| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |
| --- | --- |

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

**ОТЧЕТ**

| **по лабораторной работе №** | 2 |
| --- | --- |



Построение и программная реализация алгоритма многомерной интерполяции табличных функций

**Дисциплина:** Вычислительные алгоритмы

| Студент | ИУ7И - 46Б |  |  | Андрич К. |
| --- | --- | --- | --- | --- |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | В.М. Градов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

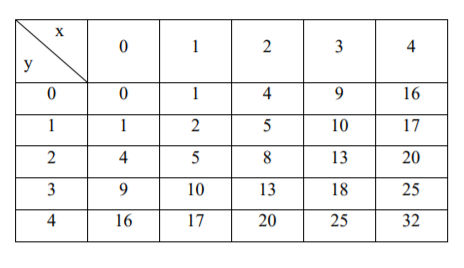
Москва, 2021.

Цель работы

Получение навыков построения алгоритма интерполяции таблично заданных функций двух переменных

Исходные данные

1. Таблица функции с количеством узлов 5x5



1. Степень аппроксимирующих полиномов - nx и ny
2. Значение аргументов x, y, для которого выполняется интерполяция

Описание алгоритма

1. Выбираем узли - nx + 1 x ось и ny + 1 для y ось. Строим таблицу разделенных разностей.
2. Делаем интерполацию по х. При значениях y[i], i = 0...ny выполняется ny+1 итерации и получаются значения функции f(x, y[i])
3. Как получим значения функции выполняется одна итерация

Код программы

У программе есть 5 файлов: 2 заголовочных файла (functions.h и errors.h) и 3 файла кода в СИ (main.c, in\_out.c и methods.c)

Файл: errors.h

| #ifndef ERRORS\_H #define ERRORS\_H  #define OK 0 #define ERR\_IO 1 #define ERR\_ARGS 2 #define ERR\_FILE 3  #endif //ERRORS\_H |
| --- |

Файл: functions.h

| #ifndef FUNCTIONS\_H #define FUNCTIONS\_H  #include <stdio.h> #include <stdlib.h>  #define N 6  typedef struct {  float x;  float y; }data\_t;  int read\_file(FILE \*f, float matrix[N][N]); void print\_matrix(float matrix[N][N]); int input\_values(float \*a, float \*b, float \*c, float \*d);  int comparator(const void \*q, const void \*p); void check\_beg\_end(int \*b, int \*e, int n); void get\_cur\_pos(float arr[N][N], float x, float y, int \*a, int \*b); void get\_point(float mtx[N][N], float tbl[N][N], float x\_arr[N], float y\_arr[N], int nx, int ny, float x, float y); int get\_pos(data\_t \*arr, float x, int len); float newton(data\_t \*arr, int len, float x, int n); float to\_data(data\_t \*arr, float x\_arr[N], float y\_arr[N], float tbl[N][N], float new[N], int nx, int ny, float x, float y);  #endif //FUNCTIONS\_H |
| --- |

Файл: in\_out.c

| #include "functions.h" #include "errors.h"  int read\_file(FILE \*f, float matrix[N][N]) {  for (int i = 0; i < N; i++)  for (int j = 0; j < N; j++)  if (fscanf(f, "%f", &matrix[i][j]) != 1)  return ERR\_IO;  return OK; }  void print\_matrix(float matrix[N][N]) {  printf("| x/y |");  for (int i = 1; i < N; i++)  printf(" %11f |", matrix[0][i]);  printf("\n");  for (int i = 1; i < N; i++)  {  for (int j = 0; j < N; j++)  {  printf("| %11f ", matrix[i][j]);  if (j == N - 1)  printf("|\n");  }  } }  int input\_values(float \*a, float \*b, float \*c, float \*d) {  printf("Введите nx: ");  if (scanf("%f", a) != 1)  return ERR\_IO;  printf("Введите ny: ");  if (scanf("%f", b) != 1)  return ERR\_IO;  printf("Введите x: ");  if (scanf("%f", c) != 1)  return ERR\_IO;  printf("Введите yx: ");  if (scanf("%f", d) != 1)  return ERR\_IO;  return OK; } |
| --- |

