

Article

On the Approximated Solution of a Special Type of Nonlinear Third-Order Matrix Ordinary Differential Problem

Emilio Defez ¹, Javier Ibáñez ^{2,*}, José M. Alonso ², Michael M. Tung ¹ and Teresa Real-Herráiz ¹

- ¹ Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain; edefez@imm.upv.es (E.D.); mtung@mat.upv.es (M.M.T.); tereaher@upv.es (T.R.-H.)
- ² Instituto de Instrumentación para Imagen Molecular, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain; jmalonso@dsic.upv.es
- * Correspondence: jjibanez@upv.es

Abstract: Matrix differential equations are at the heart of many science and engineering problems. In this paper, a procedure based on higher-order matrix splines is proposed to provide the approximated numerical solution of special nonlinear third-order matrix differential equations, having the form $Y^{(3)}(x) = f(x, Y(x))$. Some numerical test problems are also included, whose solutions are computed by our method.

Keywords: higher-order matrix splines; third-order matrix differential equations



Citation: Defez, E.; Ibáñez, J.; Alonso, J.M.; Tung, M.M.; Real-Herráiz, T. On the Approximated Solution of a Special Type of Nonlinear Third-Order Matrix Ordinary Differential Problem. *Mathematics* **2021**, *9*, 2262. <https://doi.org/10.3390/math9182262>

Academic Editor: Luca Gemignani

Received: 23 July 2021

Accepted: 11 September 2021

Published: 15 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In different areas of engineering and applied science, such as those described in [1,2], third-order ordinary differential equation arise in the form of:

$$y^{(3)} = f(x, y), \quad x \in [a, b], \quad (1)$$

with initial conditions $y(a) = y_a, y'(a) = y'_a$, and $y''(a) = y''_a$. Here, first and second derivatives are not part of the equation to be solved.

For the resolution of this type of equation, various different numerical techniques have been developed and exist in the literature [3–7]. Nonetheless, they frequently depend on very well-known numerical methods, once the original equations have been converted into a system of first-order ordinary differential equations. However, this conversion requires an increase in the number of unknowns of the problem, with the unavoidable growth in the computational cost that this entails. On the other hand, it is possible to use direct-integration methods to solve higher-order differential equations, such as (1), which have been proven to be computationally efficient and offer results with a very satisfactory accuracy (see [8–10] and the references therein).

In this paper, we present a new algorithm for solving matrix third-order differential equations of the type shown in Equation (1). Regarding the mathematical notation to be used hereinafter, we decided to adopt the formulation commonly used in matrix calculus, which is also part of the papers [11,12], previously written by the authors. Accordingly, given a vector $x \in \mathbb{C}^n$, its Euclidean norm is defined as $\|x\|_2 = (x^T x)^{1/2}$ and its 1-norm as $\|x\|_1 = \sum_{i=1}^n |x_i|$. Likewise, for a rectangular $m \times n$ matrix $A \in \mathbb{C}^{m \times n}$, its 2-norm is:

$$\|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}.$$

Suppose $A \in \mathbb{C}^{m \times n}$ and $B \in \mathbb{C}^{p \times q}$. Then, the Kronecker product $A \otimes B$ is the $pm \times qn$ block matrix:

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{pmatrix}.$$

If $A \in \mathbb{C}^{m \times n}$, the column-vector operator on it is defined by:

$$\text{vec}(A) = \begin{pmatrix} \mathbf{A}_{\bullet 1} \\ \vdots \\ \mathbf{A}_{\bullet n} \end{pmatrix}, \text{ being } \mathbf{A}_{\bullet k} = \begin{pmatrix} a_{1k} \\ \vdots \\ a_{mk} \end{pmatrix}.$$

The derivative of a matrix $A \in \mathbb{C}^{m \times n}$ with respect to a matrix $B \in \mathbb{C}^{p \times q}$ was defined by [13] as:

$$\frac{\partial A}{\partial B} = \begin{pmatrix} \frac{\partial A}{\partial b_{11}} & \dots & \frac{\partial A}{\partial b_{1q}} \\ \vdots & & \vdots \\ \frac{\partial A}{\partial b_{p1}} & \dots & \frac{\partial A}{\partial b_{pq}} \end{pmatrix}, \text{ where } \frac{\partial A}{\partial b_{ij}} = \begin{pmatrix} \frac{\partial a_{11}}{\partial b_{ij}} & \dots & \frac{\partial a_{1n}}{\partial b_{ij}} \\ \vdots & & \vdots \\ \frac{\partial a_{m1}}{\partial b_{ij}} & \dots & \frac{\partial a_{mn}}{\partial b_{ij}} \end{pmatrix}.$$

In addition, the derivative of the product of two matrices $A \in \mathbb{C}^{m \times n}$ and $B \in \mathbb{C}^{n \times s}$ with respect to another matrix $C \in \mathbb{C}^{p \times q}$ is given by:

$$\frac{\partial AB}{\partial C} = \frac{\partial A}{\partial C} [I_q \otimes B] + [I_p \otimes A] \frac{\partial B}{\partial C}, \tag{2}$$

I_q and I_p being, respectively, identity matrices of orders q and p . Moreover, given the above matrices, even the chain rule is also applicable:

$$\frac{\partial C}{\partial A} = \left[\frac{\partial [\text{vec}(B)]^\top}{\partial A} \otimes I_p \right] \left[I_n \otimes \frac{\partial C}{\partial [\text{vec}(B)]} \right]. \tag{3}$$

This paper is organized as follows. In Section 2, the proposed method is described in detail. Then, in Section 3, an algorithm for solving this type of equation and its corresponding implementation in MATLAB are included. Next, in Section 4, distinct problems are numerically solved via the suggested method. Finally, conclusions are given in Section 5.

2. Description of the Method

Let the following third-order matrix differential equation be:

$$\left. \begin{aligned} Y^{(3)}(x) &= f(x, Y(x)) \\ Y(a) &= Y_a \\ Y'(a) &= Y'_a \\ Y''(a) &= Y''_a \end{aligned} \right\}, \quad a \leq x \leq b, \tag{4}$$

where $Y(x) \in \mathbb{C}^{p \times q}$ is the unknown matrix and $Y_a, Y'_a, Y''_a \in \mathbb{C}^{p \times q}$ are matrices that collect the initial solution. $f : [a, b] \times \mathbb{C}^{p \times q} \rightarrow \mathbb{C}^{p \times q}$ is a matrix-valued function and, as the differentiability class, $f \in \mathcal{C}^s(T)$, being $s \geq 1$, with:

$$T = \{(x, Y); a \leq x \leq b, Y \in \mathbb{C}^{p \times q}\}. \tag{5}$$

To ensure the uniqueness and the existence of $Y(x)$ as the differentiable continuously solution of problem (4) (see [14], p. 99), f also complies with the global Lipschitz condition:

$$\|f(x, Y_1) - f(x, Y_2)\| \leq L\|Y_1 - Y_2\|, \quad a \leq x \leq b, Y_1, Y_2 \in \mathbb{C}^{p \times q}, \tag{6}$$

where $L > 0$ is the Lipschitz constant.

The initial interval $[a, b]$, in which the solution must be provided, is partitioned into n subintervals, each of size $h = (b - a) / n$, as follows:

$$\Delta_{[a,b]} = \{a = x_0 < x_1 < \dots < x_n = b\}, \quad x_i = a + ih, \quad i = 0, 1, \dots, n. \tag{7}$$

In each of the n aforementioned intervals and taking s as the order of the differentiability class of the function f , a matrix spline $S(x)$ of order $m \in \mathbb{N}$, with $3 \leq m \leq s$, is defined. Thus, the differential Equation (4) is approximately solved so that $S(x) \in \mathcal{C}^3([a, b])$.

