Бабенко Анастасия Александровна,
Студент аспирантуры 2-го года обучения
НИУ «БелГУ» Россия, г. Белгород
Babenko Anastasia Alexandrovna,
2st year postgraduate student
NRU "BelGU" Russia, Belgorod
Бабенко Александр Андреевич,
Студент аспирантуры 1-го года обучения
НИУ «БелГУ» Россия, г. Белгород
Babenko Alexander Andreevich,
1st year postgraduate student
NRU "BelGU" Russia, Belgorod

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ НАХОЖДЕНИЯ ПЕРИМЕТРА И ПЛОЩАДИ ФИГУРЫ С ПОМОЩЬЮ АППРОКСИМАЦИИ

SOFTWARE IMPLEMENTATION OF FINDING THE PERIMETER AND THE AREA OF THE FIGURE USING APPROXIMATION

Аннотация: В статье реализована программа нахождения периметра и площади с использованием аппроксимации. Для аппроксимации заданной фигуры был использован алгоритм Дугласа-Пекера.

Ключевые слова: аппроксимация, алгоритм Дугласа-Пекера, формула площади Гаусса, C++ .

Abstract: The article implements a program for finding the perimeter and area using approximation. The Douglas-Pecker algorithm was used to approximate a given figure.

Keywords: approximation, the Douglas-Pecker algorithm, the Gauss area formula, C++ .

Для аппроксимации заданной фигуры был использован алгоритм Дугласа-Пекера. Алгоритм Дугласа-Пекера — это алгоритм, позволяющий уменьшить число точек кривой, аппроксимированной большей серией точек. Суть алгоритма состоит в том, чтобы по данной ломаной,

аппроксимирующей кривую, построить ломаную с меньшим числом точек. Алгоритм определяет расхождение, которое вычисляется по максимальному расстоянию между исходной и упрощённой кривыми. Упрощенная кривая состоит из подмножества точек, которые определяются из исходной кривой. Для нахождения периметра фигура разбивалась на отрезки, и вычислялись их длины с помощью расстояния Евклида между двумя точками. Существующие множество методов нахождения площади, но для конкретной задачи будут использоваться два следующих метода: формула площади Гаусса и нахождение площади по подобию многоугольников. Для написания программы была выбрана среда программирования Visual Studio 2019 C++, основанная на языке программирования C++. Данная среда выгодно отличается эффективностью и надежностью

Программная реализация нахождения периметра и площади с использованием аппроксимации:

1.Нахождения максимального расстояния

Листинг кода на C++нахождения максимального расстояния:

```
double calculateDistance(const Point& pt, const Point& lineStart, const Point& lineEnd) {
double dx = lineEnd.first - lineStart.first;
double dy = lineEnd.second - lineStart.second;
double mag = pow(pow(dx, 2.0) + pow(dy, 2.0), 0.5);
 if (mag > 0.0) {
dx /= mag; dy /= mag;}
double pvx = pt.first - lineStart.first; double pvy = pt.second - lineStart.second;
double pvdot = dx * pvx + dy * pvy; double dsx = pvdot * dx;
 double dsy = pvdot * dy;//Вычесть его из pv
 double ax = pvx - dsx; double ay = pvy - dsy;
 return pow(pow(ax, 2.0) + pow(ay, 2.0), 0.5); }
double dmax = 0.0;
 size_t index = 0;
 size_t end = pointList.size() - 1;
 for (size_t i = 1; i < end; i++) {
  double d = calculateDistance(pointList[i], pointList[0], pointList[end]);
  if (d > dmax) {
   index = i; dmax = d;} }
```

