Miklós Farkas
Seminar on Applied Analysis, Budapest, 18 Oct 2018

Numerical Solution of Nonlinear Two-Point Boundary Value Problems

The linearization methods as a basis to derive the FDMs and the shooting methods. Successive application of the linear shooting method.

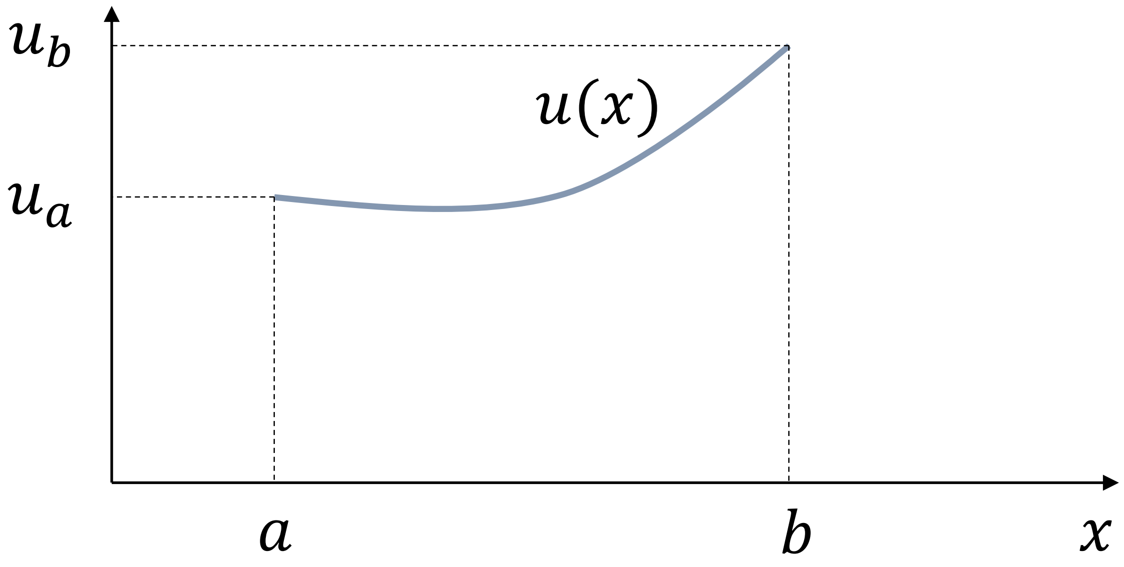
Stefan Filipov [1], Ivan Gospodinov [1], István Faragó [2]

[1] Department of Computer Science, Faculty of Chemical System Engineering,
University of Chemical Technology and Metallurgy, Sofia, Bulgaria

[2] Department of Applied Analysis and Computational Mathematics, Faculty of Science, MTA-ELTE Research Group,
Eötvös Loránd University, Budapest, Hungary

Nonlinear two-point boundary value problem

$$u''(x) = f\big(x, u(x), u'(x)\big), x \in (a, b),$$
$$u(a) = u_a, u(b) = u_b \text{ (Dirichlet)}$$



f – nonlinear function of u and/or u'

We have also considered:
Neumann, general linear, nonlocal BCs and integral condition.

Shooting methods and relaxation methods (FDM)

Shooting Method

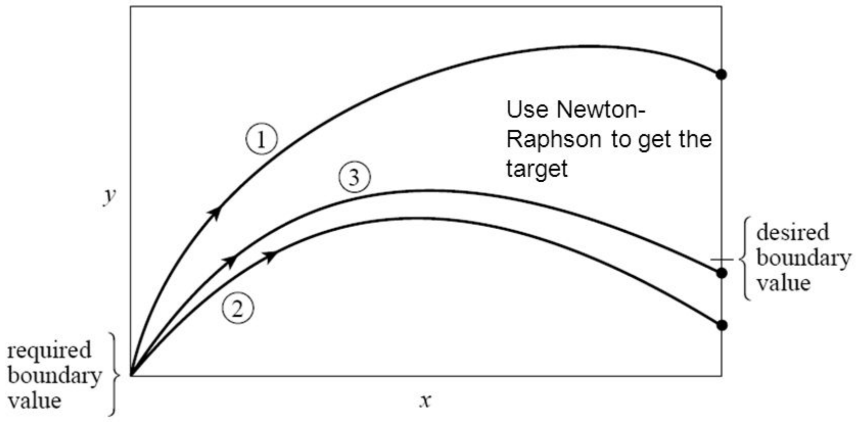


Figure 17.0.1. Shooting method (schematic). Trial integrations that satisfy the boundary condition at one endpoint are "launched." The discrepancies from the desired boundary condition at the other endpoint are used to adjust the starting conditions, until boundary conditions at both endpoints are ultimately satisfied.

Relaxation Methods (FDM)

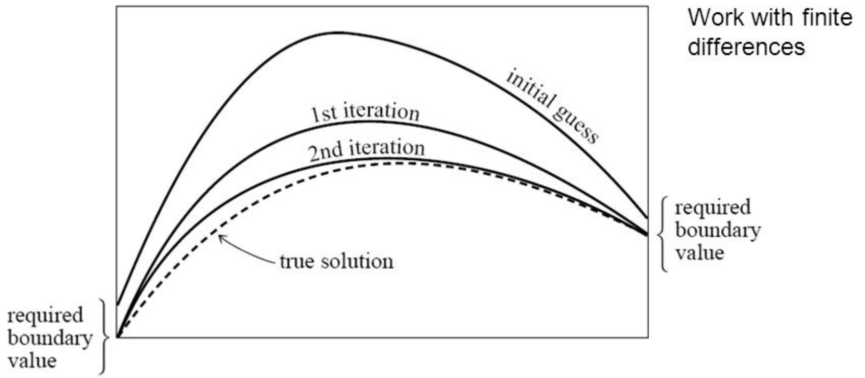


Figure 17.0.2. Relaxation method (schematic). An initial solution is guessed that approximately satisfies the differential equation and boundary conditions. An iterative process adjusts the function to bring it into close agreement with the true solution.

