Numerical Solution of Matrix Differential Models using Cubic Matrix Splines

E. DEFEZ*, L. SOLER, A. HERVAS AND C. SANTAMARÍA
Instituto de Matemáctica Multidisciplinar
Universidad Politécnica de Valencia, P.O. Box 22.012, Valencia, Spain
edefez|ahervas|crissana@imm.upv.es

(Received and accepted April 2005)

Abstract—This paper deals with the construction of approximate solution of first-order matrix linear differential equations using matrix cubic splines. An estimation of the approximation error, an algorithm for its implementation and some illustrative examples are included. © 2005 Elsevier Ltd. All rights reserved.

Keywords—Matrix linear differential equations, Cubic spline.

1. INTRODUCTION

Matrix differential equations appear frequently in a wide variety of models in physics and engineering [1-4]. Apart from the problems where the mathematical pattern is written in matrix form, they also appear when special techniques to solve scalar or vectorial problems are used. Examples of such situations are the embedding methods for the study of linear boundary value problems [5], shooting methods to solve scalar or vectorial problems with boundary values conditions [6], lines method for the numerical integration of partial differential equations [7], or homotopic methods to solve nonlinear systems equations [8].

The vectorization techniques to transform a matrix problem in a set of scalar equations or vectorial independent has several drawbacks, [9]. First, the physical sense of the magnitudes is lost with vectorization techniques. Secondly, the computational cost increases. Finally, the vectorization techniques waste the advantages of those symbolic languages adapted to matrix expressions.

This work we will develop a method for the numerical integration of the first-order matrix differential linear equation given by

\[ Y'(x) = A(x)Y(x) + B(x), \quad a \leq x \leq b, \]
\[ Y(a) = Y_a, \quad (1) \]

This work has been partially supported by the Spanish MCYT and FEDER Grant DPI2004-08383-C03-03 and The Generalitat Valenciana.

*Author to whom all correspondence should be addressed.

0898-1221/05/$ - see front matter © 2005 Elsevier Ltd. All rights reserved. Typeset by A&M-TEX
where \( Y, Y \in \mathbb{C}^{r \times q} \), \( A : [a, b] \rightarrow \mathbb{C}^{r \times r}, B : [a, b] \rightarrow \mathbb{C}^{r \times q} \) verifies \( A, B \in C^1([a, b]) \), it guarantees the existence of only one solution \( Y(x) \) of (1), which is continuously differentiable, [10, p. 99].

Problem (1) appears in the mathematical modelling of many different technological applications, see [11] and also dealing with nonlinear problems such as Riccati equations [12–14] after using some linealization technique. Numerical methods for the calculation of approximate solutions of problems of the type (1) by means of linear multistep methods with constant step have been studied in [15]. Although for these methods exist a priori error bound given in function of the problem data, this error bound is given in terms of an exponential which depends on the integration step \( h \), forcing in the practice to take a value of \( h \) too small. Therefore, these methods require some interpolation techniques in order to get a continuous solution, [15]. Other methods, based on the developments of Magnus or Fer, [16], require the calculation of the exponential of some matrices involving high computational cost.

In the scalar case, the cubic splines were used in [17] for the resolution of ordinary differential equations, obtaining approaches that, among other advantages, they were of \( C^1 \) class in the interval \([a, b]\), easily valuates and with an approach error \( O(h^4) \). Recently, the splines have been used in the resolution of other scalar problems, [18]. In [19] have been developed an implicit spline method by means of Hermite interpolation techniques for vector problems.

In this paper, we propose a method using cubic matrix splines for the numerical approximation to the solution of (1). The present work extends this important advantages obtained in [17] for the scalar case to the matrix framework.

This paper is organized as follows. In Section 2, we develop the proposed method including the study of the approximation error and a constructive algorithm. Finally, some examples will be presented in Section 3.