For the first subinterval, the spline is designed as:

$$\begin{aligned} S_{|[a,a+h]}(x) &= Y(a) + Y'(a)(x - a) + \frac{1}{2!}Y''(a)(x - a)^2 + \frac{1}{3!}Y^{(3)}(a)(x - a)^3 \\ &+ \frac{1}{4!}Y^{(4)}(a)(x - a)^4 + \dots + \frac{1}{(m - 1)!}Y^{(m-1)}(a)(x - a)^{m-1} \\ &+ \frac{1}{m!}A_0(x - a)^m. \end{aligned} \tag{8}$$

As can be easily verified, this spline fulfills Equation (4) at $x = a$, since:

$$S_{|[a,a+h]}(a) = Y(a) = Y_a, \quad S'_{|[a,a+h]}(a) = Y'(a) = Y'_a, \quad S''_{|[a,a+h]}(a) = Y''(a) = Y''_a$$

and:

$$S^{(3)}_{|[a,a+h]}(a) = Y^{(3)}(a) = f(a, Y(a)) = f(a, S_{|[a,a+h]}(a)).$$

Obviously, the values corresponding to $Y^{(4)}(a), Y^{(5)}(a), Y^{(6)}(a), \dots, Y^{(m-1)}(a)$ and the matrix $A_0 \in \mathbb{C}^{p \times q}$ must be previously determined to fully define the spline (8). Based on this, the procedure given in [12] can be followed to calculate the fourth-order derivative $Y^{(4)}(x)$, that is,

$$\begin{aligned} Y^{(4)}(x) &= \frac{\partial f(x, Y(x))}{\partial x} + \left[[\mathbf{vec} f(x, Y(x))]^\top \otimes I_p \right] \frac{\partial f(x, Y(x))}{\partial \mathbf{vec} Y(x)} \\ &= g_1(x, Y(x), Y'(x)), \end{aligned} \tag{9}$$

where $g_1 \in \mathcal{C}^{s-1}(T)$. As a result, $Y^{(4)}(a)$ can be estimated as:

$$Y^{(4)}(a) = g_1(a, Y(a), Y'(a)) = g_1(a, Y_a, Y'_a).$$

Let us assume that $f \in \mathcal{C}^s(T)$, for $s \geq 2$. Thus, the second partial derivatives of f exist and are also continuous. The fifth-order derivative $Y^{(5)}(x)$ can be expressed as:

$$\begin{aligned} Y^{(5)}(x) &= \frac{\partial^2 f(x, Y(x))}{\partial x^2} + \left([\mathbf{vec} f(x, Y(x))]^\top \otimes I_p \right) \frac{\partial}{\partial x} \left(\frac{\partial f(x, Y(x))}{\partial \mathbf{vec} Y(x)} \right) \\ &+ \left(\frac{\partial [\mathbf{vec} f(x, Y(x))]^\top}{\partial x} \otimes I_p \right) \frac{\partial f(x, Y(x))}{\partial \mathbf{vec} Y(x)} \\ &+ \left([\mathbf{vec} f(x, Y(x))]^\top \otimes I_p \right) \frac{\partial}{\partial \mathbf{vec} Y(x)} \left(\frac{\partial f(x, Y(x))}{\partial x} \right) \\ &+ \left([\mathbf{vec} f(x, Y(x))]^\top \otimes I_p \right) \left(\frac{\partial [\mathbf{vec} f(x, Y(x))]^\top}{\partial \mathbf{vec} Y(x)} \otimes I_p \right) \frac{\partial f(x, Y(x))}{\partial \mathbf{vec} Y(x)} \\ &+ \left([\mathbf{vec} f(x, Y(x))]^\top \otimes I_p \right) \left([\mathbf{vec} f(x, Y(x))]^\top \otimes I_{p^2q} \right) \frac{\partial^2 f(x, Y(x))}{(\partial \mathbf{vec} Y(x))^2} \\ &= g_2(x, Y(x), Y'(x), Y''(x)) \in \mathcal{C}^{s-2}(T). \end{aligned} \tag{10}$$

Therefore, $Y^{(5)}(a) = g_2(a, Y(a), Y'(a), Y''(a)) = g_2(a, Y_a, Y'_a, Y''_a)$ can be immediately evaluated. For the calculation of all the other higher-order derivatives, from $Y^{(6)}(x)$ to $Y^{(m-1)}(x)$, the same previously described procedure can be carried out. In this way, we have:

$$\left. \begin{aligned} Y^{(6)}(x) &= g_3(x, Y(x), Y'(x), Y''(x), Y'''(x)) \in \mathcal{C}^{s-3}(T) \\ &\vdots \\ Y^{(m-1)}(x) &= g_{m-4}(x, Y(x), Y'(x), \dots, Y^{(m-4)}(x)) \in \mathcal{C}^{s-(m-4)}(T) \end{aligned} \right\}. \tag{11}$$

At this point, $Y^{(6)}(a), \dots, Y^{(m-1)}(a)$ can be determined by just replacing $x = a$ in (11). Once all these derivatives have been worked out, the matrix parameter A_0 should be determined as well. For this purpose, suppose that (8) is the solution of Equation (4) at the point $x = a + h$. Then, one obtains:

$$S_{|[a,a+h]}^{(3)}(a+h) = f(a+h, S_{|[a,a+h]}(a+h)), \tag{12}$$

Precisely, it is from here that the following implicit matrix equation can be formulated, with A_0 being its only unknown:

$$A_0 = \frac{(m-3)!}{h^{m-3}} \left[f(a+h, Y(a) + Y'(a)h + \dots + \frac{h^{m-1}}{(m-1)!} Y^{(m-1)}(a) + \frac{h^m}{m!} A_0) - Y^{(3)}(a) - Y^{(4)}(a)h - \dots - \frac{1}{(m-4)!} Y^{(m-1)}(a)h^{m-4} \right]. \tag{13}$$

Once this equation has been solved for A_0 , under the assumption that this solution is unique, all the parameters that form the spline in the interval $[a, a + h]$ will have been determined, and its analytical expression can be obtained.

For the next interval $[a + h, a + 2h]$, the matrix spline can be expressed as:

$$S_{|[a+h,a+2h]}(x) = \sum_{j=0}^2 \frac{S_{|[a,a+h]}^{(j)}(a+h)}{j!} (x - (a+h))^j + \sum_{j=3}^{m-1} \frac{\overline{Y^{(j)}(a+h)}}{j!} (x - (a+h))^j + \frac{A_1}{m!} (x - (a+h))^m, \tag{14}$$

where

$$\overline{Y^{(3)}(a+h)} = f(a+h, S_{|[a,a+h]}(a+h)). \tag{15}$$

After having evaluated the corresponding derivatives of $Y(x)$ using $S_{|[a,a+h]}(a+h)$ in (9)–(11), the expressions for $\overline{Y^{(4)}(a+h)}, \dots, \overline{Y^{(m-1)}(a+h)}$ are obtained, and they can be rewritten in the form:

$$\left. \begin{aligned} \overline{Y^{(4)}(a+h)} &= g_1(a+h, S_{|[a,a+h]}(a+h), S'_{|[a,a+h]}(a+h)) \\ &\vdots \\ \overline{Y^{(m-1)}(a+h)} &= g_{m-4}(a+h, S_{|[a,a+h]}(a+h), \dots, S_{|[a,a+h]}^{(m-4)}(a+h)) \end{aligned} \right\}. \tag{16}$$

The matrix spline $S(x)$, defined by means of (8) and (14), is of differentiability class $\mathcal{C}^3([a, a + h] \cup [a + h, a + 2h])$, and therefore, it satisfies Equation (4) at the point $x = a + h$. Similar to the previous interval, all spline coefficients have already been identified, with the exception of $A_1 \in \mathbb{R}^{p \times q}$. Nevertheless, its value can be determined if the mentioned spline is considered as a solution of (4) at $x = a + 2h$:

$$S_{|[a+h,a+2h]}^{(3)}(a+2h) = f(a+2h, S_{|[a+h,a+2h]}(a+2h)),$$

which leads to the following matrix equation, whose only unknown is precisely A_1 :

$$A_1 = \frac{(m-3)!}{h^{m-3}} \left[f \left(a + 2h, \sum_{j=0}^2 \frac{S_{|[a,a+h]}^{(j)}(a+h)}{j!} h^j + \sum_{j=3}^{m-1} \frac{\overline{Y^{(j)}(a+h)}}{j!} h^j \right) + \frac{A_1 h^m}{m!} \overline{Y^{(3)}(a+h)} - \overline{Y^{(4)}(a+h)} h - \dots - \frac{h^{m-4}}{(m-4)!} \overline{Y^{(m-1)}(a+h)} \right]. \tag{17}$$

Assuming once again the uniqueness of the solution of the above equation for A_1 , the spline is completely specified in the interval $[a + h, a + 2h]$.

