

В.И.ДВОРЖЕЦ
КОМПЛЕКТ ПРОЦЕДУР ВЫВОДА ГРАФИКОВ.

I. Общая постановка задачи.

Обычно при выводе графиков выделяют две задачи - вычерчивание осей координат и собственно рисование графиков. Для работы задается некоторое поле. На нем может быть нарисовано несколько графиков в одной системе координат. Задать функцию, график которой необходимо выдать, можно одним из трех видов: явно, параметрически или сеточно. Оси координат рисуются в виде двух перпендикулярных векторов со штриховкой и маркировкой (цифровкой).

Естественно возникает разделение функций процедур комплекта. Одна процедура - с и к о р - вычерчивает оси координат со штриховкой и маркировкой, а другие вычерчивают на определевшемся поле линии графиков.

При разработке комплекта основное внимание было обращено на создание максимального разнообразия возможностей и, вместе с тем, максимальную простоту обращения с комплектом.

Например, связь между с и к о р и остальными процедурами осуществляется только через процедуры первого уровня (см. [1]). А именно, в конце своей работы процедура с и к о р устанавливает декартовую систему координат на листе в соответствие со штриховкой осей. Процедуры вычерчивания узнают масштабы и начало координат с помощью процедур л и к и д е к из первого уровня. С и к о р можно использовать не только в комплекте "график". Например, для отрисовки осей координат при вычерчивании изолиний, векторных полей, линий тока и т.д.

Соответственно, другие процедуры вывода графиков могут быть использованы без процедуры с и к о р . Ведь декартовую систему координат нужного вида можно получить и просто обращением к процедуре д е к а р (из [1]).

Процедура г ф предназначена для вычерчивания графиков явно, а г ф п - параметрически заданных функций. Для

г ф лин и г ф кин функция задается таблично. Они производят соответственно линейную или кубическую интерполяции и обращаются к г ф . Эти процедуры, кроме вычерчивания графиков, могут использоваться для черчения различных кривых, например, ограничивающих какую-либо область.

2. Процедура с и к о р .

Процедура с и к о р (x_n , x_k , шаг x , u_n , u_k , шаг y , n_x , n_y).

Все параметры вещественные. Последние два параметра задают размер поля в миллиметрах, на котором будут выдаваться графики. Размер оси x на 20 мм, а y на 10 мм меньше. Слева от оси y и ниже оси x оставляется место для маркировки.

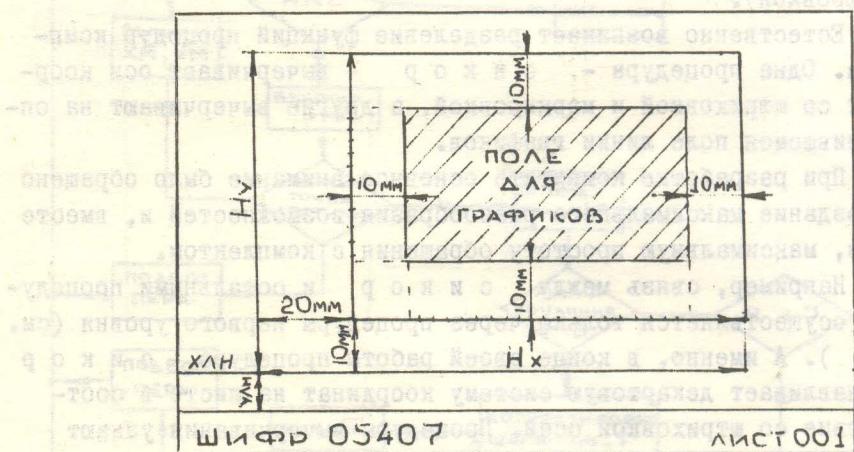


Рис. I

x_n , x_k - диапазон значений x ;

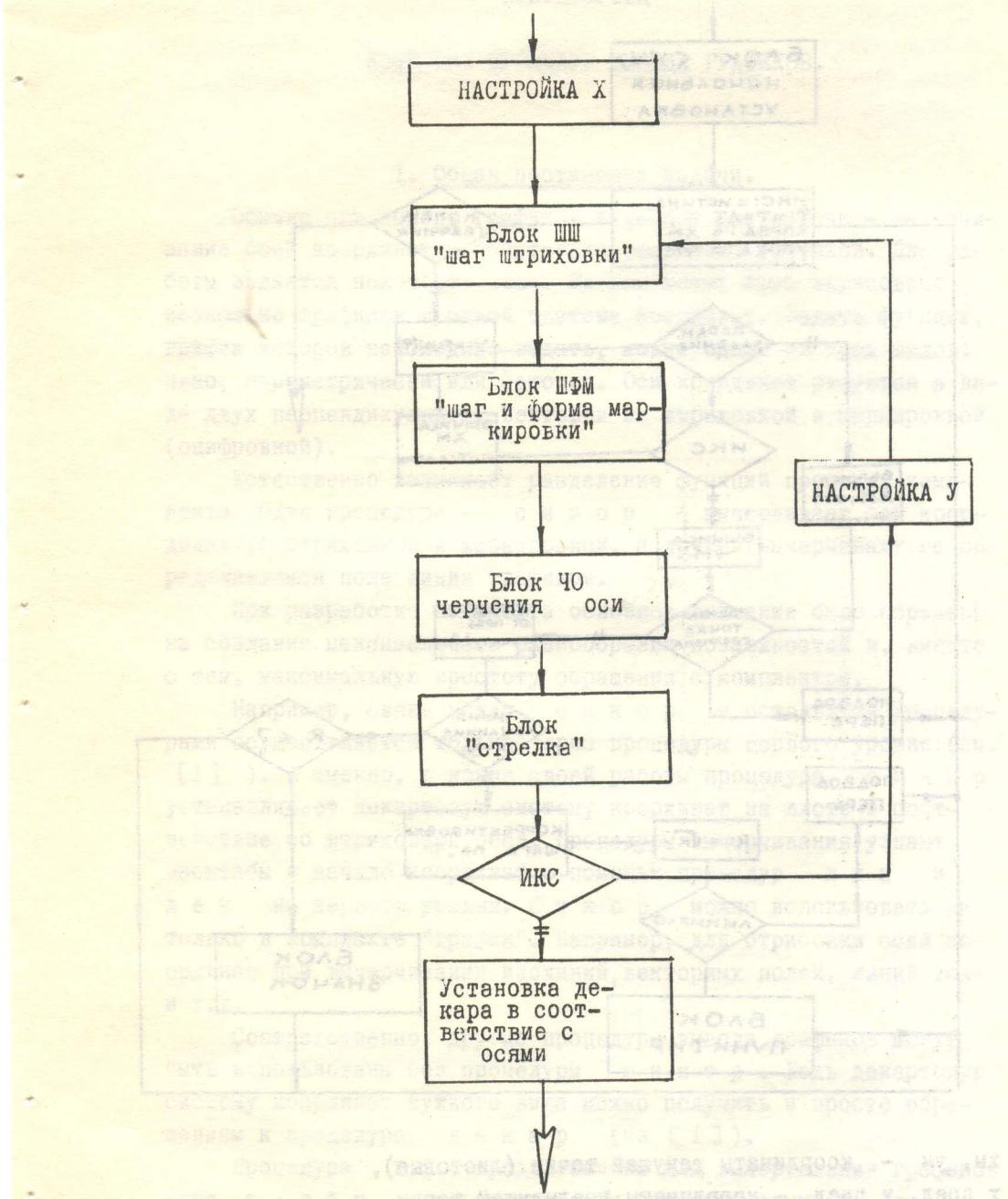
u_n , u_k - диапазон значений y ;

шаг x , шаг y - шаги штриховки по осям x и y .

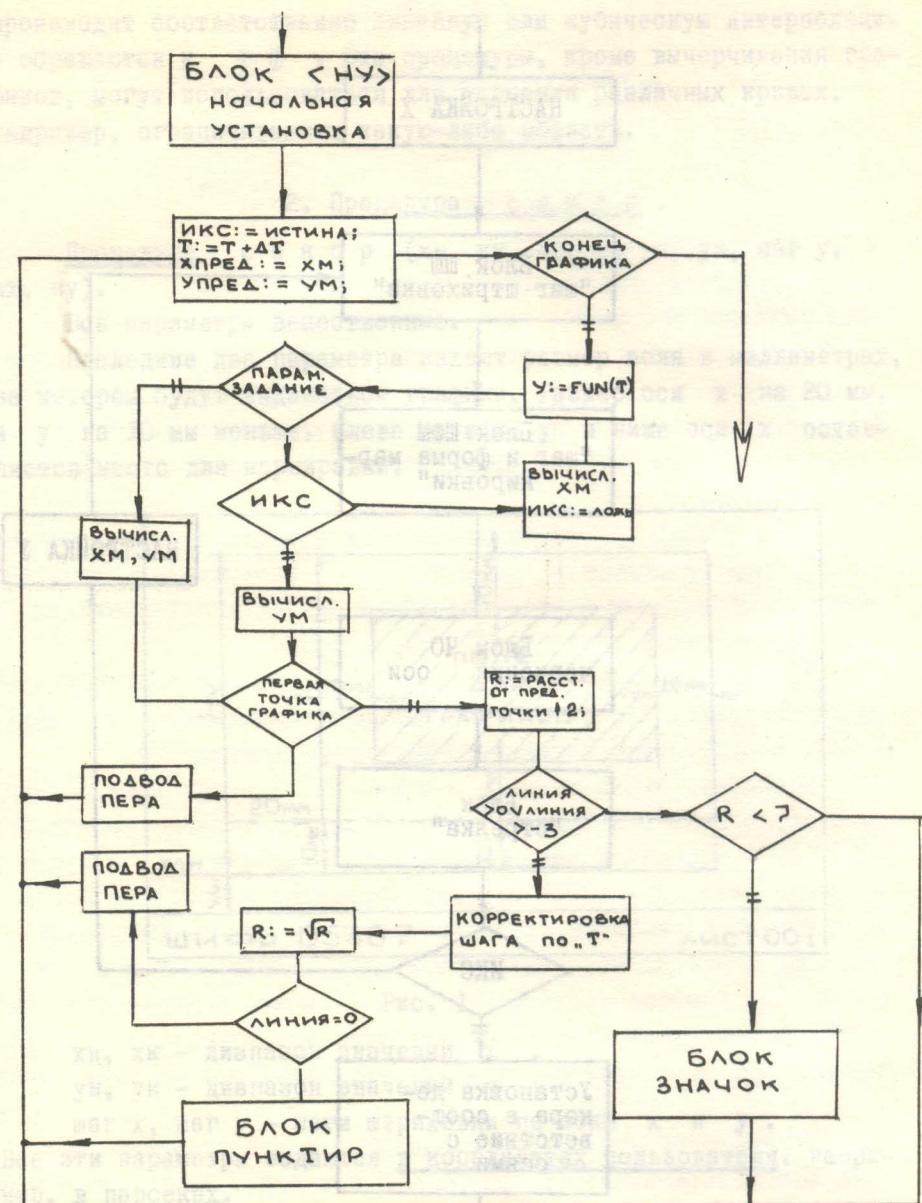
Все эти параметры задаются в координатах пользователя, например, в парсеках.

Имеется возможность предоставить выбор шага штриховки по любой оси самой процедуре с и к о р . Для этого соответствующий параметр шаг x или шаг y в обращении к с и к о р должен быть равен нулю. При таком задании процедура с и к о р вы-

Блок-схема с и к о р .



Блок-схема ГФ.



ХМ, УМ - координаты текущей точки (листовые),
Х пред, У пред - координаты предыдущей точки.

берет шаг штриховки так, чтобы расстояние между соседними штрихами было равно I или 2 миллиметрам. При этом x_n (ун) может несколько уменьшиться, а x_k (ук) - увеличиться по сравнению с заданными.

Перед обращением к сикор перо должно стоять в точке, которая будет левым нижним углом поля. Перо можно подвести обращением к процедуре тра.

Основная часть сикор работает дважды: чертятся оси "x" и "y". Перед черчением очередной оси работают блоки выбора шага штриховки и маркировки, а также устанавливается вид (формат), в котором будет выдаваться маркер.

Блок нахождения шага штриховки работает в случае задания шаг x (шаг y) равным нулю. Обозначим через ШШР значение шага штриховки в координатах пользователя, а через ШАМ - то же в листовых координатах. Сначала производится грубый выбор шага из расчета $I,5 \text{ мм} \leq \text{ШАМ} \leq 2 \text{ мм}$. Затем ШШР округляется до ближайшего числа, имеющего вид $y \times 10^x$, где $y = 1, 2$ или 5 . После этого ШАМ заменяется на свою целую часть - это будет I или 2 .

Следующий блок, анализируя первые 2-3 маркера, находит шаг и формат маркировки так, чтобы исключить возможность накладывания маркеров друг на друга. У маркеров незначащие нули справа подавляются.

Ясно, что при работе этих блоков может возникнуть противоречие между заданным процедуре масштабом (вытекающим из параметров x_n , x_k , un , uk , Hx , Hu) и желанием получить штриховку через целое число миллиметров, при ШШР, имеющем вид $y \times 10^x$. Поэтому производится необходимое расширение промежутка x_n , x_k (un , uk) при неизменном размере в миллиметрах (некоторое уменьшение масштаба по сравнению с заданным). Если значение этого масштаба существенно, то лучше пользоваться нормальным заданием шагов штриховки.

Блок черчения оси производит штриховку. При этом через каждые пять штрихов длиной по 2 мм ставится штрих 3 мм, а в случае, если штрих маркируется, то 4 мм.

3. Процедура ГФ.

Процедура ГФ (FUN , линия, $x_{нг}$, $унг$, $тм$);

Здесь **FUN** - функция, вычисляющая значение $y = f(x)$ в любой точке $x \in [x_{\text{НГ}}, x_{\text{КГ}}]$. Таким образом, задание функции для этой процедуры явное.

Линия - параметр, обеспечивающий выдачу графиков с помощью различных линий, пунктиров, значков и т.д. Если линия равна 0, то график выводится в виде сплошной линии, если она равна -1, -2, -3, то с помощью различных видов пунктиров.

Если $0 < \text{линия} < 205_{10}$, то график состоит из символов кодировки ГОСТ соответствующего номера. Если линия равна -4, то изображаются треугольники, -5 - квадратики, -6 - значки **Y**. Если линия > 205 мм или < -6 , то график выдается в виде сплошной линии.

Параметр "тм" - точность в миллиметрах, т.е. максимальный размер звена ломаной, аппроксимирующей график функции.

Следует отметить, что графики в виде пунктиров и сплошных линий являются точными (т.е. с точностью "тм"). Центры символов **Δ**, **□**, **Y** находятся на графике, центры остальных символов ($0 < \text{линия} < 205_{10}$) лежат на линии с ошибкой $\pm 0,8$ мм, т.е. их следует применять для изобразительных, а не измерительных целей.

ГФ приспособлена и для выдачи графиков параметрически заданных функций. Для этого необходимо, чтобы в обращении к ГФ значение параметра "линия" было больше 900, и процедура-функция (параметр "гф") была соответствующим образом организована. В этом случае от значения параметра "линия" вычитается 1000 и процедура переключается на параметрический режим - в каждой точке "т" производится два обращения к своему параметру **FUN**. Значение, полученное в первый раз, считается за $x(t)$, а во второй - за $y(t)$. Здесь $x_{\text{НГ}}$ и $x_{\text{КГ}}$ есть границы, в которых изменяется параметр "т". Естественно ограничение - в случае явного задания функции - линия < 900 .

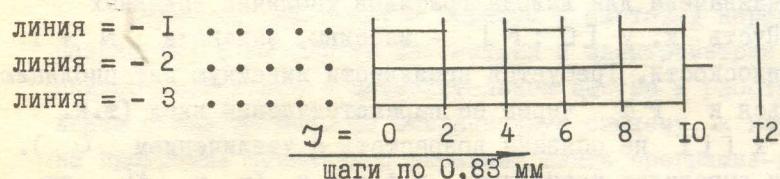
Общий цикл процедуры производит вычисление точек "x" и "y" графика при некотором шаге по "т".

При явном задании $t=x$ шаг по "т" выбирается произвольно, а затем корректируется до получения заданной точности (по параметру "тм").

Если линия равна 0, то график чертится сразу. Если же линия равна -1, -2, -3 или -4, -5, -6, то происходит переход на блоки черчения пунктиров и значков соответственно.

Блок "пунктир" накапливает точки линии до шага длиной 0,83 мм (остаток запоминается), и вычерчивает его (с положением пера, зависящим от параметра "линия"), затем управление передается обратно в общий цикл. Когда очередная точка графика поступает на вход блока "пунктир", то к длине "линия" между этой точкой и предыдущей добавляется хранимый остаток и т.д.

Пунктир рисуется участками по 10 мм (12 шагов по 0,83 мм). Следующий рисунок поясняет черчение различных видов пунктиров (темные участки указывают на опущенное перо).



По окончании участка все повторяется заново ($J :=$ шаг : = 0).

Блок "значок" также накапливает точки до длины 2,3 мм, а затем ставит значок. Если расстояние между соседними точками больше 2,3 мм, то оно делится на части по 2,3 мм и ставится нужное количество значков. После обработки очередной точки блок возвращает управление в основной цикл по "т".

4. Процедура г ф п .

Процедура осуществляет вывод графиков параметрически заданных функций, обращаясь к г ф через параметрический вход. Для иллюстрации параметрических обращений к г ф приведем полный текст г ф п .

```

процедура г ф п (фх, фу, лин, тн, тк, тм);
значение лин, тн, тк, тм;
целый лин; вещественный тн, тк, тм;
вещественный процедура фх, фу;
начало логический икс;
вещественный процедура эрзац (т);
значение т; вещественный т;
начало вещественный RESULT ;
RESULT : = если икс то фх (т) иначе фу (т);

```

икс := икс; эрзац := RESULT конец;
икс := истина;
г ф (эрзац, лин + 1000, тн, тк, тм);
конец;

Здесь фх, фу - процедуры-функции, вычисляющие х (т) и
у (т). Очевидно ограничение: лин > - 94.
тн, тк - границы изменения параметра "т".

5. Процедура г ф лин .

Предназначена для вывода графиков таблично заданных функций. Пусть х, у [0 : N] - массивы, задающие N + 1 точку на плоскости. Требуется произвести линейную интерполяцию и обратиться к г ф через ее параметрический вход (т.к. значения х [i] не обязаны возрастать с увеличением i). Эту задачу выполняет процедура г ф лин (x, y, N , линия, тм).

Параметры "линия" и "тм" имеют тот же смысл, что и в процедуре г ф . Ограничение такое же, как в г ф п - линия > - 94.

В начале работы вычисляется длина ломаной, соединяющей данные точки (x[i], y[i]). Затем идет обращение к г ф через параметрический вход, причем параметр "т" изменяется от 0 до вычисленной длины.

6. Процедура г ф кин .

Предназначена также для вывода графиков таблично заданных функций. Исходная информация - массивы x, y [-I : N +I].

Задача . Через точки (x[i], y[i]) i = 0,...,N провести гладкую кривую. Интерполяция кубическая. Обращение к г ф происходит через параметрический вход, т.е. функция $y = f(x)$ не обязана быть однозначной (точки (x[i], y[i]) могут располагаться в произвольном порядке). Значения x[i], y[i] при i = -I, N +I пользователь не задает (эти значения заполняются самой процедурой).

Описание :

Процедура г ф кин (x, y, N , т, линия, тм);

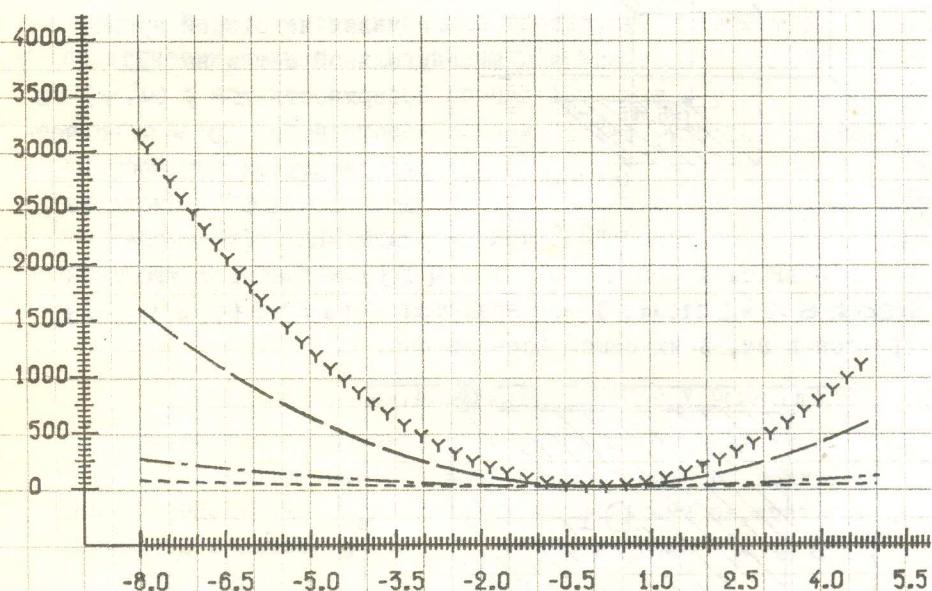
Параметры "x", "y", "N ", "линия", "тм" имеют тот же смысл, что и в процедуре г ф лин .

Массив T - рабочий массив процедуры. Он должен быть описан пользователем как массив $T [0 : 4, -1 : N + 1]$.

При работе процедуры на сетке $0, \dots, N$ вычисляются коэффициенты кубических многозвенников при $Z_i = x_i$ и $Z_i = y_i$. Коэффициенты $a_2^{(i)}$ для $f_i = x_i$ засыпаются в $T [4, 1]$, а для $f_i = y_i$ - в $T [2, 1]$.

Затем идет обращение к ГФ через параметрический вход. Параметр меняется от 0 до N . Процедура-функция вычисляет остальные коэффициенты $a_0^{(i)}$, $a_1^{(i)}$ и $a_3^{(i)}$ и затем значения функций $x(t)$ и $y(t)$. О кубических многозвенниках см. [2], а также статью "Процедуры вывода изолиний".

Пример использования процедур вывода графиков.



ЛИТЕРАТУРА

1. Математическое обеспечение графопостроителей (I уровень).
Под ред. Куртукова А.Я. ВЦ СО АН СССР, 1971.
2. Завьялов Ю.С. Интерполяция кубическими многозвенниками. В сб. "Вычислительные системы". Ин-т математики СО АН СССР, № 38.

Чтобы решить задачу вычерчивания изолиний, необходимо знать некоторые свойства функции $f(x, y)$.
 В. И. ДВОРЖЕЦ
 ПРОЦЕДУРЫ ВЫЧЕРЧИВАНИЯ ИЗОЛИНИЙ.

I. Общая постановка задачи.

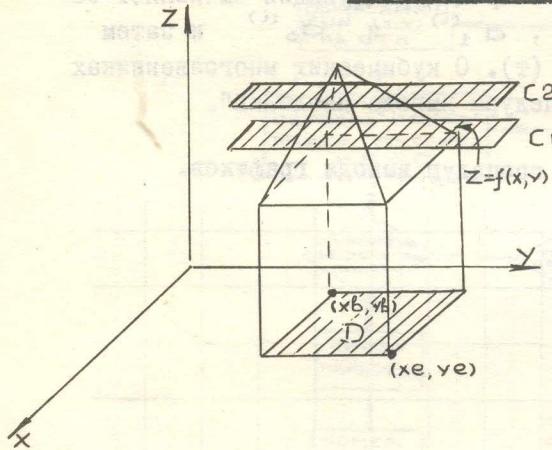


Рис. I

$D_1 = [a_1, b_1] \times [c_1, d_1]$. Параметры a_1, b_1, c_1, d_1 , измеряются в мм, а их смысл ясен из рис. 2.

Задана прямоугольная область $D = [x_b, x_e] \times [y_b, y_e]$ в R^2 и функция $f: D \rightarrow R$, заданная явно или табличей значений в узлах некоторой прямоугольной сетки на D . Поверхность $z = f(x, y)$ в пространстве $D \times R$ сеется семейством плоскостей $z = \text{const}$. Пусть на листе L (в смысле [I]) задан прямоугольник

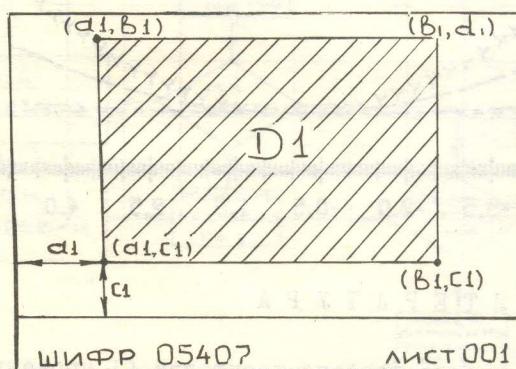


Рис. 2

На прямоугольнике D_1 требуется начертить изолинии функции

f заданных уровней $z = \text{const}$. В этой статье будут описаны 3 процедуры: линур, кинт и линт. Процедура линур вычерчивает изолинии явно заданных функций и является основной. Процедуры линт и кинт работают с таблично заданными функциями. Они производят линейную и кубическую интерполяции и обращаются к линур, которая и вычерчивает нужные уровни изолиний.

Следует отметить, что из всех программ П уровня комплект "изолиний" предъявляет наиболее высокие требования к памяти и времени счета, что потребовало особенно тщательной полировки алгоритмов. Поскольку процедуры были написаны на языках типа алгол, то данное требование повлекло за собой следующие особенности реализации:

1. Самый внутренний цикл процедуры линур написан практически целиком в машинных командах.

2. В главном массиве процедуры линур используется отдельно первый разряд каждой ячейки, что также потребовало работы в машинных командах.

3. Процедуры кинт и линт в варианте АЛЬФА оказалось необходимым реализовать с помощью автономно оттранслированных программ кинт и линт-генератора. (См. статью "Система для М-220" настоящего сборника).

2. Процедура линур.

Вычерчивает изолинии явно заданных функций, т.е. функций, для которых могут быть написаны вычисляющие процедуры-функции. Такая процедура-функция с двумя формальными параметрами должна вычислять значение $f(x, y)$ в любой точке $(x, y) \in D$.

Обращение.

линур ($H_x, H_y, F_{\text{fun}}, c, n, x_b, y_b, x_e, y_e$,
маркер);

Смысл параметров x_b, y_b, x_e, y_e ясен из пункта I - это координаты углов прямоугольника в системе координат пользователя (например, в километрах).

$H_x = b_1 - a_1 \quad \left. \right\}$ - размеры рисунка по "X" и "Y" в
 $H_y = d_1 - c_1 \quad \left. \right\}$ миллиметрах.

Этими параметрами еще не задано положение рисунка на листе L.

Для задания точки (d_1, c_1) используется процедура I уровня - декар.

Если $d_1 \neq 0$ или $c_1 \neq 0$, то перед обращением к линур следует поставить оператор декар (I, I, $d_1, c_1, 0, 0$). Ниже показано, как можно регулировать точность вычерчивания изолиний.

C - массив значений уровней с граничной парой $C[0:n]$.

Примечание: элементы массива C должны быть упорядочены:

$C[0] < C[1] < \dots < C[n]$.

Маркер - параметр, обеспечивающий маркировку изолиний разных уровней. Маркировка может производиться путем постановки на линии либо номера уровня (начиная с нуля) при маркер = 10, либо значения $C[i]$ уровня i с заданным числом цифр в мантиссе или дробной части при dbs (маркер) ≤ 4 . Во втором случае маркер задает вид выдачи $C[i]$ так же, как в процедуре число (см. [I]). Если маркер $\neq 10$ и dbs (маркер) ≥ 5 , то маркировка не производится.

Опишем алгоритм процедуры, который не зависит от конкретных языков и машин.

Рисунок разбивается на стандартные квадраты (СТК) размером 30 x 30 МК. Размер единицы МК задается процедурой декар (см. ниже описание регулировки точности). Сначала отрисовываются все уровни изолиний на первом СТК, затем на втором и т.д. Внутри каждого СТК черчение ведется по уровням, начиная от 0 до n . Для вычерчивания уровней на СТК наводится сетка малых квадратов (МК). В процедуре линур описан массив КЕКС с граничными парами $[0:30, 0:30]$, в котором хранятся значения функции в узлах сетки МК. Вычисление этих значений происходит до вычерчивания уровней на СТК. Младший (первый) разряд каждой ячейки этого массива используется для пометки тех МК, через которые нужно провести рассматриваемый уровень изолиний. Для ячеек массива КЕКС, соответствующих этим квадратам, первый разряд полагается равным 1, а для остальных ячеек - 0. Ясно, что изменение первого разряда незначительно изменит значение функции и не окажет существенного влияния на картину уровня. Определение прохождения уровня через данный МК

производится путем сравнения знаков разности КЕКС $[f, k] - [c_i]$ на концах сторон МК. (Здесь c_i - значение уровня). Если хотя бы на одной из сторон эта разность меняет знак, то через рассматриваемый МК проходит уровень c_{i+1} .

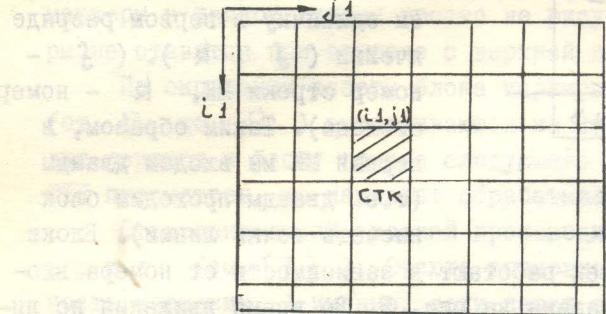


Рис. 3

После заполнения первых разрядов ячеек массива КЕКС для данного уровня производится проверка, была ли занесена в них хотя бы одна единица (это означает, что через этот МК проходит уровень). Если этого не произошло, то данный уровень не пересекает поверхность на рассматриваемом СТК. При этом, если на текущем СТК уже отрисовывались какие-либо уровни, то можно переходить к следующему СТК, иначе, к следующему уровню. Действительно, так как уровни упорядочены, уровень k пересекает поверхность, а уровень $k+1$ не пересекает, то уровни $k+x$ ($x > 1$) не могут пересекать поверхность из однозначности функции f . Например, на рис. 4 уровни, следующие за третьим, могут не анализироваться.

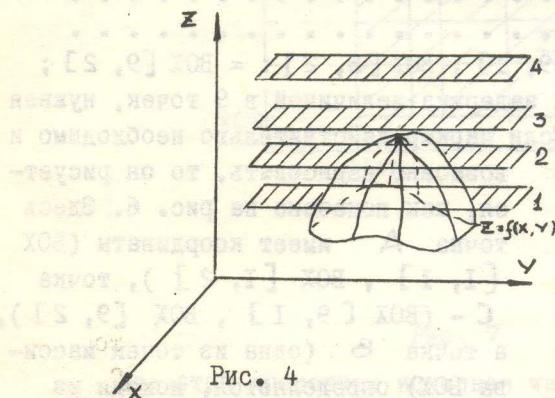


Рис. 4

Пусть теперь уровень пересекает поверхность на данном СТК. В этом случае начинается поиск единицы в первых разрядах ячеек массива КЕКС.

Просмотр идет построчно. Пусть найдена единица в ячейке номера (f_0, k_0). Насчитываем первую точку линии, про-

изводя линейную интерполяцию внутри МК. Насчитав первую точку, дальнее действуем стандартно.

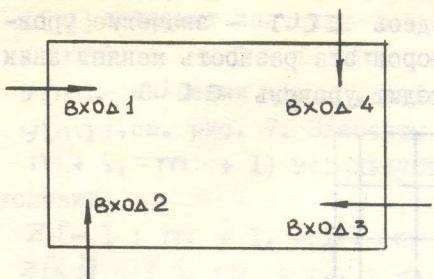


Рис. 5

черчения и насчета точек работают в зависимости от номера входа в МК. Нумерация показана на рис. 5. Во время движения по линии до входа в следующий квадрат проверяется значение первого разряда соответствующей ячейки КЕКС $[f, k]$. Если он равен нулю (линия замкнулась или вышла за границу СТК), то производится переход на блок маркировки.

При движении по линии заполняется буферный массив BOX [I:9, I:2] размером—9 точек линии. При заполненности BOX происходит вычерчивание первой его точки и сдвиг информации в этом массиве по схеме:

BOX [I, I] := BOX [2, I]; BOX [I, 2] := BOX [2, 2];

.....

.....

BOX [8, I] := BOX [9, I]; BOX [8, 2] := BOX [9, 2];

Таким образом, получается задержка величиной в 9 точек, нужная для постановки маркера. Если маркер действительно необходимо и возможно нарисовать, то он рисуется, как показано на рис. 6. Здесь точка A имеет координаты (BOX [I, I], BOX [I, 2]), точка C - (BOX [9, I], BOX [9, 2]), а точка B (одна из точек массива BOX) определяется, исходя из размеров маркера. Стрелкой показано направление вычерчивания линии.

Блок маркировки принимает меры против постановки маркеров близко друг от друга. Предотвратить наложение маркеров на линии уровней трудно и, видимо, не нужно. При большой густоте

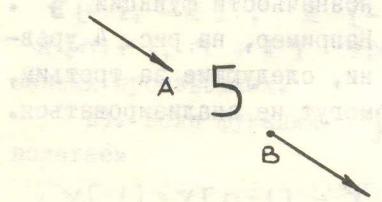


Рис. 6

линий лучше применять маркировку номером уровня (маркер = 10). Для предотвращения накладывания маркеров их координаты хранятся в специальном массиве. При переходе к следующей полосе СТК этот массив обновляется. При переполнении массива следующие маркеры в данной полосе просто не рисуются. Кроме того, маркеры не ставятся при встрече с верхней и левой границами СТК.

По окончании работы блока маркировки оставшиеся точки (от B до C) выталкиваются из накопителя BOX, и происходит переход к блоку поиска следующего куска линии. Если весь СТК просмотрен, то начинает обрабатываться следующий уровень.

Отрисовка линий уровней производится через процедуру т р а д (см. [1]). Отсюда возможность регулировать точность черчения. А именно, если перед обращением к т р а д поставить оператор д е к а р ($2, 2, \text{c1}, \text{c1}, 0, 0$) и задать H_x и H_y в два раза меньше, чем необходимо, то размер МК будет в два раза больше стандартного. Это уменьшит время счета почти в четыре раза. Стандартный размер МК равен одному миллиметру.

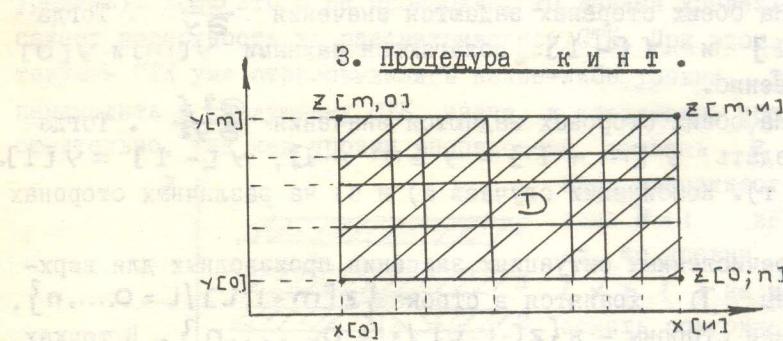


Рис. 7

Для этой процедуры исходная информация включает таблицу значений функции в узлах прямоугольной (не обязательно равномерной) сетки на области D .

Обращение.

`кинт (x, y, z, n, m, Hx, Hy, c, cn, маркер);`

массив $x[-I : n + I]$ в своих элементах $x[0], \dots, x[n]$ содержит значения x_i разбиения отрезка ($x[0], x[n]$), а $y[-I : m + I]$ в элементах $y[0], \dots, y[m]$ — значения y_j , задающие разбиение отрезка ($y[0], y[m]$), см. рис. 7. Элементы с крайними номерами ($-I, n + I, m + I$) используются для указания типа граничных условий.

$z[-I : m + I, -I : n + I]$ — массив, ячейки $z[i, j]$ ($i = 0, \dots, n; j = 0, \dots, m$) которого задают значения функции в узлах сетки по формуле $z[i, j] = f(x[i], y[j])$. Элементы краевых строк и столбцов массива z используются для задания граничных условий.

Интерполяция ведется с помощью аппарата сплайн-функций (бикубических многозвенников). Математическая сторона дела описана в работе [3].

Способы задания граничных условий по "x" и по "y" аналогичны, поэтому мы приведем их только в случае "y" (верхняя и нижняя стороны D).

а). На обеих сторонах задаются значения $\frac{\partial f}{\partial y}$. Тогда $y[m + I]$ и $y[-I]$ полагаются равными $y[m]$ и $y[0]$ соответственно.

б). На обеих сторонах задаются значения $\frac{\partial^2 f}{\partial y^2}$. Тогда следует задать $y[m + I] = y[m - I]$, $y[-I] = y[I]$.

в) и г). Комбинации случаев а) и б) на различных сторонах D .

В перечисленных ситуациях значения производных для верхней стороны D хранятся в строке $\{z[m+1, i] / i = 0, \dots, n\}$, а для нижней стороны — в $\{z[-1, i] / i = 0, \dots, n\}$. В точках $z[-I, -I]$, $z[-I, n + I]$, $z[m + I, -I]$, $z[m + I, n + I]$ хранятся значения соответствующих смешанных производных.

д). Если функция f периодична с периодом Y , то полагаем

$y[-I] = y[n - I] - Y$ и $y[m + I] = y[1] + Y$,
причем $z[m + I, i] = z[I, i]; z[-I, i] = z[m - I, i]$ для $i = 0, \dots, n$.

Если функция оказывается периодичной и по x , то условия в угловых точках не задаются. В противном случае в углах

задаются только производные $\frac{\partial f}{\partial x}$ или $\frac{\partial^2 f}{\partial x^2}$.

Пользователь, не имеющий никаких сведений относительно граничных условий, может задать значения вторых производных на границе D нулями. Это соответствует продолжению функции за границу линейно.

Параметры H_x , H_y , c , c_n , маркер имеют тот же смысл, что и в процедуре `линуr`.

Процедура `кинt` состоит из обращения к процедуре `линуr` и описаний процедур `кит` и `SIM`. Процедура `кит` осуществляет одномерную кубическую интерполяцию, алгоритм которой был впервые реализован Михалевичем и Омельченко в стандартной процедуре `CUM` ([4]). По сравнению с `CUM`, в процедуре `кит` количество рабочих массивов сокращено до минимума — пяти, против восьми в `CUM`, без особого изменения алгоритма.

Процедура `SIM` осуществляет бикубическую интерполяцию по заданной точке (x_0, y_0) и массивам x , y , z . Алгоритм процедуры сильно отличается от алгоритма процедуры `BICUM` Михалевича и Омельченко (см. [4]).

Отличия связаны с невозможностью хранения огромных матриц `BICUM` в памяти М-220 и даже БЭСМ-6. Это потребовало в ряде случаев повторного просчета коэффициентов бикубического многозвенника. Кроме того, использовано знание способа просмотра СТК (см. пункт 2) в процедуре `линуr`, которая обращается к `SIM` (так как `SIM` в обращении из `кинt` является фактическим параметром `линуr`). Из описания алгоритма построения бикубического многозвенника в [4] видно, что для вычисления коэффициентов на некоторую клетку интерполяционной сетки достаточно сделать n обращений к `кит` до работы `линуr`, 4 обращения в каждой новой строке и одно обращение при смене столбца внутри строки. При этом хранится одна матрица $n \times m$, одна $n \times 12$ и $5 \times k$ — матрицы `кита` ($k = \max(n, m)$).

Матрица $n \times m$ получается при расчете всех коэффициентов $d_{0e}^{(i,j)}$, матрица $n \times 12$ — коэффициенты на одну строку вида $d_{3e}^{(i,j)}, d_{2e}^{(i,j)} \text{ и } d_{1e}^{(i,j)}$ для i и $i+1$, где $e = 0, 2$.

4. Процедура линт.

Для этой процедуры исходная информация также включает таблицу значений функции f в узлах сетки, однако граничные условия не задаются. Массив Z для этой процедуры имеет граничные пары $[0 : m, 0 : n]$, а массивы x и y - $[0 : n]$ и $[0 : m]$. Как и в пункте 3, $z[i,j] = f(x[i], y[j])$ для $i = 0, \dots, n; j = 0, \dots, m$.

Смысл остальных параметров такой же, что и в процедуре линур.

Процедура линт состоит из описания процедуры линейной интерполяции и обращения к линур с этой процедурой.

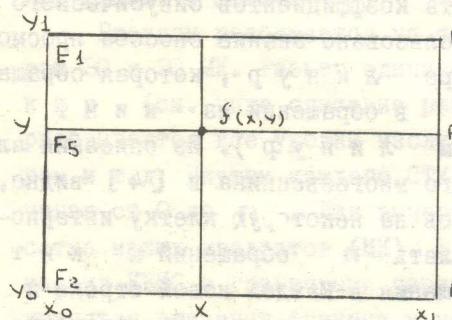
Линейная интерполяция производится по формуле:

$$f(x,y) = F_5 + (F_6 - F_5) \times (x - x_0) / (x_1 - x_0),$$

где

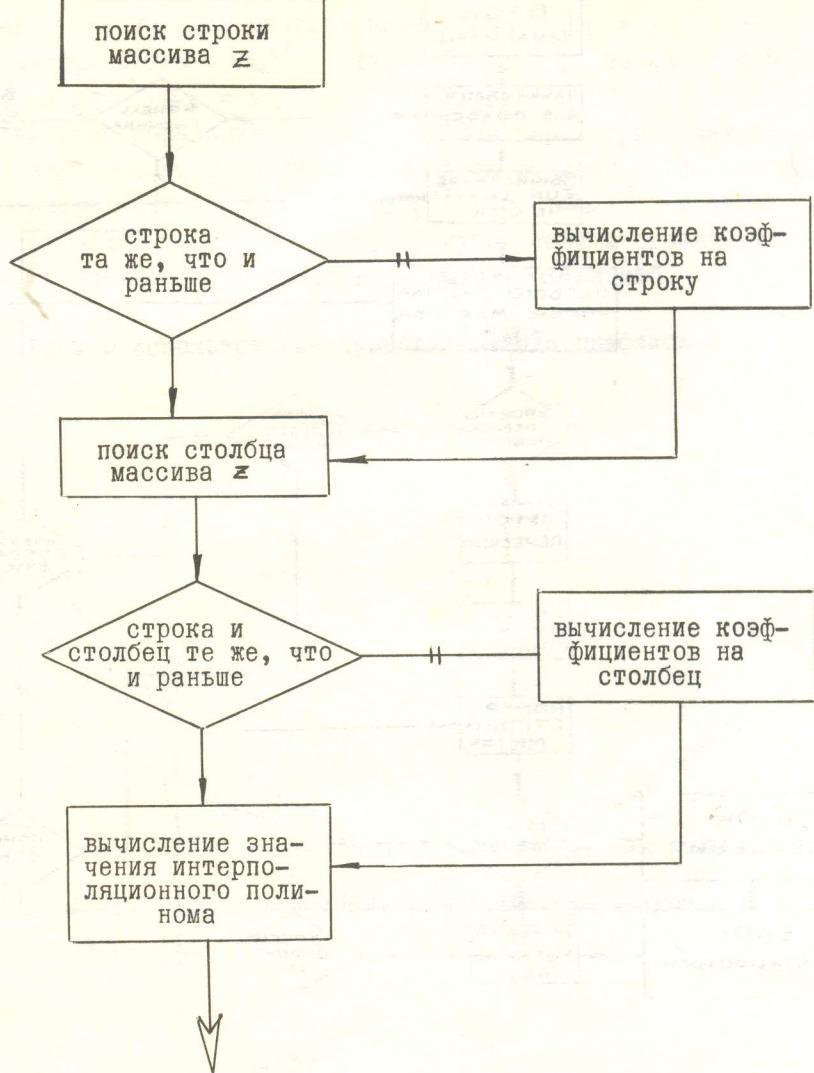
$$F_5 = F_2 + (F_1 - F_2) \times (y - y_0) / (y_1 - y_0)$$

$$F_6 = F_3 + (F_4 - F_3) \times (y - y_0) / (y_1 - y_0).$$

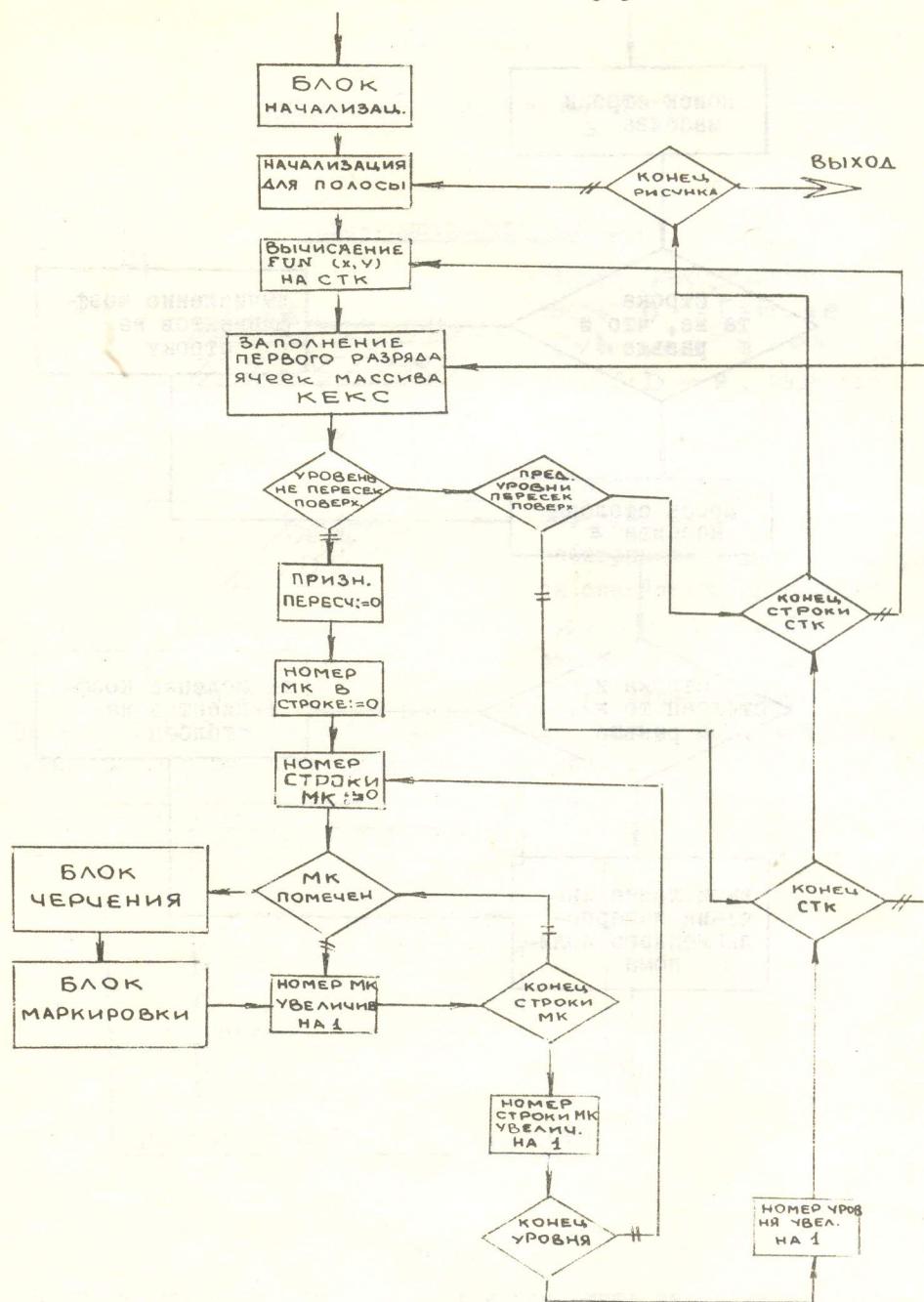


Здесь F_1, F_2, F_3, F_4 - значения функции в узлах нужного прямоугольника сетки; y_0, y_1, x_0, x_1 - координаты сторон прямоугольника, а x, y - точка $x \in [x_0, x_1], y \in [y_0, y_1]$.

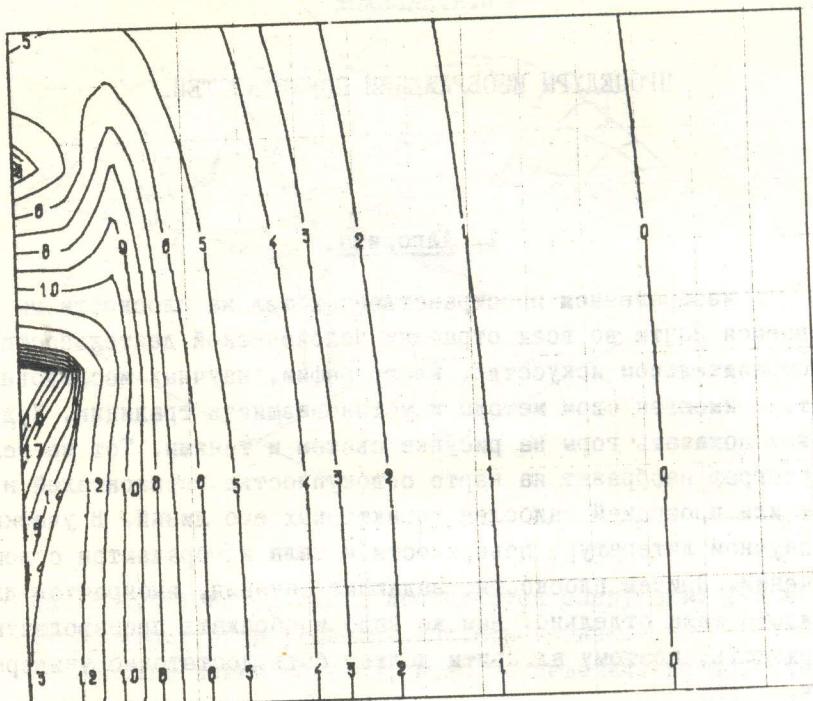
Блок-схема SIM.



Блок-схема линур.



Пример использования процедур вывода изолиний.



ЛИТЕРАТУРА

1. Математическое обеспечение графопостроителей (I уровень).
Под ред. Куртукова А.Я. ВЦ СО АН СССР, 1971.
2. Автоматическое построение изолиний. САСМ № 7, 1969.
3. Завьялов Ю.С. Интерполяция кубическими многозвенниками.
Интерполяция бикубическими многозвенниками. В сб. "Вычислительные системы". Институт математики СО АН СССР, № 38.
4. Михалевич Ю.И., Омельченко О.Н. Процедуры кусочно-полиномиальной интерполяции функций одного и двух переменных. В сб. "Стандартные программы и процедуры". Вып. П, Новосибирск, 1970.

ПРОЦЕДУРЫ ИЗОБРАЖЕНИЙ ПОВЕРХНОСТЕЙ.

1. Алгоритм.

С изображением пространственных тел на плоскости мы сталкиваемся почти во всех отраслях человеческой деятельности. В изобразительном искусстве, картографии, научных исследованиях и т.д. имеются свои методы и установившиеся традиции. Художник может показать горы на рисунке светом и тенями. Тот же рельеф картограф изобразит на карте совокупностью горизонталей и высот или проекцией наиболее характерных его линий. В учебниках и научной литературе поверхности и тела изображаются с помощью сечений. Причем плоскости, задающие сечения, выбираются для каждого тела отдельно. Нам же надо изображать произвольную поверхность, поэтому алгоритм должен быть достаточно универсальным.

Пусть поверхность определяется однозначной и непрерывной функцией $Z = Z(x, y)$, где x и y изменяются в прямоугольнике

$$\Delta = \{x, y; \begin{array}{l} x_n \leq x \leq x_k, \\ y_n \leq y \leq y_k \end{array}\}$$

Произведем сечения поверхности плоскостями, параллельными координатным x и y . В пересечении их с поверхностью получается как бы каркас из кривых, на который она натянута. Его то мы и будем проектировать на плоскость рисунка.

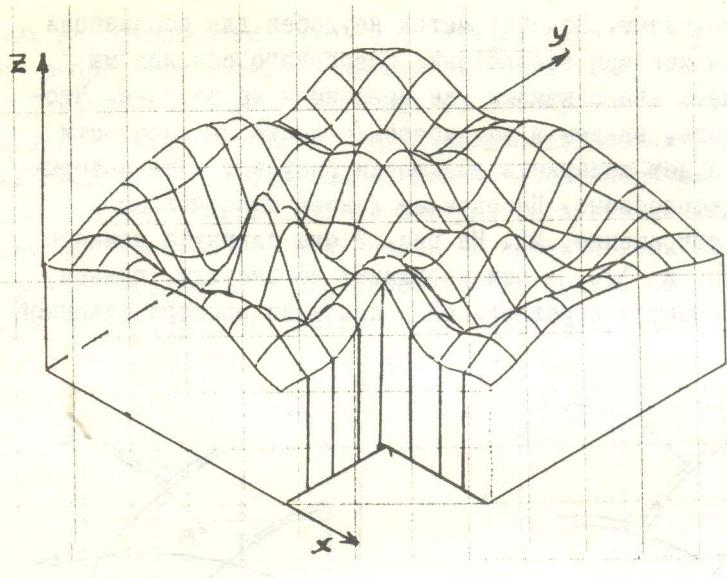


Рис. I

Наглядность изображения достигается следующими условиями:

1. Проекция прямоугольная изометрическая.
2. Участки линий каркаса, которые невидимы на поверхности, на рисунке не изображаются.
3. Поверхность представляется в виде крыши прямоугольного непрозрачного бруса (см. рис. I).

Рассмотрим механизм уборки невидимых линий. Пусть сечения поверхности выглядят так, как показано на рисунке 2. Для большей наглядности они представлены как кривые, ограничивающие

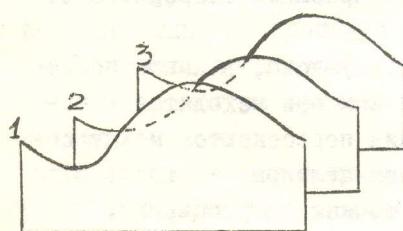


Рис. 2

непрозрачные пластины. Пластина 1 не закрыта никакой другой, поэтому проекция первого сечения изображается полностью. У пластины 2 часть АВ ограничивающего сечение невидима из-за первой пластины. Следовательно, второе сечение изображается не полностью. Аналогич-

ную операцию проделываем и для третьего сечения. Правда, для него необходимо определять превышение не только над первым, но

и над вторым сечением. Но этот метод неудобен для реализации его на ЭВМ, так как при изображении очередного сечения мы должны сравнивать его с каждым уже имеющимся на рисунке. Чтобы избежать этого, введем в рассмотрение линию на плоскости проекции. Она будет содержать все точки рисунка, выше которых уже ничего не нарисовано. Назовем ее линией максимального рельефа, или, сокращенно, МР. На рис. 3 она сдвинута немного вверх, где U и V - оси координат на рисунке, причем ось V выбирается параллельной к проекции пространственной оси Z .

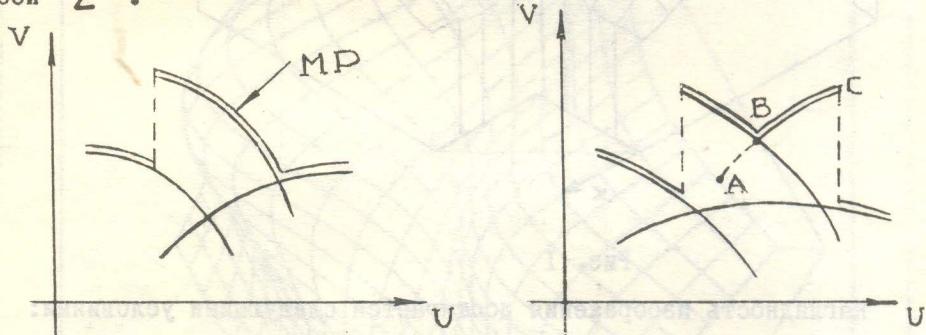


Рис. 3

Теперь на каждом шаге алгоритма будем считать видимыми участки проекций сечений, которые расположены выше линии максимального рельефа. После изображения видимой части сечения на рисунке определяем МР и переходим к следующему шагу. Первоначально МР задается уравнением $V = M$, где M достаточно большое отрицательное число. На рис. 3 АС - проекция очередного сечения и АВ - ее невидимая часть.

В рассмотренном выше случае (рис. 2) ясно, в какой последовательности изображать сечения, так как они находятся в параллельных плоскостях. У нас же сечения пересекаются между собой. Поэтому надо изображать не сечения целиком, а отдельно каждый участок между двумя соседними точками пересечений.

Сечениям в области D будет соответствовать прямоугольная сетка, а каждому элементарному участку - одно единственное ребро этой сетки. Тогда последовательность изображений можно задать следующим образом. Рассмотрим совокупность ребер сетки Q и соответствующую ей совокупность отрезков

кривых на поверхности R . На каждом шаге из Q выбираем такое ребро, что отрезок, соответствующий этому ребру, не может быть "закрыт" никаким другим из совокупности R . При этом необходимо учитывать, что поверхность произвольная. Из обеих совокупностей удаляем соответственно выбранное ребро и участок кривой, видимую часть которой изображаем. Так делаем до тех пор, пока обе совокупности не исчерпаются. На рисунках 4.1 и 4.2 приводятся два способа обхода сетки. Стрелка под рисунком указывает направление проектирования.

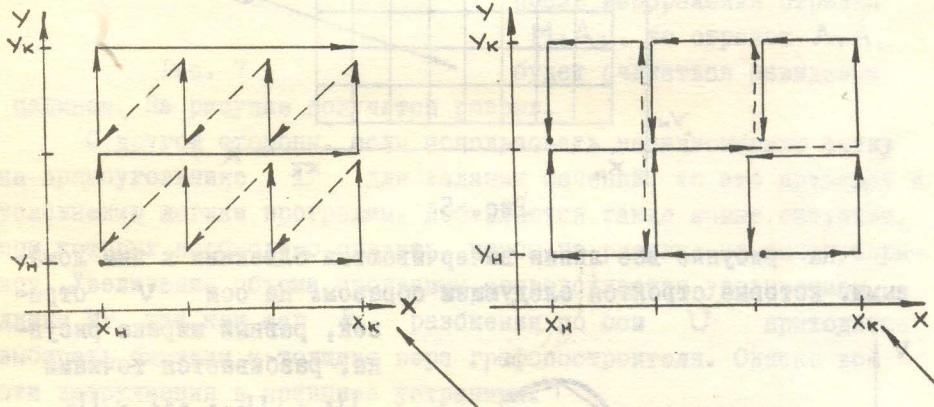


Рис. 4.1

Рис. 4.2

2. Конкретная реализация.

Изложенный алгоритм был реализован автором в виде комплекта процедур на языках АЛГОЛ и АЛЬФА в Вычислительном центре СО АН СССР. Одна из процедур изображает поверхности, заданные во всей области процедурой-функцией, другая - заданные таблично. Вторая процедура применяет кубическую интерполяцию сплайн-функциями.

Одним из параметров процедуры является желательный размер рисунка по горизонтали в миллиметрах. Поэтому процедура автоматически деформирует поверхность одинаково по осям x и y до получения нужного размера. По оси z искажений нет, и пользователю необходимо учитывать, что по вертикали рисунок не контролируется.

Секущие плоскости выбираются так, чтобы прямоугольная сеть

сетка на области D была равномерной по обеим осям. Но выбор подходящего шага разбиения не всегда возможен, поэтому из прямоугольника D отбрасывается полоса, параллельная оси x и по ширине меньшая, чем шаг разбиения (см. рис. 5). В конце работы программы подрисовывает снизу к поверхности бруса, который как бы вырезается из пространства.

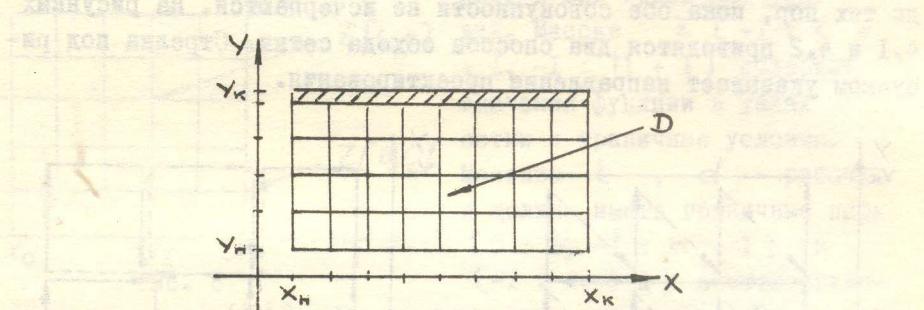


Рис. 5

На рисунке все линии вычертываются близкими к ним ломаными, которые строятся следующим образом. На оси V отре-

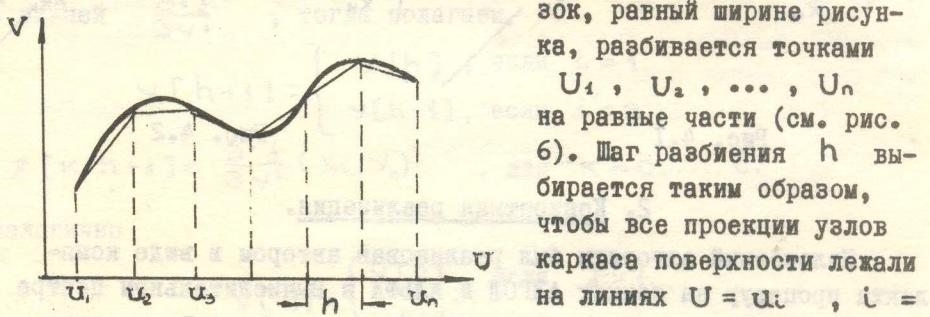


Рис. 6

зок, равный ширине рисунка, разбивается точками U_1, U_2, \dots, U_n на равные части (см. рис. 6). Шаг разбиения h выбирается таким образом, чтобы все проекции узлов каркаса поверхности лежали на линиях $U = u_i$, $i = 1, 2, \dots, n$. За вершины ломаной, приближающей кривую, возьмем точки пересечения этой кривой с прямыми $U = u_i$. Точно так же зададим и линии максимального рельефа. В памяти машины она хранится в виде массива ординат для точек u_i . Остальные ломаные в памяти не хранятся, т.к. достаточно проектировать каждое их звено отдельно, вычисляя координаты концов непосредственно перед изображением. Программа для каждого звена определяет видимую его часть, изображает ее и правит массив MP в точках, соответствующих концам звена, если это необходимо. Правка MP происходит не сразу, а только после изображения следующего звена ломаной.

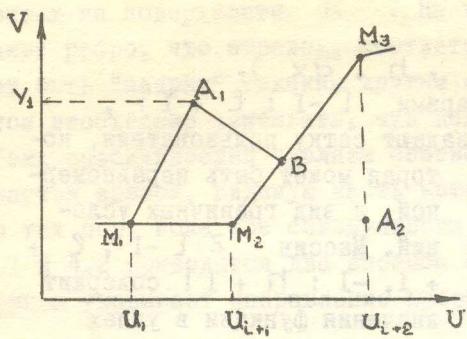


Рис. 7
целиком. На рисунке получится разрыв.

С другой стороны, если использовать неравномерную сетку на прямоугольнике D для задания сечений, то это приведет к усложнению логики программы. Добавляются также новые ситуации, при которых необходимо следить, чтобы на рисунке не было разрывов. Увеличение объема программы сопровождается увеличением длины MP, так как шаг h разбиения по оси U приходится выбирать близким к толщине пера графопостроителя. Однако все эти затруднения в принципе устранимы.

2.1. Процедура `приз` применяется для изображения поверхностей, заданных аналитически.

Обращение.

`приз (xн, xк, ун, ук, сx, f)`.

Первые четыре параметра задают прямоугольник $D = [xн, xк] \times [ун, ук]$ и выражаются в координатах пользователя. f — процедура от двух параметров, которая по $(x, y) \in D$ вычисляет z . Параметр `сx` — желательный размер рисунка по горизонтали, выраженный в миллиметрах.

2.2. Процедура `рифт` применяется для изображения поверхностей, заданных таблично. Информация о поверхности состоит из массивов точек разбиения осей x и y , которые задают сетку пользователя на D , и массива значений функции в узлах сетки. Так как процедура в процессе работы использует интерполяцию сплайн-функциями, то необходимо задать граничные условия.

Поясним это на рис. 7. Отрисовав отрезок $M_1 A_1$, запоминаем значение y_1 . В массиве MP изменений не делаем. И лишь после изображения

$A_2 B$ — видимой части отрезка $A_1 A_2$, значение y_1 засылаем в MP $[i + 1]$. Если данную засылку сделать сразу после изображения отрезка $M_1 A_1$, то отрезок $A_1 A_2$ будет считаться невидимым

Обращение.

рифт $(x, y, z, t, d, l, h, \alpha_x)$.

Массивы x и y с граничными парами $[-I : l + I]$,

$[-I : h + I]$ соответственно задают сетку пользователя, ко-

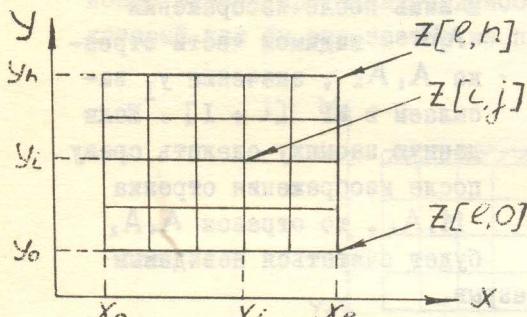


Рис. 8

торая может быть неравномерной, и вид граничных условий. Массив $z [-I : l + I, -I : h + I]$ содержит значения функции в узлах сетки и граничные условия. Массивы t, d - рабочие и должны иметь граничные пары $[0 : 3, -I : k + I]$ и $[-I : k + I]$ соответствен-

но, где $k = \max(l, h)$. На сторонах прямоугольника D могут быть заданы значения первых или вторых производных, либо периодичность функции.

Пусть на верхней границе задаются значения $\frac{\partial^i f}{\partial x^i}$, а на нижней $\frac{\partial^j f}{\partial y^j}$, тогда полагаем

$$y[h+1] = \begin{cases} y[h], & \text{если } i=1 \\ y[h-1], & \text{если } i=2 \end{cases}$$

а $z[k, h+1] = \frac{\partial^i f}{\partial y^i}(x_i, y_h)$, для $k=0, \dots, l$.

Аналогично

$$y[-1] = \begin{cases} y[0], & \text{если } j=1 \\ y[1], & \text{если } j=2 \end{cases}$$

$z[k, -1] = \frac{\partial^j f}{\partial x^j}(x_k, y_0)$, для $k=0, 1, \dots, l$.

Точно так же задаются граничные условия на левой и правой сторонах прямоугольника D . В угловых элементах матрицы

z : $z[-I, -I]$, $z[-I, h+I]$, $z[l+I, -I]$, $z[l+I, h+I]$ хранятся значения соответствующих смешанных производных. Если функция периодична с периодом Y , то полагаем

$$y[-1] = y[h-1] - Y$$

$$y[h+1] = y[1] + Y,$$

причем

$$z[\ell+1, i] = z[1, i]$$

$$z[-1, i] = z[\ell-1, i],$$

для

$$i = 0, \dots, h$$

Если функция f оказывается периодичной и по x , то условия в угловых точках не задаются. В противном случае в углах задаются только производные $\frac{\partial f}{\partial x}$ или $\frac{\partial^2 f}{\partial x^2}$.

Математическая сторона дела изложена в [2].

2.3. Связь между процедурами.

Было бы невыгодно иметь две процедуры изображения, многие функции которых совпадают. Поэтому изображение поверхности осуществляет процедура `приз`, а `рифт` занимается только интерполяцией предложенной ей таблицы. В `рифте` описывается процедура-функция от двух параметров — $f(x, y)$, которая при вычислении значения функции использует побочный эффект.

Обращение к `призу` в `рифте` выглядит следующим образом:

`приз (x[0], x[\ell], y[0], y[h], dx, f).`
Ввиду того, что для изображения поверхности нужны значения функции лишь на сечениях, в `рифте` используется одномерная интерполяция. В каждый момент в памяти хранятся коэффициенты кубического многозвенника (см. [2]) только для изображаемого сечения. Для этого `приз` каждый раз при переходе на новое сечение сообщает об этом процедуре $f(x, y)$ специальным обращением. Такое обращение `приз` делает только в том случае, если он работает с `рифтом`. На рис. 9 приводится рисунок, полученный с помощью процедуры `рифт`.

ИСТ 004

ШИФР 54287

чертежи 1

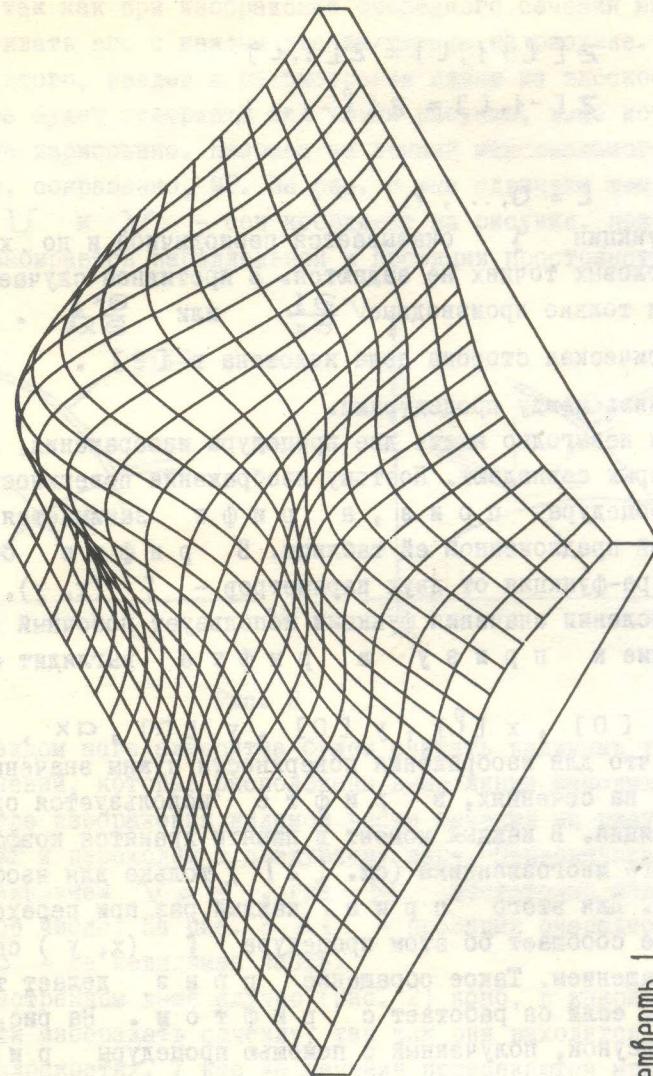


Рис. 9. Пример выдачи программы.

ЛИТЕРАТУРА

1. Математическое обеспечение графопостроителей (I уровень).

Под ред. Куртукова А.Я. ВЦ СО АН СССР, 1971.

2. Завьялов Ю.С. Интерполирование кубическими многозвенниками. В сб. "Вычислительные системы". Институт математики СО АН СССР, № 38.

3. Михалевич Ю.И., Омельченко О.Н. Процедуры кусочно-полиномиальной интерполяции функций одной и двух переменных. В сб. "Стандартные программы и процедуры".

Вып. П, Новосибирск, 1970.

Задача 1. Рассмотрим задачу о построении кубической кривой, проходящей через заданные точки x_1, x_2, x_3, x_4 . Для этого воспользуемся процедурой $\text{INTERP}(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$, которая возвращает коэффициенты кубической кривой в виде

$y = a_3x^3 + a_2x^2 + a_1x + a_0$, где a_3, a_2, a_1, a_0 — коэффициенты кубической кривой. Для этого воспользуемся процедурой $\text{INTERP}(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$, которая возвращает коэффициенты кубической кривой в виде

$y = a_3x^3 + a_2x^2 + a_1x + a_0$, где a_3, a_2, a_1, a_0 — коэффициенты кубической кривой. Для этого воспользуемся процедурой $\text{INTERP}(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$, которая возвращает коэффициенты кубической кривой в виде

$y = a_3x^3 + a_2x^2 + a_1x + a_0$, где a_3, a_2, a_1, a_0 — коэффициенты кубической кривой. Для этого воспользуемся процедурой $\text{INTERP}(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$, которая возвращает коэффициенты кубической кривой в виде

$y = a_3x^3 + a_2x^2 + a_1x + a_0$, где a_3, a_2, a_1, a_0 — коэффициенты кубической кривой. Для этого воспользуемся процедурой $\text{INTERP}(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$, которая возвращает коэффициенты кубической кривой в виде

ПРОЦЕДУРЫ ВЫВОДА ВЕКТОРНЫХ ПОЛЕЙ.

(анесят 1) ПРОЦЕДУРЫ ВЫВОДА ВЕКТОРНЫХ ПОЛЕЙ
предназначены для вывода векторных полей на экран монитора.

• ПРОЦЕДУРЫ ВЫВОДА ВЕКТОРНЫХ ПОЛЕЙ СОСТАВЛЕНЫ ИЗ ОДНОГО УЧЕБНИКА И ДОДАЧИ
• ПРОЦЕДУРЫ ВЫВОДА ВЕКТОРНЫХ ПОЛЕЙ СОСТАВЛЕНЫ ИЗ ОДНОГО УЧЕБНИКА И ДОДАЧИ
• ПРОЦЕДУРЫ ВЫВОДА ВЕКТОРНЫХ ПОЛЕЙ СОСТАВЛЕНЫ ИЗ ОДНОГО УЧЕБНИКА И ДОДАЧИ

I. Общая постановка задачи.

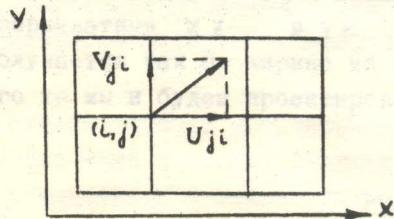
На плоскости задана составленная из прямоугольников область D . Требуется изобразить векторное поле, заданное на этой области.

I.I. Конкретная реализация. Область D дополняется до прямоугольной D' . Полученная область покрывается сеткой, вообще говоря, неравномерной. В узлах этой сетки задаются векторы. Вектор можно задать двумя способами. Можно задавать его проекции на оси, либо задавать длину вектора и угол наклона его к оси X . В данной статье рассматривается только первый тип задания векторов.

Проекциям векторов, не принадлежащим D , присваивается определенное число. В дальнейшем будем обозначать его "пусто". Предусмотрено несколько режимов вычерчивания полей.

Во-первых, можно вычерчивать вектора на заданной сетке (процедура `поле`).

Во-вторых, можно пересчитать вектора на новую сетку и уже на ней вычерчивать поле. Пересчитать можно при помощи линейной интерполяции (процедура `полин`), либо интерполяции сплайн-функциями (процедура `купол`).

2. Процедура `поле`.

2.1. Процедура вычерчивает поле на заданной сетке, т.е. в каждом узле (i, j) рисуется вектор с проекциями (U_{ij}, V_{ij}) .

2.2. Обращение.

`поле (a, b, x, y, u, v, n, m, пусто, гран);`

a , b - размеры листа (в миллиметрах), на котором требуется изобразить поле;
 x - массив абсцисс сетки с граничной парой $[0:n]$;
 y - массив ординат сетки с граничной парой $[0:m]$;
 U - массив проекций векторов на ось x } с граничными
 V - массив проекций векторов на ось y } парами
 n - число вертикалей сетки без 1; $[0:m, 0:n]$;
 m - число горизонталей сетки без 1;
пусто - число, которое перед обращением присваивается проекциям векторов, не попадающим в D ;
гран - если этот параметр имеет значение истина, то граница поля обводится, в противном случае этого не делается.

3. Процедура ПОЛИН.

3.1. Процедура пересчитывает значения проекций векторов на новую сетку и уже на ней вычерчивает поле. Для пересчета значений проекций векторов используется линейная интерполяция.

3.2. Обращение.
ПОЛИН ($a, b, x, y, u, v, x_1, y_1, u_1, v_1$, пусто, гран);

x - массив абсцисс новой сетки с граничной парой $[-I:n]$,
 $x[-I]$ должно быть равно n ;
 y - массив ординат новой сетки с граничной парой $[-I:m]$,
 $y[-I]$ должно быть равно m ;
 x_1 - массив абсцисс первоначальной сетки с граничной парой $[-I:n_1]$, $x_1[-I]$ должно быть равно n_1 ;
 y_1 - массив ординат первоначальной сетки с граничной парой $[-I:m_1]$, $y_1[-I]$ должно быть равно m_1 ;
 u, v - массивы с граничными парами $[0:m_2, 0:n]$,
где m_2 - максимальное из m и m_1 .

Перед обращением эти массивы не заполняются. В процессе работы процедуры в них заносятся значения проекций векторов в узлах новой сетки.

u_1 - массив проекций векторов на ось x } с граничными парами $[0:m_1, 0:n_1]$;
 v_1 - массив проекций векторов на ось y }

На длины массивов накладывается ограничение:

n, n_1, m, m_1 не должны быть больше 40.
Остальные параметры те же, что и в процедуре `поле`.

4. Процедура `купол`.

4.1. Процедура пересчитывает значения проекций векторов на новую сетку и уже на ней вычерчивает поле. Для пересчета значений проекций векторов используется бикубическая интерполяция. Постановка задачи бикубической интерполяции и численное ее решение описаны в работе [1].

4.2. Обращение.

`купол (a, b, x, y, u, v, x1, y1, u1, v1,`
`пусто, гран).`

Все параметры имеют тот же самый смысл, что и в процедуре `полин`. Ограничения на длины массивов сохраняются.

5. Общие замечания.

Вектора с проекциями, равными "пусто", не рисуются. Проверка на равенство "пусто" производится по проекции на ось y . Вектора с длиной менее 1,5 мм рисуются стандартно. Они имеют длину 4 мм и их основание заключается в квадратик со стороной 1 мм.

Имеется возможность поместить поле в заданном месте листа. Координатами левого нижнего угла поля считаются координаты пира перед обращением к процедуре. После работы процедуры пиро возвращается в эту же точку. Система координат пользователя также запоминается и восстанавливается.

6. Описание алгоритмов.

6.1. Процедура `поле`. Рассмотрим работу основных блоков процедуры `поле`.

a). Нормировка векторов.

Для того, чтобы вектора не выходили за рамку листа и, по возможности, не пересекались, они нормируются. Нормировка векторов проводится так, чтобы каждый вектор не выходил за пределы прямоугольника сетки, в котором он находится.

Номером прямоугольника сетки будем называть пару (i, j), соответствующую левому нижнему углу. Нормировка достигается проходом по всем узлам и выбором нормы, подходящей для всех векторов. Итак, имеем узел ($\Delta x, \Delta y$). Ищется номер прямоугольника сетки, в котором расположен вектор, соответ-

ствующий узлу (N_x , N_y). Таким номером будет (N_{nx} ,
 N_{ny}). Затем вычисляются длины сторон прямоугольника
(N_{nx} , N_{ny}), которые обозначим через I_x и I_y . Из
всех возможных отношений $v[N_y, N_x]/I_x$ и $v[N_y, N_x]/I_y$ выби-
рается максимальное. Это и будет множитель нормировки.

б). Выбор масштаба.

Масштаб выбирается так, чтобы рисунок полностью поместил-
ся на заданном листе. Возможно, что граничные вектора будут
лежать вне D' . Тогда в R_1 заносится максимальная проек-
ция вектора на ось x , лежащая вне D' с левой стороны
области, в R_2 - с правой. В R_3 и R_4 заносятся макси-
мальные проекции векторов на ось y , лежащие, соответ-
ственно, с верхней и нижней стороны D' . Затем с учетом
этих проекций ищется масштаб.

в). Рисующая часть.

Координатами левого нижнего угла поля считаются координа-
ты пера перед обращением к процедуре. Перо сдвигается на R_1
по оси x и на R_2 по оси y , и считается, что теперь
оно находится в точке с координатами ($x[0]$, $y[0]$). Для
запоминания номера столбца, который вычерчивается в данный
момент, служит логическая переменная P . Если P имеет
значение истина, то столбец четный и рисование начинается с
первого элемента. В противном случае столбец нечетный и рисо-
вание начинается с последнего элемента. Это позволяет эконо-
мить "холостое" движение пера. Исходными данными для рисова-
ния вектора служат R_x , R_y (проекции его на оси) и x ,
 y (координаты вектора). Во всех случаях сначала ставится
точка (чтобы выделить основание вектора), а затем, в зависи-
мости от величины вектора, он рисуется без изменений, либо ему
присваивается определенная длина и основание вектора заключа-
ется в квадратик со стороной 1 мм.

г). Обводка границ.

Сначала соединяются угловые точки сетки. Затем обводится
граница области D . Это достигается четырьмя обращениями к
процедуре рис. Во всех случаях ищется узел, в котором
проекции вектора равны "пусто", и просматриваются окружающие
его узлы. Если в двух рядом лежащих узлах проекции вектора не
равны "пусто", то эти узлы соединяются. Процедура рис
для каждого "пустого" узла просматривает узлы, лежащие ниже

его. Ясно, что, поворачивая сетку на 90° , можно перебрать все случаи.

5.2. Рассмотрим теперь работу основных блоков процедур *полин* и *купол*. В отличие от процедуры *полин* в рассматриваемых процедурах добавлен блок пересчета проекций векторов на новую сетку. Процедура *полин* использует для этой цели линейную, а *купол* кусочно-кубическую интерполяцию. Алгоритм пересчета одинаков для обеих процедур.

Рассмотрим работу блока пересчета для процедуры *купол*. Пусть $D' = \{(x, y) : x \in [a, b], y \in [c, d]\}$ – некоторая прямоугольная область. В D' заданы сетки

$$S = \{(x_k, y_e) : a = x_0 < x_1 < \dots < x_n = b, c = y_0 < y_1 < \dots < y_m = d\}$$

и

$$S_1 = \{(x'_k, y'_e) : a = x'_0 < x'_1 < \dots < x'_n = b, c = y'_0 < y'_1 < \dots < y'_m = d\}$$

В узлах сетки S_1 заданы значения проекций векторов на оси x и y , равные $\{U_{j,i}\}_{i=0}^m$ и $\{V_{j,i}\}_{i=0}^m$ соответственно. Причем в узлах сетки S_1 , не принадлежащих D' , проекциям векторов присвоено "пусто". Требуется, используя кусочно-кубическую интерполяцию, найти значения $\{U_{j,i}\}_{i=0}^m$ и $\{V_{j,i}\}_{i=0}^m$ в узлах сетки S . Значения проекций векторов в узлах, не принадлежащих D' , должны быть равны "пусто".

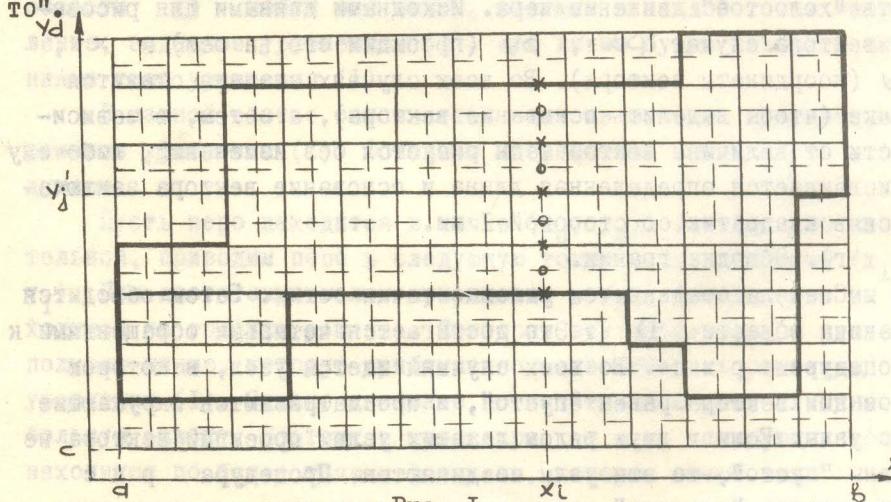


Рис. I.001. Квадрат "Куботочку" с сетками

На рис. I сплошными линиями нанесена сетка S_1 и пунктиром — S . Область D выделена жирными линиями.

Значения проекций векторов в узлах сетки S ищутся по столбцам. Пусть требуется найти эти значения в узлах, лежащих на пересечении прямой $x = x_i$ с прямыми $y = y_j$ ($j = 0, 1, \dots, m$). Тогда для каждой прямой $y = y_j$ ($j = 0, 1, \dots, m$) ищется целое H такое, что для узла (H, j) сетки S_1 выполняются следующие условия:

- 1) проекции вектора в узле (H, j) не равны "пусто";
- 2) либо $H = 0$, либо проекции в узле $(H-1, j)$ равны "пусто";
- 3) если $x_{e-1} < x_i \leq x_e$, то для любого $q = H, H+1, \dots, e$ проекции в узле (q, j) не равны "пусто".

Аналогично ищется целое K такое, что для узла (K, j) выполняются условия:

- 1) проекции вектора в узле (K, j) не равны "пусто";
- 2) либо $K = m$, либо проекции в узле $(K+1, j)$ равны "пусто";
- 3) если $x_e < x_i < x_{e+1}$, то для любого $q = \ell, \ell+1, \dots, K$ проекции в узле (q, j) не равны "пусто".

Иначе говоря, ищется максимальная часть строки сетки S_1 , целиком лежащая в D и содержащая узел, лежащий на пересечении прямых $x = x_i$ и $y = y_j$. Если таких H и K найти не удается, то этот узел не принадлежит D и проекциям присваивается "пусто". Далее проводится кусочно-кубическая интерполяция по узлам сетки S_1 вдоль прямой $y = y_j$, ограниченной прямыми $x = x_H$ и $x = x_K$. В результате мы будем иметь значения проекций в узлах, лежащих на пересечении прямой $x = x_i$ с прямыми $y = y_j$ ($j = 0, 1, \dots, m$).

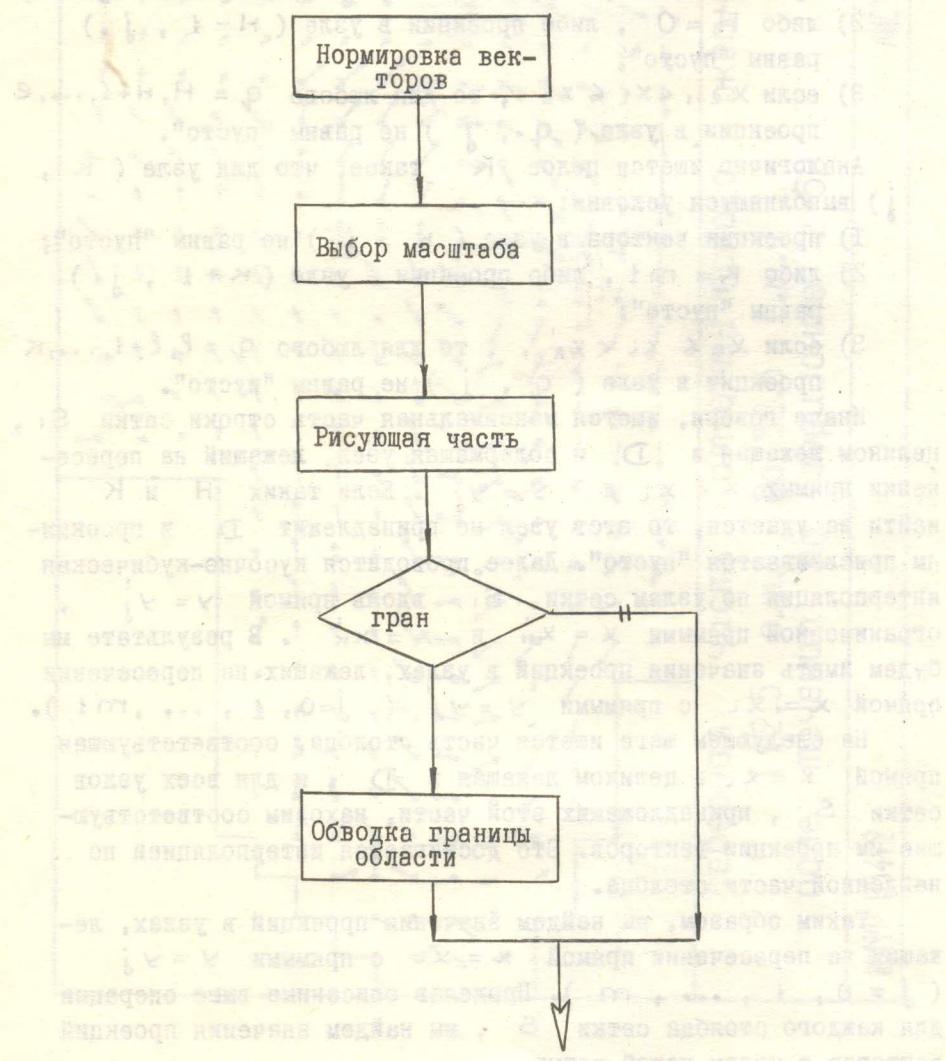
На следующем шаге ищется часть столбца, соответствующая прямой $x = x_i$ и целиком лежащая в D , и для всех узлов сетки S , принадлежащих этой части, находим соответствующие им проекции векторов. Это достигается интерполяцией по найденной части столбца.

Таким образом, мы найдем значения проекций в узлах, лежащих на пересечении прямой $x = x_i$ с прямыми $y = y_j$ ($j = 0, 1, \dots, m$). Проделав описанные выше операции для каждого столбца сетки S , мы найдем значения проекций векторов в узлах новой сетки.

Дальнейшие действия полностью аналогичны действиям процедуры **поле**.

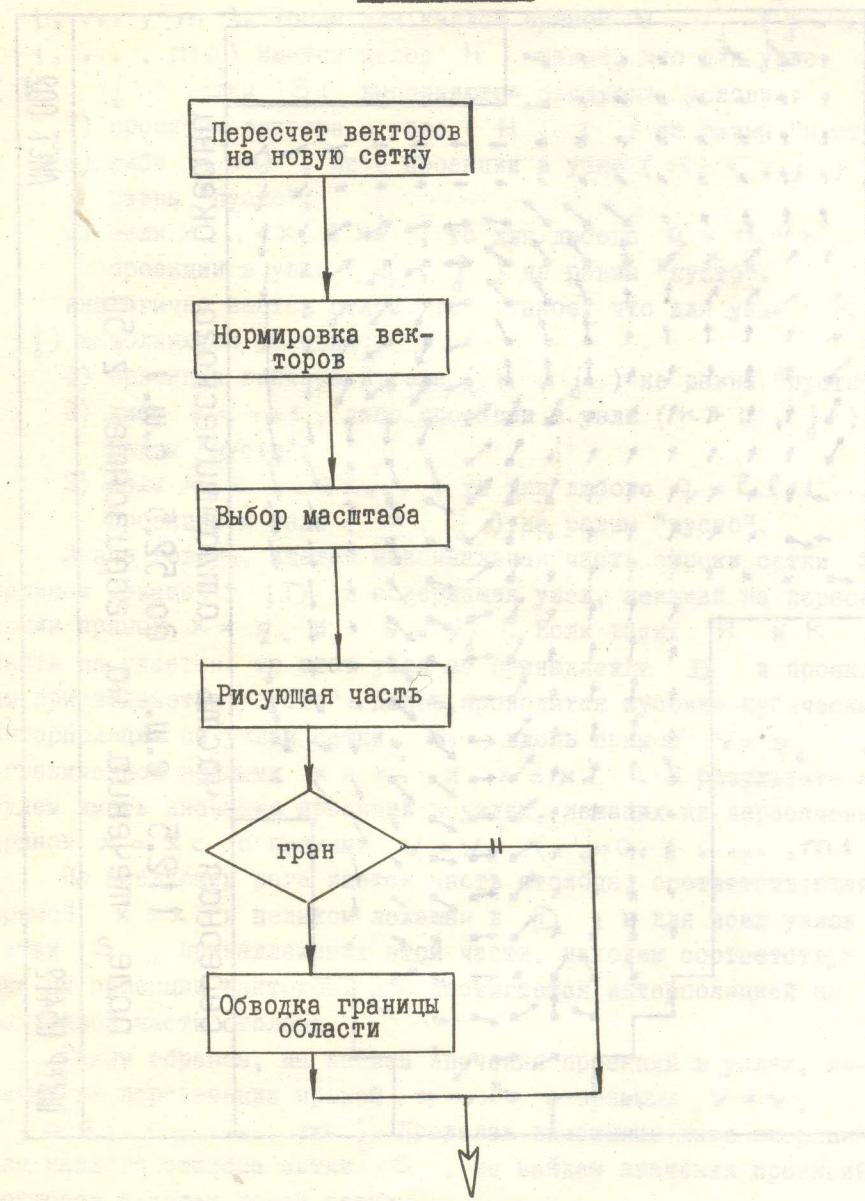
Процедура **поле** ведет здесь аналогично

Блок-схема I.

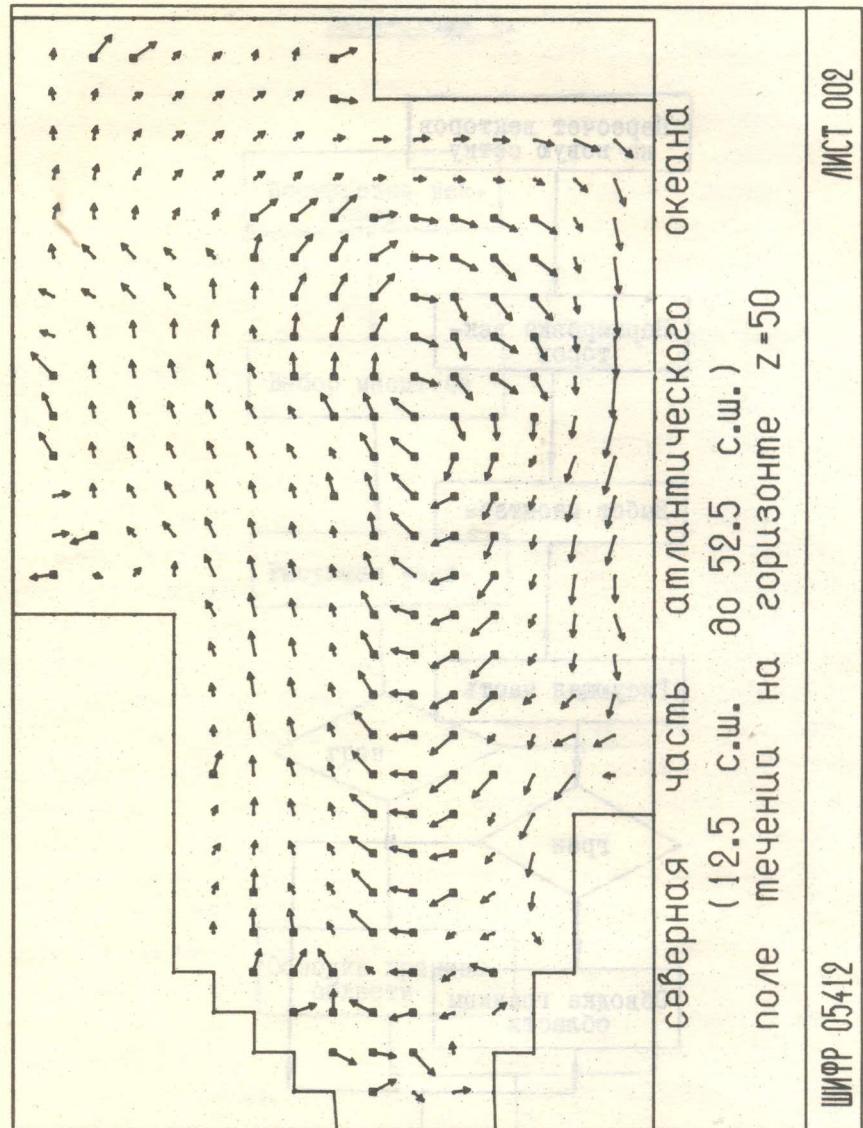


Процедуры полиномиального

Блок-схема П.



Пример использования процедуры поле.



ЛИТЕРАТУРА

1. Марчук Г.И. Методы вычислительной математики. Новосибирск, 1972.
2. Математическое обеспечение графопостроителей (I уровень). Под редакцией Куртукова А.Я. ВЦ СО АН СССР, 1971.
3. Завьялов Ю.С. Интерполяция кубическими многочленами. Интерполяция бикубическими многочленами. В сб. "Вычислительные системы". Ин-т математики СО АН СССР, № 38.