Solving two classes of ODEs by using Chebyshev's method

Wesley V. I. Shirabayashi*
Department of Mathematics, State University of Maringá,
87020-900, Maringá, PR, Brazil
wvishirabayashi@uem.br

Emerson Vitor Castelani
Department of Mathematics, State University of Maringá
87020-900, Maringá, PR, Brazil
evcastelani@uem.br

Juliana V. Shirabayashi
Federal University of Paraná, Campus of Jandaia do Sul,
86900-000, Jandaia do Sul, PR, Brazil
juliana.verga@ufpr.br

Abstract

This paper presents a study on the use of Chebyshev's method for solving second-order equations with boundary conditions. This study involves the production of an algorithm for solving a family of second order problems with multiple points in the boundary by finite difference scheme. The resulting nonlinear systems are solved by iterative method of Chebyshev. A comparison with Newton's method is accomplished.

keywords. Newton's method, Chebyshev's method, Nonlinear systems

1 Introduction

The study of methods for solving nonlinear systems is not new, but the growing emergence of real-world problems formulated under this optics has made these studies also focused on practical efficacy of the methods.

A great example is the development of quasi-Newton methods which have no theoretical power of the local quadratic convergence of Newton's method, but are quite efficient in practice [4, 18].

There are also methods of Chebyshev-Halley's family [9, 10, 11, 12]. These methods are of type tensorial and its main advantage is the cubic local convergence. On the other hand its main drawback is the use of high computational processing and storage due to structure tensorial.

The goal of this paper is to present an application of the Chebyshev's method for solving second order differential equations with two and three points on the boundary.

*Corresponding author

More specifically, let us consider the following equations:

$$\begin{cases} u'' = f(x, u, u') \\ u(0) = u(1) = 0, \end{cases} \tag{1}$$

$$\begin{cases} u'' = f(x, u, u') \\ u(0) = 0, u(1) = g(u(\eta)), \end{cases} \tag{2}$$

where $f : [0, 1] \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ are continuous nonlinear functions and $\eta \in (0, 1)$.

The idea that we use is based on the finite difference scheme. Briefly, we discretize the equations (1)-(2) and generate nonlinear systems to solve them. The performance of such a scheme depends largely on the strategy that we use to solve the nonlinear system. In this case, we will use Chebyshev's method. We will see that the resulting structure of the problem is favorable to use the Chebyshev's method and will verify the efficiency of this method by comparing it with Newton's method.

This work is divided as follows. In Section 2 we present the methods of Newton and Chebyshev. In section 3, we present the discretization of equations (1) and (2) and the computation of the tensors. Sections 4 and 5 are devoted to algorithms and examples. In section 6, we have the final considerations.

Remark. Finite difference scheme are widely used to solve differential equations. However, strategies that use the Chebyshev's method are poorly explored due to the calculation of the tensors. Moreover, considering the equation (2), we have that the numerical methods used in this class are fixed-point type [3], or using approximations in reproducing kernel spaces [15]. Thus, in this article we provide a new algorithm for solving (2).

2 Newton and Chebyshev's Methods for nonlinear systems

Let us consider a nonlinear function $F : \mathbb{R}^n \to \mathbb{R}^n$ and the problem of finding $w \in \mathbb{R}^n$ such that:

$$F(w) = 0. \tag{3}$$

There are several iterative methods to solve this problem and probably, the Newton's method is the most know. An important property of this method is its fast local convergence. In general we assume that F is sufficiently smooth. The solution w^* for the problem (3) is called as degenerate or singular, if the Jacobian matrix $F'(w^*)$ is singular. Else, the solution is called nondegenerate. Some interesting works related to development of Newton's method are Galántai [4] e Yamamoto [18].

In Newton's method, if we have an approximation to the solution w^k, the iterative squeme is defined by:

$$w^{k+1} = w^k + d^k, \tag{4}$$

where d^k is solution of the following linear system:

$$F'(w^k)d^k = -F(w^k), \tag{5}$$

and $F'(w^k)$ is nonsingular.