* Numerical Recipes in C

The linearization methods as a basis to derive the FDMs and the shooting methods

Linearization method

	Shooting-projection procedure		FDM discretization	
Shooting by Newton	$\longleftarrow$	Newton (QLM)	$\longrightarrow$	NFDM
[1]	$\longleftarrow$	Picard	$\longrightarrow$	PFDM
Shooting by const. sl.	$\longleftarrow$	constant-slope	$\longrightarrow$	CFDM

[1] S. M. Filipov, I. D. Gospodinov, I. Faragó (2017). Shooting-projection method for two-point boundary value problems. Appl. Math. Lett. 72 (2017) 10–15

$$F(x) = 0 \iff x = \underbrace{x - \frac{F(x)}{m}}_{f(x)}, m \neq 0$$

$x = f(x)$. Expand $f(x)$ around x_k:

$$x = f(x_k) + f'(x_k)(x - x_k) + \cdots \qquad (1)$$

We are going to drop terms in the rhs of (1) and replace x by its approximation x_{k+1}. We consider the following three types of *linearization* around x_k:

- (i) Newton: $x_{k+1} = f(x_k) + f'(x_k)(x_{k+1} - x_k) \Longrightarrow x_{k+1} = x_k - \dfrac{F(x_k)}{F'(x_k)}$
- (ii) Picard: $x_{k+1} = f(x_k) \Longrightarrow x_{k+1} = x_k - \dfrac{F(x_k)}{m}$
- (iii) constant-slope: $x_{k+1} = f(x_k) + f'(x_0)(x_{k+1} - x_k) \Longrightarrow x_{k+1} = x_k - \dfrac{F(x_k)}{F'(x_0)}$

$$u''(x) = f\big(x, u(x), u'(x)\big), x \in (a, b),$$

Expand f around $\big(x, u_{(k)}(x), u'_{(k)}(x)\big)$, drop terms higher than linear, and replace $u(x)$ by its approximation $u_{(k+1)}(x)$:

$$u''_{(k+1)}(x) = f_{(k)}(x) + q_{(k)}(x)\big(u_{(k+1)}(x) - u_{(k)}(x)\big) + p_{(k)}(x)\big(u'_{(k+1)}(x) - u'_{(k)}(x)\big)$$
$$u_{(k+1)}(a) = u_a, u_{(k+1)}(b) = u_b, \quad k = 0,1, \ldots$$

where $f_{(k)}(x) = f\big(x, u_{(k)}(x), u'_{(k)}(x)\big)$ and

	Newton (QLM)	Picard	constant-slope
$q_{(k)}(x) =$	$\partial_2 f\big(x, u_{(k)}(x), u'_{(k)}(x)\big)$	0	$\partial_2 f\big(x, u_{(0)}(x), u'_{(0)}(x)\big)$
$p_{(k)}(x) =$	$\partial_3 f\big(x, u_{(k)}(x), u'_{(k)}(x)\big)$	0	$\partial_3 f\big(x, u_{(0)}(x), u'_{(0)}(x)\big)$

Discretize the ODE using the central finite difference approximation for $u''(x)$:

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = f(x_i, u_i, \mathcal{D}u_i), i = 2,3, \dots, N-1$$

$$u_1 = u_a, u_N = u_b$$

where $\mathcal{D} \in \{\mathcal{D}_+, \mathcal{D}_-, \mathcal{D}_0\}$, and

$$\mathcal{D}_+ u_i = \frac{u_{i+1} - u_i}{h}, \qquad \mathcal{D}_- u_i = \frac{u_i - u_{i-1}}{h}, \qquad \mathcal{D}_0 u_i = \frac{u_{i+1} - u_{i-1}}{2h}.$$

This is a nonlinear system of N equations for the N unknowns $u_i, i = 1,2, \dots, N$.

Solving the nonlinear system

The nonlinear system can be written in the form: $\mathbf{G}(\mathbf{u}_h) = 0$,

where $\mathbf{u}_h = [u_1, u_2, \dots, u_N]^T$, and $\mathbf{G}$ is $N \times 1$ vector with components:

$$G_1 = u_1 - u_a, G_N = u_N - u_b, G_i = u_{i+1} - 2u_i + u_{i+1} - h^2 f(x_i, u_i, \mathcal{D}u_i), i = 2,3,\dots,N-1$$

The nonlinear system can solved by the following iteration:

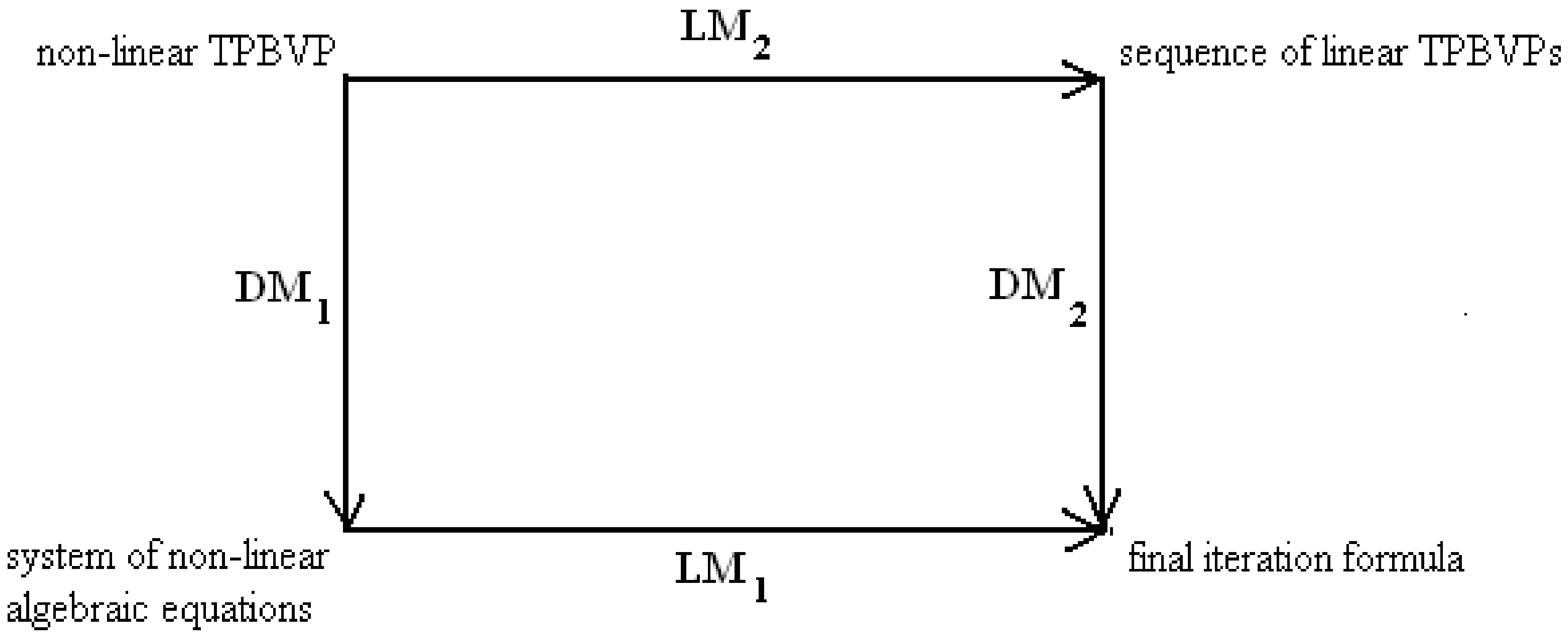
$$\mathbf{u}_h^{(k+1)} = \mathbf{u}_h^{(k)} - \left(\mathbf{L}_h^{(k)}\right)^{-1} \mathbf{G}\left(\mathbf{u}_h^{(k)}\right), k = 0,1,\dots \qquad L_{1,1}^{(k)} = 1, L_{N,N}^{(k)} = 1$$

$$L_{i,i-1}^{(k)} = 1 - h^2 p_i^{(k)} \frac{\partial \mathcal{D}u_i^{(k)}}{\partial u_{i-1}^{(k)}}, L_{i,i}^{(k)} = -2 - h^2 q_i^{(k)} - h^2 p_i^{(k)} \frac{\partial \mathcal{D}u_i^{(k)}}{\partial u_i^{(k)}}, L_{i,i+1}^{(k)} = 1 - h^2 p_i^{(k)} \frac{\partial \mathcal{D}u_i^{(k)}}{\partial u_{i+1}^{(k)}}$$

	Newton	Picard	constant-slope
$q_i^{(k)} =$	$\partial_2 f\left(x_i, u_i^{(k)}, \mathcal{D}u_i^{(k)}\right)$	0	$\partial_2 f\left(x_i, u_i^{(0)}, \mathcal{D}u_i^{(0)}\right)$
$p_i^{(k)} =$	$\partial_3 f\left(x_i, u_i^{(k)}, \mathcal{D}u_i^{(k)}\right)$	0	$\partial_3 f\left(x_i, u_i^{(0)}, \mathcal{D}u_i^{(0)}\right)$

Equivalence between NFDM (PFDM,CFDM) and QLM (PLM,CLM) with FDM

Theorem: Let $\mathrm{LM}_1, \mathrm{LM}_2 \in \{$ Newton, Picard, constant-slope$\}$ be two linearization methods and let $\mathrm{DM}_1, \mathrm{DM}_2 \in \{\mathrm{FDM}\ \mathcal{D}_+, \mathrm{FDM}\ \mathcal{D}_-, \mathrm{FDM}\ \mathcal{D}_0\}$ be two FDM discretization schemes. If $\mathrm{LM}_1 = \mathrm{LM}_2$ and $\mathrm{DM}_1 = \mathrm{DM}_2$, then $\mathrm{LM}_1 \circ \mathrm{DM}_1 = \mathrm{DM}_2 \circ \mathrm{LM}_2$.



