

СПЛАЙН-АППРОКСИМАЦІЯ НА ОСНОВЕ ТРИАНГУЛЯЦІЇ

Е.В. Назаренко

Інститут кібернетики ім. В.М. Глушкова НАН України,
03028, Київ, проспект Науки 43, кв. 70.
Тел.: 524 5330, eugn@ukr.net

Описан алгоритм поліноміальної сплайн-аппроксимации на основе триангуляції, використовується при побудові карт. Метод забезпечує більш точне співпадіння з заданими точками, ніж В-сплайни.

An algorithm of the polynomial spline approximation based on triangulation is described. The algorithm is used for building maps. The method provides more precise fitting to defined points than B-splines.

Введення

Задача побудови карт добре вивчена і має широке застосування. Розроблено ряд методів її рішення: триангуляція, метод мінімальної кривизни [1, 2], методи кригінга [3, 4], методи сплайн-аппроксимации [5], метод Шепарда, скользящей плоскости, інверсних відстаней, N найближчих сусідів, натуральної околиці, радіальних базисних функцій і др.

Методи сплайн-аппроксимации мають певні переваги: можливість використання стабілізаторів для задання загальних властивостей картируваної поверхні, адитивне включення в функціонал довільного числа якісної і кількісної інформації з використанням вагових коефіцієнтів в якості керуючих параметрів [6], стійкість і надійність.

Недоліки сплайн-аппроксимации [5]:

- значення на карті можуть виходити за діапазон відомих значень в області картопобудови;
- обчислювальна складність;
- слабкий порівняно з кригінгом контроль помилки аппроксимации;
- кластерний ефект (приводить до того, що точна аппроксимация всередині щільних скоплень відомих точок компенсує великі відхилення в окремих точках і на краях області аппроксимации).

Алгоритм поліноміальної сплайн-аппроксимации на основі триангуляції забезпечує кращу для обчислень систему рівнянь, ніж аппроксимация на основі радіальних базисних функцій, а також відсутність «ефекту бичього ока», характерного для більшості інших методів.

Побудова триангуляції

Діаграма Вороного. Нехай S – кінцеве множинство точок на площині. Многоугольником Вороного $V^*(p)$ для точки $p \in S$ називається множинство точок на площині, більш близьких до p , ніж до будь-якої іншої точки множинства S [7]. Нехай p_i і p_j – дві точки. Тоді множинством точок, більш близьких до p_i , ніж до p_j , є область, що містить точку p_i , півплощина $H(p_i, p_j)$, обмежена прямою, що проходить через середину відрізка $[p_i, p_j]$ перпендикулярно до нього. $V^*(p_i) = \bigcap_{i \neq j} H(p_i, p_j)$ [8]. Границю многоугольника $V^*(p_i)$ будемо позначати як $V(p_i)$. Для заданого множинства точок S діаграма Вороного складається з множинства многоугольників Вороного для кожної з точок множинства S (рис. 1). Ребро діаграми Вороного, що лежить між двома многоугольниками Вороного $V^*(p_i)$ і $V^*(p_j)$, складається з точок на площині, еквідистантних точкам p_i і p_j , і розташоване ближче до цих точок, ніж до інших точок множинства S [7]. Вершини діаграми Вороного визначаються вершинами многоугольників Вороного.

Розглянемо алгоритм побудови діаграми Вороного, наведений в [8]. Визначимо:

aP – масив точок множинства S ;

$lEdg$ – список ребер діаграми;

$aVor$ – масив многоугольників Вороного.

Список $lEdg$ складається з кортежів $VOR_EDGE \langle n_1 \in N, n_2 \in N, e \in E, o \in \{0,1,2\} \rangle$. E – множинство відрізків виду $\langle org \in R^2, dest \in R^2 \rangle$. Елементи n_1, n_2 кортежа задають відповідно ліву (p_l) і праву (p_r) відносно вектора e (направленого з org до $dest$) точки множинства S такі, що $e \in V(p_l) \cap V(p_r)$. Ребро e , в свою чергу, є ребром діаграми Вороного. Ребра бувають трьох типів: пряма, промінь і відрізок. Їм відповідають значення 0, 1, 2 елемента o . Ребро є прямою тоді і тільки тоді, коли всі точки діаграми

лежат на одной прямой. Лучом ребро является лишь в случае, когда $n_1 \in CH(S) (\Leftrightarrow n_2 \in CH(S))$. Обратное тоже верно. Этот факт нам понадобится при вычислении $CH(S^*), S^* \subset S$. Отрезком ребро является во всех остальных случаях.

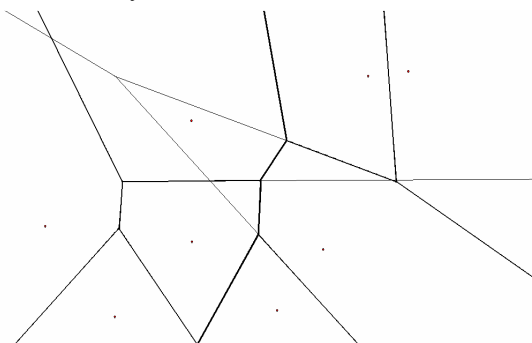


Рис. 1. Диаграмма Вороного

Каждой точке $aP[i]$ отвечает i -й элемент массива $aVor$, который содержит многоугольник Вороного для данной точки. Многоугольник $aVor[i]$ организован в виде списка ребер. Ребро имеет следующую структуру (EDGE_FOR_NODE): $\langle pE \in E_*, n \in N, pNE \in F_* \rangle$, где E_* – множество указателей на кортежи типа VOR_EDGE, F_* – множество указателей на кортежи типа EDGE_FOR_NODE. Элемент pE кортежа задает собственно ребро. n задает индекс соседа по ребру, на которое указывает pE , в массиве aP . Элемент pNE указывает на то же самое ребро у соседа. Взаимосвязь структур данных показана на рис. 2. Отметим, что ребра в многоугольнике отсортированы по часовой стрелке, что иллюстрирует, в частности, рис. 2.

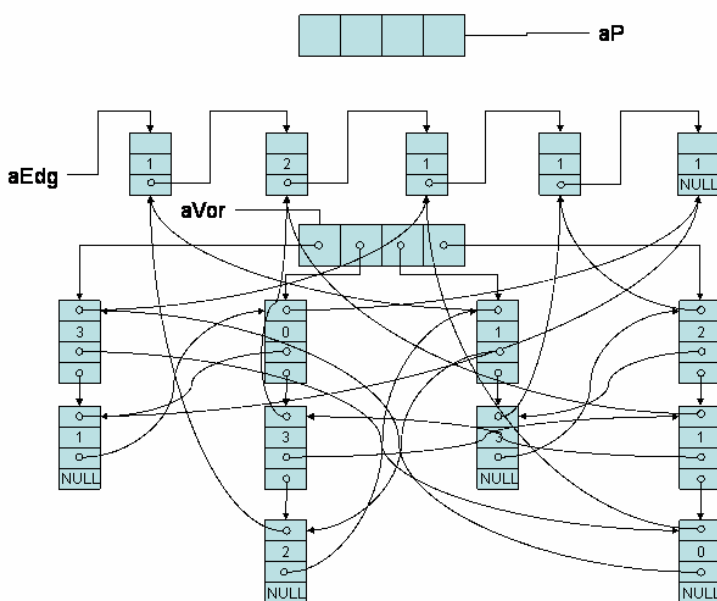


Рис. 2. Представление диаграммы Вороного

Метод “разделяй и властвуй” состоит в следующем. Задача делится на несколько подзадач, которые решаются рекурсивно и затем достигнутые результаты объединяются для получения решения исходной задачи. По своему виду подзадачи сходны с исходной задачей, но они меньше по размеру и, естественно, сумма размеров подзадач равна размеру исходной задачи. Если задача достаточно мала или проста, она может быть решена непосредственно и нет необходимости ее дальнейшего деления – это условие является критерием для окончания процесса деления [7].

В нашем случае задача каждый раз делится на две подзадачи, а условием окончания является случай, когда на вход рекурсивной функции подана одна точка.

Алгоритм построения диаграммы Вороного:

- 1) $sort(S)$;
- 2) удалить одинаковые точки;
- 3) если $|S| > 1$ $VorD(0, |S| - 1)$.

Функция $VorD(fPoint \in N, lPoint \in N)$ имеет следующий вид:

если $fPoint \neq lPoint$ то

$$VorD\left(fPoint, \left\lceil \frac{fPoint + lPoint}{2} \right\rceil\right); VorD\left(\left\lfloor \frac{fPoint + lPoint}{2} \right\rfloor + 1, lPoint\right);$$

$$\text{Объединить } Vor(S_1) \text{ и } Vor(S_2), S_1 = \bigcup_{i=fPoint}^{\lceil \frac{fPoint+lPoint}{2} \rceil} \{aP[i]\}, S_2 = \bigcup_{i=\lceil \frac{fPoint+lPoint}{2} \rceil+1}^{lPoint} \{aP[i]\}.$$

Первым этапом решения задачи является сортировка исходного множества точек S по координате x в порядке возрастания. В случае совпадения значений, точки сортируются по координате y (тоже в порядке возрастания).

Для разделения задачи на подзадачи в нашем случае используется функция VorD. В общих чертах алгоритм объединения двух диаграмм описан далее.

Пусть два множества точек S_1 и S_2 ($S_1, S_2 \subset S$) линейно разделимы вертикальной прямой $x = const$. Пусть также имеем диаграммы Вороного $Vor(S_1)$ и $Vor(S_2)$ (рис. 1). Наша задача состоит в том, чтобы путем объединения данных диаграмм за линейное время получить диаграмму Вороного для множества $S_{1,2} = S_1 \cup S_2$. Введем понятие разделяющей цепи. Для заданного разбиения $\{S_1, S_2\}$ множества $S_{1,2}$ разделяющая цепь $\sigma(S_1, S_2)$ обозначает множество ребер диаграммы Вороного, общих для пар многоугольников $V(p_i)$ и $V(p_j)$ диаграммы $Vor(S_{1,2})$, где $p_i \in S_1$ и $p_j \in S_2$ [8]. На рис. 1 изображена разделяющая цепь между множествами S_1 и S_2 . Поскольку данные множества линейно разделимы вертикальной прямой, то σ однозначно разбивает плоскость на левую π_l и правую π_r части. Пусть S_1 лежит слева от S_2 . Тогда диаграмма Вороного $Vor(S_{1,2})$ представляет собой объединение $Vor(S_1) \cap \pi_l$, $Vor(S_2) \cap \pi_r$ и σ .

Получить σ можно следующим образом. Пусть $I(e_1, e_2)$ означает пересечение линий e_1 и e_2 , каждая из которых – прямая, луч или отрезок. $|v_1, v_2|$ – расстояние между точками v_1 и v_2 . λ – отсутствие пересечения. Пусть также ось абсцисс направлена вправо, ось ординат – вниз.

1. Построить верхний опорный отрезок $[aP[p_L^T], aP[p_R^T]]$; построить нижний опорный отрезок $[aP[p_L^B], aP[p_R^B]]$; $p_l = p_L^T$; $p_r = p_R^B$; $opnd = 0$; $\sigma = \emptyset$;
2. $l =$ прямая, перпендикулярная $[aP[p_L^T], aP[p_R^T]]$ и делящая $[aP[p_L^T], aP[p_R^T]]$ пополам; $v =$ точка из l , бесконечно удаленная вправо, если l – горизонтальная и точка из l , бесконечно удаленная вверх иначе;
3. $e =$ прямая, перпендикулярная $[aP[p_L^T], aP[p_R^T]]$ и делящая $[aP[p_L^T], aP[p_R^T]]$ пополам;
4. $p_l^* = \{ I(e, m) \mid m \in V(p_l) \text{ и } |v, I(e, m)| \text{ – минимально, если } opnd = 0 \text{ и } I(e, m) \neq v \text{ иначе} \}$;
 $p_r^* = \{ I(e, m) \mid m \in V(p_r) \text{ и } |v, I(e, m)| \text{ – минимально, если } opnd = 0 \text{ и } I(e, m) \neq v \text{ иначе} \}$;
5. если $p_l^* = \lambda$ и $p_r^* = \lambda$ перейти на 8;
 если $p_l^* = \lambda$ перейти на 6.3;
 если $p_r^* = \lambda$ перейти на 6.2;
6. если $|v, p_l^*| = |v, p_r^*|$
 - 6.1. $\sigma = \sigma \cup \{(v, p_l^*)\}$, если $opnd = 0$ и $\sigma \cup \{(v, p_l^*)\}$ иначе;
 $v = p_l^*$; $opnd = opnd + 1$;
 $p_l =$ сосед по ребру e^* (см. рис. 3.1, 3.2);
 $p_r =$ сосед по ребру e^{**} (см. рис. 3.1, рис. 3.2);
 иначе если $|v, p_l^*| < |v, p_r^*|$
 - 6.2. $\sigma = \sigma \cup \{(v, p_l^*)\}$, если $opnd = 0$ и $\sigma \cup \{(v, p_l^*)\}$ иначе;
 $v = p_l^*$; $opnd = opnd + 1$;
 $p_l =$ сосед по ребру e^* (см. рис. 3.2);
 иначе
 - 6.3. $\sigma = \sigma \cup \{(v, p_r^*)\}$, если $opnd = 0$ и $\sigma \cup \{(v, p_r^*)\}$ иначе;
 $v = p_r^*$; $opnd = opnd + 1$;
 $p_r =$ сосед по ребру e^{**} (см. рис. 3.2);
7. если $p_l \neq p_L^B \vee p_r \neq p_R^B$ перейти на 3;

8. e – прямая, перпендикулярная $[aP[p_i], aP[p_r]]$ и делящая $[aP[p_i], aP[p_r]]$ пополам; v^* – точка из e , бесконечно удаленная влево, если e – горизонтальная и точка из e , бесконечно удаленная вниз иначе; $\sigma = \sigma \cup \{v, v^*\}$, если $opnd = 0$ и $\sigma \cup \{v, v^*\}$ иначе.

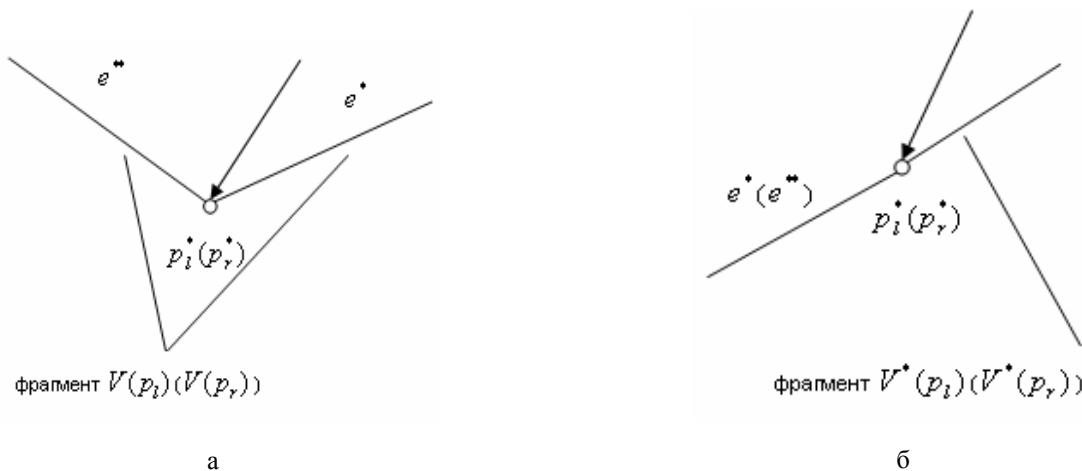


Рис. 3

Алгоритм построения верхнего и нижнего опорного отрезков подробно описан в [7]. Заметим только, что он использует выпуклые оболочки множеств S_1 и S_2 , которые могут быть эффективно вычислены следующим образом. Ранее мы установили связь между неограниченными многоугольниками и выпуклой оболочкой. Ребра в неограниченном многоугольнике отсортированы таким образом, что лучами являются первое и последнее ребро в списке ребер многоугольника. Для простоты предположим, что $Vor(S_{1,2})$ не содержит прямых.

Следующая последовательность приводит к вычислению $CH(S_{1,2})$:

1. если $|S| = 1$
 $A = \{p\}$; // $aP[p]$ – точка, принадлежащая выпуклой оболочке, например, точка с наибольшей координатой x
 конец вычисления;
2. $A = \emptyset$;
 $p_i = p$;
3. $A = A \cup \{p_i\}$;
4. p_i – элемент n первого элемента $aVor[p_i]$;
5. если $p_i \neq p$ перейти на 3.

Таким образом, A – искомая выпуклая оболочка множества $S_{1,2}$. В приведенном алгоритме она вычисляется в порядке обхода по часовой стрелке. $CH(S_{1,2})$, упорядоченную против часовой стрелки, можно вычислить, заменив шаг 4 шагом 4*:

- 4* . p_i – элемент n последнего элемента $aVor[p_i]$.

Случай, когда $Vor(S_{1,2})$ состоит из прямых, требует внесения лишь незначительных уточнений в приведенный алгоритм.

Шаг 4. алгоритма построения разделяющей цепи имеет следующую особенность. Цепь σ , оставаясь внутри некоторого многоугольника $V(p_i)$, может многократно менять свое направление, прежде чем она пересечет какое-либо ребро многоугольника $V(p_i)$. Однако, ввиду особенностей алгоритма, при поиске пересечений (начиная со второго) следует рассматривать только ребра, идущие по часовой стрелке после предыдущего пересечения для левой диаграммы и против часовой стрелки для правой диаграммы.

Чтобы завершить слияние диаграмм, остается удалить все ребра (и части ребер) диаграммы $Vor(S_2)$, расположенные слева от σ , и все ребра (и части ребер) $Vor(S_1)$, расположенные справа от σ . Обрезать и удалять ребра можно одновременно с построением разделяющей цепи.

Триангуляция Делоне. Задача триангуляции множества точек может быть сформулирована следующим образом. На плоскости заданы N точек. Соединить их непересекающимися отрезками так, чтобы каждая область внутри выпуклой оболочки этого множества точек являлась треугольником [8].

Выпуклой триангуляцией называется такая триангуляция, для которой минимальный многоугольник, охватывающий все треугольники, будет выпуклым. Говорят, что триангуляция удовлетворяет условию Делоне,

если внутрь окружности, описанной вокруг любого построенного треугольника, не попадает ни одна из заданных точек триангуляции. Триангуляция называется триангуляцией Делоне, если она является выпуклой и удовлетворяет условию Делоне [9].

Пусть S – множество точек на плоскости, причем все точки множества S не лежат на одной прямой. Пусть также никакие четыре точки исходного множества не лежат на одной окружности. Тогда имеет место утверждение.

Теорема Делоне. Граф, двойственный диаграмме Вороного (рис. 4), является триангуляцией (Делоне) множества S [8].

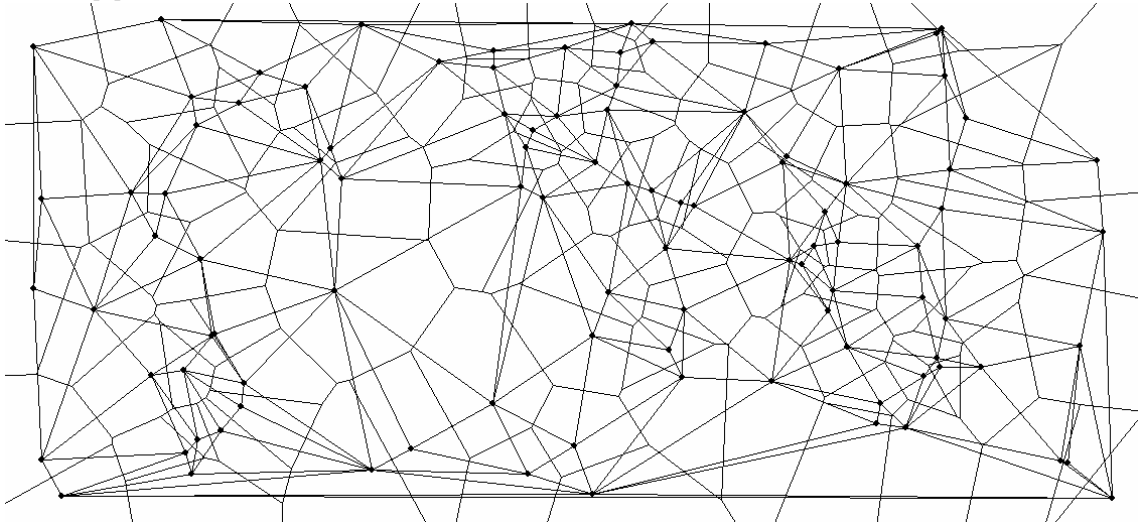


Рис. 4. Диаграмма Вороного и двойственная ей триангуляция Делоне

Таким образом, триангуляция Делоне может быть получена из диаграммы Вороного. В случае, когда четыре точки множества S лежат на одной окружности, после “обращения” ребер графа диаграммы, помимо треугольников, образуются также многоугольники. Для получения триангуляции они должны быть разбиты на треугольники произвольным образом.

Заметим, что вершины диаграммы Вороного являются центрами описанных окружностей вокруг треугольников триангуляции Делоне того же множества, для которого построена диаграмма. Это свойство может быть использовано для получения триангуляции Делоне из диаграммы Вороного:

1. Для каждого ребра, если оно не является лучом, сделать дубликат.
2. Отсортировать ребра по *org* (или по *dest*).
3. Сформировать триангуляцию.

В результате сортировки ребра, инцидентные одной вершине, будут расположены подряд. Располагая информацией о точках множества S , между которыми построено ребро (элементы n_1, n_2 кортежа VOR_EDGE), триангуляция Делоне может быть построена за один проход по отсортированному списку ребер диаграммы.

Рассмотрим шаг 2. алгоритма подробнее. Будем говорить, что e_1 больше, чем e_2 если функция $less(e_1, e_2)$ возвращает *true* и e_1 меньше, чем e_2 – иначе. Функция $less(e_1, e_2)$ имеет следующий вид:

1. v_1 = вершина, которой инцидентно ребро e_1 ; v_2 = вершина, которой инцидентно ребро e_2 ;
2. если $v_1.x = v_2.x$

если $v_1.y = v_2.y$

α = угол, который образуется при переходе вектора (0,-1) в вектор e_1 при вращении вектора (0,-1) вокруг точки v_1 по часовой стрелке;

β = угол, который образуется при переходе вектора (0,-1) в вектор e_2 при вращении вектора (0,-1) вокруг точки v_2 по часовой стрелке;

если $\alpha < \beta$, то $less = true$; иначе $less = false$;

иначе

если $v_1.y < v_2.y$, то $less = true$; иначе $less = false$;

3. если $v_1.x < v_2.x$, то $less = true$;

4. $less = false$;

Примечание. “.” – элемент кортежа.

Для задания триангуляции используем массив, состоящий из кортежей $\langle p_1 \in N, p_2 \in N, p_3 \in N \rangle$. Элементы кортежа задают индексы вершин треугольника в массиве aP .

Построение карты

В общем случае задача построения карты может быть сформулирована так. На двумерной ограниченной области аппроксимации по известной количественной и качественной информации необходимо отыскать неизвестную функцию f , принадлежащую пространству H функций, непрерывных вместе с первыми производными на всей области аппроксимации, за исключением заданного множества меры нуль [10]. Однако на практике для представления карт используются значения f в узлах заранее заданной регулярной прямоугольной сетки значений (заданной прямоугольником r и шагами Δ_x, Δ_y).

В нашем случае количественная информация представлена известными значениями в некоторых точках и описывается функционалом вида

$$J_z(f) = \sum_i (f(x_i, y_i) - z_i)^2.$$

Качественная информация чаще всего бывает следующих трех видов:

1. Искомая функция f напоминает другую известную функцию f_a – например, карту вышележащего горизонта, крупномасштабную карту или карту, полученную на основании других способов измерения.

$$J_a(f, f_a) = \sum_{i,j} \left[\left(\frac{\partial f(x_i, y_j)}{\partial x} - \frac{\partial f_a(x_i, y_j)}{\partial x} \right)^2 + \left(\frac{\partial f(x_i, y_j)}{\partial y} - \frac{\partial f_a(x_i, y_j)}{\partial y} \right)^2 \right].$$

2. Искомая функция f должна изменяться плавно, с минимальными искривлениями. Физическая модель плавно изменяющейся функции – плоская упругая пластина, искривляемая силами притяжения к точкам, чье положение в пространстве определяется известными значениями:

$$J_2(f) = \sum_{i,j} \left[\left(\frac{\partial^2 f(x_i, y_j)}{\partial x^2} \right)^2 + \left(\frac{\partial^2 f(x_i, y_j)}{\partial y^2} \right)^2 \right].$$

3. Искомая функция f соответствует стационарному процессу, ожидаемое значение в случайной точке не зависит от горизонтального смещения:

$$J_1(f) = \sum_{i,j} \left[\left(\frac{\partial f(x_i, y_j)}{\partial x} \right)^2 + \left(\frac{\partial f(x_i, y_j)}{\partial y} \right)^2 \right].$$

Таким образом подавляются всплески значений на краях области построения карты, где отсутствуют известные значения.

С учетом вышеизложенного задача картопостроения сводится к минимизации функционала:

$$\Phi^* = \lambda_z J_z + \lambda_a J_a + \lambda_s (\gamma J_2 + (1 - \gamma) J_1) \rightarrow \min.$$

Далее вместо функционала Φ^* рассмотрим функционал (поскольку конечно-разностное представление J_a аналогично конечно-разностному представлению J_1) $\Phi = \lambda_z J_z + \lambda_s (\gamma J_2 + (1 - \gamma) J_1)$.

Пусть S – множество точек на плоскости, в которых известно значение искомой функции f . Область аппроксимации ограничим прямоугольником $r = \{(x, y) | x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}\}$. Пусть $S^* = S \cup \{(x_{\min}, y_{\min}), (x_{\min}, y_{\max}), (x_{\max}, y_{\min}), (x_{\max}, y_{\max})\}$. Область аппроксимации разобьем сеткой триангуляции множества S^* на множество непересекающихся подобластей, каждая из которых представляет треугольник с вершинами в точках множества S^* . Тогда пространство H аппроксимируется конечномерным пространством H_s функций, совпадающих на каждой подобласти с полиномом Лагранжа третьей степени:

$$f_L = b_0 + b_1 x + b_2 y + b_3 x^2 + b_4 y^2 + b_5 xy + b_6 x^3 + b_7 y^3 + b_8 xy^2 + b_9 x^2 y$$

и удовлетворяющих условиям непрерывности функции/производных на границах подобластей [5].

Для задачи восстановления f в узлах регулярной сетки производную целесообразно считать методом конечных разностей:

$$J_1(f) = \sum_{i=2}^{n_x} \sum_{j=1}^{n_y} \left(\frac{f(x_i, y_j) - f(x_{i-1}, y_j)}{\Delta_x} \right)^2 + \sum_{i=1}^{n_x} \sum_{j=2}^{n_y} \left(\frac{f(x_i, y_j) - f(x_i, y_{j-1})}{\Delta_y} \right)^2.$$

Точки, между которыми проходит разлом, не должны быть включены в J_1 .

$$J_2(f) = \sum_{i=3}^{n_x} \sum_{j=1}^{n_y} \left(\frac{f(x_i, y_j) - 2f(x_{i-1}, y_j) + f(x_{i-2}, y_j)}{\Delta_x^2} \right)^2 + \sum_{i=1}^{n_x} \sum_{j=3}^{n_y} \left(\frac{f(x_i, y_j) - 2f(x_i, y_{j-1}) + f(x_i, y_{j-2})}{\Delta_y^2} \right)^2.$$

Точки, между которыми проходит разлом, не должны быть включены в J_2 .

Для минимизации функционала Φ составляется система линейных уравнений $\frac{\partial \Phi(a_1, \dots, a_n)}{\partial a_k} = 0, k = 1, n$.

Введем следующие обозначения:

$$f_L^i = a_{10i} + a_{10i+1}x + a_{10i+2}y + a_{10i+3}x^2 + a_{10i+4}y^2 + a_{10i+5}xy + a_{10i+6}x^3 + a_{10i+7}y^3 + a_{10i+8}xy^2 + a_{10i+9}x^2y;$$

$$c_k(x, y) - \text{коэффициент функции } f_L^i(a_{10i}, \dots, a_{10i+9}) \text{ при переменной } a_k \text{ (т.е., } c_k(x, y) = \frac{\partial f_L^i}{\partial a_k}, i = \left[\frac{k}{10} \right];$$

$$c_k^i(x, y) = \begin{cases} c_k(x, y), & \text{если } 10i \leq k \leq 10i + 9 \\ 0, & \text{иначе} \end{cases};$$

$l(x, y)$ – индекс функции, построенной на треугольнике, содержащем точку (x, y) .

Тогда

$$\begin{aligned} \frac{1}{2} \frac{\partial \Phi(a_1, \dots, a_n)}{\partial a_k} &= \lambda_z \sum_{j=1}^3 c_k \left(x_j^{\left[\frac{k}{10} \right]}, y_j^{\left[\frac{k}{10} \right]} \right) \left(f_L^{\left[\frac{k}{10} \right]} \left(x_j^{\left[\frac{k}{10} \right]}, y_j^{\left[\frac{k}{10} \right]} \right) - z_j^{\left[\frac{k}{10} \right]} \right) + \\ &+ \lambda_s (1 - \gamma) \left(\frac{1}{\Delta_x} \sum_{i=2}^{n_x} \sum_{j=1}^{n_y} \left(c_k^{l(x_i, y_j)}(x_i, y_j) - c_k^{l(x_{i-1}, y_j)}(x_{i-1}, y_j) \right) \left(f_L^{l(x_i, y_j)}(x_i, y_j) - f_L^{l(x_{i-1}, y_j)}(x_{i-1}, y_j) \right) \right) + \\ &+ \lambda_s (1 - \gamma) \left(\frac{1}{\Delta_y} \sum_{i=1}^{n_x} \sum_{j=2}^{n_y} \left(c_k^{l(x_i, y_j)}(x_i, y_j) - c_k^{l(x_i, y_{j-1})}(x_i, y_{j-1}) \right) \left(f_L^{l(x_i, y_j)}(x_i, y_j) - f_L^{l(x_i, y_{j-1})}(x_i, y_{j-1}) \right) \right) + \\ &+ \lambda_s \gamma \left(\frac{1}{\Delta_x} \sum_{i=3}^{n_x} \sum_{j=1}^{n_y} \left(c_k^{l(x_i, y_j)}(x_i, y_j) - 2c_k^{l(x_{i-1}, y_j)}(x_{i-1}, y_j) + c_k^{l(x_{i-2}, y_j)}(x_{i-2}, y_j) \right) \left(f_L^{l(x_i, y_j)}(x_i, y_j) - 2f_L^{l(x_{i-1}, y_j)}(x_{i-1}, y_j) + f_L^{l(x_{i-2}, y_j)}(x_{i-2}, y_j) \right) \right) + \\ &+ \lambda_s \gamma \left(\frac{1}{\Delta_y} \sum_{i=1}^{n_x} \sum_{j=3}^{n_y} \left(c_k^{l(x_i, y_j)}(x_i, y_j) - 2c_k^{l(x_i, y_{j-1})}(x_i, y_{j-1}) + c_k^{l(x_i, y_{j-2})}(x_i, y_{j-2}) \right) \left(f_L^{l(x_i, y_j)}(x_i, y_j) - 2f_L^{l(x_i, y_{j-1})}(x_i, y_{j-1}) + f_L^{l(x_i, y_{j-2})}(x_i, y_{j-2}) \right) \right). \end{aligned} \quad (1)$$

Отметим, что $c_k^{l(x, y)}(x, y)$ для большинства x и y равняется нулю, так, что при составлении (1) при суммировании можно учитывать только те точки, которые отстоят от треугольника, которому соответствует функция $f_L^{\left[\frac{k}{10} \right]}$, не более чем на $2\Delta_x$ и $2\Delta_y$ соответственно.

Далее эта система решается, что приводит к вычислению неизвестных коэффициентов a_1, \dots, a_n и, таким образом, к определению функции f . Для решения задачи остается лишь вычислить значения уже известной функции f в узлах сетки. Заметим, что во избежание переполнений при составлении системы линейных уравнений, значения x и y заменяются соответственными значениями с интервалов $[x_{\min}, x_{\max}]$ и $[y_{\min}, y_{\max}]$,

причем должно выполняться равенство $\frac{\Delta_x}{\Delta_y} = \frac{\Delta_x^*}{\Delta_y^*}$, где $\Delta_x^* = \frac{x_{\max} - x_{\min}}{n_x - 1}$, $\Delta_y^* = \frac{y_{\max} - y_{\min}}{n_y - 1}$. Интервалы могут быть

выбраны, например, такими: $[0, \Delta_x - 1]$, $\left[0, \frac{\Delta_y}{\Delta_x} (n_y - 1) \right]$.

Реализация метода в пакете «ГеоПоиск»

Метод включен как один из методов картопостроения в пакет «ГеоПоиск» [11].

Рассмотрим применение метода для получения кровли пласта по имеющимся скважинам (рис. 5). Положение скважин определяет разбиение области аппроксимации (рис. 6). Каждому треугольнику отвечает функция – полином Лагранжа третьей степени, которая представлена в системе линейных уравнений десятью коэффициентами. Таким образом, количество скважин определяет порядок системы.

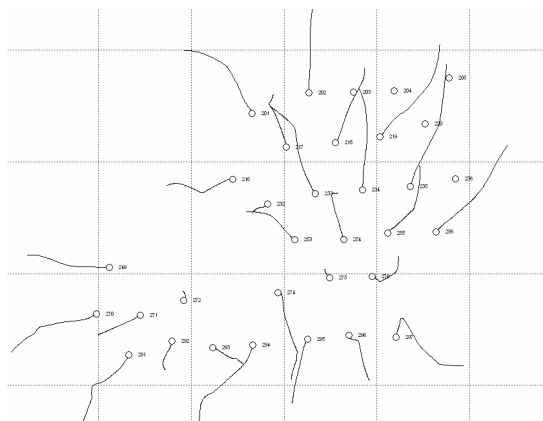


Рис. 5

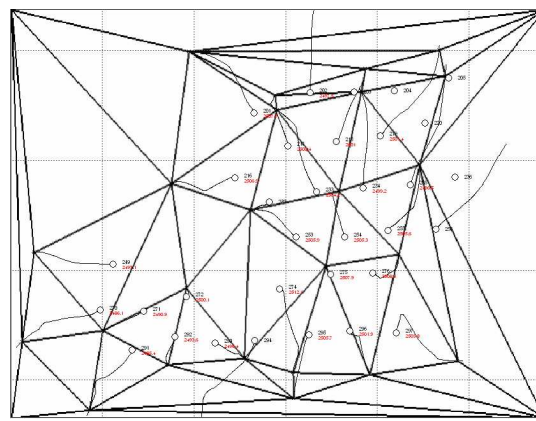


Рис. 6

Для расчета поверхности необходимо задать следующие параметры: размер сетки, вес известных значений, вес поверхности-аналога, вес стабилизатора, доля 1-й производной в стабилизаторе. Размер сетки определяет количество узлов, в которых будет вычислена поверхность. Кроме того, более густая сетка обеспечивает более точную аппроксимацию поверхности в известных точках (за счет увеличения точности вычисления производных), однако требует больше времени на составление системы линейных уравнений. Веса позволяют управлять влиянием соответствующего фактора (упругость, поверхность-аналог и т.п.) на поверхность. На рис. 7 показано вычисленную приведенным методом карту.

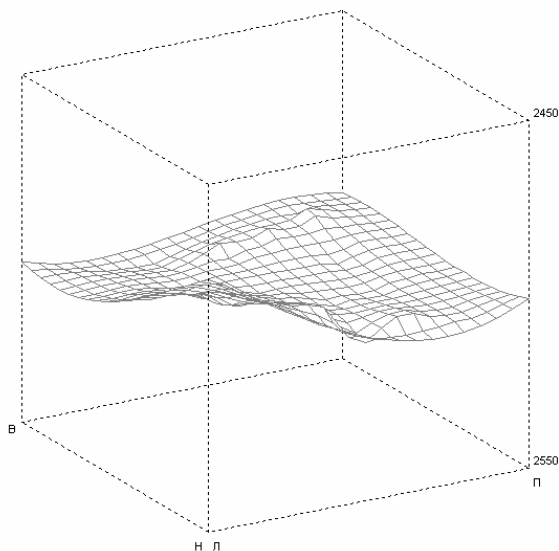


Рис. 7

Заключение

Рассмотренный метод обеспечивает построение достаточно точной реалистичной карты. Его эффективная реализация включена в набор методов картопостроения пакета «ГеоПоиск», предназначенного для обработки и интерпретации данных геофизического исследования нефте-газовых скважин с привлечением смежной геологической и геофизической информации. Разломы (нарушения) с известной амплитудой, проходящие через кровлю/подошву пласта, реализованы путем предварительного вычитания из известных значений вертикального сдвига в соответствующих точках и последующего прибавления к карте поверхности карты сдвига из-за разломов. Усовершенствовать реализацию

разломов с неизвестной амплитудой можно, отдельно рассматривая части треугольника, разделенных разломом. Также планируется реализовать правку сетки триангуляции вручную, так как триангуляция Делоне, оптимизированная по форме треугольников, не отвечает требованию выстраивания длинного ребра треугольника вдоль направления напластования пород, в соответствии геологическими представлениями специалистов.

1. Smith W. H. F., Wessel P. Gridding with continuous curvature splines in tension // Geophysics. – 1990. – Vol. 55, N 3. – P. 293 – 305.
2. Красавчиков В.О. Математические методы, алгоритмы и технология геомоделирования для решения задач геологии нефти и газа. Автореф... докт. дис. – Кемерово. – 2007.
3. Abdullah Arik. Area Influence Kriging // Mathematical Geology. – 2002. – Vol. 34, N 7. – P. 783 – 796.
4. Schenkova Z., Schenk V., Kalogeras I., Pichl R., Kottnauer P., Papatsimba C., Panopoulou G. Isoseismal maps drawing by the kriging method // Journal of Seismology. – 2007. – Vol. 11, N 3. – P. 345 – 353.
5. Тульчинский В.Г. Технология картопостроения // Кибернетика и системный анализ. – 1999. – № 3. – С. 17 – 21.
6. Плавник А.Г. Алгоритмизация геоинформационных технологий в задачах, связанных с картопостроением. Автореф. канд. дис. – Тюмень. – 2004.
7. Майкл Ласло. Вычислительная геометрия и компьютерная графика на C++. – 1997. – 304 с.
8. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. – 1989. – 478 с.
9. Сворцов А.В. Триангуляция Делоне и её применение. – 2002. – 128 с.
10. Стечкин С.В., Субботин Ю.Н. Сплайны в вычислительной математике. – М.: Наука, 1976. – 248 с.
11. www.geopoisik.com.