The Newton's method solves at each iteration a linear model to F around the point w^k. In recent years, some research has been focused on developing methods that have fast local convergence [9, 10, 11, 12] but many of these methods have been known for decades and have been rewritten and adapted to reduce the computational cost.

In particular, if we consider scalar equations, methods with cubic local convergence were gathered in a family called Chebyshev-Halley [12]. Subsequently, Gutirrez and Herndez [9] extended the results for equations in Banach spaces.

Notation: In the present text we will use the following terms:

$$F_k = F(w^k),\ \ F'_k = F'(w^k),\ \ F''_k = F''(w^k),$$

As established in [9], methods of the Chebyshev-Halley's family are defined by:

$$w^{k+1} = w^k - \left[I + \frac{1}{2}(F'_k)^{-1}F''_k(F'_k)^{-1}F_k\left[I - \lambda(F'_k)^{-1}F''_k(F'_k)^{-1}F_k\right]^{-1}\right](F'_k)^{-1}F_k, \qquad (6)$$

where $\lambda \in [0,1]$ is the parameter that defines each member of this family.

If $\lambda = 0$ we have the Chebyshev's method, if $\lambda = \frac{1}{2}$, we have Halley's method and if we consider $\lambda = 1$, the method defined is called super-Halley, [12].

Classical results of quadratic local convergence for Newton's method or cubic local convergence for Chebyshev-Halley's family are valid only $\|(F'(w))^{-1}\|$ can be quoted in a neighborhood of a solution w^*.

Gundersen and Steihaug in [5] showed that this family of methods can be defined as follows:

$$w^{k+1} = w^k + d_N + d, \qquad (7)$$

where d_N and d are solutions of the linear systems given by:

$$F'_k d_N = -F_k, \qquad (F'_k + \lambda F''_k d_N)d = -\frac{1}{2}F''_k d_N d_N, \qquad (8)$$

with $\lambda \in [0,1]$. The operation $F''_k d_N d_N$ is the vector whose components are given by:

$$(F''_k d_N d_N)_l = \sum_{j=1}^{n-1}\sum_{i=1}^{n-1}(F''_k)_{i,j,l}(d_N)_i(d_N)_j, \quad l = 1,\dots,n-1.$$

In [5, 6] the authors prove that for certain classes of problems, the ratio between the cost per iteration of each method of Chebyshev-Halley's family and Newton's method is independent of the size of problem, being asymptotically constant.

In this paper we use the Chebyshev's method, because it is necessary to solving two linear systems with the same coefficient matrix, and that generates an economy in the calculation of directions.

In tests in [5, 6] for unconstrained optimization problems, the methods of Chebyshev-Halley's family had performance compatible with the Newton's method.

3 Discrete formulation of the problems

In this section we discretize the equations (1) and (2). Then we define the parameters needed to utilize the above-mentioned methods. Consider first the equation (1).

Let $x_0 = 0$, $x_1 = x_0 + h$, ..., $x_n = x_{n-1} + h$ be a uniformly spaced mesh in $[0,1]$. The problem (1) can be discretized by finite difference formulas:

$$u'(x_i) \approx u'_i = \frac{u_{i+1} - u_{i-1}}{2h} \qquad (9)$$

and
$$u''(x_i) \approx u''_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}, \tag{10}$$
where $i = 1, ..., n-1$. Thus replacing (9) and (10) into (1) we have the following nonlinear system:

$$u_2 - 2u_1 - h^2 f(x_1, u_1, \frac{u_2}{2h}) = 0,$$

$$u_{i+1} - 2u_i + u_{i-1} - h^2 f(x_i, u_i, \frac{u_{i+1} - u_{i-1}}{2h}) = 0,\ i = 2, ..., n-2,$$

$$-2u_{n-1} + u_{n-2} - h^2 f(x_{n-1}, u_{n-1}, \frac{-u_{n-2}}{2h}) = 0.$$

In order to use the Chebyshev's method we need to compute the Jacobian matrix and tensor of the function $F : \mathbb{R}^{n-1} \to \mathbb{R}^{n-1}$ defined by:

$$F(u_1, ..., u_{n-1}) = \begin{pmatrix} u_2 - 2u_1 - h^2 f(x_1, u_1, \frac{u_2}{2h}) \\ u_3 - 2u_2 + u_1 - h^2 f(x_2, u_2, \frac{u_3 - u_1}{2h}) \\ \vdots \\ u_{n-1} - 2u_{n-2} + u_{n-3} - h^2 f(x_{n-2}, u_{n-2}, \frac{u_{n-1} - u_{n-3}}{2h}) \\ -2u_{n-1} + u_{n-2} - h^2 f(x_{n-1}, u_{n-1}, \frac{-u_{n-2}}{2h}) \end{pmatrix}.$$

Denoting the Jacobian matrix of F by $F' = (F')_{i,j}$ we have that F' is an tridiagonal matrix. Consequently, if ∇_i represents the i-th term of the gradient vector of f, we have:

$$F'_{1,1} = -2 - h^2 \nabla_2 f(x_1, u_1, \frac{u_2}{2h}), \quad F'_{1,2} = 1 - \frac{h}{2} \nabla_3 f(x_1, u_1, \frac{u_2}{2h}),$$

$$F'_{i,i-1} = 1 + \frac{h}{2} \nabla_3 f(x_i, u_i, \frac{u_{i+1} - u_{i-1}}{2h}), \quad F'_{i,i} = -2 - h^2 \nabla_2 f(x_i, u_i, \frac{u_{i+1} - u_{i-1}}{2h}),$$

$$F'_{i,i+1} = 1 - \frac{h}{2} \nabla_3 f(x_i, u_i, \frac{u_{i+1} - u_{i-1}}{2h}), \text{ for } i = 2, ..., n-2,$$

and $F'_{i,j} = 0$ if $j \neq i-1, i, i+1$.

$$F'_{n-1,n-2} = 1 + \frac{h}{2} \nabla_3 f(x_{n-1}, u_{n-1}, \frac{-u_{n-2}}{2h}),$$

$$F'_{n-1,n-1} = -2 - h^2 \nabla_2 f(x_{n-1}, u_{n-1}, \frac{-u_{n-2}}{2h}).$$

According to [6][Theorem 2], in unconstrained optimization, the structure of sparsity of the tensor depends on the structure of sparsity of the Hessian matrix. In our case, as the Jacobian matrix is tridiagonal we have that the tensor is sparse and its structure is kind of diagonal. In fact, denoting the tensor by $F'' = F''_{i,j,l}$ of F, we have:

For $l = 1$,

$$F''_{1,1,1} = -h^2\nabla^2_{2,2}f(x_1, u_1, \frac{u_2}{2h}), \qquad F''_{1,2,1} = -\frac{h}{2}\nabla^2_{2,3}f(x_1, u_1, \frac{u_2}{2h}),$$

$$F''_{2,1,1} = -\frac{h}{2}\nabla^2_{3,2}f(x_1, u_1, \frac{u_2}{2h}), \qquad F''_{2,2,1} = -0.25\nabla^2_{3,3}f(x_1, u_1, \frac{u_2}{2h}).$$

For $l = 2, ..., n-2$,

$$F''_{l-1,l-1,l} = -0.25\nabla^2_{3,3}f_l, \quad F''_{l,l-1,l} = F''_{l-1,l,l}, \qquad F''_{l+1,l-1,l} = F''_{l-1,l+1,l},$$

$$F''_{l-1,l,l} = \frac{h}{2}\nabla^2_{2,3}f_l, \qquad F''_{l,l,l} = -h^2\nabla^2_{2,2}f_l, \qquad F''_{l+1,l,l} = F''_{l,l+1,l},$$

$$F''_{l-1,l+1,l} = 0.25\nabla^2_{3,3}f_l, \qquad F''_{l,l+1,l} = -\frac{h}{2}\nabla^2_{2,3}f_l, \quad F''_{l+1,l+1,l} = F''_{l-1,l-1,l},$$

where $\nabla^2_{i,j}f_l = \nabla^2_{i,j}f(x_l, u_l, \frac{u_{l+1} - u_{l-1}}{2h})$.

