Application of the shooting method for the solution of second order boundary value problems

Ifeoluwa F. Edun¹, Grace O. Akinlabi¹*

¹Department of Mathematics, Covenant University, Canaanland, Ota, Nigeria

*Corresponding Author’s e-mail: grace.akinlabi@covenantuniversity.edu.ng

Abstract. Due to the applications of Boundary Value Problems (BVPs) in real-life phenomena, the shooting method has proven itself useful and efficient in handling BVP. The shooting method works by first reducing the BVP to an Initial Value Problem (IVP), then one/two initial value guesses are made. The IVPs are then solved using an iterative solution, and this process is then repeated until the second boundary condition is reached to a satisfactory level. The iterative formula used in this study is the Euler’s method while the initial value estimation method used is the Secant method (interpolation formula).

Keywords: Shooting Method, Boundary Value Problem, Initial Value Problem, Ordinary Differential Equation

1. Introduction
A Boundary Value Problem (BVP) is one of the types of Ordinary Differential Equation (ODE) with derivative roots identified at more than one point. BVPs are useful in the design of mechanics such as pivots and bearings and are usually solved by the Shooting Method (SHM) and the Standard Finite Difference Methods (SFDMs). The SHM is a recognized technique used to find solutions to BVP, usually by reducing the BVP to Initial Value Problems (IVPs) [1]. In this study, we will describe the shooting method and its use in solving second-order BVPs. Most research studies tend to concentrate on IVPs with Ordinary Differential Problems using numerical methods such as Euler's method, Runge-Kutta method, and so on [2-3].

The shooting approach can be very effective on basic issues like the projectile problem. It can be conveniently expanded to propose a solution approach for almost any boundary value problem based on its boundary conditions and has been implemented in several pieces of mathematical tools. The performance of the SHM highly depends on the stability of the IVP formed [4]. The SHM was studied, and a Modified Simple Shooting Method (MSSM) was proposed for solving two-point boundary value problems in [5]. In the study according to [6], an iterative formula was created that changes BVP to IVP and then solving the nonlinear BVP using the SHM and interpolation; also comparisons of solutions were made from the proposed SHM and other methods such as Euler method and the Runge-Kutta method. It was discovered that the blend of the SHM and the Newton's method provides a very efficient tool [7]. In [8], the authors made way for difficulties in solving the BVPs and also concluded that the SHM is the best and easiest way to solve BVPs. The multiple SHM was also illustrated using several examples in [9]. Clearly stated in the article, according to [10], are the existence and uniqueness of the solutions of BVP. The BVP to be considered in this work is of the form:
\[ y'' + r(x)y = f(x), \quad a \leq t \leq b \]
\[ y(a) = A, \quad y(b) = B \tag{1} \]

