УДК 004.94+519.65

РЕШЕНИЕ ЗАДАЧИ ИНТЕРПОЛЯЦИИ В МАТЕМАТИЧЕСКОМ ПАКЕТЕ MATHCAD

Калиев И.А., Суйундукова А.К.

Стерлитамакский филиал ФГБОУ ВО «Башкирский государственный университет»

E-mail: kalievia@mail.ru

Использование таких пакетов моделирования, как MathCAD, Matlab, Maple и т.д., позволяет решать очень широкий круг задач. В данной работе изучены возможности пакета MathCAD для решения различных аппроксимационных задач: нахождения значений функции с помощью глобальной и локальной интерполяции, задачи экстраполяции. Эта работа будет полезной для подготовки магистрантов направления «Математическое моделирование» во время самостоятельной работы.

Ключевые слова: глобальная и локальная интерполяция, экстраполяция, математическое моделирование, пакет MathCAD.

SOLVING THE PROBLEM OF INTERPOLATION IN THE MATHCAD MATHEMATICAL PACKAGE

Kaliev I.A., Suyundukova A.K.

The use of such modeling packages as MathCAD, Matlab, Maple, etc., allows solving a very wide range of tasks. In this paper, the capabilities of the MathCAD package for solving various approximation problems are studied: finding the values of a function by means of global and local interpolation, extrapolation problems. This work will be useful for preparing the undergraduates of the direction "Mathematical Modeling" during self-study.

Keywords: global and local interpolation, extrapolation, mathematical modeling, MathCAD package.

The main task of numerical interpolation is to find the values of a table-defined function at those points within a given interval, where it is not specified. You can calculate the desired value of the original function at any point.

The main problems of the production are considered:

1) the choice of the interpolation function;

2) estimation of interpolation error.

Special interpolation methods allow us to determine the desired value of a function without directly constructing an interpolation function. In principle, all interpolation methods, based on the use of polynomials as an interpolation function, give the same results, but with different costs [1, 2]. This is because the polynomial n containing $n + 1$ parameter that does not pass through all given $n + 1$ points is the only Taylor series into which the original differentiable function can be decomposed.

This is one of the main advantages of a polynomial, as an interpolation function. Therefore, the first problem of interpolation is often solved by choosing a polynomial as the interpolation function, although other functions can be used. Selecting the type of interpolation function is an important task, especially if you remember that you can perform any number of functions through specified points.

It should be noted that there is an obvious way of constructing an interpolation function: from the conditions of passing through all points a system of equations is compiled, from the solution of which its parameters are found. However, this path is far from effective, especially with a large number of points.

The simplest *problem of interpolation* is as follows. For given $n + 1$ points $x_i = x_0, x_1, \ldots, x_n$, which are called *interpolation nodes*, and the values at these points of some function $f(x_i) = y_0, y_1, \ldots, y_n$ construct a polynomial $\varphi(x)$ (*interpolation polynomial*) of degree n of the form

$$\varphi(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0, \qquad (1)$$

taking at the interpolation points x_i the same values of y_i as the function $f(x_i)$:

$$\varphi(x_0) = y_0, \varphi(x_1) = y_1, \ldots, \varphi(x_n) = y_n, \; i = 0,1,\ldots,n \qquad (2)$$

The simplest kind of *global interpolation* is *parabolic interpolation*, when, using the conditions described above (2), to find unknown $n + 1$ coefficients $a_0, a_1, \ldots, a_n$ of expression (1), we obtain a system of $n + 1$ equations:

$$\begin{cases} a_n x_0^n + a_{n-1} x_0^{n-1} + \ldots + a_1 x_0 + a_0 = y_0, \\ a_n x_1^n + a_{n-1} x_1^{n-1} + \ldots + a_1 x_1 + a_0 = y_1, \\ \cdot \quad \cdot \quad \cdot \quad \cdot \\ a_n x_n^n + a_{n-1} x_n^{n-1} + \ldots + a_1 x_n + a_0 = y_n \end{cases} . \qquad (3)$$

It can be shown that this system has a unique solution if there are no coincident nodes among the interpolation nodes, i.e. if $x_i \neq x_j$ at $i \neq j$. Solving this system, we find the coefficients of the interpolation polynomial (1). We note, however, that such a way of constructing an interpolation polynomial requires a considerable amount of computation, especially for a large number of nodes. There are simpler algorithms for constructing interpolation polynomials [3].