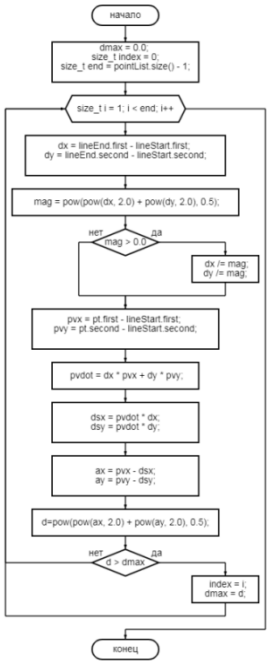

Рисунок 1 - Блок схема нахождения максимального расстояния

2. Алгоритм аппроксимации методом Дугласа-Пекера

Листинг кода на C++ алгоритма Дугласа-Пекера:

```
if (dmax > epsilon) {
  //Если максимальное расстояние больше ε, упростить кривую
  vector<Point> recResults1; vector<Point> recResults2;
  vector<Point> firstLine(pointList.begin(), pointList.begin() + index + 1);
  vector<Point> lastLine(pointList.begin() + index, pointList.end());
  mainAlgorithm(firstLine, epsilon, recResults1); mainAlgorithm(lastLine, epsilon, recResults2);
  //Построить список с результатами
  out.assign(recResults1.begin(), recResults1.end() - 1);
  out.insert(out.end(), recResults2.begin(), recResults2.end());
  if (out.size() < 2) throw runtime_error("Проблема с построением списка вывода"); }
 else {//Иначе вернуть точки start и end
  out.clear(); out.push_back(pointList[0]); out.push_back(pointList[end]); }}
```

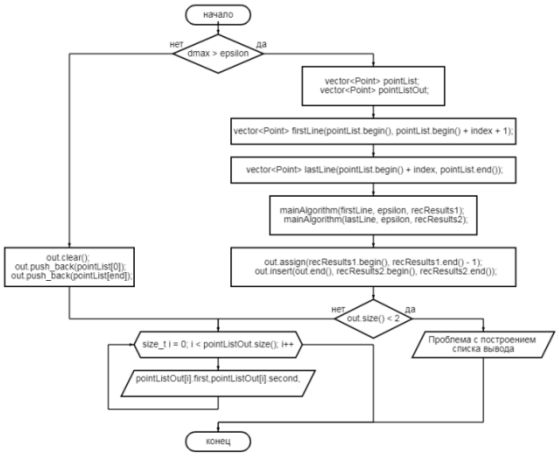

Рисунок 2 - Блок схема алгоритма Дугласа-Пекера

3. Нахождение я периметра и площади

Листинг кода на C++ нахождения периметра

```
for (size_t i = 0; i < pointList1.size(); i++) {
if (i != pointList1.size() - 1) {
P1 += sqrt(pow((pointList1[i + 1].first - pointList1[i].first), 2.0) + pow((pointList1[i + 1].second
- pointList1[i].second), 2.0));
S1 += 0.5 * abs(pointList1[i].first * pointList1[i + 1].second - pointList1[i + 1].first *
pointList1[i].second);}
else{
P1 += sqrt(pow((pointList1[0].first - pointList1[i].first), 2.0) + pow((pointList1[0].second -
pointList1[i].second), 2.0));
S1 += 0.5 * abs(pointList1[i].first * pointList1[0].second - pointList1[0].first *
pointList1[i].second);}}
for (size_t i = 0; i < pointListOut.size(); i++) {
if (i != pointListOut.size() - 1) {
P2 += sqrt(pow((pointListOut[i + 1].first - pointListOut[i].first), 2.0) + pow((pointListOut[i +
1].second - pointListOut[i].second), 2.0));}
else{
P2 += sqrt(pow((pointListOut[0].first - pointListOut[i].first), 2.0) + pow((pointListOut[0].second
- pointListOut[i].second), 2.0));}}
S2 = S1 * pow(P2,2.0) / pow(P1,2.0);
cout « "Периметр P1 = " « P1 « endl;
cout « "Периметр P2 = " « P2 « endl;
cout « "Площадь S1 = " « S1 « endl;
cout « "Площадь S2 = " « S2 « endl;
```

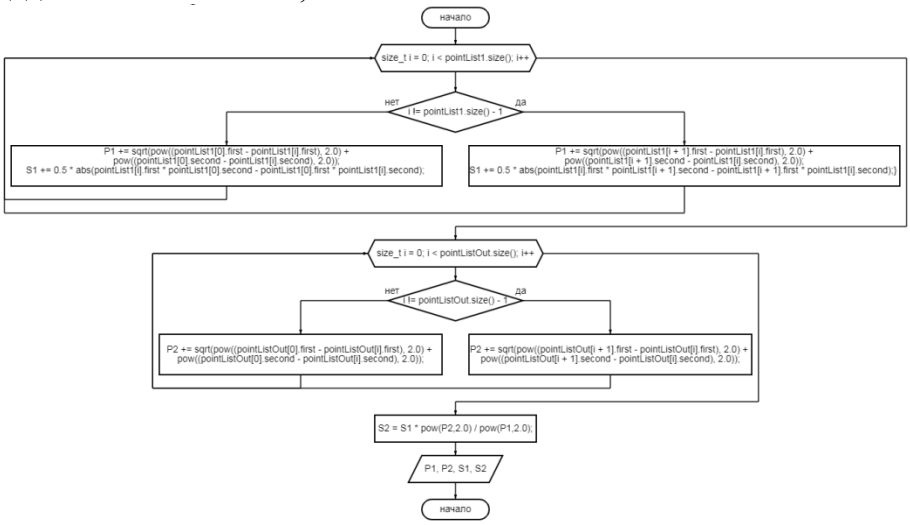

Рисунок 3 - Блок нахождения периметра и площади

Тестирование алгоритма Дугласа-Пекера

Фигура для аппроксимации представлена на рис. 4.

