Construction of cubic splines for interpolating functional dependencies and processing the results of experimental studies

V V Osadchyi1 and V S Yeremieiev1
1 Bogdan Khmelnitsky Melitopol State Pedagogical University, 20 Hetmanska Str., Melitopol, 72300, Ukraine
E-mail: osadchyi@mdpu.org.ua, eremeev_v@mdpu.org.ua

Abstract. Algorithms for constructing cubic splines with different boundary conditions have been developed: 1. The second derivative takes arbitrary values at the ends of the spline. A special case is the “natural spline”, when the second derivative at its ends is equal to zero. 2. At one end of the spline, the first derivative is given, and at the opposite end – the second derivative. 3. The first derivative is given at the two ends of the spline without any restrictions on the second derivatives. The proposed methods were tested on the example of interpolation of the function $10x \sin xe^{-x}$ for the interval $[0, 1]$. The relative calculation error for the number of nodes $n = 10$ was about $10^{-2}$. With an increase in the number of nodes to 1000 and 10000, the error increases, respectively, to $10^{-6}$ and $10^{-7} – 10^{-8}$. It is shown that a number of problems with other boundary conditions can be solved using the proposed methods. The considered algorithms can be used for interpolating functional dependencies and processing the results of experimental studies, represented as a discrete set of pairs of numbers.

1. Introduction
Restoring functional dependence on the basis of a discrete set of numbers is one of the important issues of computational mathematics. In the simplest case, this set is specified in the form of a table with two variables $(x_k, y_k), k = 0, 1, 2, \ldots, n$, belonging to the interval $[a, b]$. Identification of the dependence $y(x)$ for large $n$ using interpolation polynomials, like the Lagrange polynomial, encounters insuperable difficulties that can be avoided using splines. The theory of polynomial splines was developed in the works [1–9] and others. First of all, it expanded the possibilities of interpolating functional dependencies [10,11].

Later, the field of application of spline functions covered many sections in the theory of differential equations, mathematical modeling [12], in the design of various profiles [13], in computer graphics [14–16], when processing experimental data [17–19]. Using splines, it is possible to describe the trajectories of moving objects, a change in the distribution of temperature and electromagnetic fields, diffusion processes in multilayer systems [8], they can also be used when solving other problems of applied mathematics and geometry [3]. The most widespread is the cubic spline, which is distinguished by its simplicity and ease of implementation when solving a wide range of problems [20,21]. Most often, the so-called “natural spline” is used, which corresponds to the assumption that its second derivatives at the boundaries of the interval $[a, b]$ are equal to zero [3]. The use of “natural spline” guarantees a sufficiently high accuracy...
of interpolation of continuous functions with continuous first and second derivatives. In many cases, there is a need to use other boundary conditions. In particular, these include the following tasks:

- finding profiles of various types with given boundary conditions,
- constructing the lines on a closed contour,
- processing the results of experimental studies on the dependence of one factor on another with the selected boundary conditions.

This work is devoted to the creation of algorithms that provide the construction of cubic splines of this type.

2. Research work objectives

Supposing on the interval \([a, b]\) there is a continuous function \(F(x)\) with continuous derivatives up to the third order. Imagine that at the nodal points \(x_0 = a, x_1, x_2, ..., x_n = b\), this function takes known values equal, respectively, to \(y_0, y_1, y_2 ... y_n\). Let us consider the possibility of recovering this function for the entire interval \([a, b]\) with a certain accuracy \(\delta F\). Theoretically, the use of well-known interpolation polynomials makes it possible to obtain a solution to the problem posed with an arbitrarily high accuracy. However, the accumulation of computational errors for sufficiently large values of \(n\) does not allow achieving the desired results. In this case, the spline interpolation method can be used. Cubic splines are polynomials of the least degree that have continuous second derivatives. The most widespread is the cubic spline \(S(x)\), given by polynomials not higher than the third order [3].

\[
f_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3, k = [1, ..., n], x_{k-1} <= x <= x_k, \quad (1)
\]

