Advances in the Theory of Nonlinear Analysis and its Applications **4** (2020) No. 4, 432–442.
https://doi.org/10.31197/atnaa.753561
Available online at www.atnaa.org
Research Article

Advances in the Theory of Nonlinear Analysis and its Applications

ISSN: 2587-2648

Peer-Reviewed Scientific Journal

Forward-Backward Alternating Parallel Shooting Method for Multi-layer Boundary Value Problems

Mohamed Ali Hajji[a]

[a]*Department of Mathematical Sciences, UAE University, Al Ain, UAE.*

Abstract

Multi-layer boundary value problems have received a great deal of attention in the past few years. This is due to the fact that they model many engineering applications. Examples of applications include fluid flow though multi-layer porous media such as ground water and oil reservoirs. In this work, we present a new method for solving multi-layer boundary value problems. The method is based on an efficient adaption of the classical shooting method, where a boundary value problem is solved by means of solving a sequence of initial value problems. We propose, an alternating forward-backward shooting strategy that reduces computational cost. Illustration of the method is presented on application to fluid flow through multi-layer porous media. The examples presented suggested that the method is reliable and accurate.

Keywords: Boundary value problem, Shooting method, Fluid flow, Porous media.
2010 MSC: 65L10; 65N99; 65L99.

1. Introduction

Multi-layer boundary value problems (MLBVPs) have received a great deal of attention in the past few years. Many engineering problems are modelled by multi-layer boundary problems, where the system is described by a different differential equation on different subsets of the physical domain. For example, MLBVPs are used as mathematical models for underground fluid flow through porous channels with different characteristics. It is well known that the classical shooting method [1] is one of the most powerful methods used to solve single boundary value problems.

Email address: `mahajji@uaeu.ac.ae` (Mohamed Ali Hajji)

Received June 16, 2020, Accepted: December 18, 2020, Online: December 20, 2020.

The classical shooting method is based on successively solving a sequence of initial value problems (IVPs) whose initial conditions are iteratively updated. The iterations continuously change (update) the assumed initial conditions until the solution of the initial-value problem satisfies the boundary conditions. Many modifications to the classical shooting method either to improve the rate of convergence or to fit a certain class of boundary value problems have been made. Examples of these modifications include the modified simple shooting method [2], the multiple shooting method [3, 4], and the parallel shooting method [5]. All of these modifications deal with using the shooting method to solve a single boundary value problem over a finite domain. In this work, we consider an adaptation of the shooting method to solve multi-layer boundary value problems, which we refer to as multi-layer parallel shooting method (MLPSM). The advantage of this method is that it is suitable for parallel computing implementation. This kind of problem has been considered in [16], for a two layer problem, and in [17] for a multi-layer setting, both using a finite-difference approach.

As an application of the suggested multilayer parallel shooting method, we will investigate the fluid mechanics of multi-layer flows through porous media. The problem of the two layers flows was investigated by several authors [10]-[13]. In [10], the authors considered the interface region between two different porous media. They employed a special formulation of the shooting method to study the characteristics of the fluid flow. In [11], a finite difference approach was employed for the two channel problem. The multi-channel case was treated in [16, 17] using finite differences. In the present work, we extend the earlier works and formulate a multi-layer parallel shooting method for the solution of the multi-layer case using an alternating forward-backward shooting strategy that reduces the computational cost.

The paper is organised as follows. In section 2, the mathematical formulation of the problem is presented. The proposed method is presented in section 3. In section 4, application of the fluid flow through multi-layer porous media is presented. Concluded remarks are given in section 5.

2. Mathematical Formulation of the Problem

We consider a second order multi-layer boundary value problem consisting of the following set of second order boundary value problems

$$y'' = f_i(x, y, y'), \quad x_{i-1} \le x \le x_i, \quad i = 1, ..., n, \tag{1}$$

with outer boundary conditions

$$y(x_0) = \alpha, \qquad y(x_n) = \beta, \tag{2}$$

and interior boundary conditions at the interface nodes

$$y(x_i^-) = y(x_i^+) \text{ and } y'(x_i^-) = y'(x_i^+). \tag{3}$$

The functions f_i, $i = 1, ..., n$, are assumed to be C^2. The interior nodes x_i, $i = 1, \ldots, n-1$, divide the overall domain $[x_0, x_n]$, not necessarily uniformly, into subdomains $[x_{i-1}, x_i]$, as shown in Fig. 1. The set of interior boundary conditions, equation (3), are imposed such that the overall solution $y(x)$ defined over the whole domain $[x_0, x_n]$, is smooth across the nodes x_i. This requirement is important in many physical problems such as in fluid flow in multi-layer porous channels [11, 12].

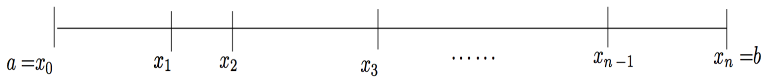

Figure 1: Domain subdivision

It is important to mention that the interior boundary conditions (3) make the problem more restrictive in the sense that the different solutions in neighbouring subdomains must agree smoothly at the nodes x_i. Therefore, it is important that any numerical scheme should produce a numerical solution satisfying (3) as accurate as possible. As mentioned earlier, finite difference methods can be, and in fact has been used [17], to solve equation (1) subject to (2) and (3). However, the accuracy in satisfying (3) dependents on the mesh size, whether it is a uniform mesh size across $[x_0, x_n]$ or different mesh sizes h_i for each subdomain $[x_{i-1}, x_i]$, as was done in [17].