Файл: methods.c

| #include "functions.h" #include "errors.h"  int comparator(const void \*q, const void \*p) {  const data\_t \*a = q;  const data\_t \*b = p;  return a->x - b->x; }  void check\_beg\_end(int \*b, int \*e, int n) {  if (n % 2 == 0)  \*b += 1;  if (\*b < 0)  {  \*b += 1;  \*e += 1;  } }  void get\_cur\_pos(float arr[N][N], float x, float y, int \*a, int \*b) {  int i = 1;  int j = 1;  while (i < N)  {  if (arr[0][i] > x)  break;  i++;  }  while (j < N)  {  if (arr[j][0] > y)  break;  j++;  };  \*a = i;  \*b = j; }  void get\_point(float mtx[N][N], float tbl[N][N], float x\_arr[N], float y\_arr[N], int nx, int ny, float x, float y) {  int beginning1, end1;  int beginning2, end2;  int i, j, k, l;  get\_cur\_pos(mtx, x, y, &i, &j);  beginning1 = i - (nx + 1) / 2 - 1;  end1 = i + (nx + 1) / 2 - 1;  check\_beg\_end(&beginning1, &end1, nx + 1);  beginning2 = j - (ny + 1) / 2 - 1;  end2 = j + (ny + 1) / 2 - 1;  check\_beg\_end(&beginning2, &end2, ny + 1);  j = 0;  for (i = beginning1; i <= end1; i++)  {  x\_arr[j] = mtx[0][i];  j++;  }  j = 0;  for (i = beginning2; i <= end2; i++)  {  y\_arr[j] = mtx[i][0];  j++;  }  k = 0;  for (j = beginning2; j <= end2; j++)  {  l = 0;  for (i = beginning1; i <= end1; i++)  {  tbl[k][l] = mtx[i][j];  l++;  }  k++;  } }  int get\_pos(data\_t \*arr, float x, int len) {  int i = 0;  while (i < len)  {  if (arr[i].x > x)  break;  i++;  }  return i; }  float newton(data\_t \*arr, int len, float x, int n) {  float table[N][N], rem, result;  int beginning, end;  int i, k, j = 0;  qsort(arr, len, sizeof(data\_t), comparator);  i = get\_pos(arr, x, len);  beginning = i - ((n + 1) / 2) - 1;  end = i + ((n + 1) / 2) - 1;  check\_beg\_end(&beginning, &end, n + 1);  for (i = beginning; i <= end; i++, j++)  table[j][0] = arr[i].y;  for (i = 1; i <= n; i++)  for (j = 0, k = beginning; k < (end - i + 1); j++, k++)  table[j][i] = (table[j][i - 1] - table[j + 1][i - 1]) / (arr[k].x - arr[k + i].x);  result = table[0][0];  rem = x - arr[beginning].x;  beginning++;  for (i = 1; i <= n; i++, beginning++)  {  result += rem \* table[0][i];  rem = rem \* (x - arr[beginning].x);  }  return result; }  float to\_data(data\_t \*arr, float x\_arr[N], float y\_arr[N], float tbl[N][N], float new[N], int nx, int ny, float x, float y) {  int i, j;  for (i = 0; i < nx + 1; i++)  arr[i].x = x\_arr[i];  for (i = 0; i < ny + 1; i++)  {  for (j = 0; j < nx + 1; j++)  arr[j].y = tbl[i][j];  new[i] = newton(arr, nx+1, x, nx);  }  for (i = 0; i < ny + 1; i++)  {  arr[i].x = y\_arr[i];  arr[i].y = new[i];  }  return newton(arr, ny+1, y, ny); } |
| --- |

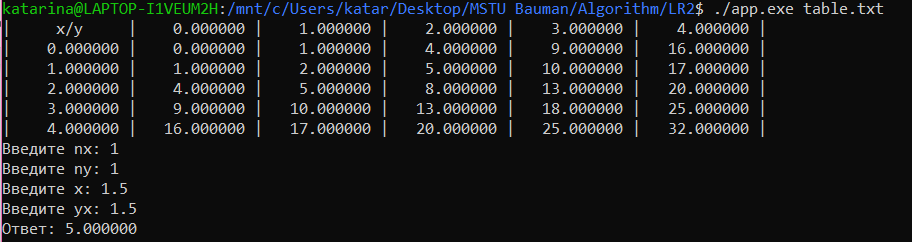
Файл: main.c

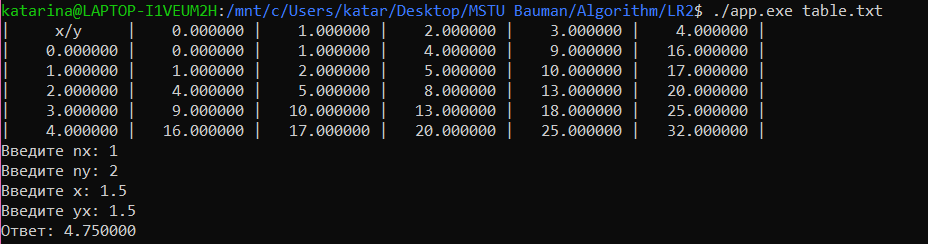
| #include "functions.h" #include "errors.h"  int main(int argc, char \*argv[]) {  int rc = OK;  float enter[N][N], table[N][N];  float x\_ans[N], y\_ans[N], new[N];  float nx, ny, x, y;  float res;  data\_t data[N];  FILE \*f;  if (argc < 2)  {  printf("Недостаточно количество аргументов\n");  rc = ERR\_ARGS;  }  else  {  f = fopen(argv[1], "r+");  if (f == NULL)  {  printf("Не возможно октрыть файл\n");  return ERR\_FILE;  }  else  {  rc = read\_file(f, enter);  if (!rc)  {  print\_matrix(enter);  rc = input\_values(&nx, &ny, &x, &y);  if (!rc)  {  get\_point(enter, table, x\_ans, y\_ans, nx, ny, x, y);  res = to\_data(data, x\_ans, y\_ans, table, new, nx, ny, x, y);  printf("Ответ: %f \n", res);  }  else  printf("Неправильный ввод\n");  }  else  printf("Невозможно прочитать файл\n");  fclose(f);  }  }  return rc; } |
| --- |