Proof of the equivalence theorem – part 1

The QLM equation is discretized using the central difference approximation for $u''_{(k+1)}(x)$:

$$\frac{u_{i-1}^{(k+1)} - 2u_i^{(k+1)} + u_{i+1}^{(k+1)}}{h^2} - p_i^{(k)} \mathcal{D}u_i^{(k+1)} - q_i^{(k)} u_i^{(k+1)} = r_i^{(k)}, i = 2,3,\dots,N-1 \qquad (2)$$

where $r_i^{(k)} = f_i^{(k)} - q_i^{(k)} u_i^{(k)} - p_i^{(k)} \mathcal{D}u_i^{(k)}$. Euler's theorem on homogenous functions:

$$\mathcal{D}u_i^{(k+1)} = u_{i-1}^{(k+1)} \frac{\partial \mathcal{D}u_i^{(k+1)}}{\partial u_{i-1}^{(k+1)}} + u_i^{(k+1)} \frac{\partial \mathcal{D}u_i^{(k+1)}}{\partial u_i^{(k+1)}} + u_{i+1}^{(k+1)} \frac{\partial \mathcal{D}u_i^{(k+1)}}{\partial u_{i+1}^{(k+1)}} \qquad (3)$$

Using property (3) and $\partial \mathcal{D}u_i^{(k+1)} / \partial u_j^{(k+1)} = \partial \mathcal{D}u_i^{(k)} / \partial u_j^{(k)}$, we write equation (2) as:

$$\left(1 - h^2 p_i^{(k)} \frac{\partial \mathcal{D}u_i^{(k)}}{\partial u_{i-1}^{(k)}}\right) u_{i-1}^{(k+1)} + \left(-2 - h^2 q_i^{(k)} - h^2 p_i^{(k)} \frac{\partial \mathcal{D}u_i^{(k)}}{\partial u_i^{(k)}}\right) u_i^{(k+1)} + \left(1 - h^2 p_i^{(k)} \frac{\partial \mathcal{D}u_i^{(k)}}{\partial u_{i+1}^{(k)}}\right) u_{i+1}^{(k+1)} = h^2 r_i^{(k)}$$

or, in a matrix form: $\mathbf{L}_h^{(k)} \mathbf{u}_h^{(k+1)} = \mathbf{R}_h^{(k)}\left(\mathbf{u}_h^{(k)}\right)$, with $\mathbf{R}_h^{(k)}\left(\mathbf{u}_h^{(k)}\right) = \left[u_a, h^2 r_1^{(k)}, h^2 r_2^{(k)}, \dots, u_b\right]^T$

Proof of the equivalence theorem – part 2

Now, we rearrange the rhs of $\mathbf{L}_h^{(k)}\mathbf{u}_h^{(k+1)} = \mathbf{R}_h^{(k)}\left(\mathbf{u}_h^{(k)}\right).$ (4)

Using property (3) for $\mathcal{D}u_i^{(k)}$, we can write: $\mathbf{R}_h^{(k)}\left(\mathbf{u}_h^{(k)}\right) = \mathbf{L}_h^{(k)}\mathbf{u}_h^{(k)} - \mathbf{G}\left(\mathbf{u}_h^{(k)}\right).$ (5)

Substituting (5) into (4) we get: $\mathbf{L}_h^{(k)}\mathbf{u}_h^{(k+1)} = \mathbf{L}_h^{(k)}\mathbf{u}_h^{(k)} - \mathbf{G}\left(\mathbf{u}_h^{(k)}\right).$ (6)

Finally, multiplying both sides of (6) by the inverse of $\mathbf{L}_h^{(k)}$ we obtain:

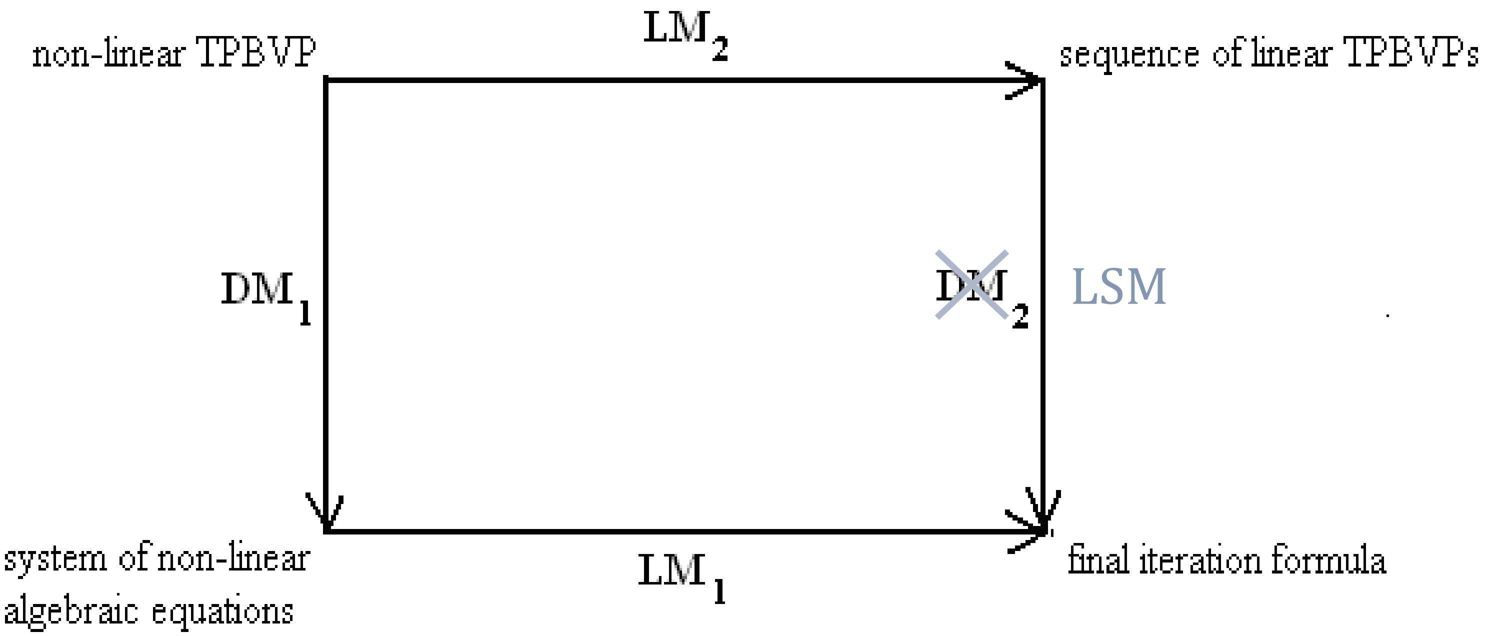
$$\boxed{\mathbf{u}_h^{(k+1)} = \mathbf{u}_h^{(k)} - \left(\mathbf{L}_h^{(k)}\right)^{-1}\mathbf{G}\left(\mathbf{u}_h^{(k)}\right)}$$

QED

Replacing NFDM (PFDM,CFDM) by successive application of the LSM

Why is this result useful?