Along this work we will denote by \( \mathbb{C}^{r \times q} \) the set of the rectangular \( p \times q \) complex matrices. If \( A \in \mathbb{C}^{r \times s} \), we will denote for \( \|A\| \) their 2-norm, defined by

\[
\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|},
\]

where for a vector \( x \) in \( \mathbb{C}^r \), \( \|x\| = (x^*x)^{1/2} \) is the usual Euclidean norm of \( x \). By [20, p. 56], it follows that

\[
\max_{ij} |a_{ij}| \leq \|A\| \leq \sqrt{sr} \max_{ij} |a_{ij}|. \tag{2}
\]

We will denote by \( P_n[x] \) the set of matrix polynomials with degree \( n \) and real variable \( x \). We will say that one matrix function \( g : [a, b] \rightarrow \mathbb{C}^{r \times q} \) is of class \( k \) \( 0 \geq k \) \( \geq 0 \), and we will represent it \( g \in C^k([a, b]) \), if \( g \) is \( k \) times differentiable, and its \( k \)th derivative is continuous in \([a, b]\). Let be \([a, b] \subseteq \mathbb{R} \) and be

\[
\Delta = \{a = x_0 < x_1 < \cdots < x_n = b\},
\]

a partition of \([a, b]\). Given \( m \) an integer bigger or equal to zero, we define the set of matrix splines of order \( m \) as

\[
M_{\mathbb{C}^{r \times q}}(\Delta)^{m-1}_{m-1} = \left\{ Q : [a, b] \rightarrow \mathbb{C}^{r \times q}, \left\{ Q_{|[x_{i-1}, x_i]}(x) \in P_m[x], \ i \in \{1, \ldots, n\}, \right\} \right\}.
\]

If \( m = 3 \) the matrix splines are called \textit{matrix cubic splines}, [21].

\section{Construction of the Approach}

Let us consider the partition of the interval \([a, b]\) given by

\[
\Delta_{[a, b]} = \{a = x_0 < x_1 < \cdots < x_n = b\}, \quad x_k = a + kh, \quad k = 0, 1, \ldots, n, \tag{3}
\]

where \( h = (b - a)/n \), being \( n \) a positive integer.
We will build in each subinterval \([a + k h, a + (k + 1) h]\) a matrix cubic spline approximating the solution of problem (1). For the first interval \([a, a + h]\), we consider that the spline is defined by

\[
S_{[a,a+h]}(x) = Y(a) + Y'(a)(x - a) + \frac{1}{2!}Y''(a)(x - a)^2 + \frac{1}{3!}\alpha_0(x - a)^3,
\]

where the matrix \(\alpha_0 \in C^{r \times q}\) is a parameter to be determined. As \(S_{[a,a+h]}(x)\) defined in (4) verifies:

\[
S_{[a,a+h]}(a) = Y(a), \quad S'_{[a,a+h]}(a) = Y'(a) = A(a)Y(a) + B(a),
\]

to obtain the spline we must determine \(Y''(a)\) and \(\alpha_0\). A simple calculation, using \(A, B \in C^1([a, b])\), shows that

\[
Y''(x) = (A'(x) + A^2(x)) Y(x) + A(x)S(x) + B'(x),
\]

for what we can evaluate \(Y''(a)\) using (5).

We have left the problem of determining \(\alpha_0\) when the spline is totally determined. For that, we will impose the spline be a solution of the problem (1) in the point \(x = a + h\),

\[
S'_{[a,a+h]}(a + h) = A(a + h) S_{[a,a+h]}(a + h) + B(a + h).
\]

From (6) we obtain the matrix equation with an only unknown matrix \(\alpha_0\):

\[
\left( I - \frac{h}{3}A(a + h) \right) \alpha_0 = \frac{2}{h^2} \left[ A(a + h) \left( Y(a) + Y'(a)h + \frac{1}{2!}Y''(a)h^2 \right) + B(a + h) - Y'(a) - Y''(a)h \right].
\]

Assuming that matrix equation (7) has one solution only \(\alpha_0\), this technique determines the spline in the interval \([a, a + h]\).

In the interval \([a + h, a + 2h]\), the matrix cubic spline takes the form

\[
S_{[a+h,a+2h]}(x) = S_{[a,a+h]}(a + h) + S'_{[a,a+h]}(a + h)(x - (a + h)) + \frac{1}{2!}S''_{[a,a+h]}(a + h)(x - (a + h))^2 + \frac{1}{3!}\alpha_1(x - (a + h))^3,
\]

so \(S(x)\) defined on \([a, a + h] \cup [a + h, a + 2h]\) it is of class \(C^2([a, a + 2h])\), and all the coefficients of the spline \(S_{[a+h,a+2h]}(x)\) are determined with the exception of \(\alpha_1 \in C^{r \times q}\). It is easy to check that the spline (8) satisfies the differential equation (1) in \(x = a + h\), for what we will determine \(\alpha_1\) by imposing that (8) be also a solution of (1) in \(x = a + 2h\),

\[
S'_{[a+h,a+2h]}(a + 2h) = A(a + 2h) S_{[a+h,a+2h]}(a + 2h) + B(a + 2h).
\]