A great advantage of cubic splines is the simplicity of the algorithms for their construction, which ensures their wide use in many applied problems with a relatively small error. To ensure the continuity of the spline, it is necessary to require that the value of the spline at the nodal points \(S(x_k)\) has to be equal to \(f_k(x_k) = y_k\) and \(f_k(x_{k-1}) = f_k - 1(x_{k-1})\). In this case, the constants in the polynomial \(a_k\) are equal to \(y_k\). Let us emphasize the differences between neighboring coordinates of nodes and the values of functions at these nodes by \(\delta x_k = x_k - x_{k-1}, \delta y_k = y_k - y_{k-1}, k = [1, ..., n]\). The condition of the spline continuity implies that the equalities are fulfilled as follows

\[
g_k = b_k - c_k\delta x_k + d_k\delta x_k^2, k = [1, ..., n], \quad (2)
\]

where \(g_k = \delta y_k/\delta x_k\).

The polynomials (1) obtained in this way restore the function \(F(x)\) on the entire interval \([a, b]\). To ensure the smoothness of the spline, it is necessary to require the continuity of the derivatives of the spline at the nodal points. Let’s take \(f'_{k-1}(x_{k-1-1}) = b_k - 1, f''_{k-1}(x_{k-1}) = c_k-1\) where, taking into account the formula (1), it follows that

\[
b_k = b_{k-1} + 2c_k\delta x_k - 3d_k\delta x_k^2, k = [2, ..., n], \quad (3)
\]

\[
d_k = (c_k - c_{k-1},)/3(\delta x_k), k = [2, ..., n], \quad (4)
\]

After the substitution of \(d_k\delta x_k\) from (4) into formulas (2), (3) we will get

\[
b_k = g_k + c_k\delta x_k/2 + c_{k-1}\delta x_k/3, k = [2, ..., n], \quad (5)
\]

\[
b_k - b_{k-1} - c_k\delta x_k - c_{k-1}\delta x_k = 0, k = [2, ..., n], \quad (6)
\]

From formula (5) it follows that \(b_{k-1} = c_{k-1}\delta x_{k-1}/2 + c_{k-2}\delta x_{k-1}/3 + g_{k-1}, k = [3, ..., n]\).
Substituting the obtained result and formula (5) into (6), we will get
\[ 2\delta x_k c_{k-1} + (4\delta x_{k+1} + 3\delta x_k)c_k + 3\delta x_{k+1} c_{k+1} + 1 = 6h_{k+1}, k = [2, ..., n - 1], \]
(7)
where \( h_k = g_k - g_{k-1} \).

The solution of the system of equations (7) makes it possible to identify the coefficients of the spline \( d_k \) and \( b_k \) by formulas (4), (5) and, thus, to ensure the restoration of the function. \( F(x) \) using polynomials (1). Thus, the main problem is reduced to solving a system of \( n - 2 \) linear algebraic equations (7), which contains \( n \) unknowns \( c_k, k = [1, ..., n] \). For its numerical solution, it is necessary to take two additional relations. In many applications, they are set by the conditions for the second derivatives at the ends of the interval \([a,b]\) in the form of \( S''(a) = 0, S''(b) = 0 \) [9].

The purpose of the work is to develop an algorithm for constructing cubic splines with different boundary conditions.

3. Work outcomes

3.1. General solution of the problem
Let us write down the equations (7) in the form
\[ A_k c_{k-1} - B_k c_k + C_k c_{k+1} = D_k, k = [2, \ldots, n - 1], \]
(8)
or in matrix form
\[
\begin{pmatrix}
A_2 & -B_2 & C_2 & D_2 \\
A_3 & -B_3 & C_3 & D_3 \\
& & \ddots & \ddots \\
& & & A_{n-1}
\end{pmatrix}
\begin{pmatrix}
c_{k-1} \\
c_k \\
c_{k+1}
\end{pmatrix}
= \begin{pmatrix}
D_2 \\
D_3 \\
\ddots \\
D_{n-1}
\end{pmatrix},
\]
where
\[ A_k = 2\delta x_k, B_k = -(4\delta x_{k+1} + 3\delta x_k), C_k = 3\delta x_{k+1}, D_k = 6h_{k+1}, k = [2, \ldots, n - 1]. \]

