

The word "SPACE" is written in a bold, three-dimensional, geometric font. It is oriented diagonally, sloping upwards from left to right. The letters are primarily white with thick black outlines, set against a light beige background. The background features several concentric, curved red bands that wrap around the text. A single, thick red diagonal line cuts across the composition, starting from the bottom left and ending at the top right, intersecting the text.

SPACE

SPACE

С С С Р

МИНИСТЕРСТВО ПРИБОРОСТРОЕНИЯ,
СРЕДСТВ АВТОМАТИЗАЦИИ И СИСТЕМ УПРАВЛЕНИЯ
СОЮЗСИСТЕМПРОМ

Научно-производственное объединение
"ЦЕНТРПРОГРАМСИСТЕМ"

Версия О201

ПАКЕТ ПРИКЛАДНЫХ ПРОГРАММ
ДЛЯ РЕШЕНИЯ ТРЕХМЕРНЫХ
ГРАФИЧЕСКИХ ЗАДАЧ
(ППП SPACE)

Калинин 1985

Авторы разработки пакета прикладных программ для решения трехмерных графических задач (**ППП SPACE**): Машокин А. М., Упольников С. А.

ППП SPACE сопровождается отделом ЗО6 НПО "Центрпрограммсистем"
(зав. отделом Никонов А. С.).

1. ОПИСАНИЕ ПРИМЕНЕНИЯ

1.1. Назначение ППП

Пакет прикладных программ для решения трехмерных графических задач (ППП **SPACE**) предназначен для обработки трехмерной геометрической информации на ЕС ЭВМ.

ППП **SPACE** реализует следующие функции:

- описание пространственных кривых, участков поверхностей и границ тел как набора участков поверхностей;
- преобразование сдвига, поворота и масштабирования объектов, теоретико-множественные операции над объектами;
- построение объектов с помощью смещения и вращения объектов меньшей разности;
- построение сечений и визуализация проекций с удалением невидимых линий;
- вычисление длины, площади, объема, центра масс, момента инерции.

Пакет **SPACE** предназначен для решения прикладных задач, связанных с расчетом геометрии трехмерных объектов, а также для автоматизации проектных и конструкторских работ на этапе технической подготовки производства. Пакет **SPACE** целесообразно использовать в научно-исследовательских и проектно-конструкторских организациях при решении задач, требующих представления, обработки и визуализации трехмерной геометрической информации с использованием ЭВМ, если имеются необходимые технические средства.

Ограничений на область применения нет.

1.2. Условия применения ППП

ППП **SPACE** структурно представлен в виде набора подпрограмм, управление которыми осуществляется с помощью обычных средств операционной системы, то есть является пакетом библиотечного типа. При работе пакета не используются никакие стандартные системные средства.

Для описания геометрических объектов пользователь определяет количество необходимых подпрограмм и порядок их работы на основе информации, представленной

в разд. 2. Здесь дается подробное описание всех подпрограмм пакета, включающее в себя наименование подпрограммы, способ использования, список формальных параметров, описание принципов работы.

Обращение к подпрограммам пакета производится по правилам ФОРТРАН.

Для функционирования пакета пользователь определяет область оперативной памяти в виде трех, специальным образом поименованных, общих (**COMMON**) блоков, определенного объема. Создается структурированный информационный массив (архив). Информация, содержащаяся в нем, необходима для работы подпрограмм пакета. Пользователь задает объем информационного массива и первичную структуру (объемы общих блоков). Дальнейшая редакция содержимого этого массива осуществляется подпрограммами пакета.

Подробное описание структуры информационного массива, способов доступа к нему и методов редакции представлено в разд. 2.

Величина объема оперативной памяти для информационного массива определяется пользователем в зависимости от сложности моделируемого геометрического объекта. Случай недостаточности выделенного объема оперативной памяти или неудачной его структуры идентифицируются по входной информации соответствующих подпрограмм, что позволяет пользователю скорректировать свою программу.

Кроме оперативной памяти, выделенной под архив, при выполнении программ ППП **SPACE** занимает 180К байтов оперативной памяти.

Для размещения программных компонент пакета на магнитном диске требуется 1,4 Мбайта памяти.

ППП **SPACE** используется автономно в пакетном режиме под управлением ОС ЕС версии 4.1 и выше.

Для работы пакета необходимы следующие технические средства:

- процессор ЕС ЭВМ;
- устройство ввода с перфокарт ЕС-6012;
- алфавитно-цифровое печатающее устройство (АЦПУ) ЕС-7030;
- накопитель на магнитной ленте ЕС-5010;
- накопитель на магнитном диске ЕС-5061;
- печатно-пишущая машинка или дисплей в качестве операторского терминала.

Вывод графических результатов работы пакета осуществляется через систему математического обеспечения графопостроителей. Для вывода необходимо дополнительно наличие следующих технических средств:

- устройство вывода на перфоленту ЕС-7022;
- графопостроители ЕС-7051, ЕС-7052, ЕС-7053, ЕС-7054, BENSON-220, ИТЕКАН в любом наборе.

Входными данными пакета являются описания обрабатываемых геометрических объектов в виде обращений к подпрограммам ППП **SPACE** на языке ФОРТРАН.

Выходными данными пакета являются данные, выводимые на АЦПУ операционной системой ОС ЕС, и графические данные, выводимые с использованием системы математического обеспечения графопостроителей (СМОГ) [1].

1.3. Состав и функции ППП

Перечень функций ППП **SPACE** приведен в п. 1.1.

Структура и взаимосвязи компонентов пакета описаны в разд. 3. Перечень подпрограмм пакета приведен в прил. 1 - прил. 10.

ППП **SPACE** не имеет сервисных средств.

На этапе вывода информации на графопостроители ППП **SPACE** использует СМОГ.

Пользователь в головной программе должен описать вызов подпрограмм СМОГ, необходимых для вывода информации на устройства графического отображения.

1. 4. Описание задачи

ППП **SPACE** разработан как универсальное программное средство для решения трехмерных геометрических задач и использует только средства языка ФОРТРАН. Он состоит из трех независимых комплексов **MC**, **SP**, **YP**.

Комплект подпрограмм **MC** является составной частью ППП **SPACE**, но может использоваться и автономно. Комплект предоставляет средства описания пространственных кривых, участков поверхностей и границ тел как набора участков, поверхностей. В предлагаемом варианте комплекса подпрограмм **MC** кривые суть ломаные или последовательный набор кубических кривых Фергюсона. Под участком поверхности понимается параметрически заданная поверхность, граница которой совпадает с указанными (ранее описанными) четырьмя кривыми, а ее параметры меняются в пределах единичного квадрата. Фактически участок (порция) поверхности задается своими краями и способом интерполяции (полилинейным или полукубическим). Участку поверхности автоматически присваивается направление внешней нормали, что позволяет использовать его как часть границы тела.

Подпрограммы масштабирования, сдвига и поворота указанных геометрических объектов упрощают в некоторых случаях описание нужного объекта. Для хранения геометрических объектов в пакете используется достаточно простая база данных, которая полностью расположена в оперативной памяти ЭВМ. Последнее условие ограничивает применение комплекса **MC**, но в следующих версиях это ограничение будет снято. Подпрограммы построения многогранных представлений пространственных тел позволяют использовать изобразительные и расчетные средства пакета **SPACE**.

Комплект подпрограмм **SP** является составной частью ППП **SPACE**, но может использоваться и самостоятельно. Комплект является, в основном, недоступным непосредственно пользователю. Комплект, в свою очередь, можно разделить на следующие части: подпрограммы, реализующие функции векторной алгебры; подпрограммы, аппроксимирующие плоские кривые 2-го порядка; подпрограммы обработки многоугольников и многогранников в пространстве.

Подпрограммы пакета реализуют: преобразования сдвига, поворота и масштабирования объектов, теоретико-множественные операции над объектами; операции построения объектов с помощью смещения и вращения объектов меньшей размерности; построение сечений и визуализацию проекций с удалением невидимых линий и некоторые другие специальные преобразования массивов точек.

Подпрограммы комплекса, не вошедшие ни в одну из упомянутых частей, отнесены к служебным подпрограммам, которые могут быть использованы так же, как и остальные программы комплекса.

Комплект **SP** носит базовый характер и является основой, на которой строится входной язык пакета **SPACE**.

Комплект подпрограмм **YP** является составной частью ППП **SPACE**. Комплект предоставляет средства описания плоских кривых, многоугольников и многогранников. В предлагаемом варианте комплекса подпрограмм **YP** плоская кривая строится с использованием отрезков, дуг окружностей, эллипсов, парабол и гипербол в качестве

ее элементов. Аппроксимация замкнутой кривой набором отрезков позволяет строить многоугольники, над которыми определены (в комплекте) теоретико-множественные операции и аффинные преобразования. Поскольку ППП **SPACE** предназначен для описания трехмерных геометрических объектов, то операция построения многоугольника необходима прежде всего для описания чертежей многогранников.

Над многогранниками также определены теоретико-множественные операции и аффинные преобразования. Эти средства позволяют строить модели деталей практически любой геометрической сложности. Кроме средств построения, комплект **SP** предоставляет средства изображения и вычисления площадей, объемов, центров масс и моментов инерции многоугольников и многогранников.

Фактически все задачи комплекта **YP** решаются подпрограммами комплекта **SP**. Основное назначение комплекта **YP** заключается в создании обрамления (или входного языка) комплекта **SP**, упрощающего задачу использования средств комплекта **SP** в прикладных программах.

1.5. Входные и выходные данные

Описания обрабатываемых геометрических объектов производятся ППП **SPACE** в виде обращений к подпрограммам на языке ФОРТРАН. При этом могут использоваться все соответствующие устройства подготовки данных. Данные, используемые для описания объектов, могут быть следующих типов:

- вещественные массивы, содержащие координаты точек на плоскости или в пространстве;
- целые переменные, идентифицирующие архивные объекты;
- целые и вещественные константы и переменные, являющиеся параметрами описываемых объектов (размерности массивов, пределы изменения углов и т. п.).

Формат и использование входных данных полностью соответствуют стандартам языка ФОРТРАН. Входные данные каждой подпрограммы ППП **SPACE** подробно описаны в п. 2.2.

Выходные данные выводятся на АЦПУ с помощью штатных программных средств ОС либо на графические устройства с использованием системы СМОГ и необходимого для функционирования этой системы набора аппаратуры. Графические выходные данные выводятся с помощью подпрограммы **SP2015**, использующей СМОГ, и при необходимости могут быть обработаны любой другой графической системой при соответствующей модификации подпрограммы **SP2015** (см. п. 2.2).

1.6. Рекомендации по освоению ППП

Для освоения и эксплуатации ППП **SPACE** необходимо знать язык программирования ФОРТРАН.

При использовании режима вывода и визуализации графической информации на графопостроителе необходимо знание СМОГ.

2. РУКОВОДСТВО ПРОГРАММИСТА

2.1. Назначение и условия применения ППП

ППП **SPACE** предназначен для решения прикладных задач, связанных с расчетом геометрии трехмерных объектов, а также для автоматизации проектных и конструкторских работ на этапе технической подготовки производства.

Более подробно назначение, условия применения и функции пакета описаны в разд. 1.

2.2. Характеристика ППП

2.2.1. Архив объектов

2.2.1.1. Общие сведения

ППП **SPACE** разработан как универсальное программное средство для решения трехмерных геометрических задач и использует средства языка ФОРТРАН. Он состоит из трех комплектов **MC**, **SP**, **УР**. Каждый комплект состоит из подпрограмм, каждая из которых выполняет определенную функцию. Обращение к подпрограммам описано далее.

Требования к составу технических средств и используемым ресурсам приведены в разд. 1.

Перечень функциональных элементов приведен в прил. 1 – прил. 10.

Комплект подпрограмм **MC** разработан для описания кривых, участков поверхностей, опирающихся на заданные кривые, и границ пространственного тела как набора порций поверхностей. Несложный анализ позволяет сделать вывод, что описание объекта должно включать в себя идентификацию самого объекта, перечень объектов, в состав которых входит рассматриваемый объект, перечень объектов, входящих в состав рассматриваемого, и числовую информацию, например, перечень точек ломаной.

В соответствии с этим выводом в комплекте **MC** приняты следующие определения.

В описание объекта входят:

- блок идентификации объекта;
- массив ссылок вверх (ap);
- массив ссылок вниз (down);
- массив информационной (числовой) части.

Под описание объектов выделены три общих (COMMON) блока: KIEL, KCEL и AREL. Первые два - целые, последний - вещественный. Блоки идентификации объектов хранятся в COMMON/KIEL/, массивы ссылок - в COMMON/KCEL/, а массивы информационных частей - в COMMON/AREL/. Первые 10 (десять) слов COMMON /KIEL/ являются служебными, остальная часть COMMON /KIEL/ разбита на участки одинаковой длины - блоки идентификации объектов. Минимальная длина блока равна 5 (пяти):

- 1-е слово - адрес массива информационной части в COMMON /AREL/;
- 2-е слово - длина массива информационной части,
- 3-е слово - адрес массива ссылок вверх в COMMON /KCEL/;
- 4-е слово - адрес массива ссылок вниз в COMMON /KCEL/;
- 5-е слово - тип объекта.

При организации архива длина блока идентификации задается и может быть больше 5. Дополнительные слова могут быть использованы для хранения текстового имени объекта. Количество дополнительных слов хранится в KIEL (4), возможное количество блоков в KIEL (9), адрес первого блока в KIEL (6), длина блока в KIEL (5).

Блок считается свободным, если его первое слово нулевое.

Массив ссылок у объекта отсутствует, если его адрес в блоке идентификации объекта нулевой.

Структура массива ссылок:

- 1-е слово - длина массива,
 - 2-е слово - количество ссылок (KC),
 - 3-е слово - первая ссылка - адрес начала блока идентификации объекта, на который сформирована ссылка,
- ...
- (KC+2) слово - последняя ссылка.

Каждая ссылка в массиве встречается только один раз. Каждой ссылке вверх должна соответствовать ссылка вниз. В комплекте МС эти требования выполняются автоматически.

Объект считается не заданным, если его тип нулевой или отрицательный. Незаданный объект обязательно имеет ссылки вверх, но не имеет ссылок вниз.

Подпрограммы комплекта МС различают объекты по числовым именам - адресам соответствующих блоков идентификации объектов в COMMON /KIEL/. Это условие распространяется и на пользователя комплекта подпрограмм МС.

Имя объекта - идентификатор целой переменной, значение которой либо равно нулю (объект не задан), либо равно адресу блока в COMMON /KIEL/. Значение переменной определяется автоматически подпрограммами комплекта.

Для диагностики работы подпрограмм комплекта МС используется 10-е слово **COMMON /KIEL/**. Если по выходу из подпрограммы **KIEL(10) ≠ Ø**, то подпрограмма свою задачу не выполнила.

2.2.1.2. Инициация архива

В программе, использующей подпрограммы комплекта МС, должны быть описаны общие блоки **KIEL, KCEL, AREL**:

COMMON /KIEL/
COMMON /KCEL/
COMMON /AREL/

Задание их длин зависит от количества описываемых объектов, длин массивов информационных частей и наличия массивов ссылок.

