

8. ПОСТРОЕНИЕ ПЛОСКИХ ИЗОБРАЖЕНИЙ ТРЕХМЕРНЫХ ОБЪЕКТОВ

В этой главе описаны программные средства, позволяющие строить плоские проекции пространственных объектов. Для достижения большей реалистичности и наглядности изображения те части образа объекта, которые невидимы наблюдателю, стираются.

В Графоре реализовано несколько различных методов *удаления невидимых линий*, однако большинство из них предназначено для изображения графических объектов, представляющих собой однозначные функции двух переменных, хотя они могут использоваться и в некоторых более общих случаях.

С помощью программ, изложенных в 8.1 и 8.2 (см. также [1, г, 1, ж]), можно строить проекции поверхностей, описываемых однозначными непрерывными функциями двух переменных, заданными на прямоугольной области. В основе первого метода - будем называть его *методом слоев* - лежит способ параллельных сечений. Секущие плоскости покрывают поверхность семейством кривых, которые затем проецируются на картинную плоскость. В зависимости от расположения секущих плоскостей в пространстве поверхность может быть покрыта сечениями, параллельными плоскостям XZ или YZ , или же она покрывается одновременно двумя этими семействами сечений, образующими криволинейную сетку (совокупность криволинейных четырехугольников). С помощью этого метода можно построить произвольные аксонометрические проекции, а также центральные проекции на плоскость, перпендикулярную *лучу зрения* (отрезку, соединяющему центр проекции с началом координат).

В другом алгоритме, построенном на основе *метода ячеек*, элементарными единицами являются участки поверхности, соответствующие лишь одной ячейке криволинейной сетки. Поэтому изображения, полученные с использованием этого алгоритма, обладают несколько более высоким качеством, чем проекции, построенные с помощью метода слоев. В алгоритме, реализующем метод ячеек, кроме того, применяется более простой способ проецирования по сравнению с традиционными подходами (см. [1, г, 1, к, 1, л]), основанными на аппарате однородных координат. Однако с помощью второго метода можно получить лишь центральные проекции поверхностей.

В 8.3 описаны программные средства, позволяющие получать практически произвольные (лишь с небольшими ограничениями) центральные проекции поверхностей (см. также [1, к]). Они позволяют "строить изображения функций, которые заданы в произвольно расположенных точках. Более того, область определения может быть неодносвязной. Поверхности аппроксимируются с помощью треугольной сетки, формируемой по заданной границе области и узлам внутри нее с помощью *триангуляции*.

Алгоритмы удаления невидимых линий, применяемые в трех упомянутых случаях, во многом схожи. В основе их лежит упорядочение соответствующих элементарных единиц поверхности, будь то сечения параллельными плоскостями или треугольные, или четырехугольные элементы - "кусочки". При этом выбирается такой порядок построения элементарных единиц поверхности (назовем их элементами), при котором каждый элемент поверхности может быть закрыт только ранее построенными участками поверхности.

Такой упорядоченный перебор элементов поверхности позволяет ввести *экран*, определяемый как некоторый многоугольник на плоскости рисунка. При проецировании очередного элемента изображаются только те его участки, которые выходят за границы экрана, и экран соответствующим образом модифицируется. Практически, экран отображает на плоскости рисунка проекцию участка поверхности, который соответствует построенным к данному моменту элементам поверхности. Для экономного задания экрана на картинной плоскости вводится *физическая сетка*, и экран определяется с точностью до этой сетки.

В 8.4 описан еще один способ построения проекций трехмерных объектов (см. также [1, л]). Для создания эффекта глубины в нем используется *метод ореола*, когда дальние, невидимые отрезки стираются не полностью, но в них вносятся разрывы, которые вызваны ореолами вокруг более близких к наблюдателю отрезков.

Основное отличие метода ореола от других способов удаления невидимых линий состоит в том, что он позволяет работать с неструктурированной графической информацией, т. е. он предназначен для изображения объектов, представленных в виде неупорядоченной совокупности отрезков прямых.

С точки зрения быстродействия первый и второй методы обладают преимуществом по сравнению с третьим и четвертым, поскольку время их работы линейно зависит от числа изображаемых отрезков, тогда как у третьего и четвертого наблюдается зависимость более высокого порядка.

8.1. Изображение функций двух переменных. Метод параллельных сечений

С помощью программ, описанных в этом параграфе, можно строить проекции поверхностей, заданных однозначной непрерывной функцией двух переменных, с удалением невидимых линий. Функция должна быть определена на прямоугольной сетке (не обязательно равномерной). Для

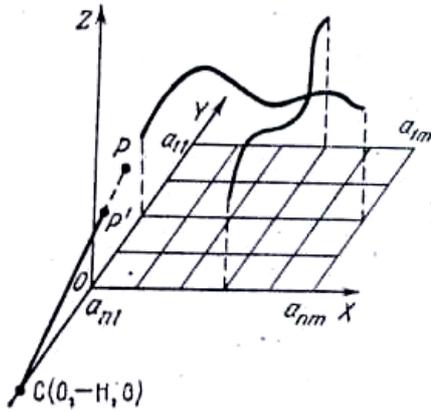


Рис. 8.1 Декартова система координат, проекция P' точки P на плоскость XZ , сетка, на которой задана поверхность, и сечения, параллельные плоскостям XZ и YZ .

аффинных преобразований пространственных объектов и для их проецирования на картинную плоскость используется аппарат однородных координат. С помощью этого метода можно получать как аксонометрические, так и центральные проекции на плоскость перпендикулярную "линии визирования", проведенной из центра проекции в начало координат. Для графического представления функции используется способ параллельных сечений. Упорядочение сечений лежит в основе алгоритма удаления невидимых линий. Имеются также средства, позволяющие строить изображения пространственных кривых.

8.1.1. Преобразование координат в трехмерном пространстве.

В основе программ аффинных преобразований пространственных объектов, а также их проецирования на картинную плоскость лежит аппарат однородных координат (см., например, [15, 16]). При этом все необходимые для построения проекции и установления нужного ракурса преобразования координат описываются матрицами размером 4×4 и представляются в виде

суперпозиции некоторых основных преобразований: переноса точки в пространстве на фиксированный вектор, поворота вокруг указанной оси на заданный угол, масштабирования вдоль какой-либо оси, сдвига, перспективы или проецирования на одну из главных координатных плоскостей.

Основные преобразования координат. Рассмотрим некоторую декартову систему координат (рис. 8.1). Любая точка пространства представляется в ней вектор-матрицей вида $(x \ y \ z \ 1)$. Мы будем пользоваться однородными координатами точки в пространстве $(x \ y \ z \ 1)$.

В качестве *картинной плоскости* выберем плоскость XZ , описываемую уравнением $Y = 0$. Проекция точки объекта на эту плоскость получается в результате умножения $(x \ y \ z \ 1) A$, где

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

задает преобразование проецирования на плоскость XZ .

Поворот вокруг заданной оси (X , Y и Z соответственно) на указанный угол α описывается следующими матрицами:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & b & a & 0 \\ 0 & -a & b & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_y(\alpha) = \begin{bmatrix} b & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & b & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_z(\alpha) = \begin{bmatrix} b & a & 0 & 0 \\ -a & b & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

где $a = \sin \alpha$, $b = \cos \alpha$. Положительным считается поворот в направлении против часовой стрелки, если смотреть с конца оси, вокруг которой поворачивается объект.

Матрицы преобразований переноса на фиксированный вектор и масштабирования имеют следующий вид:

$$TR = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}, \quad S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & s \end{bmatrix}.$$

Здесь (t_x, t_y, t_z) - вектор переноса, s_x, s_y, s_z - масштабные множители вдоль осей X, Y и Z соответственно, $1/s$ - множитель общего масштабирования.

Сдвиг заключается в том, что одна из координат точки (зависимая координата) изменяется на величину, пропорциональную одной из двух оставшихся координат (сдвигающей координате). Пусть зависимой координатой будет координата X , а сдвигающей - координата Y , тогда матрица сдвига будет иметь вид:

$$SH = \begin{bmatrix} 1 & 0 & 0 & 0 \\ F & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

где F - коэффициент сдвига. Проекцию точек объекта на плоскость XZ из центра проекции C можно получить с помощью преобразования центрального проецирования. Его матрица:

$$P(H) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/H \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Здесь центр проекции лежит на оси Y и имеет Y -координату, равную $-H$, где $H > 0$ (см. рис. 8.1).

С помощью основных преобразований координат можно получить практически произвольные плоские геометрические проекции.

Рассмотрим сначала случай параллельного проецирования. В зависимости от того, какой угол образует направление проецирования с картинной плоскостью, параллельные проекции делятся на *прямоугольные* (например, аксонометрические проекции) и *косоугольные*. В случае прямоугольных проекций направление проецирования перпендикулярно картинной плоскости. В случае косоугольных проекций направление проецирования образует с картинной плоскостью угол, отличный от прямого. Более подробные сведения об этих типах проекций можно найти, например, в [16].

Более общие аксонометрические проекции можно получить с помощью двух последовательных поворотов объекта (сначала вокруг оси Z на некоторый угол Az , а потом вокруг оси X на угол Ax) и затем ортогонального проецирования на плоскость XZ . Для двух наиболее распространенных типов аксонометрических проекций - изометрии и диметрии - углы поворота имеют следующие значения: $Az = -45^\circ, Ax = 35^\circ$ и $Az = -20^\circ, Ax = 20^\circ$.

На рис. 8.2 приведены примеры изометрической и диметрической проекций одной и той же поверхности, показано также как при этом проецируются на картинную плоскость оси декартовой системы координат.

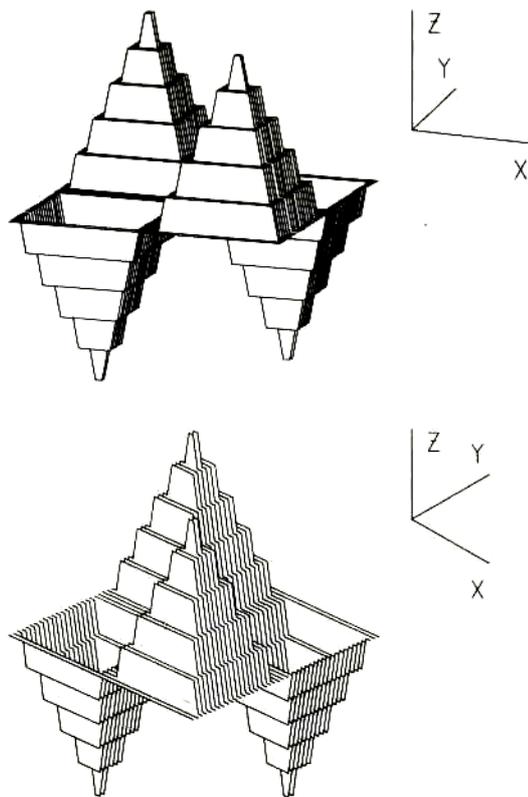


Рис. 8.2. Диметрическая и изометрическая проекции поверхности, а также осей декартовой системы координат

Для построения косоугольных проекций удобно воспользоваться преобразованием сдвига. Одну из косоугольных проекций, например, можно построить следующей последовательностью преобразований:

- 1) сдвиг, в котором зависимой осью является ось X , сдвигающей осью - ось Y ; коэффициент сдвига $F = 1$ в случае, если задана "положительная" проекция (рис. 8.3, б), и $F = -1$, если требуется "отрицательная" проекция (рис. 8.3, а);
- 2) сдвиг, в котором зависимой является ось Z , сдвигающей - ось Y и коэффициент сдвига $F = 1$;
- 3) проецирование на плоскость XZ .

В случае, если ни одна из упомянутых стандартных параллельных проекций (изометрия, диметрия и косоугольная проекция) по каким-либо причинам не устраивает, можно построить требуемую проекцию с помощью переноса, поворота, масштабирования и сдвига.

Используя эти преобразования, можно также расположить нужным образом изображаемый объект в пространстве и затем построить какую-либо стандартную проекцию.

С помощью основных преобразований координат легко также формируется преобразование, которое позволит получать центральную проекцию объекта из произвольного центра проекции на плоскость, проходящую через начало координат перпендикулярно лучу зрения. Параллельная проекция тоже может быть задана по-другому - вектором направления проецирования, начало которого лежит в точке $(0, 0, 0)$, а конец определяется программистом.

Таким образом, для построения произвольной проекции графического объекта достаточно сформировать матрицу преобразования, являющегося суперпозицией перечисленных выше основных преобразований координат. Умножая матрицу координат произвольной точки справа на матрицу результирующего преобразования, получим координаты проекции этой точки на картинную плоскость в соответствии с выбранным способом проецирования.

Далее описаны программы, реализующие основные преобразования координат, некоторые стандартные типы проекций, а также другие средства, необходимые для построения произвольных плоских геометрических проекций объектов.

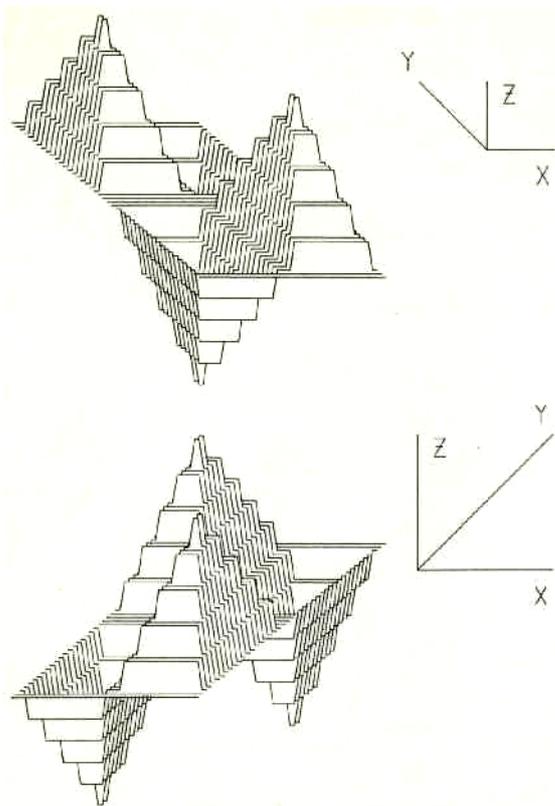


Рис. 8.3. "Отрицательная" (а) и "положительная" (б) косоугольные проекции поверхности, а также осей декартовой системы координат

Программы преобразований. Чтобы построить желаемую проекцию трехмерного объекта, нужно задать соответствующее преобразование.

Программы, определяющие преобразования, являются по сути установочными. Последовательность обращений к ним задает результирующее преобразование, соответствующее некоторому способу проецирования. Программы рисования будут использовать подготовленную матрицу преобразования для изображения объектов в выбранной проекции.

Каждая из программ, устанавливающих свое преобразование, формирует матрицу размером 4×4 и умножает ее слева на матрицу текущего преобразования. В результате преобразования будут выполняться в том порядке, в котором они задавались. Начальные установки выполняет программа INIT, которая формирует единичную матрицу. Обращение к ней отменяет уже накопленное преобразование. Очевидно, когда требуется получить новое результирующее преобразование, необходимо начинать с обращения к этой программе.

Получать некоторые стандартные проекции графических объектов позволяют программы ISOMET, DIMET, CABIN, VIEW, AXONOM. Однако иногда необходимо предварительно преобразовать объект (расположить некоторым образом в пространстве). Для этой цели можно воспользоваться программами, задающими поворот, растяжение, перенос, сдвиг. Это программы: TDROT, TDSCAL, TDTRAN, SHEAR.

Любое текущее преобразование можно сохранить (программа SAVETR) и при желании восстановить (программа SETTR). Вообще с помощью программы SETTR можно установить в качестве текущего преобразования произвольное преобразование, расширив тем самым круг основных преобразований координат.

Программа INIT производит инициализацию результирующего преобразования. Программа без параметров.

Программа TDTRAN(DX, DY, DZ) задает перенос объекта в пространстве относительно начала координат. Параметры программы DX, DY, DZ определяют вектор переноса.

Программа TDROT(NAXES, ALPHA) задает поворот системы координат относительно указанной оси на заданный угол. Ее параметры:

NAXES - номер оси, относительно которой выполняется поворот:

NAXES = 1 - ось X, NAXES = 2 - ось Y, NAXES = 3 - ось Z (кроме того, если NAXES < 0, угол поворота считается заданным в радианах, NAXES > 0 - в градусах);

ALPHA - угол поворота;

$ALPHA > 0$ - поворот выполняется против часовой стрелки, относительно оси, вокруг которой выполняется поворот;

$ALPHA < 0$ - поворот выполняется по часовой стрелке.

Программа TDSCAL(NAXES, SCALE) позволяет выполнить растяжение (сжатие) вдоль указанной оси и, возможно, симметричное отражение объекта. Параметры программы следующие:

NAXES - номер оси, вдоль которой выполняется растяжение (сжатие): NAXES = 1 - ось X, NAXES = 2 - ось Y, NAXES = 3 - ось Z; NAXES = 4 - растяжение (сжатие) по всем осям;

SCALE - коэффициент растяжения (сжатия):

$SCALE \geq 1$ - растяжение в SCALE раз,

$0 < SCALE < 1$ - сжатие в $1/SCALE$ раз,

$SCALE < 0$ - симметричное отражение относительно соответствующей координатной плоскости или начала координат и растяжение в $|SCALE|$ раз или сжатие в $1/|SCALE|$ раз.

Программа SHEAR(I, J, F) определяет сдвиг. Параметры программы:

I - номер сдвигающей координаты: I = 1 - координата X, I = 2 - координата Y, I = 3 - координата Z;

J - номер зависимой координаты;

F - коэффициент сдвига.

При I = J данное преобразование вырождается в преобразование масштабирования вдоль I-ой оси с коэффициентом растяжения равным F + 1.

Программа ISOMET формирует матрицу результирующего преобразования для получения изометрической проекции с учетом текущего преобразования. Программа без параметров.

Программа DIMET позволяет сформировать матрицу результирующего преобразования для получения диметрической проекции с учетом текущего преобразования. Программа без параметров.

Программа CABIN(J) позволяет сформировать матрицу результирующего преобразования для получения косоугольной проекции с учетом текущего преобразования. Параметр программы J определяет вид косоугольной проекции. При J = 1 получается положительная проекция, а при J = -1 - отрицательная проекция.

Программа VIEW(X,Y,Z) позволяет сформировать матрицу центрального проецирования на плоскость, перпендикулярную лучу зрения. Параметры программы:

X, Y, Z - координаты центра проекции (точки зрения).

Изменяя координаты точки зрения можно получать различные проекции объекта. Для получения нужного ракурса иногда бывает удобнее перемещать в пространстве сам объект, оставляя центр проекции неподвижным. Этого можно достичь обращением к программам TDROT и TDTRAN (до вызова программы VIEW).

При обращении к программе VIEW надо следить, чтобы центр проекции не оказался внутри изображаемого объекта, иначе результаты работы программы рисования THREEED будут непредсказуемы.

Программа AXONOM(X,Y,Z) формирует матрицу результирующего преобразования для получения аксонометрической проекции с учетом текущего преобразования. Направление проецирования определяется вектором, соединяющим точку (X, Y, Z) с началом координат.