To solve system (8) of \( n - 2 \) equations with respect to unknowns \( c_k \), we “extend” it to \( n \) equations by including two relations,
\[ c_0 = \alpha_0 c_1 + \beta_0, c_{n-1} = \alpha_{n-1} c_n + \beta_{n-1}. \]
(10)

Coefficients \( \alpha_0, \beta_0, \alpha_n, \beta_n \) will be later specified taking into account the selected boundary conditions.

Substitute the expression for \( c_0 \) from (10) into formula (9), which gives
\[ A_1 (\alpha_0 c_1 + \beta_0) - B_1 c_1 + C_1 c_2 = D_1, \]
(11)
where follows the dependence of the first radical \( c_1 \) on the second radical \( c_2 \)
\[ c_1 = \alpha_1 c_2 + \beta_1, \]
(12)
where \( \alpha_1 = C_1/(B_1 - A_1 \alpha_0), \beta_1 = (A_1 \beta_0 - D_1)/(B_1 - A_1 \alpha_0) \).

Similarly, substituting \( c_1 \) from (12) into formula (11), we obtain an expression from which we can identify \( c_2 = \alpha_2 c_3 + \beta_2 \). Next, using the induction method, we find the \( k \) radical
\[ c_k = \alpha_k c_{k+1} + \beta_k, k = [1, ..., n - 1], \]
(13)
with known coefficients
\[\alpha_k = C_k/(B_k - A_k\alpha_{k-1}), \beta_k = (A_k\alpha_{k-1} - D_k)/(B_k - A_k\alpha_{k-1}), k = [1, \ldots, n - 1].\] (14)

After substituting the coefficients of equation (9) \(A_k, B_k, C_k, D_k\) into expressions (14), we find
\[\alpha_k = -3\delta x_{k+1}/(4\delta x_{k+1} + 3\delta x_k\alpha_{k-1})k = [1, \ldots, n - 1],\] (15)
\[\beta_k = (-2\delta x_k\beta_{k-1} + 6h_{k+1})/(4\delta x_{k+1} + 3\delta x_k + \alpha_{k-1}2\delta x_k), k = [1, \ldots, n - 1].\] (16)

To identify the first coefficients \(\alpha_1, \beta_1\) and the last radical \(c_n\), additional information is needed on the behavior of the spline on its boundaries.

3.2. Algorithms for constructing splines with different boundary conditions

Let’s consider the following cases which can be encountered while analyzing the applied problems:

1. The second derivatives of the spline at the ends of the interval \([a, b]\) are determined by the conditions
\[S''(a) = s_1, S''(b) = s_2.\] (17)

2. At one end of the interval \([a, b]\), the first derivative of the spline \(S'(a) = s_3\), is specified, at the opposite end – the second derivative \(S''(b) = s_2\).

3. On the boundaries of the interval \([a, b]\), the first derivatives are set: \(S'(a) = s_3, S'(b) = s_2\).

4. It provides a link between the values of the spline and its first derivative at the boundaries of the area: \(f_1(a) = f_n(b), f'_1(a) = f'_n(b)\).

Case 1. Construction of a spline under condition (17).

The second derivative at the nodal point \(x_n = b\) is equal to \(2c_n\); therefore, condition (17) implies the equality \(c_n = s_2\). From the second condition at the nodal point \(x_1 = a\) it follows that \(c_1 - 3d_1\delta x_1 = s_1\), which makes it possible to expand the possibility of using formula (4) to the interval \(k = [1, \ldots, n]\) by introducing an additional constant \(c_0 = s_1\). The first coefficient \(\alpha_1\) is obtained from formula (14) by setting the value of \(\alpha_0\) in (10) equal to zero
\[\alpha_1 = -3\delta x_2/(4\delta x_2 + 3\delta x_1).\] (18)

Taking the value \(\beta_0\) in (10) equal to zero, we identify the coefficient \(\beta_1\) from the formula (16)
\[\beta_1 = 6h_2/(4\delta x_2 + 3\delta x_1).\] (19)

The numerical algorithm for constructing a spline is as follows:

Step 1. Calculation of the coefficients of the tridiagonal matrix \(A_k, B_k, C_k, D_k, k = [1, \ldots, n - 1]\), by formulas (9).

