БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Лабораторная работа №7

**Метод наименьших квадратов построения**

**элемента наилучшего приближения**

Выполнил:

Травничев Иван

2 курс 1 группа

Преподаватель:

Будник А. М.

Постановка задачи

Построить элемент наилучшего приближения степени n = 5 методом наименьших квадратов по равномерной сетке узлов для функции используя полиномиальную аппроксимацию, т. е. . Рассчитать истинную погрешность в точках:

Ход решения

1. Для определения коэффициентов полинома   
   составим систему:  
   где ,   
   Данную систему решим методом квадратного корня, т.к. матрица системы является симметрической.
2. Элемент наилучшего приближения определим по формуле:
3. Подсчитаем значения элемента наилучшего приближения в узлах
4. Для точек считаем истинную погрешность:

Листинг программы

**public static void main(String[] args){**

**//constructing equations for solving**

**equations = new double[POWER + 1][POWER + 2];**

**for (int i = 0; i <= POWER; i++){**

**for (int j = 0; j <= POWER; j++){**

**//calculating (x^i, x^j)**

**for (int k = 0; k < POINTS\_AMOUNT; k++){**

**equations[i][j] += WEIGHT \* Math.pow((BEGIN + STEP \* k), i + j);**

**}**

**}**

**//calculating (f, x^i)**

**for (int k = 0; k < POINTS\_AMOUNT; k++){**

**equations[i][POWER + 1] += WEIGHT \* f(BEGIN + STEP \* k) \* Math.pow(BEGIN + STEP \* k, i);**

**}**

**}**

**//discover coefficients and approximations in check points**

**double a[] = method\_sqrt(equations, POWER + 1);**

**double r[] = approximation(a);**

**//show the results**

**output(a, r);**

**}**

**static double[] approximation(double[] a){**

**double[] r = new double[3];**

**for (int i = 0; i < 3; i++){**

**r[i] = Math.abs(f(CHECK\_POINTS[i]) - q(CHECK\_POINTS[i], a));**

**}**

**return r;**

**}**

**//objective function**

**static double f(double x){**

**return (*KOEF* \* Math.*pow*(Math.*E*, x) + (1 - *KOEF*) \* Math.*sin*(x));**

**}**

**//approximating function**

**static double q(double x, double[] a){**

**double res = 0;**

**for (int i = 0; i <= *POWER*; i++){**

**res += Math.*pow*(x, i) \* a[i];**

**}**

**return res;**

**}**

**final static double KOEF = 1.9;**

**final static int POINTS\_AMOUNT = 11;**

**final static double BEGIN = 1;**

**final static double END = 2;**

**final static int POWER = 5;**

**final static double WEIGHT = 1;**

**final static double STEP = (END - BEGIN) / (POINTS\_AMOUNT - 1);**

**final static double[] CHECK\_POINTS = {**

**BEGIN + STEP \* 0 + STEP / 3.0,**

**BEGIN + STEP \* 5 + STEP / 3.0,**

**BEGIN + STEP \* 10 - STEP / 3.0**

**};**

Результаты

Элемент наилучшего приближение:

Таблица узлов:

|  |  |
| --- | --- |
|  |  |
| 1.0 | 4.40740946368801 |
| 1.1 | 4.905835577060976 |
| 1.2 | 5.4693830067066544 |
| 1.3 | 6.104456123368785 |
| 1.4 | 6.817976735068382 |
| 1.5 | 7.617469486199058 |
| 1.6 | 8.51114725662233 |
| 1.7 | 9.50799656076295 |
| 1.8 | 10.617862946704214 |
| 1.9 | 11.851536395283269 |
| 2.0 | 13.220836719186458 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| *значения* | 1.0333333333333334 | 1.5333333333333334 | 1.9666666666666666 |
|  | 4.566685511779873 | 7.904464707492094 | 12.748551479052693 |
|  | 4.566692414684976 | 7.904470034131815 | 12.748559049891274 |
|  | 6.902905102634804E-6 | 5.326639720415471E-6 | 7.570838580051031E-6 |

Вывод

Полученные показатели погрешности соответствуют степени построенного элемента наилучшего приближения (n=5). Однако система не ортогональна и при больших значениях *n* система плохо обусловлена. Можно обойти эту трудность, строя и используя многочлены, ортогональные с заданным весом на заданной системе точек. В нашем же случае(при n = 5) обусловленность задачи удволетворительна.