

Задача Изолинии – построение портрета функции

Реализовать приложение для построения "портрета" однозначной скалярной функции двух переменных в виде цветовой карты и карты изолиний.

Постановка задачи:

Итак, дана однозначная функция

$$z = f(x, y), (x, y) \in D = [a, b] * [c, d].$$

На клиентской области окна приложения выбираем прямоугольник P с углами (u_0, v_0) и (u_1, v_1) в экранных координатах (пикселях), ось V - сверху вниз. Пусть даны n чисел:

$$z_1 < z_2 < \dots < z_n,$$

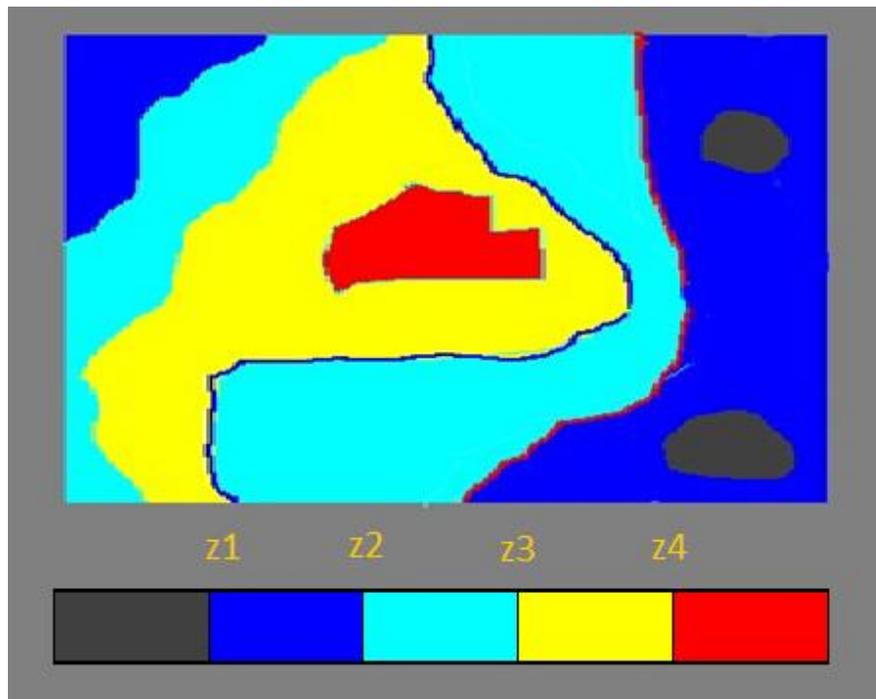
и $n + 1$ цвет:

$$c_0, c_1, c_2, \dots, c_n.$$

Будем говорить, что значению z соответствует цвет c_0 , если $z < z_1$, иначе значению z соответствует цвет c_i ($i = 1..n$), где i определяется как

$$i = \min(j) \{z \geq z_j\}.$$

Рассмотрим довольно простое линейное отображение D на P , когда точка (a, d) переходит в пиксель (u_0, v_0) , а точка (b, c) в (u_1, v_1) . Очевидное обратное преобразование позволяет по центру каждого пикселя (u, v) из P получать точку (x, y) из D . И, таким образом, пикселю присваивается цвет, соответствующий значению $f(x, y)$. Полученное изображение прямоугольника P и будем называть цветной картой функции f . Снизу от цветовой карты функции на свободном месте окна приложения рисуется горизонтальная *легенда* с подписями над ней, показывающая параметры цветной карты функции:



Границы, разделяющие зоны, раскрашенные в разные цвета, называются линиями уровня или изолиниями. Так, граница раздела областей, залитых цветами i и $(i - 1)$, – это изолиния уровня z_i – линия, являющаяся решением уравнения $f(x, y) = z_i$.

Параметры задачи:

- $f(x, y)$ – программируется внутри приложения. Выбор конкретной функции – за исполнителем. Локализовать в какой-либо функции;
- a, b, c, d – область определения функции;
- n – число ключевых значений, вводится из файла;
- Цвета $c_0, c_1, c_2, \dots, c_n$ – вводятся из файла в виде 255 170 24 (красный, зелёный, синий). Одно значение цвета на строку файла;
- z_1, z_2, \dots, z_n – ключевые значения. Для определения этих значений: подсчитываются минимум и максимум функции, в полученном интервале равномерно распределяются остальные значения.

Примечание: полученную карту вместе с легендой вписать в клиентскую область окна.

Изолинии сеточной функции

Имеем

$$a = x_1 < x_2 < \dots < x_k = b,$$

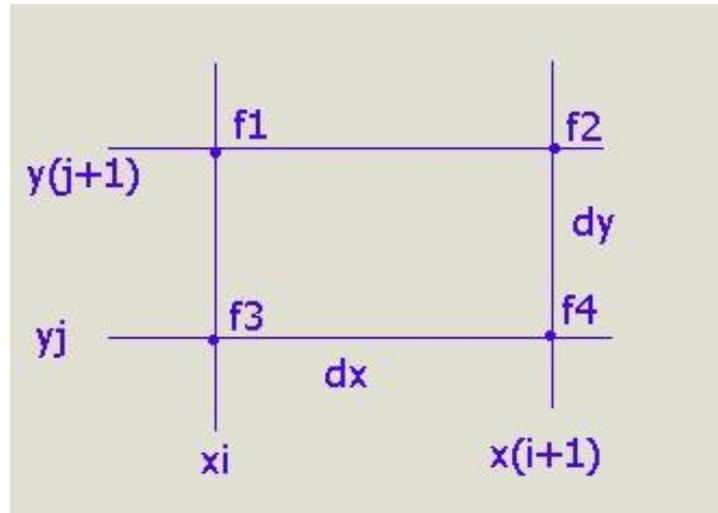
$$c = y_1 < y_2 < \dots < y_m = d$$

некоторую сетку на прямоугольнике D (см. выше). И набор из $k \cdot m$ значений

$$z_{ij} = f(x_i, y_j)$$

некоторой неизвестной функции f , т.е. имеем функцию, заданную на сетке, аналог задания рельефа высотами на прямоугольной сетке. В принципе, можно воспользоваться какой-либо интерполяцией и определить функцию f во всех точках области, а затем построить изолинии последней. Здесь же предлагается реализовать следующий очень простой алгоритм марширующих кубиков (marching cubes).

1. Делается цикл по всем ячейкам сетки, т.е. построение изолинии на каждой ячейке делается независимо.
2. Пусть мы в настоящее время строим изолинию уровня z в очередной ячейке:

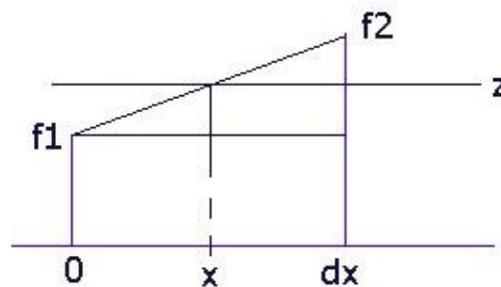


здесь $dx = x(i+1) - x_i$, $dy = y(j+1) - y_j$, а через $f_1..f_4$ обозначены для краткости значения сеточной функции в соответствующих узлах.

3. Необходимо найти точки "входа/выхода" изолинии на границах ячейки (если таковые точки есть). Поскольку все 4 стороны рассматриваются независимо, то проведём поиск для верхней границы ячейки. В данном алгоритме считается, что неизвестная функция ведёт себя между узлами линейно. Изолиния уровня z не пересекает верхнюю сторону ячейки, если

$$z < f_1 \& z < f_2 \text{ или } z > f_1 \& z > f_2.$$

Иначе пересечение есть, найдём его, приняв для определённости, что $f_1 < f_2$.



Получаем $x = dx * (z - f_1) / (f_2 - f_1)$ в пределах клетки, а на области определения функции точка входа изолинии в ячейку

$$(x[i] + (x[i+1] - x[i]) * (z - z[i, j+1]) / (z[i+1, j+1] - z[i, j+1]), y(j+1)).$$

Аналогично определяются и точки на других границах ячейки.

4. Рассмотрим возможные ситуации.
 - Ни одной точки пересечения изолинии с границей ячейки. Это значит, что изолиния не проходит через ячейку.

- Ровно две точки, т.е. нашли точки входа на 2-х из 4-х сторон. Соединяем эти точки отрезком прямой линии.
- Может быть и 3 точки. Добавить анализ как ведёт себя изолиния уровня $z + eps$ или $z - eps$.
- 4 точки, т.е. на каждой стороне имеется точка входа. Разбиваем на 4 треугольника, сводим проблему к предыдущим двум случаям.

Задание

1. Пусть сетка равномерная по каждой из осей. Насчитать сеточную функцию по функции f , запрограммированной в предыдущей части задания. Значения k и m взять из файла.
2. Цвет изолинии взять из файла.

Преобразование координат $D \rightarrow P$

$(x, y) \rightarrow (u, v)$, на примере горизонтальной координаты

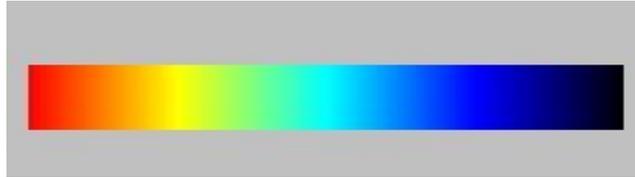
$$u = [(u_1 - u_0) * (x - a) / (b - a) + u_0 + 0.5]$$

$$x = (b - a) * (u - u_0) / (u_1 - u_0) + a$$

$[\dots]$ - целая часть.

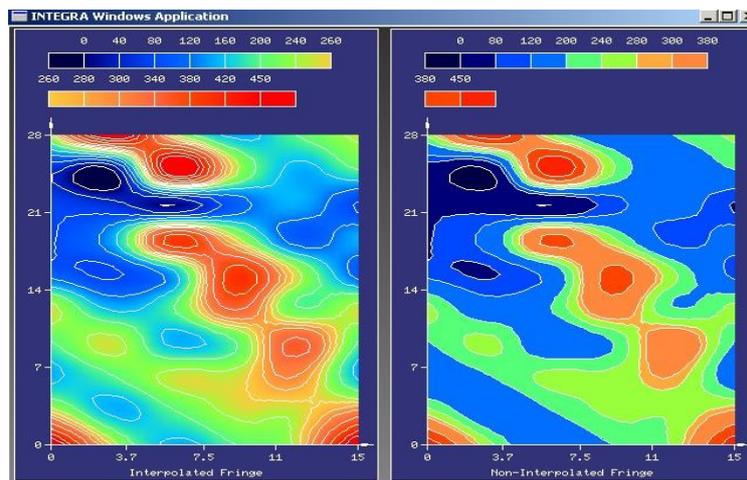
Интерполяция цвета

Режим "плавной" закрашки. Легенда тоже должна иметь интерполированный вид в этом случае.



Надо проинтерполировать цвета от пикселя с координатой $v1$ и цветом $(r1, g1, b1)$ до пикселя с координатой $v2$ и цветом $(r2, g2, b2)$. По каждой компоненте цвета интерполяция выполняется отдельно. Пусть нам надо получить красный r для пикселя с координатой u . Применяем формулу линейной интерполяции:

$$r = r1 * (v2 - v) / (v2 - v1) + r2 * (v - v1) / (v2 - v1).$$



Слева применена интерполяция, справа нет.

Требования к программе

1. Читать параметры задачи из файла (должен быть подготовлен свой файл данных). Файл должен поддерживать комментарии, как и в предыдущих задачах.
2. Изменять параметры задачи в диалоге: k, t, a, b, c, d (обязательно), остальные (цвета) по желанию, можно разрешить изменение количества цветов, дать возможность задания новых цветов. Диалог вызывается по кнопке на тулбаре.
3. Отображать функцию в одном из двух видов: цветовая карта, интерполяция. Переключать виды кнопками на тулбаре. Перерисовывать легенду для каждого вида. Легенда — это как бы отдельная функция, равномерно возрастающая по горизонтали, плюс подписи; будет красиво, если вы отобразите легенду с помощью того же кода, что и основную картинку, просто подсунув в этот код другую функцию.
4. Отображать поверх функции изолинии для уровней z_1, z_2, \dots, z_n ; возможность включить и выключить изолинии кнопкой на тулбаре

5. Отображать поверх функции сетку $k \times m$; возможность включить и выключить кнопкой на тулбаре;
6. Добавить интерактивное построение изолиний по клику мыши: определяете по координатам пикселя, куда пользователь кликнул, точку в области определения функции f , берёте её значение и строите изолинию.
7. Режим динамической изолинии. В нём пока клавиша мыши нажата в области функции изолиния строится в след за движением мыши. В каждый момент определяется значение функции под мышью и строится соответствующая изолиния.
8. Режим отображения точек входа в прямоугольники (и треугольники). Отображать, например, маленькими цветными кружками.
9. При движении мыши над полем значения x , y и $f(x, y)$, соответствующее точке экрана, выводится в статусбаре (не в пикселях!).

Замечание: легенда состоит не только из палитры используемых цветов, но и цифровых значений уровней, которые рисуются над легендой. Расположение надписей должно быть правильно центрировано – центр надписи совпадает с границей цветов на легенде. То есть надо определять габаритный бокс надписи и размещать его корректно. Значения разумно округлять, чтобы они не напоздали друг на друга.

На что следует обратить внимание

1. Все действия осуществляются по кнопкам: вкл/выкл цветной карты, вкл/выкл карты из n изолиний, вкл/выкл режима интерактивного построения, вкл/выкл сетки (и другие режимы). Сетка строится либо при помощи линий, либо в виде узловых точек (отдельным цветом). Возможны варианты личных решений.
2. Поле изолиний и легенда должны **ВПИСЫВАТЬСЯ** в клиентскую область окна, т.е. необходимо производить расчёт в программе. (Нарушение – максимум 3).
3. Отсутствие разрывов в линиях. (Нарушение – максимум 2).
4. Наличие анализа случаев 4-х и 3-х точек пересечения изолинии с клеткой (Нет – максимум 3).
5. В описании точно указать, где в программе задаётся функция. Это должен быть отдельный метод.
6. Отсутствие диалога для задания и смены параметров (минус 1).
7. Наличие слежения за мышью, т.е. отображение координат точки в области определения функции f и самого значения функции. (нет – максимум 3)

Итак, максимум – 4. Для 5-ки надо

8. Всё остальное.

Формат файла

```
k m // число значений сетки по X и по Y
n // число уровней
r0 g0 b0 // цвета легенды
r1 g1 b1
```

...

rn gn bn

riz giz biz // цвет построения изолиний

EOF