We can substitute DM_2 by some *alternative method* (AM). Then, the method $\mathrm{AM} \circ \mathrm{LM}_2$ will produce, at each iteration step k, the same result (up to numerical accuracy) as the method $\mathrm{LM}_1 \circ \mathrm{DM}_1$. We propose the linear shooting method (LSM) as AM. Hence, we can replace NFDM (PFDM,CFDM) by successive application of the LSM.



This substitution reduces the number of operations from $O(N^3)$ to only $O(N)$.

QLM (PLM, CLM) with linear shooting method

We apply the linear shooting method (LSM) to solve the sequence of linear sub-problems arising from Newton (QLM), Picard, or constant-slope linearization. We refer to this approach as NLSM, PLSM, or CLSM, respectively.

Let $\bar{u}(x)$ and $\bar{\bar{u}}(x)$ be solutions to the following IVPs (Cauchy problems), respectively:

$$\bar{u}''(x) = p_{(k)}(x)\bar{u}'(x) + q_{(k)}(x)\bar{u}(x) + r_{(k)}(x), \text{ where } r_{(k)} = f_{(k)} - q_{(k)}u_{(k)} - p_{(k)}v_{(k)}$$

$$\bar{u}(a) = u_a, \bar{u}'(a) = 0, \tag{7}$$

$$\bar{\bar{u}}''(x) = p_{(k)}(x)\bar{\bar{u}}'(x) + q_{(k)}(x)\bar{\bar{u}}(x),$$

$$\bar{\bar{u}}(a) = 0, \bar{\bar{u}}'(a) = 1. \tag{8}$$

The LSM gives the solution as:

$$\boxed{u_{(k+1)}(x) = \bar{u}(x) + \left(\frac{u_b - \bar{u}(b)}{\bar{\bar{u}}(b)}\right)\bar{\bar{u}}(x)} \tag{9}$$

To compare numerically the NFDM with the NLSM we choose an IVP method with the same discretization as the FDM. Let $\mathcal{D} = \mathcal{D}_{-}$ and let $v_{i-1} = (u_i - u_{i-1})/h$. Then, the FDM discrete equation (page 7) can be written as:

$$\begin{aligned} u_i &= u_{i-1} + h v_{i-1}, \\ v_i &= v_{i-1} + h f(x_i, u_i, v_{i-1}), \ \ i = 2,3,\dots,N. \end{aligned} \tag{10}$$

The method (10) is explicit Euler but with x_i, u_i instead of x_{i-1}, u_{i-1} in f. We call it EE_.

$\bar{u}_1 = u_a, \bar{v}_1 = 0$ (*initial conditions*)

$$\begin{aligned} \bar{u}_i &= \bar{u}_{i-1} + h\bar{v}_{i-1}, \quad i = 2,3,\dots,N \\ \bar{v}_i &= \bar{v}_{i-1} + h\left(p_i^{(k)}\bar{v}_{i-1} + q_i^{(k)}\bar{u}_i + r_i^{(k)}\right), \end{aligned}$$

$\bar{\bar{u}}_1 = 0, \bar{\bar{v}}_1 = 1$ (*initial conditions*)

$$\begin{aligned} \bar{\bar{u}}_i &= \bar{\bar{u}}_{i-1} + h\bar{\bar{v}}_{i-1}, \quad i = 2,3,\dots,N \\ \bar{\bar{v}}_i &= \bar{\bar{v}}_{i-1} + h\left(p_i^{(k)}\bar{\bar{v}}_{i-1} + q_i^{(k)}\bar{\bar{u}}_i\right), \end{aligned}$$

$$\boxed{\mathbf{u}_h^{(k+1)} = \bar{\mathbf{u}}_h + \left(\frac{u_b - \bar{u}_N}{\bar{\bar{u}}_N}\right)\bar{\bar{\mathbf{u}}}_h,} \tag{11}$$

At each step k, we can use (11), instead of FDM (page 8), avoiding $\left(\mathbf{L}_h^{(k)}\right)^{-1}\mathbf{G}\left(\mathbf{u}_h^{(k)}\right)$.