This limitation of finite differences method and the good performance of the classical shooting method prompt us to consider an adaptation of the classical shooting method to solve equations (1) subject to (2) and (3). The next section describes the proposed method. It is well-known that the shooting method is an accurate method for solving boundary value problems via solving a sequence of initial value problems. The classical shooting mechanism is performed in a forward way as shown in Fig. 2. Using, this classical mechanism for problem (1) subject to (2) and (3) requires the introduction of $2n-1$ parameters (1 at x_0 and 2 at each $x_i,\ i = 1, 2, \dots, n-1$). However, as we will see in the section the proposed strategy requires the introduction of n parameters, which reduces the computational complexity by half.

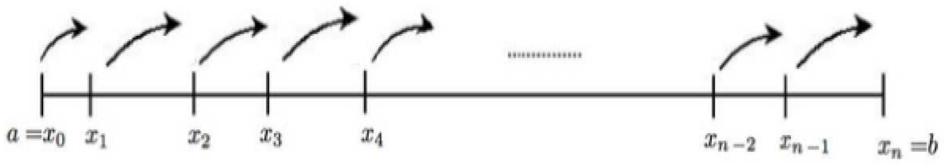

Figure 2: Forward shooting strategy.

3. The Proposed Method

The proposed method to solve problem (1)–(3) consists of a multi-layer shooting method which can be performed in parallel. The shooting strategy we propose is in a forward-backward alternating fashion as illustrated in Fig. 3. The derivations are the same for both n even and odd with a slight modification in the Jacobian matrix. We, therefore, consider in detail the case n even.

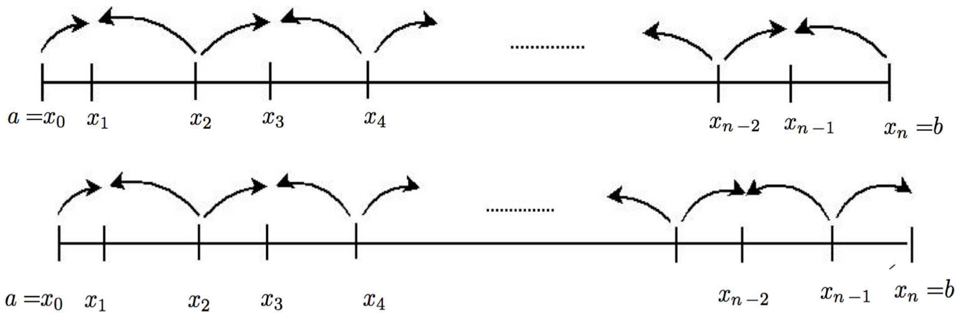

Figure 3: Proposed shooting strategy; top for n even and bottom for n odd.

According to the shooting strategy suggested in Fig. 3, we introduce n parameters $\lambda_i,\ i = 1, \dots, n$, such that

$$y'(x_0) = \lambda_1, \tag{4}$$

$$y(x_{2i}) = \lambda_{2i},\ y'(x_{2i}) = \lambda_{2i+1},\ i = 1, \dots, n/2 - 1, \tag{5}$$

$$y'(x_n) = \lambda_n. \tag{6}$$

Then we solve, in parallel, the following set of initial-value problems:

1. In $[x_0, x_1]$ solve the I.V.P.

$$y''(x) = f_1(x, y, y'), \qquad y(x_0) = \alpha,\ y'(x_0) = \lambda_1. \tag{7}$$

2. In the interior subintervals $[x_{2i-1}, x_{2i}]$, $i = 1, .., n/2 - 1$, solve the I.V.P.

$$y''(x) = f_{2i}(x, y, y'), \qquad y(x_{2i}) = \lambda_{2i},\ y'(x_{2i}) = \lambda_{2i+1}. \tag{8}$$

3. In the interior subintervals $[x_{2i}, x_{2i+1}]$, $i = 1, .., n/2 - 1$, solve the I.V.P.

$$y''(x) = f_{2i+1}(x, y, y'), \qquad y(x_{2i}) = \lambda_{2i},\ y'(x_{2i}) = \lambda_{2i+1}. \tag{9}$$

4. In $[x_{n-1}, x_n]$, solve the I.V.P.

$$y''(x) = f_n(x, y, y'), \qquad y(x_n) = \beta,\ y'(x_n) = \lambda_n. \tag{10}$$

The parameters $\lambda_i,\ i = 1, \ldots, n$, are to be determined such that the interior boundary conditions (3) are satisfied to within a prescribed tolerance. Let $y_1(x; \lambda_1)$, $y_{2i}(x; \lambda_{2i}, \lambda_{2i+1})$, $y_{2i+1}(x; \lambda_{2i}, \lambda_{2i+1})$, $i = 1, ..., n/2-1$, and $y_n(x; \lambda_n)$, be the solutions to (7)–(10), respectively. Imposing conditions (3) at the oddly-indexed nodes $x_{2i-1},\ i = 1, 2, \ldots, n/2 - 1$, (eq. (3)) is already satisfied at x_{2i}), we obtain the set of equations

