**Creating a Bivariate Spline Function Model and Applying it to Real Data**

Bunyod Azimov1, a), Boburjon Ahmadaliev1,b) and Mukhriddin Abduganiev1,c)

1*Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Tashkent 100084, Uzbekistan*

*a)* [*bunyodbekazimov@mail.ru*](mailto:bunyodbekazimov@mail.ru) *b) Corresponding author:* [*ahmadaliyevbobur0717@gmail.com*](mailto:ahmadaliyevbobur0717@gmail.com) *c)* [*mr\_muhriddin\_20@mail.ru*](mailto:mr_muhriddin_20@mail.ru)

**Abstract.** This article considers the construction of a bivariate spline function model and its application to real data. In this work, the construction of a bivariate spline model was first considered. In this work, a bivariate bicubic spline model was first constructed and interpolated based on the values of trigonometric and Gaussian functions. The high accuracy of the interpolation is shown in Figures 1 and 2. Considering the obtained results, a spline model was adopted to calculate the temperature field variations (printed circuit boards), and the error of the spline model was evaluated based on the interpolation work. The error results show that the accuracy of this bivariate spline model is very good, which can be used to analyze the temperature fields at points and predict the failure of devices. This approach can be widely applied to the process of optimizing the heat balance in microelectronic systems, industrial equipment and Internet of Things platforms.

**Keywords**: bivariate spline function, function error, trigonometric and Gaussian functions, printer circuit board, polyonimi

**INTRODUCTION**

Piecewise polynomials and classical polynomials are used in most signal reconstruction and signal interpolation applications. While classical polynomials define only one polynomial (over the entire interval), piecewise polynomials have separate polynomials (over each subinterval). Cubic splines are the most popular type of piecewise polynomial, and are defined as a degree 3 polynomial over each interval. Two-dimensional splines are applied to a two-dimensional space, where they divide the space into smaller regions of a group of rectangles or triangles, each region of which may have a unique law polynomial.Based on these polynomials, the interpolation process is performed in a given area. From this study, it is known that if the form of the function y = F(x) is not known exactly, but the values of the function satisfying it are known in tabular form, the process of replacing the function F(x) with the function G(x) corresponding to these other values is called interpolation. If G(x) is a polynomial, this process is called polynomial interpolation.. Polynomials can be used in different ways in different problems. For example, in cases where accuracy is required, low error of the polynomial is relevant, and in cases where speed is required, the speed of calculating the values of the polynomial is relevant. In general, problems with accuracy are often encountered in real life. Spline functions are usually used to solve these difficulties, and they are generated by many piecewise polynomials, which give smooth and flexible functions on each interval. In such processes, the reconstruction of functions of two variables using bicubic-spline interpolation is considered very effective. When constructing cubic spline functions, at least 4 points are usually used on each interval (this ensures that the interpolating functions are continuously differentiable on the entire interval, and also ensures that the second-order derivative is also continuous). The smoothness of these conditions makes it one of the most popular interpolation splines, namely cubic spline interpolation. The requirement in the two-dimensional construction of a cubic spline is that the spline, as well as all its partial derivatives, must be continuous on the sides of any grid cell. These conditions make splines successful in modeling smooth and precise surfaces. Consequently, cubic splines are very common in smooth and precise approximation of spatial data, particularly in signal reconstruction, image processing, geostatistics and computer graphics [1, 2].

