

# Восстановление донной поверхности по различным картографическим данным

А.А. Сухинов

В статье описан новый метод интерполяции, учитывающий форму области интерполяции и погрешность исходных данных. Метод применён к задаче восстановления донной поверхности по различным картографическим данным (отметки глубины, изолинии глубины, области, отмеченные различными цветами). Разработан многомасштабный параллельный алгоритм, выполняющий интерполяцию.

## 1. Введение

Для построения гидродинамических моделей мелководных водоёмов и прибрежных систем требуются точные электронные карты этих объектов [1]. Как правило, обычные бумажные карты являются основным источником информации для построения электронных карт. В данной статье рассмотрен последний этап построения карт, когда различная картографическая информация (отметки высоты, изолинии высоты, области, отмеченные различным цветом) уже переведена в цифровой вид. Предлагаемый метод также подходит для интерполяции других измеренных данных. Основное отличие предлагаемого метода от других методов интерполяции заключается в том, что он может учитывать точность измерений и форму области интерполяции.

Математически задача формулируется, как оптимизационная задача с естественными ограничениями. Для получения результата должна быть решена система линейных уравнений с ограничениями. Это требует большого объёма вычислений при использовании итерационных методов. Для существенного сокращения объёма вычислений использован многомасштабный подход; дополнительное ускорение получено за счёт использования параллельных вычислений.

## 2. Оптимизационная задача восстановления поверхности

В плоской области  $G$  заданы ограничения на диапазон изменения высот: функции  $m(x, y)$  и  $M(x, y)$  такие, что  $m(x, y) < M(x, y)$ , где  $(x, y) \in G$  — горизонтальные координаты некоторой точки области. Это означает, что значение реальной глубины/высоты  $e(x, y)$  в области  $G$  удовлетворяет неравенствам  $m(x, y) \leq e(x, y) \leq M(x, y)$ . Допустимы случаи, когда в некоторых точках или подобластях  $m(x, y) = -\infty$  и/или  $M(x, y) = +\infty$ .

Требуется построить в области  $G$  функцию  $f(x, y)$ , которая имеет непрерывные производные первого порядка и ограниченные производные второго порядка, удовлетворяющую следующим условиям:

$$m(x, y) \leq f(x, y) \leq M(x, y), \quad (1)$$

$$\iint_{(x,y) \in G} \alpha^2 \left( \frac{\partial f}{\partial x} \right)^2 + \alpha^2 \left( \frac{\partial f}{\partial y} \right)^2 + \left( \frac{\partial^2 f}{\partial x^2} \right)^2 + \left( \frac{\partial^2 f}{\partial y^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2 dx dy \rightarrow \min_f, \quad (2)$$

где  $\alpha \geq 0$  — параметр, значение которого будет описано ниже.

Постановка задачи имеет следующие отличия по сравнению с классической задачей интерполяции [2]:

- исходные данные представлены в интервальном виде:  $[m(x, y), M(x, y)]$ ;
- форма области  $G$  также берётся в расчёт; это важно при интерполяции физических величин, существующих только в пределах некоторой области (например, концентрации загрязнений в водоёме).

Перечисленные особенности расширяют область применения предлагаемого интерполяционного метода.

Функции  $m(x, y)$  и  $M(x, y)$  обычно представляют собой результаты измерений. Например, если мы имеем  $N$  измерений глубины/высоты  $(x_i, y_i, e_i)$ ,  $1 \leq i \leq N$ , то функции  $m$  и  $M$  могут быть заданы следующим образом:

$$m(x, y) = \begin{cases} e_i - \Delta e_i, & \text{если } \exists i : x_i = x, y_i = y; \\ -\infty, & \text{иначе.} \end{cases}$$

$$M(x, y) = \begin{cases} e_i + \Delta e_i, & \text{если } \exists i : x_i = x, y_i = y; \\ +\infty, & \text{иначе.} \end{cases}$$

Здесь величины  $e_i$  имеют погрешность измерения  $\Delta e_i > 0$ . Предполагается, что координаты  $x_i, y_i$  известны точно.

Обсудим постановку задачи. Минимизация суммы квадратов вторых производных в выражении (2) приводит к тому, что поверхность в каждой точке стремится быть «наиболее близкой к плоскости». Член со смешанной производной присутствует для того, чтобы всё выражение (2) было нечувствительно к повороту системы координат.

Минимизация суммы квадратов первых производных даёт тенденцию горизонтального положения интерполированной поверхности. Это соответствует минимизации потенциальной энергии земной поверхности. Коэффициент  $\alpha \geq 0$  (обычно достаточно малый) предназначен для настройки этой особенности.

В общем, модель (1), (2) можно понимать, как построение «наиболее гладкой» поверхности, соответствующей имеющимся ограничениям.

Лучшим алгоритмом интерполяции точно заданных данных считается крайгинг [3]. Однако он вычислительно трудоёмок и не учитывает форму области интерполяции. Подбор параметра  $\alpha$  позволяет значительно приблизить результаты предлагаемого метода к результатам интерполяции, получаемым при помощи крайгинга. По видимому, в случае точечных исходных данных, имеющих одинаковую погрешность измерения  $\Delta e_i = \Delta e$ , значение параметра  $\alpha$  следует выбирать таким, чтобы кривая

$$\gamma(h) = \frac{2}{3} (\Delta e)^2 + k \cdot (1 - e^{-\alpha h})^2, \quad h \geq 0 \quad (3)$$

при некотором параметре  $k > 0$  наиболее близко аппроксимировала экспериментальную вариограмму [3], вычисленную по точкам  $(x_i, y_i, e_i)$ ,  $1 \leq i \leq N$ .