Numerical comparison between NFDM and NLSM

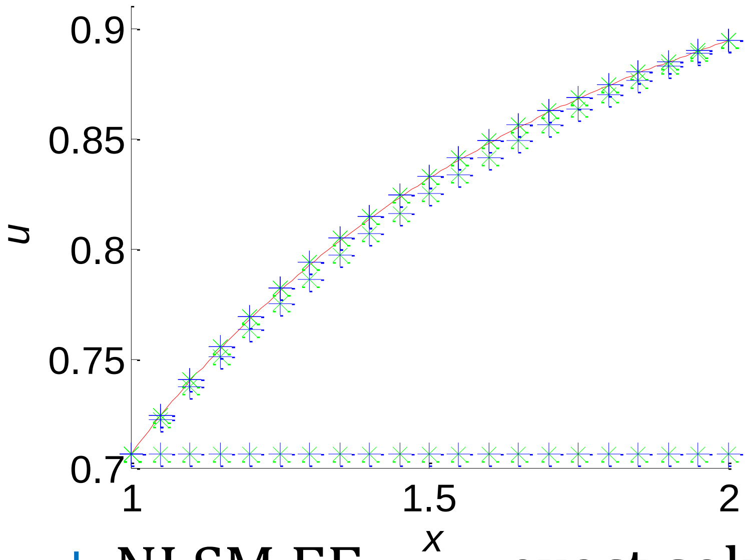
For the solution of the IVPs, we choose EE_. For the FDM, we choose $\mathcal{D} = \mathcal{D}\_$. Consider:

$$u'' = -\frac{3u^2u'}{x}, x \in (1,2), u(1) = \frac{1}{\sqrt{2}}, u(2) = \frac{2}{\sqrt{5}} \quad (12)$$

Exact solution: $u(x) = x/\sqrt{1+x^2}$.

Table 1. $\epsilon^{(k)} = \left\| \mathbf{u}_h^{(k+1)} - \mathbf{u}_h^{(k)} \right\|_{L2}$

k	$\epsilon^{(k)}$, NFDM _	$\epsilon^{(k)}$, NLSM EE_
0	1.257774292959600e-01	1.257774292959598e-01
1	5.908388441300371e-03	5.908388441300540e-03
2	7.827639094112682e-06	7.827639094064679e-06
3	1.207704920791167e-11	1.207704131132550e-11



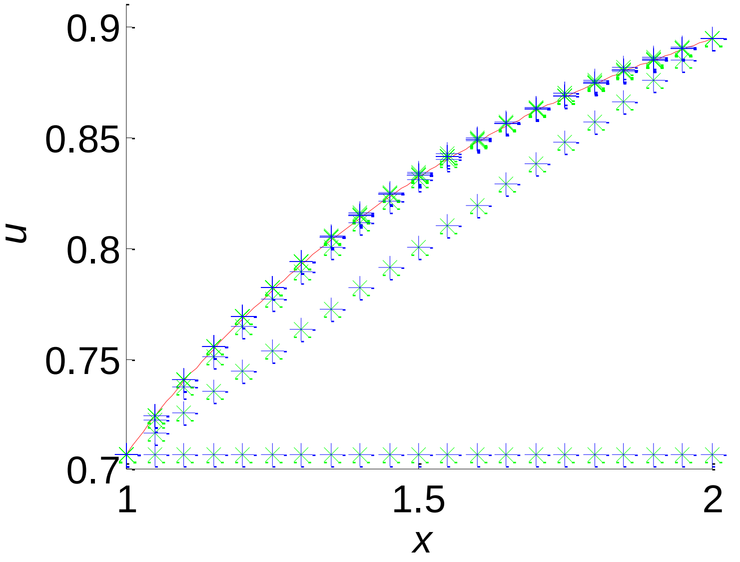
× NFDM_ , + NLSM EE_ , — exact solution

Numerical comparison between PFDM and PLSM

We solve the TPBVP (12) using PFDM and PLSM. Again, we choose $\mathcal{D} = \mathcal{D}$_ and EE_.

Table 2. $\epsilon^{(k)} = \left\| \mathbf{u}_h^{(k+1)} - \mathbf{u}_h^{(k)} \right\|_{L2}$

k	$\epsilon^{(k)}$, PFDM _	$\epsilon^{(k)}$, PLSM EE_
0	1.121969507867195e-01	1.121969507867196e-01
1	2.213503602145248e-02	2.213503602145243e-02
2	3.057696505976492e-03	3.057696505976490e-03
3	5.647562667135585e-04	5.647562667136228e-04
4	1.806427862599132e-04	1.806427862598797e-04
5	2.463853234546461e-05	2.463853234551921e-05
6	8.371405221603600e-06	8.371405221586881e-06
7	1.771619590213677e-06	1.771619590120436e-06
8	3.078247434095909e-07	3.078247434000902e-07
9	1.062797393253808e-07	1.062797392684467e-07
10	1.552916954477012e-08	1.552916953923758e-08
11	4.739721265430622e-09	4.739721250544968e-09
12	1.114973121540600e-09	1.114973108446678e-09
13	1.736426690506943e-10	1.736426723812498e-10
14	6.326895982029898e-11	6.326885861115776e-11



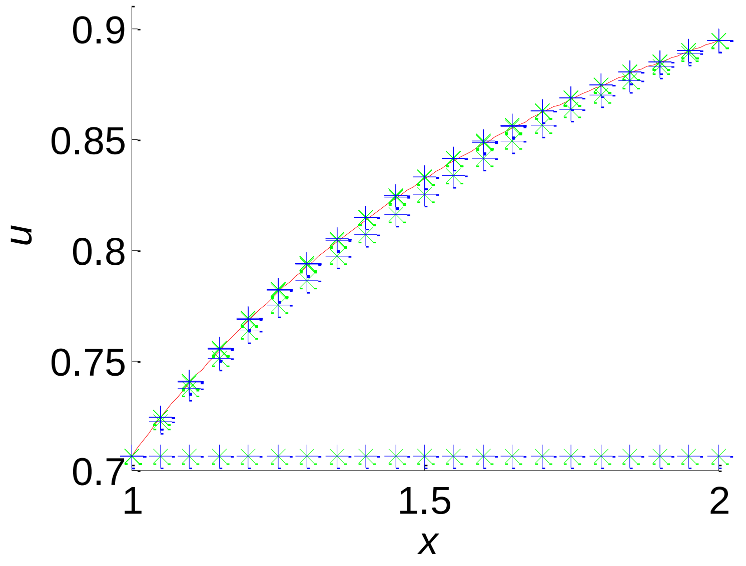
× PFDM_ , + PLSM EE_ ,
— exact solution

Numerical comparison between CFDM and CLSM

We solve the TPBVP (12) using CFDM and CLSM. Again, we choose $\mathcal{D} = \mathcal{D}\_$ and EE_.