**LITERATURE REVIEW**

Interpolation of bivariate cubic splines is widely used in fields such as image processing and signal processing where high accuracy and precision are required. At the same time, in a three-chapter fieldwork, D. Ramachandran and Dr. V.M. Mallaya [3] analyzed the mathematical properties of bivariate cubic spline interpolations and showed that the method can be reconstructed using various geometric scientific foundations. In the work, the authors derived formulas, according to which the given formulas were determined as the basis for creating a mathematical model of the temperature surface. B. Azimov and co-authors [4] restored the proposed algorithms for processing seismic signals in personal computer signal systems, and the basis of the document is the effectiveness of interpolation algorithms in real-time and distributed systems. It is possible to transmit temperature data from the board in real-time modeling processes through the laboratory. H. Zaynidinov et al. [7] have provided information on how to optimally perform spline interpolation in a multi-core environment. This problem plays an important role in quickly and efficiently reconstructing the temperature of many points, especially on large-sized boards [10], proposed a method of linear bicubic spline interpolation for processing geophysical signals. Using this method, the authors practically proved that the interpolation has high accuracy, increasing the recovery of values in unclear or interrupted parts of the signal. In general, the reviewed scientific literature reveals a wide range of applications of spline interpolation, the mathematical foundations of the algorithm, and its advantages in a practical environment. The methodologies used in the laboratory served as a theoretical and practical basis for our work in reliably reconstructing the temperature surface of the board.

**METHODOLOGY**

**The aim of our study was to reconstruct the surface of real data using the bivariate bicubic spline interpolation method. The research activities include the following stages:**

**1. Development of a mathematical model:**

**The development of bivariate natural cubic splines is given in the theoretical component of the study. To obtain a smooth surface on each of the mesh segments, the function values at the mesh nodes, the first and second derivatives, and the mixed derivatives are used to determine the model. The continuity and smoothness conditions of the spline functions are given here. The following formulas and procedures were implemented as a mathematical basis:**

**a) General formula for a cubic spline in two variables:**

=a(x- (1)

**b) Continuity and smoothness conditions for nodes and mixed derivatives:**

(2)

**c) In the presented case, the determination of the spline coefficients is based on the values of the x = 0, 0,1,0,2,...,10 and y = 0, 0,1,0,2,...,10 in the interval x = [0,10] and y = [0, 10].**

**We compared some of these processes with the spline model on the surfaces of the Trigonometric:**  **and Gaussian functions.**

**2. Real data application:**

**The interpolation results were predicted and a spline model was created and evaluated using the values of the heat distribution on the surface of the printed circuit board.**

**3. Calculation and visualization:**

**A two-cubic spline model was built in the Python environment and the results were discussed in tabular and graphical form.**

**BUILDING A TWO-VARIABLE SPLINE FUNCTION MODEL**

A cubic spline in two variables is a unique function that corresponds to the values of in every rectangular lattice for all and. Since s a cubic spline in two variables, all its partial derivatives of order must be linear and continuous. Here we consider the second derivative with respect to the variables x [1, 3]. It will be as follows:

(3)

let:

The process of integrating (1) with respect to x,

(4)

Integrating (4) with respect to x,

(5)

Since the spline interpolates at the knots:

Applying the conditions a, b, c, d to (3):

(6)

(7)

(8)

(9)

Solving (4), (5), (6), (7) we get the following constants

So from (3) the two variable cubic spline is

(10)

In two variable spline function there exist a unique tangent plane at the two surfaces in every node. So corresponding to the node we have =

(11)

(12)

Equating (11) and (12)

(13)

Assume: where i = 1,2,…,(n−1) and j=1,2,…,(m−1)

This is the first differentiated value with respect to x at the grid point (i,j), and the interval length is

(14)

Assume: where, i=1,2,…,(n−2) and j=1,2,…,(m−2).

(15)

Therefore equation (13) will become

(16)

Dividing equation (12) by we get

where i = 1, 2, ..., (n-2) and j = 1, 2, ..., (m-2) (17)

So the two variable natural cubic spline is

where i = 1, 2, ..., (n-2) and j = 1, 2, ..., (m-2) [3, 4].

For natural spline:

**EXPERIMENTAL INVESTIGATION**

Consider the two variable function The following table gives the function values x = [0,1, ... ,10] and y= [0,1, ... ,10].

We choose at the central point.