### 3. Дискретизация задачи

В дальнейшем для простоты изложения будем считать, что  $G$  — квадратная область:  $G = ((x, y) : 0 \leq x \leq H, 0 \leq y \leq H)$ . Построим в области  $G$  равномерную прямоугольную сетку  $\bar{G}$ , состоящую из  $N \times N$  квадратных ячеек  $C_{i,j}$  ( $i, j = 1, \dots, N$ ) размерами  $h = H/N$ . Далее будем считать, что  $N \geq 4$  и  $N$  кратно четырём.

Определим сеточные функции  $m_{i,j}$  и  $M_{i,j}$ :

$$\begin{aligned} \tilde{m}_{i,j} &= \sup_{(x,y) \in C_{i,j}} m(x,y); \\ \tilde{M}_{i,j} &= \inf_{(x,y) \in C_{i,j}} M(x,y); \\ m_{i,j} &= \begin{cases} \tilde{m}_{i,j}, & \text{если } \tilde{m}_{i,j} \leq \tilde{M}_{i,j}, \\ (\tilde{m}_{i,j} + \tilde{M}_{i,j}) / 2, & \text{иначе;} \end{cases} \\ M_{i,j} &= \begin{cases} \tilde{M}_{i,j}, & \text{если } \tilde{m}_{i,j} \leq \tilde{M}_{i,j}, \\ (\tilde{m}_{i,j} + \tilde{M}_{i,j}) / 2, & \text{иначе.} \end{cases} \end{aligned} \quad (4)$$

$$\quad (5)$$

Тогда дискретные аналоги выражений (1), (2) могут быть записаны следующим образом:

$$m_{i,j} \leq f_{i,j} \leq M_{i,j}, \quad i, j = 1, \dots, N; \quad (6)$$

$$\begin{aligned} R \equiv & \frac{\alpha^2}{h^2} \sum_{i=2}^N \sum_{j=2}^N \left( (f_{i,j} - f_{i-1,j})^2 + (f_{i,j} - f_{i,j-1})^2 \right) + \\ & + \frac{1}{h^4} \sum_{i=2}^{N-1} \sum_{j=2}^{N-1} \left( (f_{i-1,j} - 2f_{i,j} + f_{i+1,j})^2 + (f_{i,j-1} - 2f_{i,j} + f_{i,j+1})^2 \right) + \\ & + \frac{2}{h^4} \sum_{i=2}^N \sum_{j=2}^N (f_{i,j} - f_{i-1,j} - f_{i,j-1} + f_{i-1,j-1})^2 \rightarrow \min, \end{aligned} \quad (7)$$

где  $f_{i,j}$  — неизвестные величины. Для сокращения записи будем считать, что

$$f_{i,j} = 0, \quad \text{если } C_{i,j} \notin \bar{G}. \quad (8)$$

#### 4. Решение дискретной задачи

В выражении (7)  $R$  является функцией аргументов  $f_{i,j}$ ,  $i, j = 1, \dots, N$ . Если зафиксировать значения всех аргументов функции  $R$ , кроме некоторого аргумента  $f_{r,s}$ , то она будет представлять собой многочлен второй степени с положительным коэффициентом при  $f_{r,s}^2$ . Такая функция обладает минимумом, который достигается в единственной точке  $\tilde{f}_{r,s}$ , определяемой из уравнения:

$$\left. \frac{\partial R}{\partial f_{r,s}} \right|_{f_{r,s} = \tilde{f}_{r,s}} = 0. \quad (9)$$

Тогда минимум функции  $R$ , как аргумента  $f_{r,s}$ , при ограничениях (6) будет достигаться в точке  $f_{r,s}^*$ , определяемой следующим образом:

$$f_{r,s}^* = \max \left( m_{r,s}, \min \left( M_{r,s}, \tilde{f}_{r,s} \right) \right) \equiv L_{r,s} \left( \tilde{f}_{r,s} \right). \quad (10)$$

Решим уравнение (9). Дифференцируя  $R$  по  $f_{i,j}$ , получим:

$$\begin{aligned}
\frac{\partial R}{\partial f_{i,j}} = & \frac{\alpha^2}{h^2} \frac{\partial}{\partial f_{i,j}} \left[ b_{i-1} \cdot (f_{i,j} - f_{i-1,j})^2 + b_{i+1} \cdot (f_{i+1,j} - f_{i,j})^2 + \right. \\
& \left. + b_{j-1} \cdot (f_{i,j} - f_{i,j-1})^2 + b_{j+1} \cdot (f_{i,j+1} - f_{i,j})^2 \right] + \\
& + \frac{1}{h^4} \frac{\partial}{\partial f_{i,j}} \left[ b_{i-1,i+1} \cdot (f_{i-1,j} - 2f_{i,j} + f_{i+1,j})^2 + b_{j-1,j+1} \cdot (f_{i,j-1} - 2f_{i,j} + f_{i,j+1})^2 + \right. \\
& + b_{i-2} \cdot (f_{i-2,j} - 2f_{i-1,j} + f_{i,j})^2 + b_{i+2} \cdot (f_{i,j} - 2f_{i+1,j} + f_{i+2,j})^2 + \\
& + b_{j-2} \cdot (f_{i,j-2} - 2f_{i,j-1} + f_{i,j})^2 + b_{j+2} \cdot (f_{i,j} - 2f_{i,j+1} + f_{i,j+2})^2 \left. \right] + \\
& + \frac{2}{h^4} \frac{\partial}{\partial f_{i,j}} \left[ b_{i-1,j-1} \cdot (f_{i,j} - f_{i-1,j} - f_{i,j-1} + f_{i-1,j-1})^2 + \right. \\
& + b_{i+1,j-1} \cdot (f_{i+1,j} - f_{i,j} - f_{i+1,j-1} + f_{i,j-1})^2 + \\
& + b_{i-1,j+1} \cdot (f_{i,j+1} - f_{i-1,j+1} - f_{i,j} + f_{i-1,j})^2 + \\
& \left. + b_{i+1,j+1} \cdot (f_{i+1,j+1} - f_{i,j+1} - f_{i+1,j} + f_{i,j})^2 \right], \quad (11)
\end{aligned}$$