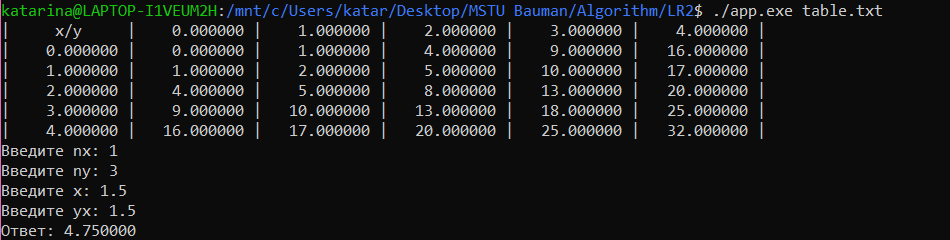
Результаты работы

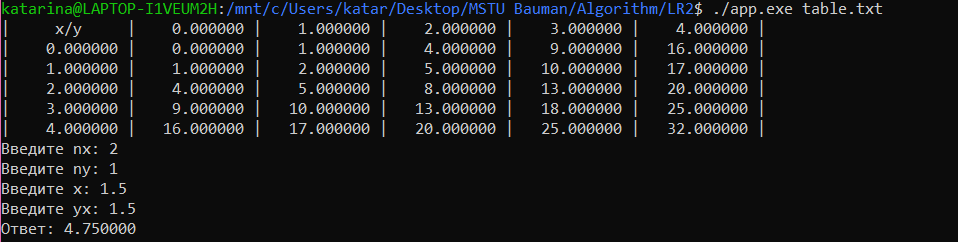
| ny / nx | 1 | 2 | 3 |
| --- | --- | --- | --- |
| 1 | 5.0 | 4.75 | 4.75 |
| 2 | 4.75 | 4.5 | 4.5 |
| 3 | 4.75 | 4.5 | 4.5 |

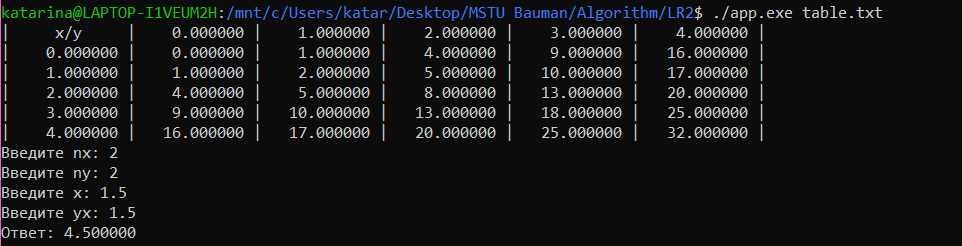
(x = y = 1.5)