Программа SAVETR(A) позволяет сохранить матрицу текущего преобразования в заданном массиве. Параметр программы: A - одномерный массив длины 16.

Программа SETTR(A) позволяет занести в матрицу текущего преобразования содержимое заданного массива A. Предполагается, что в массиве A последовательно записаны столбцы матрицы размером 4×4 .

Вспомогательные и служебные программы.

Программа HCUNIT(A) формирует единичную матрицу A размером 4×4 .

Программа HCMULT(A, B) перемножает две квадратные матрицы четвертого порядка $A \times B$. Результат помещается на место матрицы A.

Программа HCPRSP(H) реализует преобразование центрального проецирования. Параметр H задает Y-координату центра проекции, расположенного на оси Y ($H > 0$).

Программа HGINV(X,Y,Z, XP,YP,ZP) вычисляет координаты (XP,YP,ZP) центра проекции с учетом обратного преобразования координат. Предварительно вычисляется матрица обратного преобразования.

Программа HCROT1(X,Y,Z) позволяет найти результирующее преобразование, переводящее двумя последовательными поворотами точку A(X, Y, Z) в точку с координатами $(0, -\sqrt{X^2 + Y^2 + Z^2}, 0)$.

8.1.2. Построение проекций поверхностей. Удаление невидимых линий. Класс трехмерных объектов, проекции которых могут быть изображены с помощью предлагаемых ниже программ, ограничивается поверхностями, заданными однозначной и непрерывной функцией двух переменных $z = f(x,y)$. Поверхность определяется следующим образом. На плоскости XY вводится прямоугольная сетка в общем случае с неравномерным шагом по осям X и Y. В узлах этой сетки задаются значения функции. Порядок расположения элементов сетки показан на рис. 8.1. Он отличается от расположения, принятого в методе ячеек, который описан в 8.2 (см. рис. 8.9).

Для построения проекций определенной таким образом поверхности используется метод сечений. Поверхность рассекается рядом параллельных плоскостей. Кривые, образованные пересечением поверхности с секущими плоскостями, затем проецируются на картинную плоскость в соответствии с выбранным способом проецирования. Разработанные программы позволяют изобразить поверхность одним из следующих способов: либо сечения, покрывающие поверхность, располагаются параллельно плоскости XZ исходной системы координат, либо секущие плоскости параллельны плоскости YZ, либо, наконец, поверхность рассекается одновременно двумя этими семействами плоскостей. В последнем случае поверхность оказывается покрытой сеткой, состоящей из криволинейных четырехугольников (криволинейная сетка). На рис. 8.4 показаны примеры таких сечений.

Задача построения изображения поверхности сводится, таким образом, к проецированию на картинную плоскость точек отдельных сечений, а именно тех точек плоскости XY, в которых заданы значения функции. Полученные образы точек затем соединяются отрезками, образуя ломаную линию - проекцию данного сечения на картинную плоскость. Это следует учитывать при задании поверхности. Например, если она не очень гладкая, то сетку, на которой задана функция, необходимо сделать более частой.

На рис. 8.4 показаны изометрические проекции некоторой поверхности. На нижнем рисунке при изображении использовались сечения параллельные плоскости YZ, на верхнем – параллельные плоскости XZ, а на среднем рисунке поверхность покрыта криволинейной сеткой.

Программа THREED, выполняющая графические построения, позволяет выбирать тот или иной из трех упомянутых способов изображения проекций. Не следует, однако, думать, что криволинейная сетка всегда является лучшим из способов. Во-первых, он требует примерно вдвое больше времени по сравнению с двумя другими (как процессорного, так и времени работы графопостроителя). Во-вторых, опыт показывает, что если угол между направлением проецирования и секущими плоскостями не превышает $5-10^\circ$, то для зрительного восприятия рисунки получаются не совсем удачными. В этом случае криволинейная сетка не обеспечивает должного качества изображения. Лучше выбрать один из двух оставшихся способов, а именно тот, при котором этот угол близок к 90° .

Для достижения большей наглядности изображения чаще всего требуется, чтобы невидимые участки поверхностей не рисовались. Основная идея алгоритма удаления невидимых линий, разработанного применительно к поверхностям, изображаемым способом параллельных сечений, достаточно хорошо известна и заключается в следующем.

Выбирается такой порядок построения сечений, при котором каждое сечение может быть закрыто только участками поверхности, определяемыми ранее построенными сечениями. В программе THREED, выполняющей графические построения, этот порядок обхода устанавливается автоматически в соответствии с выбранным направлением проецирования или заданным центром проекции. Причем в случае перспективной проекции в зависимости от расположения точки зрения относительно прямоугольной области определения функции поверхность при необходимости может быть автоматически разбита на 2 или 4 участка. Для каждого из них определяется свой порядок обхода сечений и каждый участок строится затем отдельно.

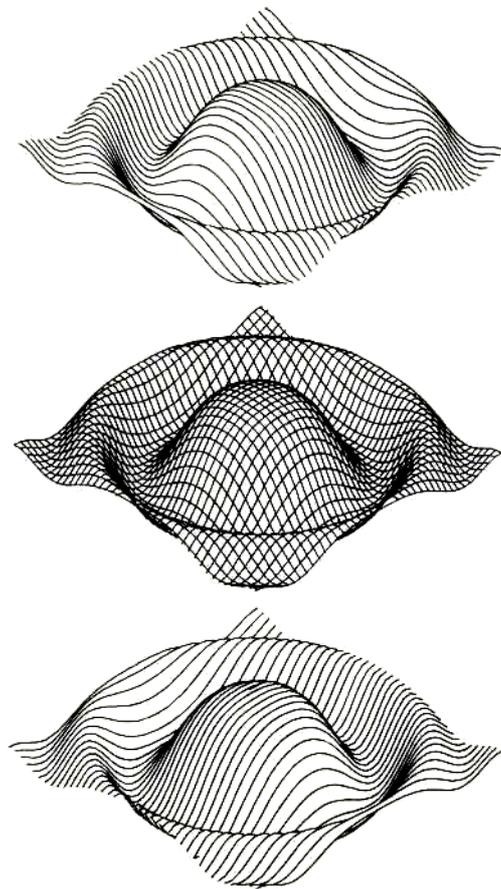


Рис. 8.4. Изометрические проекции поверхности с удаленными невидимыми линиями, построенные сечениями, параллельными плоскости XZ (верхний рисунок), сечениями, параллельными плоскости YZ (нижний рисунок), и способом "криволинейная сетка" (средний рисунок)

Если такой порядок обхода сечений фиксирован, то видимость каждой точки сечения может быть определена следующим образом.

Разобьем на картинной плоскости отрезок оси X , равный ширине рисунка, точками X_i , $i = 1, \dots, N$, на N равных частей. Будем при этом считать, что на картинной плоскости задана физическая сетка. Рассмотрим на плоскости рисунок многоугольник, вершины которого расположены на прямых X_i , $i = 1, \dots, N$. Этот многоугольник может быть невыпуклым, иногда вырожденным. Будем говорить в этом случае, что на плоскости рисунка существует экран. Этот экран представляет собой контур проекции участка поверхности, определяемого построенными к данному моменту сечениями. Выбранный порядок построения сечений позволяет использовать экран для определения видимости точек строящегося сечения. Если проекция точки попадает внутрь экрана, то такая точка считается невидимой. Рисоваться, таким образом, будут те участки сечений, которые расположены вне экрана. После построения проекции очередного сечения границы экрана изменяются в соответствии с построенными сечением.

Видимый участок сечения накапливается в виде массива абсцисс и ординат видимых точек этого участка. Служебная программа HCLINE использует для рисования видимого участка сечения программу LINEO. Заменяв программу HCLINE, можно, например, рисовать видимые и невидимые участки сечений линиями разного цвета или интенсивности, рисовать невидимые линии пунктиром и т. д.

Определение видимости отрезков сечений производится с точностью до границ экрана, которые лишь приближенно отслеживают реальный контур спроецированного к данному моменту участка поверхности. Поэтому на рисунках возможно появление незначительных дефектов. Делая физическую сетку более частой, можно уменьшить возможность появления таких искажений.

Границы экрана хранятся в массиве $AMXMN$ размером $2N$. Первые N элементов отводятся под хранение ломаной максимального рельефа (MXR), последние N элементов - для хранения ломаной минимального рельефа (MNR). Поскольку для всех точек физической сетки, где экран уже существует, $MXR(I) \geq MNR(I)$, то перед рисованием в массив MXR заносятся нули, а в массив MNR - единицы. Это означает, что в области рисунка экрана еще нет. Если возникает

необходимость на одном и том же рисунке изобразить две или более проекций поверхностей с учетом того, что они могут закрывать друг друга, то рисование последующих проекций надо производить, сохраняя то значение экрана, которое осталось после рисования предыдущих поверхностей. С этой целью в программе THREED предусмотрена возможность выбирать один из двух режимов: 1) инициализировать экран, т. е. занести в MXR нули, а в MNR единицы, 2) сохранить накопленный перед данным обращением к программе THREED экран. Используя вторую возможность, можно, например, рисовать некоторые неоднозначные поверхности.

Определение пределов изменения функции. При построении проекций поверхностей сохраняется обычный принятый в Графоре порядок работы. Сначала заводится страница, в ней определяется область (REGION) и устанавливаются пределы изменения функции. В трехмерном случае для установки пределов используется программа TDLIM. Эта программа вначале находит минимальные и максимальные значения координат точек поверхности, спроецированных на картинную плоскость: $XMIN, XMAX, ZMIN, ZMAX$. Затем выбирается одна из следующих возможностей: либо устанавливаются пределы изменения, равные найденным, либо в качестве пределов изменения берутся значения общие среди найденных и сформированных при предыдущих обращениях к программе TDLIM. В последнем случае можно рисовать в одной области несколько поверхностей, сохраняя на рисунке их взаимное расположение в пространстве. В частности, так можно получать проекции и некоторых неоднозначных поверхностей (см. пример 3 в п. 8.1.4).

Заметим, что пределы, определяемые программой TDLIM, задают в пространстве математических координат некоторый прямоугольник. Если прямоугольная область, определяемая программой REGION, не является подобной этой математической области, то изображение в плоскости рисунка произвольно подвергается еще одному сжатию или растяжению по одной из осей. Для того чтобы получить рисунок, который сохранял бы пропорции, определяемые данным способом проецирования, область графика надо задавать подобной математической области. С этой целью в программе TDLIM в качестве выходного параметра выдается коэффициент, равный отношению длины математической области к ее ширине. Учитывая этот коэффициент при задании размеров области (т. е. при вызове программы REGION), можно добиться сохранения желаемых пропорций.

Еще один способ сохранить правильные пропорции состоит в том чтобы не использовать программу TDLIM, а устанавливать пределы изменения спроецированных на картинную плоскость координат точек поверхности с помощью программы LIMITS.

Описание программ.

Программа TDLIM($X, Y, Z, NY, NX, ISTA, IFN, JST, JFN, S$) предназначена для задания пределов изменения функции на картинной плоскости. Пределы ищутся во всей области определения функции либо в некоторой ее прямоугольной подобласти. Обращение к программе может быть до и после задания REGION, но обязательно после того, как обращениями к программам преобразования координат сформировано результирующее преобразование. Эти пределы сохраняются до очередного обращения к программам, устанавливающим пределы. Параметры программы следующие:

X, Y - массивы точек сетки по осям X и Y , расположенных в порядке возрастания;

Z - двумерный массив значений функции в узлах сетки (размером (NX, NY));

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

$ISTA, IFN$ - индексы сетки по оси Y , определяющие нижнюю и верхнюю границы подобласти;

$ISTA > 0$ - в области рисования устанавливаются найденные для данной функции пределы,

$ISTA < 0$ - в области рисования устанавливаются пределы, общие среди найденных и тех, которые были получены при предыдущих обращениях к программе TDLIM;

JST, JFN - индексы сетки по оси X , определяющие левую и правую границы подобласти;

S - коэффициент, определяющий форму математической области значений функции на картинной плоскости; он равен отношению $(XMAX - XMIN)/(ZMAX - ZMIN)$, где $XMIN, XMAX, ZMIN, ZMAX$ – пределы, устанавливаемые в области рисования.

Программа THREED($X, Y, Z, NY, NX, ISTA, IFN, JSTA, JFN, LNTP, NGRD, AMXMN, AR$) предназначена для того, чтобы в соответствии с установленными пределами и выбранным способом проецирования построить в области рисования проекцию всей поверхности либо ее некоторого участка одним из имеющихся способов. Рисование может быть выполнено с удалением невидимых линий или без него. В первом случае при рисовании сечениями, параллельными плоскостям XZ или YZ , старое значение экрана, оставшееся от предыдущих

обращений к программе THREEED, может быть сохранено либо перед началом рисования уничтожено. При изображении криволинейной сетки старое значение экрана перед рисованием уничтожается. Программа имеет следующие параметры:

X, Y – массивы точек сетки по осям X и Y , расположенных в порядке возрастания;

Z - двумерный массив значений изображаемой функции в узлах сетки (размером (NY, NX));

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

$ISTA, IFN$ - индексы сетки по оси Y , определяющие нижнюю и верхнюю границы подобласти:

$ISTA > 0$ - рисование с удалением невидимых линий,

$ISTA < 0$ - рисование без удаления невидимых линий;

$JSTA, JFN$ - индексы сетки по оси X , определяющие левую и правую границы подобласти:

$JSTA > 0$ - перед рисованием старое значение экрана уничтожается,

$JSTA < 0$ - рисование выполняется с учетом имеющегося экрана;

$LNTP$ - параметр, определяющий способ построения проекции:

$LNTP = -1$ - проекция строится сечениями матрицы значений, параллельными плоскости YZ (таких сечений $JFN - JSTA + 1$),

$LNTP = 1$ - проекция строится сечениями матрицы значений, параллельными плоскости XZ (таких сечений $IFN - ISTA + 1$),

$LNTP = 0$ - проекция строится способом криволинейной сетки;

$NGRD$ - число делений физической сетки (т. е. размер экрана);

$AMXMN$ - рабочий массив для хранения границ экрана (длины $2 * NGRD$);

AR - рабочий массив, используемый для выделения видимого участка сечения (длины $2 * \text{MAX}(NY, NX)$).

Замечание. Если $ISTA < 0$, т. е. удаление невидимых линий не производится, то значения параметров $NGRD$ и $AMXMN$, а также знак $JSTA$ безразличны. В этом случае вместо $AMXMN$ можно использовать любую переменную или массив, описанный в программе, которая обращается к THREEED.

Вспомогательные и служебные программы.

Программа HCNCRD(X, Y, Z) позволяет по заданным координатам точки в декартовой системе координат определить координаты этой точки после применения к ней результирующего преобразования. Новые координаты помещаются в общий блок, описываемый как COMMON /GFCRD/XP, YP, ZP.

Программа HCsurf($X, Y, Z, NY, NX, ISTA, IFN, JSTA, JFN, LNTP, MOUX, MOUY, AMXMN, AR$) служит для построения в области рисования проекции заданного участка поверхности одним из двух способов: сечениями, параллельными плоскости XZ , ($LNTP = 1$) или сечениями, параллельными плоскости YZ , ($LNTP = -1$), - с указанным направлением обхода по строкам и столбцам матрицы значений Z . Если $MOUX = -1$, то элементы строки выбираются справа налево, если же $MOUX = 1$, то - слева направо. При $MOUY = 1$ элементы столбца выбираются в таком порядке, как они расположены в матрице Z , а при $MOUY = -1$ - в обратном порядке. Смысл остальных параметров тот же, что и в программе THREEED.

Функция HCIND(X, IST, IFN, EL) позволяет для заданного числа EL найти среди элементов массива X , расположенных в порядке возрастания, элемент с таким индексом $IST \leq I < IFN$, что для него выполняется соотношение $X(I) \leq EL \leq X(I+1)$.

Программа TDSECT($ZIJ, X, Y, IX, IY, M, N, AMXMN, AR$) предназначена для построения проекции одного сечения с указанным направлением перебора точек сечения. Построение может быть выполнено как с удалением невидимых линий, так и без него.

Программа HCLINE(X, Y, NP) рисует линию по заданным массивам X и Y длины $|NP|$. Если $NP < 0$ (участок невидим), линия не рисуется.

Функция ZINT($X1, Z1, X2, Z, X$) находит значение функции в точке X методом линейной интерполяции по заданным точкам $(X1, Z1)$ и $(X2, Z2)$.

8.1.3. Построение проекций пространственных кривых. В этом пункте приводится описание программ, позволяющих строить проекции пространственных кривых. Кривые задаются тремя одномерными массивами X, Y, Z размерности N , где N - количество точек, определяющих кривую. Так же как и при построении проекций поверхностей заводится страница, в ней задается область для рисования. Для установления пределов служит программа TDLIML. Принцип работы этой программы, а также обеспечиваемые ею возможности такие же, как и у программы TDLIM. Непосредственное построение выполняет программа TDLINE.

Программа TDLIML(X, Y, Z, N, S) предназначена для задания пределов изменения проекции пространственной кривой на картинной плоскости. Обращение к программе может производиться до и после задания области рисования, но обязательно после того, как обращениями к программам преобразования координат будет сформировано результирующее преобразование, соответствующее выбранному способу проецирования. Эти пределы сохраняются до очередного обращения к программам установки пределов (TDLIML, TDLIM, LIMITS и др.). Параметры программы:

X, Y, Z - массивы длины |N|, определяющие X-, Y- и Z-координаты точек пространственной кривой;

|N| - количество точек, задающих кривую:

N > 0 - в области рисования устанавливаются найденные для данной кривой пределы,

N < 0 - в области рисования устанавливаются пределы общие среди тех, которые были найдены для данной кривой и полученных при предыдущих обращениях к программе TDLIML;

S - коэффициент, определяющий форму математической области, в которую попадает проекция кривой на картинной плоскости, и равный отношению длины этой области к ее высоте.

Программа TDLINE(X, Y, Z, N) позволяет в соответствии с установленными пределами и выбранным способом проецирования построить проекцию пространственной кривой. Параметры программы следующие:

X, Y, Z - массивы длины N, определяющие соответственно X-, Y- и Z-координаты точек пространственной кривой;

N - количество точек, задающих кривую.

8.1.4. Примеры. Ряд примеров иллюстрирует использование описанных выше программ.