где сомножители  $b_k$  и  $b_{k_1,k_2} = b_{k_1} \cdot b_{k_2}$  предназначены для правильной записи выражения (11) в приграничных ячейках:

$$b_k = \begin{cases} 1, & \text{если } 1 \leq k \leq N; \\ 0, & \text{иначе.} \end{cases} \quad (12)$$

Вычислим производные в выражении (11):

$$\begin{aligned}
\frac{\partial R}{\partial f_{i,j}} = & \frac{2\alpha^2}{h^2} [b_{i-1} \cdot (f_{i,j} - f_{i-1,j}) - b_{i+1} \cdot (f_{i+1,j} - f_{i,j}) + \\
& + b_{j-1} \cdot (f_{i,j} - f_{i,j-1}) - b_{j+1} \cdot (f_{i,j+1} - f_{i,j})] + \\
& + \frac{2}{h^4} [-2b_{i-1,i+1} \cdot (f_{i-1,j} - 2f_{i,j} + f_{i+1,j}) - 2b_{j-1,j+1} \cdot (f_{i,j-1} - 2f_{i,j} + f_{i,j+1}) + \\
& + b_{i-2} \cdot (f_{i-2,j} - 2f_{i-1,j} + f_{i,j}) + b_{i+2} \cdot (f_{i,j} - 2f_{i+1,j} + f_{i+2,j}) + \\
& + b_{j-2} \cdot (f_{i,j-2} - 2f_{i,j-1} + f_{i,j}) + b_{j+2} \cdot (f_{i,j} - 2f_{i,j+1} + f_{i,j+2})] + \\
& + \frac{4}{h^4} [b_{i-1,j-1} \cdot (f_{i,j} - f_{i-1,j} - f_{i,j-1} + f_{i-1,j-1}) - \\
& - b_{i+1,j-1} \cdot (f_{i+1,j} - f_{i,j} - f_{i+1,j-1} + f_{i,j-1}) - \\
& - b_{i-1,j+1} \cdot (f_{i,j+1} - f_{i-1,j+1} - f_{i,j} + f_{i-1,j}) + \\
& + b_{i+1,j+1} \cdot (f_{i+1,j+1} - f_{i,j+1} - f_{i+1,j} + f_{i,j})]. \quad (13)
\end{aligned}$$

Найдём значение  $\tilde{f}_{i,j}$ , при котором частная производная (13) равна нулю:

$$\begin{aligned}
& \alpha^2 h^2 \left[ b_{i-1} \cdot (\tilde{f}_{i,j} - f_{i-1,j}) - b_{i+1} \cdot (f_{i+1,j} - \tilde{f}_{i,j}) + \right. \\
& \quad \left. + b_{j-1} \cdot (\tilde{f}_{i,j} - f_{i,j-1}) - b_{j+1} \cdot (f_{i,j+1} - \tilde{f}_{i,j}) \right] + \\
& + \left[ -2b_{i-1,i+1} \cdot (f_{i-1,j} - 2\tilde{f}_{i,j} + f_{i+1,j}) - 2b_{j-1,j+1} \cdot (f_{i,j-1} - 2\tilde{f}_{i,j} + f_{i,j+1}) + \right. \\
& \quad + b_{i-2} \cdot (f_{i-2,j} - 2f_{i-1,j} + \tilde{f}_{i,j}) + b_{i+2} \cdot (\tilde{f}_{i,j} - 2f_{i+1,j} + f_{i+2,j}) + \\
& \quad \left. + b_{j-2} \cdot (f_{i,j-2} - 2f_{i,j-1} + \tilde{f}_{i,j}) + b_{j+2} \cdot (\tilde{f}_{i,j} - 2f_{i,j+1} + f_{i,j+2}) \right] + \\
& \quad + 2 \left[ b_{i-1,j-1} \cdot (\tilde{f}_{i,j} - f_{i-1,j} - f_{i,j-1} + f_{i-1,j-1}) - \right. \\
& \quad - b_{i+1,j-1} \cdot (f_{i+1,j} - \tilde{f}_{i,j} - f_{i+1,j-1} + f_{i,j-1}) - \\
& \quad - b_{i-1,j+1} \cdot (f_{i,j+1} - f_{i-1,j+1} - \tilde{f}_{i,j} + f_{i-1,j}) + \\
& \quad \left. + b_{i+1,j+1} \cdot (f_{i+1,j+1} - f_{i,j+1} - f_{i+1,j} + \tilde{f}_{i,j}) \right] = 0.
\end{aligned}$$