For *local* interpolation between different nodes, different polynomials of low degree are chosen. In the MathCAD environment there are tools for this: *linear interpolation* tools (function *linterp*) and *interpolation by spline* (function *interp*) – linear (*lspline*), parabolic (*pspline*) and cubic (*cspline*) [4].

If it is necessary to evaluate the values of the function at points that do not belong to the interval $[x_0, x_n]$, then we can use the function *predict*.

Numerical implementation in a package MathCAD [5, 6].

Assignment 1. Calculate the values of the given function $y_i = f(x_i)$ at the interpolation nodes $x_i = a + h\,i$, where $h = (b - a)/10$, $i = 0, 1, \ldots, 10$ on the interval $[a, b]$.

Assignment 2. Based on the calculated table (x_i, y_i), conduct parabolic interpolation. To find the coefficients of the required polynomial (1), it is necessary to compile a system of linear algebraic equations (3). The system of equations is solved mathematically, using the function *lsolve*. Construct a graph of the interpolation polynomial and mark the nodal points (x_i, y_i) on it.

Assignment 3. Perform a linear interpolation of the given function using the built-in interpolation function *linterp*. Construct a graph of the *linterp* function and mark the node points (x_i, y_i) on it.

Assignment 4. Perform spline interpolation using the *lspline*, *pspline*, *cspline*, and *interp* functions. Construct a graph of the *interp* function and mark the node points (x_i, y_i) on it.

Assignment 5. Calculate the values of the given function $y_i = f(x_i)$ at the points $x_i = a + i/10$, where $i = 0, 1, \ldots, 10\ (b - a)$, on the interval $[a, b]$. Using the predict function, perform the prediction (extrapolation) of the obtained data vector y_i at the next 10 points with the last 7 values of the function. Display graphically the available data, the predicted data and the true form of the function $f(x)$.

Consider the implementation of the required tasks in MathCAD for the function $f(x) = e^{\cos(x)} \cdot \cos(x^2)$.

Define the function, construct the node points and calculate the discrete values of the function in them (Figure 1):