$$y_1(x_1; \lambda_1) - y_2(x_1; \lambda_2, \lambda_3) = 0, \tag{11}$$
$$y_1'(x_1; \lambda_1) - y_2'(x_1; \lambda_2, \lambda_3) = 0, \tag{12}$$
$$y_{2i+1}(x_{2i+1}; \lambda_{2i}, \lambda_{2i+1}) - y_{2i+2}(x_{2i+1}; \lambda_{2i+2}, \lambda_{2i+3}) = 0,\ 1 \le i \le \frac{n}{2} - 2, \tag{13}$$
$$y_{2i+1}'(x_{2i+1}; \lambda_{2i}, \lambda_{2i+1}) - y_{2i+2}'(x_{2i+1}; \lambda_{2i+2}, \lambda_{2i+3}) = 0,\ 1 \le i \le \frac{n}{2} - 2, \tag{14}$$
$$y_{n-1}(x_{n-1}, \lambda_{n-2}, \lambda_{n-1}) - y_n(x_{n-1}, \lambda_n) = 0, \tag{15}$$
$$y_{n-1}'(x_{n-1}, \lambda_{n-2}, \lambda_{n-1}) - y_n'(x_{n-1}, \lambda_n) = 0. \tag{16}$$

The above set of equations can be regarded as a nonlinear homogeneous system for the unknown parameters λ_i. Introducing the notation $\mathbf{\Lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_n)$, it can be written in vector form as

$$\mathbf{F}(\mathbf{\Lambda}) = \mathbf{0},$$

where $\mathbf{F}(\mathbf{\Lambda}) = [F_1(\mathbf{\Lambda}), ..., F_n(\mathbf{\Lambda})]^T : R^n \longrightarrow R^n$, is a vector valued function of the parameter vector $\mathbf{\Lambda}$ and $F_i(\cdot)$, $i = 1, \ldots, n$, are scalar functions given by equations (11)–(16).

A fast and efficient method for solving the above system is the well-known multidimensional Newton's method. Let

$$\mathbf{\Lambda}^{(k)} = (\lambda_1^{(k)}, \lambda_2^{(k)}, \ldots, \lambda_n^{(k)})^T$$

be the values of the parameters at the k^{th} iteration. The classical multidimensional Newton's method calculates new values $\mathbf{\Lambda}^{(k+1)}$ according to

$$\mathbf{\Lambda}^{(k+1)} = \mathbf{\Lambda}^{(k)} - \mathbf{J}^{-1}(\mathbf{\Lambda}^{(k)})\mathbf{F}(\mathbf{\Lambda}^{(k)}) \tag{17}$$

where $\mathbf{F}(\mathbf{\Lambda}^{(k)}) = [F_1(\mathbf{\Lambda}^{(k)}), \ldots, F_n(\mathbf{\Lambda}^{(k)})]^T$ and $\mathbf{J}^{-1}(\mathbf{\Lambda}^{(k)})$ is the inverse of the $n \times n$ Jacobian matrix $\mathbf{J}$ evaluated at $\mathbf{\Lambda}^{(k)}$. The entries of the Jacobian matrix are given by

$$\mathbf{J}_{ij} = \frac{\partial F_i}{\partial \mathbf{\Lambda}_j} \tag{18}$$

It is worth mentioning that there are modified versions of Newton's method which converge cubically. In the one-dimensional case, cubically convergent modified Newton's schemes were derived in [6, 8, 9] and references therein. A generalization to multivariate case was later given in [7]. The scheme given in [7] is as follows

$$\mathbf{\Gamma}^{(k)} = \mathbf{\Lambda}^{(k)} - \frac{1}{2}\mathbf{J}^{-1}(\mathbf{\Lambda}^{(k)})\mathbf{F}(\mathbf{\Lambda}^{(k)}), \tag{19}$$
$$\mathbf{\Lambda}^{(k+1)} = \mathbf{\Lambda}^{(k)} - \mathbf{J}^{-1}(\mathbf{\Gamma}^{(k)})\mathbf{F}(\mathbf{\Lambda}^{(k)}). \tag{20}$$

The Jacobian matrix **J** turn out to be pentadiagonal and has the form:

$$\boldsymbol{J}_{1,:} = \begin{bmatrix} \frac{\partial y_1(x_1)}{\partial \lambda_1} & \frac{-\partial y_2(x_1)}{\partial \lambda_2} & \frac{-\partial y_2(x_1)}{\partial \lambda_3} & \mathbf{0}_{n-3} \end{bmatrix},$$

$$\boldsymbol{J}_{2,:} = \begin{bmatrix} \frac{\partial y_1'(x_1)}{\partial \lambda_1} & \frac{-\partial y_2'(x_1)}{\partial \lambda_2} & \frac{-\partial y_2'(x_1)}{\partial \lambda_3} & \mathbf{0}_{n-3} \end{bmatrix},$$

for $i = 3, 5, 7, \dots, n-3$,

$$\boldsymbol{J}_{i,:} = \begin{bmatrix} \mathbf{0}_{i-2} & \frac{\partial y_i(x_i)}{\partial \lambda_{i-1}} & \frac{\partial y_i(x_i)}{\partial \lambda_i} & \frac{-\partial y_{i+1}(x_i)}{\partial \lambda_{i+1}} & \frac{-\partial y_{i+1}(x_i)}{\partial \lambda_{i+2}} & \mathbf{0}_{n-i-2} \end{bmatrix},$$

$$\boldsymbol{J}_{i+1,:} = \begin{bmatrix} \mathbf{0}_{i-2} & \frac{\partial y_i'(x_i)}{\partial \lambda_{i-1}} & \frac{\partial y_i'(x_i)}{\partial \lambda_i} & \frac{-\partial y_{i+1}'(x_i)}{\partial \lambda_{i+1}} & \frac{-\partial y_{i+1}'(x_i)}{\partial \lambda_{i+2}} & \mathbf{0}_{n-i-2} \end{bmatrix},$$