Table 3. $\epsilon^{(k)} = \left\| \mathbf{u}_h^{(k+1)} - \mathbf{u}_h^{(k)} \right\|_{L2}$

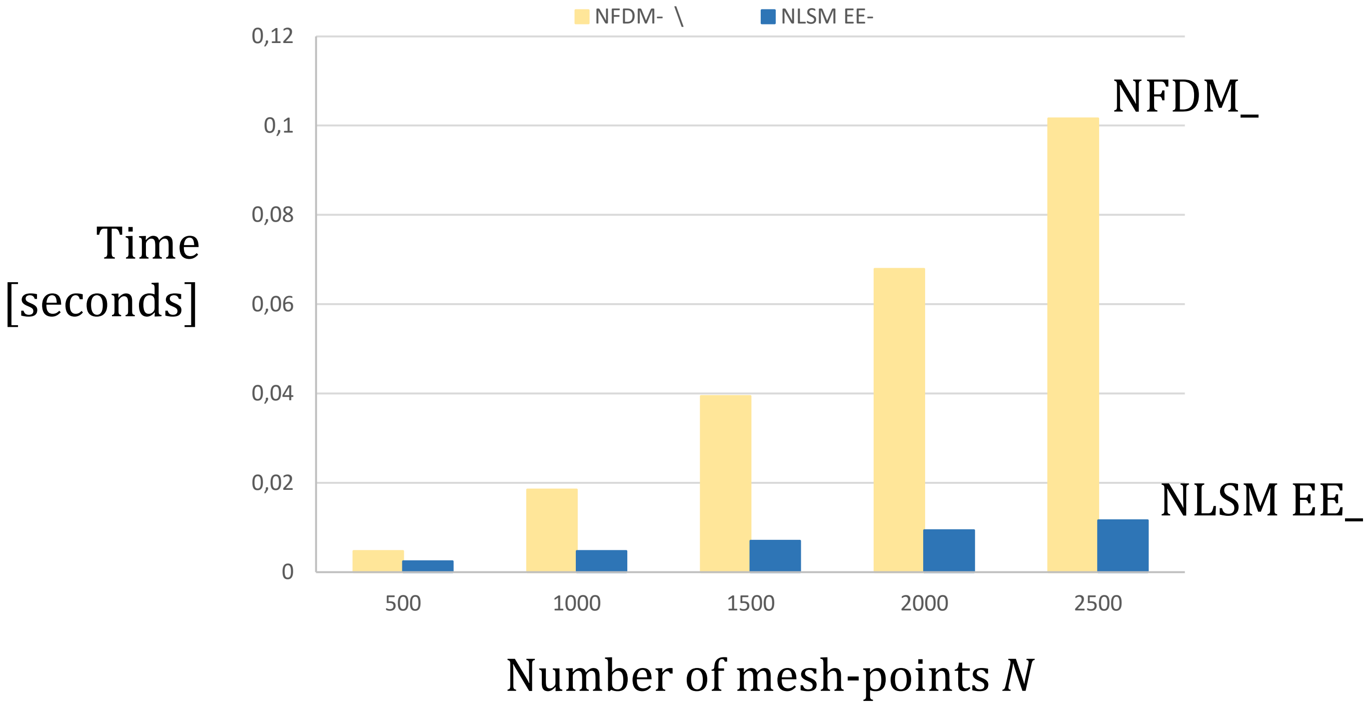
k	$\epsilon^{(k)}$, CFDM _	$\epsilon^{(k)}$, CLSM EE_
0	1.257774292959600e-01	1.257774292959598e-01
1	5.659109740927136e-03	5.659109740927313e-03
2	3.008316690785012e-04	3.008316690785886e-04
3	1.210372256400646e-05	1.210372256404309e-05
4	1.303905575804010e-06	1.303905575772519e-06
5	6.240213217768302e-08	6.240213222206902e-08
6	4.576333624407518e-09	4.576333693878961e-09
7	3.699691119783297e-10	3.699691065145836e-10
8	1.547497895480243e-11	1.547502506938748e-11



× CFDM_ , + CLSM EE_ , — exact solution

Comparing the time-efficiency of NFDM and NLSM

For NFDM we use the MATLAB backslash operator: `L\G`. It is faster than `inv(L)*G`.



The NLSM is $O(N)$ operations and is much faster than NFDM. The Thomas method is also $O(N)$ but cannot be applied directly for all BCs.

We apply the proposed NLSM with the Heun's method (which is RK2 method).

Table 2. $e_h = \|\mathbf{u}_{exact} - \mathbf{u}_h\|_{L2}$

N	h	e_h	e_{2h}/e_h
21	0.05	1.4332e-05	
41	0.025	3.5150e-06	4.0775
81	0.0125	8.7042e-07	4.0382
161	0.00625	2.1658e-07	4.0190
321		5.4017e-08	4.0094
641		1.3488e-08	4.0047
1281		3.3701e-09	4.0024
2561		8.4228e-10	4.0012
5121		2.1054e-10	4.0006

The method is the required $O(h^2)$.