Step 2. Identification of the sweep coefficients \(\alpha_k, \beta_k\) using recursion relations (15), (16). The first values \(\alpha_1, \beta_1\) are calculated for \(\alpha_1 = 0, \beta_0 = 0\).

Step 3. Calculation of coefficients \(c_k\) in polynomials (1) using recursion formula (13).

Step 4. Finding the coefficients \(d_k, b_k\) by formulas (4), (5).

Step 5. Identification of the value of the spline at a given point belonging to the interval \([a, b]\) using formula (1).
Table 1. Results of testing the interpolation of the function $10x \sin(x)e^{-x}$ ($n = 100$).

| $x$   | $f(x)$   | Spline($x$) |
|-------|----------|-------------|
| 0     | 0.980000 | 3.054610    |
| 1     | 0.981000 | 3.056718    |
| 2     | 0.982000 | 3.058820    |
| 3     | 0.983000 | 3.060915    |
| 4     | 0.984000 | 3.063005    |
| 5     | 0.985000 | 3.065088    |

The solution of this problem in a particular case with zero conditions for the second derivatives $s_1 = s_2 = 0$ is described in many publications [9]. These splines are called “natural”. The proposed algorithm was tested on the example of interpolating the function $10x \sin(x)e^{-x}$ for the interval $[0, 1]$. The relative calculation error for the number of nodes $n = 10$ was about $10^{-2}$. Table 1 shows comparative data in the case of testing at $n = 100$.

According to the table 1, the relative interpolation error is about $3 \cdot 10^{-5}$. An increase in the number of nodes to $10^3$ provides a calculation accuracy of up to $3 \cdot 10^{-8}$.

Case 2. Construction of a spline under conditions

$$S'(a) = s_3, S''(b) = s_2.$$ (20)

As the spline value $S(a)$ is equal to $y_0$, so according to (1) we have

$$g_1 - b_1 + c_1 \delta x_1 - d_1 \delta x_1^2 = 0.$$ (21)

The value of the first derivate with $x = a$ is equal to

$$f'(a) = b_1 - 2c_1 \delta x_1 + 3d_1 \delta x_1^2 = s_3.$$ (22)

From (21) and (22) we can identify the coefficient

$$b_1 = (-s_3 + 3g_1 + c_1 \delta x_1)/2.$$ (23)

The algorithm for constructing a spline under conditions (20) will change and become as follows:

Step 1. Calculation of the coefficients of the tridiagonal matrix $A_k, B_k, C_k, D_k, k = [1, \ldots, n - 1]$ and identification of the sweep coefficients $\alpha_k, \beta_k$ by formulas (9), (15) and (16). The first values $\alpha_1, \beta_1$ are calculated for $\alpha_1 = 0, \beta_0 = 0$.

Step 2. Calculation of coefficients $c_k$ in polynomials (1) using the recursion formula (13), while the value $c_n$ is taken equal to $s_2$.

Step 3. Identification of the coefficient $b_1$ using the formula (23). The rest of the coefficients for $b_k$ are calculated by formula (5).

Step 4. Finding the coefficients $d_k$ by formula (4).

