=3$	ACTV	3.0146	1.3037	0.0392	0.0151
$a=3$	HMT1	3.0146	1.3037	0.0392	0.0151
	HMT2	3.0146	1.3037	0.0392	0.0151
	MSSM	3.0146	1.3037	0.0392	0.0151
	WZQT	3.0146	1.3037	0.0392	0.0151

Table 5: Errors of new and known methods on Bratu problem for $C=3$



(a) $C = 3$, $a = 3$

(b) $C = 3.5$, $a = 3$

Figure 2: Exact errors of ACTV on Bratu problem

	Methods	$\|u^{(k+1)} - u^{(k)}\|$	$\|F(u^{(k+1)})\|$	iter
$C = 3.5$	ACTV	0.2211	8.96e-5	6
$a = 1$	HMT1	nc	nc	nc
	HMT2	nc	nc	nc
	MSSM	0.1735	8.99e-5	3
	WZQT	0.1915	9.18e-5	4
$C = 3.5$	ACTV	0.2461	9.71e-5	3
$a = 3$	HMT1	0.2384	9.09e-5	3
	HMT2	0.2309	8.93e-5	3
	MSSM	0.1949	9.01e-5	7
	WZQT	nc	nc	nc

Table 6: Approximating results of new and known methods on Bratu problem for $C = 3.5$

	Methods	$\|u-u^{(k)}\|_{LB}$	$\max\limits_{i\in 1,\ldots,n}\|u_i-u_i^{(k)}\|_{LB}$	$\|u-u^{(k)}\|_{UB}$	$\max\limits_{i\in 1,\ldots,n}\|u_i-u_i^{(k)}\|_{UB}$
$C=3.5$ $a=1$	ACTV	0.2301	0.0993	0.2510	0.1077
	HMT1	nc	nc	nc	nc
	HMT2	nc	nc	nc	nc
	MSSM	0.2189	0.0944	0.2622	0.1125
	WZQT	0.2130	0.0919	0.2680	0.1150
$C=3.5$ $a=3$	ACTV	0.2474	0.1067	0.2337	0.1002
	HMT1	0.2356	0.1016	0.2454	0.1053
	HMT2	0.2235	0.0964	0.2576	0.1105
	MSSM	0.2176	0.0939	0.2635	0.1131
	WZQT	nc	nc	nc	nc

Table 7: Errors of new and known methods on Bratu problem for $C = 3.5$