Speeding-up The Performance of The Nonlinear Shooting Method for Solving Nonlinear BVPs

Abbas Y. Al-Bayati **Ann J. Al-Sawoor**
Department of Mathematics
College of Computers and Mathematical Sciences
Mosul University

(Received 8/1/2001 , Accepted 26/2/2001)

ABSTRACT

The aim in this paper is to modify and obtain less execution time of the nonlinear shooting method which used to approximate the solution of the nonlinear boundary-value problems (BVPs) .

The modification is due to replacing single step method (Runge-Kutta for systems) by multi-step method (Adam predictor-corrector of order four for systems) which is used for solving the initial – value problems (IVPs) .

إسراع طريقة القذف غير الخطية المستخدمة في حل مسائل القيم الحدودية غير الخطية

الملخص

هدفنا في هذا البحث هو التحسين والحصول على أقل زمن تنفيذي لطريقة القذف غير الخطية والتي تستخدم لتقريب الحل لمسائل القيم الحدودية (BVPs) .

ان هذا التحسين ناتج عن استبدال طريقة الخطوة الواحدة (رانج-كوتا للأنظمة) بطـرق المتعـددة الخطوات من نوع مخمن-والمصحح للأنظمة والتي تـستخدم لحـل مـسائل القـيم الابتدائيـة (IVPs)

INTRODUCTION

Problems that frequently arises in applications is a Boundary-value problem (Burden and Faires, 1993; Khalaf, 1988; Jain and Iyengar, 1994; Stoer and Bulirsch, 1980) in which, in addition to the differential equation, information about the solution and perhaps some derivative is specified at two different values of the independent variables.

The general two points of boundary-value problems in this paper involve a second-order differential equation of the form

$$y'' = f(x, y, y') \qquad a \le x \le b \qquad ....(1)$$

together with the boundary conditions

$$y(a) = \alpha \quad \text{and} \quad y(b) = \beta \qquad ....(2)$$

There are common techniques for approximating the solution of the boundary-value problem . These techniques are similar to finite difference (See for example,Khalaf, 1988; Conte and de Boor, 1980; Fox and Mayers, 1987) and shooting methods (See for example Conte and de Boor, 1980; Fox and Mayers, 1987; Al-Assady and Al-Sawoor, 2000) .

In this paper we used nonlinear shooting method, i.e we used shooting methods for nonlinear differential equation.

In this paper we will improve and speed-up the nonlinear shooting method which is used to approximate the solution of the nonlinear BVPs by replacing single - step method (Runge-Kutta for systems) by multi-step method (Adam-Bashforth-Moulton for systems) which are used to solve the IVPs where it is well-known that numerical methods for solving ordinary differential equations have to face two different kinds of errors . The method error (or truncation error) depends on the order of the method and decreases when the step size decreases . On the contrary, the round-off error generally increases when the stepsize decreases (See Alt and Vignes, 1996), here in this paper we have used stepsize h computed by the form

$$h = (b-a) / N$$

where a and b are the endpoints of the interval [a,b] and N is the number of subintervals .

1. **The Nonlinear Shooting Method**

Al-Assady and Al-Sawoor (2000) have used shooting technique for solving linear second-order BVPs where the solution to linear problems can be simply expressed as a linear combination of the solutions to two initial-value problems. In this paper, in order to obtain the solution of the shooting technique for the nonlinear second-order boundary-value problem

$$y'' = f(x, y, y') \qquad a \le x \le b \qquad \text{..........(1.1)}$$
$$y(a) = \alpha \ , \ y(b) = \beta$$

we need to use the solutions to a sequence of the initial-value problems of the form

$$y'' = f(x, y, y') \qquad a \le x \le b \qquad \text{..........(1.2)}$$
$$y(a) = \alpha \ , \ y'(a) = t \qquad ,$$

involving a parameter t, to approximate the solution to our boundary-value problem .

Note that in this paper we have replaced the Runge-Kutta for systems by Adam predictor-corrector for systems to obtain the solutions to a sequence of IVPs . We do so to speed-up the performance of the nonlinear shooting method .

