

DEGENERATE MATRICES METHODS BY SPLINES FOR BOUNDARY VALUES PROBLEMS OF ORDINARY DIFFERENTIAL EQUATIONS

M. BELOVS and T. CĪRULIS

Institute of Mathematics of Latvian Academy of Sciences and University of Latvia

Akadēmijas laukums 1, Rīga LV-1524, Latvia

E-mail: belovs@lanet.lv; cirulis@lanet.lv

Received September 19, 2004; revised April 12, 2005

Abstract. A method for numerical solving of boundary values problems of ordinary differential equations based on the use of splines and differentiation matrices with nodes as zeroes of classical orthogonal polynomials is considered. Possibilities of the method are shown by means of different examples. The method essentially uses the results obtained by Degenerate Matrices methods and it is applied for solving initial values problems of ordinary differential equations.

Key words: boundary values problems of ODE, splines, Degenerate Matrix method, differentiation matrix, zeroes of classical orthogonal polynomials

1. Introduction

Boundary values problems of ordinary differential equations (BVP of ODE) mean a very great interest both for their direct applications in physics and for solving more complicated problems in Mathematical Physics. Such problems are discussed in [1, 2, 6] where a very extensive list of references is given.

Degenerate Matrices (DM) methods for solving problems of mathematical physics are based on the use of approximation of functions and their derivatives of the Lagrange projectors with nodes chosen as zeroes of the classical orthogonal polynomials. These methods can be applied for BVP of ODE according to the following scheme.

1. We construct the equation and boundary conditions of the problem on chosen nodes.
2. All derivatives are replaced by differentiation matrices.
3. Values of the unknown function at the end points of the given interval are excluded from the obtained system of algebraic equations.

4. By solving the corresponding system of algebraic equations, values of the unknown function at interior points of the interval are calculated.

A principal deficiency of the method stated above consists in the necessity to choose a very large order of differentiation matrices if the solution of BVP of ODE is complicated, e.g. it rapidly oscillates or has boundary layers. Large order matrices lead to a great loss of the precision of calculations. In order to increase the accuracy for solution of BVP of ODE we suggest the method which combines the approximation by means of Lagrange projectors with the ones by splines. An analogous idea was applied to solve initial value problems of ODE in [3, 4, 5] and very effective algorithms were constructed, which depend on two independent parameters: the step h and the differentiation matrices order N . The accuracy of the solution can be increased in two ways, by reducing h and increasing N . The analysis of solutions in this case leads us to a conclusion that the splines should be constructed simultaneously with an unknown solution. Following this idea, we will use splines in the same way to construct the methods for solving BVP of ODE. Therefore, these splines have the species dependent on the concrete BVP.

Section 2 is devoted to general foundation of numerical algorithms. Examples of BVP for ODE are considered in Section 3 and BVP of ODE of higher degrees and systems are investigated in Section 4. Section 5 is devoted to a decomposition of BVP of ODE. All algorithms are constructed in the matrices forms. Such formulation is very convenient for their realization in computer codes.

2. Foundations of Numerical Algorithms

We will consider the following boundary value problems of ordinary differential equations

$$\mathbf{U}_n x \equiv \sum_{s=0}^n q_s(t) x^{(s)} = f(t), \quad t \in (a, b), \quad (2.1)$$

$$\begin{aligned} \phi_j(x(a), x'(a), \dots, x^{(m)}(a), x(b), x'(b), \dots, x^{(m)}(b)) = 0, \quad (2.2) \\ j = 1, 2, \dots, n; \quad m < n \end{aligned}$$

as illustration of our method for finding the unknown function $x(t)$. It is assumed that functions $q_s(t)$ and $f(t)$ are sufficiently smooth in the interval $t \in [a, b]$ and $q_n(t) \neq 0$. The boundary conditions (2.2) can be non-linear as well.

We are using splines to approximate the solution of BVP (2.1)–(2.2). The main interval is divided into smaller subintervals:

$$\begin{aligned} [a, b] &= [a_1, b_1] \cup [a_2, b_2] \cup \dots \cup [a_{k+1}, b_{k+1}], \\ a &= a_1 < b_1 = a_2 < \dots < b_k = a_{k+1} < b_{k+1} = b. \end{aligned}$$

Let $p(t)$ be a spline of order $N - 1$. A system of nodes on every interval $[a_i, b_i]$ is fixed (i.e. the template is given) $\Sigma_i = (t_{1,i}, t_{2,i}, \dots, t_{N,i})$:

$$a_i = t_{1,i} < t_{2,i} < \dots < t_{N,i} = b_i.$$

Since all nodes are selected, the spline $p(t)$ can be identified with a column vector X whose components are values of $p(t)$ on the nodes of the whole template $\Sigma = (\Sigma_1, \Sigma_2, \dots, \Sigma_{k+1})$. Thus, we define the vector $X = p(\Sigma) \in R^{(k+1)N}$. Now instead of the spline we use a vector X from the linear space $R^{(k+1)N}$. Usually nodes are chosen as zeroes of one of classical orthogonal polynomials. In such a way the nonsaturatness of numerical algorithms is ensured. The order of splines is high enough (several dozens). For simplicity of implementation in all numerical examples subintervals $[a_i, b_i]$ are chosen of the same length, but nodes on them are zeroes of Chebyshev polynomials of the first order. Certainly, it is possible to use subintervals of different length and to choose nodes as zeroes of general Jacobi polynomials.