2. Methodology
In this section, an outline of the conventional SHM is shown.

2.1. The Shooting Method
Considering a second order BVP
\[
\begin{aligned}
&y'' = f(x, y(x), y'(x)) \\
y(a) = \alpha, \quad y(b) = \beta 
\end{aligned}
\] on the interval \([a, b]\) \tag{2}
One of the conditions is: \(y(a) = \alpha\), if we assume another condition: \(y'(a) = z\), which illustrates the slope. The resulting IVP then becomes
\[
\begin{aligned}
&y'' = f(x, y(x), y'(x)) \\
y(a) = \alpha, \quad y'(a) = z
\end{aligned}
\] \tag{3}
Let \(\varphi(z) = \beta\) \tag{4}
The main objective is to adjust \(z\) until the value agrees with \(\varphi(z) = \beta\) to a desired accuracy. First, we make an initial guess for the value of \(y'(a)\), assuming it to be \(z_0\); then we solve the resulting IVP using either Taylor’s series, Runge-Kutta or Euler method. The first resulting IVP will be of the form:
\[
\begin{aligned}
&y'' = f(x, y(x), y'(x)) \\
y(a) = \alpha, \quad y'(a) = z_0
\end{aligned}
\] \tag{5}
We carry out the resulting solution of the IVP as far as the value, \(b\) and hope that the solution is in agreement with the value, \(\beta\). If this is not so, we assume another guess for \(y'(a)\), say \(z_1\); we solve the resulting second IVP using the same process for the first IVP. Our second resulting IVP then becomes:
\[
\begin{aligned}
&y'' = f(x, y(x), y'(x)) \\
y(a) = \alpha, \quad y'(a) = z_1
\end{aligned}
\] \tag{6}
We do this until we hit the target value, \(\beta\). It is good to always take note that the final value of \(\varphi(z)\) depends on the guess we make for \(y'(a)\). Assuming the values \(z_0\) and \(z_1\) are linearly correlated, to achieve a better value \(z_2\) on the basis of the first and second guesses, we use the estimating interpolation formula:
\[
\frac{z_2 - z_1}{y(b) - \varphi(z_1)} = \frac{z_1 - z_0}{\varphi(z_1) - \varphi(z_0)} \tag{7}
\]
From which we obtain
\[
z_2 = z_1 + \left(\frac{z_1 - z_0}{\varphi(z_1) - \varphi(z_0)}\right)[y(b) - \varphi(z_1)] \tag{8}
\]
A repetition of this process causes the standard formula to be
\[ z_{n+1} = z_n + \left( \frac{z_n - z_{n-1}}{\varphi(z_n) - \varphi(z_{n-1})} \right) \left[ y(b) - \varphi(z_n) \right] \quad (n \geq 1) \quad (9) \]

all established on the starting values \( z_0 \) and \( z_1 \). We then solve the resulting IVP created on the basis of \( z_2 \) to obtain \( \varphi(z_2) \). To obtain a sharper accuracy, \( z_3 \), we can again use the linear relation (9) with \([z_1, \varphi(z_1)] \) and \([z_2, \varphi(z_2)] \). We observe the value of \( \varphi(z_{n+1}) - y(b) \) so that we can ensure improvement is made that is, the error bound reduce drastically. When it is reduced to a certain level of contentment, stop. Note that all numerically obtained values of \( \varphi(z_n) \) must be saved until closer values reaching the target are obtained. The SHM is very time consuming if each solution of the related IVP involves a small step size \( h \). Thus, we use a relatively large value of \( h \) until we observe that \( |\varphi(z_{n+1}) - y(b)| \) is satisfactorily small. After this, we can now reduce the value of \( h \) to obtain higher accuracy.

3. Numerical Examples

In this section, we shall be discussing the application of the SHM using the Euler’s method, alongside the Secant method in obtaining solutions of two BVPs. Also, comparisons of the solutions with exact solution shall be made using tables. The cases considered in the work are from [11], [12].

**Case I:** Consider the boundary value problem

\[ y'' = 2y^3 - 6y - 2x^3, \quad 1 \leq x \leq 2 \quad (11) \]

With boundary conditions: \( y(1) = 2, \quad y(2) = 2.5 \)

For the numerical solution, let step size \( h = \frac{b-a}{4} = \frac{2-1}{4} = 0.25 \)

The nodes are computed by \( x_{i+1} = x_i + h \)

\[ \therefore x_0 = 1, \quad x_1 = 1.25, \quad x_2 = 1.5, \quad x_3 = 1.75, \quad x_4 = 2 \]

Let \( y'(1) = \alpha \)

Initial guesses: \( \alpha_0 = 4, \quad \alpha_1 = 1.5 \)

Reduction to two IVPs yield

**IVP1:**
\[ y'' = 2y^3 - 6y - 2x^3 \]
\[ y(1) = 2, \quad y'(1) = 4 \]

**IVP2:**
\[ y'' = 2y^3 - 6y - 2x^3 \]
\[ y(1) = 2, \quad y'(1) = 1.5 \]

Let \( \frac{dy}{dx} = z \) and \( z' = 2y^3 - 6y - 2x^3 \)