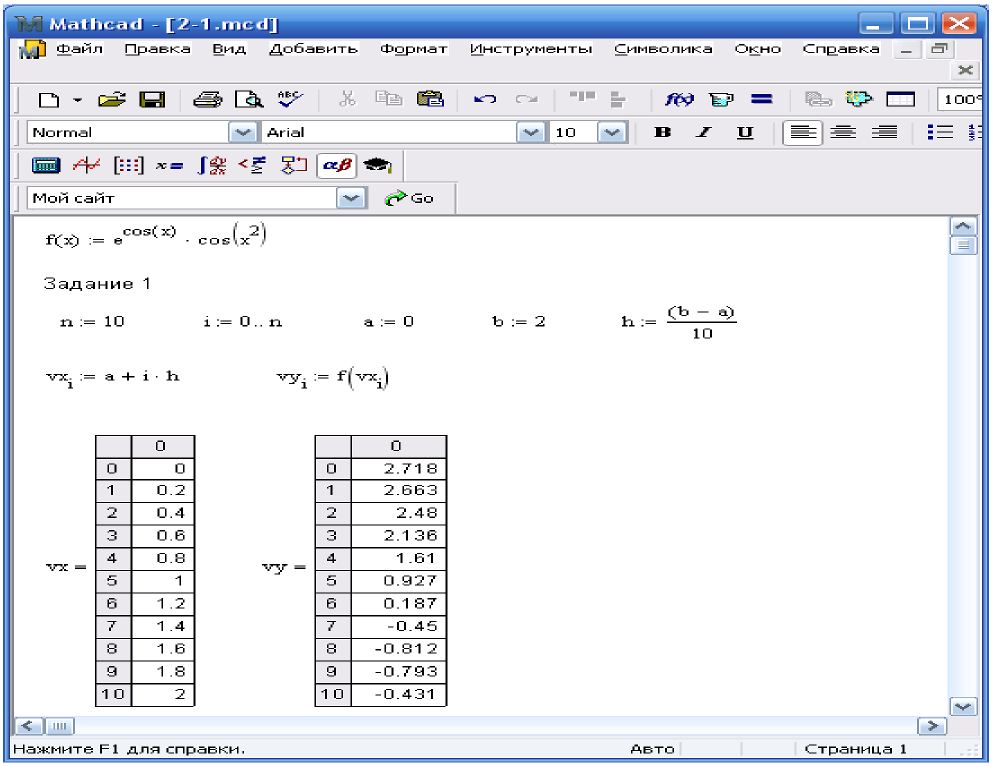

Figure 1. Calculations of the function in interpolation nodes

We perform parabolic interpolation. Solving a system of linear algebraic equations of the form (3) in a matrix way, using the *lsolve* function, we substitute the found coefficients in the desired polynomial (1).

We construct the graph of the interpolation polynomial and note the nodal points on it (x_i, y_i) (Figure 2).

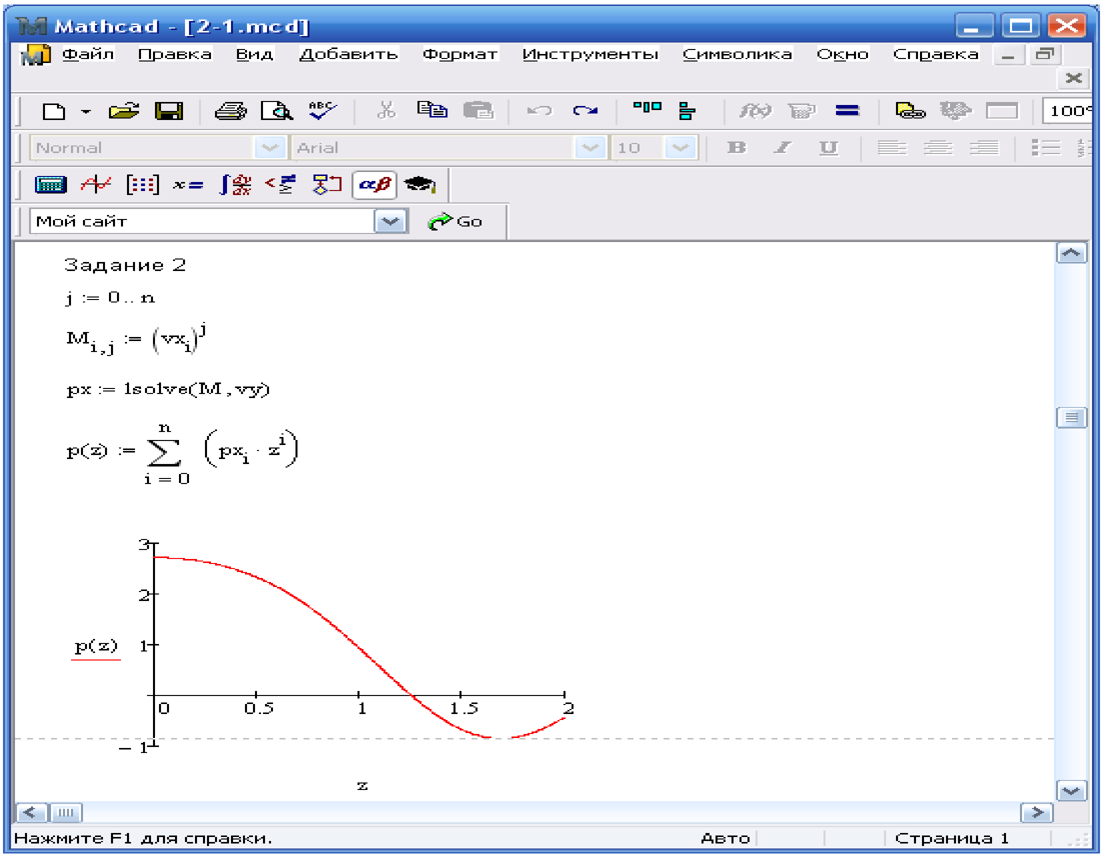

Figure 2. Parabolic interpolation

The simplest kind of interpolation is linear, which represents the required dependence in the form of a broken line. The interpolating function consists of straight line segments connecting points. To construct a linear interpolation, we use the function *linterp* (Figure 3).