We find derivatives of the spline using the block diagonal differentiation matrix

$$\Delta = \Delta^{(1)} \oplus \Delta^{(2)} \oplus \dots \oplus \Delta^{(k+1)},$$

where matrices $\Delta^{(i)}$ are primary differentiation matrices on corresponding intervals $[a_i, b_i]$ applied in DM methods for ODE. The process of construction of $\Delta^{(i)}$, its properties and applications are mentioned in [3]. Elements δ_{jk} , $j, k = 1, 2, \dots, N$ of $\Delta^{(i)}$ are computed in the following way. Let $q_N(\tau)$ be a polynomial with N different zeroes on $[-1, 1]$:

$$-1 = \tau_1 < \tau_2 < \dots < \tau_N = 1.$$

Then τ_j correspond to nodes $t_j = a_i + 0.5(b_i - a_i)(1 + \tau_j)$, $j = 1, 2, \dots, N$ for each interval $[a_i, b_i]$, and

$$\delta_{jk} = \begin{cases} H_i \frac{q'_N(\tau_j)}{(\tau_j - \tau_k)q'_N(\tau_k)}, & \text{if } j \neq k, \\ H_i \frac{q''_N(\tau_k)}{2q'_N(\tau_k)}, & \text{if } j = k, \end{cases} \quad H_i = \frac{2}{b_i - a_i}.$$

If $X = p(\Sigma)$ then $\Delta X = p'(\Sigma)$, where p' is a derivative of a spline. Higher order derivatives are defined analogically, i.e. $\Delta^2 X = p''(\Sigma)$ and so on.

Let L_j be a linear space of splines of the order $N - 1$, and the defect of splines be

$$\nu(j) = N - 1 - j, \quad -1 \leq j \leq N - 1.$$

It is known that $\dim L_j = \mu_j = (k+1)N - (j+1)k$. The number j characterises the smoothness of the spline. It must be greater than or equal to the order of ODE (2.1). In our constructions the space L_{-1} is isomorphic to $R^{(k+1)N}$. In case when $j > -1$, L_j is isomorphic to its subspace l_j of dimension μ_j . We can construct the basis of l_j by using the matrix $\Delta^{(i)}$. It is obvious that if $X \in R^{(k+1)N}$ then

$$X \in l_j \iff X = \mathbf{S}_j V_j,$$

where \mathbf{S}_j is $(k+1)N \times \mu_j$ matrix composed from a basis of l_j as its columns, but V_j is an arbitrary vector from R^{μ_j} . We will show how to build matrices \mathbf{S}_j . The sequence of subspaces l_j is a monotone one, that is:

$$R^{(k+1)N} = l_{-1} \supset l_0 \supset l_1 \supset \cdots \supset l_{N-1}.$$

It is obvious that $\mathbf{S}_{-1} = \mathbf{E}$, where \mathbf{E} is an identity matrix. We can choose \mathbf{S}_0 in the form:

$$\mathbf{S}_0 = (e_1, \dots, e_{N-1}, e_N + e_{N+1}, e_{N+2}, \dots, e_{2N-1}, \\ e_{2N} + e_{2N+1}, \dots, e_{kN} + e_{kN+1}, e_{kN+2}, \dots, e_{(k+1)N}),$$

where $\{e_j\}_1^{(k+1)N}$ are the vectors of canonical basis of $R^{(k+1)N}$.

Matrix \mathbf{S}_1 can be build in the following way. Let $\Delta_s, s = 1, 2, \dots, (k+1)N$ be rows of matrix for derivatives $\mathbf{\Delta} = \mathbf{\Delta}^{(1)} \oplus \mathbf{\Delta}^{(2)} \oplus \cdots \oplus \mathbf{\Delta}^{(k+1)}$. We define $k \times (k+1)N$ matrix \mathbf{A} with $\Delta_{iN} - \Delta_{iN+1}$ as it rows ($i = 1, \dots, k$). Rank of \mathbf{A} is equal to k . For each $X \in R^{(k+1)N}$

$$\mathbf{A}X = (x'_N - x'_{N+1}, x'_{2N} - x'_{2N+1}, \dots, x'_{kN} - x'_{kN+1})^T,$$

where $X' = \mathbf{\Delta}X$. Hence $(X \in \ker \mathbf{A}) \iff (X' = \mathbf{\Delta}X \in l_0)$. As

$$(X \in l_0 \cap l_1) \iff (X = \mathbf{S}_0 V_0 \& \mathbf{\Delta}X \in l_0) \iff (\mathbf{A}\mathbf{S}_0 V_0 = 0) \\ \iff (V_0 \in \ker \mathbf{A}\mathbf{S}_0).$$

Then taking into account that \mathbf{B}_0 is of a size $k \times \mu_0$, we get that $l_1 \cong \ker \mathbf{B}_0$, where $\mathbf{B}_0 = \mathbf{A}\mathbf{S}_0$, $\dim \ker \mathbf{B}_0 = \mu_1$ and $\text{rank } \mathbf{B}_0 = k$. So $\ker \mathbf{B}_0 = \{\mathbf{G}_1 V_1 | V_1 \in R^{\mu_1}\}$, where $\mathbf{G}_1 = (g_1, g_2, \dots, g_{\mu_1})$ is a matrix whose columns are the basis of $\ker \mathbf{B}_0$. Therefore $(X \in l_1) \iff (X = \mathbf{S}_0 \mathbf{G}_1 V_1)$, where V_1 is an arbitrary vector from R^{μ_1} . Consequently $\mathbf{S}_1 = \mathbf{S}_0 \mathbf{G}_1$. Matrix $\mathbf{S}_j, j = 2, 3, \dots, N-1$ is defined by induction. First matrix $\mathbf{B}_{j-1} = \mathbf{A}\mathbf{\Delta}^{j-1} \mathbf{S}_{j-1}$ of a size $k \times \mu_{j-1}$ is constructed. Moreover, $\text{rank } \mathbf{B}_{j-1} = k$, $\dim \ker \mathbf{B}_{j-1} = \mu_j$. Next matrix \mathbf{G}_j whose columns are the basis of $\ker \mathbf{B}_{j-1}$ is built. Then $\mathbf{S}_j = \mathbf{S}_{j-1} \mathbf{G}_j$, $\text{rank } \mathbf{S}_j = \mu_j$.