Now, repeating this procedure iteratively, the interval $[a + ih, a + (i + 1)h]$ is reached, whose resulting spline then would be defined in the form:

$$S_{|[a+ih,a+(i+1)h]}(x) = \sum_{j=0}^2 \frac{S_{|[a+(i-1)h,a+ih]}^{(j)}(a+ih)}{j!} (x - (a + ih))^j + \sum_{j=3}^{m-1} \frac{\overline{Y^{(j)}(a+ih)}}{j!} (x - (a + ih))^j + \frac{A_i}{m!} (x - (a + ih))^m, \tag{18}$$

where:

$$\overline{Y^{(3)}(a+ih)} = f \left(a + ih, S_{|[a+(i-1)h,a+ih]}(a+ih) \right), \tag{19}$$

and, similarly to what has been described previously,

$$\left. \begin{aligned} \overline{Y^{(4)}(a+ih)} &= g_1 \left(a + ih, S_{|[a+(i-1)h,a+ih]}(a+ih), S'_{|[a+(i-1)h,a+ih]}(a+ih) \right) \\ &\vdots \\ \overline{Y^{(m-1)}(a+ih)} &= g_{m-4} \left(a + ih, S_{|[a+(i-1)h,a+ih]}(a+ih), \dots, S_{|[a+(i-1)h,a+ih]}^{(m-4)}(a+ih) \right) \end{aligned} \right\}. \tag{20}$$

Under this premise, the matrix spline $S(x) \in C^3 \left(\bigcup_{j=0}^i [a + jh, a + (j + 1)h] \right)$ satisfies the differential Equation (4) at the point $x = a + ih$. If we additionally consider that $S_{|[a+ih,a+(i+1)h]}(x)$ fulfills the mentioned equation at point $x = a + (i + 1)h$, then:

$$S_{|[a+ih,a+(i+1)h]}^{(3)}(a + (i + 1)h) = f \left(a + (i + 1)h, S_{|[a+ih,a+(i+1)h]}(a + (i + 1)h) \right).$$

After expanding this previous formulation, one arrives at:

$$A_i = \frac{(m-3)!}{h^{m-3}} \left[f \left(a + (i + 1)h, \sum_{j=0}^2 \frac{S_{|[a+(i-1)h,a+ih]}^{(j)}(a+ih)}{j!} h^j + \sum_{j=3}^{m-1} \frac{\overline{Y^{(j)}(a+ih)}}{j!} h^j \right) + \frac{A_i h^m}{m!} \overline{Y^{(3)}(a+ih)} - \dots - \frac{h^{m-4}}{(m-4)!} \overline{Y^{(m-1)}(a+ih)} \right]. \tag{21}$$

Note that Equations (13) and (17) can be derived from Equation (21) when $i = 0$ or $i = 1$, respectively.

Let us now demonstrate the uniqueness of the solution for these equations by using a fixed-point argument. Thus, let the matrix-valued function $g : \mathbb{C}^{p \times q} \rightarrow \mathbb{C}^{p \times q}$ be defined as follows for given values of h and i :

$$g(T) = \frac{(m-3)!}{h^{m-3}} \left[f \left(a + (i+1)h, \sum_{j=0}^2 \frac{S_{[a+(i-1)h, a+ih]}^{(j)}(a+ih)}{j!} h^j + \sum_{j=3}^{m-1} \frac{\overline{Y^{(j)}(a+ih)}}{j!} h^j \right) + \frac{T}{m!} h^m \right] - \overline{Y^{(3)}(a+ih)} - \dots - \frac{h^{m-4}}{(m-4)!} \overline{Y^{(m-1)}(a+ih)}. \tag{22}$$

Equation (21) is satisfied if and only if $A_i = g(A_i)$, i.e., if the matrix A_i is a fixed point for function $g(T)$. Considering this definition for function $g(T)$ and the global Lipschitz condition (6) for f , we have:

$$\|g(T_1) - g(T_2)\| \leq \frac{Lh^3}{m(m-1)(m-2)} \|T_1 - T_2\|.$$

If we take $h < \sqrt[3]{m(m-1)(m-2)}/L$, the matrix function $g(T)$ will be contractive, and the solutions $A_i, i = 0, 1, \dots, n-1$, of Equation (21) will be unique and, as a consequence, entirely determine the matrix spline. We can therefore state that the following theorem has been proven.

Theorem 1. Take the third-order matrix differential equation expressed in (4) and the Lipschitz constant L described in (6). If the interval $[a, b]$ is partitioned as in (7) by considering a step size h such as:

$$h < \sqrt[3]{m(m-1)(m-2)}/L,$$

then the matrix spline $S(x)$ of order $m, 3 \leq m \leq s$, s being the order of the differentiability class of the function f , is of class $C^3[a, b]$, and it can be defined as in (18) for each subinterval $[a + ih, a + (i + 1)h], i = 0, 1, \dots, n - 1$.

If an analysis similar to the one performed in [15] were carried out, we could conclude that the defined splines have a global error of $O(h^{m-1})$.

3. Algorithm for the MATLAB Program

Consider the following third-order ordinary differential equation for the k -th step:

$$\left. \begin{aligned} Y^{(3)}(x) &= f(x, Y(x)) \\ Y(x_k) &= Y_0, Y'(x_k) = Y_1, Y''(x_k) = Y_2 \end{aligned} \right\} x_k \leq x \leq x_k + h, \tag{23}$$

where h is the step size, $x_k = a + kh$ and Y_0, Y_1 , and Y_2 are, respectively, the matrices Y, Y' , and Y'' obtained in the previous step at x_k . If we denote by $S_k(x)$ the spline of order m in the interval $[x_k, x_k + h]$, then:

$$S_k(x_k + h) = \sum_{i=0}^{m-1} \frac{Y^{(i)}(x_k)h^i}{i!} + \frac{h^m}{m!} A_k \equiv B_k^{(0)} + \frac{h^m}{m!} A_k, \tag{24}$$

$$S'_k(x_k + h) = \sum_{k=1}^{m-1} \frac{Y^{(i)}(x_k)h^{i-1}}{(i-1)!} + \frac{h^{m-1}}{(m-1)!} A_k \equiv B_k^{(1)} + \frac{h^{m-1}}{(m-1)!} A_k, \tag{25}$$

$$S''_k(x_k + h) = \sum_{k=2}^{m-1} \frac{Y^{(i)}(x_k)h^{i-2}}{(i-2)!} + \frac{h^{m-2}}{(m-2)!} A_k \equiv B_k^{(2)} + \frac{h^{m-2}}{(m-2)!} A_k, \tag{26}$$

$$S'''_k(x_k + h) = \sum_{k=3}^{m-1} \frac{Y^{(i)}(x_k)h^{i-3}}{(i-3)!} + \frac{h^{m-3}}{(m-3)!} A_k \equiv B_k^{(3)} + \frac{h^{m-3}}{(m-3)!} A_k. \tag{27}$$

If we substitute expressions (24) and (27) into differential Equation (23), we obtain:

$$B_k^{(3)} + \frac{h^{m-3}}{(m-3)!} A_k = f \left(x, B_k^{(0)} + \frac{h^m}{m!} A_k \right).$$

Then, the matrix A_k can be found by using the following fixed-point iteration:

$$A_k = \frac{(m-3)!}{h^{m-3}} \left[f \left(x, B_k^{(0)} + \frac{h^m}{m!} A_k \right) - B_k^{(3)} \right]. \quad (28)$$

Hence, the approximated values for Y , Y' , and Y'' at $x_k + h$ are:

$$\begin{aligned} Y(x_k + h) &= B_k^{(0)} + \frac{h^m}{m!} A_k, \\ Y'(x_k + h) &= B_k^{(1)} + \frac{h^{m-1}}{(m-1)!} A_k, \\ Y''(x_k + h) &= B_k^{(2)} + \frac{h^{m-2}}{(m-2)!} A_k. \end{aligned}$$