Using Euler’s method,

\[ y' = f_1(x, y, z) = z \]
\[ z' = f_2(x, y, z) = 2y^3 - 6y - 2x^3 \]
\[ y_{i+1} = y_i + h f_1(x_i, y_i, z_i) \\
= y_i + h z_i \\
z_{i+1} = z_i + h f_2(x_i, y_i, z_i) \\
= z_i + h (2y_i^3 - 6y_i - 2x_i^3) \]

**From IVP1:** \( y_0 = 2, \ x_0 = 1, \ z_0 / \alpha_0 = 4, \ h = 0.25 \)

\[ y_1 = y_0 + h z_0 \\
= 2 + 0.25(4) = 3 \]
\[ z_1 = z_0 + h(2y_0^3 - 6y_0 - 2x_0^3) \\
= 4 + 0.25 \left( 2(2)^3 - 6(2) - 2(1)^3 \right) = 4.5 \]
\[ y_2 = y_1 + h z_1 \\
= 3 + 0.25(4.5) = 4.125 \]
\[ z_2 = z_1 + h(2y_1^3 - 6y_1 - 2x_1^3) \\
= 4.5 + 0.25 \left( 2(3)^3 - 6(3) - 2(1.25)^3 \right) = 12.52344 \]
\[ y_3 = y_2 + h z_2 \\
= 4.125 + 0.25(12.52344) = 7.25586 \]
\[ z_3 = z_2 + h(2y_2^3 - 6y_2 - 2x_2^3) \\
= 12.52344 + 0.25 \left( 2(4.125)^3 - 6(4.125) - 2(1.5)^3 \right) = 39.74317 \]
\[ y_4 = y_3 + h z_3 \\
= 7.25586 + 0.25(39.74317) = 17.19165 \]

**From IVP2:** \( y_0 = 2, \ x_0 = 1, \ z_0 / (\alpha_i) = 1.5, \ h = 0.25 \)

\[ y_1 = y_0 + h z_0 \\
= 2 + 0.25(1.25) = 2.375 \]
\[ z_1 = z_0 + h(2y_0^3 - 6y_0 - 2x_0^3) \\
= 1.5 + 0.25 \left( 2(2)^3 - 6(2) - 2(1)^3 \right) = 2 \]
\[ y_2 = y_1 + h z_1 \\
= 2.375 + 0.25(2) = 2.875 \]
\[ z_2 = z_1 + h(2y_1^3 - 6y_1 - 2x_1^3) \\
= 2 + 0.25 \left( 2(2.375)^3 - 6(2.375) - 2(1.25)^3 \right) = 4.15918 \]
\[ y_3 = y_2 + h z_2 \\
= 2.875 + 0.25(4.15918) = 3.9148 \]
\[ z_3 = z_2 + h(2y_2^3 - 6y_2 - 2x_2^3) \\
= 4.15918 + 0.25 \left( 2(2.875)^3 - 6(2.875) - 2(1.5)^3 \right) = 10.04102 \]
\[ y_4 = y_3 + hz_3 \]
\[ = 3.9148 + 0.25(10.04102) = 6.42505 \]

At \( x_4 = 2 \), we have \( y_4 = 6.42505 \), while our target is \( \beta = 2.5 \).

We notice that both our initial guesses give values which are not close to our target \( \beta \), satisfactorily.

Using Secant method to deduce a more approximate value for \( y'(0) \), \( \alpha_2 \):

\[
\alpha_k = \alpha_{k-1} - \frac{(y(b, \alpha_{k-1}) - \beta)(\alpha_{k-1} - \alpha_{k-2})}{y(b, \alpha_{k-1}) - y(b, \alpha_{k-2})}
\]
\[
\alpha_2 = \alpha_1 - \frac{(y(b, \alpha_1) - \beta)(\alpha_1 - \alpha_0)}{y(b, \alpha_1) - y(b, \alpha_0)}
\]
\[ = 1.5 - \frac{(6.42505 - 2.5)(1.5 - 4)}{6.42505 - 17.19165} \]
\[ = 0.59 \]

From IVP3: \( y_0 = 2, \ x_0 = 1, \ z_0 / (\alpha_2) = 0.59, \ h = 0.25 \)

\( y_1 = y(1.25) = 2.14725 \)
\( z_1 = 1.09 \)
\( y_2 = y(1.5) = 2.41979 \)
\( z_2 = 1.84292 \)
\( y_3 = y(1.75) = 2.88052 \)
\( z_3 = 3.61013 \)
\( y_4 = y(2) = 3.78305 \)

At \( x_4 = 2 \), we have \( y_4 = 3.78305 \), while our target is \( \beta = 2.5 \).

We notice that the result obtained from the last iteration gave us a value that is close to the actual value \( y(2) \) but we are not satisfied with this result so we use the secant method to deduce another approximate value, \( \alpha_3 \):

\[
\alpha_3 = \alpha_2 - \frac{(y(b, \alpha_2) - \beta)(\alpha_2 - \alpha_1)}{y(b, \alpha_2) - y(b, \alpha_1)}
\]
\[ = 0.59 - \frac{(3.78305 - 2.5)(0.59 - 1.5)}{3.78305 - 6.42505} \]
\[ = 0.148 = 0.12 \]

From IVP4: \( y_0 = 2, \ x_0 = 1, \ z_0 / (\alpha_3) = 0.12, \ h = 0.25 \)

\( y_1 = y(1.25) = 2.03 \)
\( z_1 = 0.62 \)
\( y_2 = y(1.5) = 2.185 \)
\( z_2 = 0.78115 \)
\( y_3 = y(1.75) = 2.38029 \)
\( z_3 = 1.032 \)
At $x_4 = 2$, we have $y_4 = 2.6382$, while our target is $\beta = 2.5$. The result obtained is not still satisfactory. Using the Secant formula, we obtain an initial value approximation that give a closer result to the actual value of $y(2)$:

$$
\alpha_4 = \alpha_3 - \frac{(y(b, \alpha_3) - \beta) (\alpha_3 - \alpha_2)}{y(b, \alpha_3) - y(b, \alpha_2)}
$$

$$
= 0.12 - \frac{(2.6382 - 2.5) (0.12 - 0.59)}{2.6382 - 3.78305}
$$

$$
= 0.0632 \approx 0.06
$$

From IVP5: $y_0 = 2, x_0 = 1, z_0/(\alpha_4) = 0.12, h = 0.25$

$$
y_1 = y(1.25) = 2.015
$$

$$
z_1 = 0.56
$$

$$
y_2 = y(1.5) = 2.155
$$

$$
z_2 = 0.65161
$$

$$
y_3 = y(1.75) = 2.3179
$$

$$
z_3 = 0.73554
$$

$$
y_4 = y(2) = 2.50179
$$

We are now satisfied with the obtained result as it gives a very low error bound and a high accuracy. Table 1 shows the comparison of results obtained using the Euler’s method with the last initial guess and the exact solution.

**Table 1**: Comparison of the Exact solution and the shooting method (using Euler) for Case I

| $i$ | $x_i$ | Euler ($\alpha_4 = 0.06$) | Exact | $|e|$ | $z_i$ |
|-----|-------|--------------------------|-------|------|------|
| 0   | 1     | 2.00000                  | 2.0000 | 0.0000 | 0.0600 |
| 1   | 1.25  | 2.01500                  | 2.0500 | 0.0350 | 0.5600 |
| 2   | 1.5   | 2.15500                  | 2.1667 | 0.0117 | 0.6516 |
| 3   | 1.75  | 2.31790                  | 2.3214 | 0.0035 | 0.7355 |
| 4   | 2     | 2.50179                  | 2.5000 | 0.0017 |      |

**Case II**: A thick pressure vessel is being tested in a science laboratory to check its ability to endure pressure. It has an inner radius $a$ and an outer radius $b$. The differential equation of the radial displacement $u$ of a point is given as

$$
u''(r) + \frac{1}{r}u'(r) - \frac{u(r)}{r^2} = 0
$$

$$
\alpha = 5, \quad b = 8
$$

With boundary conditions

$$
u(5) = 0.0038731, \quad u(8) = 0.0030770
$$

(12)
Let \( u' (5) = \alpha \) and assume our initial guesses to be \( \alpha_0 = 2.5, \quad \alpha_1 = 1 \)

Reduction to two IVPs:

**IVP1:**
\[
\frac{d^2u}{dr^2} = \frac{u'}{r^2} - \frac{u}{r}
\]
\[
0.0038731, \quad 2.5
\]

**IVP2:**
\[
\frac{d^2u}{dr^2} = \frac{u'}{r^2} - \frac{u}{r}
\]
\[
0.0038731, \quad 1
\]

Let \( \frac{du}{dr} = z \), \( u' = z \), \( z' = \frac{u}{r^2} - \frac{z}{r} \)

Using Euler’s method,

\[
u'(r, u, z) = z
\]
\[
z'(r, u, z) = \frac{u}{r^2} - \frac{z}{r}
\]
\[u_{i+1} = u_i + h f_1(r_i, u_i, z_i)
\]
\[z_{i+1} = z_i + h f_2(r_i, u_i, z_i)
\]

From IVP1: \( u_0 = 0.0038731, \quad z_0 = 2.5, \quad r_0 = 5, \quad h = 0.75 \)

\[
u_1 = u_0 + h z_0 = 0.0038731 + 0.75(2.5) = 1.87887
\]
\[
z_1 = z_0 + h \left( \frac{u_0}{r_0^2} - \frac{z_0}{r_0} \right) = 2.5 + 0.75 \left( \frac{0.0038731}{5^2} - \frac{2.5}{5} \right) = 2.12511
\]
\[
u_2 = u_1 + h z_1 = 1.87887 + 0.75(2.12511) = 3.47271
\]
\[
z_2 = z_1 + h \left( \frac{u_1}{r_1^2} - \frac{z_1}{r_1} \right) = 2.12511 + 0.75 \left( \frac{1.87887}{5.75^2} - \frac{2.12511}{5.75} \right) = 1.89055
\]
\[
u_3 = u_2 + h z_2 = 3.47271 + 0.75(1.89055) = 4.89062
\]
\[
z_3 = z_2 + h \left( \frac{u_2}{r_2^2} - \frac{z_2}{r_2} \right) = 1.89055 + 0.75 \left( \frac{3.47271}{6.5^2} - \frac{1.89055}{6.5} \right) = 1.73405
\]
\[
u_4 = u_3 + h z_3 = 4.89062 + 0.75(1.73405) = 5.75765
\]

At \( r_4 = 8 \), we have \( u_4 = 5.75648 \), while our target is \( \beta = 0.0030770 \).

From IVP2: \( u_0 = 0.0038731, \quad z_0 = 1, \quad r_0 = 5, \quad h = 0.75 \)

\[
u_1 = u(5.75) = 0.75387
\]
\[z_1 = 0.85012
\]
\begin{align*}
  u_2 &= u(6.5) = 1.39146 \\
  z_2 &= 0.75633 \\
  u_3 &= u(7.25) = 1.95871 \\
  z_3 &= 0.69376 \\
  u_4 &= u(8) = 2.47903
\end{align*}

Using the results obtained from the two previous iterations, we can obtain a better approximation, \( \alpha_2 \) of the initial guess, \( u'(5) \) using the linear interpolation formula.

\begin{align*}
  \alpha_{i+1} &= \alpha_i + \left\{ \frac{\alpha_i - \alpha_{i-1}}{u(b, \alpha_i) - u(b, \alpha_{i-1})} \right\} (\beta - u(b, \alpha_i)) \\
  \alpha_2 &= \alpha_1 + \left\{ \frac{\alpha_1 - \alpha_0}{u(b, \alpha_1) - u(b, \alpha_0)} \right\} (\beta - u(b, \alpha_1)) \\
  &= 1 + \left\{ \frac{1 - 2.5}{2.47903 - 5.75765} \right\} (0.003077 - 2.47903) \\
  &= -0.13277 = -0.13
\end{align*}