**TABLE 1.** Trigonometric function values in the interval [10].

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *\* | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 0 | 50.0 | 55.88 | 59.51 | 59.51 | 55.88 | 50.0 | 44.12 | 40.49 | 40.49 | 44.12 | 50.0 |
| 1 | 50.0 | 54.76 | 57.69 | 57.69 | 54.76 | 50.0 | 45.24 | 42.31 | 42.31 | 45.24 | 50.0 |
| 2 | 50.0 | 51.82 | 52.94 | 52.94 | 51.82 | 50.0 | 48.18 | 47.06 | 47.06 | 48.18 | 50.0 |
| 3 | 50.0 | 48.18 | 47.06 | 47.06 | 48.18 | 50.0 | 51.82 | 52.94 | 52.94 | 51.82 | 50.0 |
| 4 | 50.0 | 45.24 | 42.31 | 42.31 | 45.24 | 50.0 | 54.76 | 57.69 | 57.69 | 54.76 | 50.0 |
| 5 | 50.0 | 44.12 | 40.49 | 40.49 | 44.12 | 50.0 | 55.88 | 59.51 | 59.51 | 55.88 | 50.0 |
| 6 | 50.0 | 45.24 | 42.31 | 42.31 | 45.24 | 50.0 | 54.76 | 57.69 | 57.69 | 54.76 | 50.0 |
| 7 | 50.0 | 48.18 | 47.06 | 47.06 | 48.18 | 50.0 | 51.82 | 52.94 | 52.94 | 51.82 | 50.0 |
| 8 | 50.0 | 51.82 | 52.94 | 52.94 | 51.82 | 50.0 | 48.18 | 47.06 | 47.06 | 48.18 | 50.0 |
| 9 | 50.0 | 54.76 | 57.69 | 57.69 | 54.76 | 50.0 | 45.24 | 42.31 | 42.31 | 45.24 | 50.0 |
| 10 | 50.0 | 55.88 | 59.51 | 59.51 | 55.88 | 50.0 | 44.12 | 40.49 | 40.49 | 44.12 | 50.0 |

Interval length: ==1.0, ==1.0, ==1.0, ==1.0

Using (14) and (15):

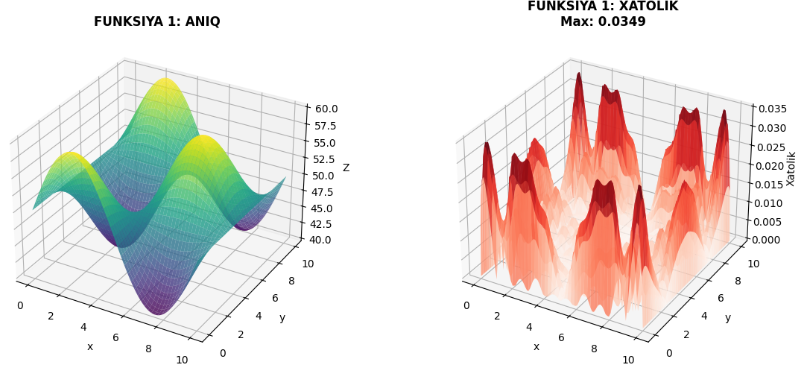
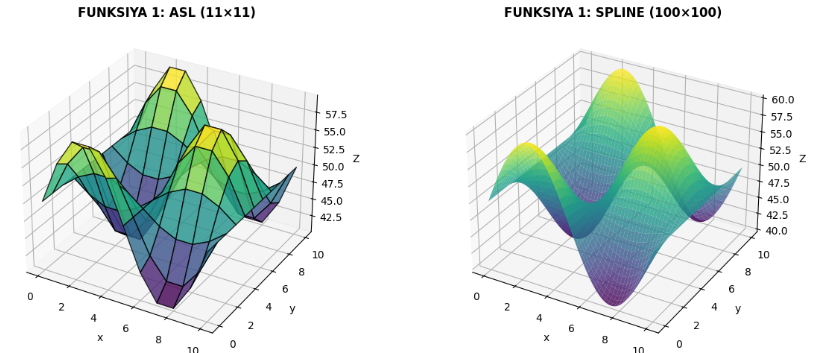
Using (17):

(18)

For a natural cubic spline

Solving (18) = 0 and   
Using (10) the two variable cubic splines are

Based on this rule, our result based on looks like this.



