

7. ПОСТРОЕНИЕ ЛИНИЙ УРОВНЯ ФУНКЦИЙ ДВУХ ПЕРЕМЕННЫХ

Карты изолиний для целей исследования функций двух переменных широко применялись еще в "домашинный" период. В сущности задача сводится к изображению линий $f(x, y) = c$, где $f(x, y)$ - исследуемая функция, а c - константа. При построении изолиний функция чаще всего задается не аналитически, а некоторым множеством значений. При сеточном задании эти значения известны в узлах сетки.

Мы будем иметь дело с функциями, которые заданы в узлах прямоугольной сетки. При построении изолиний функция доопределяется либо всюду (см. гл. 6), либо только на ребрах прямоугольных ячеек, из которых складывается сетка. В последнем случае используется линейная интерполяция.

Процесс построения изолинии разбивается на три этапа: поиск изолинии (определение начальной точки на изолинии), отслеживание изолинии (нахождение на изолинии последовательности точек) и, наконец, оформление изолинии (вписывание значений в разрывы изолиний, нумерация, расстановка берг-штрихов).

В настоящей главе описаны три варианта программ. В первом варианте (п. 7.1) используется сплайн-интерполяция (см. гл. 6), и его целесообразно применять там, где сетка значений редкая. Разумеется, расход машинного времени в этом варианте будет заметно большим, чем в двух других (пп. 7.2 и 7.3). Два других варианта не имеют принципиальных отличий. В обоих случаях используется линейная интерполяция на ребрах ячейки. Однако третий вариант экономнее в отношении памяти и имеет более широкие оформительские средства.

Описываемые варианты программ позволяют строить карты изолиний не только в декартовой системе координат, но и в произвольной системе координат, однозначно связанной с декартовой. Однако делают они это по-разному. В первом и втором вариантах линии уровня сначала строятся на плоскости в произвольной системе координат точно также, как в случае декартовой системы, а затем производится отображение этой плоскости вместе с изолиниями в декартову систему координат. В отличие от этого, в третьем варианте программ недекартово пространство сначала преобразуется в декартово, а в нем уже линии уровня строятся обычным образом.

7.1. Построение изолиний с использованием гладкого восполнения функций

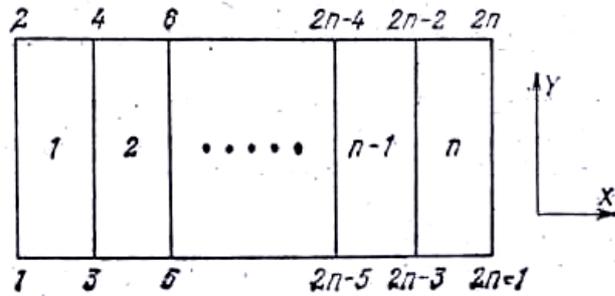
В этом параграфе рассматривается метод построения изолиний для функций, заданных таблично на прямоугольной неравномерной сетке в произвольной системе координат, однозначно связанной с декартовой (см. также [1, и]). Предполагается, что сеточная функция является достаточно гладкой, например, получена как результат восполнения одним из способов, описанных в гл. 6.

7.1.1. Метод поиска и отслеживания изолиний. Пусть задана функция $Z(X, Y)$, вычисляемая каким-либо способом в любой точке прямоугольной области, искомые изолинии будут решением уравнения

$$Z(X, Y) = C. \quad (7.1)$$

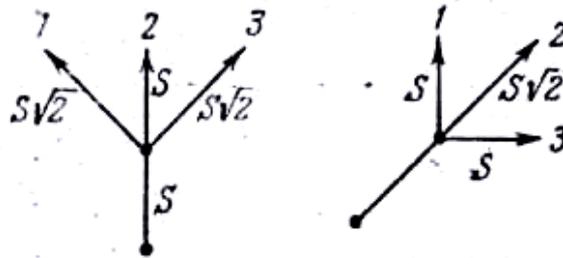
Если функция в заданной области не имеет локальных экстремумов, то решением будут кривые, пересекающие границу области. И если найти корни уравнения (7.1) на границе области, то, отслеживая изолинии, начинающиеся в найденных точках, мы получим всю карту изолиний. Эта идея и лежит в основе описываемого метода.

Пусть теперь функция, определенная на прямоугольной области, имеет локальные экстремумы. Чтобы провести линии уровня, заданные уравнениями (7.1), мы сводим задачу к предыдущей, т. е. разбиваем область на прямоугольные *окна*, в которых функция уже не имеет локальных экстремумов, или, говоря точнее, локальные экстремумы в заданных окнах несущественны для картины рельефа функции во всей области. Проведем разбиение так, как показано на рисунке, учитывая, что количеством окон можно управлять.



Поиск корней уравнения (7.1) на границе окон осуществляется путем деления сторон окон на части. Каждая часть проверяется на наличие корня, и если корень есть, то он вычисляется методом деления пополам с достаточной точностью. Таким образом, ищутся корни на границах каждого окна. После того как корни найдены, проводятся изолинии с начальными точками в этих корнях, а чтобы не было повторного отслеживания изолинии, запоминаются их точки выхода на границу окон. При выборе корня для следующей изолинии проверяется его близость к точкам выхода. Если оказалось, что корень близок к точке выхода, то он отбрасывается и рассматривается следующий.

Опишем теперь алгоритм отслеживания изолиний. Пусть нам известен корень уравнения (7.1). Начальное направление линии выбирается ортогональным к стороне окна, на которой найден корень. Пусть шаг ведения линии равен S . Следующая точка линии выбирается исходя из условия минимума отклонения функции от заданного значения C в точках 1, 2, 3:



Такой метод, вообще говоря, непригоден для функций, линии уровня которых имеют изломы, но, уменьшая шаг S , можно добиться любого приближения к идеальной изолинии. В нашем же случае, когда сеточная функция восполняется гладкой, от этого метода следует ожидать хороших результатов.

7.1.2. Описание программ. Описанный выше метод построения карт изолиний реализован в двух программах CONDEK и DRACON. Программа CONDEK строит линии уровня функции двух переменных, заданной на декартовой прямоугольной сетке с неравномерным шагом и восполняемой гладким образом программой ITPLBV (см. 6.2). Программа CONDEK может нарисовать изолинии N уровней, причем уровни могут задаваться в произвольном порядке. Имеется также аппарат маркировки изолиний, т. е. в разрывы изолиний вписывается номер уровня в порядке их расположения во входном массиве. При этом контролируется, чтобы номера разных уровней не накладывались друг на друга.

Программа DRACON также рисует изолинии функций двух переменных, но не только на декартовой сетке, а например, на полярной, логарифмической или какой-либо другой. Поэтому в эту программу введены два параметра, которыми должны быть имена подпрограмм-функций, описанных в операторе EXTERNAL. Эти подпрограммы-функции должны преобразовывать входные координаты в декартовы. Например, в случае полярной сетки две подпрограммы-функции преобразования полярных координат в декартовы будут иметь следующий вид:

FUNCTION XPOL(RO,FI)	FUNCTION YPOL(RO,FI)
XPOL=RO * COS(FI)	YPOL=RO * SIN(FI)
RETURN	RETURN
END	END

Параметры K в программе DRACON и KO в программе CONDEK имеют разный смысл. Оба параметра управляют разбиением области на окна, но параметр K задает не количество окон в области, а количество полос исходной сетки, которые будут рассматриваться в качестве одного окна области рисования. Это различие вызвано тем, что при использовании большинства преобразований сетка деформируется нелинейно, и, следовательно, выбор шага ведения изолинии в исходных координатах будет отнимать довольно много времени. Если же за окна принять полосы сетки, то выбор шага сильно упростится, что приведет к сокращению времени вычисления. Маркировка изолиний в программе DRACON производится аналогично тому, как это делается в CONDEK.

При использовании программы DRACON возникает необходимость в определении пределов изменения координат сетки, преобразованной функциями FUNX и FUNY, а также в очерчивании границ области рисунка, в которой изображаются изолинии. Для этой цели предназначены соответственно программы DRALIM и DRFRAM.

Программа CONDEK(LX,LY,X,Y,Z,STEP,KO,N,C) предназначена для вычерчивания изолиний функции, заданной на прямоугольной сетке в декартовой системе координат и восполняемой программой ITPLBV. Программа имеет следующие параметры:

LX, LY - число точек сетки по осям X и Y ;

X, Y - координаты точек сетки по осям X и Y , расположенные в возрастающем порядке;

Z - двумерный массив размером (LX,LY), задающий значения функции $Z(X, Y)$ в узлах сетки;

STEP - шаг отслеживания изолинии (в выбранных единицах измерения);

KO - целое положительное число, задающее количество окон, на которое делится область;

N - количество уровней;

C - одномерный массив значений уровней.

Программа DRACON(LX,LY,X,Y,Z,STEP,K,N,C,FUNX,FUNY) вычерчивает изолинии функции $Z(X, Y)$, заданной на произвольной сетке и восполняемой программой ITPLBV. Преобразование сетки в декартову производится с помощью подпрограмм-функций FUNX и FUNY. Программа имеет следующие параметры:

K - число полос сетки, рассматриваемых в качестве одного окна ($1 \leq K \leq LX-1$);

FUNX, FUNY - подпрограммы-функции преобразования сетки (X, Y) в декартову (U, V) (соответствие сеток устанавливается следующим образом: $U = FUNX(X, Y)$, $V = FUNY(X, Y)$); имена этих подпрограмм-функций должны быть описаны в операторе EXTERNAL).

Остальные параметры аналогичны параметрам программы CONDEK.

Замечания и рекомендации:

1. При вызове программы CONDEK нет смысла задавать большое количество окон (больше LX), так как это потребует дополнительных затрат процессорного времени. По той же причине не рекомендуется задавать параметр STEP меньшим 0,1 мм.

2. Преобразование сетки в декартову не должно сильно отличаться от изометрического, линейного или аффинного, иначе могут возникнуть побочные эффекты, приводящие к искажению картинки.