and

$$\boldsymbol{J}_{n-1,:} = \begin{bmatrix} \mathbf{0}_{n-3} & \frac{\partial y_{n-1}(x_{n-1})}{\partial \lambda_{n-2}} & \frac{\partial y_{n-1}(x_{n-1})}{\partial \lambda_{n-1}} & \frac{-\partial y_n(x_{n-1})}{\partial \lambda_n} \end{bmatrix},$$

$$\boldsymbol{J}_{n,:} = \begin{bmatrix} \mathbf{0}_{n-3} & \frac{\partial y_{n-1}'(x_{n-1})}{\partial \lambda_{n-2}} & \frac{\partial y_{n-1}'(x_{n-1})}{\partial \lambda_{n-1}} & \frac{-\partial y_n'(x_{n-1})}{\partial \lambda_n} \end{bmatrix},$$

where the notation $\boldsymbol{J}_{k,:}$ stands for the kth row of $\boldsymbol{J}$ and $\mathbf{0}_k$ stands for a row of k zeros.

The calculation of the entries of the Jacobian matrix requires the calculation of

$$\frac{\partial y_i(x_j)}{\partial \lambda_k} \quad \text{and} \quad \frac{\partial y_i'(x_j)}{\partial \lambda_k}$$

with the appropriate x_j and λ_k for a given y_i. Specifically, we require

$$\frac{\partial y_{2i}(x_{2i-1})}{\partial \lambda_k}, \ \frac{\partial y_{2i}'(x_{2i-1})}{\partial \lambda_k}, \ \frac{\partial y_{2i+1}(x_{2i+1})}{\partial \lambda_k}, \ \frac{\partial y_{2i+1}'(x_{2i+1})}{\partial \lambda_k}, \tag{21}$$

for $i = 1, 2, \dots, n/2 - 1$ and $k = 2i, \ 2i+1$, and

$$\frac{\partial y_1(x_1)}{\partial \lambda_1}, \ \frac{\partial y_1'(x_1)}{\partial \lambda_1}, \ \frac{\partial y_n(x_{n-1})}{\partial \lambda_n}, \ \frac{\partial y_n'(x_{n-1})}{\partial \lambda_n}. \tag{22}$$

The quantities in (21) and (22) are obtained from the solutions of an other set of initial value problems as explained below.

From equation (1), differentiating with respect to λ, we get

$$\frac{\partial y''}{\partial \lambda} = \frac{\partial f}{\partial y}\frac{\partial y}{\partial \lambda} + \frac{\partial f}{\partial y'}\frac{\partial y'}{\partial \lambda}.$$

Let $z_{j,k} = \dfrac{\partial y_j}{\partial \lambda_k}$. Assuming we can interchange the order of differentiation, we find that $z_{j,k}$ satisfies

$$z_{j,k}'' = \frac{\partial f_j(x, y_j, y_j')}{\partial y} z_{j,k} + \frac{\partial f_j(x, y_j, y_j')}{\partial y'} z_{j,k}'. \tag{23}$$

Therefore, we have the following I.V.Ps.

1. $z_{1,1} = \frac{\partial y_1}{\partial \lambda_1}$ satisfies (23) for $x_0 \le x \le x_1$ with the I.Cs.

$$z_{1,1}(x_0) = 0, \ z_{1,1}'(x_0) = 1. \tag{24}$$

2. For $i = 1, \dots, n/2-1$, $z_{2i,2i} = \frac{\partial y_{2i}}{\partial \lambda_{2i}}$ satisfies (23) for $x_{2i-1} \le x \le x_{2i}$ with the I.Cs.
$$z_{2i,2i}(x_{2i}) = 1,\ z'_{2i,2i}(x_{2i}) = 0. \tag{25}$$
3. For $i = 1, \dots, n/2-1$, $z_{2i,2i+1} = \frac{\partial y_{2i}}{\partial \lambda_{2i+1}}$ satisfies (23) for $x_{2i-1} \le x \le x_{2i}$ with the I.Cs.
$$z_{2i,2i+1}(x_{2i}) = 0,\ z'_{2i,2i+1}(x_{2i}) = 1. \tag{26}$$
4. For $i = 1, \dots, n/2-1$, $z_{2i+1,2i} = \frac{\partial y_{2i+1}}{\partial \lambda_{2i}}$ satisfies (23) for $x_{2i} \le x \le x_{2i+1}$ with the I.Cs.
$$z_{2i+1,2i}(x_{2i}) = 1,\ z'_{2i+1,2i}(x_{2i}) = 0. \tag{27}$$
5. For $i = 1, \dots, n/2-1$, $z_{2i+1,2i+1} = \frac{\partial y_{2i+1}}{\partial \lambda_{2i+1}}$ satisfies (23) for $x_{2i} \le x \le x_{2i+1}$ with the I.Cs.
$$z_{2i+1,2i}(x_{2i}) = 0,\ z'_{2i+1,2i}(x_{2i}) = 1. \tag{28}$$
6. $z_{n,n} = \frac{\partial y_n}{\partial \lambda_n}$ satisfies (23) for $x_{n-1} \le x \le x_n$ with the I.Cs.
$$z_{n,n}(x_n) = 0,\ z'_{n,n}(x_n) = 1. \tag{29}$$

Once the solutions $z_{j,k}$ of the above I.V.Ps (23) with (24)–(29) are obtained, the different Jacobian entries (21) and (22) are given by
$$\frac{\partial y_j(x_i)}{\partial \lambda_k} = z_{j,k}(x_i), \qquad \frac{\partial y'_j(x_i)}{\partial \lambda_k} = z'_{j,k}(x_i).$$
The proposed algorithm is summarized as follows.

1. At the k^{th} iteration, solve the IVPs (7)–(10) with parameters $\Lambda^{(k)}$.
2. Solve the IVPs (23) with (24)–(29) and construct the Jacobian matrix.
3. Use Newton's formula (17) to update the parameters Λ.
4. Repeat the process until (11)–(16) are satisfied within a desired accuracy.

It is important to mention here that at each step solving the IVPs (7)–(10) can be done in parallel. Similarly, solving the IVPs (23) with (24)–(29) can also be done in parallel. This is an important advantage of the proposed method over finite differences methods.

4. Numerical Examples

In this section we consider two *cooked* examples with exact solutions in order to test the accuracy of the proposed method. For the numerical simulations, we adopted the stopping criteria $\|\mathbf{F}\| \le \varepsilon = 10^{-6}$.

Example 4.1. *As a first example, we consider the following second-order BVP*
$$y'' = \frac{1}{2}(1 - (y')^2 - y\sin(x)), \qquad 0 < x < \pi, \tag{30}$$
with the boundary conditions $y(0) = 2$ and $y(\pi) = 2$.

The exact solution is $y(x) = 2 + \sin(x)$. We simulated this example as a multi-layer problem by decomposing the domain $[0, \pi]$ into five subintervals $[x_{i-1}, x_i]$, $i = 1, \dots, 5$, using the nodes $x_0 = 0$, $x_1 = \pi/5$, $x_2 = \pi/3$, $x_3 = 2\pi/3$, $x_4 = 3\pi/4$, and $x_5 = \pi$. Convergence was achieved after 6 iterations. Fig. 4, displays the numerical solution for the first 4 iterations and Table 1 displays the values of $y(x_i^{\pm})$ and $y'(x_i^{\pm})$ versus iterations. It can be seen from Fig. 4 that the solutions in the subdomains are converging smoothly at the interface nodes x_i. As well, from Table 1, we can see that the numerical solutions and their derivatives at x_i are converging to the those of the exact solution.

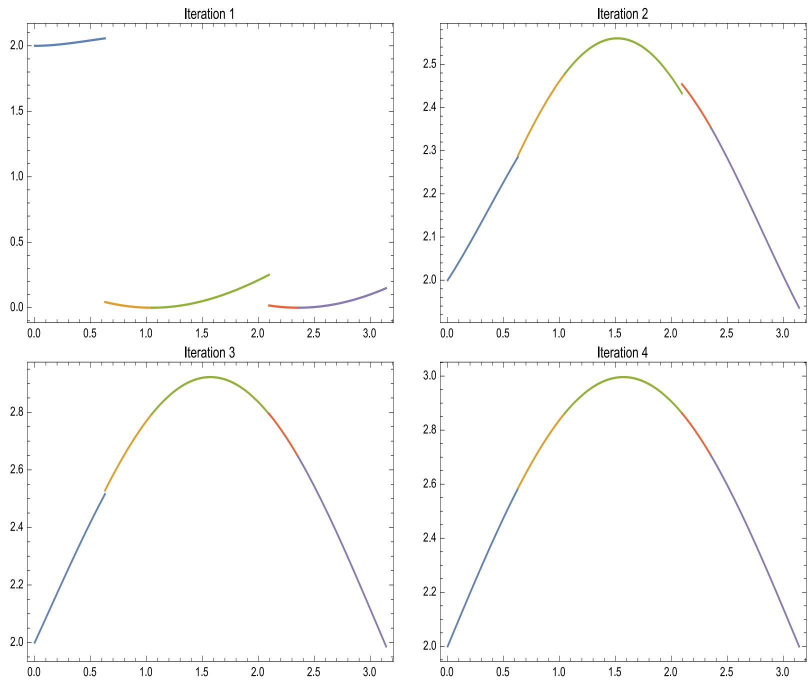

Figure 4: The first 4 iterations for Example 1.

Iter.	1	3	5	6	Exact solution
$y'(0)$	0.	0.848286	0.999972	1.	**1**
$y(\pi/5^-)$	2.05709	2.51584	2.58777	2.58779	**2.58779**
$y(\pi/5^+)$	0.0433234	2.52838	2.58778	2.58779	**2.58779**
$y'(\pi/5^-)$	0.117075	0.725929	0.809002	0.809017	**0.809017**
$y'(\pi/5^+)$	-0.204439	0.769363	0.809012	0.809017	**0.809017**
$y(\pi/3)$	0.	2.79403	2.86602	2.86603	**2.86603**
$y'(\pi/3)$	0.	0.479029	0.499998	0.5	**0.5**
$y(2\pi/3^-)$	0.251475	2.79379	2.86602	2.86603	**2.86603**
$y(2\pi/3^+)$	0.0170467	2.79511	2.86602	2.86603	**2.86603**
$y'(2\pi/3^-)$	0.441647	-0.479507	-0.499996	-0.5	**-0.5**
$y'(2\pi/3^+)$	-0.129542	-0.472927	-0.499996	-0.5	**-0.5**
$y(3\pi/4)$	0.	2.64484	2.7071	2.70711	**2.70711**
$y'(3\pi/4)$	0.	-0.667939	-0.707101	-0.707107	**-0.707107**
$y(\pi)$	0.149262	1.98639	2.	2.	**2**

Table 1: Values of $y(x_i^\pm)$ and $y'(x_i^\pm)$ vs iterations for Example 1.