The behaviour of the spline at the end points $t = a$ and $t = b$ is of a great importance for solving the BVP (2.1) and (2.2). For example, $\tilde{L}_j \subset L_j$ is the variety defined by the conditions:

$$p(a) = \xi_1, p'(a) = \eta_1, p(b) = \xi_2, p'(b) = \eta_2,$$

where $\xi_1, \xi_2, \eta_1, \eta_2$ are parameters. Variety $\tilde{l}_j \subset l_j$ corresponds to the variety \tilde{L}_j in case if we are using vectors X instead of splines $p(t)$. By using primary matrices for derivatives $\mathbf{\Delta}^{(i)}$ and matrices \mathbf{S}_j it is possible to build two matrices \mathbf{M}_1 and \mathbf{M}_2 of sizes $(k+1)N \times 4$ and $(k+1)N \times (\mu_j - 4)$ such that the following theorem holds.

Theorem 1. $X \in \tilde{l}_j$ if and only if $X = \mathbf{M}_1 \Gamma + \mathbf{M}_2 W$, where $\Gamma = (\xi_1, \eta_1, \xi_2, \eta_2)^T$ and the vector $W \in R^{\mu_j - 4}$.

Proof. We construct a matrix \mathbf{R} of size $4 \times (k+1)N$, its first row $r_1 = (1, 0, \dots, 0)$, the third one $r_3 = (0, \dots, 0, 1)$, but the second and the last rows are equal to the first and the last rows of the matrix of derivatives $\mathbf{\Delta}$. As $(X \in l_j) \iff (X = \mathbf{S}V)$, where $\mathbf{S} = \mathbf{S}_j$ and $V = V_j \in R^{\mu_j}$, then $(X \in \tilde{l}_j) \iff (\mathbf{\Phi}V = \Gamma)$, where $\mathbf{\Phi} = \mathbf{R}\mathbf{S}$

is a $4 \times \mu_j$ -matrix. It easy to see that $\text{rank } \Phi = 4$. Hence the system $\Phi V = \Gamma$ is compatible for each vector Γ , and therefore its general solution can be presented in a form

$$V = \Phi_2(\Phi\Phi_2)^{-1}\Gamma + \Phi_1W,$$

where Φ_1 is a $\mu_j \times (\mu_j - 4)$ matrix with a basis of $\ker \Phi$ as its columns, and Φ_2 is the matrix with a basis of $\ker \Phi_1^T$ as its columns and has a size of $\mu_j \times 4$, $W \in R^{\mu_j - 4}$. Since matrix $\Phi\Phi_2$ is an invertible one, so we get that

$$\mathbf{M}_1 = \mathbf{S}\Phi_2(\Phi\Phi_2)^{-1}, \quad \mathbf{M}_2 = \mathbf{S}\Phi_1.$$

■

The other varieties for solving BVP (2.1) and (2.2) can be constructed in the same way. Our main idea for the solving BVP of ODE is to construct such splines, which correspond specially to the considered BVP. Algorithms for numerical solution of different BVP are described in the following sections.

3. BVP for the Second Order ODE

Let us consider BVP (2.1) and (2.2) for $n = 2$:

$$\begin{cases} \mathbf{U}_2x = f, & t \in (a, b), \\ \phi_1(x(a), x'(a), x(b), x'(b)) = 0, \\ \phi_2(x(a), x'(a), x(b), x'(b)) = 0. \end{cases} \quad (3.1)$$

Using Theorem 1 we find an approximate solution of equation $\mathbf{U}_2x = f$ in the form of a spline $X = \mathbf{M}_1\Gamma + \mathbf{M}_2W$. Smoothness of this spline is $j \geq 2$, since the differential equation is of the second order.

Vector X depends on $\mu = \mu_j = (k + 1)N - (j + 1)k$ parameters (that are the components of Γ and W). In order to get this parameters we use the method of collocation and apply it at nodes of template Σ . So we get an approximation of equation $\mathbf{U}_2x = f$ by a system $\mathbf{A}W = \Phi\Gamma_1$, where

$$\mathbf{A} = \mathbf{D}\mathbf{M}_2, \quad \mathbf{D} = \mathbf{Q}_2\Delta^2 + \mathbf{Q}_1\Delta + \mathbf{Q}_0$$

is a $(k + 1)N \times (k + 1)N$ matrix, $\mathbf{Q}_k = \text{diag}(q_k(\Sigma))$, $k = 0, 1, 2$; Δ is a matrix for derivatives, $\Phi = (\Phi_0, F)$ is $(k + 1)N \times 5$ block matrix, $\Phi_0 = -\mathbf{D}\mathbf{M}_1$, and $F = f(\Sigma)$ is a column matrix, $\Gamma_1 = (\Gamma, 1)^T = (\xi_1, \eta_1, \xi_2, \eta_2, 1)^T$.