For $l = n-1$,

$$F''_{n-2,n-2,n-1} = -0.25\nabla^2_{3,3}f_{n-1}, \qquad F''_{n-1,n-2,n-1} = F''_{n-1,n-2,n-1},$$

$$F''_{n-2,n-1,n-1} = -\frac{h}{2}\nabla^2_{2,3}f_{n-1}, \qquad F''_{n-1,n-1,n-1} = -h^2\nabla^2_{2,2}f_{n-1},$$

where $\nabla^2_{i,j}f_{n-1} = \nabla^2_{i,j}f(x_{n-1}, u_{n-1}, -\frac{u_{n-2}}{2h})$.

The remaining terms $F''_{i,j,l}$ are null.

Thus the sparsity of the tensor of F provides the necessary structure to facilitate the implementation of the Chebyshev's method. Also,we can note that Jacobian matrix of F can be rewritten as follows:

$$F' = A + h\overline{A},$$

where

$$A = \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \cdots & -2 & 1 \\ 0 & 0 & \cdots & 1 & -2 \end{bmatrix}$$

and

$$\overline{A} = \begin{bmatrix} -h\nabla_2 f_1 & -\frac{1}{2}\nabla_3 f_1 & 0 & \cdots & 0 \\ \frac{1}{2}\nabla_3 f_2 & -h\nabla_2 f_2 & -\frac{1}{2}\nabla_3 f_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \cdots & -h\nabla_2 f_{n-2} & -\frac{1}{2}\nabla_3 f_{n-2} \\ 0 & 0 & \cdots & \frac{1}{2}\nabla_3 f_{n-1} & -h\nabla_2 f_{n-1} \end{bmatrix}$$

where $\nabla_i f_j = \nabla_i f(x_j, u_j, \frac{u_{j+1} - u_{j-1}}{2h})$ with ∇_i represents $i-$th $(i = 1, 2, 3)$ component of the gradient vector of f. Thus, by taking h sufficiently small and observing that A is negative defined, we have that F' is negative defined. Therefore, its eigenvalues are negative and consequently there is inverse of F'. Then, according Yamamoto [18] is possible to obtain results of local cubic convergence.

Now, let us consider the second order boundary value problem given by (2). Let c be an approximation to $u_n = g(u(\eta))$. Thus replacing the term $\frac{-u_{n-2}}{2h}$ by $\frac{c - u_{n-2}}{2h}$ on definitions of F, F' and F'', we get the necessary parameters in order to use the Chebyshev's method.

4 Solving a classical second order two point boundary value problem

In what follows we define two algorithms that apply Newton and Chebyshev's method to solve (1).

Algorithm 1

Step 1. Set $k \leftarrow 1$ and define a uniformly spaced mesh $\{x_j\}$.

Step 2. Discretize the problem by finite difference and choose an initial approximation $u_j^0 = u^0(x_j)$.

Step 3. Solve:

a) $F'(u^k)d_N = -F(u^k)$.

b) $F'(u^k)d = -\frac{1}{2}F''(u^k)d_N d_N$.

c) $u^{k+1} = u^k + d_N + d$.

Step 4. Set $k \leftarrow k + 1$ and go to step 3.

Algorithm 2

Step 1. Set $k \leftarrow 1$ and define a uniformly spaced mesh $\{x_j\}$.

Step 2. Discretize the problem by finite difference and choose an initial approximation $u_j^0 = u^0(x_j)$.

Step 3. Solve:

a) $F'(u^k)d_N = -F(u^k)$.

b) $u^{k+1} = u^k + d_N$.

Table 1: Comparison between Newton and Chebyshev's method considering example 1 and n=10.

It	Chebyshev		Newton	
	ε^k	ε_u^k	ε^k	ε_u^k
1	5.7493e-02	7.3065e-03	5.8231e-02	7.2954e-03
5	2.8428e-04	1.6841e-04	2.6216e-04	1.7448e-04
10	2.8740e-06	2.5760e-04	2.6498e-06	2.5768e-04
15	2.9052e-08	2.5860e-04	2.6785e-08	2.5860e-04
26	1.1844e-12	2.5861e-04	1.0920e-12	2.5861e-04
27	4.7254e-13	2.5861e-04	4.3567e-13	2.5861e-04

Table 2: Comparison between Newton and Chebyshev's method considering example 1 and n=100.