Example 4.2. *In this second made up example, we consider the following 4-layer problem*

$$y'' = \begin{cases} -(y')^2 - \frac{20}{9}y' - \frac{100}{81}, & 0 \le x < 1/2, \\ (y')^2 - \frac{196}{81} - \frac{1}{x^4} + \frac{4}{x^3} + \frac{10}{9x^2} - \frac{28}{9x}, & 1/2 \le x \le 1, \\ -\left(y' - \frac{5}{9}\right)^2 - \left(y - \frac{5}{9}x + 1\right) + \ln(3x), & 1 \le x \le 2, \\ \frac{38}{9}y + (y')^2 - \frac{19}{36}(7 + 8\ln(6)), & 2 \le x \le 3 \end{cases} \tag{31}$$

with the outer boundary conditions $y(0) = 1/3$, $y(3) = \ln(6) + \frac{1}{9}$, *and the smoothness conditions* $y(x_i^-) = y(x_i^+)$ *and* $y'(x_i^-) = y'(x_i^+)$ *for* $x_i = 1/2,\ 1,$ *and* 2.

The exact solution is

$$y(x) = \begin{cases} \ln(x+1) - \frac{10}{9}x + \frac{1}{3}, & 0 \le x < 1/2, \\ \ln(3x) + \frac{1}{x} + \frac{14}{9}x - 3, & 1/2 \le x \le 1, \\ \ln(3x) + \frac{5}{9}x - 1, & 1 \le x \le 2, \\ \ln(6) - \frac{56}{9} + \frac{95}{18}x - \frac{19}{18}x^2, & 2 \le x \le 3. \end{cases} \tag{32}$$

We applied the proposed algorithm to this example and convergence to within the prescribed accuracy was achieved after 6 iterations. Fig. 5, displays the results for the 6 iterations. Table 2 displays the values of $y(x_i^\pm)$ and $y'(x_i^\pm)$ versus iterations which can be seen to converge to those of the exact solution.

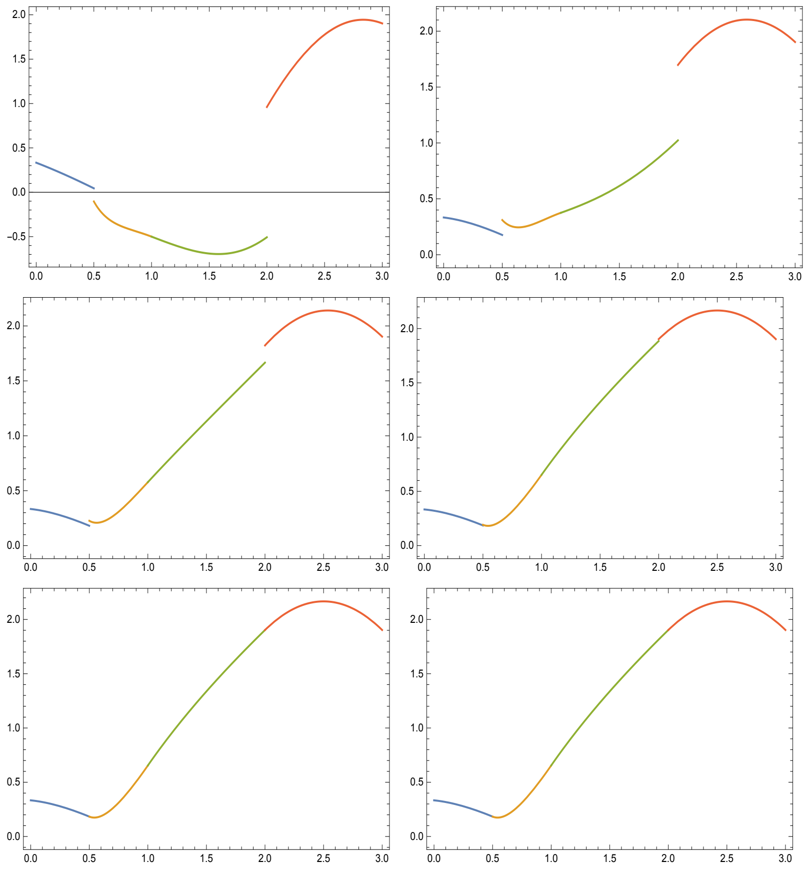

Figure 5: The first 6 iterations for Example 2.

Iter.	1	3	4	6	Exact solution
$y'(0)$	-0.5	-0.118699	-0.111101	-0.111111	**-0.111111**
$y(1/2^-)$	0.0444064	0.180711	0.183246	0.183243	**0.183243**
$y(1/2^+)$	-0.10358	0.22613	0.183307	0.183243	**0.183243**
$y'(1/2^-)$	-0.643026	-0.447825	-0.44444	-0.444444	**-0.444444**
$y'(1/2^+)$	-2.35242	-0.619855	-0.444591	-0.444444	**-0.444444**
$y(1)$	-0.5	0.577562	0.654126	0.654168	**0.654168**
$y'(1)$	-0.5	1.15635	1.55518	1.55556	**1.55556**
$y(2^-)$	-0.506763	1.66597	1.90268	1.90287	**1.90287**
$y(2^+)$	0.960851	1.82262	1.90282	1.90287	**1.90287**
$y'(2^-)$	0.933563	1.06121	1.05555	1.05556	**1.05556**
$y'(2^+)$	2.22717	1.181	1.05563	1.05556	**1.05556**
$y'(3)$	-0.5	-1.02264	-1.05554	-1.05556	**-1.05556**

Table 2: Values of $y(x_i^\pm)$ and $y'(x_i^\pm)$ vs iterations for Example 2.

