19 вопрос

Интерполяция, интерполирование (от лат. inter-polis – «разглаженный, подновлённый, обновлённый; преобразованный») – в вычислительной математике способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных

Если f совпадает с g в каком-то конечном числе точек, то f – интерполяция.

Пусть функция $f(x)$ задана набором точек (x_i, y_i) на интервале $[a,b]$:

$y_i = f(x_i)$, $i = 0,1,\dots,n$, $a \le x_i \le b$ (3.1)

Задача интерполяции – найти функцию $F(x)$, принимающую в точках x_i те же значения y_i.

Тогда, условие интерполяции: $F(x_i) = y_i$

При этом предполагается, что среди значений x_i нет одинаковых. Точки x_i называют **узлами интерполяции**.

Задача нахождения интерполяционной функции $F(x)$ имеет много решений, так как через заданные точки x_i, y_i можно провести бесконечно много кривых, каждая из которых будет графиком функции, для которой выполнены все условия интерполяции. Для практики важен случай интерполяции функции многочленами:

$F(x) = P_m(x_i) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \dots + a_m \cdot x^m$, $i = 0,1,\dots,m$

При этом искомый полином называется интерполяционным полиномом**.**

Локальная и глобальная интерполяция: Если функция $f(x)$ интерполируется на отрезке $[a,b]$ с помощью единого многочлена $P_m(x)$ для всего отрезка, то такую интерполяцию называют **глобальной**. В случае **локальной интерполяции** на каждом интервале $[x_i, x_{i+1}]$ строится отдельный интерполяционный полином невысокой степени.

Кусочно-линейная интерполяция: Простейшим и часто используемым видом локальной интерполяции является линейная (или кусочно-линейная) интерполяция. Она заключается в том, что узловые точки соединяются отрезками прямых (рис.3.1), то есть через каждые две точки (x_i, y_i) и (x_{i+1}, y_{i+1}) проводится прямая, то есть составляется полином первой степени:

$F(x) = a_0 + a_1 \cdot x$, при $x_{i-1} \le x \le x_i$

Кусочно-квадратичная интерполяция: В случае квадратичной интерполяции, для каждых трех узловых точек (x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}), строится уравнение параболы:

$F(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2$, при $x_{i-1} \le x \le x_{i+1}$

Многочлен Лагранжа:

Для решения задачи интерполяции функцию $y = f(x)$ заменяют многочленом $L_n(x)$ степени не выше n, который используют для приближенного вычисления значений функции. Многочлен полностью определяется требованием, чтобы его значения и значения $f(x)$ совпадали в узлах интерполяции:

$$f(x_0) = L_n(x_0),\ f(x_1) = L_n(x_1),\ \ldots,\ f(x_n) = L_n(x_n). \qquad (1)$$

Будем искать многочлен в виде

$$L_n(x) = \sum_{k=0}^{n} c_k \prod_{\substack{j=0 \\ j\neq k}}^{n} (x - x_j), \qquad (2)$$

Для решения задачи интерполяции необходимо определить коэффициенты $c_0, c_1, \ldots, c_n$ многочлена (2).

Положим в формуле (2) $x = x_i$ и используем условия (1). Получим

$$y_i = f(x_i) = L_n(x_i) = c_i \prod_{\substack{j=0 \\ j\neq i}}^{n} (x_i - x_j),$$

следовательно,

$$c_i = y_i / \prod_{\substack{j=0 \\ j\neq i}}^{n} (x_i - x_j).$$

Таким образом, искомый многочлен имеет вид

$$\boxed{L_n(x) = \sum_{k=0}^{n} y_k \prod_{\substack{j=0 \\ j\neq k}}^{n} \frac{x - x_j}{x_k - x_j}.} \qquad (3)$$

Он называется ***интерполяционным многочленом Лагранжа для неравноотстоящих узлов***. Это — единственный многочлен, решающий задачу интерполяции[1)].

Погрешность интерполяционной формулы Лагранжа:

$$\sigma = \frac{|L_n(x) - f(x)|}{f(x)} * 100\%$$

Код:

```
public class lagrang {
	public static void main(String[] args) {
		int n = 10;
		double [] X = new double[]  { 2, 5, -6, 7, 4, 3, 8, 9, 1, -2 };
		double [] Y = new double[]  { -1, 77, -297, 249, 33, 9, 389, 573, -3, -
21 };
		double res = lagranz(X, Y, 5);
		System.out.print("result = "+res);
	}

	static double lagranz(double [] X, double [] Y, double t) {

		int n = 10;
		double z, p1, p2;
		z = 0;
		for (int j = 0; j < n; j++){
			p1 = 1; p2 = 1;
			for (int i = 0; i < n; i++){
				if (i == j) {
					p1 = p1 * 1; p2 = p2 * 1;
				} else {
					p1 = p1 * (t - X[i]);
					p2 = p2 * (X[j] - X[i]);
				}
			}
			z = z + Y[j] * p1 / p2;
		}
		return z;
	}
}
```