Step 5. Identification of the value of the spline at a given point belonging to the interval $[a, b]$ using formula (1).
Case 3. Construction of a spline under conditions

\[ S'(a) = s_3, S'(b) = s_4. \]  \hfill (24)

The uniqueness of the solution of the problem provides the opportunity to calculate all parameters related to the spline with conditions (24). Let’s calculate the coefficients of spline at terms

\[ a_n = y_n, b_n = s_4, d_n = 0. \]  \hfill (25)

In this case

\[ c_n = (s_4 - g_n)/\delta x_n. \]  \hfill (26)

The algorithm for constructing a spline with conditions (24) will be as follows:

Step 1. Calculation of the coefficients of the tridiagonal matrix \( A_k, B_k, C_k, D_k, k = 1, \ldots, n - 1 \) under conditions \( S'(a) = s_3, S''(b) = 0 \) and identification of sweep coefficients \( \alpha_k, \beta_k \) by formulas (9), (15) and (16). The first values \( \alpha_1, \beta_0 \) are calculated for \( \alpha_1 = 0, \beta_0 = 0 \).

Step 2. Calculation of coefficients \( c_k \) in polynomials (1) using the recursion formula (13), while the value \( c_n \) is taken equal to zero.

Step 3. Identification of the coefficient \( b_1 \) by the formula (23). The rest of the coefficients for \( b_k \) are calculated by formula (5).

Step 4. Calculation of \( c_n \) according to formula (26).

Step 5. Construction of a spline under conditions (20) for \( s_2 = c_n \).

Step 6. Identification of the value of the spline at a given point belonging to the interval \([a, b]\) using formula (1).

The solution of a number of problems with other boundary conditions can be obtained using previously considered methods. Let’s dwell on a few examples. Let the conditions be given that are mirror-like with respect to formulas (20), \( S'(b) = s_3, S''(a) = s_2 \).

In this case, it is sufficient to consider the problem in the interval \([a, b]\). In applications, there is a situation when the beginning of the spline is closed at its end using the conditions \( S(a) = S(b) = y_0, S'(a) = S'(b) = s_3. \)

A similar problem is solved using the algorithm related to the third case.

3.3. Discussion of results

The relative simplicity of cubic splines construction made them an effective tool for solving various problems. There are two main directions of using splines. The first one is associated with the interpolation of functional dependencies, especially in the applied mathematics and computing. In this group of tasks there exist high requirements for the accuracy of calculations. The second group includes tasks where the most important things are necessary smoothness of lines, identification of the direction of the process development, presentation of the results of processing the experimental data in the form of approximate dependencies (approximation problems), etc.

The “natural spline” with boundary conditions (17) at \( s_1 = 0, s_2 = 0 \) is well suited for interpolating functions. In the case of continuous functions with continuous derivatives of the first and second order, it provides high accuracy. It is known that the relative error of interpolation of functions with the same step \( \delta x \) is about \( \Delta_{\text{Theory}} = 5/(2\delta x^3 f''_{\text{max}}) \), where \( f''_{\text{max}} \) is the maximum value of the third derivative in the interpolation interval \([a, b]\). Table 2 shows the data on the theoretical accuracy of the \( \Delta_{\text{Theory}} \) method and the accuracy of \( \Delta_{\text{Test}} \), obtained during testing on the example of interpolating the function \( f(x) = 10x \sin(x)e^{-x} \) depending on
Table 2. Dependence of the theoretical accuracy $\Delta_{\text{Theory}}$ and the accuracy of interpolation obtained in test experiments with the function $10x \sin(x) e^{-x}$ in the interval $[0, 1]$.

| $N$, number of steps | 10   | 100  | 1000 |
|----------------------|------|------|------|
| $\Delta x$          | $10^{-1}$ | $10^{-2}$ | $10^{-3}$ |
| $\Delta_{\text{Theory}}$ | $5 \cdot 10^{-2}$ | $5 \cdot 10^{-5}$ | $5 \cdot 10^{-8}$ |
| $\Delta_{\text{Test}}$ | $3 \cdot 10^{-3}$ | $3 \cdot 10^{-5}$ | $3 \cdot 10^{-8}$ |

the step $\delta x$ in the interval $[0, 1]$. The maximum $f'''_{\text{max}}$ for this function is approximately equal to 56.