Figure 1 shows the MATLAB code that approximately computes the solution matrices $Y(b)$, $Y'(b)$, and $Y''(b)$ of the third-order ordinary differential Equation (4). This code uses the cell-array data type for storing sets of matrices. In Line 31, $Y^{(i)}(x_k)$, $i = 0, 1, \dots, m$, are obtained and stored in the cell-array variable ym by invoking the `f` MATLAB function to return the matrices that appear in expressions (24)–(27). In Lines 11–16, the expressions $B_k^{(0)}$, $B_k^{(1)}$, $B_k^{(2)}$, and $B_k^{(3)}$ from Equations (24)–(27) are computed. In Lines 38–47, the matrix A_k is worked out by using the fixed-point iteration. Finally, in Lines 48–50, $Y(x_k + h)$, $Y'(x_k + h)$, and $Y''(x_k + h)$ are computed approximately.

```

1 function [x,y]=spline3order(f,y,a,b,h,m)
2 % This function computes the solution of a third-order
3 % matrix differential equation y'''=f(x,y), at x in [a,b], with
4 % the initial conditions y{1}=y(a), y{2}=y'(a) and y{3}=y''(a).
5 %
6 % Input data:
7 % - f is a MATLAB function which returns y and its
8 % derivatives y', y'', y''', ..., y(m).
9 % - y is a cell array of 3 elements with the matrices y, y'
10 % and y'' at the point x=a.
11 % - a and b define the considered integration interval.
12 % - h is the step size.
13 % - m is the spline order.
14 %
15 % Output data:
16 %
17 % - x stores the value, near at b, in which the solution is
18 % computed.
19 % - y is a cell array of 3 elements with the solution y, y'
20 % and y'' at the previous point x in the form y{1}=y(x),
21 % y{2}=y'(x) and y{3}=y''(x).
22 %
23 [n1,n2]=size(y{1});
24 ph(1)=h;
25 for k=2:m
26     ph(k)=ph(k-1)*h;
27 end
28 nt=round((b-a)/h);
29 x=a;
30 for k=1:nt
31     ym=f(x,y,1);
32     for j=1:4
33         Bk{j}=ym{j};
34         for i=j+1:m
35             Bk{j}=Bk{j}+ym{i}*ph(i-j)/factorial(i-j);
36         end
37     end
38     x=x+h;
39     aux=factorial(m-3)/ph(m-3);
40     Ak=ones(n1,n2);
41     ea=1;
42     while ea>=eps/2
43         Skm{1}=Bk{1}+Ak*ph(m)/factorial(m);
44         Ak1=aux*(f(x,Skm,0)-Bk{4});
45         ea=norm(Ak-Ak1);
46         Ak=Ak1;
47     end
48     for j=1:3
49         y{j}=Bk{j}+Ak*ph(m-j+1)/factorial(m-j+1);
50     end
51 end
52 end

```

Figure 1. MATLAB code for computing approximate solutions by means of m -th order splines.

Figure 2 reproduces the MATLAB function f for the thin-film flow of problem (33). The dotted Line 24 indicates the completion of the list of all derivatives until the m -th one, m being the spline order.

The memory requirements for this function are $(m + 10)$ matrices, i.e., m matrices for the cell-array variable ym , 4 matrices for the cell-array variable Bk , 3 matrices for the variables Ak , $Ak1$, and Sk_m , and 3 matrices for the cell-array variable y .

```

1 function ym=f(x,y,flag_out)
2 % Example of the thin-film flow problem.
3 % This function returns y and its derivatives y', y'', y''', ...,
4 % y(m), where m is the spline order, corresponding to the
5 % differential equation y'''(x) = f(x,y(x)).
6 %
7 % Input data:
8 % - x is the point at the X axis.
9 % - y is a cell array of 3 elements with the matrices y, y'
10 % and y'' at the previous point x.
11 % - flag_out is a boolean value: if it is equal to 0, the
12 % function just returns y'''. Otherwise, it returns
13 % y, y', y'', y''', ..., y(m).
14 %
15 % Output data:
16 % - ym is a cell array that either stores y and its
17 % derivatives or just y'''.
18 %
19 if flag_out
20     ym=y;
21     ym{4}=y{1}^(-2);
22     ym{5}=-2*y{1}^(-3)*y{2};
23     ym{6}=6*y{1}^(-4)*y{2}^2-2*y{1}^(-3)*y{3};
24     .....
25 else
26     ym=y{1}^(-2);
27 end
28 end

```

Figure 2. Function f for the thin-film flow of problem (33).

4. Numerical Test Examples

This section discusses the approximation for solutions of seven distinct third-order differential equations. For all of them, special *Mathematica* and MATLAB software packages were employed to obtain the solutions of the algebraic equations that emerge after applying the algorithm described in the preceding section.

4.1. A Scalar Inhomogeneous Non-Linear Problem

Consider this simple test problem from [1]:

$$\left. \begin{aligned} y^{(3)}(x) &= (\cos(x) - 1) \cos(x) + y^2 - 1 \\ y(0) &= 0 \\ y'(0) &= 1 \\ y''(0) &= 0 \end{aligned} \right\}, \quad 0 \leq x \leq 1, \quad (29)$$

whose exact solution $y(x) = \sin(x)$ is known. The function $f(x, y) = (\cos(x) - 1) \cos(x) + y^2 - 1$ is not Lipschitz for all of \mathbb{R}^2 . However, the particular region $D = (0, 1) \times (-1, 1) \subset \mathbb{R}^2$ can be chosen, because with constant $L = 2$, function $f(x, y)$ is locally Lipschitz. Therefore, it is possible to apply the proposed method, obtaining sixth-order splines ($m = 6$) by taking $h < \sqrt[3]{60} \approx 3.91$. The analytical expressions of the distinct polynomials that compose the spline for $h = 0.1$ are listed in Table 1. The maximum errors incurred are shown in Table 2 for each subinterval.

Table 1. Approximation spline for the problem (29).

Subinterval	Approximation Polynomials
[0, 0.1]	$-0.0000347107x^6 + 0.00833333x^5 - 0.166667x^3 + x$
[0.1, 0.2]	$-0.00031184x^6 + 0.00849961x^5 - 0.0000415695x^4 - 0.166661x^3 - 4.15695 \times 10^{-7}x^2 + 1.x - 2.7713 \times 10^{-10}$
[0.2, 0.3]	$0.0000383134x^6 + 0.00807943x^5 + 0.000168523x^4 - 0.166717x^3 + 7.9880 \times 10^{-6}x^2 + 0.999999x + 2.2133 \times 10^{-8}$
[0.3, 0.4]	$-0.00190054x^6 + 0.0115694x^5 - 0.00244893x^4 - 0.16567x^3 - 0.000227583x^2 + 1.00003x - 1.3913 \times 10^{-6}$
[0.4, 0.5]	$0.00470998x^6 - 0.00429589x^5 + 0.0134163x^4 - 0.174132x^3 + 0.00231086x^2 + 0.999621x + 0.0000256854$
[0.5, 0.6]	$-0.0205597x^6 + 0.0715132x^5 - 0.081345x^4 - 0.110957x^3 - 0.0213795x^2 + 1.00436x - 0.000369154$
[0.6, 0.7]	$0.0731793x^6 - 0.265947x^5 + 0.424846x^4 - 0.51591x^3 + 0.160849x^2 + 0.960625x + 0.00400433$
[0.7, 0.8]	$-0.277197x^6 + 1.20563x^5 - 2.15042x^4 + 1.88767x^3 - 1.10103x^2 + 1.31395x - 0.0372171$
[0.8, 0.9]	$1.02995x^6 - 5.06866x^5 + 10.3982x^4 - 11.4975x^3 + 6.93007x^2 - 1.256x + 0.305443$
[0.9, 1.0]	$-3.8489x^6 + 21.2771x^5 - 48.8798x^4 + 59.6361x^3 - 41.0851x^2 + 16.0295x - 2.28737$

Table 2. Approximation errors for the problem (29).

Subinterval	[0, 0.1]	[0.1, 0.2]	[0.2, 0.3]	[0.3, 0.4]	[0.4, 0.5]
Max. error	1.4872×10^{-11}	6.3015×10^{-11}	6.9036×10^{-10}	6.2645×10^{-10}	6.6951×10^{-9}
Subinterval	[0.5, 0.6]	[0.6, 0.7]	[0.7, 0.8]	[0.8, 0.9]	[0.9, 1.0]
Max. error	1.2629×10^{-8}	6.8791×10^{-8}	2.2188×10^{-7}	8.8033×10^{-7}	8.7895×10^{-7}

4.2. A Scalar Nonhomogeneous Linear Problem

Here, we consider the following equation, studied in [8]:

$$\left. \begin{aligned}
 y^{(3)}(x) &= y + \cos(x) \\
 y(0) &= 0 \\
 y'(0) &= 0 \\
 y''(0) &= 1
 \end{aligned} \right\} , \quad 0 \leq x \leq 1, \tag{30}$$

whose exact solution is given by $y(x) = \frac{1}{2}(e^x - \cos(x) - \sin(x))$. In this case, function $f(x, y) = y + \cos(x)$ is Lipschitz over all \mathbb{R}^2 , with constant $L = 1$. If we choose $h < \sqrt[3]{120} \approx 4.93$, sixth-order splines ($m = 6$) will be achieved. The approximation spline for step size $h = 0.1$ is given in Table 3, and the corresponding maximum errors for each subinterval are shown in Table 4.