The following part of the algorithm we will describe only heuristically without formulating and proving any theorem.

3.1. Let homogeneous system $\mathbf{A}W = 0$ approximates BVP

$$\begin{cases} \mathbf{U}_2x = 0, & t \in (a, b), \\ x(a) = x'(a) = 0, & x(b) = x'(b) = 0, \end{cases}$$

then system $\mathbf{A}W = 0$ has only a trivial solution. That means that columns of matrix \mathbf{A} are linear independent ($\text{rank } \mathbf{A} = \mu - 4$).

3.2. Non-homogeneous system $\mathbf{A}W = \Phi\Gamma_1$ is an approximation of BVP

$$\begin{cases} \mathbf{U}_2x = f, & t \in (a, b), \\ x(a) = \xi_1, & x'(a) = \eta_1, \quad x(b) = \xi_2, \quad x'(b) = \eta_2. \end{cases} \quad (3.2)$$

In general case for arbitrary $\xi_1, \eta_1, \xi_2, \eta_2$ it does not have a solution. A special set $(\xi_1, \eta_1, \xi_2, \eta_2)$ must be considered with two additional conditions (a general solution of equation $\mathbf{U}_2x = f$ has only two free parameters). We can get these conditions from a compatibility condition of system $\mathbf{A}W = \Phi\Gamma_1$.

3.3. It follows from the third Fredholm's theorem that system $\mathbf{A}W = \Phi\Gamma_1$ has a solution if and only if $\Phi\Gamma_1 \perp \ker \mathbf{A}^T$. Let $l = (k+1)N - \mu + 4$ and define $l \times (k+1)N$ matrix \mathbf{P} , whose rows are the basis of $\ker \mathbf{A}^T$. Matrix $\mathbf{B} = \mathbf{P}\Phi$ has a size $l \times 5$ and its rank is 2. In a space of rows of matrix \mathbf{B} we choose an orthonormal basis $s = (s_1, s_2, s_3, s_4, s_5), \tau = (\tau_1, \tau_2, \tau_3, \tau_4, \tau_5)$.

System $\mathbf{A}W = \Phi\Gamma_1$ is compatible if and only if

$$\begin{cases} s_1\xi_1 + s_2\eta_1 + s_3\xi_2 + s_4\eta_2 = -s_5, \\ \tau_1\xi_1 + \tau_2\eta_1 + \tau_3\xi_2 + \tau_4\eta_2 = -\tau_5. \end{cases} \quad (3.3)$$

If $f = 0$ then $s_5 = \tau_5 = 0$. The system (3.3) is a compatible one, because there always exist such $\xi_1, \eta_1, \xi_2, \eta_2$ that BVP (3.2) has a solution.

3.4. As $\xi_1 \approx x(a), \eta_1 \approx x'(a), \xi_2 \approx x(b), \eta_2 \approx x'(b)$, where $x = x(t)$ is a solution of equation $\mathbf{U}_2x = f$, then system (3.3) and boundary conditions of BVP (3.1) give a system of four equations

$$\begin{cases} s_1\xi_1 + s_2\eta_1 + s_3\xi_2 + s_4\eta_2 = -s_5, \\ \tau_1\xi_1 + \tau_2\eta_1 + \tau_3\xi_2 + \tau_4\eta_2 = -\tau_5, \\ \varphi_1(\xi_1, \eta_1, \xi_2, \eta_2) = 0, \\ \varphi_2(\xi_1, \eta_1, \xi_2, \eta_2) = 0 \end{cases} \quad (3.4)$$

to get values of four unknowns $\xi_1, \eta_1, \xi_2, \eta_2$.

3.5. System (3.4) is the main part of the proposed numerical algorithm.

- BVP (3.1) has a solution if and only if the system (3.4) is a compatible one.
- Every solution of (3.4) gives a set of approximate values of $x(a), x'(a), x(b), x'(b)$ for the problem (3.1).
- If values of $x(a), x'(a), x(b), x'(b)$ are obtained, then we can complete the solution in various ways.

For example, if $\Gamma = (\xi_1, \eta_1, \xi_2, \eta_2)^T$ is a solution of system (3.4), then forming matrix $\Gamma_1 = (\xi_1, \eta_1, \xi_2, \eta_2, 1)^T$ and solving compatible system $\mathbf{A}W = \Phi\Gamma_1$, we get an approximation of solution of the BVP as the following spline

$$X = \mathbf{M}_1\Gamma + \mathbf{R}\Gamma_1, \mathbf{R} = \mathbf{M}_2(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\Phi. \quad (3.5)$$

When matrices \mathbf{M}_1 and \mathbf{R} are known the solution of BVP can be obtained by inserting the solution of (3.4) into (3.5).