1. На рис. 8.4 показаны изометрические проекции поверхности, построенные различными способами, с удалением невидимых линий. Массивы, описывающие поверхность, заполнялись подпрограммой SURF1. Рисунок строился с помощью следующей программы:

```
DIMENSION X(41),Y(41), Z(41,41), A(200),AR(82)
CALL SURF1(X,Y,Z)
CALL INIT
CALL ISOMET
CALL PAGE(17.,26.,0, 0, 0)
CALL TDLIM(X,Y,Z, 41, 41, 1, 41, 1, 41, S)
SY = 2.
LNTP = -1
DO 3 I = 1, 3
CALL REGION(2.,SY,13.,7.5, 0, 0, 0)
CALL THREED(X,Y,Z,41,41,1,41,1,41, LNTP,100, A, AR)
SY = SY + 8.
3 LNTP = LNTP + 1
CALL ENDPG('8.4')
END
```

2. На рис. 8.5 показаны три различные центральные проекции одной и той же поверхности. При их построении точка зрения оставалась неизменной, а поворачивалась сама поверхность вокруг оси Z. С этой целью каждый раз перед обращением к программе VIEW матрица преобразования запоминалась, и это состояние матрицы восстанавливалось перед очередным поворотом поверхности. Массивы, задающие поверхность, формировались подпрограммой SURF2.

```
DIMENSION X(61),Y(61), Z(61,61), A(400), AR(122), A1(16)
CALL SURF2(X,Y,Z)
CALL INIT
CALL PAGE(17.,26.,0, 0, 0)
SY = 1.5
DO 1 I = 1, 3
CALL REGION(1.,SY,15.,7.,0, 0, 0)
CALL SAVETR(A1)
CALL VIEW(-42.,-42.,45.)
CALL TDLIM(X,Y,Z, 61, 61, 1, 61, 1, 61, S)
CALL THREED(X,Y,Z, 61, 61, 1, 61, 1, 61, 0, 200, A, AR)
```

```
CALL SETTR(A1)
CALL TDROT(3, 90.)
1 SY = SY + 8.
CALL ENDPG('8.5')
END
```

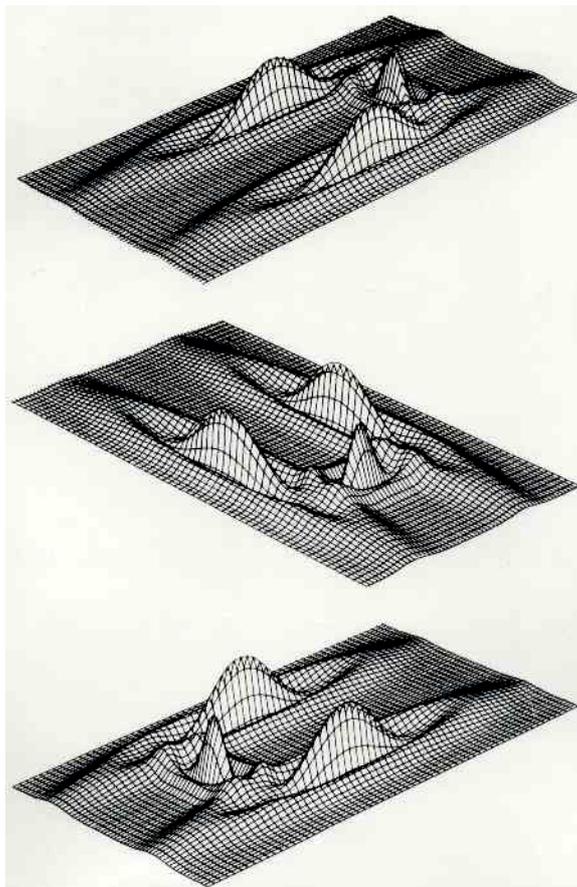


Рис. 8.5. Три различные центральные проекции одной и той же поверхности

3. На рис. 8.6 изображена неоднозначная поверхность. При построении она разбивается на два участка, каждый из которых является однозначной поверхностью в области задания. Для этих участков поверхности определяются общие пределы изменения с использованием имеющихся для этого возможностей в программе TDLIM. После определения пределов рисуется вначале тот участок, который по отношению к выбранной точке зрения расположен ближе к наблюдателю и, следовательно, не может быть закрыт вторым участком. Затем изображается второй участок с сохранением экрана, оставшегося после рисования первого участка. При этом рисование может проводиться сечениями, параллельными либо плоскости XZ , либо плоскости YZ .

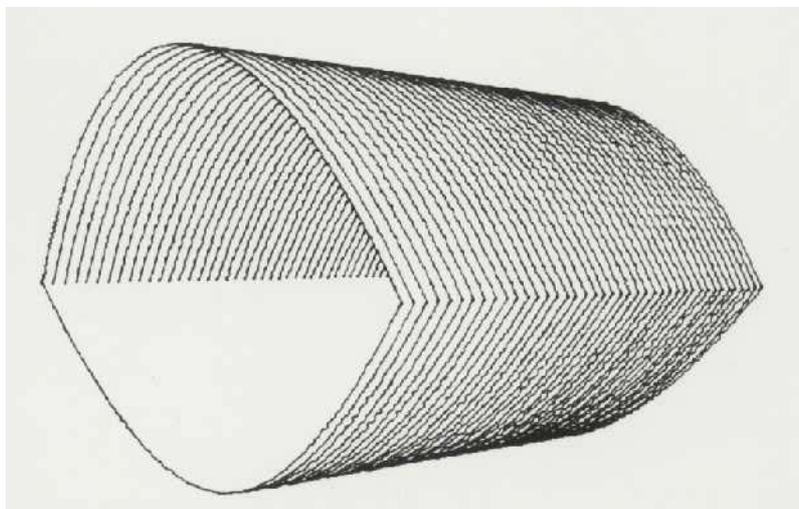


Рис. 8.6. Пример проекции неоднозначной поверхности.

```

DIMENSION X(41),Y(61), Z(61,41), A(500), AR(122), T(16), T1(16)
DATA VX, VY, VZ/-10.,-10.,2./, NY, NX/61, 41/
DO 1 I = 1, NX
1 X(I) = -3. + 6.*(I-1)/(NX-1)
  DO 2 J = 1, NY
2 Y(J) = -3. + 6.*(J-1)/(NY-1)
  DO 3 J = 1, NY
  DO 3 I = 1, NX
3 Z(J, I) = 9.-Y(J)**2
CALL PAGE(17.,9.,0, 0, 0)
CALL REGION(2.,.5, 13.,8.,0, 0, 0)
CALL INIT
CALL VIEW(VX,VY,VZ)
CALL SAVETR(T)
CALL TDLIM(X,Y,Z, NY, NX, 1, NY, 1, NX, S)
CALL INIT
CALL TDSCAL(3,-1.)
CALL VIEW(VX,VY,VZ)
CALL SAVETR(T1)
CALL TDLIM(X,Y,Z, NY, NX,-1, NY,1, NX, S)
CALL SETTR(T)
CALL THREED(X,Y,Z, NY, NX, 1, NY, 1, NX,-1, 250, A, AR)
CALL SETTR(T1)
CALL THREED(X,Y,Z, NY, NX, 1, NY,-1, NX,-1, 250, A, AR)
CALL ENDPG('8.6')
END

```

4. На рис. 8.7 изображена центральная проекция пространственной кривой. Задающие ее массивы формируются программой LINE3D.

```

DIMENSION X(3417),Y(3417), Z(3417)
CALL LINE3D(X,Y,Z)
CALL PAGE(17.,17.,0, 0, 0)
CALL REGION(1.,1.,15.,15.,0, 0, 0)
CALL INIT
CALL VIEW(15.,-8.,3.)
CALL TDLIML(X,Y,Z, 3417, FI)
CALL TDLINE(X,Y,Z, 3417)
CALL ENDPG('8.7')
END

```

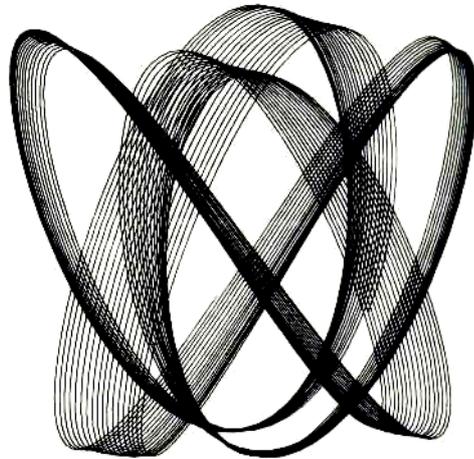


Рис. 8.7. Изображение центральной проекции пространственной кривой.

5. На рис. 8.2 и 8.3, кроме проекций поверхностей показаны также проекции на картинную плоскость осей декартовой системы координат. Их построение выполнялось с помощью подпрограммы TDAXES.

```

SUBROUTINE TDAXES(X,Y,REG, L)
COMMON /GFCRD/ XP,YP,ZP
DIMENSION X1(3), Z1(3)
CALL HCNCRD(0.,0.,0.)
X0 = XP
Z0 = ZP
XMN = XP

```

```

XMX = XP
ZMN = ZP
ZMX = ZP
DO 7 J = 1, 3
IF (J.EQ.1) CALL HCNCRD(1.,0.,0.)
IF (J.EQ.2) CALL HCNCRD(0.,1.,0.)
IF (J.EQ.3) CALL HCNCRD(0.,0.,1.)
X1(J) = XP
Z1(J) = ZP
IF ((XP-XMN).LT.0.) XMN = XP
IF ((XP-XMX).GT.0.) XMX = XP
IF ((ZP-XMN).LT.0.) ZMN = ZP
IF ((ZP-ZMX).GT.0.) ZMX = ZP
7 CONTINUE
RX = XMX - XMN
RZ = ZMX - ZMN
DX = REG
DZ = REG
IF((RX-RZ).LT.0.) DX = DZ*RX/RZ
IF((RX-RZ).GT.0.) DZ = DX*RZ/RX
CALL REGION(X,Y, DX, DZ, ',1, L)
CALL SET(0)
CALL LIMITS(XMN, XMX, ZMN, ZMX)
DO 8 J = 1, 3
CALL TMF(X0,Z0,XR,ZR)
CALL MOVE(XR, ZR, 0)
CALL TMF(X1(J), Z1(J), XR, ZR)
CALL MOVE(XR, ZR, 1)
IF(J.EQ.1) CALL SYMBOL(XR-.5, ZR-1.2,.7,'X',1, 0.)
IF(J.EQ.2) CALL SYMBOL(XR-.5, ZR+.2,.7,'Y',1, 0.)
IF(J.EQ.3) CALL SYMBOL(XR+.5, ZR-.8,.7,'Z',1, 0.)
8 CONTINUE
RETURN
END

```

8.2. Изображение функций двух переменных.

Метод ячеек.

В этом параграфе описаны программы, позволяющие строить центральные проекции поверхностей, представленных однозначными непрерывными функциями двух переменных, заданными на прямоугольной сетке (в общем случае неравномерной), с удалением невидимых линий. Применяемый способ стирания невидимых линий основывается на упорядочении изображаемых элементов поверхности, которые в данном случае представляют собой ячейки криволинейной сетки.

8.2.1. Алгоритм удаления невидимых линий. Пусть на прямоугольной области R задана поверхность S , описываемая однозначной непрерывной функцией двух переменных. Без ограничения общности можем считать, что центр области R совпадает с началом координат, а ее стороны параллельны осям X и Y . В качестве картинной плоскости будем рассматривать плоскость P , проходящую через начало координат и перпендикулярную прямой, которая соединяет центр проекции V с началом координат (рис. 8.8). Определим на картинной плоскости прямоугольную систему координат $X'Y'$ такую, что ось Y' является центральной проекцией оси Z на плоскость P .

Введем в области R прямоугольную сетку размером $LLX \times LLY$ (не обязательно равномерную). Значения функции задаются в узлах сетки, а каждый элемент поверхности определяется четырьмя точками, в промежутках между которыми функция линейно интерполируется. Образом такой линейной фигуры на картинной плоскости будет четырехугольник, причем изображаться будут только видимые ребра (или их части).

Применяемый в этом случае алгоритм удаления невидимых линий основывается на упорядочении изображаемых элементов поверхности. При этом выбирается такой порядок их построения, при котором элемент, проецируемый в данный момент на картинную плоскость, не может закрывать другие, спроецированные ранее элементы поверхности. В результате тот или иной отрезок ребра невидим из точки V в том и только в том случае, если его проекция попадает внутрь области, образованной проекциями предыдущих элементов. Образ поверхности на картинной плоскости достраивается по мере того как берутся новые элементы.

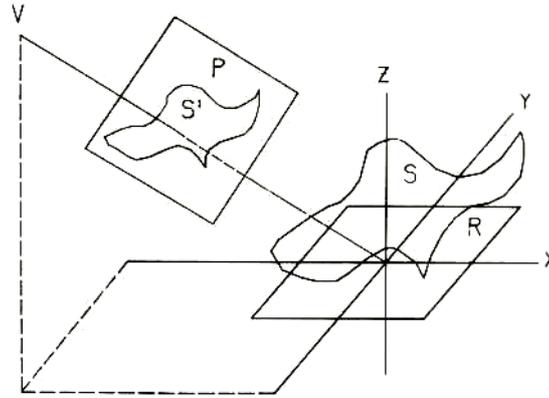


Рис. 8.8. Центральная проекция S' поверхности S на плоскость P (плоскость P для наглядности смещена относительно прямой, соединяющей центр проекции V с началом координат)

Можно предложить несколько различных способов упорядочения элементов поверхности (точнее их ребер), при которых спроецированная часть поверхности не может быть закрыта последующими элементами. На рис. 8.9 показан один из возможных способов упорядочения (для юго-западного положения проекции V_0 точки V на плоскость XY относительно области R).

Такое упорядочение элементов поверхности позволяет определить экран, т. е. заключить невидимую область на картинной плоскости между двумя кусочно-линейными непрерывными функциями (ср. с п. 8.1.2). Одна из этих функций совпадает с верхней границей построенной к этому времени на картинной плоскости фигуры (ломаная максимального рельефа). Другая функция используется для определения нижней границы проекции (ломаная минимального рельефа). Изображаться при этом будут только те части элементов поверхности, проекции которых выходят за пределы экрана.

Для экономного задания экрана разобьем отрезок на оси X' , равный ширине изображения, на NN равных частей и будем задавать значения функций $RMAX$ и $RMIN$ в узлах этой равномерной сетки. Таким образом, на картинной плоскости вводится физическая сетка. Заметим, что при этом шаг разбиения равен величине $(XMAX - XMIN)/NN$, где $XMAX$, $XMIN$ - максимальное и минимальное значения X' -координат всех спроецированных на плоскость P точек поверхности.

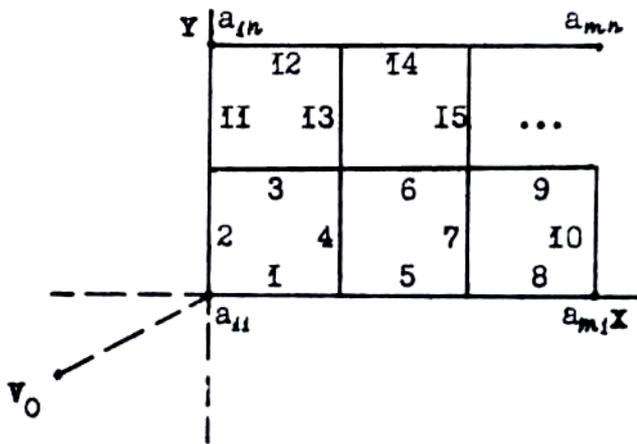


Рис. 8.9. Способ упорядочения ребер элементов поверхности (для юго-западного положения центра проекции)

В связи с тем, что границы экрана не всегда точно отображают форму видимой части проекции, в области рисунка обычно существуют участки, в которых могут появляться некоторые искажения: изменение угла наклона видимой части ребра, недоведение отдельных отрезков до границ экрана или, наоборот, вторжение некоторых отрезков в невидимую область. Эти дефекты, как правило, незначительны и мало влияют на качество изображения. Возможность появления дефектов такого рода можно уменьшить, сделав физическую сетку более частой.

Перед тем как начать проецирование, проводится инициализация экрана: в массив $RMAX$, содержащий значения ломаной максимального рельефа, заносятся нули, а в массив $RMIN$, описывающий ломаную минимального рельефа, - единицы. В результате можно легко находить те участки области рисунка, на которых экран еще не определен (в этих точках значения $RMAX$ меньше значений $RMIN$). После изображения каждого ребра элемента поверхности $RMAX$ или $RMIN$ (возможно и оба) будут модифицироваться, отражая тем самым изменение формы "нарастающего" образа поверхности на картинной плоскости.

8.2.2. Выбор ракурса. Определение пределов изменения функции. Способы упорядочения ребер элементов поверхности, аналогичные изображенному на рис. 8.9, можно предложить и для других, не только юго-западных угловых положений точки V_0 . Если же V_0 лежит непосредственно на западе, севере, востоке или юге от R или попадает внутрь R , то область R разбивается на две или четыре части (рис. 8.10), и алгоритм применяется к каждому из полученных прямоугольников, относительно которых V_0 уже имеет угловое положение.

Перед изображением каждой из таких частей происходит инициализация экрана. Причина этого заключается в следующем.

Установлено (см. [1, г, 1, ж]), что при некоторых достаточно близких к изображаемому объекту положениях центра проекции рисунок часто получается неудачным для зрительного восприятия, так как наблюдается нарушение монотонности изменения X' -координаты на картинной плоскости. В результате некоторые на самом деле видимые отрезки не изображаются или изображаются только частично, что приводит к появлению непрорисованных белых пятен на изображении. В таких случаях рекомендуется сменить ракурс, например, отодвинув центр проекции от объекта.

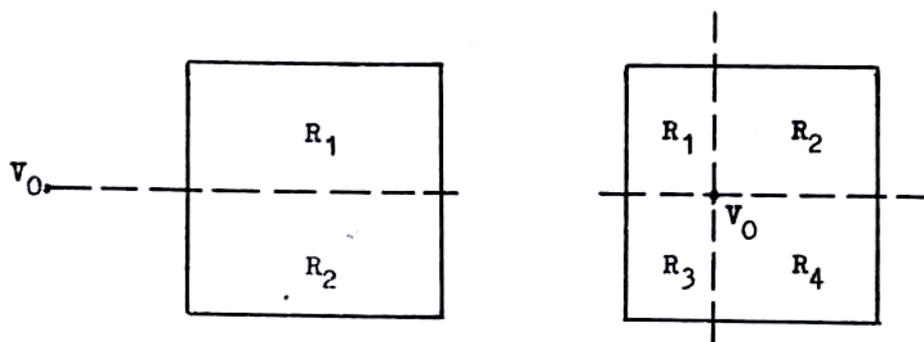


Рис. 8.10. Разбиение области R в случае неуглового положения V_0 .

Если же проецируемая поверхность состоит из нескольких частей и перед изображением каждой из них (за исключением первой) экран не инициализируется, то вероятность появления белых пятен значительно возрастает.

Хотя инициализация экрана перед проецированием и очень желательна, однако при необходимости можно сохранить прежнее содержимое экрана, полученное при рисовании предыдущих частей поверхности, т. е. можно выбирать один из следующих режимов:

- а) инициализация экрана перед изображением каждой из частей поверхности;
- б) рисование с учетом экрана, оставшегося от изображения предыдущих частей поверхности.

Вторым режимом следует пользоваться только в тех случаях, когда возникает необходимость в одной и той же области нарисовать проекции двух или более поверхностей с учетом того, что они могут закрывать друг друга. В этом случае изображение каждой из последующих поверхностей надо производить с учетом значения экрана, которое осталось после проецирования предыдущих поверхностей. В связи с этим в программе PRSP, которая выполняет построение изображения, предусмотрены две возможности:

- а) инициализация экрана перед изображением объекта;
- б) сохранение старого значения экрана, оставшегося от предыдущих обращений к программе PRSP.

Используя вторую возможность, можно, в частности, получать проекции некоторых неоднозначных поверхностей (см. п. 8.2.4).

Центром проекции, вообще говоря, может быть любая точка пространства, не лежащая на самой поверхности. При этом, однако, следует иметь в виду, что всегда существует опасность выбрать

такое взаимное расположение центра проекции и поверхности, при котором образы некоторых точек поверхности вообще могут не существовать, т. е. соответствующие проецирующие прямые будут параллельны картинной плоскости.

В связи с этим в алгоритм центрального проецирования был введен некоторый вариант отсечения - ограничивающий конус зрения с углом между образующими 160° . Все точки поверхности, лежащие за его пределами, игнорируются. Таким образом, исключается возможность проецирования участков поверхности, не имеющих образов на картинной плоскости. Следует, однако, подчеркнуть, что наличие точек поверхности, не попавших в конус зрения, свидетельствует прежде всего о необходимости изменить положение центра проекции, например, отдалив его от поверхности. Поскольку угол между образующими конуса превышает угол ясного зрения (60°), то даже полное попадание в конус проецируемого участка поверхности еще не гарантирует получения хорошего изображения. Все эти обстоятельства необходимо принимать во внимание при выборе ракурса.

Для задания пределов изменения X' - и Y' -координат на картинной плоскости используется программа MNMX. Она выполняет те же функции, что и программа TDLIM, описанная в п. 8.1.2.

8.2.3. Описание программ.

Программа MNMX(VX, VY, VZ, X, Y, A, LLX, LLY, LXINI, LXE, LYIN, LYE, S) служит для задания пределов изменения изображаемой функции на картинной плоскости. Пределы устанавливаются либо для всей области задания функции, либо для некоторой ее прямоугольной подобласти. Обращение к программе может быть как до, так и после вызова программы REGION. Установленные пределы сохраняются до очередного обращения к программе MNMX. Программа имеет следующие параметры:

VX, VY, VZ - координаты центра проекции в системе координат XYZ;

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

A - значения функции в узлах сетки (двумерный массив размером (LLX, LLY));

LLX, LLY - размеры сетки по осям X и Y;

LXINI, LXE - индексы сетки по оси X, определяющие левую и правую границы подобласти;

LXINI > 0 - в области рисования, определенной обращением к программе REGION, устанавливаются найденные для данной функции пределы,

LXINI < 0 - в области рисования устанавливаются пределы, общие для пределов изменения данной функции и пределов, полученных при предыдущих обращениях к программе MNMX;

LYIN, LYE - индексы сетки по оси Y, определяющие нижнюю и верхнюю границы подобласти;

S - коэффициент, определяющий форму математической области значений функции на картинной плоскости: он равен отношению $(XMAX - XMIN)/(YMAX - YMIN)$, где XMAX, XMIN, YMAX, YMIN - пределы изменения функции, устанавливаемые в области рисования.

Замечание. При обращении к программе MNMX надо следить, чтобы центр проекции не попал на поверхность изображаемого объекта.

Программа PRSP(VX, VY, VZ, X, Y, A, LLX, LLY, LXINIT, LXEND, LYINIT, LYEND, XF1, YF1, RMAX, RMIN, NN) позволяет построить в области рисования центральную проекцию всей заданной поверхности либо ее некоторого участка с удалением невидимых линий. Программа имеет следующие параметры:

VX, VY, VZ - координаты центра проекции в системе координат XYZ;

X, Y - координаты точек сетки по осям X и Y, расположенных в порядке возрастания;

A - значения функции в узлах сетки (двумерный массив размером (LLX, LLY));

LLX, LLY - размеры сетки по осям X и Y;

LXINIT, LXEND - индексы сетки по оси X, определяющие левую и правую границы проецируемого участка поверхности;

LXINIT > 0 - перед рисованием экран инициализируется, т. е. старое значение его, оставшееся от предыдущих обращений к программе PRSP, уничтожается;

LXINIT < 0 - рисование осуществляется с учетом экрана, оставшегося от предыдущих обращений к программе PRSP;

LYINIT, LYEND - индексы сетки по оси Y, определяющие нижнюю и верхнюю границы проецируемого участка поверхности;

LYINIT > 0 - перед изображением проецируемого участка поверхности или каждой из его частей, полученных при разбиении в случае неуглового положения центра проекции, происходит инициализация экрана;

$LYINIT < 0$ - рисование выполняется с учетом экрана, оставшегося от изображения предыдущих участков поверхности или их частей, полученных при разбиении (это дает возможность получить полное значение экрана от всего проецируемого участка поверхности независимо от положения центра проекции);

$XF1, YF1$ - рабочие массивы длины соответственно $2 \times LLX$ и LLX , предназначенные для запоминания X - и Y -компонент координат точек сетки, расположенных вдоль оси X ;

$RMAX, RMIN$ - массивы, используемые для хранения верхней и нижней границ экрана (они должны иметь длину не менее NN);

NN - число делений физической сетки на области рисунка (т. е. размер экрана);

Замечания.

1) Если при изображении нескольких проекций поверхностей с учетом того, что они могут закрывать друг друга, центр проекции находится в угловом положении относительно первой поверхности, то при обращении к программе $PRSP$ для первой поверхности знак $LYINIT$ безразличен.

2) При обращении к программе $PRSP$ надо следить, чтобы центр проекции не попал на поверхность изображаемого объекта.

Программа $PRJT(VX, VY, VZ, X, Y, A, LLX, LLY, LXIN, LXE, LYIN, LYE, XF1, YF1)$ позволяет построить в области рисования центральную проекцию всей заданной поверхности либо ее некоторого участка без удаления невидимых линий. Параметры программы следующие:

$LXIN, LXE$ - индексы сетки по оси X , определяющие левую и правую границы проецируемого участка;

$LYIN, LYE$ - индексы сетки по оси Y , определяющие нижнюю и верхнюю границы проецируемого участка;

Остальные параметры имеют то же значение, что и в предыдущей программе.

Замечание. При обращении к программе рисования $PRJT$ надо следить, чтобы центр проекции не попал на поверхность изображаемого объекта.

Вспомогательные и служебные программы.

Программа $MAP(X, Y, Z)$ по заданным координатам точки в декартовой системе координат позволяет определить координаты проекции этой точки на картинную плоскость. Вычисленные координаты XP, YP помещаются в общий блок $GFVP$, описываемый как $COMMON/GFVP/V1, V2, V3, XP, YP, INFIN$. При этом параметр $INFIN$ является признаком, определяющим, попадает ли данная точка поверхности в конус зрения ($INFIN = 0$ - попадает, $INFIN = 1$ - лежит за его пределами).

Программа $CORN(LXI1, LX1, LYJ1, LY1, K1, M1, K2, M2, X, Y, A, LXX, LYY, XF1, YF1, RMAX, RMIN)$ позволяет построить центральную проекцию части поверхности (задаваемой первыми четырьмя параметрами), относительно которой центр проекции имеет угловое положение. Построение производится с удалением невидимых линий. Параметры $K1, K2$ определяют координаты ближайшего к центру проекции угла проецируемой части поверхности, а $M1, M2$ задают порядок перебора элементов соответственно строк и столбцов рассматриваемой части поверхности (от начального элемента к конечному или от конечного к начальному). Остальные параметры имеют те же значения, что и в программе $PRSP$.

Программа $PATCH(XF1, YF1, RMAX, RMIN)$ служит для построения на картинной плоскости проекции одного элемента изображаемой части поверхности с удалением невидимых линий.

Программа $SEARCH(IX1, Y1, IX2, Y2, XF1, YF1, RMAX, RMIN)$ предназначена для обработки частично видимых ребер элементов поверхности (первые два параметра задают видимую конечную точку ребра, а вторые два - невидимую). С этой целью на физической сетке ищется первая видимая точка ребра, после чего изображается отрезок, соединяющий видимую конечную точку с найденной точкой.

(Параметры $IX1, IX2$ определены на физической сетке, введенной в области рисования, поэтому их значения - целые числа).

Программа $INTRSC(X1, Y1, X2, Y2, X3, Y3, X4, Y4, X0, Y0)$ позволяет найти точку пересечения $(X0, Y0)$ двух отрезков, заданных координатами конечных точек.

Программа $BUFL(IX1, Y1, IX2, Y2, RMAX, RMIN)$ запоминает информацию о видимой части заднего ребра каждого элемента поверхности и после полного просмотра элемента модифицирует экран.

Программа $LININT(JX1, Y1, JX2, Y2, RMAX, RMIN)$ позволяет линейно интерполировать функции $RMAX$ или $RMIN$ (возможно обе) между двумя заданными точками $(JX1, Y1)$ и $(JX2, Y2)$. При

этом новые значения заносятся в RMAX (RMIN) только в том случае, если они больше (меньше) старых.

8.2.4. Примеры. Следующие примеры показывают различные возможности программ центрального проецирования.

1. На рис. 8.11 изображена поверхность, описывающая распределение концентрации полезного компонента на участке сложноструктурного месторождения руды, построенное по результатам статистической обработки данных. Ниже приведена программа, с помощью которой строился этот рисунок. Массивы, описывающие поверхность, заполнялись подпрограммой MOUNT.

```
DIMENSION PHI(49,103), XP(49), YP(103), XF(98), YF(49), RX(2000),RN(2000)
DATA A,B,C/1032.,-1000.,1037./
CALL MOUNT(XP,YP,PHI)
CALL PAGE(15.,12.,0,0,0)
CALL MNMX(A,B,C,XP,YP,PHI,49,103,1,49,1,103,S)
CALL PRSP(A,B,C,XP,YP,PHI,49,103,1,49,1,103,XF,YF,RX,RN,2000)
CALL ENDPG('8.11')
END
```

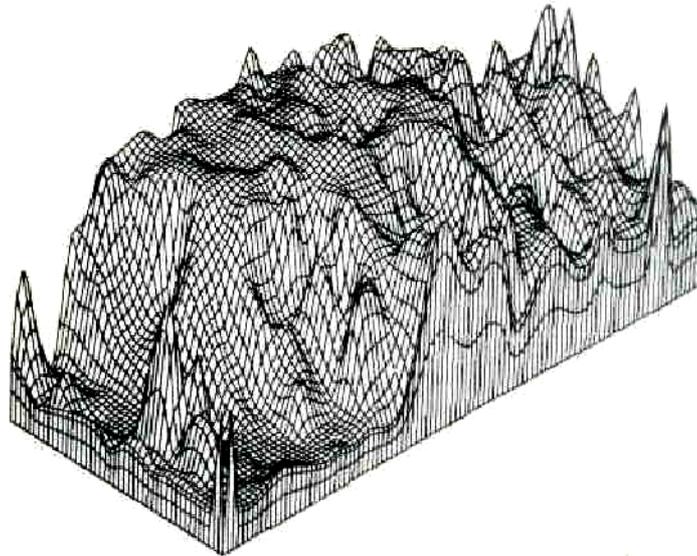


Рис. 8.11. Пример центральной проекции поверхности с удаленными невидимыми линиями

2. На рис. 8.12 показаны центральные проекции целой поверхности и некоторого ее участка. На среднем рисунке берутся пределы изменения, установленные для всей поверхности, а на верхнем - пределы, вычисленные только для рассматриваемого участка поверхности. Изображения строились с помощью следующего фрагмента программы:

```
CALL PAGE(17.,26.,0,0,0)
CALL REGION(1.,2.,15.,7.5,0,0,1)
CALL MNMX(35.,90.,100.,X,Y,A,37,37,1,37,1,37,S)
CALL PRSP(35.,90.,100.,X,Y,A,37,37,1,37,1,37,XF1,YF1,RMAX,RMIN,300)
CALL REGION(1.,10.,15.,7.5,0,0,1)
CALL PRSP(35.,90.,100.,X,Y,A,37,37,1,18,1,27,XF1,YF1,RMAX,RMIN,300)
CALL REGION(1.,18.,15.,7.5,0,0,1)
CALL MNMX(35.,90.,100.,X,Y,A,37,37,1,18,1,27,S)
CALL PRSP(35.,90.,100.,X,Y,A,37,37,1,18,1,27,XF1,YF1,RMAX,RMIN,300)
CALL ENDPG('8.12')
```

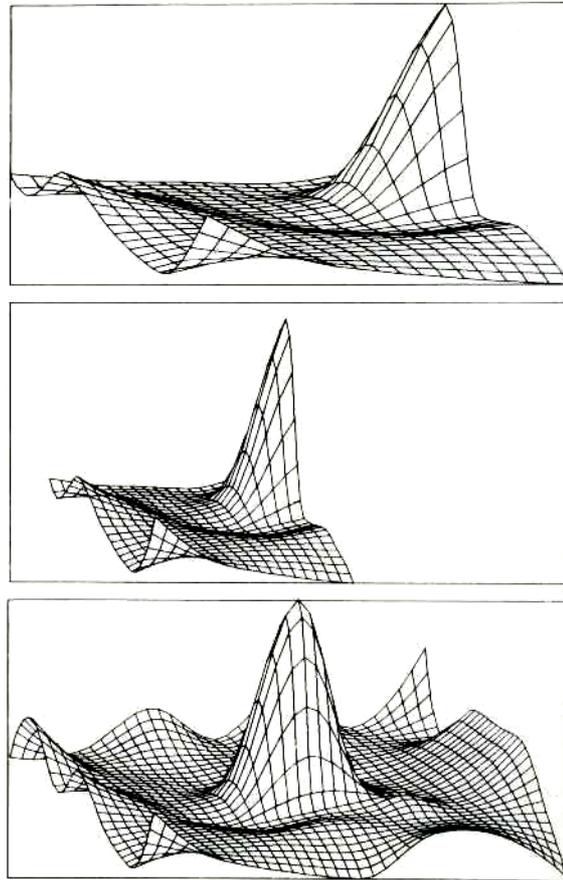


Рис. 8.12. Центральные проекции целой поверхности и некоторого ее участка

3. При желании можно получить изображение поверхности, как бы стоящей на "постаменте" определенной высоты (рис. 8.13). Для этого крайним строкам и столбцам массива следует присвоить некоторое постоянное значение (в этом примере оно равно $-0,3$).

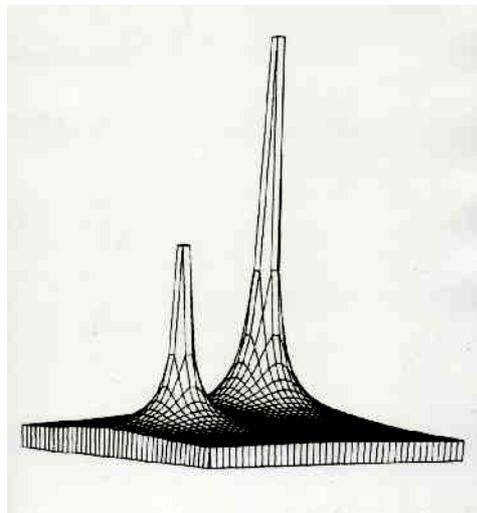


Рис. 8.13. Изображение алгебраической поверхности на "постаменте"

```

DIMENSION X(42),Y(42), A(42,42), XF1(84), YF1(42)
DIMENSION RMAX(500), RMIN(500)
DATA XA,YA,XB,YB/0.525,1.025,2*1.525/
DATA X(1),Y(1),X(2),Y(2)/-0.01,-0.01,0.,0./
DO 85 I = 3, 42
  X(I) = X(I-1) + .05
85 Y(I) = X(I)
  DO 86 K = 1, 42
    A(1,K) = -0.3
86 A(K,1) = -0.3

```

```

DO 87 J = 2, 42
DO 87 I = 2, 42
87 A(I,J) = .2*(1./SQRT((X(I)-XA)**2 + (Y(J)-YA)**2) + 2./SQRT((X(I)-XB)**2 + (Y(J)-YB)**2))
CALL PAGE(17.,15.,0, 0, 0)
CALL REGION(2.5,2.,12.,12.,0, 0, 0)
CALL MNMX(-15.,-20.,10.,X,Y,A, 42, 42,1, 42,1, 42, S)
CALL PRSP(-15.,-20.,10.,X,Y,A, 42, 42,1, 42,1, 42, XF1,YF1, RMAX, RMIN, 500)
CALL ENDPG('8.13')
END

```

4. На рис. 8.14 показана проекция неоднозначной поверхности. Заданная поверхность разбивается на два участка, каждый из которых является однозначной функцией в своей области определения. С помощью программы MNMX ищутся пределы изменения для обоих участков

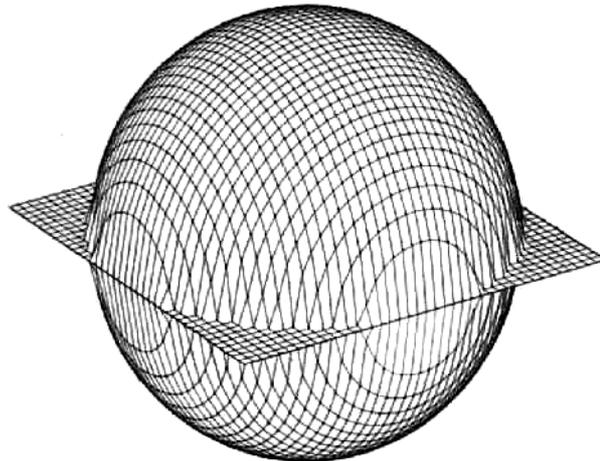


Рис. 8.14. Пример проекции неоднозначной поверхности

поверхности. Далее, сначала изображается тот участок поверхности, который расположен ближе к выбранному центру проекции и, следовательно, не может быть закрыт другим участком. Только после этого проецируется второй участок поверхности причем с сохранением экрана, оставшегося после рисования первого участка.

```

DIMENSION X(41),Y(41), A(41,41), B(41,41)
DIMENSION XF1(82),YF1(41), RMAX(500), RMIN(500)
X(1) = -2.
Y(1) = -2.
DO 1 I = 2, 41
X(I) = X(I-1) + 0.1
1 Y(I) = X(I)
DO 5 I = 1, 41
DO 5 J = 1, 41
D = X(I)*X(I) + Y(J)*Y(J) - 4.
IF(D) 4, 3, 3
3 A(I,J) = 0.
GOTO 5
4 A(I,J) = SQRT(-D)
5 B(I,J) = -A(I,J)
CALL PAGE(17.,13.,0, 0, 0)
CALL REGION(2.,2.,13.,10.,0, 0, 0)
CALL MNMX(-50.,-75.,40.,X,Y, A, 41, 41,1, 41,1, 41, S)
CALL MNMX(-50.,-75.,40.,X,Y, B, 41, 41,-1, 41,1, 41, S)
CALL PRSP(-50.,-75.,40.,X,Y, A, 41, 41,1, 41,-1, 41, XF1,YF1, RMAX, RMIN, 500)
CALL PRSP(-50.,-75.,40.,X,Y, B, 41,41,-1, 41,-1, 41, XF1,YF1, RMAX, RMIN, 500)
CALL ENDPG('8.14')
END

```

8.3. Триангуляция и изображение функции двух переменных, заданной в произвольно расположенных точках

В этом параграфе будет рассмотрен случай, когда функция задана в узлах, которые не образуют регулярную прямоугольную сетку. Более того, область определения может быть неодносвязной. Тогда целесообразно, применив триангуляцию, по заданной границе области и узлам внутри нее сформировать треугольную сетку. После этого можно построить проекцию поверхности, которая описывается исследуемой функцией. В последнем случае мы неизбежно сталкиваемся с проблемой удаления невидимых линий. Для ее решения применен существенно более общий метод, чем в п. 8.1 и п. 8.2.

8.3.1. Триангуляция. Пусть на плоскости XU определена некоторая, возможно несвязная область. Ее границы заданы одной или несколькими непересекающимися замкнутыми ломаными, и внутри области задано некоторое количество точек. Задача триангуляции состоит в том, чтобы построить в этой области треугольную сетку, узлами которой служат все вершины ломаной и все заданные точки. Пример области и одно из возможных решений для нее приведены на рис. 8.15.

Известно, что при вычислениях обеспечивается более высокая точность, если каждая точка в сетке соединяется с одинаковым числом соседних точек. В идеальном случае сетка должна состоять из равносторонних треугольников. На практике при произвольном расположении точек следует избегать тупоугольных вытянутых треугольников и по возможности добиваться, чтобы треугольники, образующие сетку, были остроугольными.

Рассмотрим алгоритм триангуляции, реализованный в одной из описанных ниже программ. Пусть в области, предназначенной для триангуляции, все узловые точки перенумерованы, например так, как на рис. 8.15. Чтобы произвести разбиение области на треугольники, необходимо научиться решать задачу нахождения третьей вершины треугольника, основанием которого будет граничный отрезок. Тогда последовательным отщеплением треугольников и переопределением области можно будет решить и задачу триангуляции.

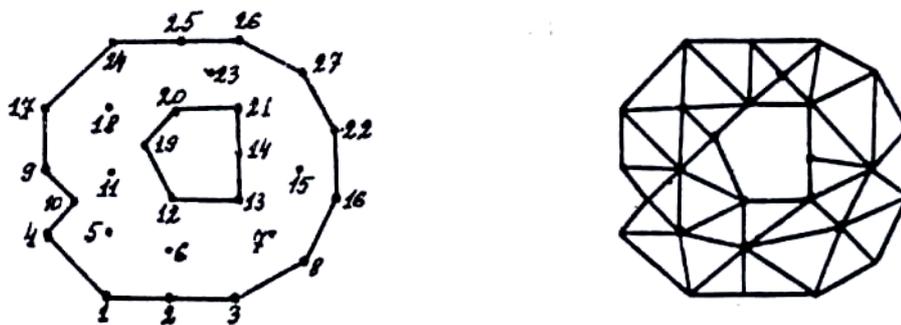


Рис. 8.15. Построение треугольной сетки в заданной области

Какая же из узловых точек области считается наиболее подходящей вершиной треугольника. Поскольку граница области предполагается ориентированной, то известно, по какую сторону отрезка должна находиться искомая точка (рис. 8.16). Среди всех точек, лежащих по указанную сторону отрезка $[A, B]$ выбирается точка K , для которой расстояние до середины отрезка минимально. При этом предпочтение отдается точкам, не попадающим в секторы α . В данном случае угол выбран таким, что $\sin \alpha = 0,2$.

Однако при таком выборе точки K в треугольник ABK могут попасть узловые точки, для которых расстояние до середины отрезка AB меньше его половины ($CK < 1/2 AB$). Если таковые имеются, они отмечаются. Точка K заменяется точкой K' , рассматривается новый треугольник и остальные пока отмеченные точки. Процесс повторяется до тех пор внутри полученного треугольника не останется узловых точек.

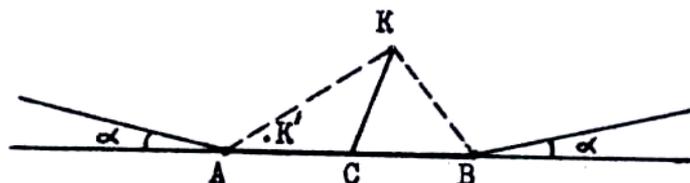


Рис. 8.16. К нахождению вершины треугольника, в основании которого - граничный отрезок

Выбор третьей вершины треугольника проводится среди так называемых доступных вершин. Первоначально в список доступных вершин включаются все узловые точки. При

последовательном отщеплении треугольников из этого списка исключаются точки, которые оказались вне текущей границы области.

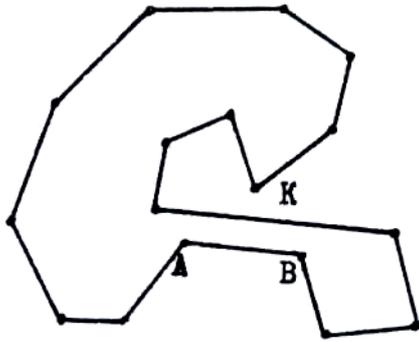


Рис. 8.17. Пример области, для которой разбиение на треугольники будет выполнено неправильно

Заметим, что последовательность граничных отрезков выбирается таким образом, что отщепленные треугольники по мере их получения образуют цепочку, скручивающуюся против часовой стрелки к центру области. Такой порядок отщепления позволяет надеяться, что если вначале граница области была не слишком изломанной, то она останется такой в процессе модификации. Это важно, поскольку алгоритм не безупречен. На рис. 8.17 приведен пример, когда разбиение будет выполнено неправильно. Для отрезка AB в качестве третьей вершины будет выбрана точка K , но полученный треугольник ABK не лежит целиком в области. Однако можно предположить, что на практике будут рассматриваться области с более или менее равномерным распределением точек и “хорошей” границей.

Рассмотрим вопросы, связанные с заданием и хранением информации об узловых точках области, ее границе и полученной треугольной сетке. Программа триангуляции TRINGL предполагает, что значения координат узловых точек или вершин находятся в массивах X и Y . Все узлы считаются перенумерованными от 1 до $N0$, где $N0$ - общее количество узлов. Будем считать, что граничный отрезок $[i, j]$ выходит из точки i и входит в точку j , если при движении из i в j область остается слева.

Для областей, ограниченных замкнутыми непересекающимися ломаными, из каждой граничной вершины будет выходить только один отрезок. Но даже в случае таких областей процесс отщепления треугольников почти всегда приводит к ситуации, в которой оставшаяся часть области будет ограничена пересекающимися замкнутыми ломаными, при этом из граничных вершин будут выходить два отрезка. В то же время три и более отрезков практически не встречаются при выбранном порядке отщепления треугольников. Учитывая это и исходя из того, что изменения при отщеплении треугольников должны быть минимальными, был выбран способ представления информации о границе.

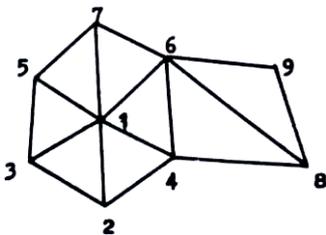
В двумерный массив IBOUND размером $(2, N0)$ заносится следующая информация о каждой узловой точке области:

а) $IBOUND(1, i) = 0, IBOUND(2, i) = 0$ - если точка i не граничная;

б) $IBOUND(1, i) = j_1, IBOUND(2, i) = 0$ или $IBOUND(1, i) = 0, IBOUND(2, i) = j_1$, если из точки i выходит один граничный отрезок (i, j_1) ;

в) $IBOUND(1, i) = j_1, IBOUND(2, i) = j_2$, или $IBOUND(1, i) = j_2, IBOUND(2, i) = j_1$, если из точки i выходят два граничных отрезка (i, j_1) и (i, j_2) .

Например, для области, изображенной ниже, массив может быть задан так:



$IBOUND(1, 1) = 0$	$IBOUND(2, 5) = 0$
$IBOUND(2, 1) = 0$	$IBOUND(1, 6) = 7$
$IBOUND(1, 2) = 4$	$IBOUND(2, 6) = 0$
$IBOUND(2, 2) = 0$	$IBOUND(1, 7) = 0$
$IBOUND(1, 3) = 0$	$IBOUND(2, 7) = 5$
$IBOUND(2, 3) = 2$	$IBOUND(1, 8) = 9$
$IBOUND(1, 4) = 0$	$IBOUND(2, 8) = 0$
$IBOUND(2, 4) = 8$	$IBOUND(1, 9) = 6$
$IBOUND(1, 5) = 3$	$IBOUND(2, 9) = 0$

Программа TRINGL($X, Y, N0, IBOUND, IDOM, NODES, NET, NT1$) позволяет разбить область на треугольники. Ее параметрами являются:

X, Y - абсциссы и ординаты узловых точек;

$N0$ - число узловых точек;

$IBOUND$ - массив описателей узловых точек (размером $(2, N0)$);

IDOM - признак вычерчивания треугольников: IDOM = 1 - вычерчиваются, IDOM = 0 - не вычерчиваются;

NODES - рабочий массив длины N0 (для списка доступных вершин);

NET - массив описателей отщепленных треугольников (длины $6*N0$);

NT1 - число отщепленных треугольников.

Следует заметить, что в результате работы программы изменяется содержимое массива IBOUND. Кроме того, если при работе программы TRINGL отщепление треугольников происходит таким образом, что в некоторый момент из какой-либо граничной вершины выходят три или более отрезков, то выдается соответствующее диагностическое сообщение. Это означает, что либо граница области с самого начала была сложной, либо узловые вершины распределены по области "плохо". Результатом работы программы TRINGL является массив описателей треугольников, которые могут быть преобразованы в информацию о сетке с помощью программы TRGRID.

Программа TRGRID(X,Y, N0, NET, IBOUND, NODES, NET0, NT1) позволяет преобразовать список описателей отщепленных треугольников в информацию о сетке. Ее параметры:

X, Y - массивы абсцисс и ординат узловых точек;

N0 - число узловых точек;

NET - массив описателей отщепленных треугольников (длины $6*N0$);

IBOUND - рабочий массив (размером $(2, N0)$);

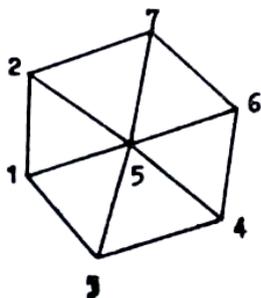
NODES - массив указателей (длины $N0 + 1$);

NET0 - массив описателей сетки (длины $6*N0$);

NT1 - число отщепленных треугольников.

Замечание. Для выпуклых областей массив NET0 достаточно задавать длиной $6*N0-2*K$, где K - число граничных точек.

Теперь опишем способ задания информации о сетке. Для каждой точки i области заводится список ее соседей в том порядке, в каком они встречаются при обходе против часовой стрелки, причем, если граничный отрезок $[j, i]$ входит в точку i , то число j заносится в список со знаком минус. Списки для каждой вершины объединяются в один общий список, хранящийся в массиве NET0. А в некоторый массив NODES заносятся указатели на начало подсписка для каждой из вершин (индексы в массиве NET0). Например, для области, изображенной ниже, информация о сетке записывается следующим образом:



NET0 = (3, 5, -2) (5, -7, 1) (4, 5, -1)
(6, 5, -3) (3, 4, 6, 7, 2, 1)
(-4, 7, 5) (-6, 2, 5) ()
NODES = 1, 4, 7, 10, 13, 19, 22, 25

В тех случаях, когда в процессе работы сетка деформируется (меняются координаты узловых точек), возникает необходимость реорганизации сетки. Такую реорганизацию позволяет выполнить программа TRIG.

Программа TRIG(X,Y, N0, NODES, NET0, IBOUND, NET, NT1) позволяет разбить область на треугольники с учетом ее предыдущего состояния. Отличие этой программы от программы TRINGL в том, что в список доступных вершин для каждого граничного отрезка заносятся его соседи по старой сетке до второго ранга. Параметры программы следующие:

X, Y - абсциссы и ординаты узловых точек (массивы длины N0);

N0 - число узловых точек;

NODES - массив указателей (длины $N0 + 1$);

NET0 - массив описателей старой сетки (длины $6*N0$);

IBOUND - рабочий массив (размером $(2, N0)$);

NET - массив описателей отщепленных треугольников (длины $6*N0$);

NT1 - число отщепленных треугольников.

Замечание. Для выпуклых областей массив NET0 достаточно задавать длиной $6*N0-2*K$, где K - число граничных узловых точек.

8.3.2. Построение центральных проекций. Рассмотрим алгоритм построения центральной проекции поверхности, заданной функцией $z = f(x, y)$, область определения которой в плоскости XU ограничена непересекающимися ломаными линиями. В области определения строится треугольная сетка, в узлах которой заданы значения функции $f(x, y)$. Эта функция приближается функцией $z = \tilde{f}(x, y)$ таким образом, что над каждым треугольником функция $f(x, y)$ заменяется линейной по трем заданным значениям. Функция $z = \tilde{f}(x, y)$ представляет собой поверхность, сложенную из треугольных элементов, их проекции и будут рассматриваться. В узловых точках значения функции $f(x, y)$ и $\tilde{f}(x, y)$ будут совпадать.

На взаимное расположение центра проекции, поверхности и картинной плоскости накладываются следующие ограничения:

- точка (XV, YV) - проекция точки зрения на плоскость XU - должна лежать вне области определения функции;
- проецируемая поверхность должна целиком находиться "перед лицом" наблюдателя, т. е. она должна полностью лежать по одну сторону от плоскости, параллельной картинной плоскости и проходящей через центр проекции.

Применяемый алгоритм удаления невидимых линий основывается на методе экранирования. Смысл этого метода заключается в том, что отмечается часть плоскости, занятая изображением, и при проведении новой линии рисуется только тот ее отрезок, который не попадает в эту уже занятую часть плоскости.

Следовательно, треугольные элементы поверхности должны быть упорядочены так, чтобы при рисовании последующие не закрывали предыдущих. Упорядочение элементов поверхности согласуется с упорядочением соответствующих треугольников в области определения функции и поэтому сначала изображается элемент, соответствующий тому треугольнику в области определения, который не закрывается никаким другим, затем - элемент, соответствующий треугольнику, который может закрываться лишь первым, и т. д. В условиях принятых ограничений на взаимное расположение центра проекции, картинной плоскости и поверхности такое упорядочение всегда существует [1, к].

Алгоритм упорядочения включает в себя алгоритм определения видимости отдельного треугольника. Будем называть сторону треугольника *условно видимой*, если угол между вектором внешней нормали и проекцией луча зрения больше 90° . В противном случае сторона называется *условно невидимой*. Треугольник ABC (рис. 8.18) с основанием AB , где AB - граничный отрезок, считается видимым и выбирается, если сторона AB условно видима, а две другие стороны либо условно невидимы, либо условно видимая сторона является граничной. При этом никакая из условно видимых граничных сторон не должна закрываться другими условно видимыми отрезками границы. Пусть выбран видимый треугольник ABC . Выбор очередного видимого треугольника производится следующим образом. Если сторона AC была условно невидимой, то в первую очередь исследуется треугольник с основанием AC . Если он не подходит, то проверяем треугольники, лежащие от него слева и справа вдоль границы. Если и эти треугольники не подходят, еще раз сдвигаемся влево и вправо и так до тех пор, пока не найдем подходящий треугольник, либо не обойдем весь контур, которому принадлежит отрезок AC . В этом случае выбираем ближайший к точке (XV, YV) граничный отрезок и выполняем для него аналогичную операцию. Если же с самого начала сторона AC была условно видимой (при выборе предыдущего треугольника), то этот процесс начинаем с отрезка CB .

После того как сделан выбор треугольника в области определения, элемент поверхности, соответствующий этому треугольнику, изображается с учетом текущего экрана. Затем экран изменяется и осуществляется переход к выбору очередного треугольника. И так до тех пор, пока не будут исчерпаны все треугольники.

8.3.3. Экранирование. По мере рисования элементов поверхности требуется запоминать ту часть картинной плоскости, которая уже занята изображением. Для проведения необходимых вычислений нужно, прежде всего, выбрать на картинной плоскости некоторую систему координат x_1, y_1 (рис. 8.19). Направим ось x_1 вправо (если смотреть от наблюдателя, стоящего вертикально в

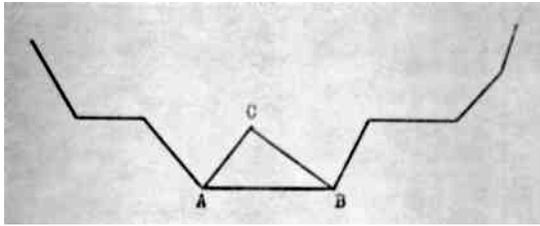


Рис. 8.18. К построению упорядоченной последовательности треугольников

центре проекции) параллельно линии пересечения картинной плоскости с плоскостью xu . Ось u_1 выберем перпендикулярной оси x_1 и направленной в сторону возрастания координаты z .

Для проведения экранирования в картинной плоскости определяется прямоугольник со сторонами, параллельными осям координат, внутри которого будет заключено изображение. Часть прямоугольника, которая уже занята изображением и в которой рисовать запрещается, запоминается следующим образом. Прямоугольник мысленно разбивается на равные части рядом вертикальных линий, назовем их *экранными линиями* (рис. 8.20), т. е. вводится физическая сетка (ср. с пп. 8.1.2, 8.2.1). На каждой экранной линии выделяются отрезки, принадлежащие той фигуре, которую нужно запомнить. Набор таких отрезков и является экраном, характеризующим эту фигуру.

Изображение поверхности строится последовательно по отдельным треугольникам в том порядке как описывалось выше. При рисовании каждого треугольника вычерчиваются те его стороны, которые до этого еще не рисовались (каждое ребро принадлежит двум треугольникам, но вычерчивается оно всего один раз). Вычерчивание происходит при неизменном экране, затем

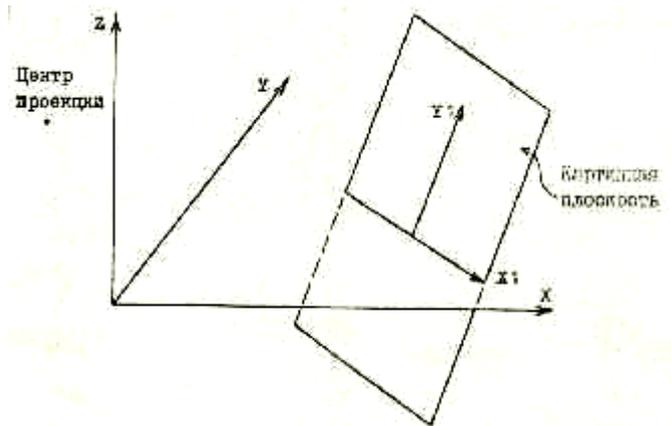


Рис. 8.19. Выбор системы координат на картинной плоскости

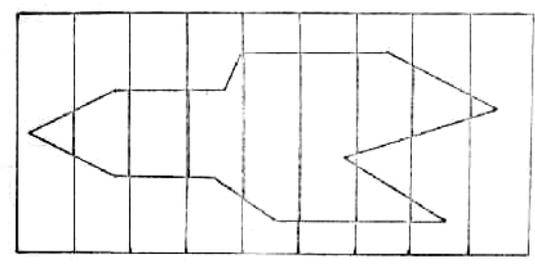


Рис. 8.20. Экранные линии. Экран

экран модифицируется. В экран заносятся отрезки экранных линий, которые оказываются внутри спроецированного треугольника. При этом, если необходимо, происходит склеивание нескольких отрезков в один.

Для запоминания экрана служат два массива ISCR и SCRN. Массив SCRN трехмерный, граничными значениями его индексов являются следующие величины: количество экранных линий, характеризующее частоту физической сетки, число отрезков на одной экранной линии и число 2. Для оценки максимального числа отрезков, которое может встречаться на экранных линиях, поступаем следующим образом [1, к]. Спроецируем центр проекции на плоскость xu и подсчитаем, сколько раз луч, выходящий из построенной точки, может пересекать область определения, и к результату прибавим единицу. Число отрезков на каждой экранной линии не будет превышать полученной величины. Например, для области, показанной на рис. 8.21, число отрезков на экранной линии не превысит трех для центра проекции в точке 1 и двух - в точке 2.

Число 2 в описании массива SCRN обусловлено тем, что запоминаются начало и конец отрезка. Поэтому k -й отрезок на i -й экранной линии записывается в этом массиве так: его координата Y меняется в пределах $SCRN(I, K, 1) \leq Y \leq SCRN(I, K, 2)$. Другой массив, ISCR, одномерный, его длина определяется количеством экранных линий. В ISCR(I) запоминается число отрезков на i -й экранной линии. Перед началом рисования массив ISCR обнуляется.

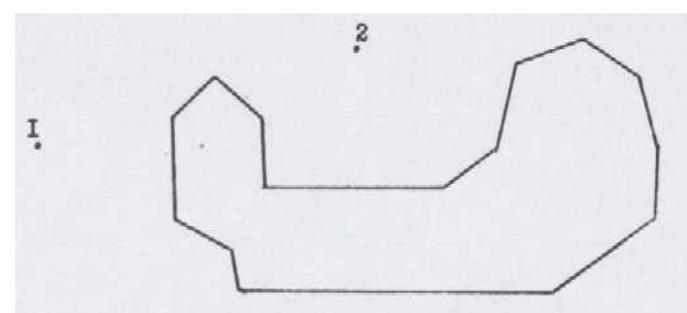
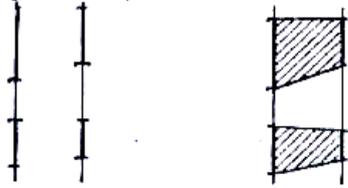


Рис. 8.21. К определению максимального числа отрезков экранной линии

При изображении очередного треугольника приходится искать точки пересечения его ребер с границей фигуры, внутри которой рисовать запрещено. Поэтому надо уметь восстанавливать границу фигуры по ее экрану.

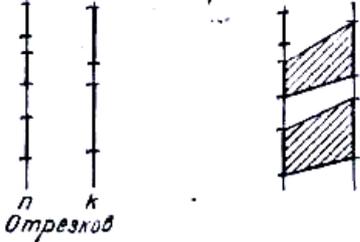
Чтобы понять как это делается, достаточно рассмотреть случай с двумя соседними экранными линиями.

а) Число отрезков на них одинаково



Тогда нижние отрезки соединяются между собой, верхние - между собой, остальные - аналогично.

б) Число отрезков различно



Построение границы идет как в а), только на той линии, где отрезков больше, $(n-k)$ наименьших по длине отрезков в построении не участвуют.

Описать коротко сам процесс изображения отрезка с учетом экрана можно следующим образом. Будем говорить, что характеристика видимости точки на картинной плоскости равна 1, если точка лежит вне экрана, и 0 в противном случае. Для того чтобы нарисовать отрезок, нужно найти точки, где меняется характеристика видимости. Эти точки ищутся так. Экранными линиями весь отрезок разбивается на меньшие отрезки, для каждого из них определяются характеристики видимости его концов. Если они различны, то ищется точка пересечения отрезка с граничной линией экрана, в противном случае исследуется следующий малый отрезок и т. д.

8.3.4. Описание программ.

Программа TRSURF(X,Y,Z,N0,XV,YV,NODES,NET0,ISCR,SCRN,IDIM1,IDIM2,NET,IBOUND) управляет работой нескольких подпрограмм, с помощью которых находит последовательно треугольник за треугольником в области определения в том порядке, в каком соответствующие им элементы поверхности должны изображаться, проецирует эти элементы, рисует их с учетом текущего состояния экрана, изменяет экран. Если в процессе проведения вычислений число отрезков на одной экранной линии превысит заданную величину IDIM2, то выдается диагностическое сообщение об ошибке. Программа имеет следующие параметры:

- X, Y, Z - массивы координат точек;
- N0 - количество точек;
- XV, YV - координаты центра проекции;
- NODES - массив указателей (длины N0+1);
- NET0 - массив описателей сетки (длины 6*N0);
- ISCR, SCRN - массивы описателей экрана (размером соответственно IDIM1 и (IDIM1,IDIM2,2));
- IDIM1 - количество экранных линий;
- IDIM2 - максимальное число отрезков на экранной линии;
- NET - рабочий массив (длины 6*N0);
- IBOUND - массив описателей условно видимых отрезков размером (2, N0).

Обращению к программе TRSURF должно предшествовать обращение к программе PREP, выполняющей ряд подготовительных операций.

Программа PREP(X,Y,Z,N0,XV,YV,ZV,XPL,YPL,ZPL,VX,VY,VZ,ISCR,IDIM1,RLYTOX) формирует матрицу однородных координат, с помощью которой будет выполняться проецирование, и устанавливает пределы изменения функции $f(x,y)$ на картинной плоскости. Ее параметры:

- X, Y, Z - массивы координат точек;
- N0 - количество точек;
- XV, YV, ZV - координаты центра проекции;
- XPL, YPL, ZPL - координаты точки на картинной плоскости;
- VX, VY, VZ - координаты вектора, перпендикулярного картинной плоскости;
- ISCR - рабочий массив характеристик экранных линий (длины IDIM1);
- IDIM1 - количество экранных линий;

RLYTOX - коэффициент, определяющий форму математической области значений функции на картинной плоскости; он равен отношению $(X_{MAX} - X_{MIN}) / (Y_{MAX} - Y_{MIN})$, где X_{MIN} , X_{MAX} , Y_{MIN} , Y_{MAX} - пределы изменения функции, устанавливаемые в области рисования.

Замечание. Проекция точки зрения $(XV, YV, 0)$ на плоскость XY должна находиться вне области определения. Вектор (VX, VY, VZ) должен быть направлен в полупространство, где находится центр проекции. В противном случае будет получено зеркальное отражение поверхности.

8.3.5. Примеры. Следующие рисунки получены с помощью программ, описанных в этом параграфе.

1. На рис. 8.22 показана простая поверхность (поверхность Каталана), которая формируется и изображается с помощью следующей программы:

```
PARAMETER(R = 2., N = 21, K = 18, ID1 = 200, ID2 = 2)
DIMENSION X(N), Y(N), Z(N), NET0(6*N + 2*K), NET(6*N)
DIMENSION SCRN(ID1, ID2, 2), ISCR(ID1), NODES(N + 1), IB(2, N)
DATA XV, YV, ZV/8., 5., -1./, VX, VY, VZ/8., 5., -1./
DATA XPL, YPL, ZPL, IDOM/3*0., 1/
DTH = 2.*3.14159/K
```

C ГРАНИЦА ОБЛАСТИ

```
DO 1 I = 1, K
  IB(1, I) = I + 1
  IB(2, I) = 0
  X(I) = R * COS(DTH * (I - 1))
  Y(I) = R * SIN(DTH * (I - 1))
```

1 Z(I) = 0.

```
  IB(1, K) = 1
```

C ВНУТРЕННИЕ ТОЧКИ

```
DO 2 I = 1, N - K
```

```
  X(I + K) = I - 2
```

```
  Y(I + K) = 0.
```

```
  Z(I + K) = 3.
```

```
  IB(1, I + K) = 0
```

2 IB(2, I + K) = 0

```
CALL PAGE(17., 26., 0, 0, 0)
```

```
CALL LIMITS(-2.1, 2.1, -2.1, 2.1)
```

```
CALL REGION(3., 1., 11., 11., 0, 0, 0)
```

```
CALL TRINGL(X, Y, N, IB, IDOM, NODES, NET, NT1)
```

```
CALL TRGRID(X, Y, N, NET, IB, NODES, NET0, NT1)
```

```
CALL REGION(3., 13., 11., 8., 0, 0, 0)
```

```
CALL PREP(X, Y, Z, N, XV, YV, ZV, XPL, YPL, ZPL, VX, VY, VZ, ISCR, ID1, RLTOX)
```

```
CALL TRSURF(X, Y, Z, N, XV, YV, NODES, NET0, ISCR, SCRN, ID1, ID2, NET, IB)
```

```
CALL ENDPG('8.22')
```

```
END
```

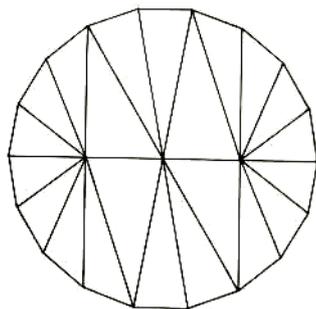
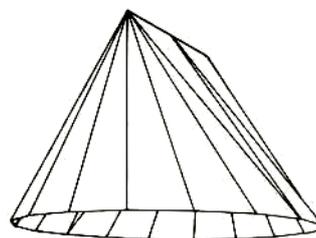


Рис. 8.22. Изображение поверхности Каталана

Программа написана с использованием расширенного фортрана, транслировалась на машине БЭСМ-6 с помощью компилятора Форекс.

2. На рис. 8.23 показаны более сложные поверхности:

а) область определения функции не является прямоугольной (как и в предыдущем примере);

б) неодносвязная область задания функции.

8.3.6. Служебные программы.

Программа TRG(L, IR, K, KL, KR, NET, NODES, IBOUND, NB, X, Y, XV, YV, IND) позволяет определить, закрывается ли чем-нибудь треугольник, в основании которого лежит граничный отрезок [L, IR]. Параметр IND = 1, если треугольник полностью видим, в противном случае IND = 0.

Программа NEXTRG(BOUND, X, Y, NET, NODES, NB, XV, YV) позволяет найти следующий треугольник, подлежащий рисованию и изменить сетку после его отщепления.

Программа PROJCT(XP, YP, ZP, XPJ, YPJ) позволяет спроецировать точку (XP, YP, ZP). Здесь (XPJ, YPJ) - координаты проекции.

Программа MATEVL(XV,YV,ZV,XPL,YPL,ZPL,VX,VY,VZ) вычисляет матрицу, с помощью которой осуществляется проецирование (эта матрица несколько модифицируется в программе PREP).

Программа SEE(L, IR, IBOUND,X,Y, NB, IND, IDRWN,XV,YV) определяет, закрывается ли отрезок [L, IR] одним из условно видимых граничных отрезков, информация о которых хранится в массиве IBOUND. Кроме того, определяется, рисовался ли этот отрезок. Параметр IND = 1, если отрезок [L, IR] видим полностью, и IND = 0, если закрывается хотя бы частично. Параметр IDRWN = 1, если отрезок [L, IR] нарисован, в противном случае IDRWN = 0.

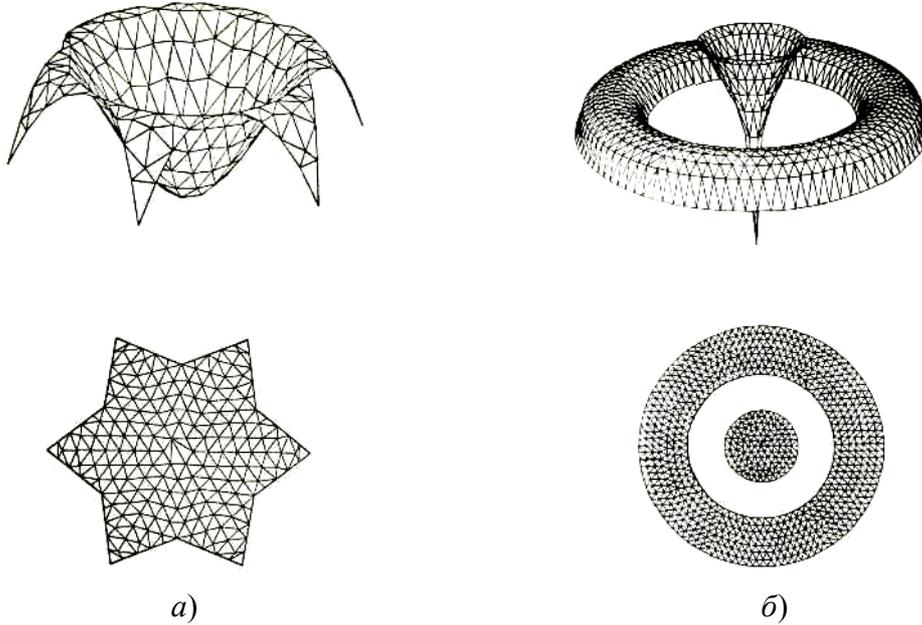


Рис. 8.23. Примеры изображений более сложных поверхностей

Программа SEE1(I1, I2, I3, I4, X,Y, XV,YV, IND) определяет, закрывается ли отрезок [I1, I2] отрезком [I3, I4], если смотреть из точки (XV, YV). Параметр IND = 1, если [I1, I2] полностью видим, и IND = 0, если [I1, I2] закрывается хотя бы частично.

Программа ILIMTN(L, IR, IBOUND, NB) позволяет исключить отрезок [L, IR] из списка условно видимых отрезков границы.

Программа LCROSS(X1,Y1, X2,Y2, X3,Y3, X4,Y4, C, D) вычисляет координаты точки пересечения двух отрезков [A1, A2] и [A3, A4] и заносит их в параметры C и D.

Программа SEGANG(X1,Y1, X2,Y2, A, STEP, IND) позволяет вычислить для отрезка [A1, A2] его угловой коэффициент A. Если отрезок почти вертикальный, то в параметр A заносится координата его пересечения с экранной линией (по выбранному критерию вертикальности такая линия будет единственной). В этом случае IND = 1.

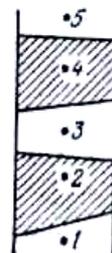
Программа SCREN1(ISCR, SCRN, IDIM1, IDIM2,Y, N, IND) позволяет определить, закрыта ли экраном точка с координатой Y, находящаяся на N-й экранной линии. Параметр IND = 1, если точка видима, в противном случае IND = 0.

Программа SCREN2(ISCR, SCRN, IDIM1, IDIM2, X,Y, NL, STEP, XBEG, A1, B1, A2, B2, IND) определяет положение точки (X, Y) относительно экрана.

A1 - угловой коэффициент граничного отрезка, отделяющего точку (X, Y) снизу. (XBEG, B1) - точка этого отрезка;

A2 - аналогично для верхнего отрезка. (XBEG, B2) - точка верхнего отрезка.

Для точки 1	IND = 1
для точки 2, 4	IND = 2
для точки 3	IND = -2
для точки 5	IND = 3



Программа TRDRAW(ISCR, SCRN, IDIM1, IDIM2, X, Y, Z) позволяет нарисовать элемент поверхности, соответствующий треугольнику L, IR, K, и модифицировать экран.

Программа PICTUR(ISCR, SCRN, IDIM1, IDIM2, X1, Y1, X2, Y2) позволяет начертить отрезок [A1, A2], где A1 = (X1, Y1), A2 = (X2, Y2).

Программа SCRMOD(ISCR, SCRN, IDIM1, IDIM2, XL, YL, XIR, YIR, XK, YK) изменяет текущее состояние экрана, добавляя к нему треугольник с координатами (XL, YL), (XIR, YIR), (XK, YK).

Программа PNORDR(X, Y, NET, L, IB, IE) упорядочивает соседей L-й точки на участке от IB до IE так, чтобы при перечислении их в NET обход осуществлялся против часовой стрелки.

Функция COSIN1(X1, Y1, X2, Y2) вычисляет скалярное произведение двух векторов.

Функция COSIN2(X1, Y1, X2, Y2) вычисляет $SIGN(\cos \varphi) * \cos^2 \varphi$, где φ - угол между векторами (X1, Y1) и (X2, Y2).

8.4. Изображение трехмерных объектов с использованием эффекта ореола

В этом параграфе рассматривается еще один алгоритм удаления невидимых линий, позволяющий распространить на векторные устройства хорошо известный в растровой графике метод приоритетов (см. [1, л]). Этот алгоритм позволяет получать центральные и параллельные проекции произвольных графических объектов, представленных в виде неупорядоченной совокупности отрезков прямых. Для создания иллюзии глубины используется метод частичного удаления невидимых линий, основанный на *эффекте ореола*.

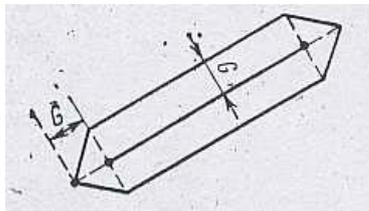


Рис. 8.24. Ореол вокруг проекции отрезка (G - ширина разрыва)

результате создания вокруг близлежащего отрезка ореола - некоторой воображаемой непрозрачной области определенной конфигурации (например, имеющей форму шестиугольника, см. рис. 8.24).

8.4.1. Особенности метода ореола. Основная идея метода ореола заключается в следующем. Если в трехмерном пространстве некоторый отрезок находится впереди другого отрезка прямой, то при проецировании на картинную плоскость более удаленный отрезок изображается с разрывом. Можно считать, что разрыв получается в

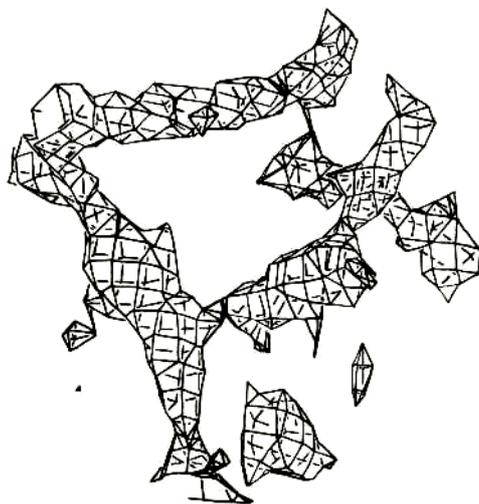


Рис. 8.25. Изображение поверхностей уровня плотности пространственного распределения галактик, построенное с использованием метода ореола

Главное отличие метода ореола от других способов удаления невидимых линий состоит в том, что этот метод позволяет изображать неструктурированную графическую информацию. Входные данные для алгоритма окружения ореолом достаточно представить в виде неупорядоченной совокупности отрезков прямых, описывающих изображаемый объект, т. е. практически в той же форме, что и для каркасного изображения. На рис. 8.25 и 8.26 приведены примеры изображений, построенных с использованием метода ореола.