A B C

**FIGURE 1.** A-function recovery, B-interpolation result, C-error graph for a trigonometric function.

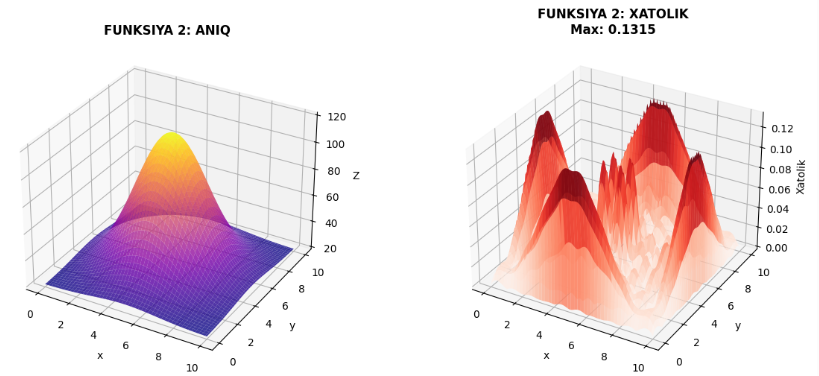
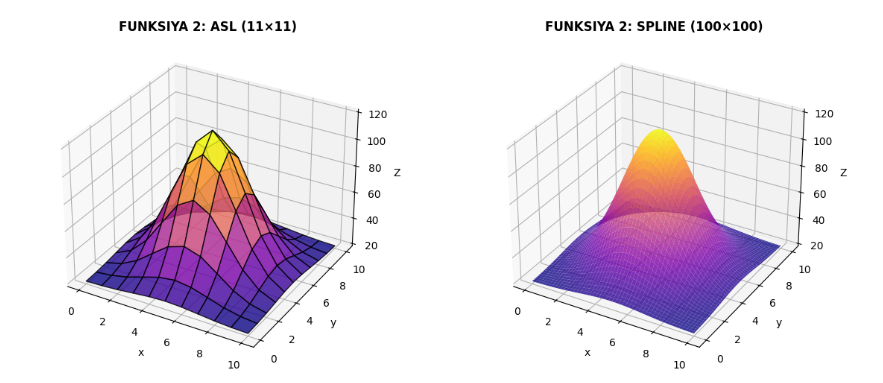
The following table lists the values of the trigonometric function at the intermediate points [0.500, 9.050] and the corresponding values obtained from interpolation of the bicubic spline function [5, 6. 7].

**TABLE 2**. Function at points x\_i and y\_j, interpolation values, and their errors

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *№* |  |  | *F(x,y)* | *Sij(x,y)* | *Error* |
| 1 | 0.500 | 0.500 | 52.938926 | 52.972251 | 0.033325 |
| 2 | 0.950 | 0.950 | 54.648882 | 54.652648 | 0.003765 |
| 3 | 1.400 | 1.400 | 54.911436 | 54.894796 | 0.016640 |
| 4 | 1.850 | 1.850 | 53.644843 | 53.640004 | 0.004840 |
| 5 | 2.300 | 2.300 | 51.243449 | 51.246310 | 0.002860 |
| 6 | 2.750 | 2.750 | 48.454915 | 48.456087 | 0.001172 |
| 7 | 3.200 | 3.200 | 46.147434 | 46.149000 | 0.001566 |
| 8 | 3.650 | 3.650 | 45.039426 | 45.042885 | 0.003459 |
| 9 | 4.100 | 4.100 | 45.475865 | 45.476138 | 0.000273 |
| 10 | 4.550 | 4.550 | 47.320866 | 47.323299 | 0.002433 |
| 11 | 5.000 | 5.000 | 50.000000 | 50.000000 | 0.000000 |
| 12 | 5.450 | 5.450 | 52.679134 | 52.676701 | 0.002433 |
| 13 | 5.900 | 5.900 | 54.524135 | 54.523862 | 0.000273 |
| 14 | 6.350 | 6.350 | 54.960574 | 54.957115 | 0.003459 |
| 15 | 6.800 | 6.800 | 53.852566 | 53.851000 | 0.001566 |
| 16 | 7.250 | 7.250 | 51.545085 | 51.543913 | 0.001172 |
| 17 | 7.700 | 7.700 | 48.756551 | 48.753690 | 0.002860 |
| 18 | 8.150 | 8.150 | 46.355157 | 46.359996 | 0.004840 |
| 19 | 8.600 | 8.600 | 45.088564 | 45.105204 | 0.016640 |
| 20 | 9.050 | 9.050 | 45.351118 | 45.347352 | 0.003765 |
| M**ax error** | | | | | 0.033325 |