3. Для того чтобы воспользоваться программой гладкого восполнения SMTSIM вместо ITPLBV, достаточно подложить в функциональную часть пакета следующую подпрограмму:

```
SUBROUTINE ITPLBV(LX,LY,X,Y,Z,N,U,V,W)
CALL SMTSIM(LX,LY,X,Y,Z,N,U,V,W)
RETURN
END
```

Если же потребуется восполнить функцию с разными степенями гладкости - 0, 1, 2, 3 - и без симметризации, то можно использовать следующую подпрограмму:

```
SUBROUTINE ITPLBV(LX,LY,X,Y,Z,N,U,V,W)
CALL SMTBVF(L,LX,LY,X,Y,Z,N,U,V,W)
RETURN
END
```

где вместо L должно быть одно из значений - 0, 1, 2 или 3. Однако при этом не следует забывать о затратах процессорного времени на восполнение.

Программа DRALIM(LX,LY,X,Y,FUNX,FUNY,R) предназначена для определения пределов UMIN, UMAX, VMIN, VMAX изменения координат сетки, преобразованной функциями FUNX и FUNY, и имеет следующие параметры:

LX, LY - число точек сетки по осям X и Y;

X, Y - координаты точек сетки по осям X и Y;

FUNX, FUNY - подпрограммы-функции преобразования сетки (X,Y) в декартову (U,V) (соответствие сеток устанавливается следующим образом:

$U = FUNX(X,Y)$, $V = FUNY(X,Y)$);

R - параметр, вычисляемый программой; $R = (VMAX - VMIN)/(UMAX - UMIN)$.

Значения пределов помещаются в общий блок DRLM, описываемый следующим образом COMMON /DRLM/UMIN,UMAX,VMIN,VMAX.

Программа DRFRAM(LX,LY,X,Y,FUNX,FUNY) служит для очерчивания границ области, в которой рисуются изолинии с помощью программы DRACON. Ее параметры имеют тот же смысл, что и в программе DRALIM.

Служебные программы.

Программа INSDEK(LX,LY,X,Y,Z) используется программой CONDEK для вычерчивания изолиний в окне (без преобразования координат).

Программа INSIDE(LX,LY,X,Y,Z,FUNX,FUNY) используется программой DRACON для вычерчивания изолиний в окне (с преобразованием координат).

Программа FNROOT(LX,LY,X,Y,Z,NROOTS,ROOTS) находит корни уравнений: $Z(X_0, Y_0) = C$, $Z(X, Y_0) = C$.

7.1.3. Примеры.

1. На рис. 7.1 приведены изолинии функции, заданной на прямоугольной сетке в декартовой системе координат. Далее приводится программа, с помощью которой выполнялось построение.

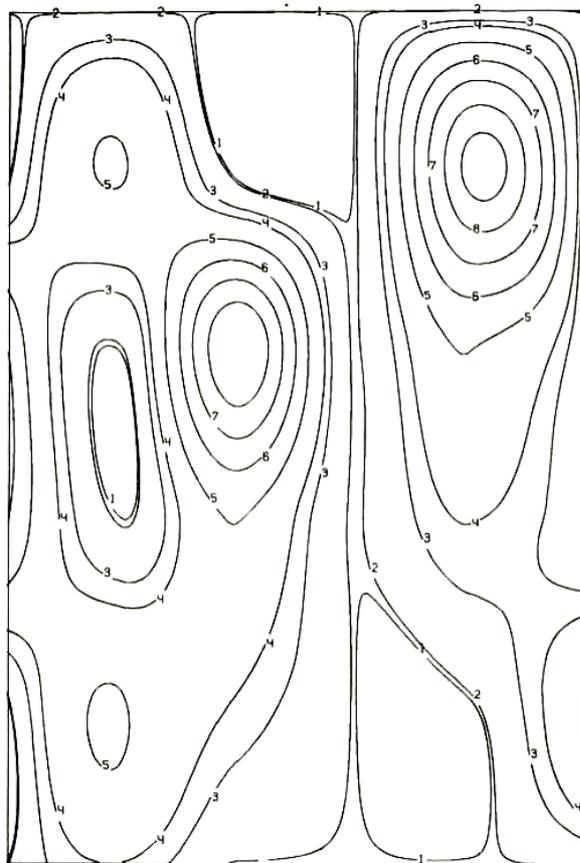


Рис. 7.1. Линии уровня функции двух переменных, заданной на прямоугольной сетке в декартовой системе координат

```
DIMENSION Z(6,6),X(6),Y(6),C(10)
```

```
DATA Z/0.,5.,5*0.,10.,5.,0.,0.,10.,15.,0.,10.,0.,5.,0.,15.,0.,30.,0.,10.,0.,0.,10.,0.,0.,30.,7*0./
```

```
DATA C/-0.1,0.1,3.,5.,10.,15.,20.,25.,29./
```

```

DATA X/1.,2.,3.,4.,5.,6./ Y/1.,2.,3.,4.,5.,6./
CALL PAGE(17.,25.,0,0,1)
CALL REGION(0.,0.,17.,25.,0,0,0)
CALL LIMITS(1.,6.,1.,6.)
CALL CONDEK(6,6,X,Y,Z,0.0125,11,9,C)
CALL ENDPG('7.1')
END

```

2. На рис. 7.2 изображены линии уровня функции, заданной на сетке в полярной и в декартовой системах координат. Ниже приводится программа рисования. Двумерный массив, описывающий поверхность, формируется с помощью программы READZ.

```

DIMENSION X(11),Y(9),Z(11,9),C(10)
DATA C/0.1,1.,5.,10.,20.,30.,40.,50.,60./
EXTERNAL XPOL,YPOL
T=ATAN(1./60)
DO 1 I=1, 11
S=5.*I
IF(I.LE.9) Y(I)=T*(S-5.)
1 X(I)=S
CALL READZ(Z)
CALL PAGE(25.,17.,0,0,0)
CALL DRALIM(11,9,X,Y,XPOL,YPOL,RR)
CALL REGION(0.,0.,25.,RR*25.,0,0,0)

```

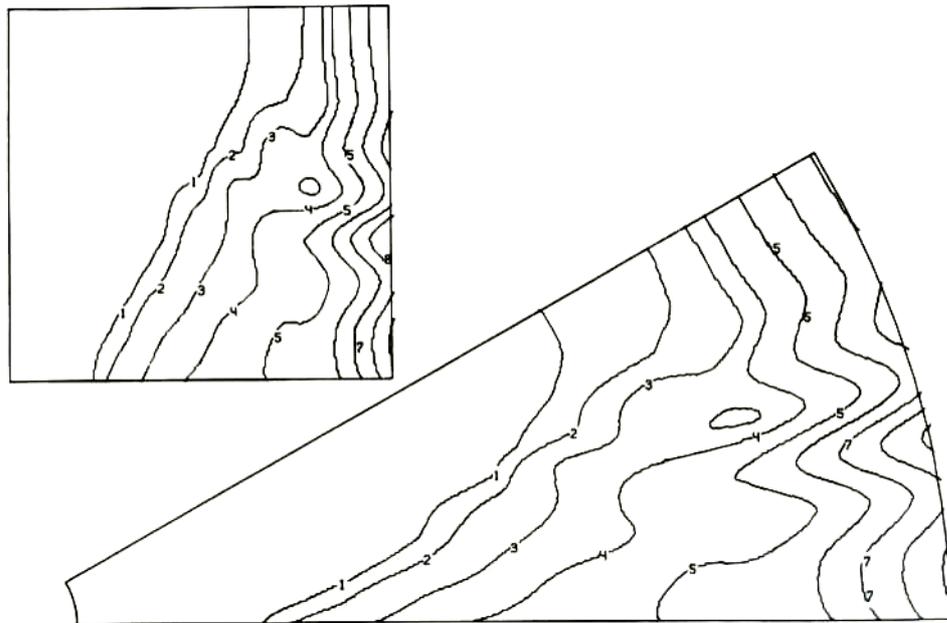


Рис. 7.2. Линии уровня функции, заданной на сетке:
а) в полярной системе координат, б) в декартовой системе координат

```

CALL LIMITS(0.,55.,0.,RR*55.)
CALL DRFRAM(11,9,X,Y,XPOL,YPOL)
CALL DRACON(11,9,X,Y,Z,0.05,10,9,C,XPOL,YPOL)
CALL REGION(0.5,6.5,10.,10.,0,0,1)
CALL LIMITS(X(1),X(11),Y(1),Y(9))
CALL CONDEK(11,9,X,Y,Z,0.05,10,9,C)
CALL ENDPG('7.2')
END

```

7.2. Построение изолиний с использованием линейной интерполяции

В этом параграфе описываются программы, позволяющие строить линии уровня для функции двух переменных. Исходная функция задается прямоугольным массивом чисел на сложной области в произвольной системе координат, однозначно отображаемой на декартову. Изолинии могут быть помечены порядковым номером и значением уровня.

7.2.1. Алгоритм построения изолиний. Данный алгоритм, реализуемый программой ISOLIN, позволяет строить линии уровня (изолинии) для функции двух переменных $Z = Z(X, Y)$, заданной в узлах прямоугольной неравномерной сетки $\{X_i, Y_j\}$, $i = 1, \dots, NX$, $j = 1, \dots, NY$. Двумерная сетка разбивает область определения функции на прямоугольные ячейки ($X_{i-1} \leq X \leq X_i, Y_{j-1} \leq Y \leq Y_j$). Если доопределить функцию $Z(X_i, Y_j)$ на ребрах ячейки, используя линейную интерполяцию, тогда линиями уровня будут ломаные, проходящие через точки пересечения отрезков функции, заданных на ребрах ячеек, с плоскостями $Z = CONT_k$, $k = 1, \dots, NCN$.

Различают *незамкнутые* изолинии, начинающиеся и заканчивающиеся на границе области, и *замкнутые* линии, лежащие целиком внутри области определения функции. Задача построения изолиний решается следующим образом. Для обнаружения начальных точек незамкнутых изолиний осуществляется обход по границе. Как только начальная точка обнаружена, изолиния отслеживается до конца, т. е. до выхода ее на границу. После того как построены все незамкнутые изолинии, производится последовательный просмотр всех горизонтальных ребер ячеек для выявления точки, принадлежащей замкнутой изолинии. После обнаружения такой точки, изолиния отслеживается до конца, т.е. до возврата в эту начальную точку. Как только построены все линии, соответствующие уровню $CONT_k$, производится переход к следующему, $CONT_{k+1}$, уровню и процедура повторяется.