Как видно из рис. 8.25, метод ореола, вообще говоря, не позволяет полностью стереть невидимые линии. Однако, когда изображаемые отрезки прямых или кривые находятся на более близком расстоянии друг от друга, чем заданная *ширина разрыва G* (рис. 8.24), то в этом случае достигается эффект близкий к эффекту полного удаления невидимых линий (рис. 8.26). От ширины разрыва, таким образом, в значительной степени зависит "чистота" удаления невидимых линий. Влияние ширины разрыва особенно хорошо проявляется в тех случаях, когда изображаемый объект представляет собой криволинейную сетку, размер ячеек которой меньше значения этого параметра. В некоторых случаях, однако, изображения, получаемые с помощью метода ореола, оказываются более наглядными, чем при полном удалении невидимых линий. Кроме того, имеется возможность получить некое представление и о невидимой части объекта.

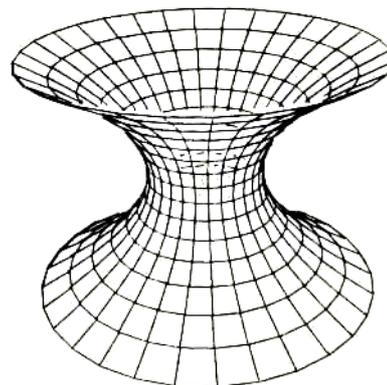


Рис. 8.26. Пример проекции алгебраической поверхности

При реализации метода ореола в рассмотрение также вводится параметр *допуска по глубине TOL* . Необходимость его введения диктуется следующими соображениями. В довольно типичном случае, когда в пространстве пересекаются два отрезка (например, при задании сеток), между ними из-за приближенности вычислений может образоваться маленький зазор. В результате отрезки будут обрабатываться не как пересекающиеся, а как скрещивающиеся, что может привести к серьезным искажениям в изображении. Если же считать, что при пересечении двух отрезков один из них, скажем, первый, закрывает второй отрезок только в тех случаях, когда первый отрезок находится впереди второго на расстоянии не меньшем, чем некоторое заданное расстояние TOL , то ошибок такого рода можно будет избежать.

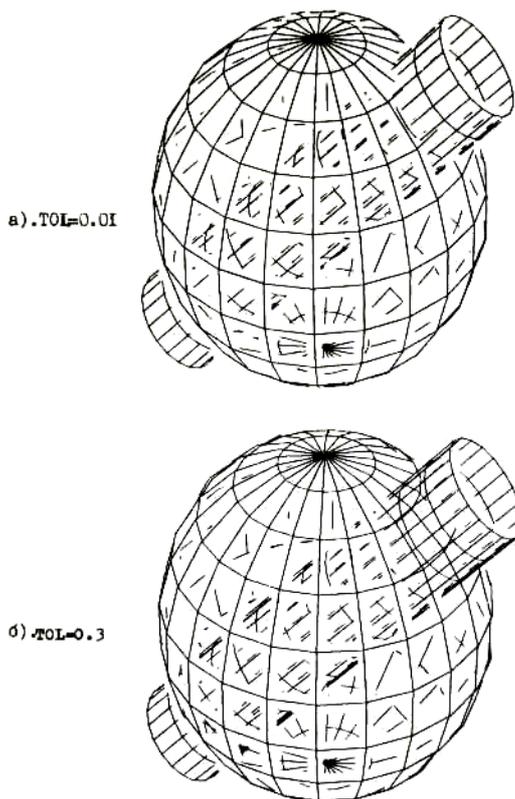


Рис. 8.27. Рисунки, показывающие влияние параметров допуска по глубине TOL на качество изображения (в результате увеличения значения TOL становятся видимыми некоторые отрезки или части отрезков, ранее скрытые от наблюдателя)

Кроме того, вводя допуск по глубине, мы получаем в свое распоряжение параметр, с помощью которого в некоторых пределах можно управлять прозрачностью изображаемого объекта (рис. 8.27).

8.4.2. Описание метода ореола. Как мы уже упоминали, входная информация для алгоритма окружения ореолом должна быть представлена в виде совокупности отрезков прямых. Эти отрезки

следует задавать координатами их крайних точек $DX(K, L)$, $DY(K, L)$, $DZ(K, L)$, где L - номер отрезка, K - индексы массива: $K = 1$ для одной крайней точки и $K = 2$ для другой.

Данные для алгоритма окружения ореолом удобно представлять в системе координат картинной плоскости. Для построения этой координатной системы можно воспользоваться практически любой схемой трехмерного проецирования. Допустимы и центральные проекции на картинную плоскость, перпендикулярную лучу зрения, проведенному из центра проекции в начало координат, и аксонометрические проекции на плоскость, перпендикулярную направлению проецирования. Без ограничения общности можем считать, что картинная плоскость всегда проходит через начало координат.

Систему координат картинной плоскости XYZ будем строить следующим образом. Пусть задана некоторая правая система координат $X'Y'Z'$. Будем называть ее объектной системой координат. Направление оси Y системы координат картинной плоскости выберем совпадающим с направлением проекции на картинную плоскость орта оси Y' объектной системы координат (рис. 8.28). Направляющий вектор оси Z в случае центральной проекции определим как вектор, начало которого лежит в точке с координатами $(0, 0, 0)$, а конец - в центре проекции. В случае аксонометрической проекции будем считать, что ось Z направлена противоположно вектору проецирования. И, наконец, орт оси X дополнит два предыдущих вектора до правой тройки.

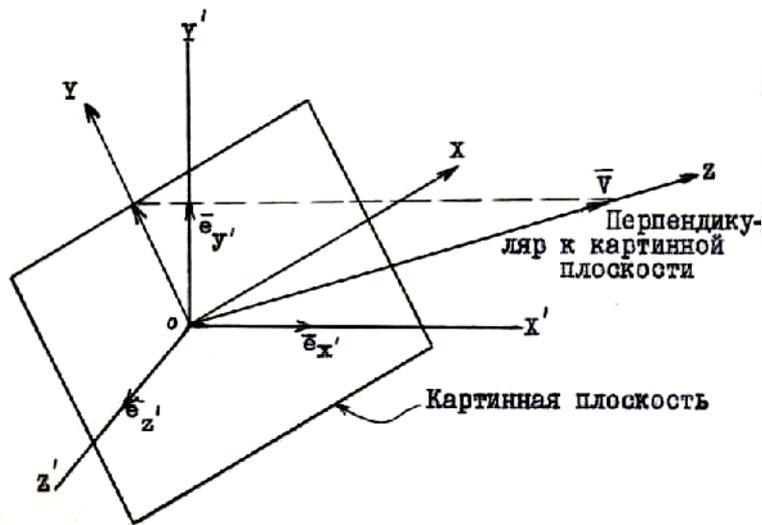


Рис. 8.28. Схема построения системы координат картинной плоскости для случая центральной проекции (здесь \bar{V} - центр проекции, а $\bar{e}_x, \bar{e}_y, \bar{e}_z$ - орты объектной системы координат)

Работу алгоритма окружения ореолом можно упрощенно представить следующим образом. Для каждого из отрезков, составляющих изображаемый объект (пусть его номер I), отбираются все отрезки (с номерами $I \neq J$), ореолы вокруг которых влияют на видимость I -го отрезка. Это означает, что проекции на картинную плоскость I -го и J -го отрезков пересекаются и J -й отрезок находится впереди I -го (относительно наблюдателя) на расстоянии большем, чем значение параметра TOL . Затем проводится отсечение проекции I -го отрезка по шестиугольным областям ореолов, окружающих отрезки с номером J . Части проекции I -го отрезка, которые попадают

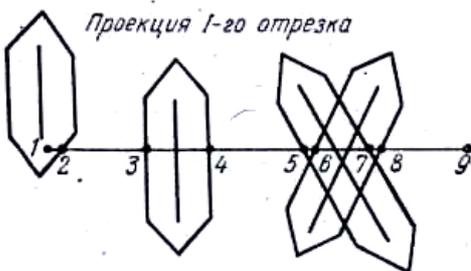


Рис. 8.29. Видимые и невидимые части проекции I -го отрезка. (Части 1-2, 3-4, 5-8 невидимы, остальные видимы)

внутри этих ореолов, считаются невидимыми, а остальные участки - видимыми (рис. 8.29). На проекции каждого отрезка допускается не более 15 отдельных интервалов невидимости. Всякий последующий (сверх 15) невидимый участок проекции отрезка будет склеиваться с ближайшим к нему интервалом невидимости и, следовательно, соответствующие видимые участки рисоваться не будут.

8.4.3. Повышение эффективности алгоритма окружения ореолом. Поскольку время работы алгоритма окружения ореолом в общем случае пропорционально квадрату числа отрезков, составляющих графический объект, важное значение имеют приемы, позволяющие уменьшить это время. Разработано несколько различных

способов сокращения времени вычислений, причем пользователю предоставлена возможность выбрать тот или иной из них.

Если все множество отрезков можно представить в виде двух или более независимых подмножеств, разделенных плоскостями, то упорядочение этих подмножеств можно использовать для сокращения времени, затрачиваемого на вычисления. Пусть, например, задана сцена, состоящая из нескольких изолированных объектов (рис. 8.30). Тогда для некоторого фиксированного центра или направления проецирования первый объект можно будет изобразить независимо от второго и третьего объекта, а второй - независимо от третьего. То есть отрезки, образующие первый объект, могут быть изображены после сравнений с отрезками, составляющими один только первый объект, отрезки, образующие второй объект, следует сравнивать с отрезками, составляющими первый и второй объекты, и только отрезки третьего объекта необходимо сравнивать со всеми отрезками сцены. В результате можно добиться довольно существенного сокращения времени, затрачиваемого на вычисления (см. пример 2 в п 8.4.5).

Существуют и другие возможные способы сокращения времени вычислений. Так, если отрезки, составляющие графический объект, удастся упорядочить таким образом, что проекция каждого последующего отрезка будет закрываться только ореолами проекций предыдущих отрезков, то проекцию каждого отрезка можно проверять на пересечение только с ореолами проекций тех отрезков, порядковые номера которых меньше, чем у рассматриваемого отрезка. (Подобное упорядочение применяется во многих алгоритмах удаления невидимых линий, предназначенных для изображения объектов, описываемых однозначными и непрерывными функциями двух переменных). В результате процессорное время, затрачиваемое на вычисления, уменьшается приблизительно вдвое.

С целью сокращения времени вычислений реализовано также автоматическое разбиение всего множества отрезков на два или четыре независимых подмножества с помощью одной секущей плоскости или двух взаимно перпендикулярных секущих плоскостей. При этом все отрезки, пересекающиеся с плоскостями, выделяются в отдельное (зависимое) подмножество. Секущие плоскости проходят через орт оси Y' (соответственно X') объектной системы координат, восстановленной из точки с координатами (C_x, C_y, C_z) , и вектор направления проецирования, помещенный в точку (C_x, C_y, C_z) , в случае параллельной проекции или вектор, соединяющий точку (C_x, C_y, C_z) с центром проекции, в случае центральной проекции (рис. 8.31). Точка (C_x, C_y, C_z) представляет собой либо центр тяжести графического объекта либо некоторую заданную пользователем точку пространства. Центр тяжести объекта вычисляется по формулам:

$$C_x = \frac{1}{2N} \sum_{I=1}^N \sum_{J=1}^2 DX(J, I); \quad C_y = \frac{1}{2N} \sum_{I=1}^N \sum_{J=1}^2 DY(J, I);$$

$$C_z = \frac{1}{2N} \sum_{I=1}^N \sum_{J=1}^2 DZ(J, I);$$

где N - число отрезков, составляющих выбранный графический объект. При проведении проверок на видимость зависимое подмножество, объединяется с каждым из независимых подмножеств.

В некоторых случаях может оказаться удобным добавить к графическому объекту дополнительные отрезки, которые не будут изображаться, но будут закрывать другие отрезки, т. е. будут принимать участие только в проверках на видимость. Программные средства, реализующие

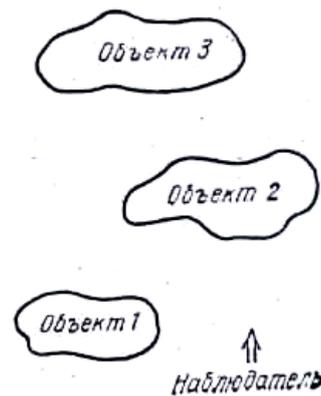


Рис. 8.30. Пример сцены, которая допускает разбиение всего множества отрезков на независимые подмножества

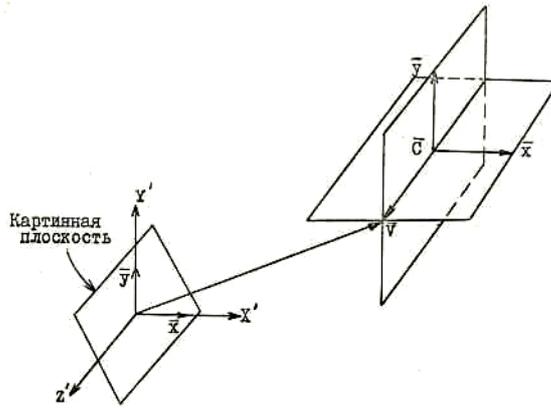


Рис. 8.31. Разбиение сцены на независимые подмножества (случай центрального проецирования).

Обозначения: \bar{x} , \bar{y} - орты соответственно осей X' и Y' объектной системы координат. $\bar{C}(C_x, C_y, C_z)$ -

либо "центр тяжести" графического объекта, либо некоторая другая заданная точка, \bar{V} - центр проекции

алгоритм, дают возможность отдельно указать отрезки, которые будут изображаться, и отрезки, которые будут принимать участие в проверках на видимость. Добавив к последнему подмножеству некоторое дополнительное количество "фиктивных" отрезков, можно добиться более полного эффекта удаления невидимых линий.

8.4.4. Программная реализация алгоритма окружения ореолом. При построении изображений с использованием метода ореола сохраняется обычный принятый в Графоре порядок работы: открытие страницы, заведение области и т. д. Сразу после определения области рисования необходимо задать режим проецирования, т. е. центр проекции или направление проецирования на картинную плоскость, перпендикулярную лучу зрения. Режим проецирования определяется с помощью обращения к программе PRJHL, которая формирует соответствующую матрицу преобразований размером 4×4 .

Для задания пределов изменения координат X и Y на картинной плоскости служит программа VPLIM. Она выполняет те же функции, что и программы TDLIM и MNMX (см. п. 8.1.2, 8.2.3), т. е. действует следующим образом. Сначала вычисляются максимальные и минимальные значения координат среди крайних точек проекций отрезков, составляющих подмножество, размер которого (т. е. число отрезков в нем) определяется самим пользователем. Далее выбирается одна из следующих возможностей: либо задаются пределы изменения, равные найденным, либо в качестве пределов изменения координат принимаются значения, общие среди найденных и пределов, полученных при предыдущих обращениях к этой программе. Используя приведенные возможности, можно в одной области рисования получать изображения сразу нескольких графических объектов с учетом их взаимного расположения в пространстве.

После того, как установлен режим проецирования и определены пределы изменения координат, можно спроецировать графический объект или некоторую его часть на картинную плоскость. Для этой цели используется программа COORDT. В результате работы этой программы формируются массивы значений координат крайних точек отрезков, спроецированных на картинную плоскость. Отметим, что поскольку графические объекты, как правило, состоят из достаточно большого числа отрезков, координаты проекций крайних точек отрезков заносятся в исходные массивы DX , DY , DZ . Таким образом, после обращения к программе COORDT первоначальные массивы DX , DY , DZ со значениями координат отрезков будут изменены, поскольку на то же место будут записаны значения координат для проекций отрезков.

При обращении к программе COORDT необходимо указать способ изображения совокупности отрезков, описывающих графический объект. Здесь возможны восемь случаев. В первом из них, самом общем, проекция каждого отрезка проверяется на пересечение с ореолами проекций всех остальных отрезков, в то время как во втором случае - только с теми отрезками, порядковые номера которых меньше, чем у рассматриваемого отрезка (см. п. 8.4.3).

В остальных шести случаях выполняется автоматическое разбиение всего множества отрезков на два или четыре независимых подмножества и одно зависимое. Первые три случая отличаются от последних трех только выбором точки (C_x, C_y, C_z) (см. п. 8.4.3): либо эта точка является центром тяжести объекта, либо она задается явно.

В случаях 3, 6 и 4, 7 разбиение выполняется одной секущей плоскостью, проходящей через орт оси Y (случаи 3, 6) или оси X (случаи 4, 7) объектной системы координат, в то время как в случаях 5 и 8 разбиение производится сразу обеими этими плоскостями. Явное задание точки (C_x, C_y, C_z) в случаях 6-8 производится с помощью программы SPLITP, обращение к которой должно предшествовать вызову программы COORDT.

В общем случае при построении изображений целесообразно проводить разбиение сразу обеими секущими плоскостями (относительно "центра тяжести" объекта либо относительно некоторой другой удобной точки пространства). При этом нам выгодна ситуация, когда зависимое подмножество содержит минимальное количество отрезков. Однако в случае вытянутых объектов или объектов, состоящих из "длинных" отрезков, в состав зависимого подмножества может входить чересчур много отрезков (в предельном случае оно будет содержать все отрезки объекта в то время как независимые подмножества будут пусты). Поэтому иногда вариант с одной секущей плоскостью может оказаться предпочтительнее.

Две программы, HALLNE и HALOED, предназначены собственно для построения графических объектов с удалением невидимых линий. Эти программы позволяют получить изображение всего графического объекта или некоторой его части восемью различными способами (см. описание программы COORDT). Различие между этими программами заключается в том, что при использовании программы HALLNE отпадает необходимость в предварительном вызове программы COORDT. То есть программу HALLNE можно рассматривать как объединение двух программ: COORDT и HALOED. Однако программа HALOED позволяет воспользоваться упорядочением независимых подмножеств отрезков, на которые разбивается все множество отрезков (если, конечно, такое разбиение возможно), или, например, ввести в рассмотрение отрезки, которые будут участвовать в проверках на видимость, но не будут изображаться (см. п. 8.4.3). Это достигается тем, что в описании параметров программы HALOED отдельно указывается диапазон отрезков, которые будут изображаться, и отдельно совокупность отрезков, которые будут участвовать в проверках на видимость.

В программах HALLNE и HALOED в качестве параметров задаются значения ширины разрыва G и допуска по глубине TOL . Как мы уже отмечали, конкретные значения этих параметров выбираются из эстетических соображений. При этом следует иметь в виду, что в случаях, когда ореолы соседних ребер элемента сетки, описывающей поверхность, закрывают весь или почти весь этот элемент, возможно пропадание отдельных отрезков или их частей на перегибах поверхности, ее краях и т. д. (см., например, рис. 8.27). Тем не менее использование достаточно больших (относительно элемента сетки) значений параметра G во многих случаях позволяет получить изображения эстетически даже более качественные (и с более полным удалением невидимых линий), чем при меньших значениях G (ср., например, рис. 8.27 и приведенный ниже рис. 8.32).

В качестве первого приближения рекомендуется выбирать параметр G таким образом, чтобы ореолы вокруг ребер наиболее типичных элементов поверхности закрывали эти элементы не полностью. Значение TOL вначале можно взять равным 0.01 - 0.1 от значения G . При таком выборе этих параметров возможные искажения на краях изображения будут сведены к минимуму, а просматриваемый задний план будет способствовать выбору наилучшего ракурса. После первой прикидки можно будет уже более уверенно подбирать значения параметров G и TOL , добиваясь желаемого качества изображения.