Сгруппируем члены:

$$\begin{aligned}
& \tilde{f}_{i,j} \cdot [\alpha^2 h^2 (b_{i-1} + b_{i+1} + b_{j-1} + b_{j+1}) + 4(b_{i-1,i+1} + b_{j-1,j+1}) + \\
& \quad + (b_{i-2} + b_{i+2} + b_{j-2} + b_{j+2}) + 2(b_{i-1,j-1} + b_{i+1,j-1} + b_{i-1,j+1} + b_{i+1,j+1})] - \\
& \quad - f_{i-1,j} \cdot [\alpha^2 h^2 b_{i-1} + 2b_{i-1,i+1} + 2b_{i-2} + 2b_{i-1,j-1} + 2b_{i-1,j+1}] - \\
& \quad - f_{i+1,j} \cdot [\alpha^2 h^2 b_{i+1} + 2b_{i-1,i+1} + 2b_{i+2} + 2b_{i+1,j-1} + 2b_{i+1,j+1}] - \\
& \quad - f_{i,j-1} \cdot [\alpha^2 h^2 b_{j-1} + 2b_{j-1,j+1} + 2b_{j-2} + 2b_{i-1,j-1} + 2b_{i+1,j-1}] - \\
& \quad - f_{i,j+1} \cdot [\alpha^2 h^2 b_{j+1} + 2b_{j-1,j+1} + 2b_{j+2} + 2b_{i-1,j+1} + 2b_{i+1,j+1}] + \\
& \quad + f_{i-1,j-1} \cdot 2b_{i-1,j-1} + f_{i+1,j-1} \cdot 2b_{i+1,j-1} + f_{i-1,j+1} \cdot 2b_{i-1,j+1} + f_{i+1,j+1} \cdot 2b_{i+1,j+1} + \\
& \quad + f_{i-2,j} \cdot b_{i-2} + f_{i+2,j} \cdot b_{i+2} + f_{i,j-2} \cdot b_{j-2} + f_{i,j+2} \cdot b_{j+2} = 0. \quad (14)
\end{aligned}$$

Введём обозначения:

$$\begin{aligned}
\tilde{k}_{i,j}^{-1,0} &= \alpha^2 h^2 b_{i-1} + 2b_{i-1,i+1} + 2b_{i-2} + 2b_{i-1,j-1} + 2b_{i-1,j+1}; \\
\tilde{k}_{i,j}^{+1,0} &= \alpha^2 h^2 b_{i+1} + 2b_{i-1,i+1} + 2b_{i+2} + 2b_{i+1,j-1} + 2b_{i+1,j+1}; \\
\tilde{k}_{i,j}^{0,-1} &= \alpha^2 h^2 b_{j-1} + 2b_{j-1,j+1} + 2b_{j-2} + 2b_{i-1,j-1} + 2b_{i+1,j-1}; \\
\tilde{k}_{i,j}^{0,+1} &= \alpha^2 h^2 b_{j+1} + 2b_{j-1,j+1} + 2b_{j+2} + 2b_{i-1,j+1} + 2b_{i+1,j+1}; \\
\tilde{k}_{i,j}^{-1,-1} &= -2b_{i-1,j-1}; \quad \tilde{k}_{i,j}^{+1,-1} = -2b_{i+1,j-1}; \\
\tilde{k}_{i,j}^{-1,+1} &= -2b_{i-1,j+1}; \quad \tilde{k}_{i,j}^{+1,+1} = -2b_{i+1,j+1}; \\
\tilde{k}_{i,j}^{-2,0} &= -b_{i-2}; \quad \tilde{k}_{i,j}^{+2,0} = -b_{i+2}; \quad \tilde{k}_{i,j}^{0,-2} = -b_{j-2}; \quad \tilde{k}_{i,j}^{0,+2} = -b_{j+2}; \\
\tilde{k}_{i,j}^{0,0} &= \tilde{k}_{i,j}^{-1,0} + \tilde{k}_{i,j}^{+1,0} + \tilde{k}_{i,j}^{0,-1} + \tilde{k}_{i,j}^{0,+1} + \tilde{k}_{i,j}^{-1,-1} + \tilde{k}_{i,j}^{+1,-1} + \\
& \quad + \tilde{k}_{i,j}^{-1,+1} + \tilde{k}_{i,j}^{+1,+1} + \tilde{k}_{i,j}^{-2,0} + \tilde{k}_{i,j}^{+2,0} + \tilde{k}_{i,j}^{0,-2} + \tilde{k}_{i,j}^{0,+2}; \\
k_{i,j}^{p,q} &= \tilde{k}_{i,j}^{p,q} / \tilde{k}_{i,j}^{0,0}, \quad |p+q| \leq 2.
\end{aligned} \tag{15}$$

Тогда уравнение (14) может быть записано следующим образом:

$$\begin{aligned} & \tilde{k}_{i,j}^{0,0} \cdot \tilde{f}_{i,j} - \tilde{k}_{i,j}^{-1,0} \cdot f_{i-1,j} - \tilde{k}_{i,j}^{+1,0} \cdot f_{i+1,j} - \tilde{k}_{i,j}^{0,-1} \cdot f_{i,j-1} - \tilde{k}_{i,j}^{0,+1} \cdot f_{i,j+1} - \\ & - \tilde{k}_{i,j}^{-1,-1} \cdot f_{i-1,j-1} - \tilde{k}_{i,j}^{+1,-1} \cdot f_{i+1,j-1} - \tilde{k}_{i,j}^{-1,+1} \cdot f_{i-1,j+1} - \tilde{k}_{i,j}^{+1,+1} \cdot f_{i+1,j+1} - \\ & - \tilde{k}_{i,j}^{-2,0} \cdot f_{i-2,j} - \tilde{k}_{i,j}^{+2,0} \cdot f_{i+2,j} - \tilde{k}_{i,j}^{0,-2} \cdot f_{i,j-2} - \tilde{k}_{i,j}^{0,+2} \cdot f_{i,j+2} = 0, \end{aligned}$$