The first test function was chosen as a combination of the sine and cosine functions. this function has complex sinusoidal surface oscillations and was used to test the interpolation accuracy. It is based on this function that the interpolation errors in the table are determined and are presented in Table 2. The maximum error value is 0.033325, which indicates a high level of accuracy

In our second Gauss function the result obtained based on the above rules also looks like this.



A B C

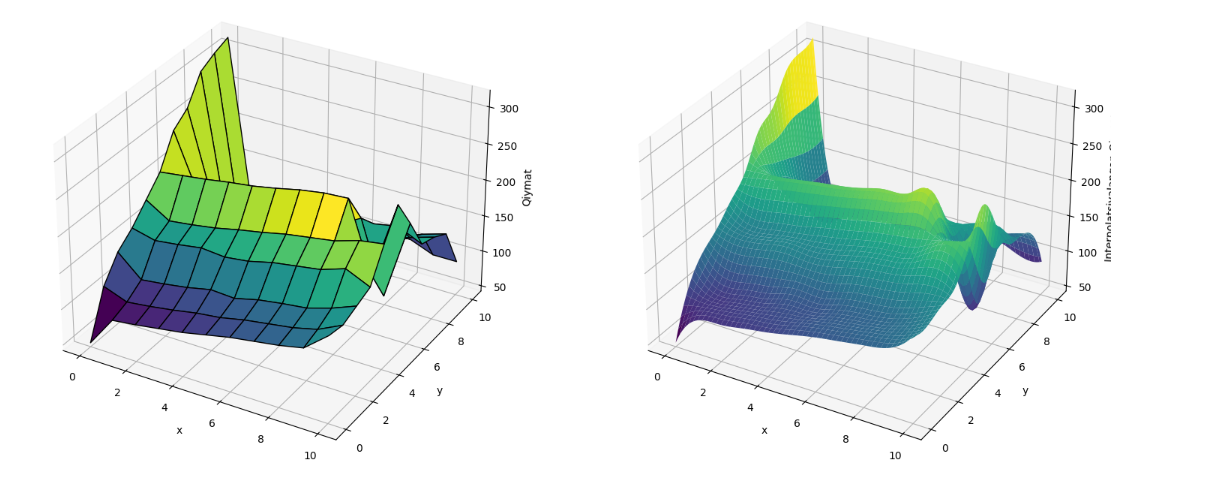
**FIGURE 2**. A-function recovery, B-interpolation recovery, C-error detection graph for the Gaussian function

Cubic splines allow the generation of smooth interpolating functions that are continuous and differentiable over the entire interval using four points in each interval [2, 3]. In this approach, the function itself, its first and second derivatives are guaranteed to be continuous on each slice of the interpolated surface. In particular, a surface of the form Z = f(x,y) was reconstructed using cubic spline interpolation in two variables. In this case, the data is arranged in the form of a grid, and the spline function is constructed in such a way that it satisfies the continuity conditions on each slice of the grid.

**TABLE 3.** Index of printed circuit board values obtained in the range [10]

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *\* | *0,0* | *1.0* | *2.0* | *3.0* | *4.0* | *5.0* | *6.0* | *7.0* | *8.0* | *9.0* | *10.0* |
| *0,0* | 48 | 90 | 94 | 99 | 104 | 111 | 118 | 123 | 128 | 135 | 162 |
| *1.0* | 110 | 98 | 102 | 107 | 113 | 117 | 124 | 129 | 134 | 142 | 158 |
| *2.0* | 141 | 112 | 116 | 121 | 128 | 126 | 134 | 139 | 143 | 153 | 166 |
| *3.0* | 158 | 150 | 154 | 159 | 156 | 160 | 168 | 173 | 178 | 189 | 174 |
| *4.0* | 181 | 161 | 165 | 170 | 177 | 183 | 190 | 195 | 200 | 210 | 215 |
| *5.0* | 204 | 208 | 212 | 217 | 221 | 227 | 234 | 239 | 241 | 246 | 251 |
| *6.0* | 246 | 170 | 161 | 165 | 175 | 180 | 185 | 180 | 198 | 210 | 210 |
| *7.0* | 262 | 175 | 153 | 158 | 161 | 170 | 165 | 165 | 174 | 180 | 165 |
| *8.0* | 298 | 140 | 115 | 124 | 130 | 124 | 130 | 128 | 137 | 148 | 158 |
| *9.0* | 309 | 120 | 96 | 100 | 110 | 115 | 118 | 115 | 123 | 136 | 146 |
| *10.0* | 317 | 115 | 93 | 92 | 90 | 100 | 90 | 100 | 103 | 90 | 90 |

We used a bivariate cubic spline to reconstruct and interpolate the original function based on the values obtained from different points on the printed circuit board. The values at the points on the printed circuit board are given in Table 3.



A B

**FIGURE 3**. A-function recovery B-interpolation graph for a printing device

Based on the values given in Table 3, we modeled the surface as a smooth surface using bivariate cubic splines. These splines were defined as 3rd degree polynomials on each mesh and ensured the continuity of the function and its partial derivatives at the boundaries. Since the interpolation process constructed a smooth function between certain points, the surface of the printed circuit board was accurately depicted. The resulting function was visualized as a 3D graph, which optimized the paths of the printing device [2, 8, 9].

**TABLE 4**. Printer. Function at points x\_i and y\_j, interpolation values, and their errors

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № |  |  | F(x,y) | Sij(x,y) | Error |
| 1 | 1.0 | 0.0 | 90.000 | 89.990 | 0.01048 |
| 2 | 5.0 | 0.0 | 111.000 | 110.992 | 0.00830 |
| 3 | 10.0 | 0.0 | 162.000 | 161.994 | 0.00614 |
| 4 | 0.0 | 1.0 | 110.000 | 110.014 | 0.01398 |
| 5 | 5.0 | 1.0 | 117.000 | 116.987 | 0.01338 |
| 6 | 10.0 | 1.0 | 158.000 | 158.006 | 0.00594 |
| 7 | 5.0 | 2.0 | 126.000 | 125.996 | 0.00382 |
| 8 | 10.0 | 2.0 | 166.000 | 165.998 | 0.00244 |
| 9 | 0.0 | 3.0 | 158.000 | 158.008 | 0.00838 |
| 10 | 5.0 | 3.0 | 160.000 | 160.013 | 0.01338 |
| 11 | 8.0 | 3.0 | 178.000 | 178.023 | 0.02343 |
| 12 | 1.0 | 4.0 | 161.000 | 161.002 | 0.00248 |
| 13 | 10.0 | 4.0 | 215.000 | 215.017 | 0.01700 |
| 14 | 1.0 | 5.0 | 208.000 | 208.007 | 0.00710 |
| 15 | 10.0 | 5.0 | 251.000 | 250.999 | 0.00064 |
| 16 | 0.0 | 6.0 | 246.000 | 246.010 | 0.01028 |
| 17 | 1.0 | 6.0 | 170.000 | 170.015 | 0.01537 |
| 18 | 9.0 | 6.0 | 210.000 | 210.006 | 0.00616 |
| 19 | 10.0 | 7.0 | 165.000 | 164.999 | 0.00111 |
| 20 | 0.0 | 8.0 | 298.000 | 298.010 | 0.00966 |
| 21 | 1.0 | 8.0 | 140.000 | 140.004 | 0.00352 |
| 22 | 0.0 | 9.0 | 309.000 | 309.003 | 0.00269 |
| 23 | 1.0 | 9.0 | 120.000 | 120.005 | 0.00485 |
| 24 | 1.0 | 10.0 | 115.000 | 115.005 | 0.00513 |
| 25 | 10.0 | 10.0 | 90.000 | 89.999 | 0.00071 |
| Max error: | | | | | 0.02343 |