При работе с графическими объектами, заданными в виде последовательности отрезков прямых, в некоторых случаях может оказаться полезной возможность построить изображение всего графического объекта или некоторой его части без удаления невидимых линий. Для этой цели предназначена программа DRWHL, которая строит заданную проекцию объекта без использования метода ореола.

Программа PRJHL(IT, X, Y, Z) позволяет задать режим проецирования. Ее параметрами являются: IT - признак проекции: $IT = 0$ - параллельная проекция, $IT = 1$ - центральная проекция; X, Y, Z - координаты вектора направления проецирования, если выбрана параллельная проекция, или координаты центра проекции, если выбрана центральная проекция.

Замечание. При выборе положения центра проекции необходимо следить, чтобы он лежал за пределами параллелепипеда, описанного вокруг изображаемого графического объекта.

Программа VPLIM($DX, DY, DZ, NLINES, NL1, NL2, S$) предназначена для задания пределов изменения координат проекций на картинную плоскость крайних точек отрезка. Установленные

пределы сохраняются до очередного обращения к программе VPLIM. Параметрами этой программы являются:

DX, DY, DZ - массивы значений координат крайних точек отрезков (размером (2, NLINES));

NLINES - число отрезков в описании графического объекта;

NL1 - номер первого отрезка подмножества, для которого ищутся пределы изменения координат:

$NL1 > 0$ - в области рисования устанавливаются пределы, найденные для данного подмножества отрезков,

$NL1 < 0$ - в области рисования устанавливаются пределы, общие среди тех, которые были найдены для данного подмножества отрезков, и тех, которые были получены при предыдущих обращениях к программе VPLIM;

NL2 - номер последнего отрезка подмножества, для которого ищутся пределы изменения координат, $|NL1| \leq NL2$;

S - коэффициент, определяющий форму математической области значений координат на картинной плоскости, он равен отношению $(XMAX - XMIN)/(YMAX - YMIN)$, где XMAX, XMIN, YMAX, YMIN - пределы, установленные в области рисования (об использовании S см. п. 8.1.2)

Программа COORDT(DX, DY, DZ, NLINES, N1, N2, NCLUST, NLNDR, ITYP) позволяет спроецировать на картинную плоскость заданное количество отрезков. Параметры программы:

N1, N2 - номера первого и последнего отрезков, проецируемых на картинную плоскость;

NCLUST - рабочий массив, используемый при разбиении всего множества отрезков на независимые подмножества (длины NLNDR);

NLNDR - длина массива NCLUST: $NLNDR \geq N2 - N1 + 1$, если ITYP = 2, и $NLNDR = 1$ в остальных случаях;

ITYP - параметр, указывающий способ изображения заданной совокупности отрезков, определяющей графический объект:

ITYP = 1 - (самый общий способ) каждый отрезок проверяется на видимость относительно всех остальных отрезков,

ITYP = 2 - (упорядоченные отрезки) каждый отрезок проверяется на видимость только по отношению к тем отрезкам, порядковые номера которых меньше, чем у рассматриваемого отрезка,

ITYP = 3, 4 - выполняется автоматическое разбиение всего множества отрезков на два независимых подмножества (разбиение выполняется плоскостью, проходящей через орт оси Y' (соответственно оси X') объектной системы координат, восстановленный из "центра тяжести" объекта),

ITYP = 5 - выполняется автоматическое разбиение всего множества отрезков на четыре независимых подмножества, (разбиение выполняется двумя плоскостями, проходящими через орты осей X' и Y' объектной системы координат, восстановленными из "центра тяжести" объекта),

ITYP = 6, 7, 8 - то же, что и при ITYP = 3, 4, 5, только орты осей координат восстанавливаются не из "центра тяжести", а из точки, координаты которой задаются с помощью программы SPLITP (обращение к этой программе должно предшествовать вызову COORDT).

Значения остальных параметров те же, что и в программе VPLIM.

Замечание. В результате работы этой программы изменяются первоначальные значения массивов координат DX, DY, DZ.

Программа SPLITP(X, Y, Z) служит для задания координат точки объектного координатного пространства, относительно которой выполняется автоматическое разбиение всего множества отрезков на два или четыре независимых подмножества. Обращение к SPLITP имеет смысл только в тех случаях, когда значение параметра ITYP в обращении к программе COORDT равно 6, 7 или 8. Эта программа реализована как входная точка программы COORDT.

Программа HALLNE(DX, DY, DZ, NLINES, N1, N2, NCLUST, NLNDR, G, TOL, ITYP) позволяет построить изображение всего графического объекта или некоторой его части с удалением невидимых линий. Ее параметрами являются:

N1, N2 - номера первого и последнего отрезков, определяющих изображаемую часть графического объекта;

G - ширина разрыва;

TOL - допуск по глубине.

Остальные параметры те же, что и в программе COORDT.

Программа HALOED(DX, DY, DZ, NLINES, NDRAW1, NDRAW2, NC1, NC2, NCLUST, NLNDR, G,

TOL,ITYP) позволяет построить изображение заданной части графического объекта после проверок на видимость с указанным диапазоном отрезков. Параметры программы:

DX, DY, DZ - массивы значений проекций координат крайних точек отрезков (размером (2,NLINES));

NLINES - количество отрезков в описании графического объекта;

NDRAW1, NDRAW2 - номера первого и последнего в совокупности отрезков, которые будут изображаться;

NC1, NC2 - номера первого и последнего в совокупности отрезков, которые будут участвовать в проверках на видимость.

Значения остальных параметров те же, что и в программах COORDT и HALLNE.

Замечание. Значения параметров NC1, NC2 могут отличаться от значений параметров NDRAW1, NDRAW2 лишь при ITYP = 1 или ITYP = 2. При других значениях ITYP параметры NC1, NC2 считаются совпадающими с NDRAW1, NDRAW2.

Программа DRWHL(DX,DY,DZ,NLINES,NL1,NL2) позволяет построить изображение всего графического объекта или некоторой его части без удаления невидимых линий. Значения параметров программы совпадают со значениями соответствующих параметров в программе HALLNE.

Вспомогательные программы. При работе с графическими объектами, представленными в виде последовательности отрезков прямых, может потребоваться повернуть объект (или его часть) на некоторый угол или перенести на некоторое расстояние. Эти задачи решаются с помощью обращений к программам ROTH и TRANH.

Кроме того, в некоторых случаях возникает необходимость представить сеточную функцию в виде последовательности отрезков прямых. Для этой цели служит программа ARRINL. Отметим, что при юго-западном положении центра проекции относительно области определения функции (рис.8.10) формируется упорядоченная последовательность отрезков прямых, которую можно изобразить вторым способом, т. е. при значении параметра ITYP = 2 (см. описание программ COORDT, HALOED, HALLNE). Порядок заполнения массивов DX, DY, DZ показан на рис. 8.10.

Программа ROTH(NAXES,FI,DX,DY,DZ,NLINES,NBEG,NEND) позволяет повернуть графический объект или некоторую его часть вокруг одной из главных координатных осей правой системы координат на заданный угол в направлении по часовой стрелке, если смотреть с конца единичного вектора этой оси на начало координат. Параметрами этой программы являются:

NAXES - номер оси, относительно которой выполняется поворот:

NAXES = 1 - ось X, NAXES = 2 - ось Y, NAXES = 3 - ось Z;

FI - угол поворота (в градусах);

DX, DY, DZ - массивы значений координат крайних точек отрезков (размером (2, NLINES));

NLINES - количество отрезков в описании графического объекта;

NBEG, NEND - номера первого и последнего в совокупности отрезков, которые подвергаются повороту.

Программа TRANH(DELX,DELY,DELZ,DX,DY,DZ,NLINES,NBEG,NEND) предназначена для переноса графического объекта или некоторой его части на заданные расстояния вдоль координатных осей. Ее параметры:

DELX, DELY, DELZ - координаты вектора переноса относительно осей X, Y и Z.

Остальные параметры те же, что и в программе ROTH.

Программа ARRINL(X,Y,A,IX,IY,IXB,IXE,IYB,IYE,DX,DY,DZ,NLINES,NLBEG) предназначена для преобразования сеточной однозначной функции в представление этой функции в виде последовательности отрезков прямых. Параметры программы:

X - массив точек сетки по координате X, расположенных в порядке возрастания;

Y - массив точек сетки по координате Y, расположенных в порядке возрастания;

A - двумерный массив значений функции в узлах сетки, определяемой массивами X и Y (размером (IX,IY));

IX, IY - размерность сетки по осям X и Y;

IXB, IXE, IYB, IYE - начальные и конечные номера столбцов и строк, определяющих на области задания функции прямоугольную подобласть, предназначенную для преобразования в последовательность отрезков;

DX, DY, DZ - массивы значений координат крайних точек отрезков (размером (2,NLINES));

NLINES - число отрезков, образующих сетку ($NLINES = 2 * IX * IY - (IX + IY)$), где $IDX = IXE - IXB + 1$, $IDY = IYE - IYB + 1$);

NLBEG - номер, начиная с которого в массивах DX, DY, DZ размещаются отрезки, составляющие рассматриваемую поверхность.

Замечание. Пользуясь параметром NLBEG, можно формировать сцены, состоящие из нескольких поверхностей, наложенных друг на друга (см. [1, л]).

Служебные программы.

Программа HLDLN(DX,DY,DZ,NLINES,I,NC1,NC2,NCLUST,NLNDR,ITYP) позволяет построить изображение отрезка с номером I после проверок на видимость с отрезками из совокупности (NC1, NC2), если параметр ITPY = 1 или ITPY = 2, или с отрезками, принадлежащими подмножествам с номерами NC1, NC2, если ITPY = 3 ÷ 8.

Программа CROSSP(X,Y,RC,T) предназначена для вычисления параметра T в диапазоне от 0 до 1, который соответствует точке с координатами RC(1), RC(2), лежащей на отрезке с крайними точками X(1), Y(1) и X(2), Y(2).

Программа EXTLN(X,Y,Z,G) позволяет продолжить отрезок с крайними точками X(1), Y(1), Z(1) и X(2), Y(2), Z(2) в обе стороны так, чтобы длина его проекции на картинную плоскость увеличилась в каждую сторону на расстояние равное значению параметра G.

Программа STORIN(AL,AU) служит для запоминания невидимой части проекции I-го отрезка, определяемой параметрами AL и AU, в массиве-буфере TE(15,2). При этом два или более пересекающихся невидимых интервала объединяются в один, отбрасываются интервалы, которые уже были учтены при предыдущих обращениях к этой программе, и т. д.

Программа SVLSEG(AL,AU,J) служит для занесения значений параметров AL и AU, определяющих невидимую часть отрезка, в J-й элемент буфера TE.

Программы SHDOWN(J) и программа SHUP(J) предназначены для перемещения на одну позицию соответственно вниз или вверх содержимого буфера TE, заключенного в диапазоне, определяемом параметрами J и NL. Значение NL выбирается из 31-го слова общего блока GFSV и представляет собой текущее число отдельных интервалов невидимости на проекции рассматриваемого отрезка.

Программа TESTVS предназначена для идентификации ситуаций, когда рассматриваемый отрезок оказывается полностью закрыт другими отрезками. В этом случае сбрасывается флаг INVSLN, который является 32-м словом общего блока GFSV.

Программа DRAWTE служит для изображения видимых частей рассматриваемого отрезка.

Программа DRAWEL(T,I) позволяет по заданному в интервале от 0 до 1 значению параметра прямой T вычислить координаты соответствующей точки. Если параметр I = 1, то проводится отрезок из точки текущего положения пера в вычисленную точку, если же I = 0, то рисование не производится.

Программа HCINIT(A) служит для формирования единичной матрицы размером 4×4.

Программа MXMULT(A,B) позволяет выполнить умножение двух матриц A и B размером 4×4. Результат умножения помещается в A.

Программа STROT3(NAXES,CS,SN,R) предназначена для формирования матрицы поворота R(4,4). Поворот выполняется вокруг одной из главных координатных осей, номер которой определяется параметром NAXES правой системы координат. Косинус и синус угла поворота задаются параметрами CS и SN.

Программа HCNCOR(X,Y,Z,T). С помощью этой программы по заданным координатам точки в объектной системе координат определяются координаты этой точки после воздействия на нее преобразования, задаваемого матрицей T размером 4×4.

Программа SORTY(DX,DY,DZ,NL,N1,N2) служит для упорядочения крайних точек отрезков, заданных массивами DX(K,NL), DY(K,NL), DZ(K,NL) таким образом, что K = 1 для точки с большим значением координаты DY, а для другой крайней точки K = 2. Упорядочению подвергаются отрезки с номерами от N1 до N2.

8.4.5. Примеры. В этом разделе приводятся примеры, иллюстрирующие использование описанных выше программ проецирования с удалением невидимых линий.

1. На рис. 8.26 показана поверхность, построенная с использованием программы HALLNE. Данные для поверхности (центр которой находится в начале координат) формируются при обращении к программе GIPER1.

```
DIMENSION DX(2,930),DY(2,930),DZ(2,930),NCLUST(930)
DATA N,G,TOL/930,0.15,0.4/
CALL GIPER1(2.,-2.,2.,15,30,DX,DY,DZ,N,1)
```

```

CALL PAGE(14.,26.,0,0,0)
CALL REGION(2.,8.,10.,10.,0,0,0)
CALL PRJHL(1,5.,5.,-10.)
CALL VPLIM(DX,DY,DZ,N,1,N,S)
CALL HALLNE(DX,DY,DZ,N,1,N,NCLUST,N,G,TOL,5)
CALL ENDPG('8.26')
END

```

Тот же результат получится при использовании фрагмента:

```

CALL PRJHL(1,5.,5.,-10.)
CALL VPLIM(DX,DY,DZ,N,1,N,S)
CALL SPLITP(0.,0.,0.)
CALL HALLNE(DX,DY,DZ,N,1,N,NCLUST,N,G,TOL,8)

```

Во втором случае явно указана точка, относительно которой должно проводиться разбиение графического объекта на независимые подмножества (см. пп. 8.4.3 и 8.4.4).

2. На рис. 8.32 показана сцена, состоящая из двух изолированных графических объектов: цилиндра и сферы. Цилиндр формируется с помощью обращения к подпрограмме CILIND, сфера – в результате вызова двух подпрограмм - SPHERG и SPHERV. Благодаря тому, что сцена допускает описание в виде независимых подмножеств отрезков, которые можно разделить плоскостью, время, затрачиваемое на вычисления, удается заметно сократить (в данном случае на 60%).

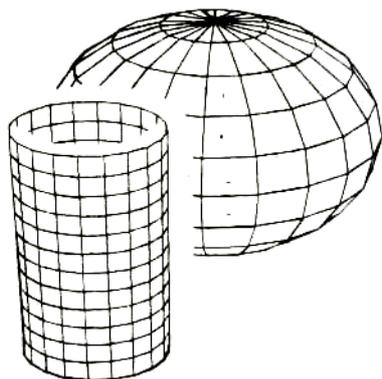


Рис. 8.32. Изображение сцены, допускающей представление в виде двух независимых подмножеств отрезков

```

DIMENSION DX(2,660),DY(2,660),DZ(2,660),NCL(280)
DATA N,NC/660,280/
С...ЦИЛИНДР (ОТРЕЗКИ 1 - 280):
CALL CILIND(0.4,-1.,0.9,12,20,DX,DY,DZ,N,1)
CALL TRANH(1.7,0.,0.,DX,DY,DZ,N,1,NC)
С...СФЕРА (ОТРЕЗКИ 281 - 660)
CALL SPHERG(1.,20,10,DX,DY,DZ,N,NC+1)
CALL SPHERV(1.,20,10,DX,DY,DZ,N,NC+1+180)
С...ИЗОБРАЖЕНИЕ СЦЕНЫ:
CALL PAGE(14.,26.,0,0,0)
CALL REGION(2.,8.,10.,10.,0,0,0)
CALL PRJHL(1,10.,5.,-3.)
CALL VPLIM(DX,DY,DZ,N,1,N,S)
С...ИЗОБРАЖЕНИЕ ЦИЛИНДРА:
CALL COORDT(DX,DY,DZ,N,1,NC,NCL,NC,3)
CALL HALOED(DX,DY,DZ,N,1,NC,1,NC,NCL,NC,0.1,0.3,3)
С...ИЗОБРАЖЕНИЕ СФЕРЫ:
CALL COORDT(DX,DY,DZ,N,NC+1,N,NCL,1,1)

```

```
CALL HALOED(DX,DY,DZ,N,NC+1,N,1,N,NCL,1,0.15,0.5,1)
CALL ENDPG('8.32')
END
```

3. На рис. 8.33 изображена центральная проекция пространственной кривой, построенная с использованием метода ореола.

```
DIMENSION DX(2,880),DY(2,880),DZ(2,880),NCLUST(880)
DATA NN,N1,M,N,K/8,110,3,4,1/,G,TOL,ITYP/0.05,0.2,5/
N2=NN*N1
H1=3.5/NN
H2=4.5/NN
FI=H1
PSI=H2
DO 21 J=1,N2
FI=FI+H1/N1
PSI=PSI+H2/N1
TH=6.283*(J-1)/(N1-1)
X=SIN(TH)
Y=1.2*SIN(M*TH+FI)
Z=1.5*SIN(N*TH+PSI)
IF(K.EQ.1) GOTO 20
DX(2,K-1)=X
DY(2,K-1)=Y
DZ(2,K-1)=Z
IF(K.EQ.N2) GOTO 21
20 DX(1,K)=X
DY(1,K)=Y
DZ(1,K)=Z
K=K+1
21 CONTINUE
```

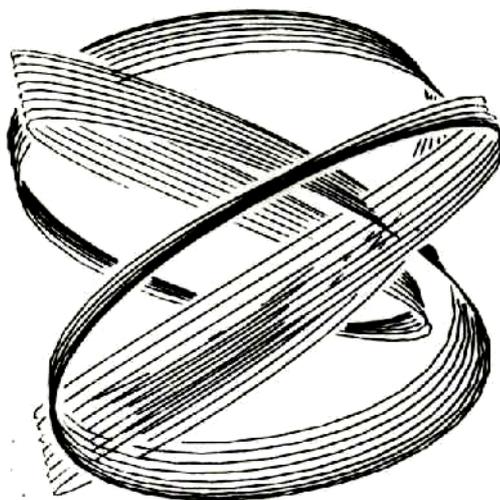


Рис. 8.33. Изображение пространственной кривой, построенное с использованием метода ореола

С...РИСОВАНИЕ ПРОЕКЦИИ ПРОСТРАНСТВЕННОЙ КРИВОЙ:

```
CALL PAGE(15.,15.,0,0,1)
CALL REGION(1.,1.,13.,13.,0,0,0)
CALL PRJHL(1,15.,8.,3.)
CALL VPLIM(DX,DY,DZ,N2,1,N2,S)
CALL HALLNE(DX,DY,DZ,N2,1,N2,NCLUST,N2,G,TOL,ITYP)
CALL ENDPG('8.33')
END
```