It	Chebyshev		Newton	
	ε^k	ε_u^k	ε^k	ε_u^k
1	6.0196e-02	9.7326e-05	6.1040e-02	9.1017e-04
5	5.9204e-06	5.5496e-06	3.4479e-06	3.1738e-06
10	1.6729e-07	2.5256e-06	9.7422e-08	2.5552e-06
15	4.7268e-09	2.5944e-06	2.7527e-09	2.5953e-06
26	1.8493e-12	2.5964e-06	1.0770e-12	2.5964e-06
27	9.0619e-13	2.5964e-06	5.2774e-13	2.5964e-06

Step 4. Set $k \leftarrow k+1$ and go to step 3.

Using examples, we present numerical results that elucidate the behavior of the methods discussed so far. In present numerical study we are considering

$$\varepsilon^k = \max_{i=1,\dots,n-1} |u_i^{k+1} - u_i^k|,$$

$$\varepsilon_u^k = \max_{i=1,\dots,n-1} |u(x_i) - u_i^k|$$

and It represents the number of iterations. The tests were run on a MacBook Pro with 4GB RAM and Core i5 2.4.

Example 1. *Consider the problem:*

$$u'' = f(x,u,u') = 4u - u^2 + u'^2 - 2 + x^4 + (2 + x + 4x^2 - 2x^3)e^{x-1} - (1+2x)e^{2x-2}$$

$$u(0) = u(1) = 0.$$

An exact solution for this equation is $u(x) = xe^{x-1} - x^2$. In this problem we consider a discretization with $n = 10$ and $n = 100$ points and set the stopping criterion by $\varepsilon^k < 10^{-12}$. The tables in sequence provide the output of the methods at each case.

As we can be seen in Tables 1 and 2 (see ε_u^k) the accuracy of results is related and depends on the approximation by finite differences. When we increase the number of points we have a better approximation for derivatives and consequently more accurate numerical results.

The runtime considering $n = 10$ was 0.162614 seconds to Chebyshev's method and 0.132883 seconds to Newton's method. Considering $n = 100$ the time increase to 0.361866 and 0.255741 seconds, respectively.

Table 3: Comparison between Newton and Chebyshev's method considering example 2 and n=10.

It	Chebyshev		Newton	
	ε^k	ε_u^k	ε^k	ε_u^k
1	2.2637e+00	1.4025e+00	2.2637e+00	1.4025e+00
5	4.9110e-02	1.6345e-02	6.8185e-02	4.2889e-02
10	1.3156e-07	1.6257e-02	2.9075e-05	1.6257e-02
15	9.3336e-13	1.6257e-02	2.1182e-10	1.6257e-02
18	-	-	1.7258e-13	1.6257e-02

Example 2. *Let's consider in this example the following problem:*

$$u'' = -\pi^2 u + u^2(u')^2 + u^3 + (u')^3 - \sin^2 x\pi - \pi^3 \cos^3 \pi x - \pi^2 \sin^2 \pi x \cos^2 \pi x,$$
$$u(0) = u(1) = 0.$$

An exact solution is given by $u(x) = \sin(\pi x)$ and the stopping criterion is again $\varepsilon^k < 10^{-12}$. The results are expressed in Table 3.

The runtime to apply the first algorithm was 0.155053 seconds and to apply the second method was 0.138070 seconds.

5 Solving a second order three-point boundary value problem

Our goal now is to adapt the previous methods to solve the following equation:

$$u'' = f(x, u, u')$$
$$u(0) = 0, u(1) = g(u(\eta)),$$

where $\eta \in (0, 1)$ and g is possibly nonlinear. This equation is commonly referenced in the literature as a second order three-point (or multi-point) boundary equation.

Due to various applications involving this type of equation, many authors have studied aspects about the existence of solution (we recommend [13, 14, 1, 2, 7, 8, 16, 17]). However, there are few numerical methods to solve it. To contribute in this direction, we present below a new algorithm that uses Chebyshev's method.