From (9) we obtain the matrix equation with an only matrix unknown \(\alpha_1\),

\[
\left( I - A(a + 2h) \frac{h}{3} \right) \alpha_1 = \frac{2}{h^2} \left[ A(a + 2h) \left( S_{[a,a+h]}(a + h) + S'_{[a,a+h]}(a + h)h + \frac{1}{2} S''_{[a,a+h]}(a + h)h^2 \right) + B(a + 2h) - S'_{[a,a+h]}(a + h) - S''_{[a,a+h]}(a + h)h \right].
\]

Assuming that matrix equation (10) has one solution only \(\alpha_1\), in this way the spline is totally determined in the interval \([a + h, a + 2h]\).
Iterating this process, let us consider the matrix cubic spline constructed until the subinterval \([a + (k - 1)h, a + kh]\) and we define it in the next subinterval \([a + kh, a + (k + 1)h]\) as

\[
S|_{[a+kh,a+(k+1)h]}(x) = \beta_k(x) + \frac{1}{3!} \alpha_k (x - (a + kh))^3, \tag{11}
\]

where

\[
\beta_k(x) = \sum_{k=0}^{2} \frac{1}{k!} S^{(k)}|_{[a+(k-1)h,a+kh]} (a + kh)(x - (a + kh))^k. \tag{12}
\]

Defined so, the matrix cubic spline \(S(x) \in C^2([a+kh,a+(k+1)h])\), and it is easy to check that it verifies the differential equation (1) in the point \(x = a + kh\). We determine \(\alpha_k\) by imposing that differential equation (1) is satisfied at the point \(x = a + (k + 1)h\)

\[
S'|_{[a+kh,a+(k+1)h]}(a + (k + 1)h) = A(a + (k + 1)h) S|_{[a+kh,a+(k+1)h]}(a + (k + 1)h) + B(a + (k + 1)h),
\]

that developing takes us to the matrix equation

\[
\left(I - \frac{h}{3} A(a + (k + 1)h) \right) \alpha_k = \frac{2}{h^3} \left[ A(a + (k + 1)h) \beta_k(a + (k + 1)h) + B(a + (k + 1)h) - \beta_k'(a + (k + 1)h) \right]. \tag{13}
\]

Note that solubility of equation (13) is guaranteed showing that the matrix \((I - (h/3)A(a + (k + 1)h))\) is invertible, for \(k = 0, 1, \ldots, n - 1\). Let us denote

\[
M = \max \{ \|A(x)\| ; a \leq x \leq b \}, \tag{14}
\]

and then

\[
\left\| I - \left(I - \frac{h}{3} A(a + (k + 1)h) \right)^n \right\| = \frac{h}{3} \|A(a + (k + 1)h)\| \leq \frac{h}{3} M, \tag{15}
\]

thus taking \(h \leq 3/M\), one has \(\| I - (I - (h/3)A(a + (k + 1)h))\| \leq 1\), which guarantees, for the perturbation lemma, [20, p. 58], that matrix \((I - (h/3)A(a + (k + 1)h))\) is invertible, and therefore, the equation (13) has unique solution \(\alpha_k\), for \(k = 0, 1, \ldots, n - 1\). Taking into account [17, Theorem 5] and (2), the following result has been established.

**Theorem 2.1.**

(i) Assume that \(A, B \in C^1([a, b])\) and let \(h > 0\) so that \(h \leq 3/M\) where \(M\) is given by (14), then the matrix cubic spline \(S(x)\) defined in each subinterval \([a + kh, a + (k + 1)h]\), \(k = 0, 1, \ldots, n - 1\) for (11) is well defined and it is only defined by the previous construction.

(ii) If \(A, B \in C^3([a, b])\) then \(\|S(x) - Y(x)\| = O(h^4), \forall x \in [a, b]\).

The following algorithm allows us to the calculation of the approximate solution of (1) by means of matrix cubic splines in the interval \([a, b]\).