Let $u(x;v_a^k)$ be a solution to the following IVP (Cauchy problem):

$$u''(x;v_a^k) = f\left(x, u(x;v_a^k), u'(x;v_a^k)\right), x \in (a,b), \tag{13}$$

$$u(a;v_a^k) = u_a, u'(a;v_a^k) = v_a^k. \tag{14}$$

The function $u(x;v_a^k)$ is called a *shooting-trajectory*.

(1) Use $u(x;v_a^k)$ as $u_{(k)}(x)$ and find a TPBVP approximation $u_{(k+1)}(x)$ using:
Newton (QLM), Picard, or constant-slope linearization (page 6).
The function $u_{(k+1)}(x)$ satisfies the BCs, and satisfies approximately the ODE. It is called relaxation-trajectory or *projection-trajectory*.
(2) Use $v_a^{k+1} = u'_{(k+1)}(a)$ as a next initial condition and find $u(x;v_a^{k+1})$.
(3) Repeat the procedure.

If we could find $v_a^{k+1} = function(v_a^k)$, then we have an iteration formula!

Shooting-projection iteration formulae (results)

It turns out that it is possible to find $v_a^{k+1} = function\left(v_a^k\right)$ for all three cases.
Results:

Newton	Picard	constant-slope
$v_a^{k+1} = v_a^k - \dfrac{u\left(b; v_a^k\right) - u_b}{\dfrac{\partial u\left(b; v_a^k\right)}{\partial v_a^k}}$ (shooting by Newton method)	$v_a^{k+1} = v_a^k - \dfrac{u\left(b; v_a^k\right) - u_b}{b - a}$ (shooting-projection method [1]) NEW	$v_a^{k+1} = v_a^k - \dfrac{u\left(b; v_a^k\right) - u_b}{\dfrac{\partial u(b; v_a^0)}{\partial v_a^0}}$ (shooting by constant-slope method)

[1] S. M. Filipov, I. D. Gospodinov, I. Faragó (2017). Shooting-projection method for two-point boundary value problems. Appl. Math. Lett. 72 (2017) 10–15

Derivation of the shooting-projection iteration formula (Picard case)

The Picard linearization method gives:

$$u''_{(k+1)}(x) = f\left(x, u\left(x; v_a^k\right), u'\left(x; v_a^k\right)\right), x \in (a, b), \tag{15}$$

$$u_{(k+1)}(a) = u_a, u_{(k+1)}(b) = u_b. \tag{16}$$

Since $u\left(x; v_a^k\right)$ is a solution to the Cauchy problem (13), (14), eqn. (15) gives:

$$u''_{(k+1)}(x) = u''\left(x; v_a^k\right), x \in (a, b), \tag{17}$$

Integrating (17) on $[a, x]$, and then integrating the result on $[a, b]$, we get:

$$u_{(k+1)}(b) - u_{(k+1)}(a) - u'_{(k+1)}(a)(b-a) = u\left(b; v_a^k\right) - u\left(a; v_a^k\right) - u'\left(a; v_a^k\right)(b-a) \tag{18}$$

Finally, denoting $u'_{(k+1)}(a) = v_a^{k+1}$, and using the BCs (16) and the ICs (14), we get:

$$\boxed{v_a^{k+1} = v_a^k - \frac{u\left(b; v_a^k\right) - u_b}{b-a}}$$

Let $y(x) = u_{(k+1)}(x) - u\left(x; v_a^k\right)$. Since $u\left(x; v_a^k\right)$ satisfies (13), the QLM (page 6) gives:

$$y''(x) = q_{(k)}(x)y(x) + p_{(k)}(x)y'(x), x \in (a,b), \tag{19}$$

$$y(a) = 0, y(b) = u_b - u\left(b; v_a^k\right). \tag{20}$$

Let us denote $u'_{(k+1)}(a) = v_a^{k+1}$, and replace the BCs (20) by the ICs:

$$y(a) = 0, y'(a) = v_a^{k+1} - v_a^k. \tag{21}$$

Now, we introduce $z(x)$ such that $y(x) = \left(v_a^{k+1} - v_a^k\right)z(x)$. Then, (19) and (21) yield:

$$z''(x) = q_{(k)}(x)z(x) + p_{(k)}(x)z'(x), x \in (a,b), \tag{22}$$

$$z(a) = 0, z'(a) = 1. \tag{23}$$

However, differentiating (13), (14) wrt v_a^k gives (22), (23). $\Longrightarrow z(x) = \partial u\left(x; v_a^k\right)/\partial v_a^k$.

At $x = b$, we have: $y(b) = \left(v_a^{k+1} - v_a^k\right)z(b) \quad \Longrightarrow$

$$\boxed{v_a^{k+1} = v_a^k - \frac{u\left(b; v_a^k\right) - u_b}{\dfrac{\partial u\left(b; v_a^k\right)}{\partial v_a^k}}}$$

Conclusions

The Newton, Picard, and constant-slope linearization methods can be used to derive the respective:

(i) FDMs (relaxation methods)
(ii) shooting-methods

Based on results (i), we have proposed a replacement of the finite-difference methods for nonlinear TPBVPs (the relaxation methods) by respective successive application of the linear shooting method. The approach removes the necessity of working with matrices altogether. Instead, it achieves the same result by solving one or two IVPs. It reduces the number of computational operations from $O(N^3)$ to only $O(N)$.

Based on results (ii), we have 'discovered' the shooting by Picard method (recently proposed by the authors as shooting-projection method). It has some advantages over the other shooting methods and the FDMs, e.g. greater stability in certain situations.

Thank you!