или

$$\tilde{f}_{i,j} = \sum_{0 < |p+q| \leq 2} k_{i,j}^{p,q} \cdot f_{i+p,j+q} \equiv S_{i,j}(f_{i-2,j-2}, \dots, f_{i+2,j+2}). \quad (16)$$

С учётом (16) выражение (10) принимает вид:

$$f_{r,s}^* = L_{r,s}(S_{r,s}(f_{r-2,s-2}, \dots, f_{r+2,s+2})). \quad (17)$$

Полученный оператор (17) даёт оптимальное значение отдельно взятой переменной  $f_{r,s}$  при фиксированных значениях остальных переменных. Используем этот оператор для построения итерационного процесса решения задачи (6), (7).

Итерационный процесс можно осуществить, обходя ячейки сетки построчно, аналогично методу Гаусса—Зейделя. Однако такой алгоритм не подходит для решения на параллельной системе из-за цепной зависимости вычисляемых величин друг от друга. Поэтому обобщим для данной задачи метод верхней релаксации с красно-чёрным упорядочиванием узлов [4]. Для этого разделим всё множество ячеек сетки на 16 классов в соответствии со следующей таблицей, имеющей размеры  $N \times N$ :

$$B = \left[ \begin{array}{cccc|cccc} 1 & 13 & 4 & 16 & 1 & 13 & 4 & \dots \\ 9 & 5 & 12 & 8 & 9 & 5 & 12 & \\ 3 & 15 & 2 & 14 & 3 & 15 & 2 & \\ 11 & 7 & 10 & 6 & 11 & 7 & 10 & \dots \\ \hline 1 & 13 & 4 & 16 & 1 & 13 & 4 & \\ \vdots & & & \vdots & & & & \ddots \end{array} \right]. \quad (18)$$

Каждая итерация выполняется следующим образом: вначале уточняются (при помощи оператора (17)) значения тех элементов, которые принадлежат первому классу, затем уточняются те элементы, которые принадлежат второму классу, и так далее до класса номер 16. Разностный шаблон оператора  $S$  ограничен квадратом  $5 \times 5$ , а элементы каждого класса стоят друг от друга на 3 ячейки сетки, поэтому элементы одного класса будут вычисляться независимо друг от друга. Это позволяет проводить вычисления параллельно.

Часть таблицы (18) размерами  $4 \times 4$  ячейки называется матрицей Байера [5]. Она широко известна в теории обработки изображений. Существуют подобные матрицы любого размера  $2^n \times 2^n$ . Применительно к задаче итерационного решения системы уравнений особенностью этой матрицы является то, что низкочастотные составляющие ошибки, вносимой выбранным порядком обработки элементов, имеют максимально высокую частоту. Для сравнения, если использовать итерационный метод Гаусса—Зейделя без переупорядочивания узлов, получим ненулевую гармонику ошибки с длиной волны, сравнимой с размерами области. Такую гармонику можно трактовать, как появление выделенного направления, вызванного выбранным порядком обхода узлов. Ситуация усугубляется тем, что подобные итерационные методы крайне медленно снижают низкочастотные составляющие матрицы ошибок.

Для формальной записи итерационного метода нам потребуется представление матрицы  $B$ , дающее смещение элементов каждого класса по его номеру:

$$\begin{aligned} B^x &= (1, 3, 1, 3, 2, 4, 2, 4, 1, 3, 1, 3, 2, 4, 2, 4)^T; \\ B^y &= (1, 3, 3, 1, 2, 4, 4, 2, 2, 4, 4, 2, 1, 3, 3, 1)^T. \end{aligned} \quad (19)$$

Тогда алгоритм итерационного процесса можно записать следующим образом:

$$\begin{array}{l} \text{Repeat} \quad //\text{повторяем итерации} \\ \quad \left| \begin{array}{l} d \leftarrow 0 \quad //\text{изменение решения на текущей итерации} \\ \text{For } c = 1, \dots, 16 \quad //\text{перебираем все классы ячеек} \\ \quad \left| \begin{array}{l} \text{For } i = B_c^y, B_c^y + 4, \dots, B_c^y + N - 4 \\ \quad \left| \begin{array}{l} \text{For } j = B_c^x, B_c^x + 4, \dots, B_c^x + N - 4 \\ \quad \left| \begin{array}{l} \tilde{f} \leftarrow S_{i,j}(f_{i-2,j-2}, \dots, f_{i+2,j+2}) \\ \tilde{f} \leftarrow L_{i,j}((1-\omega) \cdot f_{i,j} + \omega \cdot \tilde{f}) \\ d \leftarrow \max(d, |\tilde{f} - f_{i,j}|) \\ f_{i,j} \leftarrow \tilde{f} \end{array} \right. \end{array} \right. \end{array} \right. \end{array} \right. \quad (20) \\ \text{While } d > \epsilon \end{array}$$

Здесь  $S_{i,j}$  — оператор минимизации функции (7), определён в выражении (16);  $L_{i,j}$  — оператор, ограничивающий сеточную функцию в соответствии с (6), определён в выражении (10);  $\omega$  — релаксационный параметр;  $(B_n^x, B_n^y)$  — смещение элементов класса номер  $n$  в сетке; смещения определены в выражении (19);  $\epsilon$  — некоторое заданное малое число; когда максимальное изменение решения на текущей итерации становится меньше  $\epsilon$ , итерации останавливаются.