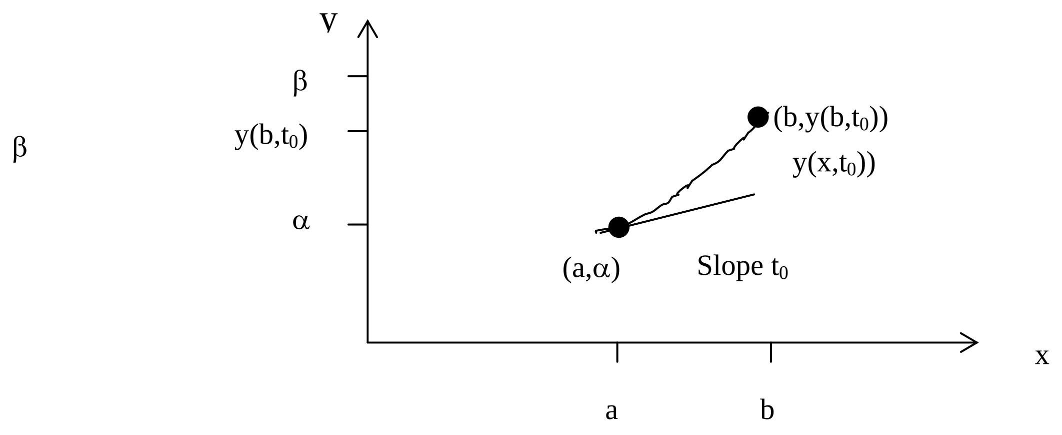

Fig.1: The procedure of firing objects at a stationary target

Now we do this by choosing the parameters $t=t_k$ in a manner to ensure that

$$\lim_{k\to\infty} y(b,t_k) = y(b) = \beta \qquad ....(1.3)$$

where $y(x,t_k)$ denotes the solution to the initial-value problem (1.2) with $t=t_k$ and $y(x)$ denotes the solution to the BVP (1.1) .

This technique is called “Shooting” method, by analogy to the procedure of firing objects at a stationary target . (See figure(1)) we start with a parameter t_0 that determines the initial elevation at which the object is fired from the point (a,α) and along the curve described by the solution to the initial-value problem

$$y'' = f(x,y,y') \qquad a \le x \le b \qquad ....(1.4)$$
$$y(a) = \alpha \ , \ y'(a) = t_0$$

If $y(b,t_0)$ is not sufficiently close to β, we attempt to correct our approximation by choosing another elevation t_1 and so on, until $y(b,t_k)$ is sufficiently close to “hitting” β. (See figure(2)) .

To determine the parameters t_k, if $y(x,t)$ denotes the solution to the initial-value problem (1.2), the problem have been determint so that :

$$y(b,t) - \beta = 0 \qquad ……….(1.5)$$

Since this is a nonlinear equation, a number of methods are available to solve it, eq. (1.5) .

One of these methods is Newton’s method . To use the more powerful Newton’s method to generate the sequence $\{t_k\}$, only one initial-value, t_0, is needed. However, the iteration has the form

$$t_k = t_{k-1} - \frac{y(b,t_{k-1}) - \beta}{dy(b,t_{k-1})/dt} \qquad ....(1.6)$$

where $dy(b,t_{k-1})/dt \equiv \dfrac{dy(b,t_{k-1})}{dt}$, and requires the knowledge of $dy(b,t_{k-1})/dt$

This presents a difficulty, since an explicit representation for $y(b,t)$ is not known; we know only the values $y(b,t_0),y(b,t_1),\ldots,y(b,t_{k-1})$.

Suppose we rewrite the initial-value problem (1.2), emphasizing that the solution depends on both x and t :

$$y''(x,t) = f(x, y(x,t), y'(x,t)) \qquad a \le x \le b \qquad ....(1.7)$$
$$y(a,t) = \alpha$$
$$y'(a,t) = t$$

retaining the prime notation to indicate differentiation with respect to x . Since we are interested in determining $(dy/dt)(b,t)$ when $t=t_{k-1}$, we first take the partial derivative of (1.7) with respect to t . This implies that

$$\begin{aligned}\frac{\partial y''}{\partial t}(x,t) &= \frac{\partial f}{\partial t}(x,y(x,t),y'(x,t)) \\ &= \frac{\partial f}{\partial x}(x,y(x,t),y'(x,t))\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}(x,y(x,t),y'(x,t))\frac{\partial y}{\partial t}(x,t) \\ &+ \frac{\partial f}{\partial y'}(x,y(x,t),y'(x,t))\frac{\partial y'}{\partial t}(x,t)\end{aligned}$$

or , since x and t are independent ,

$$\frac{\partial y''}{\partial t}(x,t)=\frac{\partial f}{\partial y}(x,y(x,t),y'(x,t))\frac{\partial y}{\partial t}(x,t)+\frac{\partial f}{\partial y'}(x,y(x,t),y'(x,t))\frac{\partial y'}{\partial t}(x,t) \quad ....(1.8)$$

for $a \le x \le b$. The initial conditions give

$$\frac{\partial y}{\partial t}(a,t)=0 \quad \text{and} \quad \frac{\partial y'}{\partial t}(a,t)=1 .$$

If we simplify the notation by using z(x,t) to denote $(\partial y/\partial t)(x,t)$ and assume that the order of differentiation of x and t can be reversed, eq.(1.8) becomes the initial-value problem

$$z''=\frac{\partial f}{\partial y}(x,y,y')z+\frac{\partial f}{\partial y'}(x,y,y')z' \quad a \le x \le b \quad ....(1.9)$$

$$z(a,t)=0 \ , \ z'(a,t)=1$$

Newton’s method therefore requires that two initial-value problems be solved for each iteration , eqs. (1.7) and (1.9) . Then from eq. (1.6)

$$t_k=t_{k-1}-\frac{y(b,t_{k-1})-\beta}{z(b,t_{k-1})} \quad ....(1.10)$$

In practice, none of these initial-value problems is likely to be solved exactly, instead approximation solutions are used .