**Algorithm.**

- Step 1. Determine the constants \(M\) and \(Y''(a)\) given by (14) and (5), respectively. Take \(n > M(b - a)/3\) and \(h = (b - a)/n\) and consider the partition \(\Delta_{[a,b]}\) given by (3).
- Step 2. For \(k = 0\), solve the matrix equation (7) and compute \(S|_{[a,a+h]}(x)\) defined by (4).
- Step 3. For \(k=1, \ldots, n-1\), solve the matrix equation (13) and compute \(S|_{[a+kh,a+(k+1)h]}(x)\) defined by (11).
3. EXAMPLES

In this section, we test the algorithm in a situation where the exact solution is known.

**Example 3.1.** Let us consider the problem

\[ Y'(x) = \frac{1}{x^3 - x - 1} \left[ \begin{array}{c} 2x^2 - 1 \\ -x - 1 \\ x^3 + x^2 - x - 1 \end{array} \right] Y(x), \quad 0 \leq x \leq 1, \] (16)

\[ Y(0) = \left[ \begin{array}{c} 1 \\ 0 \end{array} \right], \quad Y(x) \in C^2, \]

This problem has the exact solution \( Y(x) = \left[ \begin{array}{c} e^x \\ xe^x \end{array} \right] \), so we will be able to calculate the approximation error. Taking derivatives it follows that \( Y''(x) = (A'(x) + [A(x)]^2)Y(x) \), \( Y'(0) = A(0)Y(0) = \left( \begin{array}{c} 1 \\ 0 \end{array} \right) \), and \( Y''(0) = \left( \begin{array}{c} 1 \\ 2 \end{array} \right) \). As \( \max_{x \in [0,1]} \|A(x)\| \leq 3 \), we take \( M = 3 \). For \( n = 10 \), one satisfies \( n > M(b - a)/3 \), so we take \( h = (b - a)/n = 0.1 \).

The results, obtained with MATHEMATICA, V.4.0, are shown in Table 1.

In Table 1 the values in the error column are the maximum of the 2-norm in each subinterval. Note that the results are of order \( O(h^n) \) for \( h = 0.1 \). This means that bounds in Theorem 2.1 are quite conservative and that in practice results are better.

<table>
<thead>
<tr>
<th>([a_i, a_{i+1}])</th>
<th>Approximation</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>([0, 0.1])</td>
<td>(\frac{1}{x + 0.5x^2 + 0.172635x^3}{x + x^2 + 0.523192x^3})</td>
<td>(6.33721 \times 10^{-6})</td>
</tr>
<tr>
<td>([0.1, 0.2])</td>
<td>(\left( \begin{array}{c} 0.999977 + 1.00069x + 0.493104x^2 + 0.19562x^3 \ -0.0000977783 + 1.0029x + 0.970967x^2 + 0.519972x^3 \end{array} \right))</td>
<td>(6.05558 \times 10^{-6})</td>
</tr>
<tr>
<td>([0.2, 0.3])</td>
<td>(\left( \begin{array}{c} 0.999844 + 1.02368x + 0.481096x^2 + 0.212397x^3 \ -0.0000178165 + 1.01771x + 0.921992x^2 + 0.658015x^3 \end{array} \right))</td>
<td>(8.14262 \times 10^{-6})</td>
</tr>
<tr>
<td>([0.3, 0.4])</td>
<td>(\left( \begin{array}{c} 0.99916 + 1.03052x + 0.45713x^2 + 0.237618x^3 \ -0.000376158 + 1.04471x + 0.827124x^2 + 0.560748x^3 \end{array} \right))</td>
<td>(7.81749 \times 10^{-6})</td>
</tr>
<tr>
<td>([0.4, 0.5])</td>
<td>(\left( \begin{array}{c} 0.997869 + 1.05085x + 0.43312x^2 + 0.261728x^3 \ -0.000454299 + 1.08415x + 0.714712x^2 + 0.547968x^3 \end{array} \right))</td>
<td>(11.5296 \times 10^{-6})</td>
</tr>
<tr>
<td>([0.5, 0.6])</td>
<td>(\left( \begin{array}{c} 0.994113 + 1.07421x + 0.389685x^2 + 0.289154x^3 \ -0.00273102 + 1.19212x + 0.505214x^2 + 1.03395x^3 \end{array} \right))</td>
<td>(11.6396 \times 10^{-6})</td>
</tr>
<tr>
<td>([0.6, 0.7])</td>
<td>(\left( \begin{array}{c} 0.987537 + 1.09439x + 0.358899x^2 + 0.319136x^3 \ -0.0536005 + 1.53235x + 0.28642x^2 + 1.15511x^3 \end{array} \right))</td>
<td>(16.357 \times 10^{-6})</td>
</tr>
<tr>
<td>([0.7, 0.8])</td>
<td>(\left( \begin{array}{c} 0.976271 + 1.1231x + 0.26611x^2 + 0.352273x^3 \ -0.113737 + 1.5813x - 0.0817546x^2 + 1.33099x^3 \end{array} \right))</td>
<td>(17.359 \times 10^{-6})</td>
</tr>
<tr>
<td>([0.8, 0.9])</td>
<td>(\left( \begin{array}{c} 0.9566 + 1.1867x + 0.174196x^2 + 0.39065x^3 \ -0.1604 + 1.86993x - 0.467515x^2 + 1.49714x^3 \end{array} \right))</td>
<td>(23.29 \times 10^{-6})</td>
</tr>
<tr>
<td>([0.9, 1])</td>
<td>(\left( \begin{array}{c} 0.928372 + 1.29095x + 0.0695751x^2 + 0.429403x^3 \ -0.354987 + 2.41976x - 1.05624x^2 + 1.70977x^3 \end{array} \right))</td>
<td>(24.6909 \times 10^{-6})</td>
</tr>
</tbody>
</table>
**Example 3.2.** Consider the matrix problem

\[
Y'(x) = A(x)Y(x) - B(x),
\]

\[
Y(0) = \begin{pmatrix} 1 & 1 \\ 0 & -1 \\ 0 & 0 \end{pmatrix}, \quad x \in [0, 1],
\]

with

\[
A(x) = \begin{pmatrix} -1 - x & 0 & 1 + e^x + x \\ e^x & -x & 1 \\ 0 & -1 & e^x \end{pmatrix},
\]

\[
B(x) = \begin{pmatrix} -2 + (-3 + e^x)x & -1 - x - x^2 - e^x(2 + x) \\ x + e^x(1 + x) & e^x + xe^x - (5 + x)(1 + x^2) \\ -1 + xe^x & 1 - x(5 + x) \end{pmatrix},
\]

with an exact solution given by

\[
Y(x) = \begin{pmatrix} 1 + x & e^x + x \\ 0 & -1 + 5x + x^2 \\ x & 0 \end{pmatrix},
\]

so we will be able to calculate the approximation error.

<table>
<thead>
<tr>
<th>[x_i, x_{i+1}]</th>
<th>Approximation</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0, 0.1]</td>
<td>\begin{pmatrix} 1 + 1.x &amp; 1.2 + 0.65x + 0.17543x^3 \ 0 &amp; -1.3 + 5.2x + 1.35412x^2 + 5.15023x^3 \times 10^{-6} \ x &amp; -1.77022 \end{pmatrix}</td>
<td>1.39565 \times 10^{-6}</td>
</tr>
<tr>
<td>[0.1, 0.2]</td>
<td>\begin{pmatrix} 1.1 + 1.x &amp; 0.999997 + 2.00073x + 0.492747x^2 + 0.19649x^3 \ 0 &amp; -1.1 + 4.99999x + 1.00006x^2 + 0.150158x^3 \times 10^{-6} \ 1.x &amp; 10^{-6} \end{pmatrix} \begin{pmatrix} 0.00154 - 0.0464x + 0.0464x^2 - 1.92505x^3 \end{pmatrix}</td>
<td>1.39565 \times 10^{-6}</td>
</tr>
<tr>
<td>[0.2, 0.3]</td>
<td>\begin{pmatrix} 1 + 1.x &amp; 0.999996 + 2.00345x + 0.484122x^2 + 0.210864x^3 \ 0 &amp; -1.1 + 5.00005x + 0.999799x^2 + 0.000281181x^3 \times 10^{-6} \ 1.x &amp; 10^{-6} \end{pmatrix} \begin{pmatrix} -0.079 + 1.195x - 5.95x^2 + 8.07x^3 \end{pmatrix}</td>
<td>1.43614 \times 10^{-6}</td>
</tr>
<tr>
<td>[0.3, 0.4]</td>
<td>\begin{pmatrix} 1 + 1.x &amp; 0.999995 + 2.01021x + 0.458261x^2 + 0.239599x^3 \ 0 &amp; -1.00005 + 5.00342x + 0.999213x^2 + 0.00059348x^3 \times 10^{-6} \ 1.x &amp; 10^{-6} \end{pmatrix} \begin{pmatrix} -2.737 + 19.260x + 44.889x^2 + 33.603x^3 \end{pmatrix}</td>
<td>1.49774 \times 10^{-6}</td>
</tr>
<tr>
<td>[0.4, 0.5]</td>
<td>\begin{pmatrix} 1 + 1.x &amp; 0.999997 + 2.04476x + 0.348763x^2 + 0.202665x^3 \ 0 &amp; -0.999984 + 4.99986x + 1.00044x + 0.425818x^3 \times 10^{-6} \ 1.x &amp; 10^{-6} \end{pmatrix} \begin{pmatrix} -8.356 - 47.305x + 250.027x^2 - 115.19.7x^3 \end{pmatrix}</td>
<td>1.49774 \times 10^{-6}</td>