Remark 1. Let $\Gamma = (\xi_1, \eta_1, \xi_2, \eta_2)^T$ be a general solution of system (3.3). This means that $\Gamma = V_0 + c_1V_1 + c_2V_2$, where V_0 is a particular solution of the system and $c_1V_1 + c_2V_2$ is a general solution of the homogeneous system. By solving system $\mathbf{A}W = \Phi\Gamma_1$ we can get an approximation of the general solution of the equation $\mathbf{U}_2x = f$ in a form of the spline $X = X_0 + c_1X_1 + c_2X_2$, where

$$\begin{aligned} X_1 &= \mathbf{G}V_1, \quad X_2 = \mathbf{G}V_2, \quad X_0 = \mathbf{G}V_0 + \Omega F, \\ \Omega &= \mathbf{M}_2(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T, \quad \mathbf{G} = \mathbf{M}_1 + \Omega\Phi_0. \end{aligned}$$

Example 1. Let us consider the following BVP

$$\begin{cases} x'' + 2hx' - x = 1, & t \in (0, 1), \\ x^2(0) + (x'(0))^2 = r_1^2, \\ x^2(1) + (x'(1))^2 = r_2^2. \end{cases} \quad (3.6)$$

We introduce the following notations:

$$\Gamma_x = (x(0), x'(0), x(1), x'(1))^T, \quad \Gamma = (\xi_1, \eta_1, \xi_2, \eta_2)^T,$$

where Γ is a corresponding vector of approximate values, $\rho = \|\Gamma_x - \Gamma\|$, $\|\cdot\|$ stands for the Euclidean or unitary norm. The BVP (3.6) has four solutions which can also be complex. If some solution is complex then we use ρ^* instead of ρ .

Results of numerical experiments are presented in Tables 1–8. In all tables $N - 1$ is the order of the spline, $k + 1$ is the number of subintervals, j is smoothness of the spline. In Tables 4–8 we use a notation $\rho = \max(\rho_1, \rho_2, \rho_3, \rho_4)$.

We can see from Tables 1 – 3 that the accuracy of approximations is slightly dependent on the smoothness j of the spline. Since the differential equation is of the second order, we use $j \geq 2$.

It follows from Tables 4–7 that if the order of spline is not high, the precision is equivalent to the method of the finite differences. With the growth of the order the accuracy of approximation increases. In all cases there is an optimal order, the exceeding of which does not improve the accuracy of the result.

Example 2. Let us consider BVP with a singular point

$$\begin{cases} t^2y'' + ty' + (t^2 - \nu^2)y = 0, & 0 < t < b, \quad \nu \geq 0, \\ |y(0)| \leq M, \quad \alpha y(b) + \beta y'(b) = \gamma. \end{cases} \quad (3.7)$$

By means of the following substitution $y(t) = t^\nu x(t)$ we get the problem

$$\begin{cases} tx'' + (2\nu + 1)x' + tx = 0, & 0 < t < b, \\ x(t) \text{ is an entire function,} \\ \alpha_1 x(b) + \beta_1 x'(b) = \gamma_1. \end{cases} \quad (3.8)$$

Approximation with a spline of an entire function $x(t)$ is more effective than of a function $y(t) = t^\nu x(t)$ for an arbitrary $\nu \geq 0$.

Table 1. Example 1 ($r_1 = 1.881, r_2 = 2, h = 4, k = 10, N = 10$).

j	ρ_1	ρ_2	ρ_3	ρ_4
2	$5 \cdot 10^{-10}$	$5 \cdot 10^{-10}$	$4 \cdot 10^{-11}$	$4 \cdot 10^{-11}$
4	$2 \cdot 10^{-11}$	$4 \cdot 10^{-11}$	$2 \cdot 10^{-11}$	$4 \cdot 10^{-11}$
7	$5 \cdot 10^{-9}$	$5 \cdot 10^{-9}$	$5 \cdot 10^{-10}$	$3 \cdot 10^{-10}$

Table 2. Example 1 ($r_1 = 1.7, r_2 = 2, h = 4, k = 10, N = 10$).

j	ρ_1^*	ρ_2^*	ρ_3	ρ_4
2	$4 \cdot 10^{-11}$	$4 \cdot 10^{-11}$	$5 \cdot 10^{-11}$	$5 \cdot 10^{-11}$
4	$2 \cdot 10^{-11}$	$2 \cdot 10^{-11}$	$2 \cdot 10^{-11}$	$4 \cdot 10^{-11}$
7	$5 \cdot 10^{-10}$	$5 \cdot 10^{-10}$	$8 \cdot 10^{-10}$	$7 \cdot 10^{-10}$

Table 3. Example 1 ($r_1 = 1.5, r_2 = 2, h = 4, k = 10, N = 10$).

j	ρ_1^*	ρ_2^*	ρ_3^*	ρ_4^*
2	10^{-11}	10^{-11}	10^{-11}	10^{-11}
4	$3 \cdot 10^{-11}$	$3 \cdot 10^{-11}$	$4 \cdot 10^{-11}$	$4 \cdot 10^{-11}$
7	$3 \cdot 10^{-10}$	$3 \cdot 10^{-10}$	10^{-10}	10^{-10}

Table 4. Example 1 ($r_1 = 3, r_2 = 2, h = 4, k = 3, j = 3$).

N	6	8	10	15	25	30
ρ	10^{-3}	10^{-5}	10^{-8}	10^{-11}	$6 \cdot 10^{-11}$	$7 \cdot 10^{-11}$

Table 5. Example 1 ($r_1 = 3, r_2 = 2, h = 4, k = 5, j = 3$).

N	4	6	8	10	15	20
ρ	10^{-2}	10^{-3}	$4 \cdot 10^{-7}$	10^{-9}	10^{-11}	$5 \cdot 10^{-11}$

Table 6. Example 1 ($r_1 = 3, r_2 = 2, h = 4, k = 20, j = 3$).