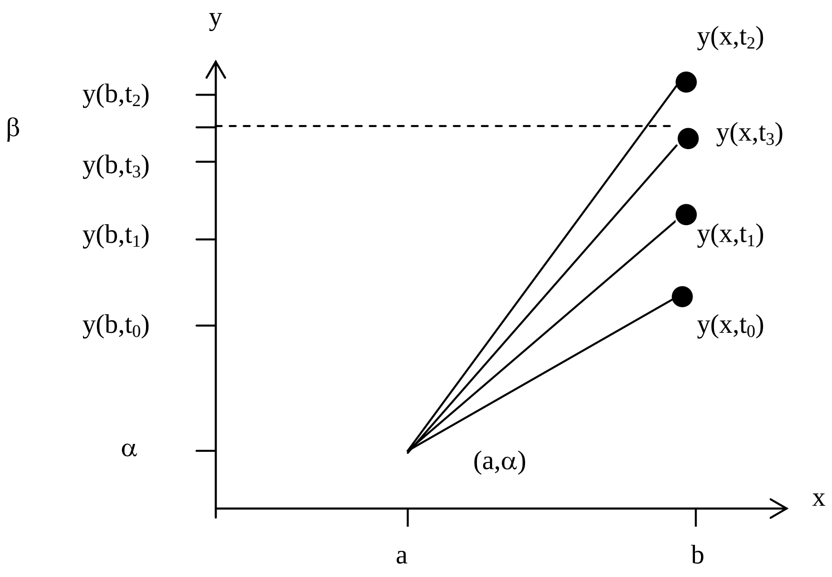

Fig. 2: Shooting technique after attempting to correct the approximation solution until y (b,t_k) is sufficiently close to “hitting” β .

Buchanan and Turner, 1992; Burden and Faires, 1993 and Ortega and Grimshaw, 1999. Have used fourth order Runge-Kutta method for systems to approximate both solutions required by Newton’s method ,like algorithm (1.1) below .

In order to speed-up the performance and obtain less time in the execute of nonlinear shooting method, we have constructed in this paper an algorithm, algorithm (2) using the Adams-Bashforth and Adams-Moulton predictor-corrector method for systems to approximate both solutions required by Newton's method .

1.1 **Algorithm** :

To approximate the solution of the nonlinear boundary-value problem

$$y'' = f(x, y, y') \ , \quad a \le x \le b \qquad y(a) = \alpha \ , \quad y(b) = \beta$$

(Note : Equations (1.7) and (1.9) are written as first order systems and solved.)

```
INPUT : endpoints a,b; boundary conditions $\alpha$, $\beta$; number of subintervals N; tolerance
        Tol; maximum number of iterations M .
OUTPUT : approximations $w_{1,i}$ to $y(x_i)$; $w_{2,i}$ to $y'(x_i)$ for each i = 0,1,…,N or a message
         that the maximum number of iterations was exceeded .
Step 1 : set h = (b-a) / N
             k = 1 ;
             Tk = ($\beta$ - $\alpha$) / (b – a)
Step 2 : while ($k \le M$) do steps 3-10
Step 3 : set $w_{1,0} = \alpha$   ($w_{1,0}$ represents y(a,t))
           $w_{2,0}$ = Tk   ($w_{2,0}$ represents $y'(a,t)$)
             $u_{1,0} = 0$    ($u_{1,0}$ represents z(a,t))
             $u_{2,0} = 1$    ($u_{2,0}$ represents $z'(a,t)$)
Step 4 : for i = 1,…,N do steps (5) and (6)
           (Runge-Kutta method for systems is used in steps (5) and (6))
Step 5 : set x = a + (i-1) h
Step 6 : $k_{1,1} = h\, w_{2,i-1}$
         $k_{1,2} = h\, f(x , w_{1,i-1} , w_{2,i-1})$
         $k_{2,1} = h\left(w_{2,i-1} + \frac{1}{2}k_{1,2}\right)$
         $k_{2,2} = hf\left(x + \frac{h}{2}, w_{1,i-1} + \frac{1}{2}k_{1,1}, w_{2,i-1} + \frac{1}{2}k_{1,2}\right)$
         $k_{3,1} = h\left(w_{2,i-1} + \frac{1}{2}k_{2,2}\right)$
         $k_{3,2} = hf\left(x + \frac{h}{2}, w_{1,i-1} + \frac{1}{2}k_{2,1}, w_{2,i-1} + \frac{1}{2}k_{2,2}\right)$
         $k_{4,1} = h(w_{2,i-1} + k_{3,2})$
         $k_{4,2} = hf(x + h, w_{1,i-1} + k_{3,1}, w_{2,i-1} + k_{3,2})$
         $w_{1,i} = w_{1,i-1} + (k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1})/6$
         $w_{2,i} = w_{2,i-1} + (k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2})/6$
```

