

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ
КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ

Допущен к защите
Заведующий кафедрой
информационных систем,
д. т. н., доцент
_____ И. Н. Глухих
“ ____ ” _____ 2003 г.

Змановский Петр Петрович

ПРОГРАММНЫЙ КОМПЛЕКС ПРИБЛИЖЕНИЯ ФУНКЦИЙ НА
НЕРЕГУЛЯРНЫХ СЕТКАХ

(дипломная работа)

Научный руководитель:

к. ф. - м. н., доцент

_____ П. К. Моор

Автор работы:

_____ Змановский П. П.

Тюмень - 2003

ВВЕДЕНИЕ.....	3
ГЛАВА 1. МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ.....	7
ГЛАВА 2. ОБЗОР МЕТОДОВ И ПРОГРАММ ПРИБЛИЖЕНИЯ ФУНКЦИЙ.....	11
2.1. Методы, их характеристики, сравнение.....	11
2.2. Программы, их характеристики, сравнение.....	21
3. ПРОГРАММНЫЙ КОМПЛЕКС ПРИБЛИЖЕНИЯ ФУНКЦИЙ КУБИЧЕСКИМИ СПЛАЙНАМИ.....	29
3.1. Описание алгоритмов.....	29
3.1.1. Построение оптимальной нерегулярной сетки.....	29
3.1.2. Построение интерполирующих функций.....	33
3.1.3. Визуализация полученных результатов.....	40
3.2. Описание программного комплекса.....	43
Заключение.....	52
Список литературы.....	54

ВВЕДЕНИЕ

При математическом моделировании физических явлений нередко возникает задача приближения функций, которая может иметь различные постановки. Наиболее распространенными из них являются задачи интерполирования и аппроксимации функций. В первом случае ставится задача построения для функции f по заданным ее значениям в дискретном множестве точек приближения \mathcal{P} . При этом интерполянт \mathcal{P} должен совпадать с функцией f , по крайней мере, в этих точках. Другая задача, близкая к проблеме интерполяции, возникает, когда значения функции в информационных точках известны не точно, а с некоторой погрешностью, максимальная величина которой в каждом узле задается в качестве априорной информации. В этом случае ставится задача аппроксимации - построения для функции f наилучшего по некоторому критерию приближения \mathcal{P} .

При математическом моделировании задач нефтяной геологии и разведочной геофизики возникают следующие задачи:

- исследование имеющихся данных;
- постановка требований, которым должна удовлетворять модель данных;
- построение модели, адекватной поставленным требованиям;
- визуализация полученных результатов (построение контурных карт, диаграмм, объемных моделей).

При исследовании данных возникают задачи статистической обработки имеющихся данных, фильтрации повторяющихся значений, выбора метода последующего интерполирования, построения экспериментальной вариограммы и ее модели (в основном для различных методов крайгинга), а также другие задачи, в зависимости от выбранного метода.

При построении модели производятся вычисления согласно выбранному методу. В некоторых методах возникает также задача разбиения интерполируемой

области на части (обычно треугольники или многоугольники), которая решается отдельно от непосредственной интерполяции. Наиболее популярные методы построения нерегулярной сетки – построение треугольников Делоне, а также треугольников Вороного.

Визуализация полученных результатов очень важна, так как в большинстве случаев именно она является основным результатом работы. Карты служат примером графических моделей геологических поверхностей, отражают их пространственную изменчивость и являются непрерывным, гладким описанием дискретных наблюдений.

В данный момент существует много алгоритмов и программ, позволяющих исследовать, интерполировать и визуализировать трехмерные данные. Такие программы имеют определенные недостатки. В [1] В.А. Мальцев разделяет такие программы на два основных класса:

- программы с хорошо проработанным математическим аппаратом, позволяющие контролировать процесс интерполяции возможностью установки огромного числа параметров, но с плохим пользовательским интерфейсом (GMT, GMS, DataMine и др.).
- программы с хорошо проработанным пользовательским интерфейсом, но с невозможностью на должном уровне контролировать процесс интерполяции (Golden Software Surfer и др.).

Такое разделение существует потому, что для проведения процесса интерполяции нужно серьезно исследовать имеющиеся данные и выбрать подходящий в данном случае метод, а для этого необходима определенная математическая подготовка. Поэтому программы первого типа разрабатывались в расчете на хорошо подготовленных специалистов, которые смогут работать и без пользовательского интерфейса. Более того, в данном случае интерфейс только мешает, так как такие программные комплексы чаще всего разделены на

множество программ, работа с которыми производится через командную строку, таким образом, появляется возможность написания скриптов, управляющих процессом. Надстройка графического интерфейса над таким программным комплексом является довольно сложной и трудоемкой задачей, так как изначально программные комплексы проектировались для работы через скрипты.

В свою очередь, программы второго типа (наиболее ярким представителем является программа Golden Software Serfer) имеют прекрасный пользовательский интерфейс, однако это достигается путем практически полной автоматизации процесса интерполяции, что во многих случаях является абсолютно неприемлемым. Хотя стоит отметить, что такие программы пользуются большой популярностью, так как более понятны для простых пользователей.

Большое число задач нефтяной геологии, разведочной геофизики и метеорологии требуют применения методов приближения функций двух переменных. Классический аппарат решения этих проблем - теория полиномиальных и мелко-линейных приближений достаточно хорошо разработана для функций одного переменного, однако мало подходит для многомерных объектов. Одним из подходов, применяемых к решению таких задач, является сплайн-интерполирование.

Необходимо отметить, что математическая и алгоритмическая база для решения задач данного класса не является хорошо проработанной, поэтому исследования в данной области довольно перспективны.

В данной работе решены следующие задачи:

- рассмотрен и реализован программно алгоритм построения триангуляции Делоне и задача кубического сплайн - интерполирования на нерегулярных триангулированных сетках.
- выведены формулы для построения системы, предназначенной для нахождения кубического сплайна.

- придуман и описан метод, позволяющий существенно оптимизировать алгоритм интерполяции.
- рассмотрены дополнительные условия, накладываемые на интерполянт, в связи с появлением в системе некоторого числа свободных переменных, выведены соответствующие формулы.
- рассмотрен метод построения поля изолиний для визуализации полученных результатов (построение контурной карты).
- создан программный комплекс, реализующий описанные методы и решающий задачи построения триангулированной сетки, интерполяции кубическими сплайнами и построения поля изолиний.

ГЛАВА 1. МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ

Пусть на плоскости в замкнутой области $G^0 \subset R^2$ определена функция $f(x, y)$. Рассмотрим задачу построения приближения $\varphi(x, y)$ на основании информации о функции f в N точках: $W = \{w_i = (x_i, y_i) : w_i \in G^0, i = 1, \dots, N\}$.

Совокупность значений функции в этих точках будем обозначать через Z :

$$Z = \{z_i = f(x_i, y_i) : (x_i, y_i) \in W, i = 1, \dots, N\}.$$

Заметим, что среди постановок задач построения приближения можно выделить следующие:

- функция задана в области G^0 аналитически или алгоритмически, а точки набора W заранее не заданы и могут выбираться произвольно;
- точки набора W заранее зафиксированы и в них заданы значения функции или её приближенные значения.

Рассмотрим построение сплайна $\varphi(x, y)$ в области G , которая является приближением на плоскости области G^0 : будем считать, что среди N точек набора W выделено N_r точек (которые мы будем называть граничными) так, что ломанная, проходящая через них в заданной последовательности, не имеет самопересечений и ограничивает область G . Остальные N_b точек будем называть внутренними ($N = N_r + N_b$). Граничные точки могут располагаться как на границе G^0 , так и внутри её. В случае, когда набор W заранее не задан, эти точки следует выбирать так, чтобы граница G хорошо приближала границу G^0 .

Принципиальным осложнением, которое возникает в многомерной задаче приближения (по сравнению с одномерной) является отсутствие упорядоченности информационных точек. При регулярном их расположении (например, в узлах прямоугольной сетки или сетке правильных треугольников) оказывается возможным применение классической теории интерполяции и аппроксимации. Отсутствие закономерности в их расположении делает это невозможным и в этом

случае целесообразно использовать нерегулярные сетки – сетки, элементы которых не являются прямоугольниками или правильными треугольниками. Популярность применения нерегулярных сеток при решении задач, когда точки набора W не определены постановкой задачи, обуславливается также и тем, что они позволяют находиться расчетным точкам на криволинейной границе области и произвольно располагаться внутри неё.

При применении нерегулярных сеток для приближения по заданным в некоторых точках значениям функции двух переменных, возникает задача построения оптимального в некотором смысле разбиения заданной области на непересекающиеся и образующие её покрытие подобласти. Наиболее часто для этого используются разбиения на многоугольные (при кусочно-постоянном приближении) или треугольные элементы.

Основным моментом в задаче построения таких сеток является вопрос о дополнительных условиях, которым должны удовлетворять их элементы, без этого построение разбиений оказывается неоднозначным. Отметим еще раз, что задача построения сетки может иметь различные постановки: построение разбиения некоторой области при произвольном или фиксированном размещении узлов. В первом случае точки, образующие вершины многоугольников или треугольников, могут выбираться в соответствии с критерием оптимизации используемого численного метода (метода конечных элементов, построения квадратурных формул, разностных схем и т.д.). Так, оптимизация разбиений в методе конечных элементов основывается на условиях минимальной энергии, которая зависит от конфигурации сетки. Во втором случае положение точек оказывается заданным. Это зачастую имеет место, например, в задачах приближения геологических поверхностей и моделирования процесса разработки нефтяных и газовых месторождений. Здесь, как правило, информационными

являются точки геофизических наблюдений или точки размещения эксплуатационных или разведочных скважин.

Пусть на плоскости в области G задано N несовпадающих точек $W = \{w_i = (x_i, y_i) : i = 1, \dots, N\}$, причем N_r из них определены как граничные, а остальные N_b как внутренние. Без ограничения общности будем считать, что внутренние точки имеют номера $1, 2, \dots, N_b$. Пусть для области G задана триангуляция: $\Sigma(W) = \{S_m, m = 1, \dots, M\}$, разбиение на замкнутые подобласти, которые являются треугольными элементами с вершинами из W и такие, что $\bigcup_{m=1}^M S_m = G$, а $S_m \cap S_n$ при $m \neq n$ есть отрезок прямой (соседние треугольники), точка, или пустое множество. Для произвольной триангуляции $\Sigma(W)$ из соотношений Эйлера следует, что справедливы формулы, связывающие общее количество точек – N , количество граничных – N_r и внутренних – N_b точек, число треугольников – M , общее количество ребер – L количество граничных L_r и внутренних ребер – L_b :

$$N = N_b + N_r,$$

$$M = N + N_b - 2 = 2N - N_r - 2,$$

$$L_r = N_r,$$

$$L_b = N + 2N_b - 3 = 3N - 2N_r - 3,$$

$$L = L_b + L_r = 2N + N_b - 3 = 3N - N_r - 3.$$

Пусть далее в точках набора W заданы значения функции $F = \{f_i\}_{i=1}^N = \{f(x_i, y_i)\}_{i=1}^N$ и ставится задача построения в области G двумерного кубического сплайна $\varphi(x, y)$.

