**interpol**

*Function of two-dimensional linear, three-dimensional linear, cubic spline-interpolation of argument x.*

**Syntax:**

*y* = **interpol**(*X, Y, х*);

*z* = **interpol**(*X, Y, Z, х, y*);

*y* = **interpol**(*x*, *M*);

**Arguments:**

*Х* – input array of coordinates of points along x axis for calculating linear interpolation,

*Y* – input array of coordinates of points along y axis for calculating linear interpolation,

*Z* – input array of coordinates of points along z axis for calculating linear interpolation,

*x, y* – arguments for interpolation,

*M* – matrix of coefficients of cubic spline-interpolation.

**Description:**

*interpol(x, M)* – function of cubic spline-interpolation of argument *x* to matrix of coefficients *M* calculated earlier by means of function *cspline*. Calculation is to be made according to formula:



*interpol(X,Y, х)* – function of two-dimensional linear interpolation of argument *x* to specified coordinates of points.

*interpol(X,Y,Z, х, y)* – function of three-dimensional linear interpolation of arguments *x, y* to specified coordinates of points.

Input arrays X, Y, Z shall be of the same dimension.

Input arrays X, Y, Z can be assigned:

* as variables of array type determined earlier:

*y* = **interpol**(*X, Y, х*);

* as variables of array type consisting of *double* variables determined earlier:

*y* = **interpol***(*[*x1,x2,x3,x4*]*,*[*y1,y2,y3,y4*]*, x*);

*y* = **interpol**([*x1,x2,x3,x4*], Y, x);

*y* = **interpol**(X, [*y1,y2,y3,y4*], x);

* as constant arrays:

*y* = **interpol**([-1.80, -1.60, -1.40, -1.20],*Y, x*);

*y* = **interpol**(*X*, [-1.80, -1.60, -1.40, -1.20], *x*);

*y* = **interpol**([-1.80, -1.60, -1.40, -1.20],[-1.40, -0.78,  
-0.53,-0.35], *x*);

**Result:**

*y* – result of interpolation of argument *x,*

*z* – result of interpolation of arguments *x, y.*

**Example 1:**

*Cubic spline-interpolation:*

|  |  |
| --- | --- |
|  | **input** Q; //input - argument  **output** n; //output – result of cubic spline-interpolation  //arrays of coordinates for calculating matrix  **const** Qpts =[-2.00, -1.80, -1.60, -1.40, -1.20, -1.00, -0.80, -0.60, -0.40, -0.20,  0.00,  0.20,  0.40,  0.60,  0.80,  1.00,  1.20,  1.40,  1.60,  1.80,  2.00];  **const** Hp=[-1.40, -0.78, -0.53, -0.35, -0.17, +0.05, +0.38, +0.80, +0.94, +1.07,  1.15,  1.26,  1.33,  1.40,  1.47,  1.40,  1.45,  1.51,  1.59,  1.69,  1.74];  Mn = **cspline**(Qpts, Hp);  n = **interpol**(Q, Mn); |

As a result, variable *n* will be assigned interpolated value of head (to arrays Hp and Qpts) according to value of consumption Q.

**Example 2**

*Two-dimension linear interpolation:*

|  |  |
| --- | --- |
|  | **input** Q; // input - argument  **output** n; // output – result of linear interpolation  // arrays of coordinates  **const** Qpts =[-2.00, -1.80, -1.60, -1.40, -1.20, -1.00, -0.80, -0.60, -0.40, -0.20,  0.00,  0.20,  0.40,  0.60,  0.80,  1.00,  1.20,  1.40,  1.60,  1.80,  2.00];  **const** Hp=[-1.40, -0.78, -0.53, -0.35, -0.17, +0.05, +0.38, +0.80, +0.94, +1.07,  1.15,  1.26,  1.33,  1.40,  1.47,  1.40,  1.45,  1.51,  1.59,  1.69,  1.74];  n = **interpol**(Qpts, Hp, Q); |

As a result, variable *n* will be assigned linearly-interpolated value of head (to arrays Hp and Qpts) according to value of consumption Q.