Table 4 shows the values of the function F(x,y) at the points (, ) for the printer, the values (x,y) calculated by bivariate spline interpolation, and their errors. The errors shown in the table (maximum error 0.02343) are very small, which indicates the high accuracy of bivariate cubic spline interpolation. The spline function (x,y) gives results very close to the original function values F(x,y). This level of accuracy is very important for printers, since even small errors can affect the quality of the printed product. The smallness of the errors confirms that the spline model approximates the data smoothly and accurately [10, 11].

**RESULTS AND DISCUSSION**

Interpretation of the results

**I. Results of the functional surfaces:**

a) The greatest interpolation error on the surface of a trigonometric function did not rise above 0.033325.

b) The largest mistake made on the Gaussian function was 0.02343 it demonstrated the proper functioning of the model.

c) The grid segments and nodes using the spline model that are illustrated in the example worked well because of their accuracy.

**II. Printed circuit board:**

a) In the spline interpolation through the surface of the printed circuit board, the difference among the true values of the respective functions and the outputs of the interpolation was very low.

c) The maximum error was less than 0.02343 as demonstrated in table 4, and this enabled us to know which areas were thermally hotspots.

c) We can succeed in designing the cooling system with the help of spline model as it simulates the heat dissipation on the board smoothly and reliably.

The bivariate bicubic spline model fitted as part of the research had a high accuracy and reliability when it comes to recovering surface of real data and functions. The strengths of the model in the interpolation process were ascertained as the very small maximum error on complex in the shape of functional surfaces like trigonometric based and Gaussian which was less than 0.033325 and 0.02343 respectively. This, by its part, created opportunities of its utilization in the modeling of complicated physical processes and in many areas.

a) The developed model operated efficiently on actual data.

b) Each spline function was guaranteed of continuity on its first and second-order derivative, hence resulted in smoothness of the surface.

c) The approach can also be used in real-time thermal monitoring, microelectronics and IoT systems.

The continuity of first and second-order derivative of the bicubic spline function is there to restore the surfaces having smooth, deformation free, and physically real shape along the whole surface. This mathematical quality is what makes it possible to apply the model even in real life activities, as opposed to artificial ones. Consider, for instance, in the case of demonstrating the correct distribution of heat on the surface of a printed circuit board the possibility was available to predict hot spots in advance, to optimize the cooling circuit and hence garner the life span of the device. In addition, such a method as bicubic spline can also be characterized as an applicable solution to thermal monitoring, real-time analysis of microelectronics and industrial equipment. Nowadays, in technologies based on the Internet of Things (IoT), there is a collection of large quantities of data and its analysis in real-time; the offered spline model opens up the prospect of smooth, consistent, and accurate modeling of data. This makes it possible to use it not just as a monitoring, forecasting, controlling of various physical parameters of temperature, pressure, humidity, vibration etc.

As can be seen in the discussed results, because of its mathematical simplicity, the speed and great accuracy, the proposed bicubic spline model can be extensively applied as a relevant and efficient tool not only in the academic studies, but also in the industrial life. This forms one of the significant processes in the fusion of science and technology.

**CONCLUSION**

In the study, the two-dimensional field provides good results in the process of image processing, surface modeling and computational computation. In this study, the temperature of the printer board was interpolated as a two-variable cubic spline interpolation. Smooth and clear surfaces were restored, because the entire interpolation process cannot achieve a perfect representation of the real function. This technique allowed us to find the temperature of all points on the board and fully understand the heat dissipation. The construction and interpolation of two-variable two-cubic splines based on the values of trigonometric and Gaussian functions were pre-arranged. The interpolation accuracy was evaluated using the trigonometric function, and the error was found to be 0.033325. The temperature distribution occurring during the operation of the printer board was determined by the bivariate cubic spline model, and the hot spots were determined. The spline model was used to determine the changes in temperature fields (printing board), and based on the interpolation results, the spline function error did not exceed 0.02343, while the bicubic spline interpolation error of the printer did not exceed 0.02343. This result proves that spline interpolation has high accuracy and is suitable for use in practical problems. Based on the obtained interpolation surfaces, it is proposed to identify the necessary areas for more effective design of the cooling system and place cooling elements in these areas. This approach can be widely used in optimizing the thermal balance in microelectronics, industrial devices, and IoT platforms.