Инициация (подготовка) архива к работе осуществляется обращением к подпрограмме **MCØ**:

CALL MCØ(N,K,KI,KC)

где **N** – целое число, длина **COMMON /AREL/**;

K – целое число, длина **COMMON /KIEL/**;

KI – целое число, количество дополнительных слов блока идентификации объектов;

KC – целое число, длина **COMMON /KCEL/**.

Результат работы подпрограммы **MCØ**:

KIEL(1)=N – длина **COMMON /AREL/**;

KIEL(2)=1 – начало свободного участка **COMMON /AREL/**;

KIEL(3)=K – длина **COMMON /KIEL/**;

KIEL(4)=KI – количество дополнительных слов блока;

KIEL(5)=5+KI – длина блока;

KIEL(6)=11 – начало блоков в **COMMON /KIEL/**;

KIEL(7)=KC – длина **COMMON /KCEL/**;

KIEL(8)=1 – начало свободного участка **COMMON /KCEL/**;

KIEL(9) – количество блоков **COMMON /KIEL/**;

KIEL(10)=Ø – сигнальное слово.

Все блоки **COMMON /KIEL/** расписываются нулем. Второе и восьмое слова **COMMON /KIEL/** в процессе работы подпрограмм изменяются и характеризуют участок с **KIEL(2)** по **KIEL(1)**, слово **COMMON /AREL/** – участок с **KIEL(8)** по **KIEL(7)**, слово **COMMON /KCEL/** – участки, в которых нет записей.

2.2.1.3. Выделение памяти под объекты

Если пользователю известны априори длины частей описания объектов, то он может выделить память под объект, используя подпрограммы **MC1, MC2** или **MC11**. Использование этих подпрограмм можно сравнить с предварительным распределением памяти, что во многих случаях повышает эффективность дальнейшей работы подпрограмм комплекта.

Обращение:

CALL MC1(KI, N, NU, ND)

где KI - целое число, имя объекта;
N - длина массива информационной части объекта KI;
NU - количество планируемых ссылок вверх;
ND - количество планируемых ссылок вниз.

Результат:

KI = 0 - память не выделилась:

KIEL(10)=1- исчерпан список блоков в COMMON /KIEL/;

KIEL(10)=2- нет места в COMMON /AREL/;

KIEL(10)=3- нет места в COMMON /KCEL/;

KI≠0 KIEL(10)=0- память выделена, тип объекта KI нулевой, т.е.
объект считается незаданным.

Обращение:

CALL MC2(KI, N, NU, ND, KIT)

где KI - целое число, имя объекта;
N - длина массива информационной части;
NU - количество планируемых ссылок вверх;
ND - количество планируемых ссылок вниз;
KIT - тип объекта.

Результат:

KI = 0 - память не выделена:

KIEL(10)=1- исчерпан список блоков в COMMON /KIEL/;

KIEL(10)=2- нет места в COMMON /AREL/;

KIEL(10)=3- нет места в COMMON /KCEL/;

KI≠0 KIEL(10)=0- память выделена, тип объекта задан и равен KIT, первое
слово массива информационной части равно нулю.

Подпрограмма MC11 служит для дополнительного выделения памяти ранее сформированному объекту. Эту подпрограмму используют модули комплекта МС при необходимости увеличения длин частей описания объекта, например, при формировании ранее не запланированных ссылок.

Обращение:

CALL MC11(KI, N, NU, ND)

где KI - числовое имя объекта;
N - количество слов, на которое необходимо увеличить массив информационной части;
NU - количество добавочно планируемых ссылок вверх;
ND - количество добавочно планируемых ссылок вниз.

Результат:

- KIEL(1Ø)=Ø** - дополнительная память выделена,
KIEL(1Ø)≠Ø - дополнительная память не выделена.

2.2.1.4. Формирование ссылок, типа и удаление объектов

Операции расформирования или формирования ссылочных связей двух объектов А и В реализуются подпрограммами MC3 и MC4.

Обращение:

CALL MC3(KI, KJ)

где **KI** - числовое имя объекта А;

KJ - числовое имя объекта В.

Результат:

У объекта А нет ссылки вниз на объект В, у объекта В нет ссылки вверх на объект А. Если объект В был не задан, а ссылка вверх на объект А последняя, то объект В из архива исключен и **KJ** присваивается нулевое значение.

Обращение:

CALL MC4(KI, KIT)

где **KI** - числовое имя объекта А;

KJ - числовое имя объекта В.

Результат:

- KIEL(IØ)=Ø** - объект А имеет ссылку вниз на объект В, объект В имеет ссылку вверх на объект А;
KIEL(IØ)≠Ø - подпрограмма не выполнена.

Изменение типа объекта заключается в изменении значения пятого слова блока идентификации объекта. Эта простая операция реализуется подпрограммой MC5.

Обращение:

CALL MC5(KI, KIT)

где **KI** - числовое имя объекта;

KIT - целое число, значение типа объекта.

Результат:

Объект имеет тип **KIT**.

Операция удаления объекта из архива в комплекте подпрограмм МС реализуется двумя принципиально разными способами. Первый, наиболее простой способ заключается в удалении объекта и соответствующего изменения ссылок на него. Второй способ заключается в удалении не только самого объекта, а и всех его составляющих объектов, если они не входят в другие структуры. Удаляемый объект всегда становится неопределенным, если у него имеются ссылки вверх. Удаление объекта осуществляется подпрограммой MC3Д. Рис. 1 иллюстрирует работу этой подпрограммы по удалению объекта K1. Обведенный на рисунке кружочком идентификатор обозначает незаданный объект, стрелки - связи между объектами. Левая часть рисунка - исходная структура, правая часть - структура объектов, полученных в результате работы.

Удаление куска объекта, т.е. его самого и всех его составляющих, осуществляется подпрограммой МСЗ1 (рис. 2).

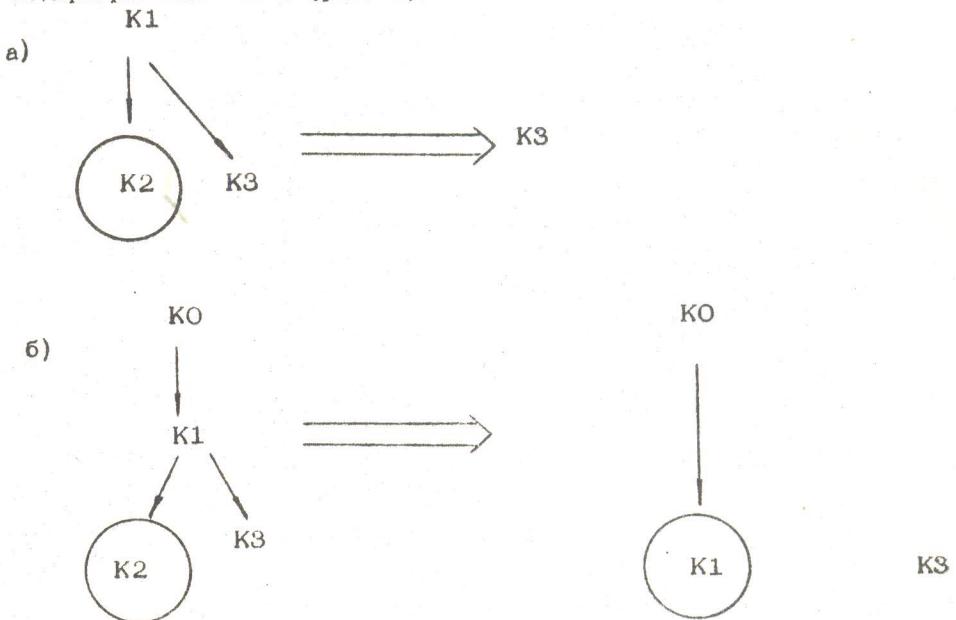


Рис. 1. Удаление объекта

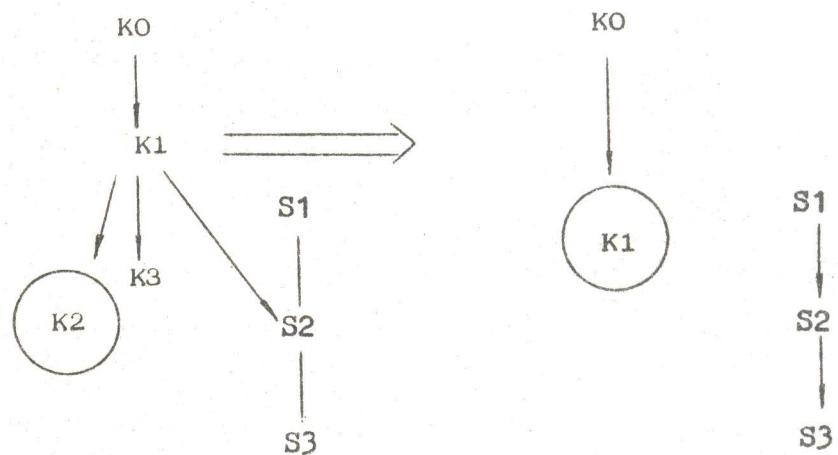


Рис. 2. Удаление куска объекта

Обращение:

CALL MC30(KI)

где KI – числовое имя объекта.

Результат:

- KIEL(10)=Ø, KI=Ø** – объект удален;
- KIEL(10)=1, KI≠Ø** – объект был не задан, не удален;
- KIEL(10)=2, KI≠Ø** – объект стал незаданным, т.е. имеет ссылки вверх, ссылок вниз уже нет.

Обращение:

CALL MC31(KI)

где KI – числовое имя объекта, определяющего куст удаления.

Результат:

- KIEL(10)=Ø, KI=Ø** – куст удален;
- KIEL(10)=1, KI≠Ø** – объект был и остался не задан;
- KIEL(10)=2, KI≠Ø** – куст удален, объект стал незаданным, у него есть ссылки вверх.

2.2.1.5. Служебные подпрограммы

К служебным подпрограммам поддержки архива относятся обмен информацией между объектами А и В (подпрограмма MC12) и сжатие COMMON /AREL/ и /KCEL/ (подпрограммы MC20 и MC21). Последние подпрограммы рекомендуется использовать в крайних случаях, так как время их выполнения может оказаться большим. В комплекте подпрограмм МС они используются для перераспределения памяти, когда длины свободных участков в COMMON /AREL/ или /KCEL/ недостаточны для продолжения работы программы.

Обращение:

CALL MC12(KI, KJ)

где KI – числовое имя объекта А;

KJ – числовое имя объекта В.

Результат:

Массив информационной части, массивы ссылок, тип и текстовое имя объекта А дублируются в соответствующих частях объекта В.

Если размеры массивов описания объекта KJ меньше размеров массивов описания объекта KI, то архив портится.

Обращение:

CALL MC20(N)

где N – длина свободного участка COMMON /AREL/, мусора в массиве нет, адреса массивов информационных частей объектов изменились, целое число.

Обращение:

CALL MC21(N)

где N - длина свободного участка COMMON /KCEL/, мусора в массиве нет, адреса массивов ссылок объектов изменились, целое число.

Естественно, что набор подпрограммы поддержки архива может быть существенно расширен. Однако для комплекта подпрограмм МС такого набора средств достаточно. Дальнейшее развитие комплекта приведет к улучшению средств архивации.

2.2.2. Описание пространственных кривых и участков поверхностей

2.2.2.1. Общие сведения

Комплект подпрограмм МС предоставляет средства описания пространственных кривых и участков поверхностей, которые применимы к описанным объектам преобразования сдвига, масштабирования и поворота вокруг оси на заданный угол. Предлагается, что геометрические объекты задаются в декартовой системе координат.

Кривая для пользователя представляет собой вектор-функцию параметра $U \in (\emptyset, 1)$. Значению параметра $U = \emptyset$ соответствует начало кривой, а $U=1$ - конец кривой. Изменение параметра U в общем случае не является естественной параметризацией кривой по ее длине. В комплекте предусмотрено задание двух типов кривых: ломаной и кубического многоэлементника.

Участок поверхности определяется с помощью интерполяционной формулы по методу Кунса и опирается на четыре заданные кривые. В пакете используются интерполяционные формулы, основанные на линейной или кубической интерполяции.

Пространственное тело можно описать как набор его участков границ, т.е. фактически будет описана граница тела. Такого описания достаточно, например, для приближенного вычисления объема, момента инерции или для построения многогранника, аппроксимирующего исходное тело. В комплекте подпрограмм нет средств проверки корректности описываемых объектов, которые бы позволили проверить, пересекаются ли кривые и поверхности, описывает ли набор участков поверхностей пространственное тело.

2.2.2.2. Описание пространственных кривых

Пространственная ломаная как геометрический объект задается подпрограммой MC110. В этом случае объекту присваивается тип 10.

Обращение:

CALL MC110(KI,N,R)

где KI - числовое имя объекта, значение KI должно быть равно либо \emptyset , либо память под объект должна быть выделена ранее, т.е. KI - имя объекта с типом 10, в этом случае построение кривой продолжается;

N - целое число, количество точек ломаной, которую следует построить или дополнить к ранее построенным;

R - вещественный массив длины $3 * N, R(3 * N)$, в котором последовательно записаны координаты точек ломаной: $X_1, Y_1, Z_1, X_2, Y_2, Z_2, \dots, X_N, Y_N, Z_N$.

Результат:

KIEL(10)= \emptyset - кривая построена, массив информационной части имеет вид: $N, U_1, X_1, Y_1, Z_1, U_2, X_2, Y_2, Z_2, \dots, U_N, X_N, Y_N, Z_N$; его длина $(4 * N + 1)$; параметры U_i определяют длину ломаной от начальной точки до i -й точки;

KIEL(1Ø)≠Ø – операция не выполнена.

Задание кубического пространственного многозвенника определяется подпрограммой MC12Ø по возрастающей последовательности параметра $U=(U_1, U_2, \dots, U_N)$, когда каждому значению параметра U соответствует точка R на кривой и вектор производной RU по этому параметру в этой точке. Объекту присваивается тип 2Ø.

Обращение:

CALL MC12Ø(KI, N, R, RU, U)

где KI – числовое имя объекта, KI = Ø – объект определяется, KI ≠ Ø и тип объекта равен 2Ø – объект достраивается;

N – количество точек на кривой;

$R=R(3 * N)$ – вещественный массив точек кривой, $R=(X_1, Y_1, Z_1, \dots, X_N, Y_N, Z_N)$;

$RU=RU(3 * N)$ – вещественный массив производных по параметру в точках кривой, $RU=(X'_1, Y'_1, Z'_1, \dots, X'_N, Y'_N, Z'_N)$;

$U=U(N)$ – вещественный массив значений параметров точек на кривой, $U=(U_1, U_2, \dots, U_N)$.

Результат:

KIEL(1Ø)=Ø – кривая построена, массив информационной части имеет вид: $N, U_1, X_1, Y_1, Z_1, X'_1, Y'_1, Z'_1, \dots, U_N, X_N, Y_N, Z_N, X'_N, Y'_N, Z'_N$, т.е. его длина $(7 * N+1)$;

KIEL(1Ø)≠Ø – операция не выполнена.

При описании кривых параметры могут выбираться произвольными. В дальнейшем пользователь должен знать имя кривой и то, что она является параметрической.

Вычисление точки на кривой или производной осуществляется подпрограммами MC115 и MC116.

Обращение:

CALL MC115(KI, W, R)

где KI – числовое имя кривой;

W – вещественное число; параметр, для которого необходимо вычислить точку на кривой $R=(X(W), Y(W), Z(W))$;

$R=R(3)$ – вещественный массив, в который записывается результат работы подпрограммы: координаты точки на кривой, соответствующие параметру W.

Результат:

KIEL(1Ø)=Ø – точка вычислена;

KIEL(1Ø)≠Ø – точка не вычислена.

Обращение:

CALL MC116(KI, W, R)

где KI – числовое имя кривой;

W – вещественное число; параметр, для которого необходимо вычислить производную в точке на кривой;

$R=R(3)$ – вещественный массив, в который записывается результат: координаты радиус-вектора производной

$R=(X'(W), Y'(W), Z'(W))$.

Результат:

$KIEL(1\emptyset)=\emptyset$ – производная вычислена;

$KIEL(1\emptyset)\neq\emptyset$ – производная не вычислена.

2.2.2.3. Описание участков поверхностей

Участок (порция) поверхности определяется математически как вектор функция $R(U, V)$, где $U \in (\emptyset, 1)$ и $V \in (\emptyset, 1)$. Кривые на поверхности $R(\emptyset, V)$, $R(U, 1)$, $R(1, V)$ и $R(U, \emptyset)$ описывают границу порции при изменении параметров U и V от \emptyset до 1. Эти кривые (пусть их имена суть K_1 , K_2 , K_3 и K_4) должны быть заданы до определения участка.

Участку поверхности приписывается направление внешней нормали таким образом, что при обходе границы по кривым K_1 и K_2 для наблюдателя со стороны внешней нормали порция остается справа (обход по часовой стрелке показан на рис. 3). Интерполяция по кривым вовнутрь порции может осуществляться линейно (тип 30) или по методу Кунса (тип 40). Задание порции осуществляется подпрограммой MC3040.

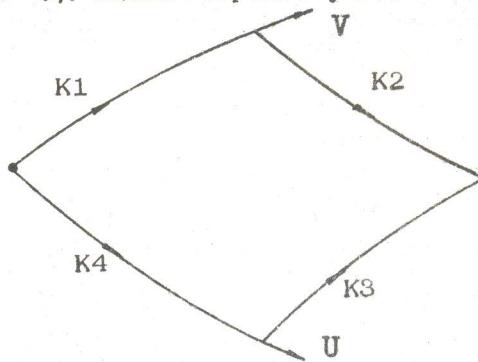


Рис. 3. Описание участка поверхности

Обращение:

CALL MC3040(K1, KIT, K1, NK1, K2, NK2, K3, NK3, K4, NK4)

где K_1 – числовое имя объекта, $K_1 = \emptyset$ – объект создается,

$K_1 \neq 0$ – под объект заранее выделена память;

K_{IT} – целое число (задается пользователем), тип порции 30 или 40;

K_1 – имя первой кривой, $R(\emptyset, V)$;

NK_1 – целое число, определяет направление кривой K_1 ;

$NK_1 > \emptyset$ означает, что направление кривой K_1 совпадает с направлением $R(\emptyset, V)$, $NK_1 < \emptyset$ означает, что направление кривой K_1 не совпадает с направлением $R(\emptyset, V)$;

K_2 – имя второй кривой, $R(U, 1)$;

NK_2 – целое число, определяет, изменять или нет направление кривой K_2 ;

K_3 – имя третьей кривой, $R(1, V)$;

NK_3 – целое число, определяет, изменять или нет направление кривой K_3 ;

K_4 – имя четвертой кривой, $R(U, \emptyset)$;

NK_4 – целое число, определяет, изменять или нет направление кривой K_4 .

Результат:

KIEL(1Ø)=Ø

- порция сформирована, массив информационной части содержит последние 8 параметров обращения к подпрограмме плюс 12 рабочих ячеек, т.е. его длина = 20, установлены ссылки между объектом K1 и объектами K1, ..., K4. (у K1, как минимум, 4 ссылки вниз);

KIEL(1Ø)≠Ø

- объект не сформирован.

Для того, чтобы на участке поверхности K1 можно было по параметрам (U, V) определять точку, необходимо произвести подготовительную операцию с помощью подпрограммы MC3Ø31. Необходимость такой операции вызвана прежде всего тем, что кривые, входящие в описание порции, могут быть переопределены.

Обращение:

CALL MC3Ø31(K1)

где K1 - числовое имя участка поверхности.

Результат:

KIEL(1Ø)=Ø

- операция выполнена, вычислены необходимые 12 вспомогательных величин, тип объекта изменился: Ø на 31, 4Ø на 41;

KIEL(1Ø)≠Ø

- операция не выполнена, не задана одна из кривых.

Вычисление точки R(U, V) на участке поверхности K1 осуществляется подпрограммой MC125.

Обращение:

CALL MC125(K1, U, V, R)

где K1 - имя участка поверхности;

U - вещественное число, значение первого параметра $U \in (\emptyset, 1)$;

V - вещественное число, значение второго параметра $V \in (\emptyset, 1)$;

R=R(3) - вещественный массив, в который записывается результат: R=(X(U, V), Y(U, V), Z(U, V)) - точка на участке поверхности.

Результат:

KIEL(1Ø)=Ø - точка вычислена;

KIEL(1Ø)≠Ø - точка не вычислена.

2.2.2.4. Описание тел с криволинейной границей

Пусть границу пространственного тела можно представить как совокупность участков поверхностей, внешние нормали которых совпадают с внешней нормалью относительно тела. В этом случае комплект подпрограмм МС предоставляет средства описания таких тел. При описании тела может случиться, что направление нормали порции поверхности следует изменить на противоположное. Эта операция реализуется подпрограммами MC135, MC2, MC150.

Обращение:

CALL MC135(K1)

где K1 - числовое имя участка поверхности.

Результат:

KIEL(1Ø)=Ø - нормаль изменилась и было обращение к MC3Ø31 для подготовки KI к вычислениям;

KIEL(1Ø)≠Ø - KI задан неверно.

Набор участков поверхностей в комплекте подпрограмм МС формируется как объект с типом 5Ø. Массив информационной части имеет длину $(N+1)$, где N - количество порций. Кроме того, объект имеет, как минимум, N ссылок вниз. Поэтому перед формированием набора порций с именем KI рекомендуется выделить память под него, обратившись к подпрограмме MC2:

CALL MC2(KI, N+1, Ø, N, 5Ø)

Подпрограмма MC2 описана в п. 2.2.1.3.

Участки поверхности KP должны иметь тип 31 или 41, т.е. быть полностью определенными. Пополнение набора KI участком KP осуществляется подпрограммой MC15Ø. Участки поверхности KP должны быть полностью определенными.

Обращение:

CALL MC15Ø(KI, KP)

где KI - имя набора, если KI = Ø, то MC15Ø образует набор из одного участка;

KP - имя участка поверхности, тип KP равен 31 или 41.

Результат:

KIEL(1Ø)=Ø - набор пополнен;

KIEL(1Ø)≠Ø - если KI = Ø, то нет места под формируемый набор;
если KI ≠ Ø, KP не включен в набор KI.

Для вычисления координат точки на поверхности набора необходимо знать:

N - количество порций в наборе, KN - имя участка поверхности, которому принадлежит точка. Затем по параметрам (U, V) определяется точка. Эти операции реализуются подпрограммами MC151, MC152, MC155.

Обращение:

CALL MC151(KI, N)

где KI - имя набора участков поверхностей;

N - целое число, количество участков в наборе.

Результат:

N - количество участков в наборе.

Обращение:

CALL MC152(KI, N, KN)

где KI - идентификатор - имя набора, участков поверхностей;

N - номер (по порядку) участка в наборе;

KN - целое число, имя участка поверхности.

Результат:

KN - имя участка поверхности. Необходимо, чтобы участок с номером N (по порядку формирования) существовал.

Обращение:

CALL MC155(KI,N,U,V,R)

где KI - имя набора участков поверхностей;

N - номер участка в наборе;

U, V - вещественные числа, параметры точки;

R=R(3) - вещественный массив, в который записывается результат.

Результат:

KIEL(1 \emptyset)= \emptyset - точка вычислена, R=(X(U,V), Y(U,V), Z(U,V));

KIEL(1 \emptyset) $\neq\emptyset$ - точка не вычислена, ошибка в данных.

Очевидно, что если известно имя KN N-го участка, то координаты точки можно вычислить обращением к подпрограмме MC125.

2.2.2.5. Преобразование сдвига, масштабирования и поворота

2.2.2.5.1. Сдвиг и масштабирование

Указанные операции над кривыми, участками поверхностей и наборами порций осуществляются подпрограммами MC170, MC174 и MC175. Однако перед дальнейшим использованием объектов необходимо обратиться последовательно к подпрограммам MC171 для всех трех типов объектов, к MC3031 - для участка поверхности и к MC176 - для набора порций. Кривые, входящие в рассматриваемые объекты, после преобразования метятся, чтобы не быть преобразованными дважды. Подпрограмма MC171 эти метки снимает. Подпрограммы MC3031 и MC176 пересчитывают заново рабочие величины участков поверхностей.

Параметр L этих подпрограмм определяет операцию сдвига ($L=\emptyset$) на вектор (X, Y, Z) или масштабирования ($L=1$), т.е. умножение координат точки объекта на коэффициенты X, Y и Z.

Обращение:

CALL MC170(KI,X,Y,Z,L)

где KI - имя кривой;

X - вещественное число, параметр преобразования по оси X;

Y - вещественное число, параметр преобразования по оси Y;

Z - вещественное число, параметр преобразования по оси Z;

L - целое число, $L=\emptyset$ - сдвиг, $L=1$ - масштабирование.

Результат:

KIEL(1 \emptyset)= \emptyset - операция выполнена;

KIEL(1 \emptyset) $\neq\emptyset$ - операция не выполнена.

Обращение:

CALL MC174(KI,X,Y,Z,L)

где KI - имя участка поверхности;

X - вещественное число, параметр преобразования по оси X;

Y - вещественное число, параметр преобразования по оси Y;

Z – вещественное число, параметр преобразования по оси Z ;

L – целое число, $L \neq \emptyset$ – сдвиг, $L=1$ – масштабирование.

Результат:

$KIEL(1\emptyset)=\emptyset$ – операция выполнена;

$KIEL(1\emptyset)\neq\emptyset$ – операция не выполнена, объект портится.

Обращение:

CALL MC175(KI,X,Y,Z,L)

где KI – имя набора участков поверхности;

X – вещественное число, параметр преобразования по оси X ;

Y – вещественное число, параметр преобразования по оси Y ;

Z – вещественное число, параметр преобразования по оси Z ;

L – целое число, $L=\emptyset$ – сдвиг, $L=1$ – масштабирование.

Результат:

$KIEL(1\emptyset)=\emptyset$ – операция выполнена;

$KIEL(1\emptyset)\neq\emptyset$ – операция не выполнена, объект портится.

Обращение:

CALL MC171

Результат:

Сняты метки у кривых, кривая считается помеченной, если ее тип 11 или 21; снять метку означает изменить тип на $1\emptyset$ или $2\emptyset$.

Обращение:

CALL MC176(KI)

где KI – имя набора участков поверхностей.

Результат:

$KIEL(1\emptyset)=\emptyset$ – набор подготовлен к вычислениям;

$KIEL(1\emptyset)\neq\emptyset$ – ошибка в данных набора, например, перед этим не было обращения к MC171 или один из участков набора не определен и не сработало обращение к MC3031.

2.2.2.5.2. Поворот вокруг оси

Пусть задана ось, проходящая через точки $(X\emptyset, Y\emptyset, Z\emptyset)$ и $(X1, Y1, Z1)$ и угол ψ : $\cos=\cos(4), \sin=\sin(4)$. Тогда кривую, участок поверхности или набор порций можно повернуть вокруг оси против часовой стрелки на угол ψ , если смотреть от точки $(X1, Y1, Z1)$ в направлении точки $(X\emptyset, Y\emptyset, Z\emptyset)$. Такая операция реализуется подпрограммами MC180, MC184 и MC185. Однако перед дальнейшим использованием объектов необходимо последовательно обратиться к подпрограммам MC171 для всех трех типов объектов, к MC3031 – для участка поверхности и к MC176 – для набора порций.

Обращение:

CALL MC180(KI,X\emptyset,Y\emptyset,Z\emptyset,X1,Y1,Z1,COS,SIN)

где KI - имя кривой;

X \emptyset - }
Y \emptyset - }
Z \emptyset - }
X1 - }
Y1 - }
Z1 - }

вещественные параметры, определяющие ось вращения;

COS - косинус угла поворота;

SIN - синус угла поворота.

Результат:

KIEL(1 \emptyset)= \emptyset - поворот выполнен;

KIEL(1 \emptyset) $\neq\emptyset$ - ошибка в данных, например, не снята метка.

Обращение:

CALL MC184(KI,X \emptyset ,Y \emptyset ,Z \emptyset ,X1,Y1,Z1,COS,SIN)

где KI - имя участка поверхности;

X \emptyset - }
Y \emptyset - }
Z \emptyset - }
X1 - }
Y1 - }
Z1 - }

вещественные параметры, определяющие ось вращения;

COS - косинус угла поворота;

SIN - синус угла поворота.

Результат:

KIEL(1 \emptyset)= \emptyset - поворот выполнен;

KIEL(1 \emptyset) $\neq\emptyset$ - поворот не выполнен, объект портится.

Обращение:

CALL MC185(KI,X \emptyset ,Y \emptyset ,Z \emptyset ,X1,Y1,Z1,COS,SIN)

где KI - имя набора участков поверхности;

X \emptyset - }
Y \emptyset - }
Z \emptyset - }
X1 - }
Y1 - }
Z1 - }

вещественные параметры, определяющие ось вращения;

COS - косинус угла поворота;

SIN - синус угла поворота.

Результат:

- KIEL(1Ø)=Ø - поворот выполнен;
KIEL(1Ø)≠Ø - поворот не выполнен, объект портится.

2.2.2.6. Дублирование кривой, порции или набора

Дублирование объектов может понадобиться, например, при конструировании нового объекта при помощи преобразований ранее заданного объекта. Поэтому в комплекте подпрограмм МС предусмотрено дублирование кривой, порции или набора участков поверхности подпрограммами MC19Ø, MC194 и MC195.

Обращения ко всем подпрограммам дублирования практически не отличаются.

Обращение:

CALL MC19Ø(KI, KD)

CALL MC194(KI, KD)

CALL MC195(KI, KD)

где KI - имя кривой в MC19Ø;
имя участка поверхности в MC194;

имя наборов участков поверхности в MC195;

KD - имя дубля кривой в MC19Ø;

дубля участка поверхности в MC194;

дубля набора участков поверхностей в MC195;

начальное значение KD=Ø.

Результат:

KIEL(1Ø)=Ø - дубль создан;

KIEL(1Ø)≠Ø - дубль не создан.

2.2.3. Проектирование объектов на плоскость

В комплекте подпрограмм МС предусмотрена возможность объемного изображения ортогональных проекций кривой, участка поверхности и набора порций с помощью алгоритма "ореола". Для этого подпрограммы MC159, MC156 и MC158 подготавливают вещественный массив звеньев кривой, порции или набора участков поверхностей. Затем этот массив в качестве параметра необходимо использовать в подпрограмме MC16Ø, реализующей ореольный алгоритм. Предварительное преобразование исходного объекта или полученного по нему массива пространственных отрезков позволяет получить объемное изображение, проекцию на любую плоскость и в любом масштабе.

Обращение:

CALL MC159(KI, N, A, NR)

где KI - имя кривой;

N - целое число, количество точек на кривой, по которым строятся отрезки;

A=A(NR) - вещественный массив, в который запишутся отрезки;

NR - длина массива A, необходимо $NR \geq 6(N-1)$.

Результат:

- $KIEL(1\emptyset)=\emptyset$ — массив А построен, т.е. число занятых слов массива $NR=6(N-1)$;
 $KIEL(1\emptyset)\neq\emptyset$ — ошибка в данных, $NR=\emptyset$.

Обращение:

CALL MC156(KI,N,M,N1,M1,A,NR)

где KI — имя участка поверхности;

N — целое число, количество кривых по переменной U, т.е.
 $U=\text{const}$; $N \geq 2$;

M — целое число, количество кривых по переменной V, т.е.
 $V=\text{const}$; $M \geq 2$;

N1 — целое число, количество точек на кривых при $U=\text{const}$; $N1 \geq 2$;

M1 — целое число, количество точек на кривых при $V=\text{const}$; $M1 \geq 2$;

A=A(NR) — вещественный массив, в который залишутся ребра;

NR — длина массива A, необходимо $NR \geq 6 [N(N1-1)+M(M1-1)]$.

Задание параметров иллюстрируется на рис. 4.

На участке поверхности заданы $N=4$ кривых $U=\text{const}$, $M=3$ кривых $V=\text{const}$. На первых кривых строятся по два ребра, $N1=3$, на второй группе кривых также строятся два ребра, $M1=3$. Если взять $N1=M$, а $M1=N$, то совокупность отрезков образует четырехугольную сетку.

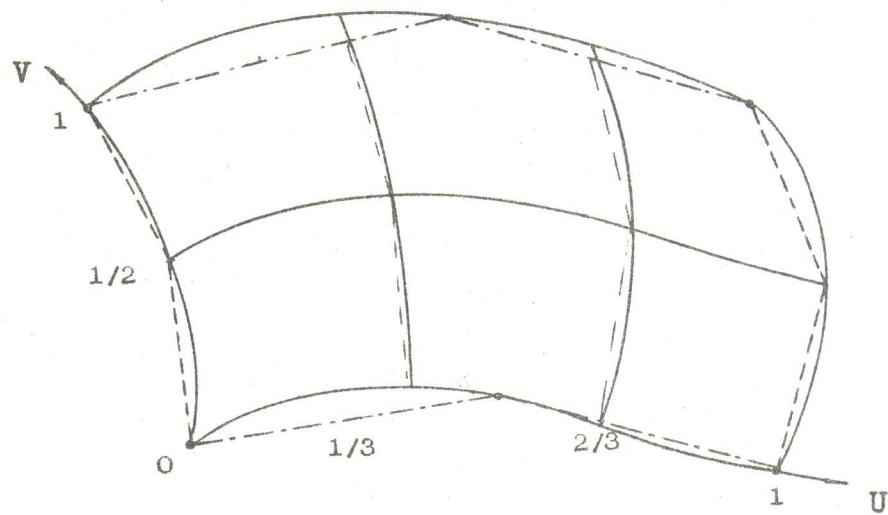


Рис. 4. Задание параметров

Результат:

- $KIEL(1\emptyset)=\emptyset$ - массив A построен, $NR=6 [N(N1-1)+M(M1-1)]$;
 $KIEL(1\emptyset)\neq\emptyset$ - ошибка в данных, $NR=\emptyset$.

Подпрограмма MC156 обращается к подпрограмме MC157, назначение и смысл параметров которой описаны в комментариях к тексту подпрограммы.

Обращение:

CALL MC158(KI,N,M,N1,M1,A, NR)

где KI - имя набора участков поверхности;

N, M, N1, M1, - целые параметры, определяющие сетку на участке набора KI (см. рис. 4);

A=A(NR) - вещественный массив, в который записутся ребра, NR - длина массива A; если в наборе KI К участков поверхности, то необходимо $NR \geq 6K[N(N1-1)+M(M1-1)]$.

Результат:

$KIEL(1\emptyset)=\emptyset$ - массив A построен, $NR=6[N(N1-1)+M(M1-1)]$;

$KIEL(1\emptyset)\neq\emptyset$ - ошибка в данных, $NR=\emptyset$.

Обращение:

CALL MC160(A, N, G, T)

где A=A(N) - вещественный массив, в который последовательно записаны координаты $N/6$ пространственных отрезков $\{(X_{i0}, Y_{i0}, Z_{i0}), (X_{i1}, Y_{i1}, Z_{i1})\} = P_i$, т.е. $A = \{P_1, P_2, \dots, P_M\}$, $M=N/6$;

N - длина массива A;

G - вещественный параметр, ширина ореола отрезка, т.е. каждый отрезок рассматривается как непрозрачная пластинка, проекция которой на плоскость XOY суть шестиугольник (рис. 5).

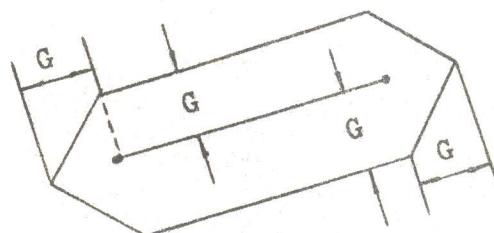


Рис. 5. Ореол отрезка

T - вещественный параметр, - толщина поверхности или пластинки ребра; назначение этого параметра заключается в том, что два отрезка считаются не затеняющими друг друга, если расстояние между ними не превышает T. T можно выбирать равным нулю.

Результат:

С помощью подпрограммы первого уровня системы СМОГ отрисовывается объемное изображение объекта - совокупности пространственных ребер; инициация системы, заказ области рисования и система декартовых координат должны быть определены до

обращения к подпрограмме MC160.

Подпрограмма MC160 обращается к подпрограммам MC161 и MC162.

2.2.4. Вычисление некоторых характеристик кривых, поверхностей и тел

При решении определенного класса задач приходится вычислять длины, площади, объемы, центры масс и моменты инерции. Часть этих задач может быть приближенно решена с помощью подпрограмм комплекта МС. Прежде всего в подпрограммах предполагается, что кривые, поверхности и тела однородны с плотностью, равной единице. Это означает, например, что масса поверхности равна ее площади, а масса тела — его объему. Во-вторых, кривые, участки или наборы участков поверхности описаны средствами комплекта подпрограмм МС. В-третьих, описанием тела является описание его поверхности как набора участков поверхностей без самопереесечений. Нарушение этого условия может привести к изменению вычисляемых характеристик. Так, например, если набор поверхностей описывает два пересекающихся тела, то плотность в области пересечения удвоится. Сформулированное условие распространяется на кривые и поверхности.

Приближенность вычислений обусловлена тем, что реально кривая $R(U), U \in (\emptyset, 1)$, заменяется набором отрезков $(R(U_i), R(U_{i+1})), U_i = i/(N-1), i = \emptyset, \dots, N-1$. Участок поверхности $R(U, V)$ заменяется набором треугольников с катетами $(R(U_i, V_i)), (R(U_i, V_{i+1}))$ и $(R(U_i, V_j)), (R(U_{i+1}, V_j))$, где $V_j = j/(M-1), j = \emptyset, \dots, M-1$. То есть каждый участок заменяется на совокупность из $2(N-1)(M-1)$ треугольников.

Приближенное вычисление массы (длины), центра масс и момента инерции относительно оси OZ кривой реализуется подпрограммой MC210.

Обращение:

CALL MC210(KI, S1, S2, N, R0, R1, R2)

где KI — имя кривой;

S1 — вещественный параметр; S1=1 — центр масс вычислять, S1=0 — центр масс не вычислять;

S2 — вещественный параметр; S2=1 — момент инерции вычислять, S2=0 — момент инерции не вычислять;

N — целое число, $N > 1$, определяет $(N-1)$ отрезков, которыми кривая приближается;

R0 — вещественный параметр, результат — масса кривой;

R1=R1(3) — вещественный массив, результат — центр масс кривой;

R2 — вещественный параметр, результат — момент инерции кривой относительно оси OZ.

Результат:

KIEL(1)=0 — R0, R1(3), R2 вычислены;

KIEL(1)≠0 — ошибка в задании кривой.

Приближенное вычисление массы, центра масс и момента инерции относительно оси OZ поверхности или тела реализуется подпрограммой MC200.

Обращение:

CALL MC200(KI, S0, S1, S2, N, M, R0, R1, R2)

- где K_1 - имя участка поверхности или набора участков поверхности;
- $S\emptyset$ - вещественный параметр; $S\emptyset=\emptyset$. - K_1 рассматривается как поверхность, $S\emptyset=1$. - K_1 рассматривается как тело;
- $S1$ - вещественный параметр; $S1=\emptyset$. - центр масс K_1 не считать, $S1=1$ - центр масс K_1 считать;
- $S2$ - вещественный параметр, $S2=\emptyset$ - момент инерции K_1 не считать, $S2=1$ - момент инерции K_1 считать;
- N - целое число; $N > 1$, определяет ($N-1$) разбиений на отрезки по переменной U каждого участка поверхности;
- M - целое число, $M > 1$, определяет ($M-1$) разбиений на отрезки по переменной V каждого участка поверхности; таким образом, каждый участок поверхности приближается $2(N-1)(M-1)$ плоскими треугольниками;
- $R\emptyset$ - вещественный параметр, результат - масса поверхности или тела K_1 ;
- $R1=R1(3)$ - вещественный массив, результат - центр масс поверхности или тела K_1 ;
- $R2$ - вещественный параметр, результат - момент инерции поверхности или тела K_1 .

Результат:

- $KIEL(1\emptyset)=\emptyset$ - $R\emptyset, R1, R2$ вычислены;
- $KIEL(1\emptyset)\neq\emptyset$ - ошибка в задании K_1 .

Подпрограмма $MC200$ обращается к служебной подпрограмме $MC201$, описание которой дано в комментариях к тексту подпрограммы.

Если требуется вычислить момент инерции объекта относительно произвольной оси, достаточно определить сдвиг и поворот объекта так, чтобы заданная ось в результате совпадала с осью OZ . Вычислить момент инерции и при необходимости выполнить обратные преобразования над объектом. Эту операцию, конечно, можно оформить отдельной подпрограммой, что в первой версии комплекта подпрограмм не сделано.

2.2.5. Описание осесимметричных участков поверхности

2.2.5.1. Поверхность вращения

В этом разделе вводятся два новых типа участков поверхности. На примере осесимметричных порций поверхности показано, насколько сложно или просто пополняется комплект подпрограмм MC .

Будем считать, что образующая поверхности вращения задана кривой K_1 , а ось поверхности - кривой K_2 . Естественно предполагать, что кривые K_1 и K_2 задаются подпрограммами комплекта MC , т.е. являются вектор-функциями параметра $V \in [\emptyset, 1]$. Поверхность K_1 как вектор-функция двух параметров $(U, V) \in (\emptyset, 1) \times (\emptyset, 1)$ получается при вращении образующей K_1 вокруг оси K_2 (против часовой стрелки). Максимальный угол поворота F (в градусах) соответствует значению параметра $U=1$. Угол F можно задавать как положительным, так и отрицательным. Внешняя нормаль к порции вращения может быть направлена либо от оси, либо к оси вращения. Итак, параметр V определяет пространственную точку кривой K_1 , параметр U определяет угол поворота $(U * F)$ этой точки вокруг кривой K_2 - оси, в результате получают точку поверхности вращения.

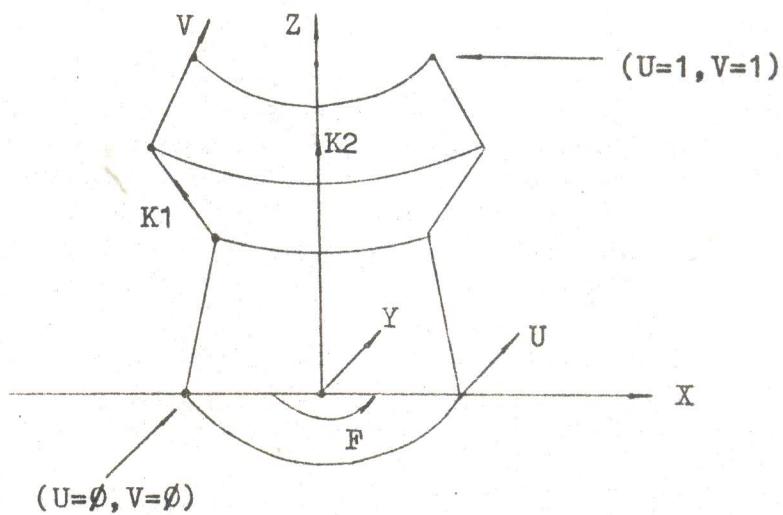
Для того, чтобы включить в комплект подпрограмм МС новый тип объекта, необходимо выполнить следующее:

- разработать подпрограмму построения объекта поверхности вращения, которая должна:
 - завести объект в архиве;
 - присвоить ему тип (выбрано значение 35),
 - сформировать массив информационной части, содержащий, например, имя кривой - образующий и определяющий ее направление параметр, имя оси и определяющий ее направление параметр, максимальный угол поворота и параметр, определяющий направление нормали;
- разработать подпрограмму определения точки на поверхности по параметрам (U, V) , а подпрограмму МС125 дополнить обращением к разработанной подпрограмме, если тип объекта равен 35;
- подпрограмму МС135 дополнить группой операторов переопределения нормали к поверхности;
- подпрограмму МС150 дополнить группой операторов, разрешающих включать в набор порций объект с типом 35;
- подпрограммы МС174 и МС184 дополнить группой операторов, разрешающих преобразование кривых порций с типом 35;
- подпрограмму МС194 дополнить группой операторов дублирования порции типа 35;
- подпрограмму МС200 дополнить группой операторов, разрешающих вычислять характеристики порции с типом 35;
- подпрограмму МС303 дополнить, если необходимо, группой операторов подготовки рабочих величин.

Последний этап для поверхности вращения отсутствует. Объем всех дополнений не превышает 50 операторов. Подпрограмма построения поверхности вращения содержит не более 25 операторов, подпрограмма вычисления точки на поверхности вращения - не более 25 операторов. На рис. 6 иллюстрируется построение поверхности вращения.

Ось вращения может отличаться от прямолинейного отрезка. Построение поверхности вращением кривой K_1 вокруг криволинейной оси K_2 не отличается по программной реализации от случая прямолинейной оси. Однако кривая K_2 , определяющая ось вращения, должна иметь непрерывные производные до второго порядка включительно. Кроме того, параметрические задания образующей K_1 и оси K_2 должны быть согласованы. Отрезок, соединяющий точки $K_1(V)$ и $K_2(V)$, должен быть ортогонален касательной к оси K_2 в точке $K_2(V)$. Точка $K_1(V)$ будет вращаться вокруг этой касательной. Более того, максимальное удаление точки $K_1(V)$ от $K_2(V)$ определяется радиусом кривизны K_2 . Чем меньше этот радиус, тем меньше должно быть удаление. Геометрически это требование можно сформулировать следующим образом. Пусть $R(V)$ - радиус круга с центром в точке $K_2(V)$, ортогонального касательной к кривой K_2 в этой же точке. Пусть никакие два круга $R(V_1)$ и $R(V_2)$ не имеют общих точек, если $V_1 \neq V_2$. Предположим, что для каждого V $R(V)$ является максимально возможным радиусом. Через $O(K_2)$ обозначим объединение всех кругов. Тогда образующая K_1 должна не выходить из $O(K_2)$ (см. рис. 7).

a) $F > \phi$, нормаль направлена от оси K2



b) $F < \phi$, нормаль направлена от оси K2

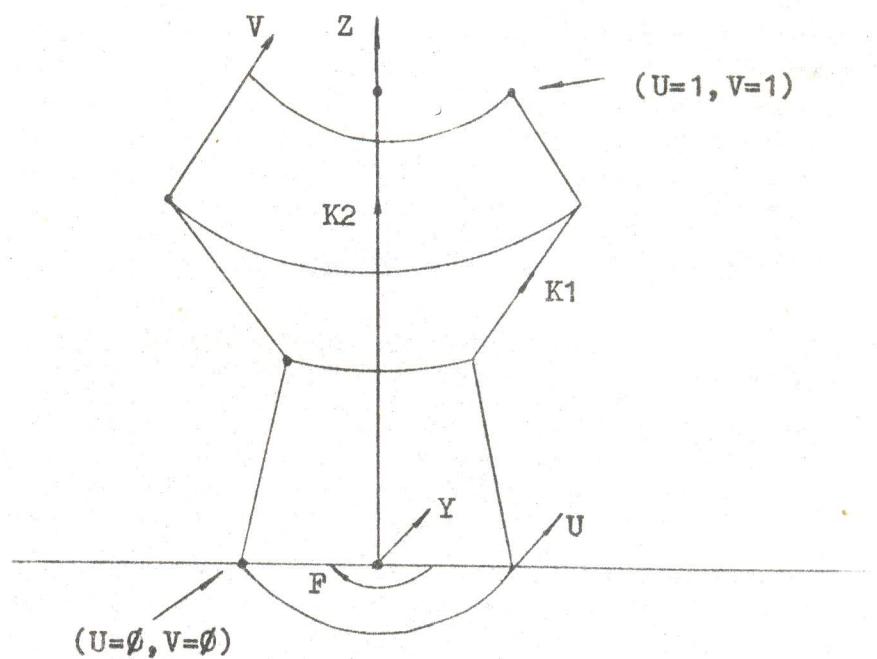
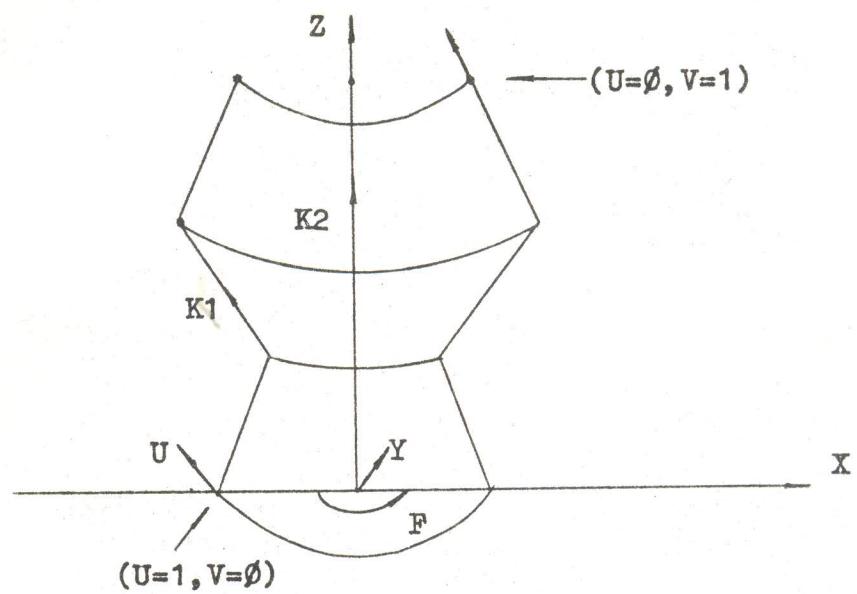


Рис. 6. Поверхность вращения кривой K1 вокруг оси K2

в) $F > \phi$, нормаль направлена к оси K2



г) $F < \phi$, нормаль направлена к оси K2

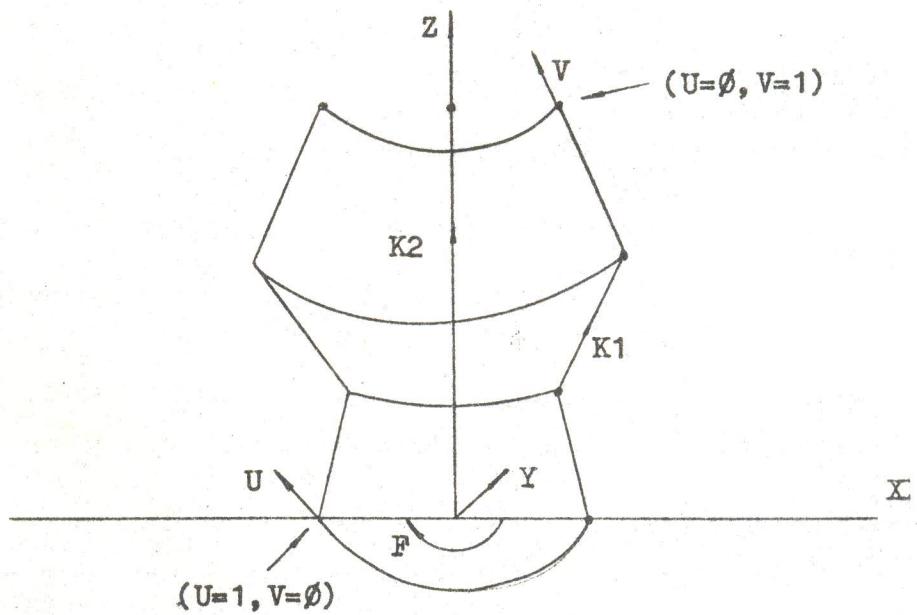


Рис. 6. Окончание

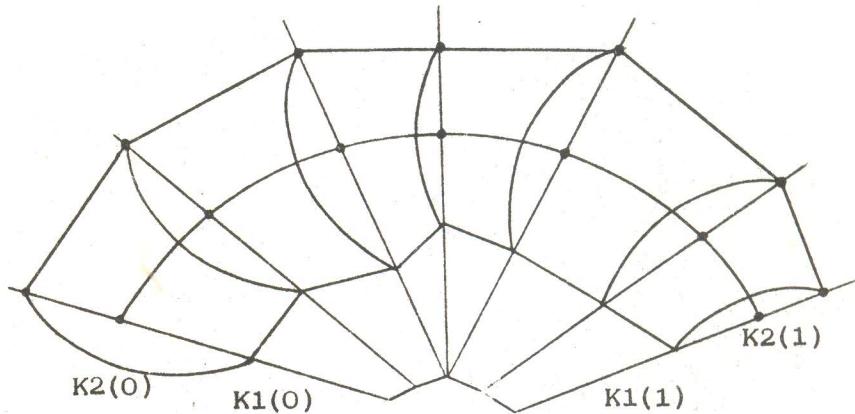


Рис. 7. Поверхность вращения кривой K_1 вокруг криволинейной оси K_2

Построение поверхности вращения кривой K_1 вокруг оси K_2 осуществляется по программе MC354Ø.

Обращение:

CALL MC354Ø(K1, K1, NK1, K2, NK2, F, N)

где K1 - имя объекта, начальное значение K1 = Ø;

K1 - имя кривой - образующей поверхности;

NK1 - целый параметр; NK1=1 означает, что направления кривой K1 и образующей совпадают, NK1=-1 - не совпадают;

K2 - имя кривой - оси поверхности;

NK2 - целый параметр; NK2=1 означает, что направления кривой K2 и оси поверхности совпадают, NK2=-1 - не совпадают;

F - вещественное число, угол максимального поворота образующей в градусах;

N - целый параметр, N=1 означает, что внешняя нормаль направлена от оси, N=-1 - нормаль направлена к оси.

Результат:

$KIEL(1\emptyset)=\emptyset$ - поверхность вращения построена, имя - K1;

$KIEL(1\emptyset)\neq\emptyset$ - поверхность вращения не построена; причина: например, нет места в архиве.

Любая точка поверхности характеризуется двумя параметрами U и V , изменяющимися от \emptyset до 1. Если V зафиксировать, а U менять от \emptyset до 1, получим окружность (или ее часть) в плоскости, ортогональной оси вращения. Если зафиксировать U , а изменять V от \emptyset до 1, получим одну из образующих поверхности вращения. Точка на поверхности может быть получена обращением к подпрограмме MC125 или к подпрограмме MC126 (K1, U, V, R). Параметры этих подпрограмм совпадают, но MC126 вычисляет точку только на поверхности типа 35.

2.2.5.2. Поверхность постоянного сечения

Пусть заданы кривые K_1 и K_2 . Предположим, что плоскость кривой K_1 ортогональна касательной к кривой K_2 в ее начальной точке. Тогда поверхностью постоянного сечения K_1 с осью K_2 назовем поверхность, образованную перемещением кривой K_1 вдоль оси K_2 таким образом, чтобы плоскость кривой K_1 , поворачиваясь, оставалась ортогональной касательной к кривой K_2 . Через PNV обозначим вектор нормали к плоскости кривой K_2 . Если K_2 – прямолинейный отрезок, то PNV – любой ортогональный ему вектор. Криволинейная ось K_2 должна иметь непрерывные производные до второго порядка включительно, а кривая K_1 при перемещении не должна выходить из $O(K_2)$, определенного в п. 2.2.5.1.

На рис. 8 показаны поверхности постоянного сечения с прямолинейной и криволинейной осями (или образующими).

Пополнение комплекта подпрограмм осуществляется по схеме, описанной в п. 2.2.5.1. Но в этом случае массив информационной части объекта (выбран тип 36) содержит кроме имен кривых, их направлений и вектора PNV еще 9 чисел: начальную точку оси K_2 , производную от кривой K_2 в начальной точке и их векторное произведение. Эти 9 чисел вычисляются подпрограммой $MC3031$. В свою очередь, подпрограмма $MC3031$ обращается к подпрограмме $MC3630$. Следовательно, перед вычислением точек на поверхности постоянного сечения или ее площади, момента инерции или центра масс необходимо подготовить данные о поверхности, обратившись к $MC3031$.

Построение поверхности постоянного сечения K_1 с осью K_2 осуществляется подпрограммой $MC3640$.

Обращение:

CALL $MC3640(KI, K1, NK1, K2, NK2, PNV)$

где KI – имя объекта, начальное значение $KI = \emptyset$;

$K1$ – имя плоской кривой – сечения поверхности;

$NK1$ – целый параметр, $NK1=1$ означает, что направление кривой $K1$ и сечения совпадают, $NK1= -1$ – не совпадают;

$K2$ – имя плоской кривой – оси (образующей) поверхности;

$NK2$ – целый параметр, $NK2=1$ означает, что направления кривой $K2$ и оси совпадают, $NK2= -1$ – не совпадают;

$PNV=PNV(3)$ – вещественный массив, вектор ортогональный плоскости кривой $K2$.

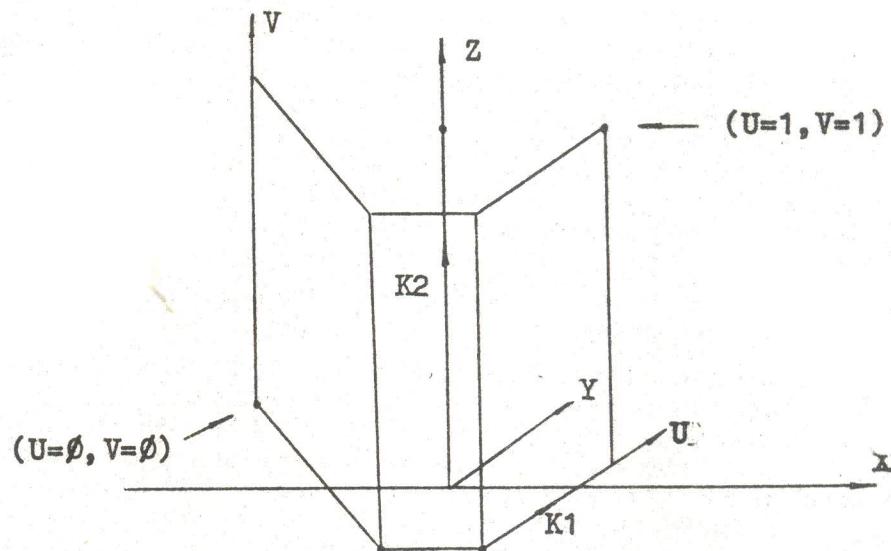
Результат:

$KIEL(1\emptyset)=\emptyset$ – поверхность постоянного сечения построена, имя KI ;

$KIEL(1\emptyset)\neq\emptyset$ – поверхность постоянного сечения не построена; причина: например, нет места в архиве.

Точка на поверхности постоянного сечения определяется подпрограммой $MC127 (KI, U, V, R)$. Пользователь должен считать ее служебной, так как подпрограмма $MC125$ вычисляет точку на любой поверхности, построенной средствами описанного комплекта подпрограмм.

a)



б)

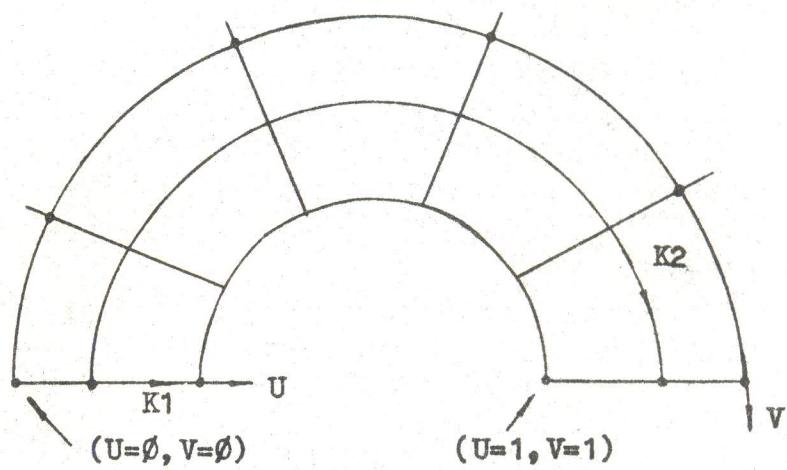


Рис. 8. Поверхность постоянного сечения K_1 с осью (образующей) K_2

2.2.6. Справочные таблицы

В табл. 1 приводятся минимально необходимые размеры выделяемых в COMMON AREL и COMMON KCEL участков под объекты, создаваемые подпрограммами комплекта МС. Эти сведения позволяют оценить объем памяти под архив в каждой конкретной задаче.

Таблица 1

Тип объекта	AREL	KCEL
10	$4 * N + 1$	0
20	$7 * N + 1$	0
30,31	20	6
40,41	20	6
35	6	4
36	16	4
50	$M + 1$	$M + 2$

В табл. 1 через N обозначено количество точек, по которым строится кривая (тип 10 или 20), через M – количество участков поверхности в наборе (тип 50).

В табл. 2 для каждой подпрограммы комплекта МС перечисляются вызываемые подпрограммы.

Таблица 2

Подпрограмма	Вызываемая подпрограмма
MC0	Нет
MC1	MC20, MC21
MC2	MC1
MC3	Нет
MC4	MC11
MC5	Нет
MC11	MC1, MC12
MC12	Нет
MC20	Нет
MC21	Нет
MC30	MC3
MC31	MC3, MC30
MC110	MC1, MC11
MC111	Нет
MC112	Нет

Продолжение табл. 2

Подпрограмма	Вызываемая подпрограмма
MC113	Нет
MC115	MC112, MC121, MC122
MC116	MC121, MC121, MC123
MC120	MC1, MC11
MC121	MC111
MC122	Нет
MC123	Нет
MC124	MC115
MC125	MC124, MC126, MC127
MC126	MC115, MC186
MC127	MC3031
MC135	MC2, MC4, MC11
MC150	Нет
MC151	Нет
MC152	MC125, MC152
MC155	MC125, MC157
MC156	MC125
MC157	MC151, MC152, MC156
MC158	MC115
MC159	MC161, MC162
MC160	Нет.
MC161	TRAD - подпрограмма СМОГ
MC162	Нет
MC170	Нет
MC171	MC170
MC174	MC174
MC175	MC3031
MC176	MC186
MC180	MC180
MC184	MC184
MC185	Нет
MC186	MC191
MC190	MC2
MC191	

Продолжение табл. 2

Подпрограмма	Вызываемая подпрограмма
MC194	MC4, MC30, MC190, MC191, MC3040
MC195	MC31, MC150, MC151, MC152, MC191, MC194
MC200	MC125, MC151, MC152, MC201
MC201	Нет
MC210	MC115
MC3031	MC115, MC3630
MC3040	MC1, MC4, MC5, MC30

2.2.7. Описание служебных подпрограмм

В этом разделе дано описание подпрограмм комплекта МС, используемых основными подпрограммами комплекта. Эти подпрограммы могут быть использованы при пополнении комплекта МС.

Подпрограмма MC111 предназначена для определения интервала, содержащего заданное значение параметра.

Обращение:

CALL MC111(A,N,M,U,I)

где A - одномерный вещественный массив длины $N*(M+1)$;

N - целое число, N-1 - количество интервалов (U_i, U_{i+1}) ;

- $i=1, \dots, N-1$; $U_i = A(1+(i-1) * (M+1))$;

M - целое число, определяет длину массива A;

U - вещественное число, значение заданного параметра;

I - целое число, результат - определяет интервал $[U_i, U_{i+1}] \ni U$.

Подпрограмма MC112 предназначена для вычисления точки на отрезке.

Обращение:

CALL MC112(A,U,R)

где A = A(8) - вещественный массив, $A=(U_1, R_1, U_2, R_2)$;

U - вещественный параметр, $U_1 \leq U < U_2$;

R=R(3) - вещественный массив, результат - $R=[(U-U_1) * R_2 + (U_2-U) * R_1] / (U_2-U_1)$.

Подпрограмма MC113 предназначена для вычисления производной в точке на отрезке.

Обращение:

CALL MC113(A,U,R)

где $A = A(8)$ – вещественный массив;

U – вещественный параметр, A и U определяются аналогично MC112;

$R = R(3)$ – вещественный массив, результат – $R = (R_2 - R_1) / (U_2 - U_1)$.

Подпрограмма MC121 предназначена для установления соответствия канонической параметризации кривой ($W \in (\emptyset, 1)$) с параметризацией, обусловленной ее построением.

Обращение:

CALL MC121(KI, W, M, KA, KIA, KNA, U)

где KI – целое число, имя кривой;

W – вещественное число, параметр кривой; $W \in (\emptyset, 1)$;

M – целое число; $M = 3$, если тип 10, $M = 6$, если тип 12;

KA – целое число, адрес начального значения параметра кривой;

KIA – целое число, адрес начала интервала параметра кривой;

KNA – целое число, адрес конечного значения параметра кривой;

U – целое число.

Подпрограмма MC122 предназначена для вычисления точки на кубическом отрезке.

Обращение:

CALL MC122(A, U, R)

где $A = A(14)$ – вещественный массив, $A = (U_1, R(U_1),$

$R'(U_1), U_2, R(U_2), R'(U_2))$;

U – параметр точки, $U_1 \leq U < U_2$;

$R = R(3)$ – вещественный массив, $R = R(U)$.

Подпрограмма MC123 предназначена для вычисления производной по параметру на кубическом отрезке.

Обращение:

CALL MC123(A, U, R)

где $A = A(14)$ – массив, задающий отрезок, $A = (U_1, R(U_1),$

$R'(U_1), U_2, R(U_2), R'(U_2))$;

U – параметр точки, $U_1 \leq U < U_2$;

$R = R(3)$ – вещественный массив, $R = R(U)$.

Подпрограмма MC124 предназначена для вычисления точки на границе участка поверхности типа 30 или 40.

Обращение:

CALL MC124(A, N, U, R)

где A – одномерный вещественный массив;

$A = A(i)$ – имя граничной кривой;

$A = A(i+1)$ – направление кривой;

N – целое число, длина массива A ;

U – вещественный параметр, $U \in (\emptyset, 1)$;

$R=R(3)$ - результат вычисления.

Подпрограмма MC157 предназначена для вычисления группы отрезков координатной кривой на участке поверхности.

Обращение:

CALL MC157(KI, H1, N1, U, K, L, A)

где KI - имя участка поверхности;

H1 - шаг изменения параметра;

N1 - количество точек на кривой;

U - фиксированный параметр задания участка поверхности;

K - целое число; определяет, первый или второй параметры фиксированы;

L - целое число, определяет направление кривой;

A - одномерный массив, в который записываются последовательно отрезки кривой, $A = (X_{10}, Y_{10}, Z_{10}, X_{11}, Y_{11}, Z_{11}, \dots, X_{MO}, Y_{MO}, Z_{MO}, X_{M1}, Y_{M1}, Z_{M1})$, $M=1-N1-1$.

Подпрограмма MC161 предназначена для определения общей точки двух отрезков на плоскости, используется только в MC160.

Обращение:

CALL MC161(P1, Q1, P2, Q2, NET, XX, YY)

где P1, Q1, P2, Q2 - вещественные числа, координаты отрезка;

XX, YY - вещественные числа, координаты точки пересечения;

NET - логический параметр, признак существования точки пересечения: $NET=\emptyset$ - не существует, $NET=1$ - существует.

Координаты концов второго отрезка передаются через бывшмянный COMMON блок.

Подпрограмма MC162 предназначена для отрисовки частей отрезка, применяется только в MC160.

Обращение:

CALL MC162(TE, IB)

где TE=TE(15, 2) - массив частей отрезка;

IB - целое число разбиений отрезка на рисуемые и нерисуемые участки.

MC162 обращается к подпрограмме TRAD системы СМОГ.

Подпрограмма MC186 предназначена для поворота точек вокруг оси.

Обращение:

CALL MC186(A, N, X \emptyset , Y \emptyset , Z \emptyset , X1, Y1, Z1, CA, SA)

где A=A(N) - массив N/3 точек (X_i, Y_i, Z_i), i=1, N/3

N - длина массива A;

X \emptyset , Y \emptyset , Z \emptyset - вещественные числа, координаты начала луча;

X1, Y1, Z1 - вещественные числа, координаты конца луча, вокруг которого против часовой стрелки (если смотреть из последней точки в начало луча) поворачиваются точки массива A, результат записывается в массив A;

CA, SA - косинус и синус угла поворота.

Подпрограмма MC191 предназначена для выделения памяти под дубль объекта.

Обращение:

CALL MC191(KI, KD)

где **KI** - имя объекта;

KD - имя объекта, выделенного под дубль **KI**, тип **KD** совпадает с типом **KI**, но в **KD** больше информации нет, только выделена память.

Подпрограмма MC201 предназначена для вычисления площадей, объемов, моментов и центров масс треугольника или тетраэдра, является частью подпрограммы MC200.

Обращение:

CALL MC201(R1, R2, R3)

R1=R1(3), R2=R2(3), R3=R3(3) - вершины треугольника или основания тетраэдра.

Остальные рабочие величины подпрограммы MC200 передаются через COMMON /CH1/ длиной 11 чисел.

В комплекте подпрограмм МС используются пять COMMON блоков: /AREL/, /KIEL/ и /KCEL/- для архива, /CH1/ - для передачи рабочих величин из подпрограммы MC200 в подпрограмму MC201 и безымянный блок для передачи рабочих величин из подпрограммы MC160 в подпрограммы MC161 и MC162.

2.2.8. Векторная алгебра

2.2.8.1. Векторная алгебра на плоскости

Все объекты, рассматриваемые в комплекте SP, базируются на точках (векторах), поэтому основой для их преобразований являются операции векторной алгебры, которые реализуются с помощью подпрограмм, описанных далее.

В операции векторной алгебры на плоскости точка представляется в виде массива из двух элементов - ее координат. В этом пункте идентификаторами A, B, C будут обозначаться точки, т.е. двухэлементные массивы. Если точка не равна (0, 0), рассматриваемые далее операции могут быть интерпретированы как операции над векторами.

Подпрограмма SP2001 предназначена для масштабирования вектора.

Обращение:

CALL SP2001(A, X, Y, C)

где A - массив, задающий вектор на плоскости;

X, Y - коэффициенты масштабирования;

C - массив, определяющий вектор-результат, C может равняться A.

Подпрограмма SP2002(A, X, Y, C) предназначена для сложения векторов.

Обращение:

CALL SP2002(A, X, Y, C)

где A - массив, задающий исходный вектор;

X, Y - координаты прибавляемого вектора;

C - массив, определяющий вектор-результат, C может равняться A.

Подпрограмма SP2003 предназначена для вращения точки относительно точки.

Обращение:

CALL SP2003(A,X,Y,RS,RC,C)

где A - массив, задающий вращаемую точку;

X, Y - координаты центра вращения;

RS, RC - величины, равные, естественно, sin и cos угла поворота;

C - массив, определяющий точку-результат, C может равняться A.

Подпрограмма SP2004 предназначена для отражения вектора относительно прямой.

Обращение:

CALL SP2004(A,X,Y,C)

где A - массив, задающий отражаемый вектор;

X, Y - координаты точки, определяющей вместе с точкой (0, 0) прямую, относительно которой происходит отражение;

C - массив, определяющий точку-результат, C может равняться A.

Если X = 0 и Y = 0, строится точка (-A(1), -A(2)), т.е. отражение относительно начала координат.

Подпрограмма SP2005 предназначена для нахождения перпендикуляра к вектору.

Обращение:

CALL SP2005(A,C)

где A - массив, задающий вектор;

C - массив, определяющий вектор, перпендикулярный к A, C может равняться A. Нормы векторов A и C совпадают. Перпендикуляр строится таким образом, что угол поворота от C к A равен 90° при отсчете против часовой стрелки в правой декартовой системе координат.

Подпрограмма SP2006 предназначена для скалярного произведения векторов.

Обращение:

CALL SP2006(A,B,R)

где A, B - массивы, задающие два вектора;

R - скалярное произведение векторов A и B.

Подпрограмма SP2007 предназначена для получения суммы модулей компонент вектора.

Обращение:

CALL SP2007(A,R)

где A - массив, задающий вектор;

R - сумма модулей его компонент.

Подпрограмма SP2008 предназначена для получения максимума модуля компонент вектора.

Обращение:

CALL SP2008(A,R)

где A - массив, задающий вектор;

R - значение максимальной по модулю компоненты.

Подпрограмма SP2009 предназначена для вычисления нормы вектора (длины).

Обращение:

CALL SP2009(A,R)

где A - массив, задающий вектор;

R - вычисленное значение его длины

Подпрограмма SP2010 предназначена для вычисления косинуса и синуса угла между векторами.

Обращение:

CALL SP2010(A,B,RS,RC)

где A,B - массивы, задающие два вектора;

RS,RC - вычисленные значения синуса (RS) и косинуса (RC) угла между векторами (от B к A против часовой стрелки).

2.2.8.2. Векторная алгебра в пространстве

Точки (векторы) задаются в правой декартовой системе координат, т.е. $\vec{k} = [\vec{i} \times \vec{j}]$, где $\vec{i}, \vec{j}, \vec{k}$ - единичные векторы по осям системы координат. В этом пункте идентификаторами A, B, C будут обозначаться точки (векторы), т.е. трехэлементные массивы, содержащие координаты.

Подпрограмма SP3001 предназначена для масштабирования вектора.

Обращение:

CALL SP3001(A,X,Y,Z,C)

где A - массив, задающий вектор;

X,Y,Z - коэффициенты масштабирования по осям;

C - массив, в который будет помещено описание вектора-результата. A может равняться C.

Подпрограмма SP3002 предназначена для сложения вектора (смещение точки).

Обращение:

CALL SP3002(A,X,Y,Z,C)

где A - массив, задающий вектор;

X,Y,Z - значения, определяющие вектор смещения;

C - массив, в который будет помещено описание вектора-результата. A может равняться C.

Подпрограмма SP3003 предназначена для вращения вектора относительно оси.

Обращение:

CALL SP3003(A,B,RS,RC,C)

где A - массив, задающий вращаемый вектор;

B - массив, задающий вектор, который определяет ось вращения, проходящую через точку (0,0,0);

RS, RC - значения синуса (RS) и косинуса (RC) угла поворота. Угол отсчитывается против часовой стрелки, если смотреть из конца вектора B;

C - массив, в который помещаются координаты вектора-результата. C может равняться A.

Подпрограмма **SP3004** предназначена для векторного произведения.

Обращение:

CALL SP3004(A, B, C)

где **A, B** - массивы, задающие два перемножаемых вектора;

C - массив, в который помещаются координаты вектора-результата. C может равняться A.

Подпрограмма **SP3005** предназначена для построения вектора, перпендикулярного к прямой.

Обращение:

CALL SP3005(A, X, Y, Z, C)

где **A** - массив, задающий вектор, который определяет прямую, проходящую через точку **(0,0,0)**;

X, Y, Z - координаты точки;

C - массив, в который будут помещены координаты вектора, перпендикулярного к прямой и лежащего в плоскости, определяемой точкой **(X, Y, Z)** и прямой. Если к вектору C добавить вектор **(X, Y, Z)**, то получится точка, лежащая на прямой **(0,0,0), A**. C может равняться A.

Подпрограмма **SP3006** предназначена для скалярного произведения векторов.

Обращение:

CALL SP3006(A, B, R)

где **A, B** - массивы, задающие два вектора;

R - значение скалярного произведения векторов.

Подпрограмма **SP3007** предназначена для вычисления суммы модулей компонент вектора.

Обращение:

CALL SP3007(A, R)

где **A** - массив, задающий вектор;

R - значение суммы модулей его компонент.

Подпрограмма **SP3008** предназначена для вычисления максимальной по модулю компоненты вектора.