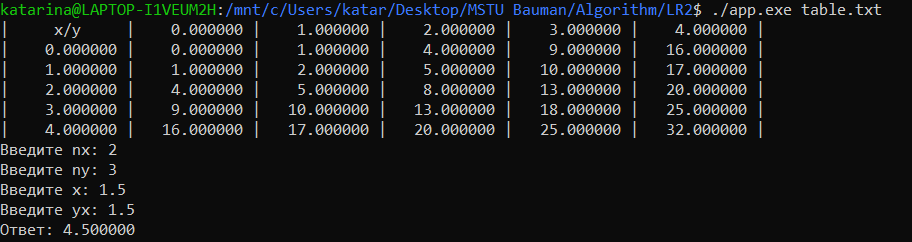


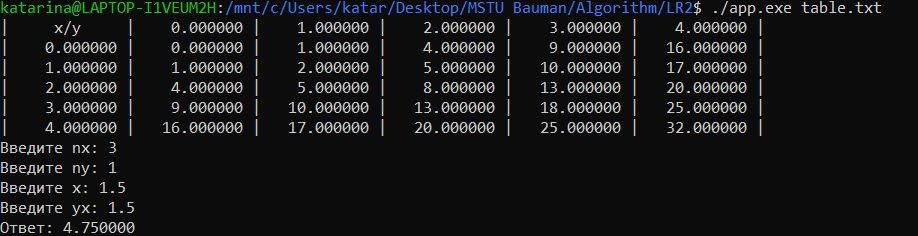


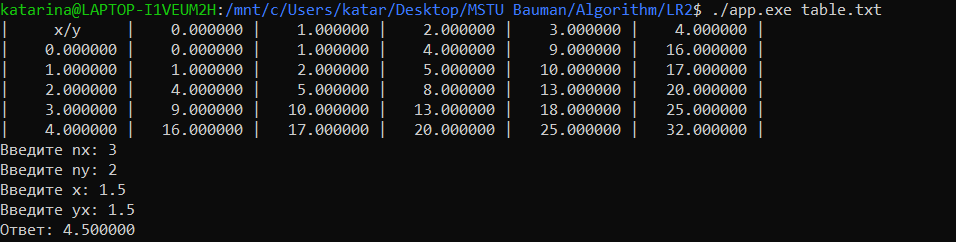


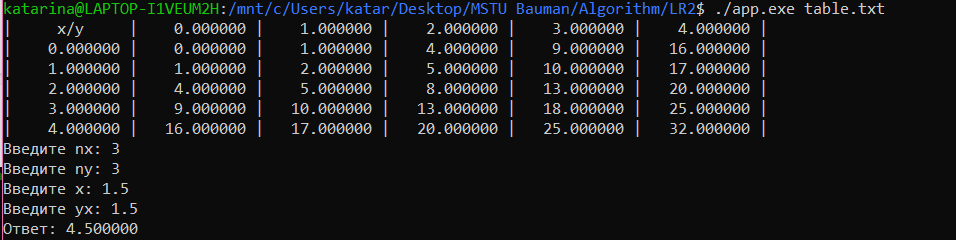












Вопросы при защите лабораторной работы

1. *Пусть производящая функция таблицы суть z(x,y) = x2 + y2 . Область определения по x и y 0-5 и 0-5. Шаги по переменным равны 1. Степени nx = ny =1, x=y=1.5. Приведите по шагам те. значения функции, которые получаются в ходе последовательных интерполяций. по строкам и столбцу.*

| y/x | ... | 1 | 2 | ... |
| --- | --- | --- | --- | --- |
| ... |  |  |  |  |
| 1 |  | 2 | 5 |  |
| 2 |  | 5 | 8 |  |
| ... |  |  |  |  |

Конфигурация узлов по х: 1, 2

Конфигурация узлов по у: 1, 2

Интерполяция по х если у = 1:

z(x, y) = 3.5

Интерполяция по х если у = 2:

z(x, y) = 6.5

Интерполяция по у когда получим значения функциях выше:

z(x, y) = 5

1. *Какова минимальная степень двумерного полинома, построенного на четырех узлах? На шести узлах?*

Полином второй степени: z = a + bx + cy + dx2 + ey2 + fxy

На таком полиноме максимум пользует 6 узлов. Из-за этого минимальна степень полинома 2 и для 4 и для 6 узлах.

1. *Предложите алгоритм двумерной интерполяции при хаотичном расположении узлов, т.е. когда таблицы функции на регулярной сетке нет, и метод последовательной интерполяции не работает. Какие имеются ограничения на расположение узлов при разных степенях полинома?*

Узлы надо выбирать в окрестности точки интерполяции (полином первой степени - 3 узла, полином второй степени - 6 узлов ...). Количество узлов соответствует числу неизвестных коэффициентов в полиноме. Потом выбранные узлы подставляются в полином. Затем образуется система из которой выражаются необходимые коэффициенты.

1. *Пусть на каком-либо языке программирования написана функция, выполняющая интерполяцию по двум переменным. Опишите алгоритм использования этой функции для интерполяции по трем переменным.*

Сначала создаем структуру узлов по третей переменной. Затем для каждого значения переменной проводим двумерную интерполяцию. На конце проводим одномерную интерполяцию по этой переменной на основе полученных результатов в втором шаге.

1. *Можно ли при последовательной интерполяции по разным направлениям использовать полиномы несовпадающих степеней или даже разные методы одномерной интерполяции, например, полином Ньютона и сплайн?*

Можно, из-за того что для последующих интерполяций других направлениях важен только результат.

1. *Опишите алгоритм двумерной интерполяции на треугольной конфигурации узлов.*

Алгоритм двумерной интерполяции на треугольной конфигурации узлов сводится к вычислению разделенных разностей и полинома Ньютона.

Пример вычисления:

Многочлен n степен в форме Ньютона для двумерной интерполяции можно представить так: