

изображение на плоскости линии $f(x, y) = C$, где C — константа. Говорят, что значение C определяет уровень проведения изолинии. Под полем изолиний будем понимать множество изолиний, составленных для различных уровней C в интервале $\min_{x,y \in D} f(x, y) \leq C \leq \max_{x,y \in D} f(x, y)$. Частота проведения изолиний определяется шагом изменения величины C . Будем полагать, что он определяется значением $delta$. В приводимых примерах на рисунках величина $delta$ составляет 1. При построении изолиний функция чаще всего задается не аналитически, а некоторым множеством значений. При сеточном задании эти значения известны в узлах сетки. Более того, не стоит сбрасывать со счетов и изолинии, подготовленные визуально с использованием различного рода специальных программных средств компьютерного моделирования.

Замечание. Предлагаемый метод градиентного заполнения поля изолиний носит самый общий характер и не использует исходное поле значений функции.

Область D рассматривается как объединение замкнутых односвязных областей $D = D_1 \cup D_2 \cup \dots \cup D_{n_D}$, $D_i \cap D_j = \emptyset, i \neq j$. Распространение предлагаемого алгоритма на многосвязную область D , когда допустимы включения $D_i \subset D_j$, не представляет особого труда. Такого рода области имеют важное практическое значение. В качестве примера можно привести область D_0, D_1, \dots, D_{n_D} , где D_0 — мировой океан, D_i — острова и материки. Самая общая интерпретация областей D_1, D_2, \dots, D_{n_D} является следствием практического решения задачи построения поля изолиний, когда области D_k могут возникать естественным образом как участки общего поля наличия данных для построения изолиний.

Исходное поле m изолиний представляется их списком P_1, P_2, \dots, P_m со значениями U_1, U_2, \dots, U_m . Каждая изолиния задается вектором своих опорных точек $P_k = (P_{k1}, P_{k2}, \dots, P_{kn_k})$, где $P_{ki} = (x_{ki}, y_{ki})$ — координаты опорных точек, n_k — количество точек на k -й изолинии. Каждая из составляющих P_1, P_2, \dots, P_m поля изолиний является замкнутой. Незамкнутые изолинии должны быть дополнены до замкнутых соответствующими участками границы области $D = D_1 \cup D_2 \cup \dots \cup D_{n_D}$. На рис. 2 показан пример такого дополнения для двух незамкнутых изолиний. Далее, границы областей $D_k, k = 1, 2, \dots, n_D$ также приводятся как изолинии в общий список замкнутых изолиний P_1, P_2, \dots, P_m . Для каждой такой изолинии—границы области D_k необходимо “правильно” установить значение ее уровня U_k , так как значение на изолинии лежит в основе процедуры определения цвета ее заполнения. Под “правильным” значением U_k изолинии—границы области D_k понимается значение, которое отвечает структуре вложенности изолиний рассматриваемой области. Такие U_k могут быть определены лишь после формирования общей структуры вложенности поля изолиний. При внесении таких изолиний—границ в общий перечень изолиний значения U_k для них устанавливаются равными $+\infty$. Указанные величины выступают лишь как метки изолиний, которым еще подлежит назначить “правильные” значения.



Рис. 2

Итак, исходными данными является лишь множество замкнутых изолиний $\Pi = \{P_1, P_2, \dots, P_m\}$ со значениями U_1, U_2, \dots, U_m . Именно при таком задании исходных данных, как правило, приходится практически решать рассматриваемую задачу градиентного заполнения.

3. Области компонент связности изолиний. В задаче градиентного заполнения исходные обозначения D_1, D_2, \dots, D_{n_D} областей не используются. Исходными данными выступает лишь множество изолиний $\Pi = \{P_1, P_2, \dots, P_m\}$.

Определение. Пусть условие $P_i \subseteq P_j$ для $P_i, P_j \in \Pi$ обозначает вложенность изолинии P_i в изолинию P_j . Определим бинарное отношение (\sim) на множестве изолиний $\Pi = \{P_1, P_2, \dots, P_m\}$. Отношение $P_i \sim P_j$ выполняется, если $\exists P_r (P_i \subseteq P_r \wedge P_j \subseteq P_r)$. Таким образом, условие $P_i \sim P_j$ означает, что существует изолиния P_r , которая охватывает изолинии P_i и P_j . Очевидно, что отношение $P_i \sim P_j$ коммутативное.

Утверждение. Введенное отношение $P_i \sim P_j$ является отношением эквивалентности. Для доказательства проверим свойства отношения эквивалентности.

1. $\forall P_i \in \Pi P_i \sim P_i$ — рефлексивность. Действительно, для $P_r = P_i$ верно $P_i \subseteq P_r$.
2. $\forall P_i, P_j \in \Pi P_i \sim P_j \rightarrow P_j \sim P_i$ — симметричность. Справедливо вследствие коммутативности отношения $P_i \sim P_j$.
3. $\forall P_i, P_j, P_k \in \Pi P_i \sim P_j \wedge P_j \sim P_k \rightarrow P_i \sim P_k$ — транзитивность.

Доказательство. Имеем $P_i \sim P_j$ и $P_j \sim P_k$. Следовательно, существуют $P_r, P_s \in \Pi$, такие, что $P_i \subseteq P_r \wedge P_j \subseteq P_r$ и $P_j \subseteq P_s \wedge P_k \subseteq P_s$. Для любой пары изолиний возможно выполнение лишь двух условий их взаимного расположения: а) одна изолиния вложена в другую; б) изолинии не вложены друг в друга. Пересечения границ изолиний быть не может (свойство изолиний). Тогда из условий $P_j \subseteq P_r$ и $P_j \subseteq P_s$ следует существование у изолиний P_r и P_s общей части. Значит, одна из них (P_r или P_s) вложена

в другую. Пусть $P_r \subseteq P_s$. Отсюда $P_i \subseteq P_r \subseteq P_s$ и $P_k \subseteq P_s$. Тогда из определения операции (\sim) следует, что $P_i \sim P_k$.

Следствие. Для изолиний $P_{k_1} \sim P_{k_2} \sim \dots \sim P_{k_{n_k}}$ существует изолиния $P_r \in \Pi$, охватывающая каждую из них $P_{k_i} \subseteq P_r$, $i = 1, 2, \dots, n_k$.

Компоненты связности. Определенное на множестве изолиний $\Pi = \{P_1, P_2, \dots, P_m\}$ отношение эквивалентности (\sim) выполняет разложение множества Π на непересекающиеся классы эквивалентности D_1, D_2, \dots, D_{n_D} , для которых $\Pi = D_1 \cup D_2 \cup \dots \cup D_{n_D}$, $D_i \cap D_j = \emptyset$, $i \neq j$ (рис. 3).

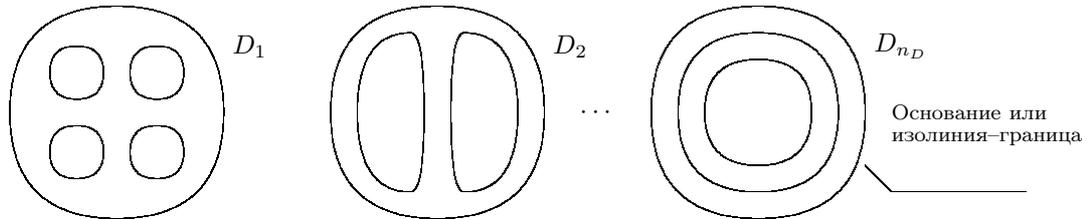


Рис. 3

Утверждение. Для каждого класса эквивалентности D_i , $i = 1, 2, \dots, n_D$, существует единственная изолиния $P_{r_i} \in D_i$, которая охватывает все множество изолиний компоненты D_i . Будем обозначать такую изолинию как основание D_i или как изолинию-границу D_i (рис. 3). Справедливость данного утверждения вытекает из последнего следствия.

Замечание. Классы эквивалентности D_i , $i = 1, 2, \dots, n_D$, далее будем обозначать как области изолиний или компоненты связности. Последнее название оправдано тем, что все изолинии одного класса D_i связаны в одну структуру вложенности с одним основанием.

Представление компонент связности изолиний. Идентификация принадлежности изолиний $\Pi = \{P_1, P_2, \dots, P_m\}$ компонентам связности D_1, D_2, \dots, D_{n_D} выполняется посредством значений элементов вектора $Z = \{Z_1, Z_2, \dots, Z_m\}$. Для каждой изолинии P_i определяется свой элемент Z_i с тем же индексом. Значение Z_i равно номеру k -й компоненты связности D_k , для которой $P_i \in D_k$. Таким образом, изолинии с одинаковыми значениями элементов вектора Z составляют одну из компонент связности, а число различных значений среди элементов вектора Z определяет количество таких компонент n_D . Назначение номеров областей D_1, D_2, \dots, D_{n_D} элементам вектора $Z = \{Z_1, Z_2, \dots, Z_m\}$ решается в рамках следующей редакции общего алгоритма выделения компонент связности [2, 4].

```
for i = 1 to m do  $Z_i = 0$ ; {Начальные номера областей  $D_1, D_2, \dots, D_{n_D}$  — их нет}
count = 0; {Количество областей  $D_1, D_2, \dots, D_{n_D}$ }
```

```
for k = 1 to m do begin {Перебор изолиний  $P_k$ }
  if  $Z_k = 0$  then begin {Начать новую область}
    count = count + 1; {Увеличить номер новой области}
     $Z_k = count$ ; {Отнести изолинию к новой области}
  end;
  for j = k + 1 to m do begin {Перебор изолиний  $P_j$ }
    if  $P_k \cap P_j \neq \emptyset$  then begin {Вложенность изолиний  $P_k$  и  $P_j$ }
      if  $Z_k \neq Z_j$  then begin {Слияние областей  $D_{Z_k} = D_{Z_k} \cup D_{Z_j}$ }
         $wZ = Z_j$ ; {Переназначить изолинии области  $D_{Z_j}$ }
        if  $Z_j \neq 0$  then for i = 1 to m do if  $Z_i = wZ$  then  $Z_i = Z_k$ ;
         $Z_j = Z_k$ ;
      end;
    end;
  end;
end;
end;
```

4. Линейная структура вложенности изолиний. Под линейной структурой вложенности поля изолиний P_1, P_2, \dots, P_m будем понимать упорядоченную их последовательность для каждой области D_i , $i = 1, 2, \dots, n_D$, таким образом, что порядковый номер по списку для каждой вложенной изолинии должен быть больше порядкового номера любой другой изолинии, которая охватывает ее. Другими словами, для каждой области D_i , $i = 1, 2, \dots, n_D$, вложенные изолинии упорядоченного списка P_1, P_2, \dots, P_m всегда располагаются правее изолиний, охватывающих их. Указанное расположение изолиний дает возможность сформировать градиентное их заполнение простым последовательным заполнением изолиний этого списка P_1, P_2, \dots, P_m соответствующими для них цветами. Решение задачи формирования линейной структуры

вложенности поля изолиний составляет основу данной работы. Дадим формальное определение этой последовательности.

Определение. Бинарное отношение (\triangleleft) неполного линейного порядка определяется для упорядоченного списка изолиний P_1, P_2, \dots, P_m . Отношение $P_i \triangleleft P_j$, где $i > j$, выполняется, если верно условие $P_i \subset P_j$ или $P_i \cap P_j = \emptyset$.

Утверждение. Упорядоченная последовательность изолиний P_1, P_2, \dots, P_m удовлетворяет свойству линейной структуры вложенности тогда и только тогда, когда эта последовательность удовлетворяет отношению неполного линейного порядка, т.е. $\forall P_i, P_j, i > j$, выполняется отношение $P_i \triangleleft P_j$.

Граф линейной структуры вложенности. Выполнение отношения неполного линейного порядка для последовательности P_1, P_2, \dots, P_m изолиний позволяет соотнести линейной структуре вложенности ориентированный граф $\Gamma(X, U)$, где $X = \{P_1, P_2, \dots, P_m\}$ — множество вершин, U — множество ребер и ориентированные ребра представляются парами вершин $u = (P_i, P_{i+1}) \in U, i = 1, 2, \dots, m - 1$. Отметим, что ребра $P_i \rightarrow P_{i+1}$ удовлетворяют отношению $P_{i+1} \triangleleft P_i$, так как $\forall P_i, P_j, i > j$ верно $P_i \triangleleft P_j$.

В данной интерпретации линейная структура вложенности изолиний представляется линейным деревом с ориентированными ребрами в направлении от корня. На рис. 4 показан пример поля изолиний P_1, P_2, \dots, P_m и несколько графов линейной структуры вложенности для них. Индексы изолиний (вершин) в графах представлены их исходными номерами.

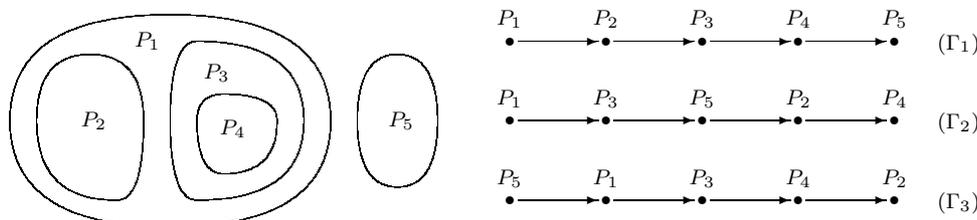


Рис. 4

Ориентированность ребер показывает последовательность обхода вершин (изолиний) графа при градиентном их заполнении.

5. Корневое дерево структуры вложенности изолиний. Введение нового отношения на множестве изолиний $\Pi = \{P_1, P_2, \dots, P_m\}$ преследует цель определения новых структур на этом множестве, которые бы обеспечили доступ к перечню изолиний P_1, P_2, \dots, P_m в соответствии с их структурой вложенности.

Определение. Бинарное отношение (\prec) охватывающего соседства определяется на множестве $\Pi = \{P_1, P_2, \dots, P_m\}$. Отношение $P_i \prec P_j$ выполняется, если $P_i \subset P_j$ и $\forall P_k \neq P_j (P_i \subset P_k \Rightarrow P_j \subset P_k)$. Это означает, что P_j — наименьшая охватывающая изолиния для P_i .

Граф структуры вложенности изолиний. Поставим в соответствие структуре вложенности изолиний P_1, P_2, \dots, P_m ориентированный граф $\Gamma = (X, U)$, где $X = \{P_1, P_2, \dots, P_m\}$ — множество вершин и U — множество ребер; ребра представляются парами вершин. Так, ориентированное ребро $u = (P_i, P_j) \in U$, если для изолиний P_i и P_j выполняется отношение $P_i \prec P_j$. Отметим, что направленное ребро (P_i, P_j) ориентировано от вложенной изолинии P_i к наименьшей охватывающей ее P_j ($P_i \rightarrow P_j$).

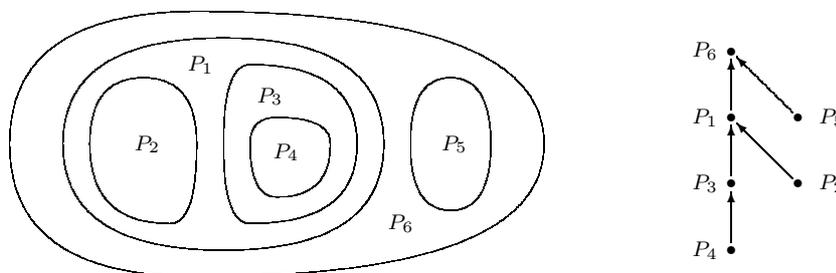


Рис. 5

Каждая область изолиний D_k может быть представлена корневым деревом. Корнем дерева будет основание (изолиния–граница) компоненты D_k . Все ребра ориентированы в направлении от листьев к корню и показывают для каждой изолинии (вершины) направление доступа к соседней охватывающей ее изолинии (вершины). На рис. 5 показан пример области изолиний D_k и соответствующий ей ориентированный граф — корневое дерево. Каждая вершина такого графа имеет соседнюю охватывающую вершину, которой нет у корня дерева, так как изолиния основания области D_k не имеет охватывающей изолинии.

Замечание. Рассмотренное корневое дерево имеет важное свойство, которое допускает последовательное перемещение по структуре вложенности соседних изолиний от произвольной изолинии к изолинии основания области D_k . Именно эта структура корневого дерева представления вложенности изолиний используется при получении линейной структуры вложенности изолиний.

Представление корневого дерева. Важно, что в рамках данной задачи представление корневого дерева можно выполнить довольно просто посредством вектора $Q = (Q_1, Q_2, \dots, Q_m)$, где элемент Q_i определяется для изолинии (вершины) P_i и равен номеру наименьшей охватывающей ее изолинии (соседней вершины по направлению ребра). Таким образом, $Q_i = j$, если выполняется $P_i \prec P_j$. Изолинии оснований областей D_1, D_2, \dots, D_{n_D} не имеют охватывающих. Для таких изолиний полагаем $Q_i = -1$.

6. Обратная перестановка изолиний. Алгоритм формирования линейной структуры вложенности использует сортировку изолиний P_1, P_2, \dots, P_m и явно изменяет их исходные номера следования. Это необходимо требует приведения в соответствие номеров соседних охватывающих изолиний в векторе $Q = (Q_1, Q_2, \dots, Q_m)$ представления корневого дерева структуры вложенности изолиний. Пусть $\pi^{-1} = (\pi_1^{-1}, \pi_2^{-1}, \dots, \pi_m^{-1}) = \begin{pmatrix} 1 & 2 & \dots & m \\ \pi_1^{-1} & \pi_2^{-1} & \dots & \pi_m^{-1} \end{pmatrix}$ — перестановка исходных номеров изолиний P_1, P_2, \dots, P_m . Здесь $1, 2, \dots, m$ — текущие новые порядковые номера изолиний после их перестановки. Значение элемента π_k^{-1} равно исходному номеру изолинии, которая сейчас занимает k -ю позицию. Данная интерпретация значения элемента π_k^{-1} позволяет сказать, что перестановка π^{-1} является обратной, так как π_k^{-1} показывает не новое место k -й изолинии, а величину, обратную к ней. Чтобы определить текущий номер изолинии в сортированном списке P_1, P_2, \dots, P_m , если в исходном перечне изолиний ее номер был Q_i , необходимо выполнить следующие действия. По номеру Q_i находим в списке $\pi_1^{-1}, \pi_2^{-1}, \dots, \pi_m^{-1}$ элемент π_k^{-1} , равный Q_i . Тогда номер искомой изолинии будет равен k . Чтобы не просматривать список $\pi_1^{-1}, \pi_2^{-1}, \dots, \pi_m^{-1}$ в поиске элемента Q_i , выполним преобразования в перестановке π^{-1} . Поменяем местами строки перестановки π^{-1} . Получим новую перестановку $\pi = \begin{pmatrix} \pi_1^{-1} & \pi_2^{-1} & \dots & \pi_m^{-1} \\ 1 & 2 & \dots & m \end{pmatrix}$. Далее сортируем столбцы новой перестановки по возрастанию элементов верхней строки. Получим $\pi = (\pi_1, \pi_2, \dots, \pi_m) = \begin{pmatrix} 1 & 2 & \dots & m \\ \pi_1 & \pi_2 & \dots & \pi_m \end{pmatrix}$ — это обратная перестановка к π^{-1} . Теперь новый номер изолинии для исходного ее номера Q_i будет определяться как $\pi[Q_i]$. По обратной перестановке π^{-1} восстановить перестановку π можно в цикле [2]: **for** $k = 1$ **to** m **do** $\pi[\pi_k^{-1}] = k$.

7. Цвет заполнения изолиний. Введем понятие *цвета заполнения* или просто цвета изолинии. Под этим будем понимать числовое значение, которое будет выступать в качестве цвета заполнения области, охватываемой изолинией. Градиентное заполнение поля изолиний предполагает назначение изолиниям различных цветов, соотношенных каким-то образом со значениями на изолиниях. Именно свойство градиентного заполнения не позволяет непосредственно использовать значения на изолиниях в качестве значений цветов. Так, например, поле изолиний на рис. 6 включает ряд соседних изолиний с равными значениями, фиксируя тем самым соответствующие изменения значений поля между такими изолиниями. Использование же значений на изолиниях в качестве цветовых значений их заполнения “смажет” одним цветом все изолинии значения 3 и 4 (рис. 6), что приведет к упрощению и искажению действительного состояния поля. С другой стороны, как будет показано ниже, поле изолиний на рис. 6 допускает однозначную градиентную раскраску с правильными оттенками заполнения соседних изолиний для равных значений.

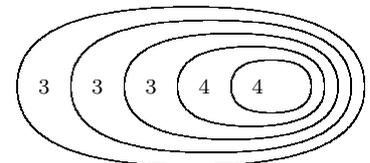


Рис. 6

Возникающие подобного рода “правильные” неоднозначности градиентной раскраски предлагается разрешить посредством некоторой процедуры (назовем ее *процедурой разметки* поля изолиний). Суть процедуры составляет подход, состоящий в том, что в качестве цвета заполнения изолиний назначается величина, равная скорректированному значению на этой изолинии. Изменение данного значения возможно как в сторону его увеличения, так и в сторону уменьшения. Знак изменения определяется по контексту значений всего поля изолиний. Величина модификации составляет половину шага частоты проведения изолиний.

8. Градиентное назначение меток изолиниям. Процедура разметки изолиний выполняет анализ значений поля для выделения пар соседних изолиний с *разными значениями*. Изменение значения поля на участке между парами таких изолиний позволяет однозначно установить метки этим изолиниям. Метки назначаются двух видов: “+” и “-”. Правило назначения меток объясняется на примере полей изолиний на рис. 7 а и 7 б. Для случая на рис. 7 а, когда наблюдается рост значения поля в глубину вложенности, изолиниям 3 и 4 назначаются метки “+”, а значение цвета их заполнения вычисляется, соответственно, как $3 + 0,5 = 3,5$ и $4 + 0,5 = 4,5$. Для случая на рис. 7 б, когда наблюдается падение значения поля в глубину

вложенности, изолиниям 4 и 3 назначаются метки “-”, а значение цвета их заполнения вычисляется, соответственно, как $4 - 0,5 = 3,5$ и $3 - 0,5 = 2,5$.

Замечание. Метка изолинии показывает рост (“+”) или падение (“-”) исходного поля в области, заключенной в границах между рассматриваемой изолинией и соседней охватывающей ее.

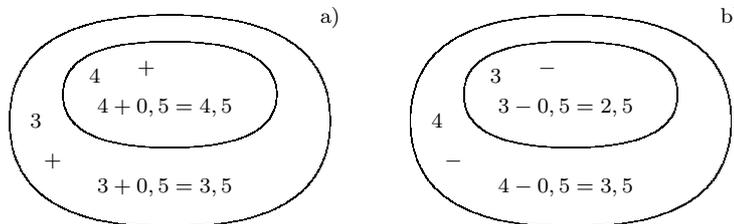


Рис. 7

Представление меток. Метки будем хранить в векторе $Mark = (Mark_1, Mark_2, \dots, Mark_m)$, где значение $Mark_i = 1$ соответствует метке “+”, $Mark_i = -1$ — метке “-”. Если $Mark_i = 0$, то метка изолинии P_i не назначена.

Приведем фрагмент алгоритма назначения градиентных меток $Mark_1, Mark_2, \dots, Mark_m$. Вложенность изолиний определяется по вектору $Q = (Q_1, Q_2, \dots, Q_m)$ представления корневого дерева структуры вложенности изолиний P_1, P_2, \dots, P_m .

```

for i = 1 to m do Marki = 0; {Начальные метки изолиний Pi — их нет}
{Назначение градиентных меток}
for i = m to 1 by -1 do begin {Перебор изолиний P1, P2, ..., Pm}
  if Qi = -1 then continue; {Изолиния-граница области — нет соседней}
  k = π[Qi]; {Номер соседней (охватывающей)}
  aU = Ui; {Значение на текущей изолинии Pi}
  bU = Uk; {Значение на охватывающей изолинии Pk}
  if (aU = bU) or (bU = +∞) then continue; {Предохранитель разметки}
  if aU > bU then Markk = Marki = 1 else Markk = Marki = -1; {Градиентные метки}
end
    
```

9. Распространение меток. Заметим, что после градиентного назначения меток непомеченными могут остаться изолинии, которые имеют в качестве соседних изолинии с равными значениями (рис. 8).

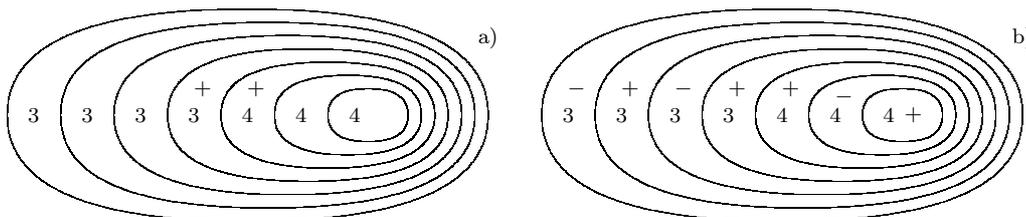


Рис. 8

Свойство вложенности изолиний отражать рост или падение поля позволяет распространить назначенные градиентные метки на непомеченные изолинии. Нетрудно заметить, что метки для оставшихся непомеченных изолиний необходимо назначить чередованием меток “+” и “-” относительно градиентных меток. Именно такое назначение меток отражает ход поведения исходного поля в области назначения меток. Так, метка “+” изолинии показывает превышение значения поля в назначенной области относительно значения этой изолинии. Наоборот, метка “-” изолинии показывает падение значения поля в указанной области. На рис. 8 b) показан пример назначения чередованием меток “+” и “-” для поля на рис. 8 а. Относительно назначенных меток вычисляются и значения цветов заполнения изолиний. Метка “+” изолинии соответствует значению цвета, равного сумме значений на изолинии и половины шага частоты проведения изолиний. Метка “-” изолинии соответствует значению цвета, равного разности значений на изолинии и половины шага частоты проведения изолиний.

Замечание. Изолинии оснований областей D_1, D_2, \dots, D_{n_D} являются составляющими общего перечня изолиний P_1, P_2, \dots, P_m . Следовательно, распространение чередования меток выполняется и для них. Если границам будут назначены метки, то цвета их заполнения вычисляются по их меткам и значениям

на соседних вложенных изолиниях. На рис. 9 приведен пример назначения цвета для границы. Соседней с границей является изолиния 3, а так как метка границы равна “-”, то значение цвета заполнения границы составит $3 - 0,5 = 2,5$.

10. Проход от центров вложенности изолиний к их основанию (сверху-вниз). Проход сверху-вниз устанавливает чередование меток для неотмеченных изолиний и назначает метки и значения изолиниям-основаниям областей D_1, D_2, \dots, D_{n_D} . Приведем фрагмент алгоритма назначения чередования меток $Mark_1, Mark_2, \dots, Mark_m$ с использованием прохода сверху-вниз по *линейной структуре вложенности изолиний* P_1, P_2, \dots, P_m .

```

for  $i = m$  to 1 by -1 do begin {Перебор изолиний  $P_1, P_2, \dots, P_m$ }
  if  $Q_i = -1$  then continue; {Изолиния-граница области — нет соседней}
   $k = \pi[Q_i]$ ; {Номер соседней изолинии по дереву вложенности}
  if  $Mark_k = 0$  or  $Mark_k \neq 0$  then continue; { $P_i$  без метки или соседняя  $P_k$  с меткой}
   $Mark_k = -Mark_i$ ; {Чередование меток изолиний}
  if  $U_k = +\infty$  then  $U_k = U_i$ ; {Назначить значение изолинии-границы}
end

```

Рассмотренный проход сверху-вниз по линейной структуре вложенности P_1, P_2, \dots, P_m не может назначить метки тем изолиниям, которые находятся выше установленных градиентных меток по вложенности изолиний. На рис. 10 приведен пример указанного случая. Последние две вложенные изолинии нельзя пометить проходом сверху-вниз.

11. Дополнительное определение меток оснований областей.

В данном пункте речь идет о тех случаях, когда основаниям областей изолиний не будут назначены метки. Пример такой области показан на рис. 11. В общем случае эти ситуации возникают для полей, которые не содержат вложенных изолиний с различными значениями. Такие поля не допускают назначения градиентных меток; как следствие, нельзя назначить метки и значения изолиниям-границам областей.

Геометрия поля изолиний для такого рода областей D_k (рис. 11) подсказывает, что значение на изолинии-границе надо положить равным минимальному значению на изолиниях рассматриваемой области D_k и назначить метку “+”. Назначение метки “-” возможно, когда все изолинии области D_k имеют равные значения. Приведем фрагмент алгоритма назначения меток $Mark_1, Mark_2, \dots, Mark_m$ и значений изолиниям-основаниям областей.

```

for  $i = 1$  to  $m$  do begin {Перебор изолиний  $P_1, P_2, \dots, P_m$ }
  if  $U_i \neq +\infty$  then continue; {Изолиния со значением}
   $bU = +\infty$ ; {Поиск соседней изолинии с минимальным значением}
  for  $j = 1$  to  $m$  do begin {Перебор изолиний  $P_1, P_2, \dots, P_m$ }
    if  $Q_j = -1$  or  $\pi[Q_j] \neq i$  then continue; {Изолиния  $P_j$  не соседняя для  $P_i$ }
    if  $bU > U_j$  then  $bU = U_j$ ; {Выбор минимального значения}
  end;
  if  $bU \neq +\infty$  then begin {Область  $D_{Z[i]}$  непустая}
     $Mark_i = +1$ ; {Начальная метка области}
     $U_i = bU$ ; {Значение на изолинии-границе области}
  end;
end

```

12. Проход от оснований к центрам вложенности (снизу-вверх). Дополнительный проход снизу-вверх позволяет назначить метки всем оставшимся неотмеченным изолиниям. Достаточным условием для назначения меток всем изолиниям является наличие меток у оснований областей D_i . Приведем фрагмент алгоритма окончательного назначения меток $Mark_1, Mark_2, \dots, Mark_m$ изолиниям проходом снизу-вверх по *линейной структуре вложенности изолиний* P_1, P_2, \dots, P_m .

```

for  $i = 1$  to  $m$  do begin {Перебор изолиний  $P_1, P_2, \dots, P_m$ }
  if  $Mark_i \neq 0$  then continue; {Изолиния  $P_i$  с меткой — назначать не надо}
   $k = \pi[Q_i]$ ; {Номер соседней изолинии  $P_k$ , соседние все с метками}
   $Mark_i = -Mark_k$ ; {Чередование меток}
end

```

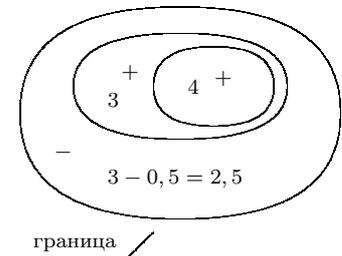


Рис. 9

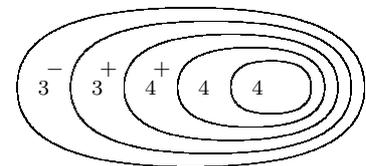


Рис. 10

13. Критерий проверки вложенности изолиний. Базовыми операциями над множеством изолиний $\Pi = \{P_1, P_2, \dots, P_m\}$ являются операции парного сравнения $\alpha, \beta \in \Pi$ относительно их взаимного расположения: $\alpha \cap \beta = \emptyset$, $\alpha \subset \beta$ или $\beta \subset \alpha$. В качестве критерия проверки соотношений $\alpha \cap \beta = \emptyset$, $\alpha \subset \beta$, $\beta \subset \alpha$ предлагается использовать следующий анализ данных об изолиниях α и β . Во-первых, определяется наличие непустого пересечения $\alpha \cap \beta \neq \emptyset$. Во-вторых, при $\alpha \cap \beta \neq \emptyset$ проверяется условие включения одной изолинии в другую: $\alpha \subset \beta$ или $\beta \subset \alpha$.

Проверка условия $\alpha \cap \beta \neq \emptyset$. Пусть γ (назовем ее *контрольной прямой*) параллельна оси OX и пересекает указанные изолинии α и β . Изолиния α , пересекая прямую γ , отсекает на ней отрезки $[\alpha_i, \alpha_{i+1}]$, $i = 1, 2, \dots, k_\alpha$, а изолиния β — отрезки $[\beta_j, \beta_{j+1}]$, $j = 1, 2, \dots, k_\beta$. Тогда наличие вложенности изолиний α и β устанавливается на основании непустого пересечения множеств $\mu_\alpha = \bigcup_{i=1}^{k_\alpha} [\alpha_i, \alpha_{i+1}]$ и $\mu_\beta = \bigcup_{j=1}^{k_\beta} [\beta_j, \beta_{j+1}]$.

Проверка условия $\alpha \subset \beta$ или $\beta \subset \alpha$. Положим значения элементов вектора S_1, S_2, \dots, S_m равными соответствующим площадям, заключенным в изолиниях P_1, P_2, \dots, P_m . Тогда характер вложенности изолиний α в β или β в α устанавливается из соотношения их площадей S_α и S_β . Так, если $S_\alpha > S_\beta$, то изолиния β вложена в α , и наоборот, если $S_\alpha < S_\beta$, то изолиния α вложена в β .

Формирование отрезков $[\alpha_i, \alpha_{i+1}]$ и $[\beta_j, \beta_{j+1}]$ осуществляется по найденным точкам α_i и β_j пересечения контрольной прямой γ с изолиниями α и β (рис. 12). Последовательные упорядоченные по возрастанию пары значений α_i и β_j составят указанные отрезки $[\alpha_i, \alpha_{i+1}]$ и $[\beta_j, \beta_{j+1}]$. Обозначим процедуру формирования этих отрезков через $Create(\gamma, \alpha, \beta)$. Важной составляющей проверки условия пересечения двух областей-изолиний является выбор положения *контрольной прямой* γ . Для любой пары вложенных изолиний

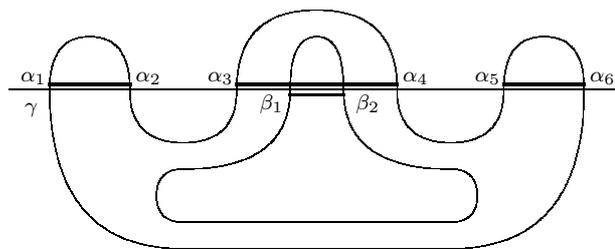


Рис. 12

прямая γ должна пересекать каждую из них. Предлагается следующее решение данной задачи. Обозначим через R_1, R_2, \dots, R_m вектор прямоугольников, где R_i — наименьший из прямоугольников, охватывающий изолинию P_i (рис. 13 а), со сторонами, параллельными осям координат. Координаты нижнего левого R_k -угла составят $(\min_i x_{ki}, \min_i y_{ki})$, а верхнего правого угла — $(\max_i x_{ki}, \max_i y_{ki})$.

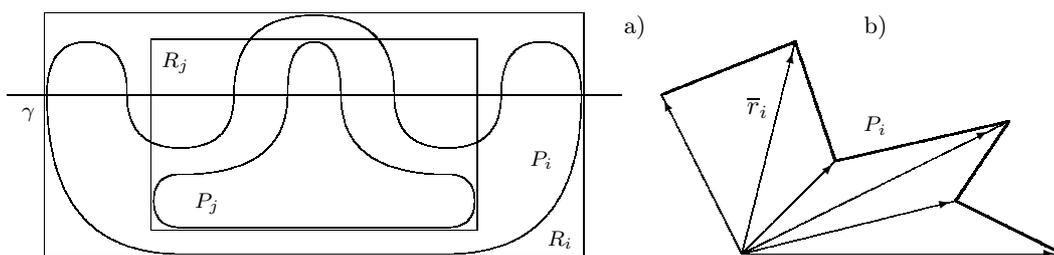


Рис. 13

Отметим, что результатом пересечения прямоугольников $R_i \cap R_j$ является прямоугольник, возможно пустой. При вложенном взаимном расположении изолиний P_i и P_j пересечение соответствующих прямоугольников $R_i \cap R_j$ будет совпадать с одним из них (рис. 13 а). Контрольная прямая γ для вложенных изолиний P_i и P_j будет их пересекать, если она располагается в пределах прямоугольника пересечения $R_i \cap R_j$. Центральная точка этого прямоугольника является наиболее очевидной для проведения через нее контрольной прямой γ .

Вычисление площадей S_1, S_2, \dots, S_m . На рис. 13 б представлена изолиния P_i и радиус-векторы \vec{r}_i , $i = 1, 2, \dots, n_k$, ее опорных точек относительно начальной точки. Координаты вектора $\vec{r}_i = (x_{ki} - x_{k1}, y_{ki} - y_{k1})$, где (x_{ki}, y_{ki}) — координаты i -й опорной точки изолинии P_k . Опорные точки (x_{ki}, y_{ki}) изолинии упорядочены по периметру. Вследствие этого площадь S_k , охватываемая такой изолинией P_k , может быть вычислена как длина вектора $\sum_{i=2}^{n_k-1} \vec{r}_i \times \vec{r}_{i+1}$.

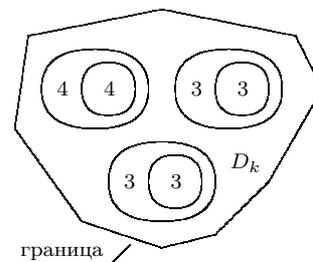


Рис. 11

14. Метод формирования линейной структуры вложенности изолиний. Линейная структура вложенности изолиний P_1, P_2, \dots, P_m (см. п. 4) имеет такое свойство: вложенные изолинии по этому списку следования всегда располагаются правее изолиний, охватывающих их. Формирование линейной структуры вложенности выполняется неявно посредством вычисления значений вектора счетчиков C_1, C_2, \dots, C_m . Элемент C_i равен количеству изолиний из общего их перечня P_1, P_2, \dots, P_m , каждая из которых охватывает рассматриваемую изолинию P_i . Так, для примера, на рис. 14 значения элементов счетчиков равны: $C_1 = 0, C_2 = 1, C_3 = 2$.

Для размещения последовательности изолиний P_1, P_2, \dots, P_m в порядке следования, отвечающего структуре вложенности, достаточно выполнить их сортировку согласно возрастанию значений счетчиков $C_i, i = 1, 2, \dots, m$, для каждой области D_k . Теперь последовательное рисование изолиний P_1, P_2, \dots, P_m будет отвечать такому свойству: каждая вложенная изолиния будет размещаться поверх охватывающей изолинии. Формирование значений $C_i, i = 1, 2, \dots, m$, предлагается выполнить способом сравнения каждой изолинии с каждой на предмет включения одной изолинии в другую, что соответствует методу сортировки перечислением [2, 4].

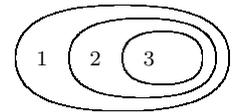


Рис. 14

Для линейной структуры вложенности не требуется упорядочивания изолиний P_1, P_2, \dots, P_m по компонентам связности D_k . С другой стороны, в ряде задач удобно иметь последовательную векторную структуру изолиний, распределенных по D_k .

15. Алгоритм формирования линейной структуры вложенности изолиний P_1, P_2, \dots, P_m .

Шаг 1. Вычисление площадей S_1, S_2, \dots, S_m .

Шаг 2. Вычисление координат охватывающих прямоугольников R_1, R_2, \dots, R_m .

Шаг 3. Формирование счетчиков вложенности изолиний C_1, C_2, \dots, C_m , номеров минимальных охватывающих изолиний Q_1, Q_2, \dots, Q_m , номеров Z_1, Z_2, \dots, Z_m областей D_1, D_2, \dots, D_{n_D} размещения изолиний.

```

for  $i = 1$  to  $m$  do begin {Начальная установка данных}
   $C_i = 0$ ; {Значения счетчиков}
   $Q_i = -1$ ; {Номера наименьших охватывающих изолиний — их нет}
   $sQ_i = +\infty$ ; {Площади наименьших охватывающих изолиний}
   $Z_i = 0$ ; {Номера областей  $D_1, D_2, \dots, D_{n_D}$  размещения изолиний — их нет}
end;
 $count = 0$ ; {Количество областей}

for  $k = 1$  to  $m$  do begin {Перебор изолиний  $P_k$ }
  if  $Z_k = 0$  then begin {Начать новую компоненту связности (область)}
     $count = count + 1$ ; {Увеличить номер новой области}
     $Z_k = count$ ; {Отнести изолинию к новой области}
  end;
  for  $j = k + 1$  to  $m$  do begin {Перебор изолиний  $P_j$ }
    if  $R_k \cap R_j = \emptyset$  then continue; {Нет пересечения прямоугольников}
     $\gamma$  — контрольная прямая для центра пересечения прямоугольников  $R_k \cap R_j$ 
     $\{\alpha, \beta$  — изолинии  $P_k$  и  $P_j\}$ 
     $Create(\gamma, \alpha, \beta)$ ; {Строим отрезки отсечения  $[\alpha_i, \alpha_{i+1}]$ ,  $[\beta_j, \beta_{j+1}]$  на прямой  $\gamma$ }
     $\mu_\alpha = \bigcup_{i=1}^{k_\alpha} [\alpha_i, \alpha_{i+1}]$ ; {Множество отрезков отсечения для  $\alpha = P_k$ }
     $\mu_\beta = \bigcup_{i=1}^{k_\beta} [\beta_i, \beta_{i+1}]$ ; {Множество отрезков отсечения для  $\beta = P_j$ }
     $\ell_\alpha = |\mu_\alpha|$ ; {Длина отрезков отсечения для  $\alpha = P_k$ }
     $\ell_\beta = |\mu_\beta|$ ; {Длина отрезков отсечения для  $\beta = P_j$ }
    if  $\mu_\alpha \cap \mu_\beta \neq \emptyset$  then begin {Вложенность изолиний  $\alpha = P_k$  и  $\beta = P_j$ }
      if  $S_k < S_j$  then  $C_k = C_k + 1$  else  $C_j = C_j + 1$ ; {Счетчики вложенности изолиний}
      if  $Z_k \neq Z_j$  then begin {Слияние компонент связности  $D_{Z_k} = D_{Z_k} \cup D_{Z_j}$ }
         $wZ = Z_j$ ; {Переназначить изолинии компоненты  $D_{Z_j}$ }
        if  $Z_j \neq 0$  then for  $i = 1$  to  $m$  do if  $Z_i = wZ$  then  $Z_i = Z_k$ ;
         $Z_j = Z_k$ ;
      end;
    if  $\ell_\alpha \geq \ell_\beta$  then {Изолиния  $\beta = P_j$  вложена в  $\alpha = P_k$ }
      if  $sQ_j \geq S_k$  then begin {Уточнить соседнюю снизу для  $\beta = P_j$ }
         $sQ_j = S_k$ ;
      end;
  end;

```

```

        Qj = k;
    end;
    else {Изолиния α = Pk вложена в β = Pj}
        if sQk ≥ Sj then begin {Уточнить соседнюю снизу для α = Pk}
            sQk = Sj;
            Qk = j;
        end;
    end;
end;
end;
end

```

Шаг 4. Сортировка изолиний P_1, P_2, \dots, P_m по возрастанию значений счетчиков уровней вложенности C_1, C_2, \dots, C_m для каждой области. Номера областей изолиний установлены в векторе Z_1, Z_2, \dots, Z_m . Применяется метод пузырьковой сортировки в его простой редакции [2, 4].

```

{Начальная пождественная перестановка π-1 изолиний P1, P2, ..., Pm}
for i = 1 to m do πi-1 = i;
{Сортировка P1, P2, ..., Pm по номерам вложенности C1, C2, ..., Cm}
for i = 1 to m do begin
    for j = 1 to m - i do begin
        if Cj > Cj+1 then begin {Номера уровней вложенности}
            w = Cj; Cj = Cj+1; Cj+1 = w;
            w = Zj; Zj = Zj+1; Zj+1 = w; {Сохраняем номера областей}
            wP = Pj; Pj = Pj+1; Pj+1 = wP; {Переставить изолинии}
            w = Qj; Qj = Qj+1; Qj+1 = w; {Соседние изолинии в исходной нумерации}
            w = πj-1; πj-1 = πj+1-1; πj+1-1 = w; {Сохраняем начальные номера изолиний}
        end;
    end;
end;
end