Table 3. Approximation spline for the problem (30).

Subinterval	Approximation Polynomials
[0, 0.1]	$0.00142361x^6 + 0.166667x^3 + 0.5x^2$
[0.1, 0.2]	$2.77797 \times 10^{-10} - 1.66678 \times 10^{-8}x + 0.5x^2 + 0.166661x^3 + 0.0000416695x^4 - 0.000166678x^5 + 0.00170141x^6$
[0.2, 0.3]	$-2.1931 \times 10^{-8} + 6.49596 \times 10^{-7}x + 0.499992x^2 + 0.166717x^3 - 0.000166538x^4 + 0.000249737x^5 + 0.0013544x^6$
[0.3, 0.4]	$1.39603 \times 10^{-6} - 0.0000277096x + 0.500228x^2 + 0.165666x^3 + 0.00245931x^4 - 0.0032514x^5 + 0.00329947x^6$
[0.4, 0.5]	$-0.0000256191 + 0.000377517x + 0.497696x^2 + 0.174109x^3 - 0.0133699x^4 + 0.0125778x^5 - 0.00329602x^6$
[0.5, 0.6]	$0.000369394 - 0.00436264x + 0.521397x^2 + 0.110906x^3 + 0.0814334x^4 - 0.0632648x^5 + 0.0219848x^6$
[0.6, 0.7]	$-0.0040011 + 0.0393423x + 0.339292x^2 + 0.515582x^3 - 0.424411x^4 + 0.273965x^5 - 0.0716901x^6$
[0.7, 0.8]	$0.0372102 - 0.313898x + 1.60086x^2 - 1.88741x^3 + 2.15022x^4 - 1.19726x^5 + 0.2786x^6$
[0.8, 0.9]	$-0.305309 + 2.255x - 6.42693x^2 + 11.4922x^3 - 10.3932x^4 + 5.07446x^5 - 1.02801x^6$
[0.9, 1.0]	$2.28658 - 15.0243x + 41.571x^2 - 59.6158x^3 + 48.8635x^4 - 21.2619x^5 + 3.84909x^6$

Table 4. Approximation errors for the problem (30).

Subinterval	[0, 0.1]	[0.1, 0.2]	[0.2, 0.3]	[0.3, 0.4]	[0.4, 0.5]
Max. error	1.4881×10^{-11}	6.3053×10^{-11}	6.9130×10^{-10}	6.3575×10^{-10}	6.7183×10^{-9}
Subinterval	[0.5, 0.6]	[0.6, 0.7]	[0.7, 0.8]	[0.8, 0.9]	[0.9, 1.0]
Max. error	1.256×10^{-8}	6.8932×10^{-8}	2.2147×10^{-7}	8.8065×10^{-7}	3.2087×10^{-6}

4.3. A Scalar Nonlinear Problem

Our next test example focuses on the problem posed in [8]:

$$\left. \begin{aligned} y^{(3)}(x) &= -e^{-y} + 3e^{-2y} - 2e^{-3y} \\ y(0) &= \log(2) \\ y'(0) &= \frac{1}{2} \\ y''(0) &= \frac{1}{4} \end{aligned} \right\}, \quad 0 \leq x \leq 1. \tag{31}$$

The exact solution to this problem is $y(x) = \log(e^x + 1)$. Considering the region $D = (0, 1) \times (0, 10) \subset \mathbb{R}^2$, function $f(x, y) = -e^{-y} + 3e^{-2y} - 2e^{-3y}$ has a partial derivative with respect to its second variable y bounded by $L = 13$ in D . Thus, function $f(x, y)$ is locally Lipschitz for constant $L = 13$. For $h < \sqrt[3]{\frac{120}{13}} \approx 2.09771$, our method is applied, providing this time sixth-order splines ($m = 6$). Table 5 displays the approximation spline for $h = 0.1$. In addition, the maximum errors for each subinterval are listed in Table 6.

Table 5. Approximation spline for the problem (31).

Subinterval	Approximation Polynomials
[0, 0.1]	$\log(2) + 0.5x + 0.125x^2 - 0.00520833x^4 + 0.000346486x^6$
[0.1, 0.2]	$0.693147 + 0.5x + 0.125x^2 + 3.51111 \times 10^{-7}x^3 - 0.00521097x^4 + 0.0000105333x^5 + 0.00032893x^6$
[0.2, 0.3]	$0.693147 + 0.5x + 0.125x^2 + 2.9433 \times 10^{-6}x^3 - 0.00522069x^4 + 0.0000299747x^5 + 0.000312729x^6$
[0.3, 0.4]	$0.693147 + 0.500001x + 0.124989x^2 + 0.0000484757x^3 - 0.00533452x^4 + 0.000181749x^5 + 0.00022841x^6$
[0.4, 0.5]	$0.693148 + 0.499994x + 0.125035x^2 - 0.000102835x^3 - 0.00505081x^4 - 0.000101957x^5 + 0.000346621x^6$
[0.5, 0.6]	$0.693137 + 0.500121x + 0.124401x^2 + 0.00158826x^3 - 0.00758746x^4 + 0.00192736x^5 - 0.000329818x^6$
[0.6, 0.7]	$0.693243 + 0.499064x + 0.128805x^2 - 0.00819903x^3 + 0.00464666x^4 - 0.00622872x^5 + 0.00193576x^6$
[0.7, 0.8]	$0.692217 + 0.50786x + 0.0973911x^2 + 0.0516366x^3 - 0.059463x^4 + 0.0304054x^5 - 0.00678664x^6$
[0.8, 0.9]	$0.700682 + 0.444372x + 0.29579x^2 - 0.279029x^3 + 0.250536x^4 - 0.124594x^5 + 0.0255049x^6$
[0.9, 1.0]	$0.636507 + 0.872202x - 0.892627x^2 + 1.48159x^3 - 1.21665x^4 + 0.527487x^5 - 0.0952508x^6$

Table 6. Approximation errors for the problem (31).

Subinterval	[0, 0.1]	[0.1, 0.2]	[0.2, 0.3]	[0.3, 0.4]	[0.4, 0.5]
Max. error	4.7340×10^{-13}	2.5465×10^{-12}	3.9703×10^{-11}	1.1270×10^{-10}	3.9472×10^{-10}
Subinterval	[0.5, 0.6]	[0.6, 0.7]	[0.7, 0.8]	[0.8, 0.9]	[0.9, 1.0]
Max. error	4.6335×10^{-10}	2.6369×10^{-9}	3.8893×10^{-9}	2.4363×10^{-8}	7.5564×10^{-8}

4.4. A Scalar Nonhomogeneous Problem

An interesting scalar problem, being nonhomogeneous and involving convoluted combinations of trigonometric functions, is:

$$\left. \begin{aligned} y^{(3)}(x) &= -\frac{3}{2}y \sin(2x) + \sin(x) \left(1 + \sin^2(x)\right) \cos(\cos(x)) \\ y(0) &= \sin(1) \\ y'(0) &= 0 \\ y''(0) &= -\cos(1) \end{aligned} \right\}, \quad 0 \leq x \leq 1. \quad (32)$$

Surprisingly, this complicated problem has an exact solution, for details see [8]. The known solution is $y(x) = \sin(\cos(x))$ (there are mistakes both in the initial conditions and in the solution of the problem, $y(x) = \cos(\sin(x))$), given in [8], p. 146). In this case, function:

$$f(x, y) = -\frac{3}{2}y \sin(2x) + \sin(x) \left(1 + \sin^2(x)\right) \cos(\cos(x))$$

is Lipschitz with constant $L = 3/2$, and the proposed method will yield sixth-order splines ($m = 6$) if $h < \sqrt[3]{\frac{240}{3}} \approx 4.30887$. For step size $h = 0.1$, this spline is summarized in Table 7. All corresponding maximum errors are also listed in Table 8.

Table 7. Approximation spline for problem (32).

Subinterval	Approximation Polynomials
[0,0.1]	$\sin[1] - \frac{\cos(1)}{2}x^2 + \left(\frac{\cos(1)}{24} - \frac{\sin(1)}{8}\right)x^4 + 0.0279824x^6$
[0.1,0.2]	$0.841471 + 7.96286 \times 10^{-8}x - 0.270153x^2 + 0.0000265429x^3 - 0.0828703x^4 + 0.000796286x^5 + 0.0266552x^6$
[0.2,0.3]	$0.841471 + 3.18879 \times 10^{-6}x - 0.270192x^2 + 0.00028564x^3 - 0.083842x^4 + 0.00273951x^5 + 0.0250359x^6$
[0.3,0.4]	$0.841466 + 0.000109689x - 0.27108x^2 + 0.00423011x^3 - 0.0937031x^4 + 0.0158877x^5 + 0.0177313x^6$
[0.4,0.5]	$0.841488 - 0.000229253x - 0.268961x^2 - 0.00283119x^3 - 0.0804632x^4 + 0.0026478x^5 + 0.0232479x^6$
[0.5,0.6]	$0.840672 + 0.00956559x - 0.317935x^2 + 0.127767x^3 - 0.27636x^4 + 0.159365x^5 - 0.0289912x^6$
[0.6,0.7]	$0.847811 - 0.0618259x - 0.0204706x^2 - 0.533266x^3 + 0.549931x^4 - 0.391495x^5 + 0.124026x^6$
[0.7,0.8]	$0.774658 + 0.565197x - 2.25984x^2 + 3.7322x^3 - 4.02021x^4 + 2.22001x^5 - 0.497762x^6$
[0.8,0.9]	$1.36838 - 3.88771x + 11.6555x^2 - 19.4601x^3 + 17.7225x^4 - 8.65135x^5 + 1.76711x^6$
[0.9,1.0]	$-3.15253 + 26.2517x - 72.0651x^2 + 104.57x^3 - 85.6362x^4 + 37.2859x^5 - 6.73979x^6$

Table 8. Approximation errors for problem (32).

Subinterval	[0,0.1]	[0.1,0.2]	[0.2,0.3]	[0.3,0.4]	[0.4,0.5]
Max. error	3.4875×10^{-11}	1.811×10^{-10}	2.989×10^{-9}	9.2962×10^{-9}	3.2810×10^{-8}
Subinterval	[0.5,0.6]	[0.6,0.7]	[0.7,0.8]	[0.8,0.9]	[0.9,1.0]
Max. error	4.8765×10^{-8}	2.1830×10^{-7}	2.0686×10^{-7}	1.845×10^{-6}	5.095×10^{-6}

4.5. A Nonlinear Scalar Problem

The following problem consists of modeling the spreading of a thin oil drop on a horizontal surface. It is the thin-film problem given by Example 4.3 in [16]:

$$\left. \begin{aligned} y^{(3)} &= y^{-2} \\ y(0) = y'(0) = y''(0) &= 1 \end{aligned} \right\}, \quad x \in [0, b]. \quad (33)$$

Table 9 details the relative errors for the method by Mechee et al. (M_1) [17], the method by Khataybeh et al. (M_2), and our spline-based method of order $m = 9$ (M_3), for $h = 0.01$ and $b = 1$. For the numerical comparison, the exact solution was obtained from Duffy and Wilson [1].

Table 9. Relative errors of the Mechee (M_1), Khataybeh (M_2), and spline-based (M_3) methods in the film flow problem (33).

x	M_1	M_2	M_3
0.2	8.397402×10^{-7}	8.399040×10^{-7}	8.397170×10^{-7}
0.4	7.596544×10^{-8}	7.643561×10^{-8}	7.598801×10^{-8}
0.6	3.485744×10^{-9}	4.315684×10^{-9}	3.496411×10^{-9}
0.8	4.587537×10^{-11}	1.192760×10^{-9}	3.625209×10^{-11}
1	3.659122×10^{-7}	3.674458×10^{-7}	3.659145×10^{-7}

These results were produced with the MATLAB code shown in Figure 2, which calculates function y and its derivatives.

4.6. A Linear Matrix Problem

Consider the matrix problem:

$$\left. \begin{aligned}
 Y^{(3)}(x) &= AY(x), A = \begin{pmatrix} \frac{817}{68} & \frac{1393}{68} & \frac{448}{68} \\ -\frac{1141}{68} & -\frac{2837}{68} & -\frac{896}{68} \\ \frac{3059}{136} & \frac{4319}{136} & \frac{1592}{136} \end{pmatrix} \\
 y(0) &= \begin{pmatrix} 2 \\ -2 \\ 12 \end{pmatrix} \\
 y'(0) &= \begin{pmatrix} -12 \\ 28 \\ -33 \end{pmatrix} \\
 y''(0) &= \begin{pmatrix} 20 \\ -52 \\ 5 \end{pmatrix}
 \end{aligned} \right\} , 0 \leq x \leq 1. \quad (34)$$

As in the other cases above, its solution (see [18], p. 334) is also known:

$$y(x) = \begin{pmatrix} e^x - 2e^{2x} + 3e^{-3x} \\ 3e^x + 2e^{2x} - 7e^{-3x} \\ -11e^x - 5e^{2x} + 4e^{-3x} \end{pmatrix}.$$

The matrix function $f(x, Y) = AY$ is Lipschitz with constant $L = \|A\|_2 \approx 90.1136$. Therefore, for $h < \sqrt[3]{\frac{210}{90.1136}} \approx 1.32579$, we employ our method and seventh-order splines ($m = 7$) are obtained. Maximum errors for $h = 0.1$ are given in Table 10.

Table 10. Approximation errors for the problem (34).

Subinterval	[0,0.1]	[0.1,0.2]	[0.2,0.3]	[0.3,0.4]	[0.4,0.5]
Max. error	7.5906×10^{-11}	7.2457×10^{-11}	4.3608×10^{-10}	1.4836×10^{-9}	3.7673×10^{-9}
Subinterval	[0.5,0.6]	[0.6,0.7]	[0.7,0.8]	[0.8,0.9]	[0.9,1.0]
Max. error	7.9945×10^{-9}	1.5020×10^{-8}	2.5835×10^{-8}	4.1559×10^{-8}	6.3425×10^{-8}

Figures 3 and 4 display errors committed when considering distinct values of h or different spline orders, respectively. As expected, the errors become smaller as either the

value of h decreases or as the spline order increases. MATLAB software was employed to provide these graphs.

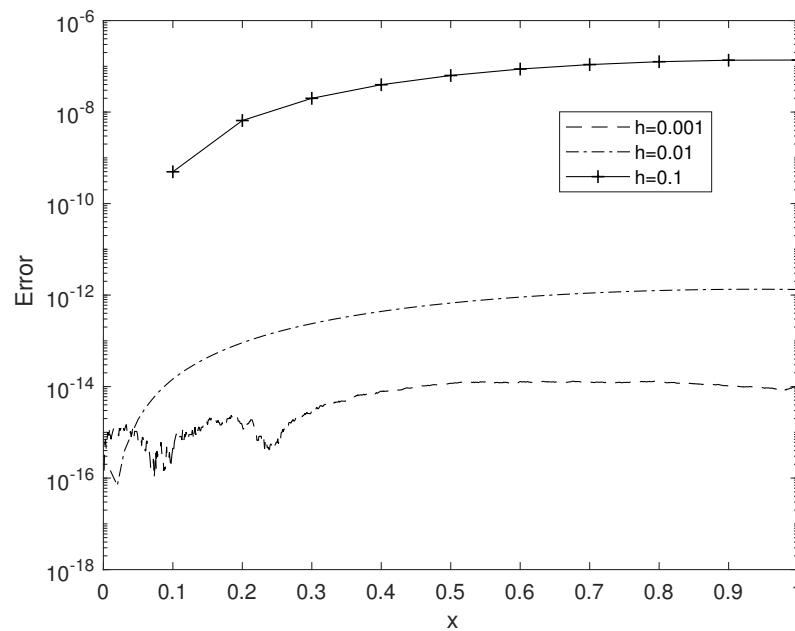


Figure 3. Approximation errors ($m = 7$) for the matrix problem (34) with various step sizes.

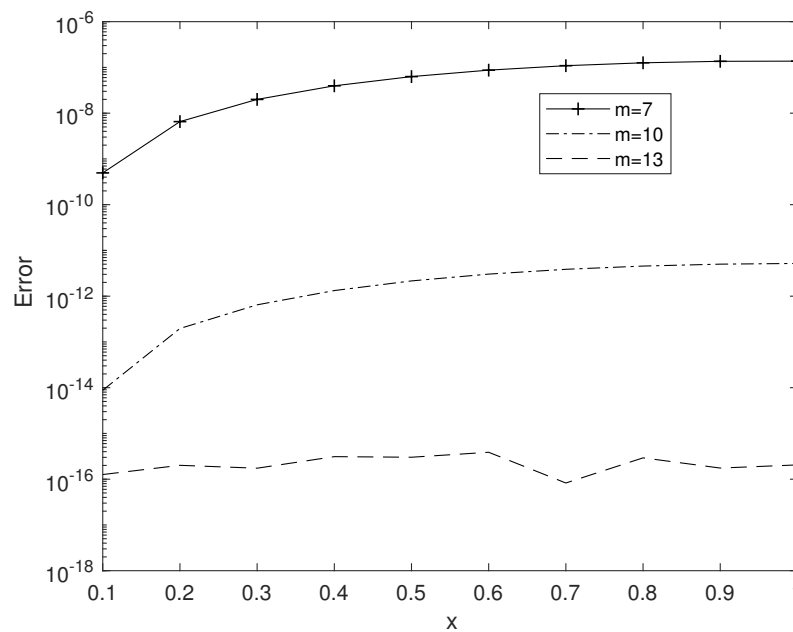


Figure 4. Approximation errors ($h = 0.1$) for the matrix problem (34) with distinct spline orders.

4.7. A Nonlinear Matrix Problem

In the next example, we focus on the following matrix problem with a nonlinear term:

$$\left. \begin{aligned} Y^{(3)} &= Y^2 \\ Y(0) &= 0_n \\ Y'(0) &= 10^{-2}I_n \\ Y''(0) &= 10^{-2}1_n \end{aligned} \right\}, x \in [0, b], \tag{35}$$

where 0_n , I_n , and 1_n are the null, the identity, and the all-ones matrices of dimension n , respectively.

To compute the “exact” solution, we employed the MATLAB Symbolic Math Toolbox (with 256 digits of precision) using the vpa (variable-precision floating-point arithmetic) function. All computations have IEEE double-precision arithmetic with round off $u = 2^{-53} \approx 1.11 \times 10^{-16}$. The “exact” solution is obtained whenever two consecutive spline orders (for fixed step size) present a relative error lower than the unit round off for this precision. For $b = 10$ and $n = 10$, we found that for $h = 0.01$ and for the spline orders 10 and 11, the relative error when the “exact” solution is calculated is equal to $1.2019 \times 10^{-17} < u$.

Table 11 shows the relative errors at $x = 2, 4, 6, 8$, and 10 , with a spline order $m = 12$, considering the step sizes $h = 0.1, h = 0.05$, and $h = 0.01$. As we can see, in general, the error committed at any point x becomes smaller and smaller for decreasing h values. Nevertheless, for $h = 0.01$, the results obtained are worse than those ones for $h = 0.05$, due to a larger number of floating-point operations and reaching the limit of the machine accuracy.

Table 11. Relative errors for the nonlinear matrix problem (35) solved by a spline of order $m = 12$.

x	2	4	6	8	10
$h = 0.5$	1.46×10^{-14}	4.32×10^{-13}	8.68×10^{-12}	4.77×10^{-10}	4.24×10^{-07}
$h = 0.1$	6.10×10^{-16}	8.35×10^{-16}	1.47×10^{-15}	3.37×10^{-15}	2.48×10^{-13}
$h = 0.05$	1.05×10^{-15}	6.27×10^{-16}	1.23×10^{-15}	2.27×10^{-15}	4.58×10^{-15}
$h = 0.01$	1.58×10^{-15}	2.48×10^{-15}	3.16×10^{-15}	8.77×10^{-15}	2.11×10^{-14}

Figure 5 plots the relative errors at $x = 10$ for distinct step sizes when varying the spline order. The relative errors decrease as we use smaller values of h and as the order of the spline m increases. When $h = 0.05$, the error can even be larger if we use a very high-degree spline versus lower degrees. Moreover, for $h = 0.01$, using a value of m greater than or equal to nine does not reduce the relative error committed. As explained for Table 11, this is because of the round off errors that appear when computing a large number of floating-point operations.

Additionally, Figure 6 displays the execution time required for different degrees of the spline and for distinct values of h . As expected, a higher value of h and m leads to an increase in the computational cost and, consequently, in the execution time. For the same value of h , it is possible that the execution time will not always rise with the degree of the spline. This is because the number of iterations required by the fixed-point method is not always identical.

Finally, Figure 7 depicts the response time versus the relative error committed, also with different step sizes. Clearly, higher values of h result in larger execution times. On the other hand, for a constant step size, a smaller relative error leads to a higher execution time, since this is achieved by working with higher spline orders.

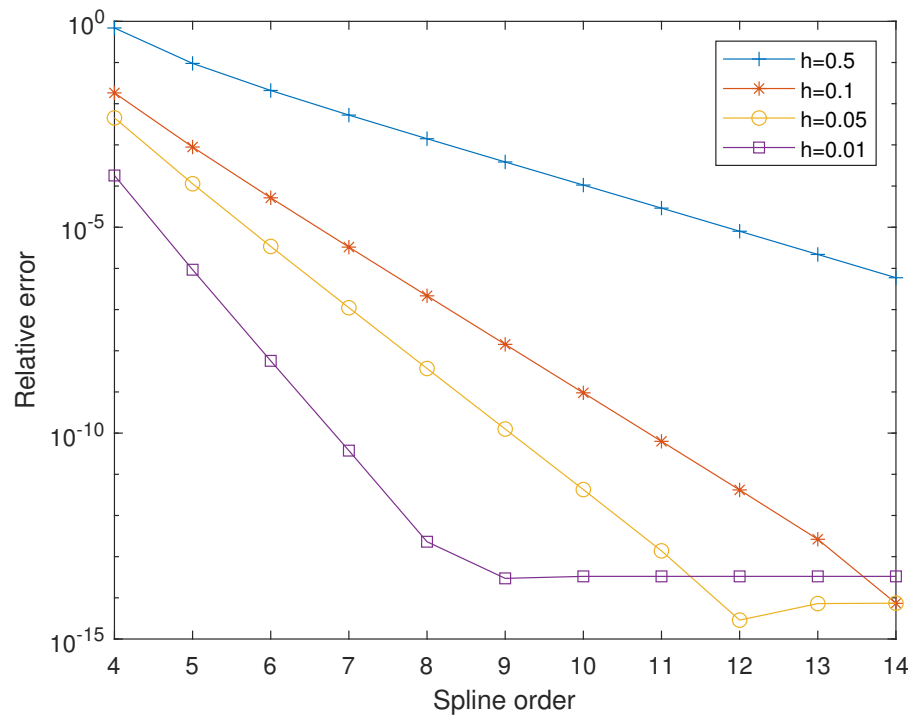


Figure 5. Relative errors, for the nonlinear matrix problem (35), depending on the spline order.

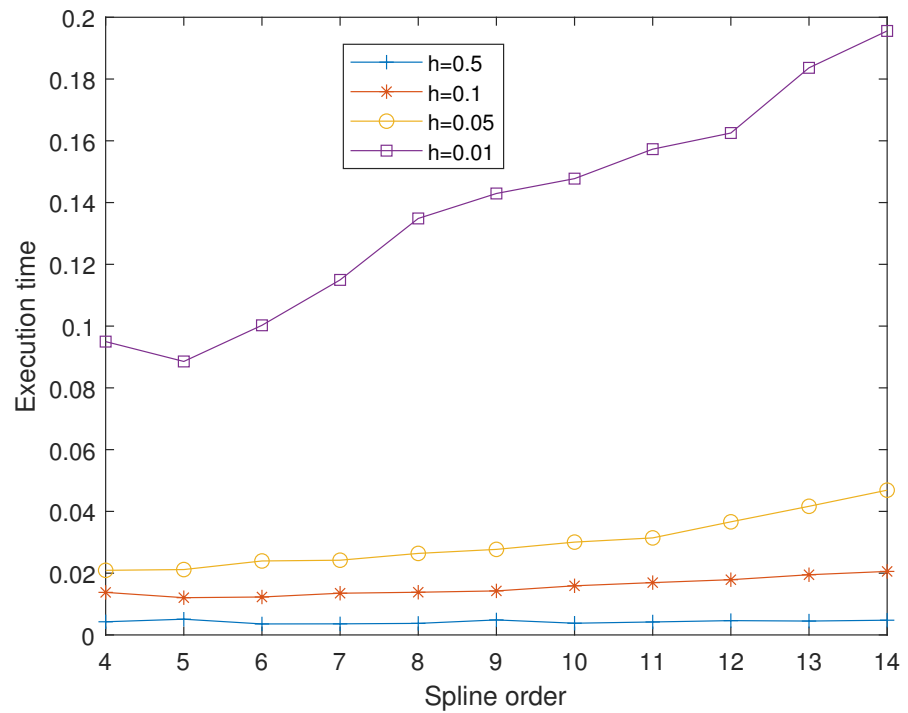


Figure 6. Execution time, for the nonlinear matrix problem (35), depending on the spline order.

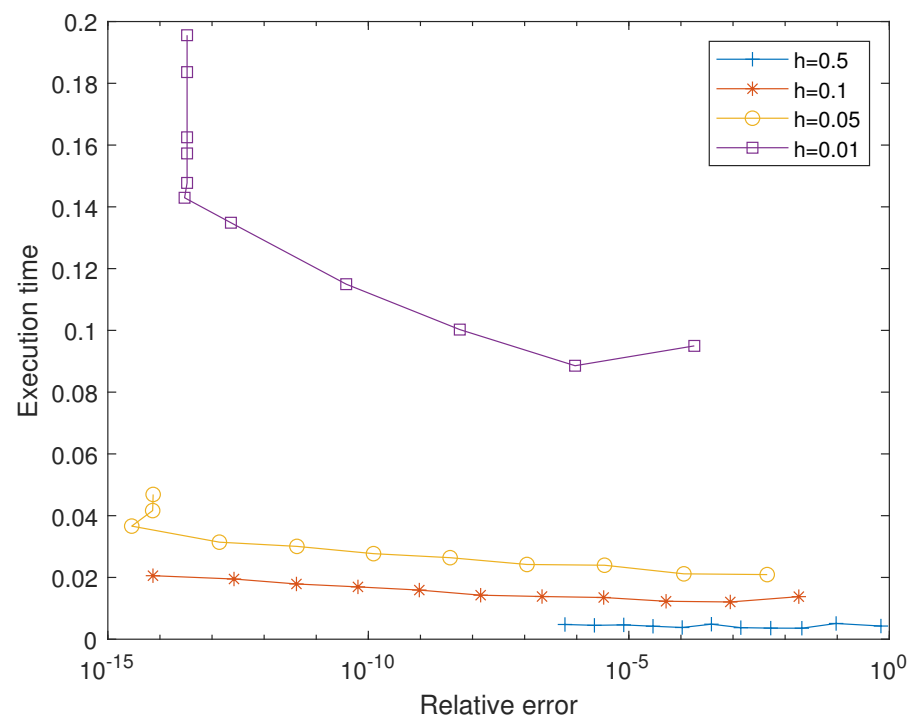


Figure 7. Execution time, for the nonlinear matrix problem (35), versus relative error.

5. Conclusions

This work outlined a numerical method for solving a specific type of third-order matrix differential equations and presents the corresponding algorithmic recipes. For illustration purposes, several standard test problems were included. All results demonstrated that our procedure provides solutions with very acceptable accuracy. Of course, the degree of accuracy can always be readjusted by decreasing the step size and/or increasing the order of the approximation spline accordingly.

Author Contributions: Conceptualization, E.D., J.I., M.M.T., T.R.-H. and J.M.A.; methodology, E.D., J.I., M.M.T., T.R.-H. and J.M.A.; software, E.D., J.I., M.M.T., T.R.-H. and J.M.A.; validation, E.D., J.I., M.M.T., T.R.-H. and J.M.A.; formal analysis, E.D., J.I., M.M.T., T.R.-H. and J.M.A.; investigation, E.D., J.I., M.M.T., T.R.-H. and J.M.A.; writing—original E.D., J.I., M.M.T., T.R.-H. and J.M.A.; writing—review and editing, E.D., J.I., M.M.T., T.R.-H. and J.M.A. All authors read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the European Regional Development Fund (ERDF) and the Spanish Ministerio de Economía y Competitividad Grant TIN2017-89314-P.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank the reviewers for their effort and time spent reviewing this paper, for their comments, suggestions, and their helpful feedback.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	Linear dichroism

References

1. Duffy, B.; Wilson, S. A third-order differential equation arising in thin-film flows and relevant to Tanner's law. *Appl. Math. Lett.* **1997**, *10*, 63–68. [[CrossRef](#)]
2. Momoniat, E. Symmetries, first integrals and phase planes of a third-order ordinary differential equation from thin film flow. *Math. Comput. Model.* **2009**, *49*, 215–225. [[CrossRef](#)]
3. Yap, L.K.; Ismail, F.; Senu, N. An Accurate Block Hybrid Collocation Method for Third Order Ordinary Differential Equations. *J. Appl. Math.* **2014**, *2014*, 549597. [[CrossRef](#)]
4. Abdel Kader, A.; Abdel Latif, M.; Nour, H. Exact solutions of a third-order ODE from thin film flow using λ -symmetry method. *Int. J. Non-Linear Mech.* **2013**, *55*, 147–152. [[CrossRef](#)]
5. Lee, K.C.; Senu, N.; Ahmadian, A.; Ibrahim, S.N.I. On Two-Derivative Runge-Kutta Type Methods for Solving $u''' = f(x, u(x))$ with Application to Thin Film Flow Problem. *Symmetry* **2020**, *12*, 924. [[CrossRef](#)]
6. Butusov, D.; Tutueva, A.; Fedoseev, P.; Terentev, A.; Karimov, A. Semi-Implicit Multistep Extrapolation ODE Solvers. *Mathematics* **2020**, *8*, 943. [[CrossRef](#)]
7. Momoniat, E.; Mahomed, F. Symmetry Reduction and Numerical Solution of a Third-Order ODE from Thin Film Flow. *Math. Comput. Appl.* **2010**, *15*, 709–719. [[CrossRef](#)]
8. You, X.; Chen, Z. Direct integrators of Runge–Kutta type for special third-order ordinary differential equations. *Appl. Numer. Math.* **2013**, *74*, 128–150. [[CrossRef](#)]
9. Famelis, I.T.; Tsitouras, C. Symbolic derivation of Runge–Kutta–Nyström type orderconditions and methods for solving $y''' = f(x, y)$. *Appl. Math. Comput.* **2017**, *297*, 50–60.
10. Senu, N.; Mechee, M.; Ismail, F.; Siri, Z. Embedded explicit Runge–Kutta type methods for directly solving special third order differential equations $y''' = f(x, y)$. *Appl. Math. Comput.* **2014**, *240*, 281–293. [[CrossRef](#)]
11. Defez, E.; Tung, M.M.; Ibáñez, J.; Sastre, J. Approximating and computing nonlinear matrix differential models. *Math. Comput. Model.* **2012**, *55*, 2012–2022. [[CrossRef](#)]
12. Defez, E.; Tung, M.M.; Solis, F.J.; Ibáñez, J. Numerical approximations of second-order matrix differential equations using higher degree splines. *Linear Multilinear Algebra* **2015**, *66*, 472–489. [[CrossRef](#)]
13. Graham, A. *Kronecker Products and Matrix Calculus with Applications*; Halsted Press: New York, NY, USA, 1981.
14. Flett, T.M. *Differential Analysis: Differentiation, Differential Equations and Differential Inequalities*; Cambridge University Press: Cambridge, UK, 1980.
15. Loscalzo, F.R.; Talbot, T.D. Spline function approximations for solutions of ordinary differential equations. *SIAM J. Numer. Anal.* **1967**, *4*, 433–445. [[CrossRef](#)]
16. Khataybeh, S.N.; Hashim, I.; Alshbool, M. Solving directly third-order ODEs using operational matrices of Bernstein polynomials method with applications to fluid flow equations. *J. King Saud Univ.-Sci.* **2019**, *31*, 822–826. [[CrossRef](#)]
17. Mechee, M.; Senu, N.; Ismail, F.; Nikouravan, B.; Siri, Z. A Three-Stage Fifth-Order Runge–Kutta Method for Directly Solving Special Third-Order Differential Equation with Application to Thin Film Flow Problem. *Math. Probl. Eng.* **2013**, *2013*, 795397. [[CrossRef](#)]
18. Defez, E.; Tung, M.M.; Ibáñez, J.; Sastre, J. Approximating a class of linear third-order ordinary differential problems. In *Progress in Industrial Mathematics at ECMI 2018*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 329–335.