Чтобы исключить повторное проведение изолиний, в процессе отслеживания регистрируется факт ее прохождения через данное ребро. При этом считается, что через каждое ребро нельзя провести более одной линии заданного уровня. Информация о пройденных ребрах сохраняется и используется для определения конца замкнутых изолиний.

При обнаружении начальной точки рассматривается поведение линии внутри ячейки. Если линия вошла в ячейку, она должна выйти из нее через одно из оставшихся трех ребер. Проверять соотношение

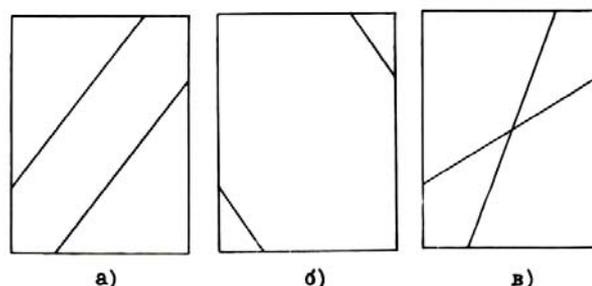
$$(Z(a) - CONT_k) \times (Z(b) - CONT_k) \leq 0,$$

где a, b - граничные точки ребра, определяем через какое ребро линия вышла. Далее рассматривается ячейка, соседняя с этим ребром, и операция повторяется. Координаты точки пересечения линии уровня с ребром находятся с помощью линейной интерполяции. Эти координаты запоминаются в некотором массиве. Линия вычерчивается, когда достигнута граница области (либо в случае замкнутых линий ранее пройденное ребро) или когда массив заполнен.

При поиске линий уровня на сетке могут встретиться вырожденные ситуации:

а) возможно проведение линии уровня через все ребра ячейки и б) линия проходит точно через вершину ячейки.

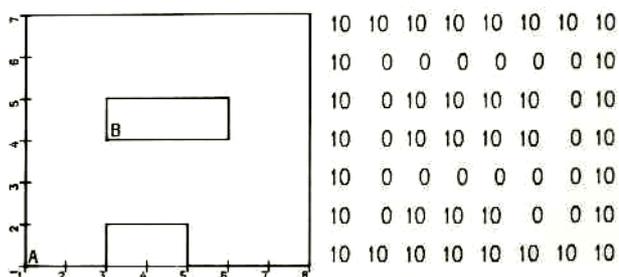
В первом случае возможны три способа соединения точек пересечения линий уровня с ребрами:



Поскольку значения функции известны только в четырех вершинах ячейки, нельзя сказать, какая из конфигураций правильна. Описываемые далее программы в такой ситуации выбирают вариант *в*). Вырождение второго вида приводит либо к повторению линии дважды, либо к ее обрыву. Чтобы избежать этого, к вырожденному узлу добавляется малое значение 10^{-5} . В результате вырождение устраняется, вызывая незаметную для глаза "ошибку" в изолинии. Заметим, что для построения визуально более качественного изображения линий уровня следует задать более частую сетку. Можно, например, воспользоваться методами гладкого восполнения функций, изложенными в гл.6.

Информация о форме области задается ее *описателем* - двумерным массивом ИТАГВ. Заметим, что эта область может иметь довольно сложную конфигурацию (см. пример 2 в п.7.2.4). Определяется она следующим образом: ИТАГВ(I, J) = 0, если точка с индексами (I, J) принадлежит области, и ИТАГВ(I, J) = 10, если точка находится на границе или не принадлежит области. Граница области задается порядковыми номерами ее граничных точек на сетке, хранящихся в массивах IXBD(K), IYBD(K), K = 1, ..., NB.

На рисунке приведен пример сложной области и ее описателя ИТАГВ.



Задание массивов IXBD и IYBD зависит от порядка обхода граничных точек. Например, если начать обход из точки А и сделать фиктивный разрез через точки А и В, то

K	1	2	3	4	5	...	29	30	31	32	...	38
IXBD(K)	1	2	3	3	4	...	1	3	3	4	...	3
IYBD(K)	1	1	1	2	2	...	1	4	5	5	...	4

В этих массивах первая точка является и последней точкой. Кроме того, следует иметь ввиду, что заданная область при обходе должна всегда оставаться слева.

Таким образом, область, на которой рисуются линии уровня, описываются дважды: массивом ИТАГВ и массивами граничных точек IXBD и IYBD. Если заданы массивы граничных точек, то массив ИТАГВ можно подготовить с помощью программы ВІТА.

При построении карт изолиний чаще всего будет использоваться простая прямоугольная область. Для задания такой области служит программа QUADRA.

Линии уровня можно строить и в произвольной системе координат. С этой целью в программе вычерчивания изолиний и в некоторых других предусмотрен параметр ICM. Если ICM = 0, то предполагается декартова система координат, а при ICM = 1 - полярная. Если же необходимо работать в какой-либо другой системе координат, то необходимо установить ICM = 1 и в функциональную часть программы добавить подпрограммы-функции XPOL и YPOL со своими преобразованиями координат (см. пример 3 в п. 7.2.4).

Для определения пределов изменения функции, заданной в недекартовой системе координат, предназначена программа EXTREM.

Для формирования массива уровней CONT предусмотрены следующие возможности. В случае, если неизвестны пределы изменения функции, для которой строятся изолинии, с помощью программы LEVFUN можно определить абсолютные максимум и минимум этой функции и построить массив уровней с равным шагом между ними. Другая программа, LEVMAP, задает массив уровней относительно нулевого уровня по положительному и отрицательному шагам. Кроме того, иногда полезно наряду с основными изолиниями построить также и промежуточные, используя другой тип линии. В этом случае для подготовки массива уровней удобно воспользоваться программой ADDLEV.

При вычерчивании изолиний бывает необходимо различать их. Для этого можно использовать программы, предоставляющие возможность рисовать различные типы линий или помечать их порядковым номером или значением уровня изолинии.

Тип линии определяется установочными программами FULL (сплошная линия) и BROKEN (штриховая и штрихпунктирная линии), вызов которых должен предшествовать обращению к программе ISOLIN. Описание этих программ приведено в п. 4.2.

Две другие программы SIZLIN и NUMLIN позволяют задать соответственно режимы вписывания в разрыв изолинии значения или порядкового номера уровня. Отменить режим вписывания позволяет программа RENUM.

При работе в недекартовой системе координат наряду с выводом изолиний бывает необходимо выводить другие линии. В этом случае следует воспользоваться программой SUPLIN, которая учтет заданное преобразование координат. Кроме того, линия может быть помечена маркером, рисоваться сплошной или прерывистой и т. д.

7.2.2. Описание программ.

Программа ISOLIN(NX,NY,X,Y,Z,ITAGB,NB,IXBD,IYBD,NCN,CONT,ICM) позволяет начертить изолинии непрерывной функции двух переменных. Программа имеет следующие параметры:

NX, NY - число точек сетки по осям X и Y ;

X, Y - массивы координат точек по осям X и Y ;

Z - массив значений функции (размером (NX, NY));

ITAGB - описатель области (массив размером (NX, NY));

|NB| - число граничных точек (если $NB > 0$, область не очерчивается, если $NB < 0$ - очерчивается);

IXBD - массив порядковых номеров граничных точек вдоль оси X (длины NB);

IYBD - массив порядковых номеров граничных точек вдоль оси Y (длины NB);

|NCN| - количество линий равного уровня (если $NCN < 0$, то при большой густоте линий применяется экранирование, в результате чего часть линий не проводится);

CONT - массив значений уровней;

|ICM| - признак системы координат: $ICM = 0$ - декартова система координат, $ICM = 1$ - полярная система координат.

В этой программе все параметры входные.

Программа QUADRA(ITAGB,IXBD,IYBD,NX,NY,NB) позволяет по заданной сетке сформировать массивы описателей области и ее границы, определяя тем самым прямоугольную область, на которой будут вычерчиваться линии уровня. Параметры программы следующие:

ITAGB - описатель области (размером (NX, NY));

IXBD, IYBD - массивы порядковых номеров граничных точек вдоль осей X и Y ;

NX, NY - число точек сетки по осям X и Y ;

NB - число граничных точек, $NB = 2 * NX + 2 * NY - 3$.

В этой программе выходные параметры ITAGB, IXBD, IYBD.

Программа VITA(ITAGB,IXBD,IYBD,NX,NY,NB) формирует описатель области ITAGB по массивам граничных точек. Параметры этой программы аналогичны параметрам программы QUADRA. Выходным здесь является параметр ITAGB.

Программа LEVFUN(Z,NX,NY,CONT,NCN) предназначена для определения абсолютных экстремумов функции и построения массива уровней с равным шагом между экстремумами.

Параметры программы следующие:

Z - массив значений функции (размером (NX, NY));

NX, NY - число точек сетки по осям X и Y ;

CONT - массив значений уровней;

NCN - число уровней.

Замечание. CONT(1) равно максимальному значению Z, а CONT(NCN) - минимальному.

Программа LEVMAP(CONT,NCN,F0,SP,SN,NP) позволяет построить массив уровней относительно нулевого уровня по указанным шагам для положительных и отрицательных уровней. Параметры программы следующие:

CONT - массив значений уровней;

NCN - число уровней;

F0 - значение нулевого уровня;

SP - шаг задания положительных уровней;

SN - шаг задания отрицательных уровней;

NP - число положительных уровней.

Число отрицательных уровней равно $NCN-NP-1$.

Программа ADDLEV(CONT,NCN,IS,CAD,NAD) позволяет построить массив дополнительных уровней. Параметры программы следующие:

CONT - значения основных уровней;

NCN - число основных уровней;

IS - число дополнительных уровней между двумя основными;

CAD - массив дополнительных уровней (длины NAD);

NAD - полное число дополнительных уровней, равно $IS * (NCN - 1)$.

Программа EXTREM(X,Y,NX,NY,XMN,XXM,YMN,YYM,ICM) предназначена для определения пределов изменения аргументов функции на области ее задания в недеклартовой системе координат. Параметры программы:

X, Y - координаты точек по осям X и Y;

NX, NY - число точек сетки по осям X и Y;

XMN, XXM - значения координаты X, соответствующие левой и правой границам ;

YMN, YYM - значения координаты Y, соответствующие нижней и верхней границе;

ICM - признак системы координат: $ICM = 0$ - декартова система координат, $ICM = 1$ - полярная система координат.

Функция XPOL(R,T) позволяет по координатам точки в полярной системе координат вычислить X-координату в декартовой системе координат. Ее параметры: R - радиус точки в полярной системе координат, T - угол между осью абсцисс и радиусом.

Функция YPOL(R,T) позволяет по координатам точки в полярной системе координат вычислить Y-координату в декартовой системе координат. Ее параметры: R - радиус точки в полярной системе координат, T - угол между осью ординат и радиусом.

Программа NUMLIN(SIZE,N) задает режим вписывания в разрыв изолинии порядковых номеров уровней. Ее параметры указывают высоту цифры в выбранных единицах измерения (SIZE) и количество цифр в номере уровня (N).

Программа SIZLIN(SIZE,N,M) задает режим вписывания в разрыв изолинии значения уровня. Ее параметры указывают высоту цифры в выбранных единицах измерения (SIZE), количество цифр в числе (N) и количество дробных знаков (M).

В программе ISOLIN используются служебные программы LOOK, CELL, LINT, ZAPIT.

Программа LOOK(NX,NY,X,Y,ITAGB,XX,YY,KT,ICM) отслеживает линию, следит за ее окончанием или за окончанием массивов XX и YY длины KT, в которых запоминаются точки пересечения линий уровня с ребрами.

Программа CELL(Z,NX,NY) позволяет найти ребро, через которое изолиния выходит из ячейки.

Программа LINT(X,Y,XX,YY,NX,NY,KT,ICM) позволяет произвести линейную интерполяцию на ребре.

Программа ZAPIT(ITAGB,NX,NY) фиксирует прохождение линии уровня через ребро.

7.2.3. Вспомогательные программы. В области можно нарисовать сетку, на которой задана функция Z. Это делается с помощью программы CMGRID. Линии сетки можно пометить как порядковым номером линии, так и соответствующим ей значением X или Y. Разметка линий производится по конечным значениям X и Y.

С помощью программы LOCEXT можно найти локальные экстремумы функций двух переменных в заданной области и нанести их на контурную карту. Поиск экстремумов производится перебором всех точек в области. Точка считается экстремальной, если функция принимает в ней значение большее (меньшее), чем значения функции во всех ближайших окружающих ее точках. Размер вписываемых цифр можно задать программой SIZLIN. По умолчанию этот размер устанавливается равным 0,3 см, и число выдается с одной цифрой после точки.

Иногда на карте изолиний бывает полезно отметить отдельные значения функции Z, так называемые *опорные* точки. Для этой цели предназначена программа KEYPO.

Программа CMGRID(X,Y,NX,NY,XX,YY,SZ,KP,M,N,ICM) позволяет построить сетку по значениям массивов координат X и Y и разметить ее. Разметка сетки производится по ее краям.

Массивы X и Y могут быть заданы в произвольной системе координат, однозначно отображаемой на декартову. Параметры программы следующие:

X, Y - координаты точек по осям X и Y ;

NX - число точек сетки по оси X (если $NX < 0$, сетка по X не рисуется);

NY - число точек сетки по оси Y (если $NY < 0$, сетка по Y не рисуется);

MX - шаг сетки по оси X , (чертится каждая $MX * i + 1$ линия, $i = 0, 1, 2, \dots$);

MY - шаг сетки по оси Y , (чертится каждая $MY * i + 1$ линия, $i = 0, 1, 2, \dots$);

SZ - высота цифр (в выбранных единицах измерения);

KP - количество требуемых дробных знаков в числе;

M - шаг разметки по оси X :

$M < 0$ - сетка метится значениями X с шагом $|M|$, т. е. метится каждый $|M|$ -й узел сетки,

$M = 0$ - сетка не метится,

$M > 0$ - сетка метится порядковыми номерами линий с шагом $|M|$, т. е. метится каждый $|M|$ -й узел сетки;

N - шаг разметки по оси Y ;

$N < 0$ - сетка метится значениями Y с шагом $|N|$, т. е. метится каждый $|N|$ -й узел сетки,

$N = 0$ - сетка не метится;

$N > 0$ - сетка метится порядковыми номерами линий с шагом $|N|$, т. е. метится каждый $|N|$ -й узел сетки;

ICM - признак системы координат: $ICM = 0$ - декартова система координат, $ICM = 1$ - полярная система координат.

Программа $LOCEXT(Z, X, Y, NX, NY, ICM)$ позволяет найти в области определения функции Z локальные экстремумы этой функции и написать их значения, пометив точки экстремумов маркером. Параметры программы следующие:

Z - массив значений функции (размером (NX, NY));

X, Y - координаты точек по осям X и Y ;

NX, NY - число точек сетки по осям X и Y ;

ICM - признак системы координат: $ICM = 0$ - декартова система координат, $ICM = 1$ - полярная система координат.

Если обнаружен максимум, то изображается маркер номер 3 (ромбик), если минимум - то номер 14 (плюс в ромбике). Чтобы задать размер цифр в числе и их количество, необходимо обратиться к программе $SIZLIN$. По умолчанию высота цифр устанавливается равной 0,3 см, и число выдается с одной цифрой после точки.

Программа $KEYPO(Z, H, Y, NX, NY, ISX, ISY, NM, SZ, KP, ICM)$ предназначена для того, чтобы отметить на графике значения функции двух переменных $Z(X, Y)$ в избранных точках.

Параметры программы следующие:

Z - массив значений функции (размером (NX, NY));

X, Y - координаты точек по осям X и Y ;

NX, NY - число точек сетки по осям X и Y ;

ISX - шаг выдачи точек по X (отмечается каждая ISX -я точка);

ISY - шаг выдачи точек по Y (отмечается каждая ISY -я точка);

NM - номер маркера, которым метятся точки;

SZ - высота цифр в пометках;

KP - количество требуемых дробных знаков в числе;

ICM - признак системы координат: $ICM = 0$ - декартова система координат, $ICM = 1$ - полярная система координат.

Программа $SUPLIN(X, Y, N, NM, JS, L, ICM)$ позволяет начертить линию по координатам точек в недекартовой системе координат. Параметры программы следующие:

X, Y - массивы абсцисс и ординат точек;

N - число точек;

NM - номер маркера (если $NM < 0$, изображается маркер уменьшенных размеров);

$|JS|$ - шаг маркировки: $JS > 0$ - линия с маркерами, $JS = 0$ - линия без маркеров, $J < 0$ - линия не проводится, вычерчиваются только маркеры;

L - признак линии: $L = 1$ - сплошная замкнутая линия с маркерами, $L = 2$ - сплошная незамкнутая линия с маркерами, $L \neq 1, 2$ - сплошная или штрихпунктирная линия (в зависимости от установок $FULL$ или $BROKEN$);

ICM - признак системы координат: ICM = 0 - декартова система координат, ICM = 1 - полярная система координат.

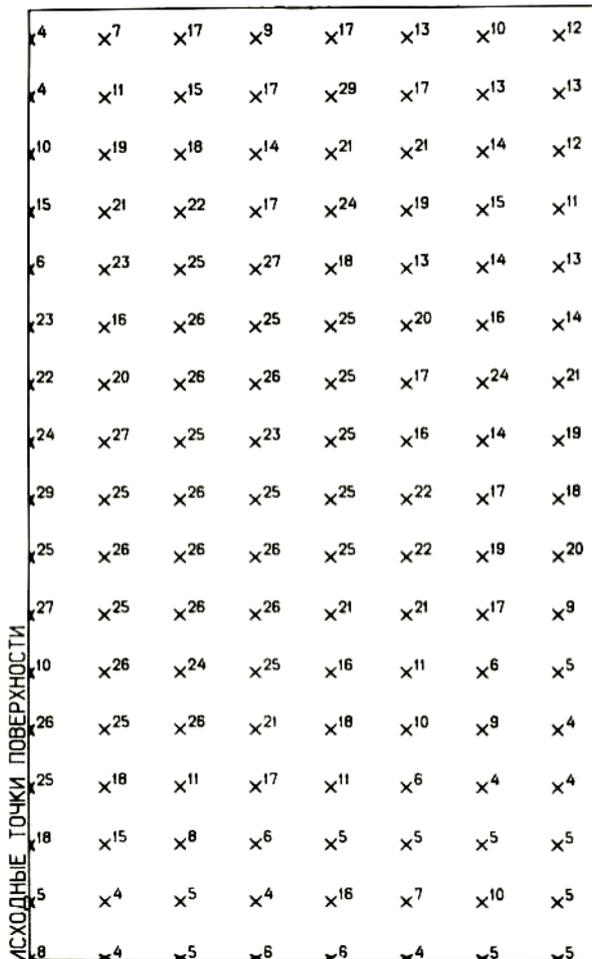


Рис. 7.3. Исходные точки, задающие некоторую поверхность

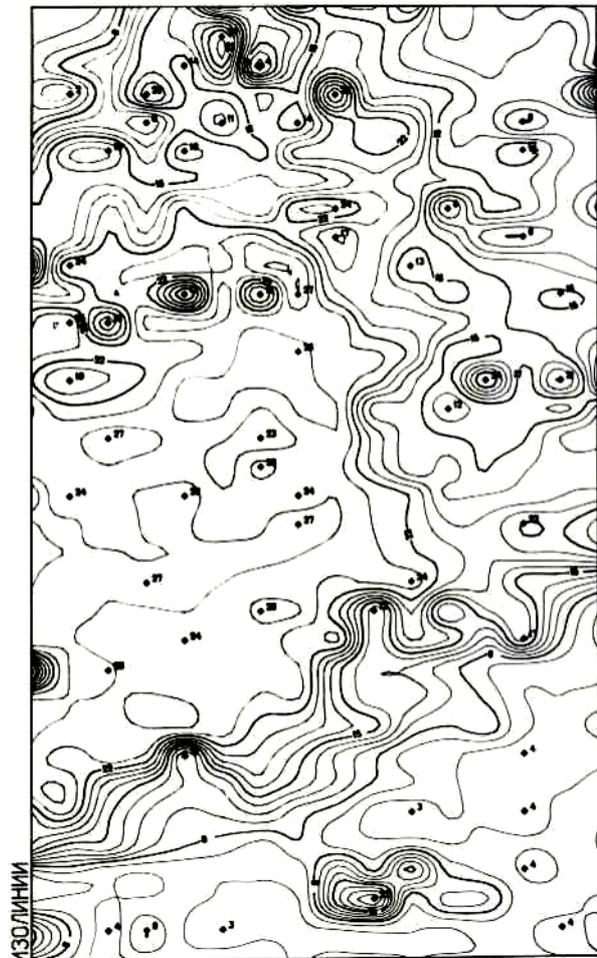


Рис. 7.4. Линии уровня поверхности, исходные точки которой показаны на рис. 7.3

7.2.4. Примеры. 1. На рис. 7.3 и 7.4 показаны соответственно исходные точки и линии уровня некоторой поверхности. Изображение ее центральной проекции приведено ниже (см.рис. 8.11).