```

```

{Сортировка P1, P2, ..., Pm по областям, номера областей в Z1, Z2, ..., Zm.}
{Сохраняется упорядочение по номерам вложенности C1, C2, ..., Cm}
{внутри каждой области D1, D2, ..., DnD}

```

```

for i = 1 to m do begin
    for j = 1 to m - i do begin
        if Zj > Zj+1 then begin {Номера областей}
            w = Cj; Cj = Cj+1; Cj+1 = w;
            w = Zj; Zj = Zj+1; Zj+1 = w;
            wP = Pj; Pj = Pj+1; Pj+1 = wP; {Переставить изолинии}
            w = Qj; Qj = Qj+1; Qj+1 = w; {Соседние изолинии в исходной нумерации}
            w = πj-1; πj-1 = πj+1-1; πj+1-1 = w; {Сохраняем начальные номера изолиний}
        end;
    end;
end;
end

```

16. Градиентное назначение меток $Mark_1, Mark_2, \dots, Mark_m$ изолиниям P_1, P_2, \dots, P_m .

Шаг 1. По обратной перестановке π^{-1} восстановить перестановку π — отображение исходных номеров изолиний в их сортированную нумерацию.

```

for i = 1 to m do π[πk-1] = k; {Исходная перестановка}

```

Шаг 2. Начальное градиентное назначение меток $Mark_1, Mark_2, \dots, Mark_m$.

```

for i = 1 to m do Marki = 0; {Начальные метки изолиний Pi — их нет}
{Назначение градиентных меток}
for i = m to 1 by -1 do begin {Перебор изолиний P1, P2, ..., Pm}
    if Qi = -1 then continue; {Изолиния-граница области — нет соседней}
    k = π[Qi]; {Номер соседней (охватывающей)}
    aU = Ui; {Значение на текущей изолинии Pi}
    bU = Uk; {Значение на охватывающей изолинии Pk}
    if (aU = bU) or (bU = +∞) then continue; {Предохранитель разметки}
    if aU > bU then Markk = Marki = 1 else Markk = Marki = -1; {Градиентные метки}
end

```

Шаг 3. Распространение градиентных меток $Mark_1, Mark_2, \dots, Mark_m$.

```
{Проход сверху–вниз по корневому дереву вложенности изолиний}
for  $i = m$  to 1 by  $-1$  do begin {Перебор изолиний  $P_1, P_2, \dots, P_m$ }
  if  $Q_i = -1$  then continue; {Изолиния–граница области — нет соседней}
   $k = \pi[Q_i]$ ; {Номер соседней изолинии по дереву вложенности}
  if  $Mark_i = 0$  or  $Mark_k \neq 0$  then continue; { $P_i$  без метки или соседняя  $P_k$  с меткой}
   $Mark_k = -Mark_i$ ; {Чередование меток изолиний}
  if  $U_k = +\infty$  then  $U_k = U_i$ ; {Назначить значение изолинии–границы}
end
```

```
{Дополнительное определение меток оснований областей}
for  $i = 1$  to  $m$  do begin {Перебор изолиний  $P_1, P_2, \dots, P_m$ }
  if  $U_i \neq +\infty$  then continue; {Изолиния со значением}
   $bU = +\infty$ ; {Поиск соседней изолинии с минимальным значением}
  for  $j = 1$  to  $m$  do begin {Перебор изолиний  $P_1, P_2, \dots, P_m$ }
    if  $Q_j = -1$  or  $\pi[Q_j] \neq i$  then continue; {Изолиния  $P_j$  не соседняя для  $P_i$ }
    if  $bU > U_j$  then  $bU = U_j$ ; {Выбор минимального значения}
  end;
  if  $bU \neq +\infty$  then begin {Область  $D_{Z[i]}$  непустая}
     $Mark_i = +1$ ; {Начальная метка области}
     $U_i = bU$ ; {Значение на изолинии–границе области}
  end;
end
```

```
{Проход снизу–вверх по корневому дереву вложенности изолиний}
for  $i = 1$  to  $m$  do begin {Перебор изолиний  $P_1, P_2, \dots, P_m$ }
  if  $Mark_i \neq 0$  then continue; {Изолиния  $P_i$  с меткой — назначать не надо}
   $k = \pi[Q_i]$ ; {Номер соседней изолинии  $P_k$ , соседние все с метками}
   $Mark_k = -Mark_i$ ; {Чередование меток}
end
```

17. Алгоритм градиентного назначения цветов числовой последовательности $Color_1, Color_2, \dots, Color_m$ изолиниям P_1, P_2, \dots, P_m .

```
{Проход по линейной структуре вложенности изолиний  $P_1, P_2, \dots, P_m$ }
for  $i = 1$  to  $m$  do begin {Перебор изолиний  $P_1, P_2, \dots, P_m$ }
  if  $Mark_i \neq 0$  then begin {Контроль назначения меток изолиниям; 0 - для пустых областей}
    if  $Mark_i > 0$  then  $Color_i = U_i + delta/2$  else  $Color_i = U_i - delta/2$ ;
    Заполнить изолинии  $P_i$  цветом  $Color_i$ ;
  end;
end
```

18. Сложность алгоритма. Пусть число изолиний равно m . Положим, что количество точек на каждой изолинии определяется числом $O(n)$ относительно n . Тогда сложность алгоритма градиентного заполнения составит $O(m^2 \cdot n)$. Это следует из описания алгоритма, где явно представлены два вложенных цикла размерности m и неявно вложенный цикл формирования отрезков отсечения размерности n .

Предложенный алгоритм реализован в рамках программного комплекса “Океан” [1]. Он может быть применен как при формировании стандартных карт погоды для заполнения полей изолиний цветовой гаммой, так и при формировании цветового заполнения динамических карт мультипликации.

СПИСОК ЛИТЕРАТУРЫ

1. Иванов Б.Н. Автоматизированный комплекс “Океан” обеспечения безопасности полетов и мореплавания // Труды ДВГТУ. Вып. 136. Владивосток, 2004.
2. Иванов Б.Н. Дискретная математика. Алгоритмы и программы. М.: Лаборатория базовых знаний, 2003.
3. Баяковский Ю.М., Галактионов В.А., Михайлова Т.Н. Графор. Графическое расширение фортрана. М.: Наука, 1985.
4. Рейнгольд Э., Нивергельт Ю., Део Н. Комбинаторные алгоритмы. Теория и практика. М.: Мир, 1980.

Поступила в редакцию
07.02.2006