In most applications it is desirable to connect the experimental points not by a broken line, but by a smooth curve. Interpolation with cubic splines, i.e. segments of cubic parabolas, is best for these purposes. This uses the built-in function *interp*. Before using the *interp* function, you must first determine the first of its arguments - the vector variable *vs*. This is done using one of the three built-in arguments functions *(vx, vy)*: *lspline*, *pspline*, *cspline* – linear, quadratic and cubic splines.

The choice of a specific function of spline coefficients affects the interpolation near the end points of the interval (Figure 4).

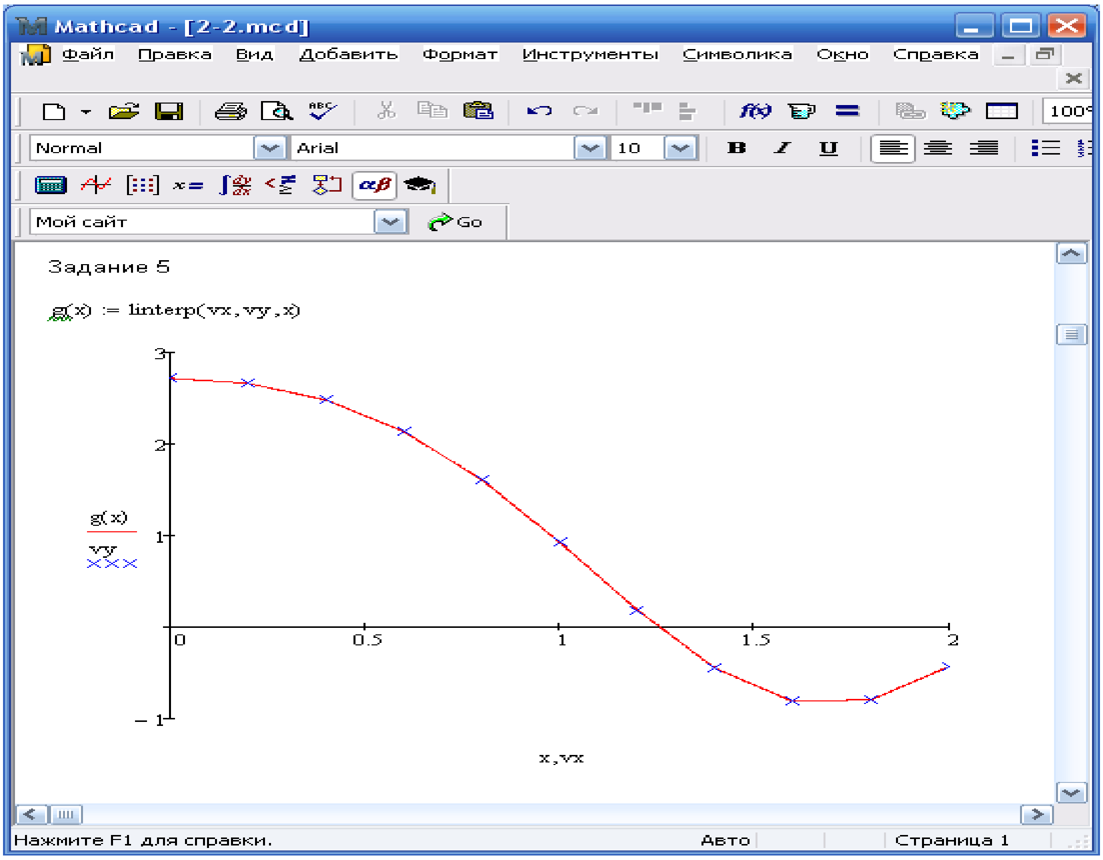

Figure 3. Linear interpolation

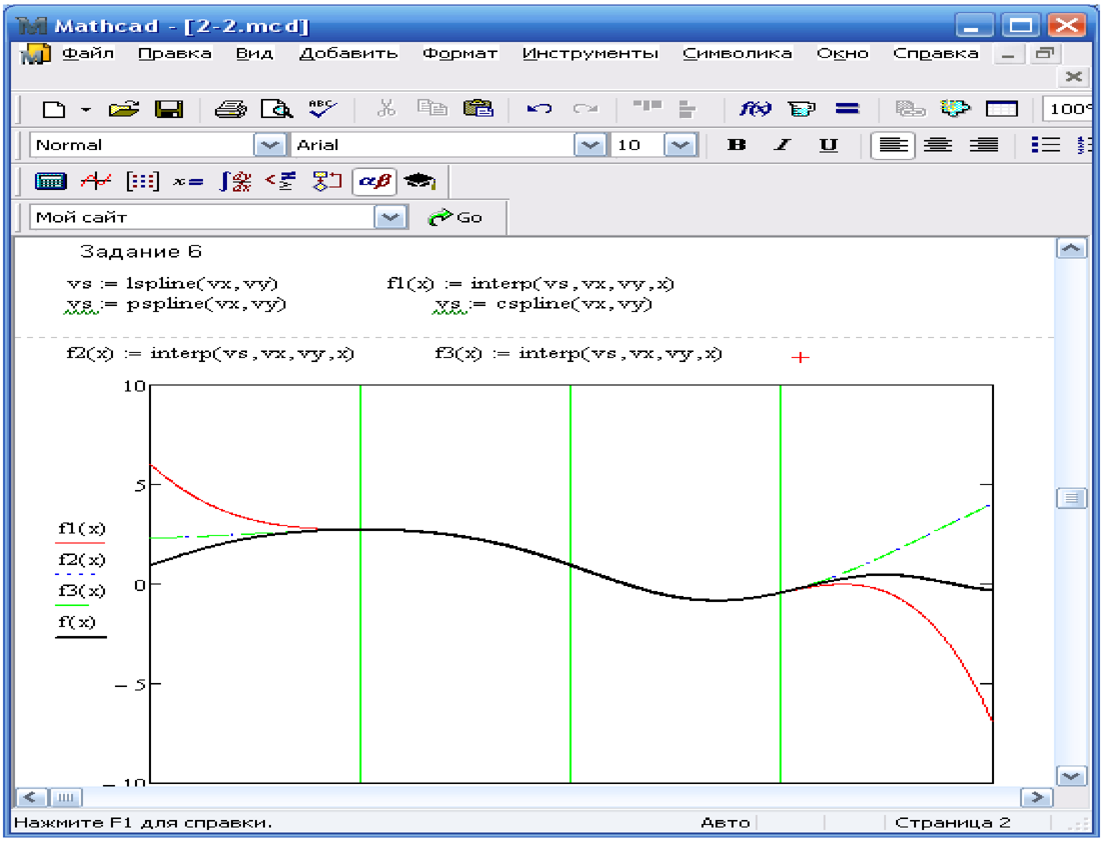

Figure 4. Spline interpolation

In MathCAD, there is another function that performs extrapolation taking into account the distribution of data along the whole interval – *predict*. In the *predict* function, a linear algorithm for predicting the behavior of a function is built-in, based on analysis, including oscillations.

Let us consider an example of using predict function to evaluate the value of a function at points not belonging to a segment [x_0, x_n], on the example of extrapolation of oscillating data y_i with a varying amplitude (see Figure 5).

The prediction function can be useful when extrapolating data over short distances. In addition, the *predict* function works well for data with a clearly observable pattern, mostly oscillating.