The previous two example demonstrate that the proposed method is a reliable, fast and accurate for the numerical simulation of multi-layer problems in (1)-(3). As an application of the proposed method to a physical problem, we shall, in the next section, use it to numerically resolve the velocity profile of fluid flow through multi-layer porous media.

5. Application to Fluid Flow Through Multi-layer Porous Media

In this section, we present an application of the algorithm presented in Section 3 to the problem of determining the velocity profile of fluid flow through multi-layer porous media. The media is composed of a number of porous layers. The top and the bottom layers are bounded above and below, respectively, by solid walls, see Fig. 6 which depicts a multi-layer porous media configuration. In each porous layer, the flow

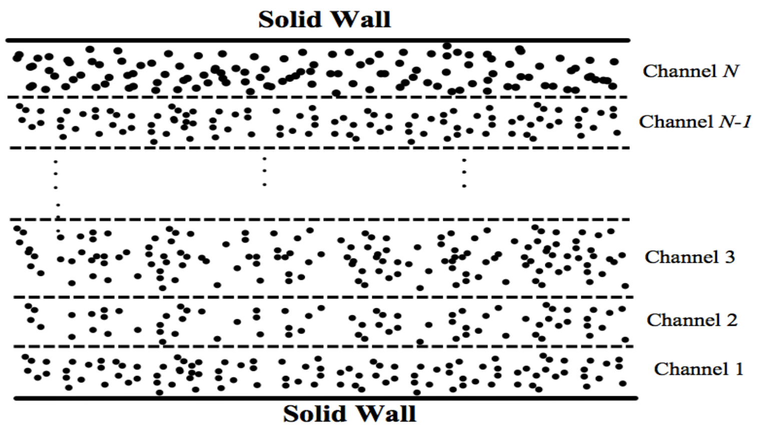

Figure 6: Configuration of a multi-layer porous media.

is assumed to be governed by the Darcy-Forchheimer-Brinkman (DFB) model [12]

$$\frac{d^2u}{dy^2} = Re\,C + \frac{u}{k} + \frac{Re\,C_d}{\sqrt{k}}u^2, \tag{33}$$

where $u(y)$, $a \leq y \leq b$, is the velocity of the fluid, and the various physical parameters in Eq. (33) are defined as follows.

1. Re $= \rho U_\infty L/\mu$ is the Reynolds number, ρ is the fluid density, U_∞ is the free stream characteristic velocity, μ is the fluid viscosity, and L is the channel characteristic length.
2. k is the permeability of the porous channel.
3. C_d is the form drag coefficient.
4. $C < 0$ is a dimensionless pressure gradient.

When $C_d = 0$, we have what is known as the Darcy-Lapwook-Brinkman (DLB) [12] model

$$\frac{d^2u}{dy^2} = Re\,C + \frac{u}{k}. \tag{34}$$

It is worth mentioning that there have been a lot of work on fluid flow through porous media [10]-[17]. The book by Chen et al. [15] presents an excellent account on computational methods for multiphase flows in porous media. The method used in [17] is based on finite difference approximation of the derivatives.

In this work, we aim to show that the proposed method is suitable for solving for the velocity profile of fluid flow through any number of porous-layer configuration. We will use the two examples considered in [17] of the three- and five-layer configurations. In our simulations, we fix the following parameter values. The Reynolds number $Re = 10$, $C = -10$ and $C_d = 0.55$, and without loss of generality, we assume that overall domain is $-1 \leq y \leq 1$. Since the top (at $y = 1$) and bottom (at $y = -1$) layers are bounded by solid impermeable walls, we assume a non-slip condition at $y = \pm 1$, that is $u(\pm 1) = 0$. At the interface nodes x_i, we assume that the flow and shear stress are continuous, that is, $u(x_i^-) = u(x_i^+)$ and $u'(x_i^-) = u'(x_i^+)$. Therefore, here, what distinguish between the different porous layers is the permeability parameter k.

We consider the cases of a three-porous media configuration and five-layer porous media configuration, as considered in [17], where the flow in all three layers is governed by the DFB model (Eq. (33)). For the three-layer configurations, we use the following permeability values: $k_b = 0.001, k_m = 0.01, k_t = 100$, where k_b, k_m, and k_t are the permeabilities of the bottom, middle, and top layers, respectively. The interface nodes are $x_1 = -1/2$ and $x_2 = 1/2$. For the five-layer configuration, we use the following permeability values: $k_1 = 0.1, k_2 = 1, k_3 = 0.001, k_4 = 10, k_5 = 0.001$, where k_1 is for the bottom layer ($-1 \leq y \leq x_1$) and k_5 for the top layer ($x_4 \leq y \leq 1$). The interface nodes for this case are $x_1 = -0.6, x_2 = 0, x_3 = 0.4, x_4 = 0.6$. The fluid velocity profiles for both cases are displayed in Fig. 7 which show an agreement with the results found in [17].