($w_{1,i}$ and $w_{2,i}$ represent $y(x_i,t)$ and $y'(x_i,t)$ respectively for i = 1,…,N which are approximation solution of eq. (1.7)) .

$$k'_{1,1} = hu_{2,i-1}$$
$$k'_{1,2} = h\left[fy(x, w_{1,i-1}, w_{2,i-1})u_{1,i-1} + fy'(x, w_{1,i-1}, w_{2,i-1})u_{2,i-1}\right]$$
$$k'_{2,1} = h\left[u_{2,i-1} + \frac{1}{2}k'_{1,2}\right]$$
$$k'_{2,2} = h\begin{bmatrix} fy\left(x+\frac{h}{2}, w_{1,i-1}, w_{2,i-1}\right)\left(u_{1,i-1}+\frac{1}{2}k'_{1,1}\right) \\ +fy'\left(x+\frac{h}{2}, w_{1,i-1}, w_{2,i-1}\right)\left(u_{2,i-1}+\frac{1}{2}k'_{1,2}\right)\end{bmatrix}$$
$$k'_{3,1} = h\left(u_{2,i-1} + \frac{1}{2}k'_{2,2}\right)$$
$$k'_{3,2} = h\begin{bmatrix} fy\left(x+\frac{h}{2}, w_{1,i-1}, w_{2,i-1}\right)\left(u_{1,i-1}+\frac{1}{2}k'_{2,1}\right) \\ +fy'\left(x+\frac{h}{2}, w_{1,i-1}, w_{2,i-1}\right)\left(u_{2,i-1}+\frac{1}{2}k'_{2,2}\right)\end{bmatrix}$$
$$k'_{4,1} = h(u_{2,i-1} + k'_{3,2})$$
$$k'_{4,2} = h\begin{bmatrix} fy(x+h, w_{1,i-1}, w_{2,i-1})(u_{1,i-1}+k'_{3,1}) \\ +fy'(x+h, w_{1,i-1}, w_{2,i-1})(u_{2,i-1}+k'_{3,2})\end{bmatrix}$$
$$u_{1,i} = u_{1,i-1} + \frac{1}{6}\left[k'_{1,1} + 2k'_{2,1} + 2k'_{3,1} + k'_{4,1}\right]$$
$$u_{2,i} = u_{2,i-1} + \frac{1}{6}\left[k'_{1,2} + 2k'_{2,2} + 2k'_{3,2} + k'_{4,2}\right]$$

($u_{1,i}$ and $u_{2,i}$ represent $z(x_i,t)$ and $z'(x_i,t)$ respectively for $i = 1,\ldots,N$ which are approximation solution of eq. (1.9)) .

```
Step 7 : If $|w_{1,N} - \beta| \le Tol$ then do steps (8) and (9)
Step 8 : for i = 0,1,…,N
            Set $x = a + ih$
            OUTPUT $(x, w_{1,i}, w_{2,i})$
```

($w_{1,i}$ and $w_{2,i}$ in this step represent the approximation solutions to $y(x_i)$ and $y'(x_i)$ respectively for $i = 0,\ldots,N$) .

```
Step 9 : (procedure is complete .)
            Stop
Step 10 : set $Tk = Tk - \left(\frac{w_{1,N} - \beta}{u_{1,N}}\right)$
            (Newton's method is used to compute Tk)
            k=k+1
Step 11 : OUTPUT ('maximum number of iterations exceeded')
            (procedure completed unsuccessfully )
            stop .
```

Now in order to speed-up the performance of the nonlinear shooting method, we have suggested an improved algorithm that is defined as follows :

2. **The Improved Nonlinear Shooting Algorithm** :

To improve and obtain less time in the execute of the approximation solution of the boundary-value problem

$y'' = f(x, y, y')$, $a \le x \le b$

$$y(a) = \alpha \quad , \quad y(b) = \beta$$

we use linear multi-step methods for systems of predictor-corrector type .
(Note : Equations (1.7) and (1.8) are written as first-order systems and solved)
i.e, in order to apply our algorithm we must, first, convert a general 2nd-order differential equation of the form

$$y'' = f(x, y, y')$$

with initial conditions $y(a) = \alpha_1$, $y'(a) = \alpha_2$ into a system of equations in the form

$$u_1' = \frac{du_1}{dx} = f_1(x, u_1, u_2)$$

$$u_2' = \frac{du_2}{dx} = f_2(x, u_1, u_2)$$

for $a \le x \le b$, with initial conditions $u_1(a) = \alpha_1$, $u_2(a) = \alpha_2$
Now in general , to convert a general mth-order differential equation of the form