Первые два рисунка построены следующей программой:

```

DIMENSION Z(16,34),X(16),Y(34),C(4),CAD(12)
DIMENSION PHI(76,166),XP(76),YP(166)
DIMENSION IT(76,166),IX(481),IY(481)
DATA NX,NY,LX,LY,MX,MY,NG,NC/16,34,76,166,5,5,481,4/
DATA IS,NAD/4,12/ C/0.,10.,20.,30./
999 FORMAT(3X,8F6.2)
READ(1,999) Z
DO 1 I=1,16
1 X(I)=I-1
DO 2 I=1,34
2 Y(I)=I-1
CALL ADDLEV(C,NC,IS,CAD,NAD)
CALL SFCFIT(NX,NY,X,Y,Z,MX,MY,LX,LY,XP,YP,PHI)
CALL QUADRA(IT,IX,IY,LX,LY,NG)
CALL PAGE(16,26,'ИСХОДНЫЕ ТОЧКИ ПОВЕРХНОСТИ',26,1)
CALL REGION(.2,.2,15.,25.,0,0,0)
CALL LIMITS(0.,15.,0.,33.)
CALL KEYPO(Z,X,Y,NX,NY,1,1,-1.,35,0,0)
CALL ENDPG('7.1')
CALL PAGE(15.,25.,'ИЗОЛИНИИ',8,1)

```

```

CALL LIMITS(0.,15.,0.,33.)
CALL SIZLIN(.3,2,0)
CALL ISOLIN(LX,LY,XP,YP,PHI,IT,NG,IX,IY,NC,C,0)
CALL LOCEXT(Z,X,Y,NX,NY,0)
CALL RENUM
CALL ISOLIN(LX,LY,XP,YP,PHI,IT,NG,IX,IY,NAD,CAD,0)
CALL ENDPG('7.4')
END

```

2. На рис. 7.5 изображены линии уровня поверхности, заданной на области со сложной границей. Граница задается массивами граничных точек. Описатель области подготавливается программой BITA.

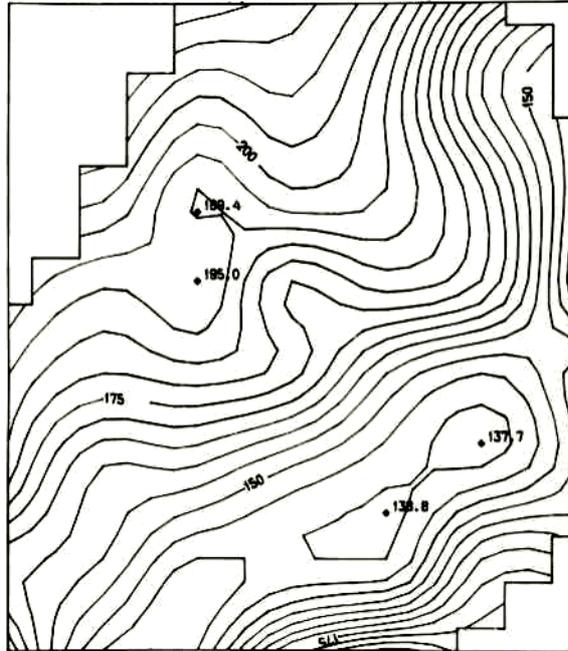


Рис. 7.5. Линии уровня поверхности, заданной на области со сложной границей

```

DIMENSION Z(7,8),X(7),Y(8),ZS(25),YS(29),IX(105),IY(105)
DIMENSION IT(25,29),C(5),CAD(16)
DATA C/125.,150.,175.,200.,225./
DATA X,Y/1.,2.,3.,4.,5.,6.,7.,1.,2.,3.,4.,5.,6.,7.,8./
99 FORMAT(7F5.2)
98 FORMAT(10I3)
READ(1,99) Z
READ(1,98) IX,IY
CALL SFCFIT(7,8,X,Y,Z,4,4,25,29,XS,YS,ZS)
CALL BITA(IT,IX,IY,25,29,105)
CALL ADDLEV(C,5,4,CAD,16)
CALL PAGE(15.,17.,0,0,1)
CALL LIMITS(1.,7.,1.,8.)
CALL LOCEXT(ZS,XS,YS,25,29,0)
CALL ISOLIN(25,29,XS,YS,ZS,IT,105,IX,IY,16,CAD,0)
CALL SIZLIN(.3,3,0)
CALL ISOLIN(25,29,XS,YS,ZS,IT,-105,IX,IY,5,C,0)
CALL ENDPG('7.5')
END

```

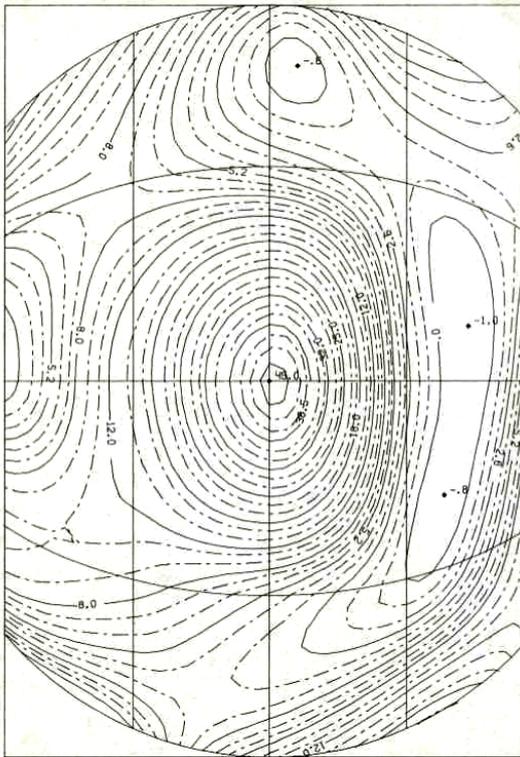


Рис. 7.6. Карта изолиний поверхности, заданной в сферической системе координат

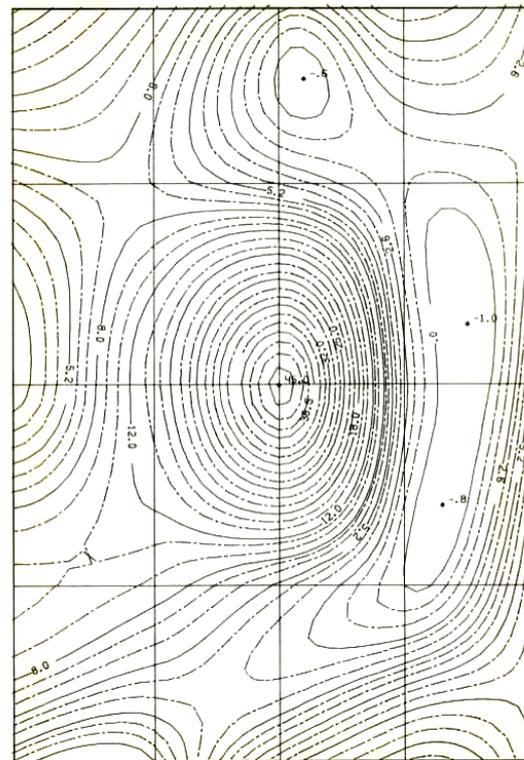


Рис. 7.7. Карта изолиний поверхности, заданной в декартовой системе координат

3. Если поверхность задана в некоторой произвольной системе координат, которая однозначно отображается на декартову, то можно построить карту изолиний такой поверхности. Для этого необходимо описать преобразование координат в подпрограммах-функциях XPOL и YPOL и вставить их в функциональную часть пакета. Границы области в этом случае устанавливаются программой EXTREM.

На рис. 7.6 и 7.7 показаны линии уровня одной и той же поверхности в предположении, что в первом случае исходные точки поверхности заданы в сферической системе координат, а во втором - в декартовой.

```

DIMENSION Z(5,5),XZ(5),YZ(5),C(10),CAD(20)
DIMENSION PHI(41,41),ITAGB(41,41),X(41),Y(41)
DIMENSION IXBD(164),IYBD(164)
DATA Z/1.,0.,10.,20.,12.,11.,9.,5.,0.,10.,2.,15.,45.,3.,5.,6.,7.,4.,1.,2.,21.,11.,3.,7.,3./
DATA XZ/0.736,1.18,1.57,1.964,2.36/
DATA YZ/-.736,-.393,0.,.393,.736/
DATA C/0.,2.6,5.2,8.,12.,18.,25.,32.,38.,44./
DATA NAD,NCN,L1,L2,N1,N2,MX,MY,N5,IS/18,10,5,5,41,41,10,10,161,2/
CALL SFCFIT(L1,L2,XZ,YZ,Z,MX,MY,N1,N2,X,Y,PHI)
CALL QUADRA(ITAGB,IXBD,IYBD,N1,N2,N5)
CALL ADDLEV(C,NCN,IS,CAD,NAD)
CALL PAGE(17.,25.,0,0,1)
CALL EXTREM(X,Y,N1,N2,XMN,XXM,YMN,XXY,1)
CALL LIMITS(XMN,XXM,YMN,XXY)
CALL CMGRID(X,Y,N1,N2,MX,MY,0.,0,0,0,1)
CALL LOCEXT(PHI,X,Y,N1,N2,1)
CALL SIZLIN(.3,4,1)
CALL ISOLIN(N1,N2,X,Y,PHI,ITAGB,N5,IXBD,IYBD,NCN,C,1)
CALL RENUM
CALL BROKEN(.5,2.,1,2)

```

```

CALL ISOLIN(N1,N2,X,Y,PHI,ITAGB,N5,IXBD,IYBD,NAD,CAD,1)
CALL ENDPG('7.6')
CALL PAGE(17.,26.,0,0,1)
CALL EXTREM(X,Y,N1,N2,XMN,XXM,YMN,XXY,0)
CALL LIMITS(XMN,XXM,YMN,XXY)
CALL LOCEXT(PHI,X,Y,N1,N2,0)
CALL CMGRID(X,Y,N1,N2,XX,YY,0.,0.,0.,0.)
CALL FULL
CALL SIZLIN(.3,4,1)
CALL ISOLIN(N1,N2,X,Y,PHI,ITAGB,N5,IXBD,IYBD,NCN,C,0)
CALL RENUM
CALL BROKEN(.5.,1.,1.,1)
CALL ISOLIN(N1,N2,X,Y,PHI,ITAGB,N5,IXBD,IYBD,NAD,CAD,0)
CALL ENDPG('7.7')
END
FUNCTION XPOL(X,Y)
XPOL=-COS(X)
RETURN
END
FUNCTION YPOL(X,Y)
YPOL=SIN(X)*SIN(Y)
RETURN
END

```

7.3 Построение изолиний и линий пересечения поверхностей

В этом параграфе представлены программы для построения изолиний и пространственных кривых, образованных пересечением поверхностей. Изолинии могут вычерчиваться линиями различного типа и снабжаться бергштрихами. Можно также вписывать в разрывы линий номера или значения уровней. Метод позволяет работать с поверхностями, заданными в произвольной системе координат, которая однозначно преобразуется в декартову.