При реализации алгоритма (20) имеет смысл вначале вычислить все возможные варианты оператора  $S_{i,j}$  (таких вариантов 25 штук в случае прямоугольной сетки), и затем использовать нужный оператор в зависимости от взаимного расположения вычисляемой ячейки и границ области.

## 5. Свойства задачи и алгоритма

Назовём ограничения (6) невырожденными, если

$$\inf_{(x,y) \in G} M(x,y) < \sup_{(x,y) \in G} m(x,y). \quad (21)$$

Для случая невырожденных ограничений и  $\alpha > 0$  доказаны следующие утверждения:

1. Дискретная задача (6), (7) имеет единственное решение для любого числа ячеек  $N$ .
2. Итерационный процесс (20) сходится к решению задачи (6), (7) при  $0 < \omega < 2$  для любого начального приближения.
3. Дополним графики функций  $m(x,y)$  и  $M(x,y)$  вертикальными поверхностями в местах разрывов, получив поверхности  $\tilde{m}$  и  $\tilde{M}$  соответственно. Если  $m(x,y)$  ограничена

сверху,  $M(x, y)$  ограничена снизу, и существует такое число  $\epsilon > 0$ , что минимальное расстояние между поверхностями  $m$  и  $M$  будет больше  $\epsilon$ , то решение дискретной задачи (6), (7) поточечно стремится к решению непрерывной задачи (1), (2) при шаге сетки  $h \rightarrow 0$ :  $f(x_i, y_j) - f_{i,j} = O(h)$ .

Первый порядок точности вызван тем, что сумма (7) не самым лучшим образом аппроксимирует выражение (2) на границах области  $G$ . Повышение порядка аппроксимации усложнит алгоритм, но не даст практической пользы, так как значения сеточной функции, вычисляемые *между* заданными точками, почти не зависят от значений вблизи границ области.

## 6. Параллельный многомасштабный алгоритм решения

Как уже было сказано выше, итерационный метод (20) очень медленно снижает низкочастотные гармоники матрицы ошибок. В частности, если расстояния между точками с известными значениями функции велики по сравнению с размером ячеек, то число итераций может иметь порядок  $O(N^2)$ . Кроме того, при использовании чисел одинарной точности итерационный процесс может не сойтись к правильному решению из-за накапливающихся ошибок округления.

Самый эффективный метод решения этих проблем — использование многомасштабного алгоритма, который легко реализуется в данной задаче.

Предположим, что  $N = 2^k$ . Сделаем уменьшенную вдвое копию сетки, объединив ячейки по 4 штуки. При объединении ячеек из четырёх ограничений сверху выбирается минимальное (см. (4), (5)), а из четырёх ограничений снизу — максимальное. Если ограничение сверху оказалось меньше ограничения снизу, то оба ограничения заменяются на их среднее арифметическое. Полученную сетку можно снова уменьшить вдвое и так далее, пока не дойдём до сетки размерами  $4 \times 4$  ячейки. Решим задачу на этой сетке при помощи алгоритма (20), и используем полученное решение в качестве начального приближения для решения на сетке  $8 \times 8$ . При переносе значений с грубой сетки на подробную достаточно использовать простейшую билинейную интерполяцию (центры ячеек грубой сетки не совпадают с центрами ячеек более подробной сетки). При решении задачи на различных сетках нужно учитывать, что оператор  $S_{i,j}$  зависит от шага сетки при  $\alpha \neq 0$ . В конце концов доберёмся до исходной самой подробной сетки, для которой будет хорошее начальное приближение, полученное с более грубой сетки.

Параллельный алгоритм проще всего реализовать для вычислительной системы с общей памятью. В этом случае не требуется пересылка данных. Вычисления всех элементов одного класса являются независимыми операциями. Это означает, что в алгоритме (20) можно разделять циклы по  $i$  и по  $j$  как угодно между процессорами. Самый простой и эффективный способ — это разделить цикл по индексу  $i$  на равные части соответственно числу процессоров/ядер процессора.

## 7. Тестовая задача

В качестве примера рассмотрим задачу интерполяции данных, показанных на рисунке 1. Прежде всего, следует отметить, что метод не требует распознавания изолиний глубины: в качестве входных данных может быть использован цвет областей, поэтому достаточно закрасить области между изолиниями различными цветами. Каждый цвет обозначает свой диапазон глубин, который записывается в ограничения по высоте при построении сетки.

Кроме областей с цветом карта имела множество отметок глубины и высоты (на рисунке 1 показаны кружочками), которые были оцифрованы и внесены в соответствующие ячейки сетки при её построении. Погрешность всех данных внутри водоёма положена равной 0.1 метра, погрешность данных за пределами водоёма — 1.0 метр.

Кроме того, была отдельная карта для маленького треугольного водоёма на юго-востоке



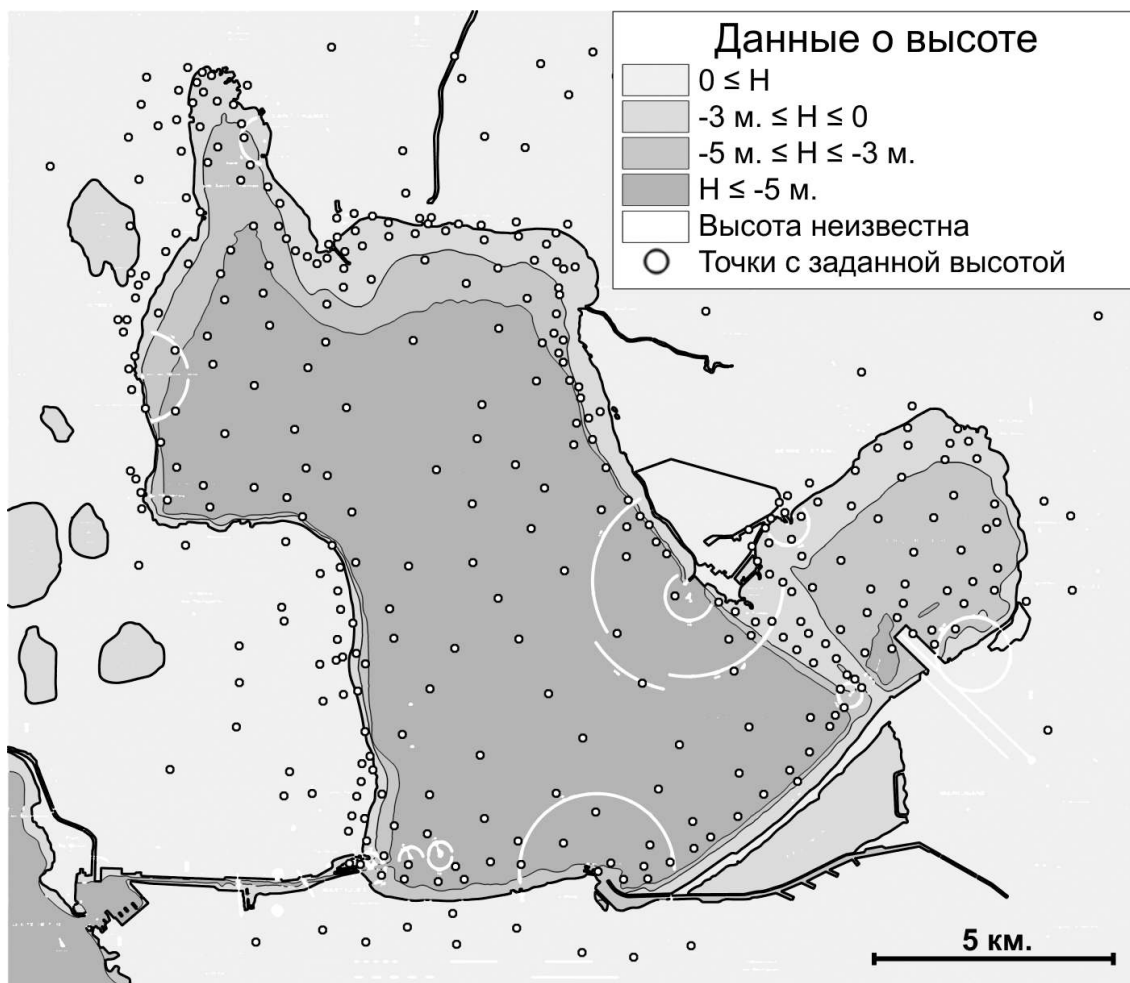


Рис. 1. Пример данных о высоте для озера Этанг-де-Берр (Франция)

карты (Этанг-де-Большмон). Её данные не показаны.

Для вычислений использовалась сетка размерами  $512 \times 512$  ячеек. Значение параметра  $\alpha = 3 \cdot 10^{-3}$  1/м. Сеточные величины — числа двойной точности. В качестве критерия сходимости итерационного процесса была выбрана величина  $\epsilon = 0.001$  метра. Процессор тестовой системы: Intel Core 2 Quad 2.4 GHz, 8 MB L2 Cache.

Вначале был произведён расчёт последовательным одномасштабным алгоритмом. Потребовалось более 50 тысяч итераций (14 минут машинного времени) для того, чтобы итерации сошлись.

Затем был произведён расчёт многомасштабным алгоритмом. На сетке каждого масштаба требовалось произвести от 10 до 100 итераций. В итоге алгоритм сошёлся за 215 итераций (2.9 секунд машинного времени). Использование многомасштабного алгоритма позволило ускорить вычисления в 300 раз!

После этого была реализована параллельная версия программы при помощи OpenMP. При запуске с двумя вычислительными потоками программа справилась с задачей интерполяции за 1.6 секунды (эффективность 90%). При запуске с четырьмя потоками интерполяция завершилась за 0.9 секунды (эффективность 81%).

Время чтения исходных данных и построения по ним сетки (вычисления ограничений) на всех тестах составляло около пяти секунд, и не принималось во внимание.

Результат интерполяции показан на рисунке 2. Изолинии высоты за пределами водоёма не показаны, так как данные о высоте на исходных картах практически отсутствовали.