$$y^{(m)}(x) = f\left(x, y, y', \ldots\ldots, y^{(m-1)}\right) \qquad , a \le x \le b$$

with initial conditions $y(a) = \alpha_1, y'(a) = \alpha_2, \ldots, y^{(m-1)}(a) = \alpha_m$ into a system of equations in the form (*) and (**)

$$\left.\begin{aligned} u_1' &= \frac{du_1}{dx} = f_1(x, u_1, u_2, \ldots, u_m), \\ u_2' &= \frac{du_2}{dx} = f_2(x, u_1, u_2, \ldots, u_m), \\ &. \\ &. \\ &. \\ u_m' &= \frac{du_m}{dx} = f_m(x, u_1, u_2, \ldots, u_m) \end{aligned}\right\} \qquad \ldots\ldots\ldots.(*)$$

for $a \le x \le b$, with the initial conditions

$$u_1(a) = \alpha_1, u_2(a) = \alpha_2, \ldots, u_m(a) = \alpha_m \qquad \ldots\ldots\ldots.(**)$$

let $u_1(x) = y(x), u_2(x) = y'(x), \ldots, u_m(x) = y^{(m-1)}(x)$.
and using this notation, we obtain the first-order system

$$\frac{du_1}{dx} = \frac{dy}{dx} = u_2$$
$$\frac{du_2}{dx} = \frac{dy'}{dx} = u_3$$
.
.
.
$$\frac{du_{m-1}}{dx} = \frac{dy^{(m-2)}}{dx} = u_m$$
and $\frac{du_m}{dx} = \frac{dy^{(m-1)}}{dx} = y^{(m)} = f\left(x, y, y', \ldots\ldots, y^{(m-1)}\right) = f\left(x, u_1, u_2, \ldots, u_m\right)$

with initial conditions
$u_1(a) = y(a) = \alpha_1, u_2(a) = y'(a) = \alpha_2, \ldots, u_m(a) = y^{(m-1)}(a) = \alpha_m$

and as a special case, algorithm (2) is done when m = 2, and then the out-line of the improved algorithm is as follows :

INPUT : endpoints a,b; boundary conditions α, β; number of subintervals N; tolerance Tol; maximum number of iterations M .

OUTPUT : approximations $w_{1,i}$ to $y(x_i)$; $w_{2,i}$ to $y'(x_i)$ for each i = 0,1,…,N or a message that the maximum number of iterations was exceeded .

```
Step 1 : set h = (b-a) / N
                k = 1 ;
                Tk = (β - α) / (b – a)
Step 2 : while (k ≤ M) do steps (3)-(16)
Step 3 : set $w_{1,0} = \alpha$    ($w_{1,0}$ represents y(a,t))
             $w_{2,0} = Tk$   ($w_{2,0}$ represents $y'$(a,t))
              $u_{1,0} = 0$    ($u_{1,0}$ represents z(a,t))
               $u_{2,0} = 1$    ($u_{2,0}$ represents $z'$(a,t))
               $x_0 = a$
Step 4 : for i = 1,2,3 do steps (5) and (6)
          (compute the starting value using Runge-Kutta method for systems)
Step 5 : $k_{1,1} = h\, w_{2,i-1}$
         $k_{1,2} = h\, f(x, w_{1,i-1}, w_{2,i-1})$
          $k_{2,1} = h\left(w_{2,i-1} + \frac{1}{2}k_{1,2}\right)$
          $k_{2,2} = hf\left(x + \frac{h}{2}, w_{1,i-1} + \frac{1}{2}k_{1,1}, w_{2,i-1} + \frac{1}{2}k_{1,2}\right)$
```

$$k_{3,1} = h\left(w_{2,i-1} + \frac{1}{2}k_{2,2}\right)$$

$$k_{3,2} = hf\left(x + \frac{h}{2}, w_{1,i-1} + \frac{1}{2}k_{2,1}, w_{2,i-1} + \frac{1}{2}k_{2,2}\right)$$

$$k_{4,1} = h(w_{2,i-1} + k_{3,2})$$

$$k_{4,2} = hf(x + h, w_{1,i-1} + k_{3,1}, w_{2,i-1} + k_{3,2})$$

$$w_{1,i} = w_{1,i-1} + (k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1})/6$$

$$w_{2,i} = w_{2,i-1} + (k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2})/6$$

($w_{1,i}$ and $w_{2,i}$ represent $y(x_i,t)$ and $y'(x_i,t)$ respectively which are the approximation solution of eq, (1.7) for i = 1,2,3 for Runge-Kutta method for systems) .