At \( h = 0.75 \) , using the Euler’s method, we get

**From IVP3:** \( u_0 = 0.0038731, \ z_0 = -0.13, \ r_0 = 5, \ h = 0.75 \)

\begin{align*}
  u_1 &= u(5.75) = -0.09363 \\
  z_1 &= -0.11038 \\
  u_2 &= u(6.5) = -0.17641 \\
  z_2 &= -0.09811 \\
  u_3 &= u(7.25) = -0.24999 \\
  z_3 &= -0.08992 \\
  u_4 &= u(8) = -0.31743
\end{align*}

Using the linear interpolation formula, we obtain another initial approximation, \( \alpha_3 \):

\begin{align*}
  \alpha_3 &= \alpha_2 + \left\{ \frac{\alpha_2 - \alpha_1}{u(b, \alpha_2) - u(b, \alpha_1)} \right\} (\beta - u(b, \alpha_2)) \\
  &= -0.13 + \left\{ \frac{-0.13 - 1}{0.00048946 - (-0.31743)} \right\} (0.003077 - (-0.31743)) \\
  &= -0.00048946
\end{align*}

At \( h = 0.75 \) , using the Euler’s method, we get

**From IVP4:** \( u_0 = 0.0038731, \ z_0 = -0.00048946, \ r_0 = 5, \ h = 0.75 \)

\begin{align*}
  u_1 &= u(5.75) = 0.003506 \\
  z_1 &= -0.0002998 \\
  u_2 &= u(6.5) = 0.003281 \\
  z_2 &= -0.0001812
\end{align*}


\[ u_3 = u(7.25) = 0.003145 \]
\[ z_3 = -0.0001021 \]
\[ u_4 = u(8) = 0.0030687 \]

We notice that the last iteration gave a result which corresponds to the actual value of \( u(8) \) to a certain extent. However, using the linear interpolation formula, we will obtain a value which gets closer to the actual value of \( u(8) \). As mentioned in the previous section, the step size and numerical technique used would have a great effect on the accuracy of the values attained [13]. Using the linear interpolation formula, we obtain another initial approximation, \( \alpha_4 \):

\[ \alpha_4 = \alpha_3 + \left\{ \frac{\alpha_3 - \alpha_2}{u(b, \alpha_3) - u(b, \alpha_2)} \right\} (\beta - u(b, \alpha_3)) \]
\[ = -0.00048946 + \left\{ \frac{-0.00048946 - (-0.13)}{0.0030687 - (-0.31743)} \right\} (0.003077 - 0.0030687) \]
\[ = -0.000486106 \approx -0.00048611 \]

At \( h = 0.75 \), using the Euler’s method, we get

**From IVP5**: \( u_0 = 0.0038731, \ z_0 = -0.00048611, \ r_0 = 5, \ h = 0.75 \)

\[ u_1 = u(5.75) = 0.003506 \]
\[ z_1 = -0.000297 \]
\[ u_2 = u(6.5) = 0.003283 \]
\[ z_2 = -0.0001787 \]
\[ u_3 = u(7.25) = 0.003149 \]
\[ z_3 = -0.0000998 \]
\[ u_4 = u(8) = 0.0030741 \]

Table 2 shows the comparison of results attained using the Euler’s method with the last initial guess and the exact methods.