7.3.1. Поиск и проведение изолиний. Задача нахождения и построения изолиний рассматривается в следующей постановке.

А. Изолинии являются решением уравнения $F(X, Y) = CONST$ на области задания $Z = F(X, Y)$.

Б. Проекция линий пересечения двух функций $Z1 = F(X, Y)$, $Z2 = \Phi(X, Y)$ на области их задания являются решениями уравнения

$$\Phi(X, Y) = F(X, Y) \quad (7.1)$$

Если $F(X, Y) - \Phi(X, Y) = P(X, Y)$, то эта задача сводится к случаю А в виде

$$P(X, Y) = 0. \quad (7.2)$$

В. Пространственные линии пересечения двух поверхностей L также являются решением уравнения (7.2). Найдя множество значений (XI, YI) , удовлетворяющих (7.2), L можно выразить через $F(X, Y)$ или $\Phi(X, Y)$ следующим образом:

$$L(XI, YI) = F(XI, YI) = \Phi(XI, YI).$$

Таким образом, все три задачи сводятся к решению первой задачи.

Алгоритм, который использовался при решении поставленной задачи, достаточно традиционен (см. п. 7.2). Он позволяет строить изолинии для функций двух переменных $Z = F(X, Y)$, заданных в узлах прямоугольной неравномерной сетки. Если доопределить функцию на ребрах ячеек, используя линейную интерполяцию, тогда линиями уровня будут

ломаные, проходящие через точки пересечения отрезков функции, заданных на ребрах ячеек, с плоскостью $Z = \text{CONST}$.

В семействе изолиний различают незамкнутые изолинии, начинающиеся и заканчивающиеся на границе области задания, и замкнутые, лежащие целиком внутри области задания функции.

При построении незамкнутых линий производится обход по границе, чтобы найти начальные точки изолиний. Как только такая точка обнаружена, изолиния отслеживается до конца, т. е. до выхода ее на границу области задания функции.

Когда все незамкнутые линии данного уровня построены, производится последовательный просмотр всех горизонтальных ребер ячеек с целью выявления точек, принадлежащих замкнутым изолиниям. Когда таковые найдены, изолинии отслеживаются до конца, т. е. до возврата в эти начальные точки. Эта процедура повторяется для всех заданных уровней. Чтобы исключить повторное проведение изолинии, в процессе отслеживания регистрируется факт ее прохождения через ребро (через каждое ребро нельзя провести более одной линии данного уровня).

Отметим некоторые особенности реализации алгоритма. Рассматриваемая поверхность предварительно "поднимается" в положительное полупространство на величину $|ZMIN|+1.0$, где $ZMIN$ - минимальное отрицательное значение функции в области задания либо нуль. Уровни также изменяются на эту величину. Поскольку все значения поверхности в результате становятся положительными, прохождение изолинии через ребро отмечается присвоением знака минус значениям функции в узлах, соединяемых этим ребром. Тем самым удастся избежать использования вспомогательного массива для хранения информации о прохождении изолинии через данное ребро. После построения всех линий уровня поверхность возвращается к исходному виду.

При отслеживании изолинии k -го уровня первое ребро, через которое проходит изолиния, определяется из следующих условий:

$$Z(I+1,J) > ZIZ(K) \geq |Z(I,J)|$$

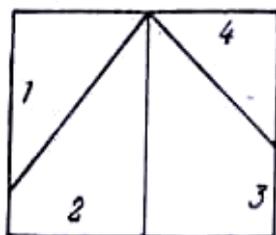
или

$$|Z(I+1,J)| > ZIZ(K) \geq Z(I,J) \geq 0,$$

где $ZIZ(K)$ - величина k -го уровня.

При выполнении любого из этих условий запоминаются соответствующие индексы I, J , и программа пытается отследить изолинию, перебирая последовательно в направлении против часовой стрелки ребра, через которые проходит изолиния. Координаты пересечения изолинии с ребром запоминаются в буферных массивах XI, YI .

На рисунке



показан элементарный просмотр одной ячейки и возможные варианты прохождения изолинии. Цифры на ребрах указывают порядок просмотра.

Отслеживание изолинии производится до тех пор пока не удастся выйти на ее начало (или на границу области, если линия начинается на границе). Когда буферные массивы координат заполнятся или изолиния будет просмотрена до конца, вся изолиния либо часть ее вычерчивается. Затем производится переход к обработке следующей изолинии или к продолжению текущей, если она выдана не полностью.

Программная реализация алгоритма позволяет строить карты изолиний как во всей области задания функции, так и в ее подобластях. Для работы программ требуются два буферных массива под координаты изолинии. Размеры этих массивов определяются программистом в зависимости от имеющейся свободной памяти.

Описываемые программные средства позволяют начертить изолинию с бергштрихами, провести ее как сплошную, штриховую или штрихпунктирную линию, вписать в разрыв линии номер или значение уровня. Для задания режимов рисования можно воспользоваться программами FULL, BROKEN, NUMLIN, SIZLIN (см. п. 4.2, пп. 7.2.1, 7.2.2) и SETBE. При этом по умолчанию вычерчиваются сплошные линии без надписей и бергштрихов.

Заметим, что алгоритм построения линий уровня устроен таким образом, что изолинии, образованные пересечением секущей плоскости с выпуклыми участками поверхности, отслеживаются в направлении против часовой стрелки, а с вогнутыми - по часовой стрелке. Отсюда вытекает и способ задания бергштрихов как векторного произведения единичного базисного вектора \mathbf{k} на отрезок-вектор $\{(XI(L) - XI(L-1)), (YI(L) - YI(L-1))\}$.

Данные программы построения линий уровня могут быть использованы не только в случае, когда исходная поверхность задана в декартовой системе координат. Допускается также обработка поверхностей, определенных в произвольной системе координат, которая однозначно отображается на декартову (например, в цилиндрической, сферической и др.). Для этого с помощью программы SETRA устанавливается режим преобразования, в результате каждая изолиния перед выводом будет преобразована указанным образом (с помощью служебной программы TRANSF).

Как обычно, при рисовании в области необходимо задавать пределы изменения функции и аргумента в этой области (LIMITS). Определить пределы в случае недекартовой системы координат удобно с помощью программы BOUND.

7.3.2. Описание программ построения карт изолиний.

Программа IZOLIN(Z,M,N,X,Y,ZIZ,L,XI,YI,NL) позволяет начертить изолинии функции двух переменных во всей области ее задания. Параметры программы следующие:

Z - массив значений функции (размером (M, N));

M, N - число точек сетки по осям X и Y;

X, Y - массивы координат сетки по осям X и Y;

ZIZ - массив значений уровней;

L - размер массива уровней;

XI, YI - буферные массивы X- и Y-координат изолинии;

NL - размер буферных массивов.

Программа IZFLIN(Z,M,N,X,Y,ZIZ,L,NX,NY,KX,KY,NF,XI,YI,NL) позволяет начертить изолинии функции двух переменных в указанных подобластях области ее задания. Параметры NX, NY, KX, KY (массивы индексов сетки по соответствующим осям) определяют нижние левые и верхние правые углы прямоугольных подобластей; NF - размер массивов (число подобластей). Остальные параметры такие же, как и в программе IZOLIN.

Программа SETBE(LSTEP,SIZEB) устанавливает режимы вычерчивания изолинии с бергштрихами. Ее параметры:

LSTEP - шаг установки бергштриха (метится каждая LSTEP-я точка, начиная с первой);

|SIZEB| - размер бергштриха: SIZEB > 0 - размер бергштриха в выбранных единицах измерения, SIZEB < 0 - размер бергштриха в рабочих единицах (математическое значение).

Программа SETRA(L) устанавливает признак системы координат: L = 0 - декартова система координат, L > 0 - недекартова система координат.

Программа BOUND(Z,M,N,X,Y,NX,NY,KX,KY,NF,XMI,XMA,YMI,YMA) позволяет определить пределы изменения аргументов функции на области ее задания в произвольной системе координат. Параметры программы следующие:

Z - массив значений функции (размером (M,N));

M, N - число точек сетки по осям X и Y;

X, Y - координаты сетки по осям X и Y;

NX, NY, KX, KY - массивы индексов сетки по осям X, Y, их элементы определяют нижние левые и верхние правые углы прямоугольных подобластей (длины NF);

NF - число подобластей;

XMI, XMA - минимальное и максимальное значения по оси X;

YMI, YMA - минимальное и максимальное значение по оси Y;

Для выполнения преобразований произвольной системы координат в декартову необходимы функции ATDX и ATDY.

Функция $ATDX(Z,X,Y)$ вычисляет координату X в декартовой системе координат по координатам точки в исходной системе координат.

Функция $ATDY(Z,X,Y)$ вычисляет координату Y в декартовой системе координат по координатам точки в исходной системе координат.