$$k'_{1,1} = hu_{2,i-1}$$

$$k'_{1,2} = h[fy(x, w_{1,i-1}, w_{2,i-1})u_{1,i-1} + fy'(x, w_{1,i-1}, w_{2,i-1})u_{2,i-1}]$$

$$k'_{2,1} = h\left[u_{2,i-1} + \frac{1}{2}k'_{1,2}\right]$$

$$k'_{2,2} = h\left[\begin{array}{l} fy\left(x + \frac{h}{2}, w_{1,i-1}, w_{2,i-1}\right)\left(u_{1,i-1} + \frac{1}{2}k'_{1,1}\right) \\ + fy'\left(x + \frac{h}{2}, w_{1,i-1}, w_{2,i-1}\right)\left(u_{2,i-1} + \frac{1}{2}k'_{1,2}\right) \end{array}\right]$$

$$k'_{3,1} = h\left(u_{2,i-1} + \frac{1}{2}k'_{2,2}\right)$$

$$k'_{3,2} = h\left[\begin{array}{l} fy\left(x + \frac{h}{2}, w_{1,i-1}, w_{2,i-1}\right)\left(u_{1,i-1} + \frac{1}{2}k'_{2,1}\right) \\ + fy'\left(x + \frac{h}{2}, w_{1,i-1}, w_{2,i-1}\right)\left(u_{2,i-1} + \frac{1}{2}k'_{2,2}\right) \end{array}\right]$$

$$k'_{4,1} = h(u_{2,i-1} + k'_{3,2})$$

$$k'_{4,2} = h\left[\begin{array}{l} fy(x + h, w_{1,i-1}, w_{2,i-1})(u_{1,i-1} + k'_{3,1}) \\ + fy'(x + h, w_{1,i-1}, w_{2,i-1})(u_{2,i-1} + k'_{3,2}) \end{array}\right]$$

$$u_{1,i} = u_{1,i-1} + \frac{1}{6}[k'_{1,1} + 2k'_{2,1} + 2k'_{3,1} + k'_{4,1}]$$

$$u_{2,i} = u_{2,i-1} + \frac{1}{6}[k'_{1,2} + 2k'_{2,2} + 2k'_{3,2} + k'_{4,2}]$$

($u_{1,i}$ and $u_{2,i}$ represent $z(x_i,t)$ and $z'(x_i,t)$ respectively which are the approximation solution of eq. (1.9) for i = 1,2,3 , i.e for Runge-Kutta method for systems) .

Step 6 : Set x = a + ih

OUTPUT (xi,$w_{1,i}$, $w_{2,i}$,$u_{1,i}$,$u_{2,i}$)

Step 7 : for i = 4 , …..,N do step (8)-(12)

Step 8 : set x = a + ih

Step 9 : $w_{1,i} = w_{1,3} + h[55w_{2,3} - 59w_{2,2} + 37w_{2,1} - 9w_{2,0}]/24$

$$w_{2,i} = w_{2,3} + h\begin{bmatrix} 55f(x_3, w_{1,3}, w_{2,3}) - 59f(x_2, w_{1,2}, w_{2,2}) \\ +37f(x_1, w_{1,1}, w_{2,1}) - 9f(x_0, w_{1,0}, w_{2,0}) \end{bmatrix} / 24$$

$$u_{1,i} = u_{1,3} + h[55u_{2,3} - 59u_{2,2} + 37u_{2,1} - 9u_{2,0}] / 24$$

$$u_{2,i} = u_{2,3} + h\begin{bmatrix} 55(fy(x_3, u_{1,3}, u_{2,3})u_{1,3} + fy'(x_3, u_{1,3}, u_{2,3})u_{2,3}) \\ -59(fy(x_2, u_{1,2}, u_{2,2})u_{1,2} + fy'(x_2, u_{1,2}, u_{2,2})u_{2,2}) \\ +37(fy(x_1, u_{1,1}, u_{2,1})u_{1,1} + fy'(x_1, u_{1,1}, u_{2,1})u_{2,1}) \\ -9(fy(x_0, u_{1,0}, u_{2,0})u_{1,0} + fy'(x_0, u_{1,0}, u_{2,0})u_{2,0}) \end{bmatrix} / 24$$

(predictor $w_{1,i}$, $w_{2,i}$, $u_{1,i}$, $u_{2,i}$)

($w_{1,i}$ and $w_{2,i}$ represent $y(x_i,t)$ and $y'(x_i,t)$ respectively which are the approximation solution of eq, (1.7) for i = 4,…..,N , by using the predictor form of Adam's method for systems) .