High accuracy of interpolation, confirmed by the data shown in table 1 and table 2, is not a compulsory condition for using splines in many application tasks. In some of them, it is more important to ensure the required curvature of the line [22] or the fulfillment of special conditions at the borders. The results, obtained in this work, do not exhaust all cases of the possibility of constructing splines with different boundary conditions. It can be seen from formulas (25), (26) that the fulfillment of condition (24) presupposes the replacement of the third degree polynomial by a parabola at the last step. A solution of such a problem can be obtained in another way, for example, by replacing it with a fourth-degree polynomial of the following type

$$f_n(x) = a_n + b_n(x-x_n) + c_n(x-x_n)^2 + d_n(x-x_n)^3 + e_n(x-x_n)^4, x_{n-1} \leq x \leq x_n.$$  (27)

Let’s emphasize one important circumstance. The fulfillment of the boundary conditions (20) or (24), when using the proposed algorithms, is reflected only in the first and last intervals adjacent to the boundaries of the interval $[0, \pi]$. Table 3 shows the results of calculations of $\text{Spline}_2(x)$ with modified boundary conditions and the “natural spline” $\text{Spline}_1(x)$ on the example of the function $10 \sin x$ for the interval $[0, \pi]$.

Table 3. Results of calculations based on $\text{Spline}_2(x)$ with modified boundary conditions and the “natural spline” $\text{Spline}_1(x)$ for the function $10 \sin x$ in the interval $[0, \pi]$.

| $i$, number of the node | Number of the point between the nodes | $x$ | $10 \sin x$ | $\text{Spline}_1(x)$ | $\text{Spline}_2(x)$ |
|------------------------|--------------------------------------|-----|-------------|----------------------|---------------------|
| 3(99)                  | 3.09761                              | 0.43968 | 0.43965 | 0.43965              |
| 4(99)                  | 3.10389                              | 0.37690 | 0.37688 | 0.37688              |
| 99                     | 5(99)                                | 3.11018 | 0.31411 | 0.31419 | 0.31419 |
|                        | 1(100)                               | 3.11646 | 0.25130 | 0.25134 | 0.24730 |
|                        | 2(100)                               | 3.12274 | 0.18848 | 0.18849 | 0.16438 |
|                        | 3(100)                               | 3.12903 | 0.12566 | 0.12566 | 0.07139 |
|                        | 4(100)                               | 3.13531 | 0.06283 | 0.06283 | -0.00150 |
| 100                    | 5(100)                               | 3.14159 | -0.00000 | -0.00000 | -0.00000 |

The information presented in Table 3, reflects the results of calculations for $n = 100$ at nodal points numbered 99, 100 and at several points between nodes with indices 3(99), 4(99), ..., 4(100), 5(100). In the case of the “natural spline” $\text{Spline}_1(x)$, there were no restrictions on the function at the boundaries of the interval $[0, \pi]$, while the first derivate at the nodes $x_0 = 0$ and...
$x_{100} = 0$ turned out to be equal to 10 and $-10$ respectively. It can be seen from table 3 that the interpolation results differ from the exact value of the $\sin(x)$ function in the fifth decimal place.

Spline2($x$) meets boundary conditions (24) for $s_3 = 10, s_4 = 10$. In this case, all the formulas used to construct the spline Spline1($x$), except for the last polynomial, which was represented by formula (27), have been preserved. Table 3 shows that the calculation results were reflected only for those values of $x$ with indices 1(100), 2(100), ..., 5(100), which were located in the half-interval $[x_99, x_{100}]$ after the prelist node with number 99. All values of the spline Spline2($x$) to the left of the $x$ coordinate remained unchanged. Similar results were obtained for a spline with modified boundary conditions at zero node $x_0$. Replacing the boundary condition $s_3 = 10$, corresponding to the “natural spline”, by any other value led to a change in the calculated data only in the half-interval $[x_0, x_1]$.