Базовой системой координат выбрана декартова. Для того чтобы можно было работать с другими системами координат, необходимо заменить функции $ATDX$ и $ATDY$ на другие, выполняющие желаемые преобразования.

```
FUNCTION ATDX(Z,X,Y)    FUNCTION ATDY(Z,X,Y)
ATDX=Φ(Z,X,Y)          ATDY=Φ(Z,X,Y)
RETURN                  RETURN
END                      END
```

Так, для цилиндрической системы координат вместо $Φ(Z,X,Y)$ следует воспользоваться функциями $ATDX1$ и $ATDY1$, для сферической - это функции $ATDX2$ и $ATDY2$. Они имеются в библиотеке Графора. В случае использования других систем координат пользователю предоставляется возможность написать соответствующие программы-функции самостоятельно. Например, функции $ATDX1$ и $ATDX2$ имеют вид:

```
FUNCTION ATDX1(Z,X,Y)  FUNCTION ATDX2(Z,X,Y)
ATDX1=X * COS(Y)      ATDX2=Z * SIN(Y) * COS(X)
RETURN                 RETURN
END                     END
```

Служебные программы. Программы $IZOLIN$ и $IZFLIN$ используют при работе подпрограммы $IZLIN$, $RAISE$, $LOWER$, $RECUR$, $LETIZO$. Непосредственный вывод вычисленной изолинии осуществляет программа $LETIZO$. В качестве параметров ей передаются массивы координат изолинии и их размеры.

Программа $IZLIN(Z,M,N,X,Y,ZMI,ZIZ,NUM,NXT,NYT,KXT,KYT,XI,YI,NL)$ позволяет начертить изолинии функции двух переменных в выделенной подобласти задания функции, если в этой подобласти функция положительна. Параметры программы:

ZMI - величина "поднятия" поверхности;

ZIZ - значение текущего уровня с номером NUM ;

NXT, NYT - индексы сетки по осям X и Y , определяющие нижний левый угол прямоугольной подобласти;

KXT, KYT - индексы сетки по осям X и Y , определяющие верхний правый угол прямоугольной подобласти;

Остальные параметры те же, что и в программе $IZOLIN$.

Программы $RAISE$, $LOWER$, $RECUR$ являются программами обработки поверхности.

Программа $RAISE(Z,M,N,ZMI)$ определяет величину ZMI , на которую надо "поднять" поверхность, и "поднимает" ее в положительное полупространство.

Программа $LOWER(Z,M,N,ZMI)$ позволяет "опустить" ($ZMI > 0$) или "поднять" ($ZMI < 0$) поверхность на заданную величину ZMI .

Программа $RECUR(Z,M,N,NX,NY,KX,KY)$ заменяет все отрицательные значения функции $Z(I, J)$ на $|Z(I, J)|$ в заданной подобласти. Параметры этой программы задают массив значений функции, размерность массива, индексы сетки по осям, определяющие нижний левый и правый верхний углы прямоугольной подобласти.

Функция $XILY(Z1,Z2,ARG1,ARG2,ZIZOL,I,J,IUS,JUS)$ производит линейную интерполяцию на ребре с крайними точками $(ARG1,Z1)$, $(ARG2,Z2)$ и регистрирует факт прохождения изолинии через ребро. Здесь: $ZIZOL$ - величина уровня (с учетом "поднятия" поверхности); I, J - индексы внешнего (по отношению к изолинии) узла; IUS, JUS - приращения индексов сетки (по осям X и Y соответственно), определяющих внутренний по отношению к изолинии узел. (Внутренним по отношению к изолинии считается узел, в котором значение функции больше величины рассматриваемого уровня. Аналогично, узел, значение функции в котором меньше величины уровня, считается внешним.)

Для вывода изолинии по вычисленным значениям координат используется программа $LETIZO$.

Программа LETIZO(XI,YI,IL,ZIZ,NIZ,KIND) вычерчивает линию по заданным массивам координат XI, YI длины IL и вписывает в разрыв значение ZIZ или номер NIZ уровня. Параметр KIND определяет тип линии (0 - незамкнутая, 1 - замкнутая).

```
SUBROUTINE LETIZO(XI,YI,ILM,ZIZ,NIZ,KIND)
DIMENSION XI(ILM),YI(ILM)
COMMON /GFBET/ KT,ISTEP,SIZE
COMMON /GFIZOL/ NC,C
NC=NIZ
C=ZIZ
IF(KIND.EQ.0) GO TO 1
XI(ILM)=XI(1)
YI(ILM)=YI(1)
1 CONTINUE
IF(KT.NE.0) CALL TRANSF(ZIZ,XI,YI,ILM)
IF(ISTEP.GT.0) CALL MARKBE(XI,YI,ILM)
CALL LINNUM(XI,YI,ILM)
RETURN
END
```

При работе этой программы используются программы LINNUM (см. п. 4.2), MARKBE и TRANSF.

Программа MARKBE(XI,YI,IL) устанавливает бергштрихи на изолинии. Ее параметры: XI, YI - массивы координат изолинии, IL - размер массивов.

Программа TRANSF(ZIZ,XI,YI,ILM) переводит массивы изолиний из произвольной системы координат в декартову в соответствии с заданными функциями преобразования. Ее параметры:

ZIZ - величина строящегося уровня;

XI, YI - массивы X- и Y-координат изолинии;

ILM - размер массивов изолиний.

7.3.3. Вспомогательные программы. Приведем описание нескольких программ, которые могут быть полезны при оформлении рисунка.

Программа EXUDE(Z,M,N,X,Y,NX,NY,KX,KY,NF,MRKA,MRKI,KD,H,TH) позволяет найти, пометить и надписать локальные экстремумы функции. Экстремальным считается тот узел, значение функции в котором больше (меньше) ее значений в смежных узлах. Параметры программы следующие:

Z - массив значений функции (размером (M, N));

M, N - число точек сетки по осям X и Y;

X, Y - массивы значений сетки по осям X и Y;

NX, NY - массивы индексов сетки по осям X и Y, определяющих нижние левые углы прямоугольников, в которых необходимо искать экстремумы функции;

KX, KY - массивы индексов сетки по осям X и Y, определяющих верхние правые углы прямоугольников, в которых необходимо искать экстремумы функции;

NF - количество прямоугольников;

MRKA - номер маркера для максимума;

MRKI - номер маркера для минимума;

KD - количество дробных знаков;

H - высота цифры в заданных единицах измерения;

TH - угол наклона текста (в градусах).

Программа TOKEN(XS,YS,MRK,FZ,KD,H,TH) помечает точку заданным маркером и надписывает заданное значение. Параметры программы:

XS, YS - декартовы математические координаты точки;

MRK - номер маркера;

FZ - надписываемое число;

KD - количество дробных знаков;

H - высота цифры в заданных единицах измерения;

TH - угол наклона текста (в градусах).

Программа EXMIMA(Z,M,N,ZMI,ZMA) позволяет определить максимальное и минимальное значения в двумерном массиве чисел. Ее параметры: Z - двумерный массив размером (M, N), ZMI и ZMA - минимальное и максимальное значения.

7.3.4. Построение пространственных кривых. Пусть имеются две поверхности A и B, заданные на одной и той же прямоугольной неравномерной сетке. Если $P = A - B$, то $P = 0$ в точках, где A и B совпадают. Построив изолинии нулевого уровня от P, получим координаты (XI, YI) проекции пространственных линий пересечения A и B на области задания. По этим координатам вычисляются значения Z пространственной линии L:

$$L(XI, YI) = A(XI, YI) = B(XI, YI).$$

Для реализации этой возможности имеется вариант программы-функции XILY - XILY1 и программы LETIZO - LETSPL. Пользователь должен обеспечить необходимую замену, т. е. поместить в пакет:

```
FUNCTION XILY(Z1,Z2,ARG1,ARG2,ZIZOL,I,J,IUS,JUS)
XILY=XILY1(Z1,Z2,ARG1,ARG2,ZIZOL,I,J,IUS,JUS)
RETURN
END
SUBROUTINE LETIZO(XI,YI,IL,ZIZ,NIZ,KIND)
DIMENSION XI(IL),YI(IL)
CALL LETSPL(XI,YI,IL,ZIZ,NIZ,KIND)
RETURN
END
```

Кроме того, необходимо в главной программе задать два общих блока:

```
COMMON /GFNAM1/M,B
COMMON /GFNAM2/KEY,PL
```

где B - массив размером (M,N), описывающий одну из поверхностей, M - количество столбцов массива, KEY - текущий индекс заполнения массивов изолиний пространственной кривой (перед обращением к программе IZOLIN должен быть установлен равным 1), PL - одномерный массив под пространственную линию (того же размера, что и буферный массив под изолинию).

Пример приведен в следующем пункте.

Следует заметить, что при рисовании пространственная линия не замыкается.

7.3.5. Примеры. В этом разделе приведены примеры, показывающие различные возможности программ IZOLIN и IZFLIN.

1. Рис. 7.8 и 7.9 иллюстрируют работу программы IZFLIN. На первом рисунке изолинии построены во всей области задания функции, на втором - в двух прямоугольных подобластях. Пределы изменения функции определены на всей области ее задания.