($u_{1,i}$ and $u_{2,i}$ represent $z(x_i,t)$ and $z'(x_i,t)$ respectively which are the approximation solution of eq. (1.9) for i = 4,…..,N , by using the predictor form of Adam's method for systems) .

$$ww_{1,i} = w_{1,3} + h[9w_{2,i} + 19w_{2,3} - 5w_{2,2} + w_{2,1}] / 24$$

$$ww_{2,i} = w_{2,3} + h\begin{bmatrix} 9f(x_i, w_{1,i}, w_{2,i}) + 19f(x_3, w_{1,3}, w_{2,3}) \\ -5f(x_2, w_{1,2}, w_{2,2}) + f(x_1, w_{1,1}, w_{2,1}) \end{bmatrix} / 24$$

$$uu_{1,i} = u_{1,3} + h[9u_{2,i} + 19u_{2,3} - 5u_{2,2} + u_{2,1}] / 24$$

$$uu_{2,i} = u_{2,3} + h\begin{bmatrix} 9(fy(x_i, u_{1,i}, u_{2,i})u_{1,i} + fy'(x_i, u_{1,i}, u_{2,i})u_{2,i}) \\ +19(fy(x_3, u_{1,3}, u_{2,3})u_{1,3} + fy'(x_3, u_{1,3}, u_{2,3})u_{2,3}) \\ -5(fy(x_2, u_{1,2}, u_{2,2})u_{1,2} + fy'(x_2, u_{1,2}, u_{2,2})u_{2,2}) \\ +(fy(x_1, u_{1,1}, u_{2,1})u_{1,1} + fy'(x_1, u_{1,1}, u_{2,1})u_{2,1}) \end{bmatrix} / 24$$

(corrector $ww_{1,i}$, $ww_{2,i}$, $uu_{1,i}$, $uu_{2,i}$)

($ww_{1,i}$ and $ww_{2,i}$ represent $y(x_i,t)$ and $y'(x_i,t)$ respectively which are the approximation solution of eq. (1.7) for i = 4,…..,N , by using the corrector form of Adam's method for systems) .

($uu_{1,i}$ and $uu_{2,i}$ represent $z(x_i,t)$ and $z'(x_i,t)$ respectively which are the approximation solution of eq. (1.9) for i = 4,…..,N , by using the corrector form of Adam's method for systems) .

```
Step 10 : OUTPUT ($x_i$ , $ww_{1,i}$ , $ww_{2,i}$ , $uu_{1,i}$ , $uu_{2,i}$)
Step 11 : for j = 0,1,2
        Set $x_j = x_{j+1}$
            $w_{1,j} = w_{1,j+1}$
            $w_{2,j} = w_{2,j+1}$
            $u_{1,j} = u_{1,j+1}$
            $u_{2,j} = u_{2,j+1}$
Step 12 : $x_3 = x$
          $w_{1,3} = ww_{1,i}$
          $w_{2,3} = ww_{2,i}$
          $u_{1,3} = uu_{1,i}$
          $u_{2,3} = uu_{2,i}$
Step 13 : If $|ww_{1,N} - \beta| \le Tol$ then do steps (14) and (15)
Step 14 : For i = 0,…..,N
          Set x = a + ih
          OUTPUT (x , $ww_{1,i}$ , $ww_{2,i}$)
Step 15 : (procedure is complete)
          Stop .
Step 16 : $Tk = Tk - \left(\frac{ww_{1,N} - \beta}{uu_{1,N}}\right)$
          (Newton's method is used to compute Tk)
          k=k+1
Step 17 : OUTPUT ('maximum number of iterations exceeded')
          (procedure completed unsuccessfully )
          stop .
```

Note that : In step (13) the best approximation to β we can expect for $ww_{1,N}(t_k)$ is $o(h^n)$.

The value t_0 selected in step (1) is the slop of the straight line through (a,α) and (b,β) .

If we give a good choice of t_0 , the convergence will improve and the procedure will work for any problem .

3. Numerical Examples :

In this section we will present some numerical examples to show that the performance of the suggested algorithm is quick and has better performance since it requires less time to execute computed by using a program written with a special compiler , which had used to compute the time of execution .

Fortran 90 program is written to carry out the experiment and the programs are executed on Pentium (PIT) computers .

Note that we had used to determine the time of execute by using the compiler : Microsoft power station Fortran , and also executed on Pentium (PIT) computers with processor speed 133 M.H.

Example 1 :

Represent the results of algorithm (1.1) and the results of the improved algorithm (2) and the time of the execution , applied to the example :

$$y'' = \frac{1}{2}y^3 \quad ,1 \le x \le 2 \ , \ y(1) = \frac{-2}{3} \ , \ y(2) = -1 \ , \ N = 20 \ , \ H = 0.05$$

where this boundary-value problem has the exact solution

$$y = \frac{2}{x-4}$$

Table 1 a :This table represents some selected results of algorithm (1.1) and the improved algorithm (2) applied to example (1) .