| \( i \) (in) | \( r_i \) (in) | Euler \( (u_i) \) \( (\alpha_i = -0.00048611) \) | Exact (in) | \( |\varepsilon| \) | \( z_i \) |
|---|---|---|---|---|---|
| 0 | 5 | \( 3.8731 \times 10^{-3} \) | \( 3.8731 \times 10^{-3} \) | 0.0000 | \( -4.8611 \times 10^{-4} \) |
| 1 | 5.75 | \( 3.5065 \times 10^{-4} \) | \( 3.5567 \times 10^{-4} \) | \( 5.02 \times 10^{-5} \) | \( -2.97 \times 10^{-4} \) |
| 2 | 6.5 | \( 3.2833 \times 10^{-4} \) | \( 3.3366 \times 10^{-4} \) | \( 5.33 \times 10^{-5} \) | \( -1.787 \times 10^{-4} \) |
| 3 | 7.25 | \( 3.1489 \times 10^{-5} \) | \( 3.1829 \times 10^{-5} \) | \( 3.4 \times 10^{-5} \) | \( -9.98 \times 10^{-5} \) |
| 4 | 8 | \( 3.0741 \times 10^{-4} \) | \( 3.0770 \times 10^{-4} \) | \( 2.9 \times 10^{-5} \) |
4. Conclusion

This work describes the shooting method for the iterative solutions of second order boundary value problems. The iterative formula used was the Euler method. Both linear and nonlinear problems were solved to describe this method.

Acknowledgement

The authors sincerely express their thanks to Covenant University for providing a good working environment and mainly for financial support.

References

[1] Filipov, S. M. Gospodino, I. D. & Farago I. (2017). Shooting –projection method for two-point boundary value problems, Applied Mathematics Letters, 72(1), 10-15. https://doi.org/10.1016/j.aml.2017.04.002
[2] T. A. Anake, D. O. Awoyemi and A. O. Adesanya, “One-step implicit hybrid block method for the direct solution of general second order ordinary differential equations”, IAENG International Journal of Applied Mathematics, 42(4), (2012), 224-228.
[3] O.P. Ogundile, S.O. Edeki, Approximate analytical solutions of linear stochastic differential models based on Karhunen-Loève expansion with finite series terms, Communications in Mathematical Biology and Neuroscience. 2020 (2020).
[4] Gladwell, I. (2008). Boundary value problem. Scholarpedia, 3(1), 2853. https://doi.org/10.4249/scholarpedia.2853
[5] Holsapple, R., Venkataraman, R., & Doman, D. (2003). A modified simple shooting method for solving two-point boundary-value problems. IEEE Aerospace Conference Proceedings, 6(0), 2783–2790. https://doi.org/10.1109/AERO.2003.1235204
[6] Ahsan, M., & Farrukh, S. (2013). A new type of shooting method for nonlinear boundary value problems. Alexandria Engineering Journal, 52(4), 801–805. https://doi.org/10.1016/j.aej.2013.07.001
[7] Bailey, P. B., & Shampine, L. F. (1968). On shooting methods for two-point boundary value problems. Journal of Mathematical Analysis and Applications, 23(2), 235–249. https://doi.org/10.1016/0022-247X(68)90064-4
[8] Adam, B., & Hashim, Mohsin, H. A. (2014). Shooting method in solving value problem. International Journal of Research and Reviews in Applied Sciences, 21(1), 8–26.
[9] Morrison, D., Riley, J., Zancanaro, J., Morrison, D., Riley, J., & Zancanaro, J. (2017). Multiple Shooting Method for Two-Point Boundary Value Problems To cite this version : HAL Id : hal-01634264 Multiple Shooting Method for Two-Point Boundary Value Problems
[10] Attili, B., & Syam, M. (2008). Efficient shooting method for solving two point boundary value problems. Chaos, Solitons & Fractals, 35, 895–903. https://doi.org/10.1016/j.chaos.2006.05.094
[11] Ha, S. N. (2001). A nonlinear shooting method for two-point boundary value problems. Computers and Mathematics with Applications, 42(10–11), 1411–1420. https://doi.org/10.1016/S0898-1221(01)00250-4
[12] Burden, R. L., Faires, J. D., & Burden, A. M. (2015). Numerical Analysis. Cengage Learning. https://books.google.com.ng/books?id=2nZvCgAAQBAJ
[13] Kaw, A. (2009). Shooting Method for Ordinary Differential Equations. Nm.Mathforcollege.Com, 1–7. http://nm.mathforcollege.com/topics/shooting_method.html