Обращение:

CALL SP3008(A, R)

где **A** - массив, задающий вектор;

R - значение максимальной по модулю компоненты вектора.

Подпрограмма SP3009 предназначена для вычисления нормы (длины) вектора.

Обращение:

CALL SP3009(A,R)

где A - массив, задающий вектор;

R - значение его нормы (длины).

Подпрограмма SP3010 предназначена для вычисления косинуса и синуса угла между векторами.

Обращение:

CALL SP3010(A,B,RS,RC)

где A,B - массивы, задающие два вектора;

RS,RC - значения синуса (RS) и косинуса (RC) угла между векторами.

2.2.9. Элементы плоских кривых

2.2.9.1. Общие сведения

Для удобства описания пространственных объектов в комплекте SP введены плоские кривые. В данной версии комплекта SP рассматриваются только кривые второго порядка. Кривая - это набор элементов или элементарных кривых. Каждая элементарная кривая - это связная кривая на плоскости без самопересечений, заданная в локальной системе координат через примитивы формы. Таким образом, элементарная кривая определяется локальным базисом на плоскости и набором примитивов, заданных в этом базисе. Примитивами являются дуги эллипса, окружности, параболы, гиперболы и многозвенная ломаная.

В комплекте SP имеются средства задания элементарных кривых и кривых в многоугольники, точнее в наборы ребер. Предполагается, что с помощью кривых будет описана граница области на плоскости, которая затем может быть преобразована в многоугольник (набор ребер) и использована для описания пространственных объектов. В последующих версиях предполагается возможность использования кривых для описания пространственных объектов без преобразования в многоугольник.

Кривые представляются в виде одномерных массивов, размещенных в оперативной памяти. Далее описываются структура представления кривых и подпрограммы работы с ними, т.е. их описания и аппроксимации.

2.2.9.2. Структура элементов и кривых

2.2.9.2.1. Структура примитивов

В комплекте SP выделены следующие примитивы: дуга окружности, дуга эллипса, дуга параболы, дуга гиперболы и ломаная линия. Каждый примитив задается массивом слов. Первое слово массива - тип примитива, а остальные - его параметры.

Дуга окружности. Тип - 5. Параметры: R - радиус; α - угол, определяющий начальную точку дуги; β - угол, задающий величину дуги. Структура массива: 5, R, α , β - всего 5 слов. Уравнение окружности: $X^2 + Y^2 = R^2$.

Дуга эллипса. Тип - 6. Параметры: A - полуось X; B - полуось Y;

α - угол, определяющий начальную точку; β - угол, задающий величину дуги.
Структура массива: 6, A, B, α , β - всего 5 слов. Уравнение эллипса:

$$X^2/A^2+Y^2/B^2=1.$$

Дуга параболы. Тип - 7. Параметры: Р - параметр из уравнения; α - угол, определяющий начальную точку дуги; β - угол, задающий величину дуги. Структура массива: 7, P, α , β - всего 5 слов. Уравнение параболы: $Y=PX^2$.

Дуга гиперболы. Тип - 8. Параметры: Р - параметр из уравнения; α - угол, определяющий начальную точку дуги; β - угол, задающий величину дуги. Структура массива: 8, P, α , β - всего 5 слов. Уравнение гиперболы: $Y=P/X$. Рассматривается только положительная ветвь гиперболы.

В описанных примитивах $_$ означает незадействованное слово массива, за-дающего примитив.

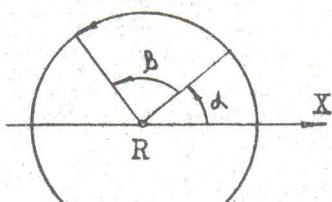
Ломаная линия. Тип - 9. Параметры: N - длина массива параметров; X_1, Y_1, X_2, Y_2 и т.д. - координаты точек ломаной линии. Структура массива: 9, N, $X_1, Y_1, \dots, X_{(N-1)/2}, Y_{(N-1)/2}$ - всего N+1 слово.

Во всех примитивах типа 5 - 8 углы задаются в радианах. Конечная точка дуги определяется углом $\alpha + \beta$. Правила отсчета угла α показаны на рис. 9 и рис. 10 (стрелками обозначена ориентация дуг для изображенных углов). Угол β отсчитывается от радиуса угла α против часовой стрелки, если $\beta > 0$, и по часовой стрелке, если $\beta < 0$.

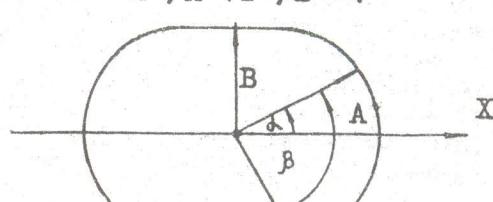
Границы изменения углов следующие: окружность ($-2\pi \leq \alpha \leq 2\pi, -2\pi \leq \beta \leq 2\pi$) эллипс ($-2\pi \leq \alpha \leq 2\pi, -2\pi \leq \beta \leq 2\pi$); парабола ($-\pi/2 \leq \alpha \leq \pi/2, -\pi/2 \leq \alpha + \beta \leq \pi/2$); гипербола ($0 < \alpha < \pi/2, 0 < \alpha + \beta < \pi/2$). При переориентации примитива происходит замена:

$$\alpha' = \alpha + \beta, \beta' = -\beta.$$

$$X^2/A^2+Y^2/B^2=1$$



$$\alpha > 0, \beta > 0$$



$$\alpha > 0, \beta < 0$$

Рис. 9. Отсчет углов для окружности и эллипса

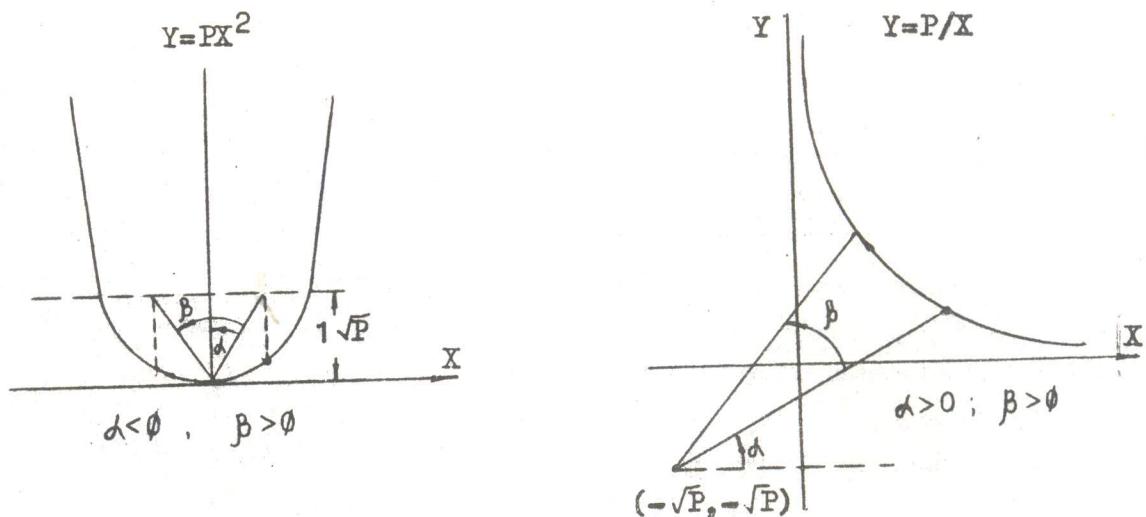


Рис. 10. Отсчет углов для параболы и гиперболы

Для примитива типа 9 задаются последовательно координаты точек ломаной линии. Ориентация от начальной точки к конечной. Отрезок – частный случай ломаной. Переориентация очевидна.

2.2.9.2.2. Структура элемента кривой

Элемент кривой – это набор примитивов, заданных в одной локальной системе координат. Элемент представляется в виде одномерного массива. Структура элемента следующая:

1-е слово – общая длина элемента в словах, включая данное;

2-7 слова – параметры, определяющие локальную систему координат;

8- N слова – последовательный набор примитивов, заданных, как было описано выше (N – значение первого слова).

Локальная система координат задается в виде трех двумерных векторов, т.е. набором из шести чисел $X_1, Y_1, X_2, Y_2, X_3, Y_3$. Здесь (X_1, Y_1) – точка начала локальной системы координат, заданная в общей системе координат плоскости; (X_2, Y_2) – точка, являющаяся концом единичного вектора оси X' локальной системы координат, заданная в общей системе координат плоскости; (X_3, Y_3) – точка, являющаяся концом единичного вектора оси Y' локальной системы координат, заданная в общей системе координат плоскости.

При аффинных преобразованиях элемента, не преобразованного в многоугольник, преобразуются только точки базиса, т.е. точки, определяющие локальную систему координат $X' Y'$.

Формулы перехода из локальной системы координат в глобальную и наоборот:

$$X = X_1 + X' (X_2 - X_1) + Y' (X_3 - X_1);$$

$$Y = Y_1 + X' (Y_2 - Y_1) + Y' (Y_3 - Y_1);$$

$$X' = (r - r_1, r_2 - r_1) / (r_2 - r_1, r_2 - r_1);$$

$$Y' = (r - r_1, r_3 - r_1) / (r_3 - r_1, r_3 - r_1);$$

где $r=(X, Y)$, $r1=(X_1, Y_1)$, $r2=(X_2, Y_2)$, $r3=(X_3, Y_3)$, $a(, ,)$ - скалярное произведение векторов.

Переориентация элемента состоит в переориентации составляющих его примитивов.

2.2.9.2.3. Структура кривой

Кривая состоит из набора элементов. Один элемент также может являться кривой. Задается кривая одномерным массивом, содержащим описание набора элементов. В силу того, что комплект SP носит базовый характер, в разных подпрограммах кривая может быть представлена следующим образом:

- 1) массив элементов и число элементов, описанных в нем;
- 2) 1-е слово - общая длина массива в словах, включая это слово;
2-е слово - число элементов у кривой;
- (3- N) слова - массив элементов, т.е. последовательно расположенные описания элементов (N - значение первого слова).

При описании подпрограмм вид задания кривой конкретизируется.

2.2.9.3. Построение элементов и кривых

В комплекте SP имеются подпрограммы для построения некоторых базисных элементов кривых и простейших кривых. В некоторых из описываемых далее подпрограмм встречаются параметры X , Y , AL , которые определяют локальную систему координат элемента. Здесь X , Y - положение начала локальной системы, а AL - угол в градусах поворота оси X локальной системы относительно оси X глобальной системы координат. Угол отсчитывается против часовой стрелки, если $AL > 0$, и наоборот. В тех подпрограммах, где параметры локальной системы координат отсутствуют, строится "пустое" описание, т.е. строится локальная система координат с параметрами $(0,0,0)$.

Подпрограмма SP2021 предназначена для построения дуги окружности.

Обращение:

```
CALL SP2021(X, Y, AL, R, AL1, AL2, C)
```

где X, Y, AL - параметры локальной системы координат;
 R - радиус окружности;
 $AL1$ - угол в градусах, определяющий начальную точку дуги;
 $AL2$ - угол в градусах, определяющий величину дуги; отсчет дуги производится против часовой стрелки, если $AL > 0$, и наоборот. Направление от начальной точки к конечной (рис. 11);
 C - массив, в который будет помещено описание элемента, $C(1)=12$.

Подпрограмма SP2022 предназначена для построения дуги эллипса.

Обращение:

```
CALL SP2022(X, Y, AL, A, B, AL1, AL2, C)
```

где X, Y, AL - параметры локальной системы координат;
 A, B - параметры эллипса, полуоси из уравнения
 $X^2/A^2 + Y^2/B^2 = 1$;

- AL1 — угол в градусах, определяющий начальную точку дуги;
 AL2 — угол в градусах, определяющий величину дуги;
 C — массив, в который будет помещено описание элемента,
 $C(1) = 12$.

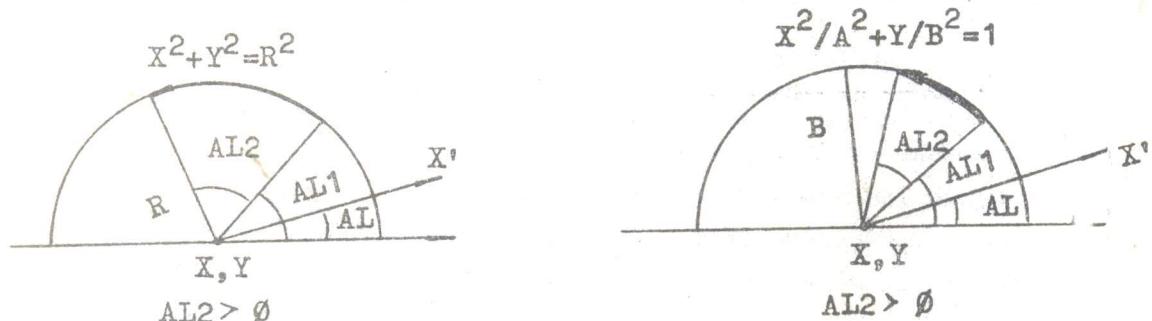


Рис. 11. Построение дуги окружности и эллипса

Подпрограмма SP2023 предназначена для построения дуги параболы.

Обращение:

CALL SP2023(X, Y, AL, P, XH, DX, C)

- где X, Y, AL — параметры локальной системы координат;
 P — параметр параболы из уравнения $Y=PX^2$;
 XH, DX — параметры, определяющие дугу параболы в соответствии с рис. 12; концевые точки дуги соответствуют параметрам XH и XH+DX;
 C — массив, в который будет помещено описание элемента, $C(1)=12$.

Подпрограмма SP2024 предназначена для построения дуги гиперболы.

Обращение:

CALL SP2024(X, Y, AL, P, XH, DX, C)

- где X, Y, AL — параметры локальной системы координат;
 P — параметр гиперболы из уравнения $Y=P/X$;
 XH, DX — параметры, определяющие дугу гиперболы в соответствии с рис. 12; начальная точка соответствует XH, конечная — XH+DX. Поскольку рассматривается только положительная ветвь, то должно выполняться условие: $XH, XH+DX > 0$;
 C — массив, в который будет помещено описание элемента,
 $C(1) = 12$.

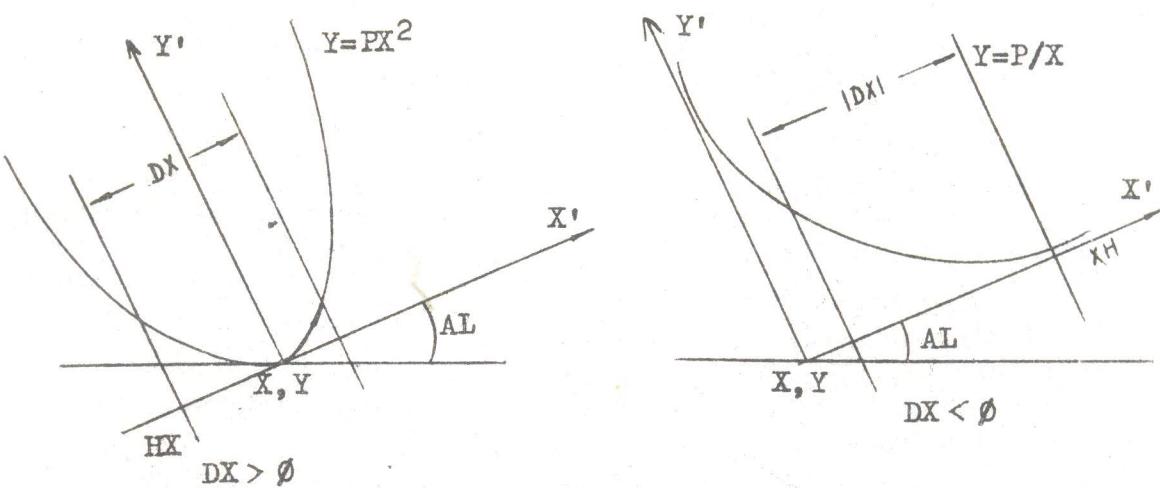


Рис. 12. Построение дуги параболы и гиперболы.