Selected value of x	The exact value of y(x)	The value of $w_{1,i}$ RK +nonlinear shooting	The value of $w_{1,i}$ Adam+ nonlinear shooting	The value of $w_{2,i}$ R-K +nonlinear shooting	The value of $w_{2,i}$ Adam+ Nonlinear shooting
1.00	-6.6666E-01	-6.6666E-01	-6.6666E-01	-2.222774E-01	-2.222774E-01
1.05	-6.779661E-01	-6.779622E-01	-6.779622E-01	-2.298740E-01	-2.298740E-01
1.10	-6.8965517E-01	-6.89654E-01	-6.89654E-01	-2.37867E-01	-2.37867E-01
1.15	-7.0175438E-01	-7.017559E-01	-7.017559E-01	-2.462845E-01	-2.462845E-01
1.35	-7.5471698E-01	-7.547296E-01	-7.546958E-01	-2.848549E-01	-2.847551E-01
1.50	-7.99999E-01	-8.000212E-01	-7.9999720E-01	-3.200583E-01	-3.199530E-01
1.80	-9.0909090E-01	-9.091309E-01	-9.090474E-01	-4.132915E-01	-4.131646E-01
1.90	-9.5238095E-01	-9.524280E-01	-9.523313E-01	-4.535888E-01	-4.534502E-01
2.00	-1.000000	-1.000055	-9.999436E-01	-5.001324E-01	-4.999280E-01

Table 1 – b :

The time of the execute , SEC.	
Runge-Kutta for system+nonlinear shooting	Adam for system + nonlinear shooting
11.2601530 SEC.	6.1276854 SEC.

Example 2 :

Present the results of algorithm (1.1) and the results of the improved algorithm (2) and the time of the execution , applied to the example :

$$y'' = \frac{1}{8}\left(32 + 2x^3 - yy'\right) ,\ 1 \le x \le 3 \ ,\ y(1) = 17 \ ,\ y(3) = \frac{43}{3} = 14.33333$$

$$N = 20 \ ,\ H = 0.1$$

where the boundary-value problem has the exact solution

$$y = x^2 + \frac{16}{x}$$

Table 2 – a :This table presents some selected results of algorithm (1.1) and the improved algorithm (2) applied to example (2) .

Selected value of x	The exact value of y(x)	The value of $w_{1,i}$ R-K +nonlinear shooting	The value of $w_{1,i}$ Adam+ nonlinear shooting	The value of $w_{2,i}$ R-K +nonlinear shooting	The value of $w_{2,i}$ Adam+ nonlinear shooting
1.1	15.755455	15.755490	15.755490	-11.023390	-11.023390
1.2	14.773333	14.773380	14.773380	-8.711340	-8.711340
1.3	13.997692	13.997740	13.997740	-6.867657	-6.867657
1.7	12.301765	12.301780	12.300810	-2.136448	-2.134655
2.0	11.999999	12.00023	11.999931	-7.692973E-05	1.34542E-03
2.5	12.650000	12.649970	12.64974	2.439964	2.440749
2.9	13.927241	13.927200	13.927200	3.897488	3.897926
3.0	14.333333	14.33329	14.33332	4.222212	4.222581

Table 2 – b :

The time of the execute , SEC.	
Runge-Kutta for system+nonlinear shooting	Adam for system + nonlinear shooting
12.0015372 SEC.	6.9864731 SEC.

Examples (1) and (2) with their tables indicate that the nonlinear shooting method, which used Adams predictor-corrector method for systems, will always require less time to execute , though possibly with some loss in accuracy .

So it is preferable to the nonlinear shooting method that used single-step (Runge-Kutta method for systems) to approximate the solution of the IVPs .

Also the fact that the Adams method that is used in this method requires only two function evaluations whereas the classical Runge-Kutta method requires four evaluations, which make it more efficient .

REFERENCES

Alt, R. and Vignes, J., 1996. Validation of results of collocation methods for ODEs with the CADNA library, Applied Numerical Mathematics, 21, 1996, pp.119-139 .

Al-Assady, N.H. and Al-Sawoor, A.J., 2000. Improving the performance of linear shooting method for solving linear BVPs , Journal of Rafidain Science Forth coming ,No. 4 , Vol. (12),2000, 107-118.

Buchanan, J.L. and Turner, P.R., 1992. Numerical Methods And Analysis,McGraw-Hill,Inc.

Burden, R.L. and Faires, J.D., 1993. Numerical Analysis,PWS-KENT Publishing Company .

Conte, S.D. and de Boor, C.D., 1980. Elementary Numerical Analysis,An algorithmic Approach-third Edition,Mc Graw-Hill international Book company .

Fox, L., and Mayers, D.F., 1987. Numerical solution of ordinary differentioa equation,Chapman and Hall .

Jain, M.K. and Iyengar, S.R.K., 1994. Numerical methods,Wiley Eastern Limited ,Publishing for one world .

Khalaf, B.M.S., 1988. Boundary value techniques in continuous simulation, M phil thesis ,University of Bradford,England.

Ortega, J.M. and Grimshaw, A.S.,1999. An Introduction to C^{++} And Numerical Method, Oxford University Press,New York.

Stoer, J. and Bulirsch, R.,1980. Introduction to Numerical Analysis, Springer-Verlag, New York Heiddbery Berlin .