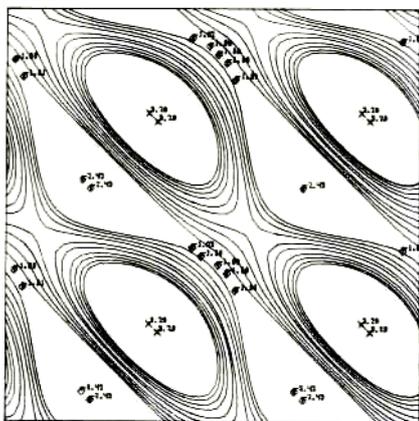


Рис. 7.8. Карта изолиний, построенная на всей области задания функции

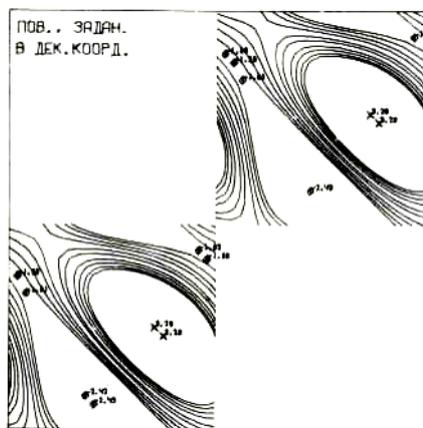


Рис. 7.9. Карта изолиний, построенная в двух прямоугольных подобластях

```
DIMENSION Z(50,50),RO(50),FI(50),XI(200),YI(200),ZIZ(10)
DIMENSION NX(10),KX(10),NY(10),KY(10),X(50),Y(50)
EQUIVALENCE (X(1),RO(1)), (Y(1),FI(1))
DATA ZIZ /0.1,0.4,0.6,0.8,0.95,-1,-4,-6,-8,-95/
```

```

DATA NLINE/200/
DATA NF/2/,NX,NY,KX,KY/1,25,8*0,1,25,8*0,25,50,8*0,25,50,8*0/,M/50/,N/50/,L/10/
H=12.56/50
DO 1 I=1,50
X(I)=H*(I-1)
Y(I)=X(I)
1 CONTINUE
DO 5 I=1,50
DO 5 J=1,50
Z(I,J)=(SIN(X(I)+Y(J))+COS(X(I)+Y(J))-(SIN(X(I))+COS(Y(J))))
5 CONTINUE
CALL PAGE(17.,17.,'ПОВ., ЗАДАН. В ДЕК.КООРД',24,0)
CALL REGION(1.,1.,15.,15.,0,0,1)
CALL BOUND(Z,M,N,RO,FI,NX,NY,KX,KY,NF,XMI,XMA,YMI,YMA)
CALL LIMITS(XMI,XMA,YMI,YMA)
CALL IZFLIN(Z,M,N,RO,FI,ZIZ,L,NX,NY,KX,KY,NF,XI,YI,NLINE)
CALL EXUDE(Z,M,N,RO,FI,NX,NY,KX,KY,NF,2,3,2,0.23,0.)
CALL ENDPG('7.9')

```

2. На рис. 7.10 приведен пример построения изолиний с бергштрихами.

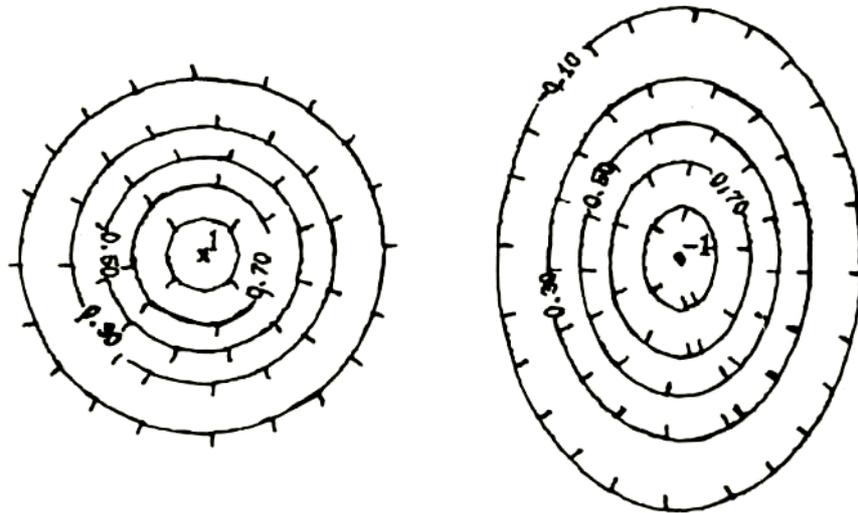


Рис. 7.10. Пример построения изолиний с бергштрихами

```

DIMENSION Z(50,50),X(50),Y(50),XI(200),YI(200),ZIZ(10)
DATA M,N,L,NLINE /50,50,10,200/
X(1)=0.
Y(1)=0.
H=0.20
DO 10 I=1,50
X(I)=H-FLOAT(I)
Y(I)=X(I)
10 CONTINUE
DO 20 I=1,50
DO 30 J=1,50
Z(I,J)=EXP(-(X(I)-3.) ** 2+(Y(J)-5.) ** 2))- EXP(-((X(I)-7.) ** 2+0.5*(Y(J)-5.) ** 2))
30 CONTINUE
20 CONTINUE
DO 40 I=1,10
ZIZ(I)=-1.1+0.200*FLOAT(I)
40 CONTINUE
CALL PAGE(17.,17.,'BERG',4,0)
CALL REGION(1.,1.,15.,15.,'SIZEB=0.15, LSTEP=3',19,0)

```

```

CALL MINMAX(X,M,XMI,XMA)
CALL MINMAX(Y,N,YMI,YMA)
CALL LIMITS(XMI,XMA,YMI,YMA)
CALL SIZLIN(0.23,4,2)
CALL SETBE(3,0.15)
CALL IZOLIN(Z,M,N,X,Y,ZIZ,L,XI,YI,NLINE)
CALL EXUDE(Z,M,N,X,Y,1,1,M,N,1,-2,-3,0,0.3,0.)
CALL ENDPG('7.10')

```

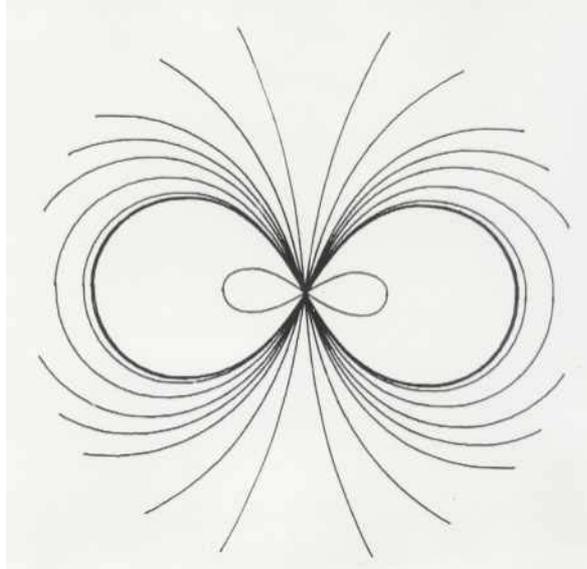


Рис. 7.11. Линии уровня в цилиндрических координатах

3. Линии уровня в цилиндрических координатах показаны на рис. 7.11.

```

DIMENSION Z(50,50),RO(50),FI(50),XI(200),YI(200),ZIZ(10)
DIMENSION NX(10),KX(10),NY(10),KY(10)
DATA ZIZ /0.1,0.4,0.8,1.,1.2,1.5,1.8,1.9,1.93,5./
DATA NF/1/,NX,NY,KX,KY/1,9*0,1,9*0,50,9*0,50,9*0/
DATA M,N,L,NLINE /50,50,10,200/
HRO=4.2/49.
HFI=3.14/49.
DO 1 I=1,50
RO(I)=HRO * (I-1)-2.1
HFI=HFI * (I-1)
1 CONTINUE
DO 5 I=1,50
DO 5 J=1,50
Z(I,J)=7. * (COS(FI(J))) ** 2/(RO(I) ** 2+1.)
5 CONTINUE
CALL PAGE(17.,17.,'ПОВ., ЗАДАН. В ЦИЛ.КООРД',24,0)
CALL REGION(1.,1.,15.,15.,0,0,1)
CALL SETRA(1)
CALL BOUND(Z,M,N,RO,FI,NX,NY,KX,KY,NF,XMI,XMA,YMI,YMA)
CALL LIMITS(XMI,XMA,YMI,YMA)
CALL IZOLIN(Z,M,N,RO,FI,ZIZ,L,XI,YI,NLINE)
CALL ENDPG('7.11')
END
FUNCTION ATDX(Z,X,Y)
ATDX=ATDX1(Z,X,Y)
RETURN
END

```

```

FUNCTION ATDY(Z,X,Y)
ATDY=ATDY1(Z,X,Y)
RETURN
END

```

4. На рис. 7.12 приведен пример построения пространственной кривой, образованной пересечением поверхностей (полусфера рассекается наклонной плоскостью).

```

SUBROUTINE PRLINE
COMMON /GFNAM1/ M,B /GFNAM2/ KEY,PL
DIMENSION A(50,50),B(50,50),X(50),Y(50)
DIMENSION XI(200),YI(200),PL(200)
DATA N,H/50,0.084/
KEY=1
M=50
X(1)=-2.1
Y(1)=-2.1
DO 10 I=2,N
X(I)=X(I-1)+H
10 Y(I)=X(I)
DO 20 I=1,N
DO 20 J=1,N
DDD=X(I) ** 2+Y(J) ** 2-4.
A(I,J)=0.
GO TO 5
IF(DDD).LT.0) A(I,J)=SQRT(-DDD)
20 B(I,J)=(X(I)+Y(J))/2.
CALL PAGE(16.,16.,0,0,0)
CALL REGION(1.,1.,14.,14.,'F1=X ** 2+Y ** 2-4 * * * F2=(X+Y)/2',29,1)
DO 30 I=1,N
DO 30 J=1,N
30 A(I,J)=A(I,J)-B(I,J)
CALL INIT
CALL ISOMET
CALL TDLIM(X,Y,A,N,N,1,N,1,N,S)
ZIZ=0
CALL IZOLIN(A,N,N,X,Y,ZIZ,1,XI,YI,200)
CALL ENDPG('7.12')
RETURN
END
SUBROUTINE LETIZO(XI,YI,ILM,ZIZ,NIZ,KIND)
DIMENSION XI(ILM),YI(ILM)
CALL LETSPL(XI,YI,ILM,ZIZ,NIZ,KIND)
RETURN
END
FUNCTION XILYI(Z1,Z2,ARG1,ARG2,ZIZOL,I,J,IUS,JUS)
XILYI=XILYI1(Z1,Z2,ARG1,ARG2,ZIZOL,I,J,IUS,JUS)
RETURN
END

```

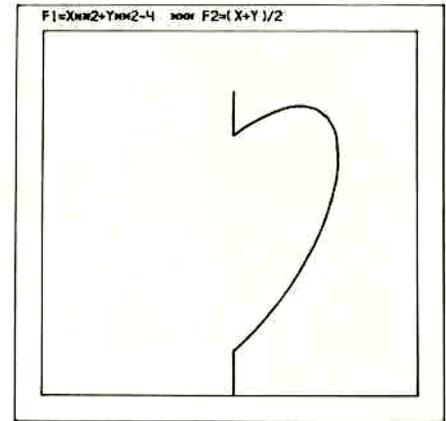


Рис. 7.12. Изображение пространственной кривой, образованной пересечением поверхностей

Примечание. Изображение пространственной кривой в этом примере производится с использованием средств, описанных в п. 8.1.