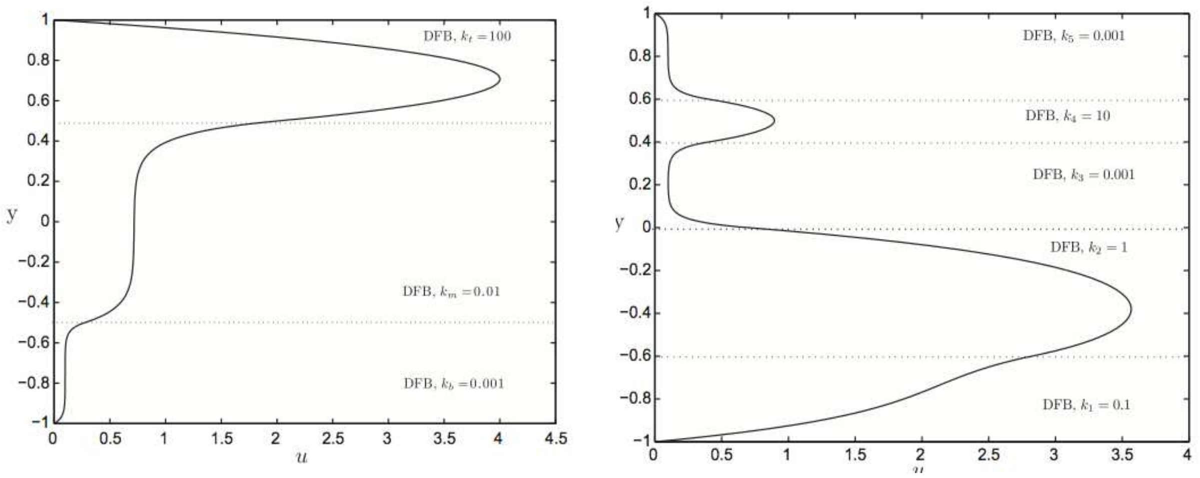

Figure 7: Velocity profiles of the 3 and 5 layer configurations.

6. Conclusion

In this paper we have presented a parallel shooting method to solve multi-layer boundary values problems. The method relies on the adaption of the classical shooting method for single boundary value problems to multi-layer boundary value problems. We have adopted an alternating forward-backward shooting strategy which decreases the size of the problem, compared to a classical forward shooting strategy. A good advantage of this method is that it can be implemented in parallel, unlike the finite-difference method, hence reducing the computation time, especially when the number of layers exceeds 3. A real application of the method is in the resolution of the velocity profile of fluid flow through multi-layer porous media. The numerical simulations suggest that the proposed method is reliable and accurate.

References

[1] R.L. Burden, and J.D. Faires, Numerical Analysis, 7^{th} ed., Brooks/Cole (2001).

[2] R. Holsapple, R. Venkataraman, and D. Doman, A modified simple shooting method for solving two-point boundary value problems, Proc. of IEEE Aerospace Conference 6 (2003) 2783-2790.

[3] J.R. Munchen, R.B. Munchen, and J.P. Innsbrusk, A multiple shooting method for the numerical treatment of stellar structure and evolution, Astron. Nachr . 315 (1994) 205-234.

[4] D.D. Morrison, J.D. Relay, and J.F. Zancanaro, Multiple shooting method for two-point boundary value problems, Communication of the ACM 5(12) (1962) 613-514.

[5] G. Gheri, and P. Marzulli, Parallel shooting with error estimate for increasing the accuracy, J. Computational and Applied Mathematics 115 (2000) 213-227 .

[6] H.H.H. Homeier, A modified Newton method for root finding with cubic convergence, J. Comput. Appl. Math. 157 (2003) 227-230.

[7] H.H.H. Homeier, A modified Newton method with cubic convergence: the multivariate case, J. Comput. Appl. Math. 169 (2004) 161-169.

[8] M. Frontini, E. Sormani, Modified Newton's method with third-order convergence and multiple roots, J. Comput. Appl. Math. 156 (2003) 345 - 354.

[9] M. Frontini, E. Sormani, Some variant of Newton's method with third-order convergence, Appl. Math. Comput. 140 (2003) 419-426.

[10] F. Allan and M.A. Hajji, (2008), Mathematical Manipulation of the Interface Region of Multilayer Flow, International Journal of Porous Media 12 (2009) 461-475.

[11] F. Allan and M. Hamdan, Fluid Mechanics of The Interface Region Between Two porous Layers, Applied Mathematics Computation 128(1) (2002) 37-43.

[12] R.A. Ford and M. Hamdan, Coupled Parallel Flow through Composite Porous Layers, Applied Mathematics Computation 97 (1998) 261-271.

[13] F. Allan, M.A. Hajji, and M. Anwar, The characteristics of fluid flow through multilayer porous media, J. Applied Mechanics 76 (2009).

[14] K. Vafai, S.J. Kim, Fluid mechanics of the interface region between a porous medium and a fluid layer: an exact solution, Int. J. Heat Fluid Flow 11 (1990) 254-256.

[15] Z. Chen, G. Huan, Y. Ma, Computational methods for multiphase flows in porous media, Soc. Indust. Appl. Math. (2006).

[16] M.A. Hajji, Multi-point special boundary value problems and applications to fluid flow through porous media, Proceedings of the 2009 IAENG International Conference on Scientific Computing, Hong Kong (2009) 18-20.

[17] M.A. Hajji, A numerical scheme for multi-point special boundary-value problems and application to fluid flow through porous layers, Applied Mathematics and Computation 217 (2011) 5632-5642.

[18] A. Ramírez, A. Romero, F. Chavez, F. Carrillo, and S. Lopez, Mathematical simulation of oil reservoir properties, Chaos, Solitons & Fractals, 38 (3) (2008) 778-788.