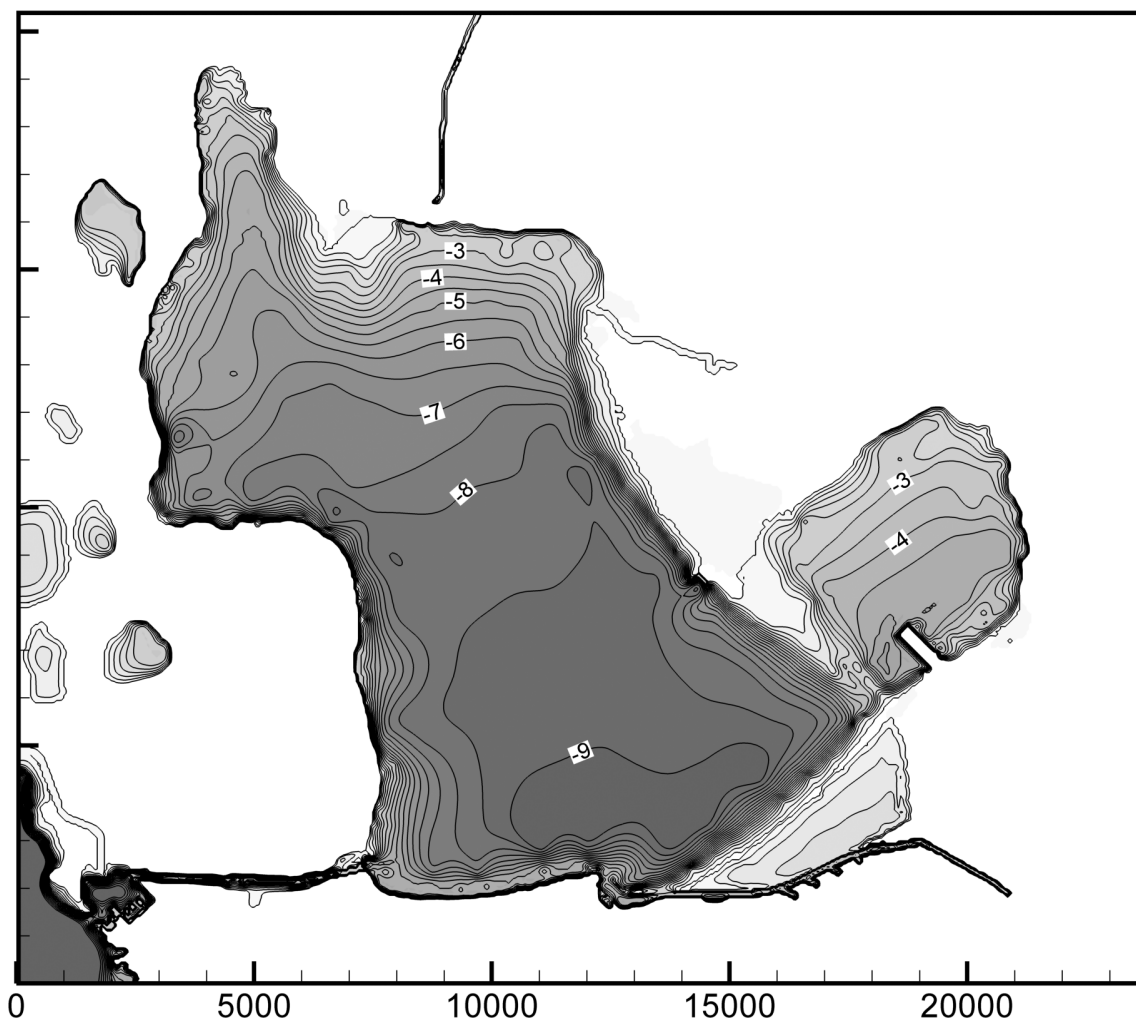


Рис. 2. Результат интерполяции. Масштаб указан в метрах

## 8. Заключение

Представленный алгоритм интерполяции может быть эффективно применён для объединения и интерполяции исходных данных различного типа. Кроме того, алгоритм учитывает возможную погрешность исходных данных, что позволяет уменьшить шум и более правильно объединить данные из различных источников.

Дискретизация задачи может быть проведена в области любой формы (не только квадратной, как описано в статье), и на различных сетках (в том числе на треугольной сетке). Учёт формы области позволяет более правильно произвести интерполяцию величин, которые распределены только в пределах заданной области.

Реализация программы с использованием многомасштабных алгоритмов и распараллеливание её с использованием нескольких вычислительных потоков позволяет получать результаты за разумное время на персональном компьютере.

Планируется усовершенствование метода путём введения в модель горизонтальной погрешности исходных данных (в дополнение к вертикальной), и нежёстких ограничений на функцию: математическое ожидание и дисперсия вместо минимального и максимального допустимых значений.

## Литература

1. Колдоба А.В., Повещенко Ю.А., Самарская Е.А., Тишкин В.Ф. Методы математического моделирования окружающей среды. –М.: Наука, 2000. –254 с.
2. Schu G.H.. Review of Interpolation Methods for Digital Terrain Models // Canadian Surveyor. –December, 1976. –Vol. 30, N. 5. –P. 389–412.
3. Oliver M.A., Webster R. Kriging: a Method of Interpolation for Geographical Information System. // INT. J. Geographical Information Systems, –1990, Vol. 4, N. 3, –P 313–332
4. Самарский А.А., Гулин А.В. Численные методы. –М.: Наука, 1989. –432 с.
5. Bayer В.Е. An Optimum Method For Two-level Rendition of Continuous-tone Pictures // Proceedings IEEE, International Conference on Communications, –Vol. 26, –1973, –P. 11–15.