Algorithm 3

Step 1. Set $k \leftarrow 1$ and define a uniformly spaced mesh $\{x_j\}$.

Step 2. Discretize the problem by finite difference and choose an initial approximation $u_j^0 = u^0(x_j)$.

Step 3. Solve:

a) $u(\eta)$ by using cubic spline interpolation.

b) $F'(u^k)d_N = -F(u^k)$.

c) $F'(u^k)d = -\frac{1}{2}F''(u^k)d_N d_N$.

d) $u^{k+1} = u^k + d_N + d$.

Table 4: Comparison between Newton and Chebyshev method's considering example 3.

It	Chebyshev		Newton	
	ε^k	ε_u^k	ε^k	ε_u^k
1	4.4959e-01	4.8747e-01	4.1625e-01	4.9412e-01
5	3.4910e-02	3.8697e-02	4.1660e-02	4.7080e-02
10	1.3925e-03	1.4157e-03	1.7087e-03	1.7636e-03
15	5.4369e-05	2.3739e-04	6.6792e-05	2.2639e-04
18	7.7635e-06	2.7921e-04	9.5378e-06	2.7762e-04

Step 4. Set $k \leftarrow k+1$ and go to step 3.

Naturally, an algorithm equivalent to Algorithm 3 can be obtained if we use the Newton's method.

Algorithm 3

Step 1. Set $k \leftarrow 1$ and define a uniformly spaced mesh $\{x_j\}$.

Step 2. Discretize the problem by finite difference and choose an initial approximation $u_j^0 = u^0(x_j)$.

Step 3. Solve:

a) $u(\eta)$ by using cubic spline interpolation.

b) $F'(u^k)d_N = -F(u^k)$.

c) $u^{k+1} = u^k + d_N$.

Step 4. Set $k \leftarrow k+1$ and go to step 3.

As we can see in Algorithms 3 and 4, each iteration requires an interpolation by cubic spline. Therefore, it is desirable to save iterations once we can save considerable processing. In this sense the Chebyshev's method is an interesting tool. Using the following example we will analyze the behavior of the algorithms. The presented results are obtained when we use the stopping criterion given by $\varepsilon_u^k < 10^{-5}$.

Example 3. *Consider a three point boundary value problem given by:*

$$f(x,u,u') = u'^2 - 9x^4 + 3x^2 - 6x - 0.25,$$

$$g(s) = s^2 - 0.5180491164$$

and

$$\eta = 0.37$$

An exact solution for this problem is $u(x) = 0.5x - x^3$. The results obtained by applying the Algorithms 3 and 4 with $n = 20$ are given in Table 4.

The runtime of Algorithm 3 and 4 were, respectively, 0.183998 and 0.159683 seconds.

Example 4. *Now consider the following problem:*

$$f(x,u,u') = u'(u'^2 - u^2 + 2xu' + x^2 - 2x - 1),$$

$$g(s) = \frac{4}{3}s$$

and

$$\eta = 0.75.$$

Table 5: Comparison between Newton and Chebyshev method's considering example 4.

It	Chebyshev		Newton	
	ε^k	ε_u^k	ε^k	ε_u^k
1	5.3551e-01	3.3113e-01	6.5472e-01	3.1971e-01
5	3.7546e-02	2.1154e-01	3.9660e-02	2.3436e-01
10	2.2918e-02	6.7411e-02	2.5227e-02	8.1517e-02
15	5.6537e-03	7.0790e-03	7.3152e-03	1.0290e-02
25	1.6789e-05	1.4404e-05	2.2764e-05	2.3670e-05
26	9.1665e-06	5.2371e-06	1.3585e-05	1.0085e-05
27	-	-	9.5378e-06	2.7762e-04

An exact solution is $u(x) = x$. We define the discretizaton with $n = 10$ and initial guess $x_0 = (0, 0.5, ..., 0.5)$ because the null vector is a solution of this problem. The result of the numerical experiment are shown in Table 5. Using the Algorithm 3 the execution time was 0.185904 seconds and using the Algorithm 4 was 0.167088 seconds.