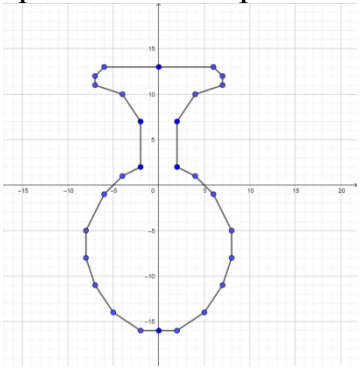

Рис. 4. Входное изображение

Периметр и площадь фигуры при ε=0,5, рис.5.

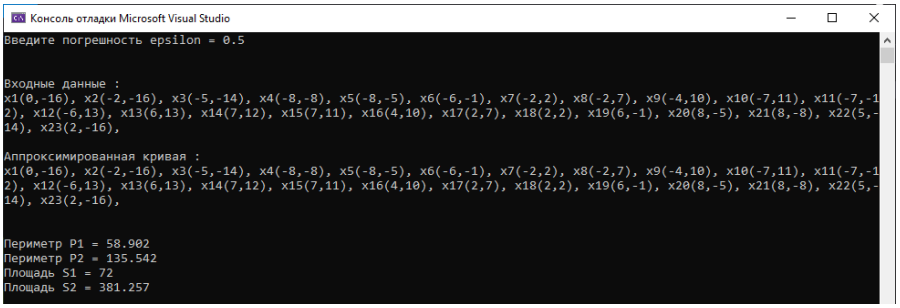

Рисунок 5 - Периметр и площадь фигуры при ε=0,5

Периметр и площадь фигуры при ε=2, рис.6.

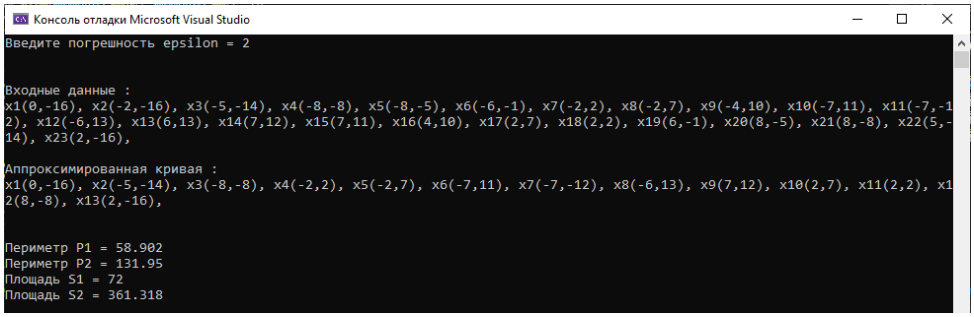

Рисунок 6 - Периметр и площадь фигуры при ε=2

Фигура после аппроксимации с ε=0,5 представлена на рис. 23.

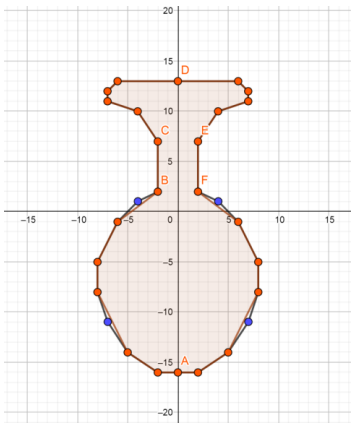

Рисунок 7 - Фигура после аппроксимации с ε=0,5

Список использованной литературы

1. Дейтел Х.М. Как программировать на C++: Пятое издание / Х.М. Дейтел, - С.Пб. ООО «Бином-Пресс», 2011. - 1456 с.

2. Понарин Я.П. Элементарная геометрия. Том 1. Планиметрия, преобразования плоскости / Я.П. Понарин, - М.: МЦНМО, 2004. — 312 с.

3. Семина Е.В. Основы алгоритмизации и структурного программирования на C++: учебно-методическое пособие / Е.В. Семина, А.В. Артемов, - Орел, 2014. – 48 с.