</tr>
<tr>
<td>[0.5, 0.6]</td>
<td>\begin{pmatrix} 1 + 1.x &amp; 0.999998 + 2.09976x + 0.343902x^2 + 0.154173x^3 \ 0 &amp; -1.000027 + 5.000127x + 0.998018x^2 + 0.00102612x^3 \times 10^{-6} \ 1.x &amp; 10^{-6} \end{pmatrix} \begin{pmatrix} -20.469 + 96.823x - 15.197x^2 + 73.300x^3 \end{pmatrix}</td>
<td>1.57205 \times 10^{-6}</td>
</tr>
<tr>
<td>[0.6, 0.7]</td>
<td>\begin{pmatrix} 1 + 1.x &amp; 0.999999 + 2.13032x + 0.256812x^2 + 0.356613x^3 \ 0 &amp; -0.999948 + 4.99784x + 1.00222x - 0.0013831x^3 \times 10^{-6} \ 1.x &amp; 10^{-6} \end{pmatrix} \begin{pmatrix} 45.190 - 18.4576x + 250.027x^2 - 113.127x^3 \end{pmatrix}</td>
<td>1.57205 \times 10^{-6}</td>
</tr>
<tr>
<td>[0.7, 0.8]</td>
<td>\begin{pmatrix} 1 + 1.x &amp; 0.999999 + 2.18597x + 0.0530041x^2 + 0.435231x^3 \ 0 &amp; -0.999829 + 4.99454x + 1.00058x^2 - 0.00204807x^3 \times 10^{-6} \ 1.x &amp; 10^{-6} \end{pmatrix} \begin{pmatrix} 164.691 - 527.578x + 561.976x^2 - 199.845x^3 \end{pmatrix}</td>
<td>1.65985 \times 10^{-6}</td>
</tr>
<tr>
<td>[0.8, 0.9]</td>
<td>\begin{pmatrix} 1 + 1.x &amp; 0.923466 + 2.30668x + 0.0300412x + 0.435231x^3 \ 0 &amp; -0.998298 + 4.99454x + 1.00058x^2 - 0.00204807x^3 \times 10^{-6} \ 1.x &amp; 10^{-6} \end{pmatrix} \begin{pmatrix} 164.691 - 527.578x + 561.976x^2 - 199.845x^3 \end{pmatrix}</td>
<td>1.65985 \times 10^{-6}</td>
</tr>
<tr>
<td>[0.9, 1]</td>
<td>\begin{pmatrix} 1 + 1.x &amp; 0.923466 + 2.30668x + 0.0300412x + 0.435231x^3 \ 0 &amp; -0.998298 + 4.99454x + 1.00058x^2 - 0.00204807x^3 \times 10^{-6} \ 1.x &amp; 10^{-6} \end{pmatrix} \begin{pmatrix} 164.691 - 527.578x + 561.976x^2 - 199.845x^3 \end{pmatrix}</td>
<td>1.65985 \times 10^{-6}</td>
</tr>
</tbody>
</table>
As \( \max_{x \in [0,1]} \| A(x) \| \leq 5 \), we take \( M = 5 \). Taking derivatives it follows that

\[
Y'(x) = \left[ A'(x) + A^2(x) \right] Y(x) - A(x)B(x) - B'(x),
\]

and by (18),

\[
Y''(0) = A(0)Y(0) - B(0) = 0.
\]

For \( n = 10 \) it is verified \( n > (b - a)/3 \), so we take \( h = (b - a)/n = 0.1 \). The values obtained with Mathematica V.4.0 are shown in Table 2. The values in the error column are obtained as in Example 3.1. Note that results are of order \( O(h^6) \) for \( h = 0.1 \).

REFERENCES