N	4	6	8	12
ρ	$4 \cdot 10^{-3}$	10^{-6}	$2 \cdot 10^{-10}$	$3 \cdot 10^{-10}$

Table 7. Example 1 ($r_1 = 3, r_2 = 2, h = 4, k = 50, j = 3$).

N	4	6	8	10
ρ	10^{-3}	10^{-7}	10^{-10}	$2 \cdot 10^{-10}$

Table 8. Example 1 ($r_1 = 3, r_2 = 2, j = 2$).

k	10	10	10	10	10	14	18	10
N	10	10	10	10	10	10	10	16
h	4	10	20	30	40	40	40	40
ρ	$3 \cdot 10^{-11}$	10^{-8}	$6 \cdot 10^{-7}$	$2 \cdot 10^{-5}$	$2 \cdot 10^{-5}$	$8 \cdot 10^{-6}$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-9}$

The numerical algorithm is the same. The only difference is that $\text{rank } \mathbf{B} = 3$. Instead of system (3.3) we have analogical system of three equations. Together with equation $\alpha_1 \xi_2 + \beta_1 \eta_2 = \gamma_1$ they give a system of four equations to get values of

$$\xi_1 \approx x(0), \quad \eta_1 \approx x'(0), \quad \xi_2 \approx x(b), \quad \eta_2 \approx x'(b).$$

On a set of four free parameters $\xi_1, \eta_1, \xi_2, \eta_2$ one must give three additional conditions. Equation (3.8) has only one linear independent solution which satisfies conditions: $x(0) = \xi_1, x'(0) = \eta_1$.

In Table 9 we present results of numerical experiments for $\alpha = 1, \beta = -0.01, \gamma = 2$. We have used the following parameters:

$$k = 10, \quad N = 10, \quad j = 3, \quad Y = y(\Sigma),$$

where $y(t)$ is the exact solution of problem (3.7), Σ is a set of all node points, Y_p is a vector of the approximate values, $\rho = \|Y - Y_p\|$, where $\|\cdot\|$ stands for l_∞ -norm.

Table 9. Example 2 (Spline parameters: $k = 10, N = 10, j = 3$).

b	1	1	1	1	6	6	6	12	12
ν	1	2	3	4	1	2	5	1	2
ρ	$7 \cdot 10^{-10}$	$2 \cdot 10^{-10}$	$3 \cdot 10^{-11}$	$4 \cdot 10^{-11}$	$5 \cdot 10^{-10}$	$3 \cdot 10^{-10}$	10^{-11}	10^{-9}	10^{-8}

4. Higher Degrees ODE and Systems

4.1. The algorithm for solving BVP (2.1)–(2.2) for $n > 2$ is analogous to the case $n = 2$. Let, for example, $n = 4$ and boundary conditions (2.2) be of the form

$$\begin{aligned} \phi_k(x(a), x^{(1)}(a), x^{(2)}(a), x^{(3)}(a), \\ x(b), x^{(1)}(b), x^{(2)}(b), x^{(3)}(b)) = 0, \quad k = 1, 2, 3, 4. \end{aligned} \quad (4.1)$$

Let $\Gamma = (\xi_1, \eta_1, \mu_1, \kappa_1, \xi_2, \eta_2, \mu_2, \kappa_2)^T$ be a vector of parameters and

$$\Gamma_y = (y(a), y^{(1)}(a), y^{(2)}(a), y^{(3)}(a), y(b), y^{(1)}(b), y^{(2)}(b), y^{(3)}(b))^T$$

be a vector of values of function $y = y(t)$ and its derivatives at the end points. Let spline $p(t) \in L_j$. By using condition $\Gamma_p = \Gamma$ we construct a variety $\widetilde{L}_j \subset L_j$. If splines are given as vectors $X = p(\Sigma)$ then variety $\widetilde{l}_j \subset l_j$ corresponds to the variety \widetilde{L}_j . It is easy to get a result analogous to Theorem 1.

Finally we approximate the solution of equation $\mathbf{U}_4 x = f$ by a spline

$$X = \mathbf{M}_1 \Gamma + \mathbf{M}_2 W, \quad W \in R^{\mu-8},$$

where \mathbf{M}_1 and \mathbf{M}_2 are given matrices, $\mu = \mu_j = (k+1)N - (j+1)k$. Smoothness of this spline $j \geq 4$. Vector X depends on μ parameters. The remaining analysis is equivalent to the case $n = 2$. The only difference is that the dimension of row-space of matrix \mathbf{B} is equal to 4. Hence instead of system (3.3) we get an analogical system of four equations. Together with boundary conditions (4.1) we get a system of 8 equations, that is analogical to system (3.4). Solving it we get vector Γ which approximates vector Γ_x , where $x = x(t)$ is the solution of BVP.

Example 3. Let us consider the following BVP

$$\begin{cases} x^{(4)} + 2x^{(3)} + x^{(2)} - x^{(1)} + x = 1, & t \in (0, b), \\ x(0) = x^{(3)}(0) = 0, & x(b) = x^{(3)}(b) = 0. \end{cases} \quad (4.2)$$

The results of numerical experiments are presented in Tables 10–12. Here we use the following notation: k, N, j are parameters of the spline, $\rho = \|\Gamma - \Gamma_x\|$, where $\|\cdot\|$ stands for the l_∞ norm.