6 Conclusion and final remarks

Although Chebyshev type methods have cubic convergence, its application in nonlinear systems is limited due to the tensors calculation. Thus, nonlinear systems that allow good use of the method need to have a favorable structure. In this paper, we showed that the finite difference scheme for solving two classes of differential equations have this structure. This is the main reason that justify this work.

This study includes advances in two fields: the first, on the applicability of the Chebyshev's method and the second for solving second order three-point boundary value problems.

Naturally, the examples indicate that the runtime of algorithms using Newton's method were lower than the algorithms that use the method of Chebyshev. However, the use of the Chebyshev's method is promising in the following sense: In general, the algorithms that are based on Chebyshev's method consumed less iterations indicating that, with suitable parallel programming techniques or automatic differentiation, may have a very efficient algorithm for solving certain classes of nonlinear problems.

References

[1] Agarwall, R. P., Thompson, H. B., Tisdell, C. C., Three-Point Boundary Value Problem for second order discrete equation, Computers and Mathematics with Applications (2003).

[2] Bai Z., Wang Y. and Ge, W., Triple positive solutions for a class of two-point boundary-value problems, Electron. J. Differential Equations (2004).

[3] Castelani, E.V., Ma, T. F., Numerical solutions for a three-point boundary value problem, Communications in Applied Analysis , V11, N1, 87-96 (2007).

[4] Galántai A., The theory of Newton's method. J. of Comp. and Appl. Math., V124, 25-44 (2000).

[5] Gundersen G., Steihaug, T., On large scale unconstrained optimization problems and higher order methods. Optim. Meth. and Soft., 25(2), 337-358 (2010).

[6] Gundersen, G., Steihaug T., Sparsity in higher order methods in optimization, Reports in Informatics - Dep. of Informatics - universty of Bergen - Norway, ISSN 0333-3590, n. 327, (2006).

[7] Gupta, C. P., Solvability of a three-point nonlinear boundary value problem for a second order ordinary differential equation, J. Math. Anal. Appl, V168, N2, 540-551 (1992).

[8] Gupta, C. P., Trofimchuk, S. I., A sharper condition for the solvability of a three-point second order boundary value problem, J. Math. Anal. Appl, V205, N2, 586-597 (1997).

[9] Gutiérrez, J. M., Hernández, M. A., A family of Chebyshev-Halley type methods in Banach spaces, Bull. Austral. Math. Soc., V55, 113-130, (1997).

[10] Gutiérrez, J. M., Hernández, M. A., Recurrence relations for the super-Halley method, Comput. Math. Appl., V36, 1-8 (1998).

[11] Gutiérrez, J. M., Hernández, M. A., An acceleration of Newton's method: Super-Halley method, Appl. Math. and Comp., V117, 223-239 (2001).

[12] Hernández M.A., Salanova, M. A., A family of Chebyshev-Halley type methods, Intern. J. of Computer Math., V47, 59-63 (1993).

[13] Il'in, V. A., Moiseev, E. I., Nonlocal boundary-value problem of the secod kind for a Sturm-Liouville operator, Differential Equations, V23, 979-987 (1987).

[14] Il'in, V. A., Moiseev, E. I., A nonlocal boundary value problem for the Sturm-Liouville operator in the differential and difference treatments, Sov. Math. Dokl, V34, 507-511 (1987).

[15] Lin, Y., Cui, M., A numerical solution to nonlinear multi-point boundary value problems in the reproducing kernel space, Mathematical Methods in the Applied Sciences, V34, 44-47 (2011).

[16] Ma, R., Existence theorems for a second order m-point boundary value problem, J. Math. Anal. Appl, V211, 545-555 (1997).

[17] Ma, R., Multiplicity of positive solutions for second-order three-point boundary value problems, Comput. and Math. with Appl, V40. N2, 193-204 (2000).

[18] Yamamoto, T., Historical developments in convergence analysis for Newton's and Newton-like methods, J. of Comp. and Appl. Math., V124, 1-23 (2000).