The data obtained can be used while training specialists in the field of applied mathematics and computing. At the first stage, it is recommended for students to study the dependence of the accuracy of extrapolation of a known function on the number of nodal points for a fixed interval of its variation within the interpolation interval $[a, b]$. At the second stage, it is desirable to investigate the dependence of the accuracy on the form of the function, in particular, when using oscillating functions. At the third stage, it is useful to offer students to construct approximating type splines, which are often used to process the results of experimental studies.

4. Conclusions
The algorithms for constructing cubic splines with various boundary conditions have been developed:

1. The second derivative takes arbitrary values at the ends of the spline. A special case is the “natural spline”, when the second derivative at spline ends is equal to zero.
2. At one end of the spline, the first derivative is given, and at the opposite end there is the second derivative.
3. The first derivative is given at two ends of the spline without any restrictions on the second derivatives.

It is shown that a number of problems with other boundary conditions can be solved using the proposed methods. The considered algorithms are recommended to use while interpolating functional dependencies and processing the results of experimental studies, represented as a discrete set of pairs of numbers.

References
[1] Schoenberg I J 1988 Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions (Boston, MA: Birkhäuser Boston) pp 3–57
[2] Stechkin S B and Subbotin Y N 1976 Splines in computational mathematics (Moscow: Nauka)
[3] Zav’yalova Y S, Kvasov B I and Miroshnichenko V L 1980 Spline function methods (Moscow: Nauka)
[4] Voskoboynikov Y E, Preobrazhensky N G and Sedelnikov A I 1984 Mathematical processing of the experiment in molecular gas dynamics (Novosibirsk: Nauka)
[5] Korneichuk N P 1984 Splines in approximation theory (Moscow: Nauka)
[6] Kalitkin N N and Kuzmina L V 1995 Mathematical modeling 7 77–94
[7] Malozemov V N 1998 Computational mathematics and mathematical physics 38 1181–1192
[8] Zadorozhny A G and Kiselev D S 2019 Building splines using the OpenGL library (Novosibirsk: Publishing house of NSTU)
[9] Volkov Y S and Subbotin Y N 2014 Trudy Instituta Matematiki i Mekhaniki UrO RAN 20 52–67
[10] Volkov E A 1987 Numerical Methods 2nd ed (Moscow: Nauka)
[11] Samarskii A A and Gulin A B 2003 Numerical Methods of Mathematical Physics (Moscow: Nauka)
[12] Volkov Y S, Bogdanov V V, Miroshnichenko V L and Salienko A E 2016 Calculation of the parameters of a model hydraulic turbine by the value of the speed coefficient Intellectualization of information processing IOI-2016 pp 192–193
[13] Struchenkov V I 2018 Russian technological journal 6 40–52
[14] Degtyarev V M 2010 Engineering and computer graphics (Moscow: Academy)
[15] Chemerys H, Osadchyi V, Osadcha K and Kruhlyk V 2019 CEUR Workshop Proceedings 2393 17–28
[16] Osadchyi V, Chemerys H, Osadcha K, Kruhlyk V, Koniukhov S and Kiv A 2020 CEUR Workshop Proceedings 2731 328–340
[17] Miroshnichenko V L and Selyunin D O 2009 About modern methods of approximation of experimental data Mat. X Int. symp. Application of MAES analyzers in industry (Novosibirsk) p 22
[18] Yeremeev V S and Rakovich G M 2012 Theory of planning and processing experiments (Melitopol: Bogdan Khmelnitsky Melitopol State Pedagogical University)
[19] Bilousova L, Kolgatin O and Kolgatina L 2019 CEUR Workshop Proceedings 2393 880–894
[20] Vlasenko M 2019 Bank Bulletin 31–36
[21] Shevaldin V T 2019 Proceedings of the Institute of Mathematics and Mechanics, Ural Branch of the Russian Academy of Sciences 25 279
[22] Gavrilenko E A, Gavrilenko E A, Holodnyak Y V, Holodnyak Y V, Antonova G V, Antonova G V, Chaplinsky A P and Chaplinsky A P 2020 Pratsi Taurysko ho derzhavnoho ahroteknolohichnoho universytetu 3 293–303