Table 10. Example 3 ($b = 1$).

k	N	j	ρ
3	10	5	10^{-8}
5	10	5	10^{-8}
10	10	5	10^{-8}
10	10	7	10^{-7}

Table 11. Example 3 ($b = 20$).

k	N	j	ρ
10	10	5	$8 \cdot 10^{-5}$
20	10	5	$7 \cdot 10^{-6}$
20	16	5	$4 \cdot 10^{-8}$

The behavior of numerical results is analogical to the case $n = 2$.

4.2. The given algorithm can be applied to solve BVP of systems of ODE. Let us consider the following system

Table 12. Example 3 ($b = 30$).

k	N	j	ρ
10	10	5	$5 \cdot 10^{-4}$
10	10	7	$3 \cdot 10^{-3}$
20	10	5	$2 \cdot 10^{-5}$
20	16	5	$3 \cdot 10^{-7}$

$$\mathbf{q}x' + \mathbf{s}x = f(t), \quad t \in (a, b), \quad (4.3)$$

where

$$x = x(t) = (x_1(t), \dots, x_4(t))^T, \quad \mathbf{q} = \mathbf{q}(t) = \text{diag}(q_1(t), \dots, q_4(t)),$$

$$\mathbf{s} = \mathbf{s}(t) = \|s_{ij}(t)\|_{4,4}, \quad f(t) = (f_1(t), f_2(t), f_3(t), f_4(t))^T.$$

The boundary conditions are given by

$$\phi_k(x_1(a), x_1(b), \dots, x_4(a), x_4(b)) = 0, \quad k = 1, 2, 3, 4. \quad (4.4)$$

It is assumed that all functions are smooth enough and $\det \mathbf{q}(t) \neq 0, t \in [a, b]$. The following notation is introduced: $\Gamma = (\nu_1, \nu_2, \mu_1, \mu_2, \eta_1, \eta_2, \xi_1, \xi_2)$ is a vector of parameters, $\Gamma_y = (y_1(a), y_1(b), \dots, y_4(a), y_4(b))^T$ is a vector of function $y(t) = (y_1(t), \dots, y_4(t))^T$ values at the end points, $L_j^4 = L_j \oplus L_j \oplus L_j \oplus L_j$ and the smoothness of the spline is $j \geq 1$.

Let $p(t) = (p_1(t), p_2(t), p_3(t), p_4(t)) \in L_j^4$. Using a condition $\Gamma_p = \Gamma$ we construct a variety $\widetilde{L}_j^4 \subset L_j^4$. If splines are given as vectors $X = (p_1(\Sigma), \dots, p_4(\Sigma))^T \in R^{4(k+1)N}$, then variety $\widetilde{l}_j^4 \subset l_j^4 = l_j \oplus l_j \oplus l_j \oplus l_j$ corresponds to variety \widetilde{L}_j^4 . It is easy to prove a result equivalent to Theorem 1.

Finally we approximate the solution of system (4.3) with the spline

$$X = \mathbf{M}_1\Gamma + \mathbf{M}_2W, \quad W \in R^{4\mu-8},$$

where $\mathbf{M}_1 = \mathbf{M}'_1 \oplus \mathbf{M}'_1 \oplus \mathbf{M}'_1 \oplus \mathbf{M}'_1$ is $4(k+1)N \times 8$ block diagonal matrix, $\mathbf{M}_2 = \mathbf{M}'_2 \oplus \mathbf{M}'_2 \oplus \mathbf{M}'_2 \oplus \mathbf{M}'_2$ is $4(k+1)N \times (4\mu-8)$ block diagonal matrix, $\mu = \mu_j = (k+1)N - (j+1)k$. Using the method of collocation on nodes of the whole template Σ we get system $\mathbf{A}W = \Phi\Gamma_1$ as an approximation of system of ODE (4.3). Here \mathbf{A} and Φ are known matrices of sizes $4(k+1)N \times (4\mu-8)$ and $4(k+1)N \times 9$ respectively and $\Gamma_1 = (\Gamma, 1) = (\nu_1, \nu_2, \mu_1, \mu_2, \eta_1, \eta_2, \xi_1, \xi_2, 1)^T$.

The rest part of construction is analogical to Section 3. The only difference is that the dimension of a row-space of matrix $\mathbf{B} = \mathbf{P}\Phi$ is equal to four. So the condition of compatibility of system $\mathbf{A}W = \Phi\Gamma_1$ is equivalent to a system of four equations. Together with boundary conditions (4.4) it gives a system of 8 equations, which is analogical to system (3.4), for finding vector Γ that approximate a vector Γ_x . Here $x = x(t)$ is a solution of the given BVP.

Example 4. Let us consider the following BVP

Table 13. Example 4 ($b = 10$).

k	5	10	10	20
N	10	10	10	10
j	2	2	6	2
ρ	$5 \cdot 10^{-7}$	$4 \cdot 10^{-9}$	$2 \cdot 10^{-8}$	10^{-12}

$$\begin{cases} x_1' + 2x_1 + x_2 - x_3 + x_4 = 1, & t \in (0, b), \\ x_2' - x_1 = 0, \\ x_3' - x_2 = 0, \\ x_4' - x_3 = 0, \\ x_1(0) = x_4(0) = 0, & x_1(b) = x_4(b) = 0. \end{cases} \quad (4.5)$$

The results of numerical experiments are presented in Table 13. Here we use the following notation: k, N, j are parameters of the spline, $\rho = \|\Gamma - \Gamma_x\|$, where $\|\cdot\|$ stands for the l_∞ norm. Notice that problem (4.5) is equivalent to problem (4.2). A better approximation is explained by increased number free parameters (4μ parameters instead of μ).