This paper has suggested a bivariate bicubic spline model and has shown that it can work with real data. They revealed the high perfection and usability of this model. Further working steps are seen to play out important results through testing of the method in physical processes (pressure, humidity), by incorporating it into real time monitoring systems, and in the future of development of an adaptive spline method on larger grids segments.

**REFERENCES**

1. S. Shen, H. Zhou, K. He, and L. Zhou, Principal component analysis of two-dimensional functional data with serial correlation, J. Agric. Biol. Environ. Stat. **29**(3), (2024). https://doi.org/10.1007/s13253-023-00585-8
2. A. Kroizer, T. Routtenberg, and Y. C. Eldar, Bayesian estimation of graph signals, IEEE Trans. Signal Process. **70**, (2022). https://doi.org/10.1109/TSP.2022.3159393
3. D. Ramachandran and V. M. Mallayya, Two variable cubic spline interpolation, IOSR J. Math. **13**(1), (2017). https://doi.org/10.9790/5728-1301060105
4. H. Zaynidinov, S. Bahromov, B. Azimov, and M. Kuchkarov, Lacol interpolation bicubic spline method in digital processing of geophysical signals, Adv. Sci. Technol. Eng. Syst. J. **6**(1), (2021). https://doi.org/10.25046/aj060153
5. H. Zaynidinov, J. Nurmurodov, and S. Qobilov, Digital signal and image processing using neural networks, in *Proc. Seminar on Information Systems Theory and Practice (ISTP 2023)* (IEEE, 2023). https://doi.org/10.1109/ISTP60767.2023.10426883
6. H. Zaynidinov, S. Makhmudjanov, F. Rajabov, and D. Singh, IoT-enabled mobile device for electrogastrography signal processing, in *Lecture Notes in Computer Science (LNCS)*, including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*, (Springer, 2021). https://doi.org/10.1007/978-3-030-68452-5\_36
7. H. Zaynidinov, S. Bakhromov, B. Azimov, and S. Makhmudjanov, Comparative analysis spline methods in digital processing of signals, Adv. Sci. Technol. Eng. Syst. J. **5**(6), (2020). https://doi.org/10.25046/aj0506180
8. Z. Xakimjon and K. Muslimjon, Modeling of geophysical signals based on the second-order local interpolation splay-function, in *Proc. Int. Conf. Information Science and Communications Technologies: Applications, Trends and Opportunities (ICISCT 2019)* (IEEE, 2019). https://doi.org/10.1109/ICISCT47635.2019.9011853
9. A. Kroizer, Y. C. Eldar, and T. Routtenberg, Modeling and recovery of graph signals and difference-based signals, in *Proc. IEEE Global Conf. Signal and Information Processing (GlobalSIP 2019)* (IEEE, 2019). https://doi.org/10.1109/GlobalSIP45357.2019.8969536
10. O. Mallaev, B. Azimov, K. Muslimjon, and A. Kamola, Algorithm for digital processing of seismic signals in distributed systems, in *Lecture Notes in Computer Science (LNCS)*, including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics* (Springer, 2023). https://doi.org/10.1007/978-3-031-27199-1\_60
11. H. Zaynidinov, B. Sayfiddin, A. Bunyod, and J. Umidjon, Parallel processing of signals in local spline methods, in *Proc. Int. Conf. Information Science and Communications Technologies: Applications, Trends and Opportunities (ICISCT 2021)* (IEEE, 2021). https://doi.org/10.1109/ICISCT52966.2021.9670409