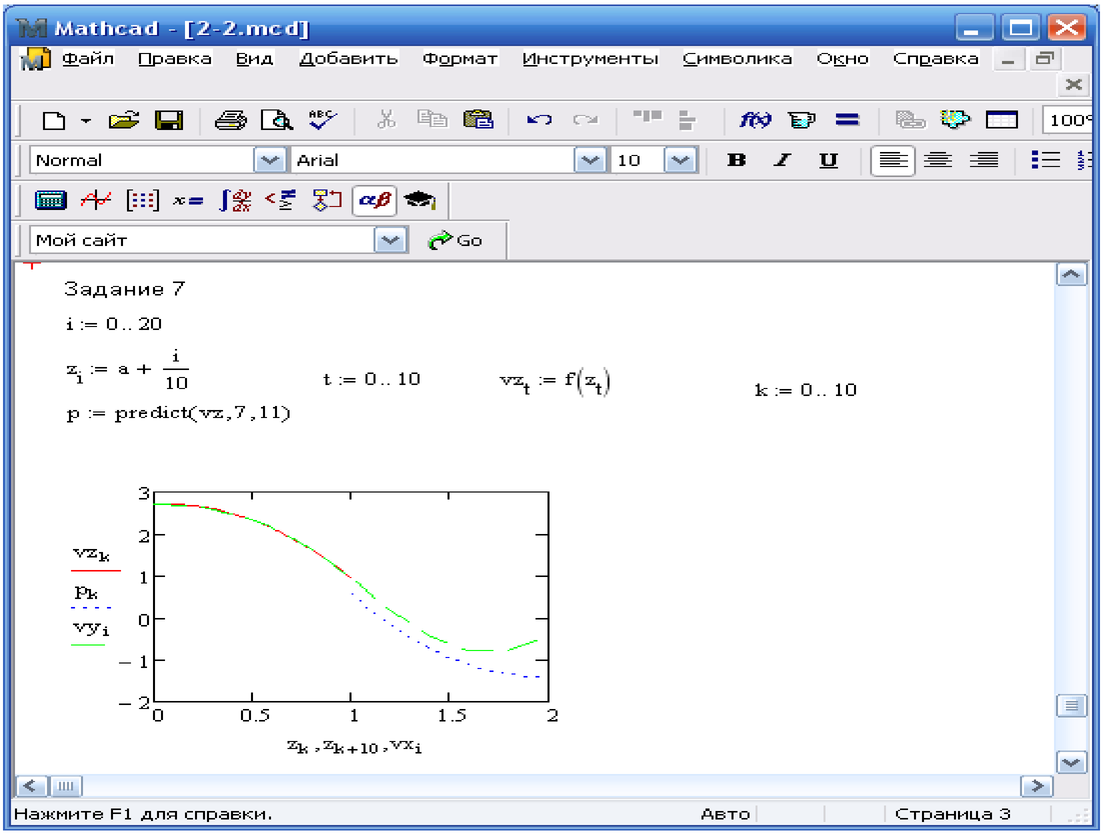

Figure 5. Function prediction

The interpolation problem is considered in the article, when the function has a complex analytical description that causes certain difficulties in its use, or is tabulated. The implementation of the problem using the built-in functions of the mathematical package MathCAD.

References

1. Kaliev I.A., Sabitova G.S. Practicum at the rate «Mathematical packages». // Navigator in the world of science and education. – M.: FGBNU IOO RAO. – 2011. – No. 8 (16). – P. 47-48.

2. Porshnev S.V., Belenkova I.V. Numerical methods based on MathCAD. – St. Petersburg: BVH-Petersburg. – 2005. – 456 p.

3. Sabitova G.S. Laboratory Workshop on Information Technologies in Mathematics: Textbook. allowance. Sterlitamak: SGPA. – 2008. – 216 p.

4. Sabitova G.S., Kaliev I.A. Workshop on Information Technology. Part 1: Training. allowance. – Sterlitamak: RIO of Sterlitamak branch of BashGU. – 2015. – 127 p.

5. Sabitova G.S., Kaliev I.A. Computer workshop on the discipline "Packages of mathematical modeling." // Chronicles of the united fund of electronic resources "Science and Education". – 2017. – № 08 (99). – P. 56.

6. Sabitova G.S., Myrzabekova A.M. Creation of the electronic manual "Packages of mathematical modeling". // Theory. Practice. Innovation. Electronic scientific journal. – 2018. – №1. – P. 48-53.