Определение. Двумерным сплайном степени α дефекта β на триангулированной сетке Σ области G будем называть функцию $\varphi(x, y)$, являющуюся на каждом элементе $S_m \in \Sigma$ многочленом двух переменных x и y степени α и имеющую в G непрерывные производные до порядка $\alpha - \beta$ включительно (то есть $\varphi(x, y) \in C^{\alpha - \beta}$).

Применение сплайнов первой и второй степени хорошо изучено. В первом случае интерполирующий сплайн имеет дефект 1, а во втором 2, то есть в обоих случаях удастся построить только сплайн класса C^0 .

Рассмотрим задачу приближения функций двумерными сплайнами третьей степени, которые будем называть также кубическими. То есть ставится задача построения функции $\varphi(x, y)$:

$$\varphi(x, y) = \{\varphi^m(x, y) : (x, y) \in S_m, S_m \in \Sigma(W)\}$$

$$\varphi^m(x, y) = a_0^m + a_1^m x + a_2^m y + a_3^m x^2 + a_4^m xy + a_5^m y^2 + a_6^m x^3 + a_7^m x^2 y + a_8^m xy^2 + a_9^m y^3,$$

приближающей функцию f в точках W_i . Задачу приближения функций мы будем рассматривать в постановке интерполирования. В этом случае интерполянт должен удовлетворять условиям:

$$\varphi(W_i) = \varphi(x_i, y_i) = f_i, \quad w_i \in W.$$

Вопросы существования и единственности для заданного набора точек W , значений F и триангуляции $\Sigma(W)$ интерполяционного кубического сплайна $\varphi \in C^2$ и способы применения дополнительных условий при наличии свободных переменных в решаемой системе рассмотрен в [2].

При нахождении интерполяционного кубического сплайна перед нами встают следующие задачи:

- построение триангуляции Делоне;
- вывод уравнений для нахождения коэффициентов сплайна;
- избавление от свободных переменных;
- построение интерполирующего кубического сплайна на данной сетке.

ГЛАВА 2. ОБЗОР МЕТОДОВ И ПРОГРАММ ПРИБЛИЖЕНИЯ ФУНКЦИЙ

2.1. Методы, их характеристики, сравнение

В настоящее время существует большое число алгоритмов интерполяции трехмерных данных, использующих разнообразные интерполяционные или аппроксимационные методы приближения. Это множество можно разделить на четыре основные группы:

- методы, приближающие данные некоторыми функциями.
- методы взвешенного интерполирования и крайгинги (геостатистические методы);
- методы приближения сплайн – функциями;
- методы конечных элементов.

Наиболее известными являются:

1. Метод линейной интерполяции (Linear Interpolation)
2. Методы взвешенного интерполирования:
 - i. Метод Шепарда (Shepards Method);
 - ii. Метод градиентных нодальных функций;
 - iii. Метод квадратичных нодальных функций;
3. Интерполяция Клау-Тахера;
4. Метод натурального соседа;
5. Различные методы крайгинга (Kriging);
6. Методы сплайн – интерполирования.

2.1.1. Линейная интерполяция (Linear Interpolation).

Линейная интерполяция является самой простой и наименее точной. Однако она является очень быстрой, поэтому используется для быстрой обработки данных. Кроме того, при равномерном и достаточно большом количестве точек, этот метод может показывать неплохие результаты.

Уравнение плоскости, определенной тремя точками треугольника:

$$Ax + By + Cz + D = 0,$$

где A , B , C и D вычисляются по координатам (x_1, y_1, z_1) , (x_2, y_2, z_2) и (x_3, y_3, z_3) :

$$A = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2);$$

$$B = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2);$$

$$C = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2);$$

$$D = -Ax_1 - By_1 - Cz_1.$$

Уравнение плоскости также может быть записано в следующем виде:

$$z = f(x, y) = -\frac{A}{C}x - \frac{B}{C}y - \frac{D}{C}.$$

2.1.2. Методы взвешенного интерполирования обратно пропорционального расстоянию (inverse distance weighted (IDW) interpolation).

Одной из наиболее часто применяемых технологий при интерполяции точек на нерегулярной сетке являются методы взвешенного интерполирования. Такие методы базируются на предположении, что интерполируемая поверхность зависит в большей степени от тех точек, которые расположены ближе к ней, чем от тех, которые расположены относительно далеко. Интерполируемая поверхность вычисляется путем вычисления взвешенного среднего исходных точек. Вес каждой исходной точки зависит от расстояния от нее до интерполируемой точки. Чем меньше расстояние, тем больше вес.

2.1.3. Метод Шепарда (Shepards Method).

Наиболее простым методом inverse distance weighted интерполяции является так называемый «Метод Шепарда» (Shepard 1968).

Используется следующее уравнение:

$$F(x, y) = \sum_{i=1}^n w_i f_i,$$

где n – число исходных точек в множестве, f_i – значение функции в этих точках (т.е. множество значений), и w_i – весовая функция, определенная для каждой исходной точки. Классическая форма весовой функции это:

$$w_i = \frac{h_i^{-p}}{\sum_{j=1}^n h_j^{-p}},$$

где p – некоторое положительно действительно число, называемое степенным параметром (обычно, $p=2$) и h_i – расстояние от исходной точки до интерполируемой, т.е.:

$$h_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$$

где (x, y) – координаты интерполируемой точки, а (x_i, y_i) – координаты i -ой исходной точки. Весовые функции нормализуются.

Также можно использовать следующее уравнение для нахождения весовых функций:

$$w_i = \frac{\left[\frac{R - h_i}{Rh_i} \right]^2}{\sum_{j=1}^n \left[\frac{R - h_j}{Rh_j} \right]^2}$$

где h_i – расстояние от интерполируемой точки до исходной, R – расстояние от интерполируемой точки до наиболее отдаленной исходной, n – число исходных точек. Это уравнение дает лучшие результаты, чем классическое (Franke & Nielson, 1980).

Метод Шепарда используется очень часто из-за его простоты.

2.1.4. Метод градиентных нодальных функций.

Ограничение метода Шепарда заключается в том, что интерполируемая поверхность является просто взвешенным средним значений функции в исходных точках, и поэтому все получаемые значения лежат в промежутке экстремумов исходного множества значений. Эта проблема решается использованием другого

уравнения для нахождения значений в интерполяционных точках: $F(x, y) = \sum_{i=1}^n w_i Q_i(x, y)$,

где Q_i – нодальные функции, определенные в каждой исходной точке (Franke 1982; Watson & Philip 1985).

$$Q_i(x, y) = f_x(x - x_i) + f_y(y - y_i) + f_i,$$

где f_x и f_y – частные производные в исходной точке, предварительно вычисленные, основываясь на взаимном расположении окружающих исходных точек. Градиенты обычно подсчитываются путем триангуляции исходных точек и подсчете градиента в каждой исходной точке, как среднего среди градиентов треугольников, примыкающих к данной точке.

2.1.5. Метод квадратичных нодальных функций.

Квадратичные нодальные функции могут использоваться также как и градиентные. Для их нахождения используется следующее уравнение в некоторой точке (x_k, y_k) :

$$Q_k(x, y) = f_k + a_{k2}(x - x_k) + a_{k3}(y - y_k) + a_{k4}(x - x_k)^2 + a_{k5}(x - x_k)(y - y_k) + a_{k6}(y - y_k)^2$$

Коэффициенты находятся путем приближения квадратичной функции к ближайшей исходной точке по методу наименьших квадратов. Для того чтобы

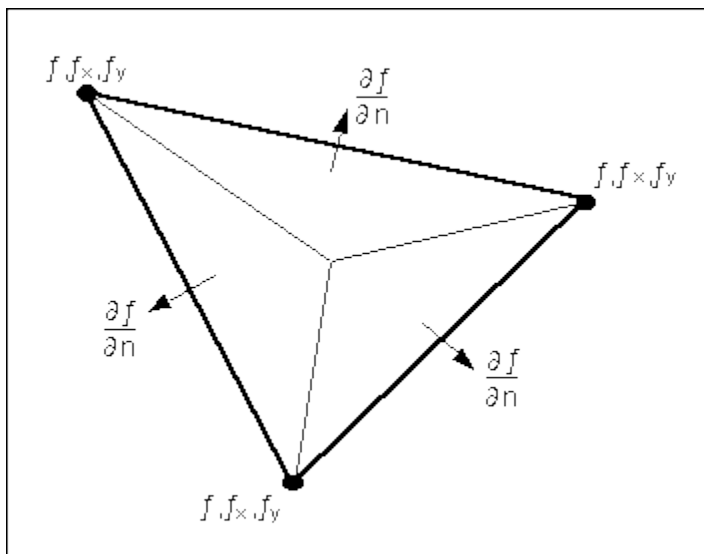
система была совместной, необходимо, что в исходном множестве было, по крайней мере, пять точек.

Далее веса подсчитываются также как и в других методах взвешенного интерполирования обратно пропорционального расстоянию (Шепарда и градиентных нодальных функций) с использованием найденных коэффициентов.

2.1.6. Интерполяция Клау-Тахера.

Перед использованием метода Клау-Тахера необходимо провести предварительную триангуляцию исследуемой области. На каждом треугольнике в итоге находится полином двух переменных. Таким образом, создается непрерывная поверхность, состоящая из треугольников и определенных на них многочленах.

В методе Клау-Тахера используются кубические полиномы, определяемые двенадцатью параметрами (как показано на следующей иллюстрации): значения функций, первые производные f_x и f_y в каждой вершине, а также нормальные производные $\frac{\partial f}{\partial n}$ в центре каждой стороны треугольника (Clough & Tocher, 1965; Lancaster & Salkauskas, 1986). Первые производные приближаются при использовании значений примыкающих треугольников.



Кубический многочлен вида $F(x, y) = \sum_{j=0}^{3-i} c_{ij} x^i y^j$ находится на каждом треугольнике.

Схема Клау-Тахера является локальной, поэтому имеет большое преимущество в скорости. Также, эта схема получает гладкую интерполяционную поверхность, которая хорошо удаляет локальный тренд в исходных данных.

Поскольку триангуляция покрывает часть плоскости внутри описанного вокруг множества точек многоугольника, экстраполяция на точки, находящиеся вне данного многоугольника невозможна при использовании данной схемы.

2.1.7. Метод натурального соседа.

Метод натурального соседа может использоваться как для интерполяции, так и для экстраполяции. Метод натурального соседа был впервые представлен Сибсоном (Sibson) (1981). Более детальное описание этого метода сделано Оуеном (Owen) (1992).

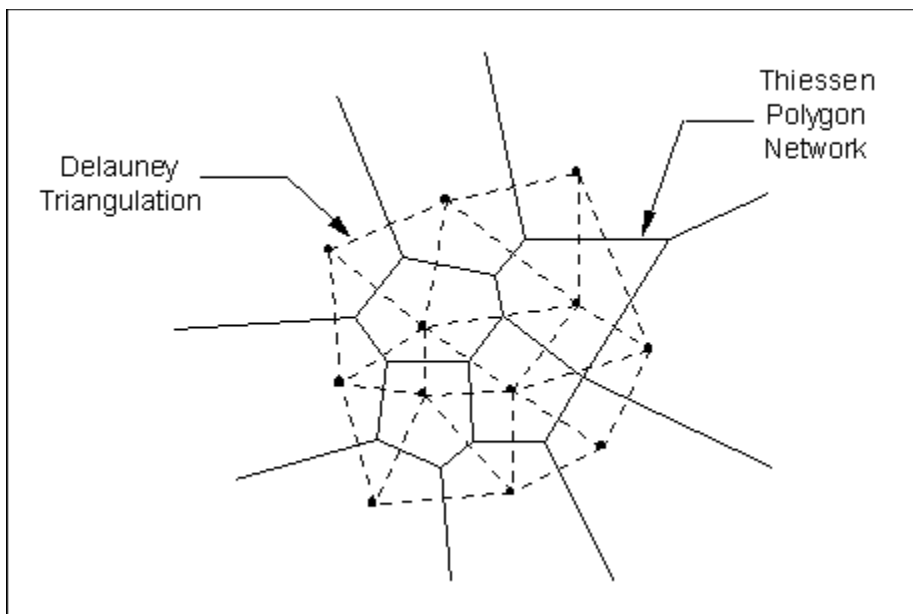
Основные уравнения, используемые в интерполяции натуральным соседом, идентичны тем, которые используются в методе взвешенного интерполирования, обратно пропорционального расстоянию (IDW):

$$F(x, y) = \frac{\sum_{i=1}^n w_i Q_i(x, y)}{\sum_{i=1}^n w_i}.$$

Как и в IDW - интерполяции, нодальные функции могут быть константами, градиентными плоскостями или квадратичными функциями. Отличие метода натурального соседа от IDW-интерполяции заключается только в методе подсчета весов и в методе выбора исходных точек, используемых в интерполяции.

Метод натурального соседа основывается на сети многоугольников Тиссена (Thiessen polygon network) множества исходных точек. Сеть многоугольников

Тиссена может быть построена на основе триангуляции Делоне данного множества точек.



Триангуляция Делоне и соответствующая ей сеть многоугольников Тиссена.

Для каждой исходной точки существует только один многоугольник Тиссена. Такой многоугольник покрывает площадь, каждая точка которой ближе к выбранной исходной точке, чем к любой другой. Многоугольники для граничных точек являются открытыми.

Веса в методе натурального соседа основаны на концепции локальных координат. Значения локальных координат зависят от области влияния многоугольников Тиссена на остальные точки. Чтобы найти локальную координату точки, необходимо знать площади всех многоугольников. Для нахождения значения в данной точке, эта точка временно вставляется в сеть многоугольников, после чего находятся новые значения локальных координат, а также площадь многоугольника, образуемого новой точкой. Чем больше, общая область нового многоугольника и какого-либо другого, тем сильнее влияние соответствующей точки на значение в интерполируемой точке.

Если определить $k(n)$ как площадь многоугольника Тиссена для P_n и $k_m(n)$, как разница площади многоугольника (для одной из соседних точек) до и после

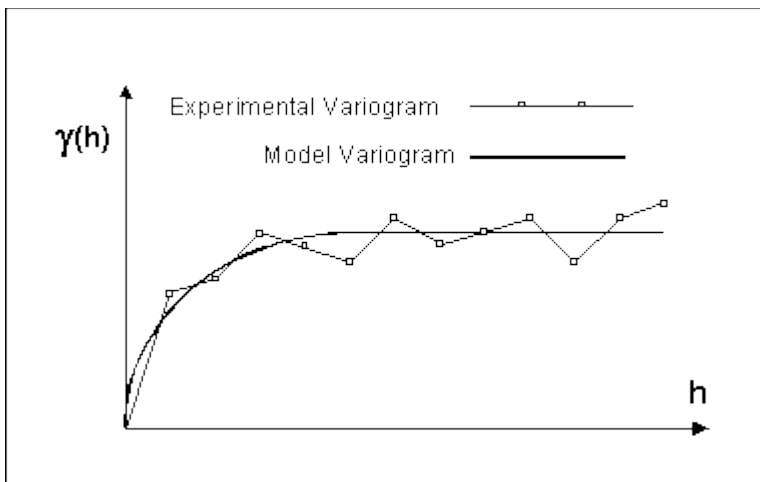
вставки временной точки, то локальная координата находится по следующей формуле:

$$\lambda_m(n) = \frac{\kappa_m(n)}{\kappa(n)}$$

2.1.8. Крайгинг (Kriging).

Крайгинг - это метод интерполяции, названный в честь Южно-Африканского инженера Крайга (D. G. Krige), который его изобрел. В последние десятилетия крайгинг стал фундаментальным методом в области геостатистики.

Первый шаг в крайгинге – построение вариограммы на основе исходных данных. Вариограмма состоит из двух частей: экспериментальная вариограмма и вариограмма модели. Экспериментальная вариограмма находится путем подсчета вариации (g) каждой точки множества относительно остальных исходных точек с учетом расстояния между точками. Разные формулы могут использоваться для учета расстояния, однако наиболее часто используемой является квадрат половины разности значений функций в точках.



Experimental and Model Variogram Used in Kriging

После нахождения экспериментальной вариограммы, находится ее модель, которая является простой математической функцией, моделирующей тренд экспериментальной вариограммы.

После нахождения модели вариограммы, она используется для нахождения весов, используемых в методе крайгинга. Основное уравнение обычного крайгинга выглядит следующим образом:

$$F(x, y) = \sum_{i=1}^n w_i f_i, \text{ где } n - \text{число исходных точек, } f_i - \text{значения функции в этих}$$

точках, w_i – веса, приписанные каждой точке. Это уравнение такое же, как в методе обратного расстояния, за исключением того, что в крайгинге веса основываются на построенной модели вариограммы. Например, для интерполяции точки P , основываясь на окружающих точках P_1 , P_2 , and P_3 , веса w_1 , w_2 , and w_3 находятся путем решения следующей системы уравнений:

$$w_1 S(d_{11}) + w_2 S(d_{12}) + w_3 S(d_{13}) = S(d_{1p});$$

$$w_1 S(d_{12}) + w_2 S(d_{22}) + w_3 S(d_{23}) = S(d_{2p});$$

$$w_1 S(d_{13}) + w_2 S(d_{23}) + w_3 S(d_{33}) = S(d_{3p});$$

где $S(d_{ij})$ – модель вариограммы, вычисленная на расстоянии равном расстоянию между точками i и j . К системе добавляется ее одно уравнение:

$$w_1 + w_2 + w_3 = 1.0.$$

Поскольку, переменных становится меньше, чем уравнений, то добавляется дополнительная переменная. В итоге система выглядит так:

$$w_1 S(d_{12}) + w_2 S(d_{22}) + w_3 S(d_{23}) + \lambda = S(d_{2p});$$

$$w_1 S(d_{12}) + w_2 S(d_{22}) + w_3 S(d_{23}) + \lambda = S(d_{2p});$$

$$w_1 S(d_{13}) + w_2 S(d_{23}) + w_3 S(d_{33}) + \lambda = S(d_{3p});$$

$$w_1 + w_2 + w_3 + 0 = 1.0.$$

После нахождения весов, находится значение в интерполируемой точке:

$$f_p = w_1 f_1 + w_2 f_2 + w_3 f_3.$$

При использовании вариограммы таким способом, ожидаемая ошибка минимизируется методом наименьших квадратов. Поэтому крайгинг получает лучшее линейное приближение. Однако минимизация ожидаемой ошибки не всегда наиболее значимый критерий и в некоторых случаях другие методы дают лучшие результаты (Philip & Watson, 1986).

Существует много разновидностей крайгинга. Перечислим некоторые из них:

Простой крайгинг является абсолютно таким же, как и нормальный, за исключением того, что к системе не добавляется следующее уравнение:

$$w_1 + w_2 + w_3 = 1.0.$$

В результате, простой крайгинг является менее точным методом, однако производит более гладкое приближение.

Универсальный крайгинг используется для уничтожения тренда между группами точек. Использование данных, имеющих тренд, дает худшие результаты, а такие возможности всегда существуют. Для этого строится простая полиномиальная функция, приближающая значения в исходных точках и на основе нее удаляется тренд между точками.

Зональный крайгинг основывается на разделении исходных точек на группы и построении для каждой зоны своей вариограммы. Зональный крайгинг может применяться только для регулярных сеток.

2.2. Программы, их характеристики, сравнение.

Рассмотрим программные средства управляемого решения одной из наиболее распространенных геологических задач - интерполяции данных и оценивания по интерполяционным поверхностям (гиперповерхностям) интегральных характеристик заданных площадей (объемов). Такими задачами являются автоматизированное картирование, увязка рудных интервалов, подсчет запасов и т.д. Аналогичные задачи встречаются и в негеологических отраслях знания.

2.2.1. Общий анализ имеющихся средств.

В настоящее время на рынке имеется большое количество программных средств рассматриваемого назначения: DataMine, BluePACK, GEOPACK, GEOEAS, Geostatistical Toolbox, TechBASE, Rockworks и некоторые менее распространенные. Имеется также большое число средств, отвечающих предметной области только на первый взгляд. Классический пример - пакет Surfer, который используется отечественными пользователями чрезвычайно широко, несмотря на полное отсутствие управления интерполяцией и механизмов оценивания объектов, отличных от точечных. Пакеты типа Surfer есть средства полностью автоматической интерполяции, не имеющие отношения к рассматриваемому классу задач.

Проанализировав имеющиеся программные средства с упором на эффективность эксплуатации, можно сразу сделать следующие выводы:

1. Имеющиеся программные средства достаточно жестко делятся на два класса:

средства, созданные для работы в пакетном режиме и далее адаптированные для интерактивной работы. Этот класс средств имеет развитую и мобильную математику, но интерфейс пользователя является "надстройкой сверху", не дающей возможности мобильного управления и оперативного эксперимента, что

блокирует эффективное использование без привлечения математика. Наиболее типичные примеры такого рода - программы Geostatistical Toolbox и DataMine;

средства нового поколения, изначально создаваемые для работы в интерактивной среде. Этот класс программных средств имеет весь спектр возможностей по управлению данными, процессом решения и представлением результатов, при этом управление ориентировано на практического пользователя с низким уровнем математической подготовки (StratiFact, GEOPACK, Surfer).

При этом практически отсутствуют "гибриды", которые объединяли бы достоинства этих двух классов и не имели их недостатков, так как есть соблазн использовать имеющиеся математические модули, а для получения реального интерактива их необходимо полностью переписать, что и трудоемко, и дорого.

2. Среди программных средств, эффективных в эксплуатации, длительно используются только средства, позволяющие пользователю вводить в интерполяционные процедуры свое видение объекта изучения, то есть средства, интерактивно моделирующие пространственную изменчивость геологических характеристик и строящие интерполяционные процедуры на основе этих моделей. Реально этому условию отвечают только программы, использующие математику линейной геостатистики и в полной мере реализующие интерактивную вариографию с моделированием иерархических структур изменчивости. Программные средства полностью автоматической интерполяции довольно быстро перестают устраивать пользователя. Если он решает задачи, прямо связанные с интерполяцией (картирование концентраций по опробованию), это происходит через год-полтора, а если интерполяция вспомогательна (например, построение мелкомасштабных карт прогноза, где основная содержательная нагрузка в алгоритмах распознавания образов), программа автоматической интерполяции (типа Surfer) может даже оказаться оптимальной.

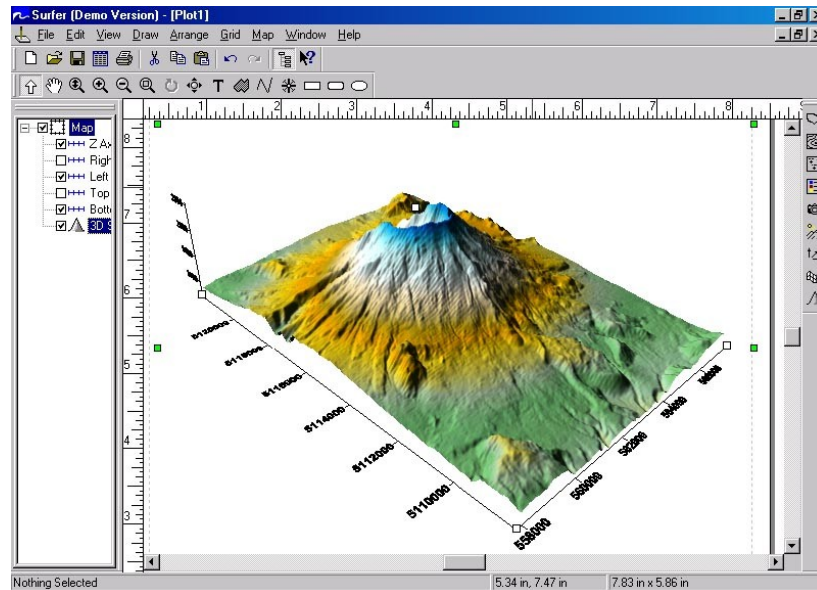
Во всех широко тиражируемых программных средствах данного направления, авторы программ ограничиваются тезисом о предпочтительности математики линейной геостатистики. Таким образом, наиболее популярные методы, реализованные в таких программах – это методы крайгинга, различные весовые методы. Геостатистика пока является единственным распространенным аппаратом управляемой интерполяции, но из этого никоим образом не следует, что альтернативы невозможны. Поясним, почему они не используются.

Пользователь, удовлетворяющий предложенной модели, будет искать физический смысл во всех предлагаемых ему математических моделях и построениях. Именно это блокирует широкое применение, например, нелинейной геостатистики. Для линейной каждое построение имеет физический смысл, если и не прямо применимый к объекту исследования, то хотя бы легко обобщаемый на него (например, в крайгинге сферическая вариограмма моделирует конкреции). Для нелинейных процедур все это остается непонятным, применение той или иной модели никакого физического содержания не имеет. Даже для используемой при ко-крайгинге кросс-вариограммы трудно найти физический смысл, не говоря уж о более труднопонижаемых методах. Это существенно сдерживает развитие пакетов, ориентированных на передовую математику нелинейной геостатистики. Их время придет только, когда авторы методик смогут объяснить физический смысл того моделирования, которым они предлагают заняться пользователю, несущему ответственность за результат. Единственная возможность их эффективного применения пока состоит в использовании только в научно-технических центрах, берущих ответственность за результат на себя. Тиражируемый же продукт, к сожалению, пока вынужден использовать ограниченную, но хорошо обоснованную математику.

Рассмотрим наиболее ярких представителей обоих классов программных средств двумерной интерполяции.

2.2.2. Пакет Surfer - обработка и визуализация двухмерных функций

Версия Surfer 5.0 для Windows 3.x была выпущена в 1994 году. Год спустя, одновременно с выходом Windows 95, вышел пакет Surfer 6.0, который был представлен двумя вариантами - 32-разрядным для работы в среде Windows NT и Windows 95 и 16-разрядным для Windows 3.1. В 2002 году была выпущена версия Surfer 8.0.



Основным назначением Surfer является обработка и визуализация двухмерных наборов данных, описываемых функцией типа $z=f(x, y)$. Логiku работы с пакетом можно представить в виде трех основных функциональных блоков: а) построение цифровой модели поверхности; б) вспомогательные операции с цифровыми моделями поверхности; в) визуализация поверхности.

Новая версия популярной программы создания трехмерных карт Surfer 8 позволяет обрабатывать огромные объемы данных и создавать реалистичные трехмерные карты с полным контролем освещенности и теней, накладывая при необходимости изображения местности в различных форматах, экспортировать их

в популярные графические форматы, в том числе форматы геоинформационных систем, а также выводить на цветную печать с размером до 50 м по диагонали.

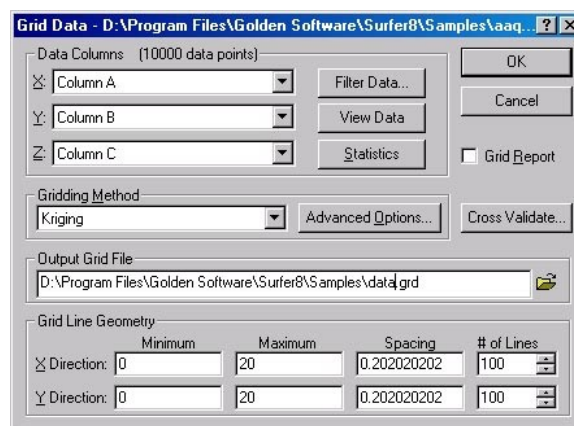
Этот программный комплекс позволяет создавать трехмерные поверхностные карты с полным контролем освещения и теней, положения, углов наклона и поворота. Surfer 8 производит наложение растровых и векторных карт, контурных карт, фотографий местности, карт рельефа местности с тенями, и задает изменение цвета (поверхности и функций наложения).

Существует функция для применения к данным сглаживающих фильтров. Она включает более 60 предустановленных фильтров. Кроме того, существует возможность определить собственные параметры фильтров.

С точки зрения математического аппарата построения поверхности очень важным представляется реализация еще одного алгоритма интерполяции – Ближайший сосед (Nearest Neighbor), а также трех уровней вложения вариограмм, что позволяет создавать более 500 результирующих комбинаций.

Созданные ранее изображения на основе различных типов карт (Contour Map, Shaded Relief Map, Post Map, Image Map) можно использовать в качестве шаблона путем подстановки в уже существующие карты нового файла. Кроме того, теперь, объединив в начале в одно изображение несколько слоев различных карт, можно затем разделить их на исходные элементы и переделать на основе новых данных.

Из чисто сервисных функций следует выделить возможность занесения



данных оцифровки линий границ и произвольных точек с экрана прямо в ASCII-

файл, а также автоматическое создание легенды для различных типов точек карты Post Map. В качестве цифровой модели поверхности можно импортировать файлы формата Digital Elevation Model (DEM) прямо из Internet (или любого другого источника информации). И, наконец, новые форматы экспорта данных позволяют сохранять изображения карт практически во всех растровых форматах (PCX, GIF, TIF, BMP, TGA, JPG и многих других).

Таким образом, пользовательский интерфейс и сервисные функции данного программного комплекса на высоте, а вот управления математическим аппаратом на «низком» уровне нет.

2.2.3. GMT – комплекс консольных приложений.

Программный комплекс GMT состоит из шестидесяти консольных приложений, каждое из которых решает определенную задачу. Такое разделение очень удобно, так как, во-первых, позволяет использовать скрипты в виде bat-файлов, а во-вторых, делает возможным контролировать процесс построения модели до мельчайших деталей. Каждая из таких программ имеет в среднем около двадцати различных параметров, передающихся через командную строку. Таким образом, существует возможность очень тонко контролировать процесс построения модели.

Программный комплекс оперирует следующими типами файлов: xyz-файлы, в которых содержится первоначальная информация для интерполирования, grd-файлы, в которых содержится полученная после интерполирования информация, ps-файлы – файлы формата «PostScript», предназначенные для вывода графической и текстовой информации, а также для дальнейшей ее печати. Данный комплекс содержит следующие основные программы:

Фильтрация 1-D и 2-D данных:

blockmean, blockmedian, blockmode – удаление трендов;

filter1d –фильтрация одномерных данных;

grdfilter –фильтрация двумерных данных;

Графический вывод 1-D и 2-D данных:

grdcontour – программа построения изолиний;

grdimage, grdvector, grdview- программы трехмерной визуализации;

pshistogram – построение гистограмм;

psrose – построение секторных диаграмм;

pstext –вывод текста;

psxy, psxyz – вывод символов, многоугольников и линий в 2-D и 3-D соответственно;

Интерполирование (x,y,z) данных:

nearneighbor – интерполяция методом ближайшего соседа;

surface – алгоритм интерполяции непрерывной поверхностью;

triangulate – производит оптимальную триангуляцию Делоне;

minmax – нахождение экстремальных значений в файле с данными;

Конвертирование данных:

gmtmath –калькулятор в обратной польской нотации;

grd2xyz – конвертирование grd-файла в xyz-файл;

grdcut – удаление подрегиона из grd-файла;

grdpaste – склеивание двух grd-файлов;

splitxyz – деление xyz-файла на несколько сегментов;

xyz2grd - конвертирование данных из xyz-файла в grd-файл;

Определение трендов в 1-D и 2-D данных:

fitcircle, grdtrend, trend1d, trend2d;

Пример скрипта программного комплекса GMT:

gmtset GRID_PEN 0.25pta

```

psxy table_5.11 -R0/7/0/7 -JX3.06i/3.15i -B2f1WSNe -Sc0.05i -G0 -P -K -Y6.45i >
example.ps
blockmean table_5.11 -R0/7/0/7 -I1 > mean.xyz
surface mean.xyz -R0.5/7.5/0.5/7.5 -I1 -Gdata.grd
grdcontour data.grd -JX3.06i/3.15i -B2f1WSne -C25 -A50 -G3i/10 -S4 -X2.25i -
Y4.55i> example.ps
grdcontour data.grd -JX3.06i/3.15i -B2f1WSne -C3000 -A50 -G3i/10 -S4 -X2.25i -
Y4.55i> example.ps

```

Этот скрипт строит интерполированную поверхность и помещает сетку в файл data.grd при помощи подпрограммы surface. Предварительно данные проверяются на наличие тренда и ненужные значения удаляются. Затем строится контурная карта при помощи модуля grdcontour и результат помещается в файл example.ps.

Очевидно, что такой программный комплекс серьезно подходит для специалистов с хорошей математической подготовкой, однако для простого пользователя он абсолютно не подходит из-за сложности в его освоении и использовании.

Проблема, прежде всего, состоит в том, что математический аппарат не дает прямого ответа на то, какие методы лучше всего использовать. Поэтому пользователю приходится экспериментировать, применяя различные методы и их настройки. Очевидно, что при использовании программ со слабо выраженными возможностями контроля математических алгоритмов, наступает момент, когда пользователь понимает, что возможности программы ограничены для решения его задачи из-за нехватки пространства для эксперимента.

3. ПРОГРАММНЫЙ КОМПЛЕКС ПРИБЛИЖЕНИЯ ФУНКЦИЙ КУБИЧЕСКИМИ СПЛАЙНАМИ

3.1. Описание алгоритмов

В процессе работы был создан программный комплекс, решающий следующие задачи:

- построение триангуляции Делоне;
- реализован метод интерполяции кубическими сплайнами на полученной сетке;
- реализован метод построения поля изолиний для визуализации полученных результатов (построение контурной карты).

3.1.1. Построение оптимальной нерегулярной сетки.

Приближение функций полиномиальными сплайнами тесно связано с задачей построения сетки - разбиения рассматриваемой области на непересекающиеся подмножества. В одномерном случае в качестве узлов сетки, как правило, выбирают точки, в которых задана информация, а элементами разбиения являются отрезки прямых между ними. Двумерная задача построения сетки, особенно при хаотическом расположении информационных точек, оказывается значительно сложнее. Вместе с тем, разбиение области может в значительной степени определять качество приближения функций и, в связи с этим встает вопрос о выборе критерия их оптимальности и методах построения таких сеток.

Пусть на плоскости в ограниченной замкнутой области W задан набор N несовпадающих точек

$$W = \{w_i = (x_i, y_i); i = 1, \dots, N\}$$

причем N_b из них являются внутренними, а N_r - лежат на границе области G^0 . Будем предполагать, что границей области G является ломаная линия, не имеющая

самопересечений, причем концы ее прямолинейных отрезков являются точками набора W . Без ограничения общности можно считать, что внутренние точки (если есть такие) имеют номера $1, \dots, N_b$, а граничные – N_b+1, \dots, N_b+N_r . Пусть далее в точках набора W заданы значения функции

$$Z = \{z_i = f(w_i); w_i \in W, i = 1, \dots, N\}$$

удовлетворяющей условию Липшица: существует константа C , такая, что для любых $W', W'' \in G$ выполняется неравенство

$$|f(W') - f(W'')| \leq C \|W' - W''\| \quad (I)$$

Ставится задача построения в G наилучшей интерполирующей функции f некоторого класса Φ .

Алгоритмы построения оптимальных сеток $\Sigma(W, F, \Phi^0)$ и $\Sigma(W, F, \Phi^1)$ могут быть реализованы путем перебора всех возможных разбиений, оценки их качества по соответствующим критериям и выбора оптимального из них. Однако такой подход требует большого количества переборов. В случае нахождения триангуляции Делоне имеется возможность построения эффективных алгоритмов, что основывается на свойствах данного разбиения.

В программе был использован алгоритм, предложенный в [2].

Смысл метода заключается в том, что построение триангуляции Делоне эквивалентно выбору из множества окружностей, проходящих через три точки набора W , такого подмножества, что любая окружность не содержит внутри себя других точек. Принадлежность точек одной окружности определяет соседство точек.

При построении такой системы окружностей на некоторых из них может оказаться более одной точки. В этом случае все разбиения образованного ими многоугольника диагоналями оказываются эквивалентными.

Пусть построено некоторое разбиение $\Sigma(W, F, \Phi^0)$ и соответствующая ему триангуляция. Алгоритм построения оптимального разбиения может быть

реализован следующим образом: будем рассматривать каждый четырехугольник, образованный соседними треугольниками (имеющими общей границей одну из сторон), и если его разбиение диагональю не является оптимальным, то изменим его.

Утверждение 2. Триангуляция, каждый четырехугольник (образованный соседними треугольниками), который разбит наилучшим образом, является триангуляцией Делоне [2].

Если для данного набора точек триангуляция Делоне определяется неоднозначно, то в результате применения такого подхода будет получена одна из них. Например, если точки набора лежат на одной окружности, то все триангуляции будут эквивалентными.

Оптимальное разбиение четырехугольника на треугольники может быть реализовано путем выбора такого, у которого максимальный радиус описанных окружностей минимален.

Рассмотренные выше утверждения позволяют построить эффективный алгоритм, не требующий полного перебора всех допустимых триангуляций. Исходной информацией для алгоритма являются координаты точек набора W и список тех из них, которые являются граничными. Предполагается, что граничные точки в списке соответствуют положительному обходу области G .

Алгоритм может быть успешно применен и к триангуляции неодносвязных областей. В этом случае достаточно ввести дополнительные точки в список граничных так, чтобы "разрезать" эту область. После выполнения триангуляции остается только "удалить" дополнительные точки и "склеить" триангуляцию. Результатом работы алгоритма является триангуляция области, информацию о которой мы храним в виде двух троек чисел для каждого треугольника. Первая тройка – это номера точек, образующие треугольник, а вторая – номера соседних треугольников. Вторая тройка чисел является избыточной информацией, но она

используется как для ускорения работы алгоритма триангуляции, так и при реализации алгоритмов приближения функций.

Остановимся на основных этапах алгоритма,

Этап I. На этом этапе производится построение триангуляции Σ_r – области G без учета внутренних точек, то есть для набора $W_r = \{w_i \in W : i = N_{B+1}, \dots, N_B + N_R\}$.

Этап II. Построение триангуляции для всего набора точек может быть выполнено последовательным добавлением внутренних точек. При этом каждый раз производится преобразование триангуляции следующим образом: в ней находится треугольник S_{ijk} , в который попадает добавляемая точка e , и вместо S_{ijk} вводятся треугольники S_{ije} , S_{jke} и S_{kie} .

Остановимся на двух моментах, позволяющих повысить эффективность работы алгоритма на этом этапе. Во-первых, для описания треугольника, в который попадает добавляемая точка, наиболее простым оказывается анализ площадей треугольников: точка принадлежит треугольнику S_{ijx} , если площади S_{ije} , S_{jke} и S_{kie} положительные. Здесь предполагается, что нумерация вершин в треугольниках всегда имеет положительное направление.

Во-вторых, поиск такого треугольника оказывается возможным выполнять целенаправленно. Если для некоторого треугольника S_{ijk} площадь S_{ije} отрицательна, то следующий треугольник для анализа выберем соседний относительно стороны ij . В качестве первоначального треугольника целесообразно выбирать один из тех, что были построены на предыдущем шаге. Это связано с тем, что точки с близкими номерами в W находятся, как правило, недалеко друг от друга. При сложной конфигурации области G такой поиск треугольника может привести в тупик и в этом случае приходится прибегать к полному перебору.

Этап III. В соответствии с утверждением 2 для построения триангуляции Делоне из полученного на предыдущем этапе разбиения необходимо выполнить

цепь последовательных преобразований. То есть рассмотреть все четырехугольники и изменить их разбиение, если это необходимо. Очевидно, что одного просмотра для этого может оказаться недостаточно. Фиксируя все треугольники, на которые оказывает влияние изменение способа разбиения, (то есть соседние), можно и здесь сократить объем вычислений.

При выполнении этапа III будет построена триангуляция Делоне и, следовательно, оптимальное разбиение

Процесс оптимизации (этап III) может выполняться также после этапа I и каждый раз после добавления очередной внутренней точки. При хаотическом расположении точек на плоскости каждое новое добавление не будет, как правило, вызывать перестройки всей триангуляции. То есть добавление точки будет вносить локальные изменения в триангуляцию.

Построение оптимальных триангуляции $\Sigma(W, F, \Phi^{1,2,3})$ может выполняться по этому же алгоритму, но на этапе III при этом необходимо применять соответствующие критерии оптимизации,

3.1.2. Построение интерполирующих функций.

Вкратце напомним постановку задачи построения интерполирующего кубического сплайна.

Пусть на плоскости в области G задано N несовпадающих точек

$$W = \{w_i = (x_i, y_i) : i = 1, \dots, N\}$$

причем N_r из них определены как граничные, а остальные N_b как внутренние. Без ограничения общности будем считать, что внутренние точки имеют номера $1, 2, \dots, N_b$. Пусть для области G задана триангуляция:

$\Sigma(W) = \{S_m, m = 1, \dots, M\}$ – разбиение на замкнутые подобласти, которые являются треугольными элементами с вершинами из W и такие, что $\bigcup_{m=1}^M S_m = G$, а

$S_m \cap S_n$ при $m \neq n$ есть отрезок прямой (соседние треугольники), точка, или пустое множество.

Двумерным сплайном степени α дефекта β на триангулированной сетке Σ области G будем называть функцию $\varphi(x, y)$, являющуюся на каждом элементе многочленом двух $S_m \in \Sigma$ переменных x и y степени α и имеющую в G непрерывные производные до порядка $\alpha - \beta$ включительно (то есть $\varphi(x, y) \in C^{\alpha - \beta}$).

Ставится задача построения функции $\varphi(x, y)$:

$$\varphi(x, y) = \{\varphi^m(x, y) : (x, y) \in S_m, S_m \in \Sigma(W)\}$$

$$\varphi^m(x, y) = a_0^m + a_1^m x + a_2^m y + a_3^m x^2 + a_4^m xy + a_5^m y^2 + a_6^m x^3 + a_7^m x^2 y + a_8^m xy^2 + a_9^m y^3,$$

приближающей функцию f в точках W_i .

Интерполянт должен удовлетворять условиям

$$\varphi(W_i) = \varphi(x_i, y_i) = f_i, \quad w_i \in W.$$

Кроме того, вводятся условия непрерывности первой и второй производной на стыках сплайнов. Для двух соседних треугольников построенные сплайны должны удовлетворять условиям:

$$\varphi^1(x, y) = \varphi^2(x, y);$$

$$\frac{\partial \varphi^1}{\partial x}(x, y) = \frac{\partial \varphi^2}{\partial x}(x, y);$$

$$\frac{\partial \varphi^1}{\partial y}(x, y) = \frac{\partial \varphi^2}{\partial y}(x, y);$$

$$\frac{\partial^2 \varphi^1}{\partial x^2}(x, y) = \frac{\partial^2 \varphi^2}{\partial x^2}(x, y);$$

$$\frac{\partial^2 \varphi^1}{\partial y^2}(x, y) = \frac{\partial^2 \varphi^2}{\partial y^2}(x, y);$$

$$\frac{\partial^2 \varphi^1}{\partial xy}(x, y) = \frac{\partial^2 \varphi^2}{\partial xy}(x, y);$$

$$(x, y) \in d_{ik}.$$

Уравнение прямой d_{ik} запишем в параметрической форме:

$$x = \beta t, \quad y = \gamma t - b.$$

При переходе от переменных x и y к переменной t получаем систему уравнений.

Из равенства значений функций на ребре d_{ik} :

$$a_0^1 - a_2^1 b + a_5^1 b^2 - a_9^1 b^3 = a_0^2 - a_2^2 b + a_5^2 b^2 - a_9^2 b^3$$

$$a_1^1 + a_2^1 \gamma - a_4^1 b - 2a_5^1 \gamma b + a_8^1 b^2 + 3a_9^1 \gamma b^2 = a_1^2 + a_2^2 \gamma - a_4^2 b - 2a_5^2 \gamma b + a_8^2 b^2 + 3a_9^2 \gamma b^2$$

$$a_3^1 + a_4^1 \gamma + a_5^1 \gamma^2 - a_7^1 b - 2a_8^1 \gamma b - 3a_9^1 \gamma^2 b = a_3^2 + a_4^2 \gamma + a_5^2 \gamma^2 - a_7^2 b - 2a_8^2 \gamma b - 3a_9^2 \gamma^2 b$$

$$a_6^1 b + a_7^1 \gamma + a_8^1 \gamma^2 + a_9^1 \gamma^3 = a_6^2 b + a_7^2 \gamma + a_8^2 \gamma^2 + a_9^2 \gamma^3$$

Из равенства первых производных по x на ребре d_{ik} :

$$a_1^1 - a_4^1 b + a_8^1 b^2 = a_1^2 - a_4^2 b + a_8^2 b^2$$

$$2a_3^1 + a_4^1 \gamma - 2a_7^1 b + 2a_8^1 \gamma b = 2a_3^2 + a_4^2 \gamma - 2a_7^2 b + 2a_8^2 \gamma b$$

$$3a_6^1 + 2a_7^1 \gamma + a_8^1 \gamma^2 = 3a_6^2 + 2a_7^2 \gamma + a_8^2 \gamma^2$$

Из равенства первых производных по y на ребре d_{ik} :

$$a_2^1 - 2a_5^1 b + 3a_9^1 b^2 = a_2^2 - 2a_5^2 b + 3a_9^2 b^2$$

$$a_4^1 + 2a_5^1 \gamma - 2a_8^1 b - 6a_9^1 \gamma b = a_4^2 + 2a_5^2 \gamma - 2a_8^2 b - 6a_9^2 \gamma b$$

$$a_7^1 + 2a_8^1 \gamma + 3a_9^1 \gamma^2 = a_7^2 + 2a_8^2 \gamma + 3a_9^2 \gamma^2$$

После различных преобразований, приведения подобных слагаемых и сложения различных уравнений между собой, можно получить следующую систему (1):

$$a_0^1 - a_2^1 b + a_5^1 b^2 - a_9^1 b^3 = a_0^2 - a_2^2 b + a_5^2 b^2 - a_9^2 b^3$$

$$a_2^1 - 2a_5^1 b + 3a_9^1 b^2 = a_2^2 - 2a_5^2 b + 3a_9^2 b^2$$

$$a_3^1 + a_4^1 \gamma + a_5^1 \gamma^2 = a_3^2 + a_4^2 \gamma + a_5^2 \gamma^2$$

$$a_6^1 b + a_7^1 \gamma + a_8^1 \gamma^2 + a_9^1 \gamma^3 = a_6^2 b + a_7^2 \gamma + a_8^2 \gamma^2 + a_9^2 \gamma^3$$

$$a_1^1 - a_4^1 b + a_8^1 b^2 = a_1^2 - a_4^2 b + a_8^2 b^2$$

$$2a_3^1 + a_4^1 \gamma - 2a_7^1 b + 2a_8^1 b = 2a_3^2 + a_4^2 \gamma - 2a_7^2 b + 2a_8^2 b$$

$$a_4^1 + 2a_5^1 \gamma - 2a_8^1 b - 6a_9^1 \gamma b = a_4^2 + 2a_5^2 \gamma - 2a_8^2 b - 6a_9^2 \gamma b$$

$$a_7^1 + 2a_8^1 \gamma + 3a_9^1 \gamma^2 = a_7^2 + 2a_8^2 \gamma + 3a_9^2 \gamma^2$$

Эту систему можно решать различными методами.

В программе она решается двумя способами. Первый способ заключается в использовании метода Гаусса, а второй – в использовании разработанного мною алгоритма. Опишем подробнее эти методы.

При решении системы методом Гаусса основной проблемой является наличие линейно-зависимых уравнений. То есть в процессе решения необходимо пометать их, а затем не использовать в процессе работы. Поскольку уравнений меньше, чем неизвестных, то какое-то количество неизвестных будут являться свободными. Для нахождения коэффициентов нужно использовать какие-то дополнительные условия. Поскольку при выводе системы, условия неразрывности вторых производных мы опускали (так как при этих условиях система становится в общем случае несовместной), то логично использовать условия, связанные с минимизацией разрывов вторых производных на границах сплайнов.

В программе были использованы условия минимизации суммы квадратов разрывов вторых производных. Однако они не добавлялись в общую систему, так как это значительно увеличило бы количество уравнений и объем используемой памяти. Для минимизации разрывов вторых производных использовался следующий метод.

После прямого хода метода Гаусса, выбираются свободные переменные: $a_{x1}^{n1}, a_{x2}^{n2}, \dots, a_{xt}^{nt}$. Затем в процессе обратного хода метода Гаусса все остальные переменные выражаются через свободные. Для каждой переменной составляется массив коэффициентов d_{it}^j :

$$a_i^j = \sum_t d_{it}^j a_{xt}^{nt}.$$

Затем на каждом треугольном элементе разбиения выражается вторая производная сплайна через свободные переменные $f_m'' = \sum_t c_{it}^j a_{xt}^{nt}$.

Далее сумма квадратов разрывов вторых производных также выражается через свободные переменные. Для того чтобы найти значения переменных, при которых функционал достигает минимума, нужно производную функционала по каждой из переменных приравнять к нулю. Получается система для нахождения свободных переменных, которая решается также методом Гаусса.

Недостаток такого решения системы заключается в следующем:

- используется большое количество оперативной памяти ($\sim 400N^2$, где N - количество узлов интерполяции);
- в процессе использования метода Гаусса при делении возникают довольно большие погрешности;
- метод работает очень медленно по сравнению с геостатистическими методами;

Для частичного устранения этих недостатков мною был разработан метод, позволяющий снизить объем используемой памяти до $\sim 56N^2$, а также снизить время выполнения метода. Очень важным свойством нового метода также является отсутствие деления в вычислениях, поэтому искомые значения коэффициентов получаются практически без погрешности.

Идея метода заключается в преобразовании уравнений системы (3) к такому виду, чтобы любые восемь переменных были выражены через оставшиеся две. В данном случае мы выразили все переменные через переменные a_5^i и a_9^i для каждого ребра ($i=1..M$). Получилась система (4):

$$a_0^1 - a_5^1 b^2 + 2a_9^1 b^3 = a_0^2 - a_5^2 b^2 + 2a_9^2 b^3$$

$$a_2^1 - 2a_5^1 b + 3a_9^1 b^2 = a_2^2 - 2a_5^2 b + 3a_9^2 b^2$$

$$a_3^1 - a_5^1 y^2 + 6a_9^1 y^2 b = a_3^2 - a_5^2 y^2 + 6a_9^2 y^2 b$$

$$a_6^1 b - 2a_9^1 y^3 = a_6^2 b - 2a_9^2 y^3$$

$$a_1^1 + 2a_5^1 y b - 6a_9^1 y b^2 = a_1^2 + 2a_5^2 y b - 6a_9^2 y b^2$$

$$a_4^1 + 2a_5^1 y - 6a_9^1 y b = a_4^2 + 2a_5^2 y - 6a_9^2 y b$$

$$a_7^1 + 3a_9^1 y^2 = a_7^2 + 3a_9^2 y^2$$

$$a_8^1 + 3a_9^1 y^2 = a_8^2 + 3a_9^2 y^2$$

Теперь мы можем выразить любую переменную (кроме a_5^i и a_9^i) через переменные a_5^i и a_9^i и переменные какого-то соседнего треугольника (вариант, когда $N=3$, считаем тривиальным и не учитываем).

Далее выбираем любой треугольник и в нем считаем все переменные известными. После этого выражаем переменные соседних треугольников через них и вновь добавившиеся свободные переменные на этом треугольнике a_5^i и a_9^i . Продолжаем процесс на треугольники, смежные с данными, не заходя в те треугольники, в которых мы уже были, и добавляя на каждом элементе разбиения две новые переменные. В результате все переменные выражены через десять переменных первого треугольника и две переменные с каждого другого треугольника. Таким образом, остается примерно $10+2*(2N-1)$ переменных (при большом количестве точек количество треугольников приблизительно равно N). Соответственно, количество переменных уменьшается примерно в 5 раз.

Условия непрерывности первых переменных на многих ребрах мы уже использовали, но не на всех. Теперь составляем систему, которая будет состоять из уравнений непрерывности первых производных на неиспользованных ребрах, а также условия равенства сплайнов в исходных точках.

Также как и в первом методе, после решения системы остаются свободные переменные, через которые выражаются все остальные. Затем через свободные переменные выражается функционал (равный сумме квадратов разрывов вторых производных) и минимизируется тем же способом, что и в первом методе.

Как уже было сказано, преимущество данного метода ощутимо, так как количество переменных в пять раз меньше, а также не используется деление.

Оба метода были реализованы в программе и дают схожие результаты, однако в первом методе возникает и быстро накапливается погрешность, второй же дает очень точные результаты (погрешность возникает в восьмом знаке после запятой). Тесты проводились для количества точек около 100 для различных аналитически заданных функций.

3.1.3. Визуализация полученных результатов.

Одним из широко распространенных и удобных методов описания поведения функций двух переменных является графическое изображение поверхностей в виде полей изолиний. Особенно часто такое изображение используется в задачах геологии, геофизики и разработки нефтяных и газовых месторождений. Применение для этих целей устройств графического изображения информации ЭВМ – графопостроителей и графических дисплеев требует вычисления с достаточно маленьким шагом координат точек, лежащих на изображаемых кривых.

Как правило, получение явных аналитических выражений для линий уровня, которые должны быть нанесены на карту, оказывается невозможным. Традиционным подходом при численном решении этой задачи является вычисление значений функции в узлах прямоугольной сетки и анализ значений в вершинах ее четырехугольных элементов. Однако здесь, как и ранее возникают определенные трудности при работе с областями сложной конфигурации. Кроме того, для обеспечения достаточной точности в этом случае приходится производить вычисление функции в значительном количестве точек, хотя количество изображаемых изолиний может быть и невелико. С другой стороны, при недостаточно мелкой сетке в некоторых случаях поведение кривой

определяется неоднозначно. В [2] был предложен алгоритм, позволяющий в значительной степени снизить объем вычислений, необходимый для построения полей изолинии. Ниже будем называть процесс вычисления координат точек изолиний в последовательности, которой они должны быть соединены, прослеживанием изолиний.

Пусть в области G , границей которой является ломанная, задана функция двух переменных $f(x, y)$ и требуется в этой области проследить изолинию со значением z_0 , то есть найти координаты точек, лежащих на кривой $z_0 = f(x, y)$.

Без ограничения общности можно считать $z_0 = 0$.

Вычислительный процесс начинается с обхода (например, в положительном направлении) области & по границе с достаточно малым шагом h , вычислением значений функции в точках, лежащих на ней и анализе выражения $f(x_a, y_a) * f(x_b, y_b)$, где точки A и B с координатами (x_a, y_a) и (x_b, y_b) лежат на границе G и расстояние между которыми равно h . Отрицательные значения этого выражения означают, что на этом отрезке изолиния пересекает границу. Координаты этой точки (которая является первой точкой изолинии) может быть приближенно найдена по формулам линейного интерполирования или с помощью некоторого более точного метода. Итак, пусть нами найден отрезок AB и первая точка изолинии F_1 . Рассмотрим равносторонний треугольник с основанием AB и третьей вершиной C , координаты которой могут быть вычислены по формулам:

$$x_c = (x_b + x_a + \sqrt{3}(y_a - y_b))/2$$

$$y_c = (y_b + y_a + \sqrt{3}(x_a - x_b))/2$$

Если шаг h достаточно мал, то внутри этого треугольника содержится только один связный кусок изолинии, и точка C будет лежать внутри области G . В зависимости от значения функции в точке другой конец изолинии (точка F_2) располагается на отрезках AB или BC . Будем считать, что точка F_2 лежит на

отрезке BO . Тогда на этом отрезке построим новый равносторонний треугольник BDC , координаты точки D вычисляются по формулам:

$$x_d = x_b + x_c - x_a$$

$$y_d = y_b + y_c - y_a$$

Для полученного треугольника описанная процедура повторяется до тех пор, пока изолиния не пересечет второй раз границу области (таким образом будет получена последовательность точек F_1, F_2, \dots, F_N , последовательно расположенных на изолинии). Описанный алгоритм позволяет проследивать изолинии, пересекающие границу области. Однако с небольшой модификацией он применяется и для замкнутых линий уровня. Для этого можно, например, границу области G дополнить отрезками, лежащими внутри ее или использовать в качестве ломанных ребра триангулированной сетки.

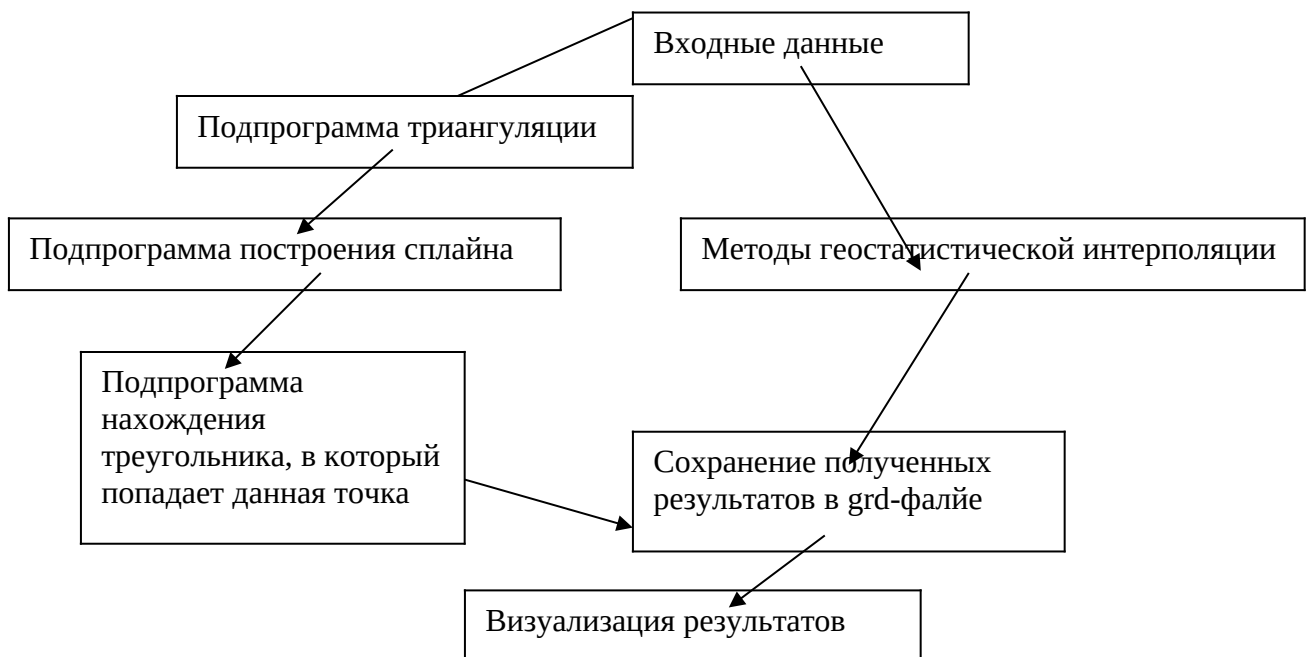
Данный алгоритм был реализован мною в программном комплексе.

3.2. Описание программного комплекса

Используя вышеописанные алгоритмы, реализованный программный комплекс решает следующие задачи:

- построение нерегулярной триангулированной сетки;
- нахождение коэффициентов сплайна;
- нахождение значения полученного сплайна в любой точке интерполируемой области;
- визуализация полученной информации путем построения поля изолиний;
- возможность экспорта всей получаемой информации в файлы соответствующих форматов.

В процессе проектирования системы, с учетом вышеописанных недостатков других программных комплексов, была построена следующая модель.



После построения модели будущего приложения возникла проблема выбора средств разработки. Существовало два основных варианта: Microsoft Visual Studio 6.0, либо Borland Delphi 6.0. Рассмотрим все плюсы и минусы данных средств.

3.2.1. Средства разработки

Проведем краткое сравнение средств разработки приложений Microsoft Visual Studio 6.0 и Borland Delphi 6.0.

Borland Delphi 6.0 является прекрасным средством для тех, кто хочет быстро и легко создать Windows-приложение со стандартным интерфейсом. За несколько минут при помощи одной лишь мышки, можно создать окно приложения с огромным выбором различных управляющих элементов. Однако при создании собственных графических компонентов возникают трудности, так как среда все же для этого не предназначена.

С другой стороны, Microsoft Visual Studio позволяет легко и естественно создавать новые управляющие элементы, однако при использовании стандартного интерфейса, возникает намного больше проблем по сравнению с использованием Delphi.

Такая субъективная вещь, как пользовательский интерфейс, традиционно на высоте у продукта компании Microsoft и оставляет желать лучшего у продукта компании Borland. В Delphi совершенно неудобные средства отладки: окно просмотра значений переменных (Watch), дерево процедурных вызовов (рекурсии) если где-то и есть, то мне его, к сожалению, обнаружить не удалось, отсутствует возможность отправки текстовых сообщений из программы при отладке. В Microsoft Visual Studio 6.0 все это реализовано на высоком уровне.

В Delphi технология объектно-ориентированного программирования является «лишней», так как основными принципами программирования в этой среде являются идеи модульного программирования, пришедшие из более ранних версий продукта. Кроме того, даже сами эти принципы полностью не реализованы. Так, например, нельзя породить объект от двух предков. Такая возможность существует в Microsoft Visual Studio.

Кроме того, Microsoft Visual Studio создает намного более компактные по размеру и быстрые по скорости приложения. А скорость нередко является основным критерием для работы программ.

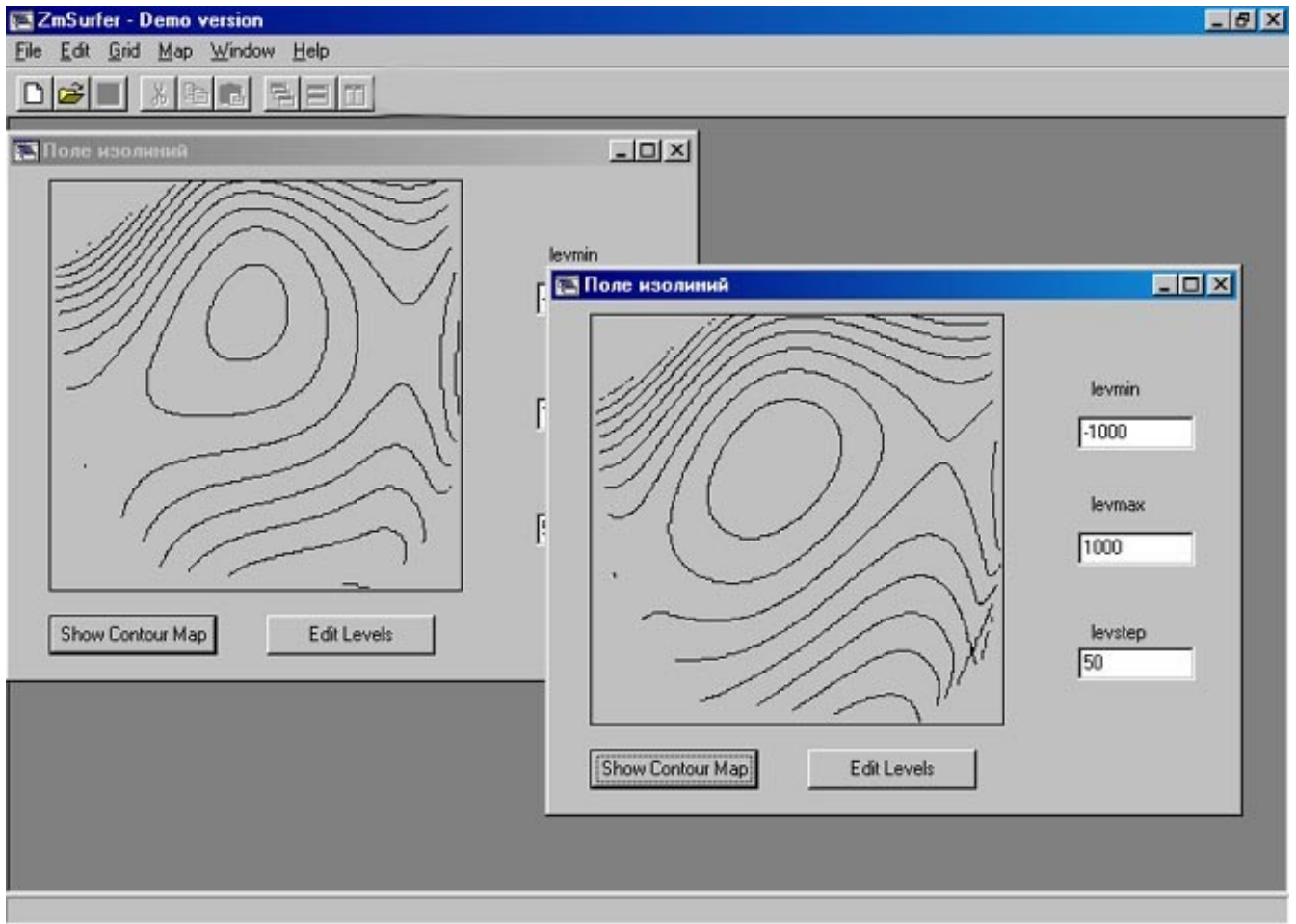
Подводя итог, опишем основные черты вышеописанных средств разработки Windows-приложений. В Delphi существует возможность быстрого и простого создания пользовательского интерфейса, но в то же время тяжелая отладка, неудобная работа, долгое создание собственных компонентов, которые в итоге получаются очень медленными. В Visual Studio существует возможность создания легких и быстрых программ, удобная работа и отладка, однако долго и тяжело делать пользовательский интерфейс.

При создании серьезных коммерческих продуктов использование Microsoft Visual Studio является практически стандартом. Однако в моем случае я все-таки выбрал среду Borland Delphi 6.0, для того чтобы снизить время разработки пользовательского интерфейса.

3.2.2. Описание реализованного программного средства

В процессе работы было создано MDI-приложение, для того чтобы существовала возможность одновременного отображения нескольких контурных карт.

Функциональные возможности приложения делятся на 3 части: триангуляция, интерполяция и визуализация. Результаты любой из этих частей можно сохранить в файл и использовать в дальнейшей работе. Результаты интерполирования хранятся в виде таблицы значений функции в узлах выбранной заранее сетки. Таблица хранится в простом текстовом файле тройками чисел x , y , z . Это позволяет использовать данный формат для визуализации любыми другими программами аналогичного характера.



Обычная последовательность действий при работе с программным комплексом состоит в следующем. В начале создается файл с данными в текстовом формате. Затем производится процесс интерполяции. Кроме метода интерполирования кубическими сплайнами, реализован также метод линейной интерполяции и метод Шепарда. Результаты интерполирования сохраняются в текстовый файл. Далее запускается процесс визуализации. Его результат можно сохранить в файле формата bmp.

Также существует возможность сохранения в файл результатов триангуляции в виде троек чисел в каждой строке, которые обозначают набор вершин, принадлежащих какому-то отдельному треугольнику.

рис.2. Окно интерполяции.

Grid Data - D:\zm\diplom\mywork\zmserfer_gui\mean.xyz

Data columns

X: Column A

Y: Column B

Z: Column C

☒ Grid Report

Gridding Method

Shepard's

Output Grid File

D:\zm\diplom\mywork\zmserfer_gui\mean.grd

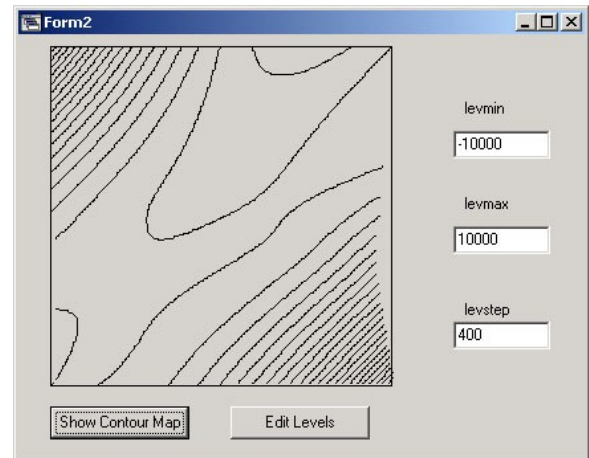
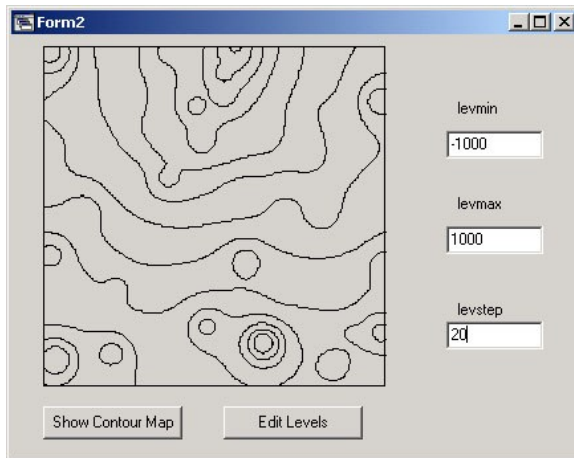
Grid Line Geometry

	Minimum	Maximum	Spacing	# of Lines
X Direction	0,2	6,3	0,061616161616	100
Y Direction	0	6,2	0,062626262626	100

В процессе интерполяции можно выбрать метод интерполяции, файл для сохранения результатов, ограничения на вводимые координаты, количество узлов в выводимой сетке. Также можно отфильтровать данные по различным критериям.

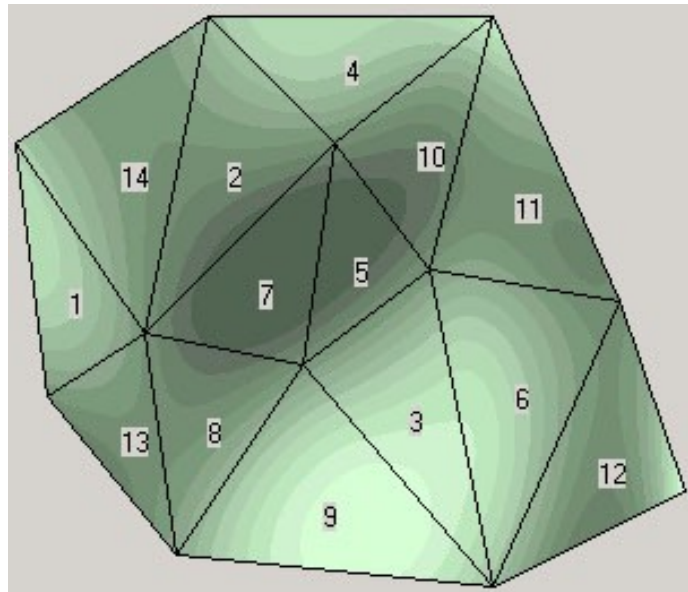
После построения сетки, можно производить над ней различные действия. Реализована возможность применения к ней простого девятиточечного сглаживающего фильтра для удаления случайно возникших возмущений.

В процессе визуализации можно задавать список отображаемых на контурной карте уровней.

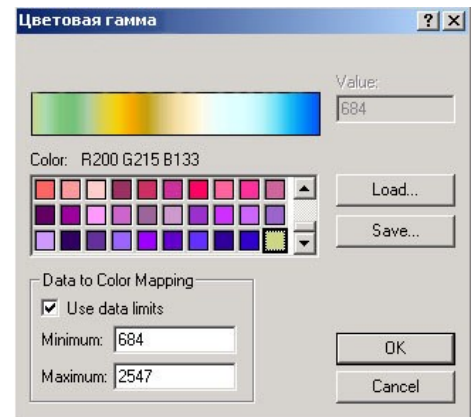
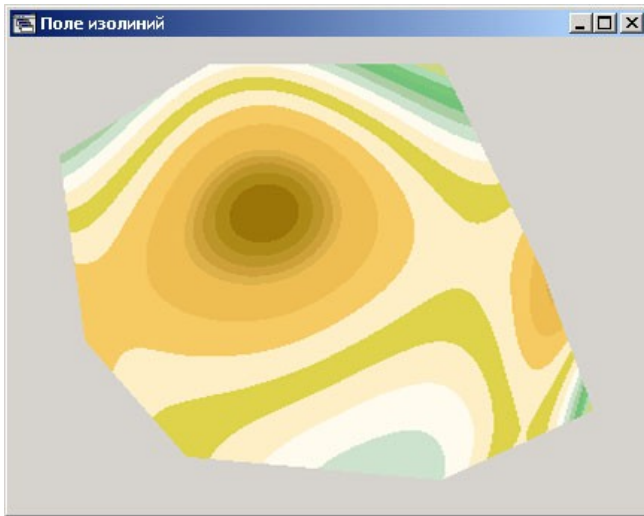


Существует возможность визуализации триангуляции и вывода номеров соответствующих треугольников. При этом можно также заполнять площадь между уровнями каким-либо цветом.

Можно выбрать автоматическую раскраску уровней.



Существует возможность выбора цветов при раскраске уровней. Для этого используется соответствующее диалоговое окно.



3.2.3. Программная реализация

Программа состоит из трех модулей. 1) построение нерегулярной сетки, 2) построение интерполирующего сплайна, 3) построение изолиний.

Создан класс CWTriangleGrid, предоставляющий методы для построения триангулированной сетки.

Список public-методов:

+LoadSetOfPoints(TPoint *points)	Загружает список точек на плоскости со значениями функции в этих точках.
+SetMethod(METHOD methodtype)	Устанавливает метод нахождения нерегулярной сетки
+CreateGrid()	Создает триангулированную сетку.

+GetGrid(TGrid *grid)	Загружает найденную сетку в особом формате в массив.
+SaveGridToFile(CString filename)	Сохраняет найденную сетку в особом формате в файл.

Для построение интерполирующего сплайна создан класс CWSpline, предоставляющий методы для построения сплайна.

Список public-методов:

+LoadGrid(TGrid *grid)	Загружает триангулированную сетку.
+CreateSpline()	Создает сплайн.
+GetSpline (TSpline *spline)	Загружает найденный сплайн в особом формате в массив.
+SaveSplineToFile(CString filename)	Сохраняет найденный сплайн в файл в виде сетки.

Для построение изолиний создан класс CWIsoline, визуализирующий построение изолиний. Порожден от класса CWnd.

Список public-методов:

- BOOL visualize	Флаг, определяющий, визуализировать загруженный сплайн или нет.
+LoadSpline(TSpline *spline)	Загружает сплайн.

+Visualize(BOOL true)	Устанавливает переменную visualize.
-----------------------	-------------------------------------

Применение объектно-ориентированного программирования существенно упрощает создание больших программных комплексов.

Заключение

Большое число задач нефтяной геологии, разведочной геофизики и метеорологии требуют применения методов приближения функций двух переменных. Одним из подходов, применяемых к решению таких задач, является сплайн-интерполирование.

В данной работе рассмотрены алгоритмы построения нерегулярных триангулированных сеток, алгоритмы сплайн-интерполирования, алгоритмы визуализации полученных данных.

Рассматривается задача приближения функций двумерными сплайнами третьей степени. Ставится задача построения функции $\phi(x, y)$:

$$\phi(x, y) = \{\phi^m(x, y) : (x, y) \in S_m, S_m \in \sum(W)\}$$

$$\phi^m(x, y) = a_0^m + a_1^m x + a_2^m y + a_3^m x^2 + a_4^m xy + a_5^m y^2 + a_6^m x^3 + a_7^m x^2 y + a_8^m xy^2 + a_9^m y^3,$$

приближающей функцию f в точках W_i . Задачу приближения функций мы будем рассматривать в постановке интерполирования. В этом случае интерполянт должен удовлетворять условиям

$$\phi(W_i) = \phi(x_i, y_i) = f_i, \quad w_i \in W.$$

Выведена система уравнений для нахождения непрерывного кубического сплайна с непрерывными первыми производными. В связи с наличием в системе свободных переменных, сформулированы критерии для нахождения решения. Основным критерием является минимизация суммы квадратов разрывов вторых производных на каждом внутреннем ребре триангуляции.

Система уравнений приведена к такому виду, что восемь искоемых коэффициентов выражены через два оставшихся и коэффициенты на каком-то соседнем треугольнике. Благодаря этому удалось изобрести алгоритм, существенно опережающий обычный алгоритм решения системы по различным критериям. Новый алгоритм лучше в скорости, количестве используемой памяти и точности, так как количество переменных в решаемой системе уменьшается

примерно в пять раз, и в алгоритме практически не используется операция деления.

Создан программный комплекс, реализующий описанные методы и решающий задачи построения триангулированной сетки, интерполяции кубическими сплайнами и построения поля изолиний. Реализовано два алгоритма нахождения коэффициентов сплайнов – решение системы методом Гаусса и решение собственным алгоритмом.

При создании контурных карт реализована возможность задавать значения уровней, которые должны быть отображены, а также заполнять промежуточные значения каким-либо цветом.

Все результаты работы данной программы могут сохраняться в файлах и использоваться при работе с другими программами автоматической интерполяции двумерных данных.

Список литературы

1. В.А. Мальцев. Руды и металлы. №1, 1994, с. 79 –89.
2. Моор П.К. Построение оптимальных нерегулярных сеток и алгоритмов приближения функций в задачах разведки и разработки нефтегазовых месторождений. Диссертация на соискание ученой степени кандидата физико-математических наук. М. 1990.
3. Фихтенгольц Г.М. Курс дифференциального и интегрального исчисления. Том III. - М.: Наука, 1970. - с. I74-I8I.
4. Васильев Ф. П. Численные методы решения экстремальных задач. М. Наука. 1988.
5. Кремер Н. Ш. Исследование операций в экономике. М. Юнити. 2001.
6. Кнут Д. Э. Искусство программирования. т.1. М. Вильямс. 2001.
7. Моор П.К. Построение оптимальных разбиений при кусочно-постоянной и кусочно-линейном интерполировании. - В сб.: Применение численного анализа и ЭВМ при моделировании и решении задач Западно-Сибирского региона /Сборник научных трудов/ - Тюмень, 1988, с. 61-66.
8. Карманов В.Г. Математическое программирование. М. Наука, 1975.
9. Василенко В.А. Сплайн-функции: теория, алгоритмы, программы. Новосибирск: Наука, 1983.