5. Decomposition of BVP

The theory that has been introduced helps us also in case of decomposition of BVP of ODE or system of ODE. Let us consider a scheme of decomposition of BVP (2.1)–(2.2) for $n = 2$ with boundary conditions of the third kind, i.e.

$$\begin{cases} \mathbf{U}_2 x = f, & t \in (a, b), \\ \alpha_0 x(a) + \beta_0 x'(a) = \gamma_0, & \alpha_1 x(b) + \beta_1 x'(b) = \gamma_1. \end{cases} \quad (5.1)$$

The main interval $[a, b]$ is divided into n subintervals

$$a = a_1 < b_1 = a_2 < b_2 = a_3 < \dots < b_{n-1} = a_n < b_n = a_{n+1}.$$

On every subinterval $[a_i, b_i], i = 1, 2, \dots, n$ we get a system of compatibility equations (3.3), that can be written as

$$\begin{cases} s_1^{(i)} \xi_1^{(i)} + s_2^{(i)} \eta_1^{(i)} + s_3^{(i)} \xi_2^{(i)} + s_4^{(i)} \eta_2^{(i)} = -s_5^{(i)}, \\ \tau_1^{(i)} \xi_1^{(i)} + \tau_2^{(i)} \eta_1^{(i)} + \tau_3^{(i)} \xi_2^{(i)} + \tau_4^{(i)} \eta_2^{(i)} = -\tau_5^{(i)}, \end{cases} \quad (5.2)$$

where

$$\xi_1^{(i)} \approx x(a_i), \quad \eta_1^{(i)} \approx x'(a_i), \quad \xi_2^{(i)} \approx x(b_i), \quad \eta_2^{(i)} \approx x'(b_i), \quad b_i = a_{i+1}.$$

Let $Z_x = (x(a_1), x'(a_1), x(a_2), x'(a_2), \dots, x(a_{n+1}), x'(a_{n+1}))^T$, and Z is a corresponding vector of approximate values. Then equation (5.2) together with the boundary conditions of BVP (5.1) give system $\mathbf{A}Z = B$. Matrix \mathbf{A} is 5-diagonal,

Table 14. Example 5 (Spline parameters: $k = 10, N = 10, j = 2$).

b	46	66	86	106	126	146	186	246	246	246
n	10	10	10	10	10	10	10	10	20	40
ρ	$2 \cdot 10^{-11}$	$5 \cdot 10^{-11}$	$2 \cdot 10^{-9}$	$5 \cdot 10^{-9}$	$5 \cdot 10^{-8}$	10^{-7}	10^{-6}	10^{-4}	10^{-6}	10^{-9}

$B = (\gamma_0, -s_5^{(1)}, -\tau_5^{(1)}, \dots, -s_5^{(n)}, -\tau_5^{(n)}, \gamma_1)^T$. By solving this system we get approximate values $x(a_i), x'(a_i), i = 1, 2, \dots, n + 1$. After that we can get values of $x(t)$ inside the subintervals in various ways.

Example 5. Let us consider the following BVP

$$\begin{cases} t^2 x'' + tx' + (t^2 - 16)x = 0, & t \in (6, b), \\ x_1(6) - 0.12x'(6) = 2, \\ x(b) + 0.12x'(b) = 1. \end{cases}$$

The results of a numerical experiments are presented in Table 14. Here we use the following notation: k, N, j are parameters of the spline, $\rho = \|Z - Z_x\|/\|Z_x\|$, where $\|\cdot\|$ stands for Euclidean norm.

6. Conclusion

A method for numerical solving of boundary values problems of ordinary differential equations is constructed. It is based on the use of Degenerate Matrices methods and splines. All algorithms are constructed in the matrices form. Possibilities of the method are shown by solving different test examples.

References

- [1] I. Babuška, M. Práger and E. Vitásek. *Numerical Processes in Differential Equations*. Mir, Moscow, 1969. (in Russian)
- [2] G. Hall and J.M. Watt(Eds.). *Modern Numerical Methods for Ordinary Differential Equations*. Claren Press, Oxford, 1976.
- [3] T. Čirulis and O. Lietuvietis. Degenerate matrix method for solving nonlinear systems of differential equations. *Mathematical Modeling and Analysis*, **3**, 45 – 56, 1998.
- [4] T. Čirulis and O. Lietuvietis. Degenerate matrix method for solving nonlinear systems of differential equations. In: *Numerical Mathematics and Advanced Applications. Proceeding of 3-rd European Conference*, volume 6. Word Scientifi c, 456 – 461, 2000.
- [5] T. Čirulis, D. Čirule and O. Lietuvietis. Multistep matrix method for ordinary differential equations. *Mathematical Modeling and Analysis*, **6**(1), 58 – 67, 2001.
- [6] V.F. Zajcev and A.D. Poljanin. *Hand Book of Ordinary Differential Equation*. Fizmatlit, Moscow, 2001. (in Russian)

Splainais sudaromų nereguliacijų matricų metodų taikymas paprastųjų diferencialinių lygčių kraštiniamis uždaviniamis spręsti

M. Belovs, T. Cīrulis

Darbe nagrinėjamas paprastųjų lygčių su kraštinėmis sąlygomis skaitinis sprendimo metodas. Šio metodo pagrindą sudaro nereguliacijų matricų bei splainų konstravimas klasikinių ortogonalinių polinomų pavidalu. Nagrinėjamo straipsnyje metodo taikymo galimybės parodytos įvairiais pavyzdžiais.