Подпрограмма **SP2030** предназначена для построения ломаной по массиву точек.

Обращение:

CALL SP2030(X, Y, AL, A, N, C)

где **X, Y, AL** – параметры локальной системы координат;

A=A(N) – массив, содержащий координаты точек ломаной, заданных в локальной системе координат, **N/2** точек;

N – количество координат точек ломаной в массиве **A**;

C – массив, в который будет помещено описание элемента,
C(1)=N+8.

Далее описываются подпрограммы, предназначенные для построения кривых, ограничивающих специальные области.

Подпрограмма **SP2025** предназначена для построения кривой, описывающей часть кольца.

Обращение:

CALL SP2025(R1, R2, AL, C, N, NC)

где **R1** – внутренний радиус кольца;

R2 – внешний радиус кольца;

AL – угол в градусах, определяющий часть кольца; отсчет угла стандартный;

C – массив, содержащий набор элементов, описывающих область – часть кольца (рис. 13);

N – число элементов в массиве **C**;

NC – общее число слов массива **C**.

Если **R1=0**, получим часть круга; если **|AL| > 360°**, получим полное кольцо. Ограничение **0 < R1 < R2**. Ориентация элементов, описывающих кольцо, соответствует правостороннему обходу области, т.е. внутренность локальна справа.

Подпрограмма SP2026 предназначена для построения кривой, описывающей часть круга.

Обращение:

CALL SP2026(R, H, DH, C, N, NC)

где R - радиус круга;

H, DH - параметры, выделяющие часть круга (рис. 13);

C - массив, содержащий набор элементов, описывающих область - часть круга;

N - число элементов в массиве C;

NC - общее число слов в массиве C.

Если $H = \emptyset$ и $DH > 2R$, получим полный круг.

Ограничения: $H \geq \emptyset$; $DH, R > \emptyset$. Ориентация элементов кривой соответствует правостороннему обходу границы области.

Подпрограмма SP2027 предназначена для построения кривой, описывающей часть оживала.

Обращение:

CALL SP2027(R, H, H1, DH, C, N, NC)

где R - радиус окружности;

H, H1, DH - параметры области в соответствии с рис. 13;

C - массив, содержащий набор элементов, описывающих границу области;

N - число элементов в массиве C;

NC - общее число слов в массиве C.

Ограничения: $R > \emptyset$; $H \leq 2R$; $H1 > \emptyset$; $DH > \emptyset$. Ориентация элементов кривой соответствует правостороннему обходу границы области.

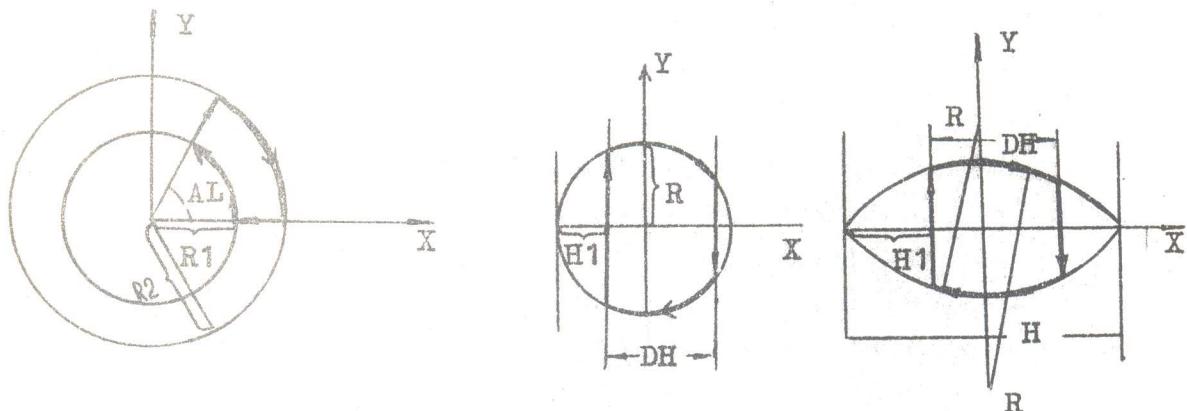


Рис. 13. Кривые, ограничивающие части кольца, круга и оживала

Подпрограмма **SP2028** предназначена для построения кривой, описывающей область, ограниченную параболой.

Обращение:

CALL SP2028(R, H, DH, C, N, NC)

где **R, H, DH** — параметры области в соответствии с рис. 14;

C — массив, содержащий набор элементов, описывающих границу области;

N — число элементов в массиве **C**;

NC — общее число слов в массиве **C**.

Ограничения: $R > \emptyset, DH > \emptyset, H \geq \emptyset$. Ориентация элементов соответствует правостороннему обходу области. Уравнение параболы

$$X = \frac{H_1 + H_2}{R^2} Y^2, \quad X = \frac{H + DH}{R^2} Y^2.$$

Подпрограмма **SP2029** предназначена для построения кривой, описывающей область, ограниченную эллипсом.

Обращение:

CALL SP2029(A, B, H, DH, C, N, NC)

где **A, B** — параметры эллипса, его полуоси;

H, DH — параметры области в соответствии с рис. 14;

C — массив, содержащий набор элементов, описывающих границу области;

N — число элементов в массиве **C**;

NC — общее число слов в массиве **C**.

Ограничения: $A, B, DH > \emptyset; H \geq \emptyset$. Ориентация элементов соответствует правостороннему обходу области.

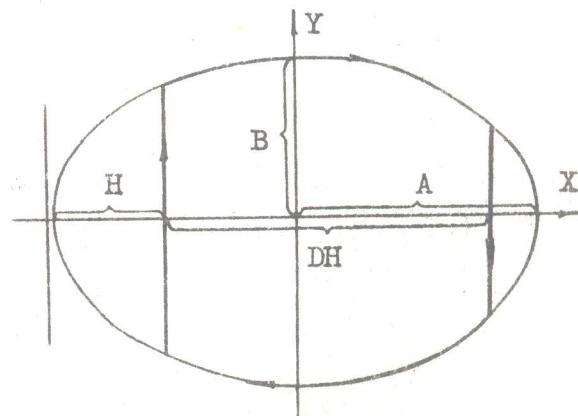
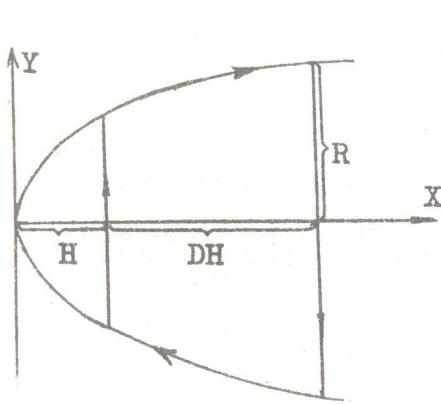


Рис. 14. Кривые, ограничивающие области параболы и эллипса

Следует отметить, что если кривая по каким-либо причинам не может быть построена, то на выходе в подпрограммах $SP2\theta 25 - SP2\theta 29$ $N < \emptyset$. Средства работы с кривыми, не преобразованными в многоугольники, в данной версии комплекта SP не рассматриваются.

2.2.9.4. Аппроксимация элементов и кривых

Для дальнейшей работы с построением кривой необходимо преобразовать ее в многоугольник. Далее описываются подпрограммы, которые дают базовые средства аппроксимации примитивов и кривых.

В п. 2.2.9.2 были определены канонические способы задания примитивов формы. Каждый примитив определяется углом. Аппроксимация примитива всегда равномерна по углу, как показано на рис. 15.

Канонические способы задания выбраны таким образом, чтобы обеспечить достаточно "хорошую" аппроксимацию при равномерном шаге по углу, и дают рекуррентные соотношения для последовательного вычисления точек на примитиве.



Рис. 15. Аппроксимация примитива кривой

Если исходным параметром для параметризации является E – величина максимального отклонения ломаной от кривой, то величина угла приращения определяется в точке максимальной кривизны для каждого типа примитива. В подпрограммах $SP21\theta 0 - SP21\theta 4$ рассматриваются примитивы типов 5 ÷ 8 (см. п. 2.2.9.2).

Подпрограмма $SP21\theta 0$ предназначена для определения длины ломаной и угла приращения для аппроксимации примитива по величине максимального отклонения.

Обращение

CALL $SP21\theta 0(A, E, N, DA)$

где $A=A(5)$ – массив, содержащий описание примитива в соответствии с п. 2.2.9.2;

E – величина максимального отклонения ломаной от дуги кривой второго порядка, определяемой A ;

N – число точек ломаной (включая концы дуги), аппроксимирующей дугу с точностью E ; параметр вычисляется;

DA – угол приращения параметра дуги в радианах, обеспечивающий эту аппроксимацию; параметр вычисляется.

Эта подпрограмма предназначена для использования совместно с подпрограммой $SP21\theta 3$.

Подпрограмма $SP21\theta 1$ предназначена для построения ломаной, приближаю-

шей дугу второго порядка с точностью Е.

Обращение:

CALL SP2101(A,E,B,M)

где A=A(5) - массив, содержащий описание примитива в соответствии с п. 2.2.9.2;

E - величина максимального отклонения ломаной от дуги кривой, определяемой A;

B=B(M) - массив координат точек ломаной, содержащий описание M/2 точек;

M - количество координат точек ломаной в массиве B.

Подпрограмма SP2102 предназначена для определения точки, лежащей на дуге кривой второго порядка, при параметризации примитива отрезком $[0,1]$.

Обращение:

CALL SP2102(A,T,X,Y)

где A=A(5) - массив, содержащий описание примитива в соответствии с п. 2.2.9.2;

T - параметр, определяющий точку примитива $T \in [0,1]$;

X, Y - координаты точки, лежащей на дуге, определяемые значением T.

Следует отметить, что при вычислении координат X, Y в этой подпрограмме считаются косинус и синус угла.

Подпрограмма SP2103 предназначена для определения точки примитива по известным точке и параметру угла приращения с использованием рекуррентных соотношений.

Обращение:

CALL SP2103(A,X,Y,RS,RC,X1,Y1)

где A=A(5) - массив, содержащий описание примитива в соответствии с п. 2.2.9.2;

X, Y - координаты точки, лежащей на дуге;

RS, RC - величины синуса и косинуса угла приращения (при использовании совместно с подпрограммой SP2100);

X1, Y1 - координаты вычисляемой точки.

Подпрограмма SP2104 предназначена для построения ломаной, аппроксимирующей дугу второго порядка, заданной некоторым количеством точек.

Обращение:

CALL SP2104(A,N,B,M)

где A=A(5) - массив, содержащий описание примитива в соответствии с п. 2.2.9.2;

N - число точек ломаной, включая концы, т.е. дуга разбита на N-1 звено равномерно по углу;

B=B(M) - массив, содержащий координаты точек ломаной, т.е. координаты N точек;

M=2 - длина массива B.

Далее описываются две подпрограммы, обеспечивающие преобразование кривой, состоящей из набора элементов, в многоугольник, т.е. в набор ребер (указан в п. 2.2.10.1). При этом происходит преобразование локальных координат точек элементов в глобальные координаты плоскости.

Подпрограмма **SP2Ø31** предназначена для определения общей длины массива, описывающего многоугольник (набор ребер), приближающий данную кривую с точностью Е.

Обращение:

```
CALL SP2Ø31(C,N,NC,E,M)
```

где **C** – массив набора элементов, описывающих кривую;

N – количество элементов в массиве **C**;

NC – общее число слов массива **C**;

E – величина максимального отклонения ребра от кривой;

M – общая длина массива ребер (в словах), приближающих данную кривую.

Эта подпрограмма предназначена для использования совместно со следующей.

Подпрограмма **SP2Ø32** предназначена для построения многоугольника (массива ребер) по заданной кривой и точности аппроксимации Е.

Обращение:

```
CALL SP2Ø32(C,N,NC,E,A,M)
```

где **C** – массив набора элементов, описывающих кривую;

N – количество элементов в массиве **C**;

NC – общая длина массива **C** в словах;

E – величина максимального отклонения ребра от кривой;

A – массив набора ребер, описывающих многоугольник, приближающий данную кривую;

M – общая длина массива **A** в словах.

2.2.10. Многоугольники на плоскости

2.2.10.1. Структура многоугольника

Многоугольник описывается одномерным массивом, содержащим координаты концов его ребер. Каждое ребро задается четверкой чисел. Многоугольники предлагаются заданными в плоскости ХУ в прямоугольной декартовой системе координат. Все ребра считаются направленными, причем направление ребер соответствует правостороннему обходу, т.е. при движении по ребру внутренность многоугольника локальна справа. В силу того, что многоугольник задается неупорядоченным набором ребер, возможно его поэтапное построение. Для этого, в частности, могут быть использованы подпрограммы **SPØØØ1** – **SPØØØ3** работы с массивами, которые описываются в п. 2.2.14, и подпрограммы **SP2Ø12**, **SP2Ø13**, описываемые далее.

Многоугольник задается массивом координат концов ребер А и его общей длиной в словах **N**, т.е. парой **A, N**. Каждое ребро в массиве А задается четырьмя словами.

В некоторых подпрограммах комплекта **SP** многоугольник задается в виде пары **A, M**, где **M** – число ребер, описанных в **A**. Тот или иной вид задания много-

угольника конкретизируется при описании подпрограмм.

Подпрограмма **SP2Ø12** предназначена для преобразования ломаной в многоугольник.

Обращение:

CALL SP2Ø12(A,N,B,M)

- где A=A(N) - массив, содержащий описание ломаной, т.е. координаты $N/2$ точек;
- N - количество координат ломаной в массиве A;
- B=B(M) - массив, в который будет помещено описание ребер многоугольника, построенного по ломаной A;
- M - общая длина массива B в словах. Можно: B = A, т.е. преобразование в одном массиве, $M=2N-4$.

Подпрограмма **SP2Ø13** предназначена для изменения ориентации многоугольника.

Обращение:

CALL SP2Ø13(A,N,B)

- где A=A(N) - массив набора ребер, описывающих многоугольник;
- N - общая длина массива A в словах;
- B - массив-результат, содержащий описание того же набора ребер, но с противоположной ориентацией. Можно: B = A, т.е. преобразование на месте.

В п. 2.2.11.3 описана подпрограмма **SP3Ø12**, объединяющая функции описанных подпрограмм **SP2Ø12, SP2Ø13**.

Для рисования многоугольников могут быть использованы подпрограммы **SP2Ø14** и **SP2Ø15**, которые описываются в части служебных подпрограмм.

2.2.10.2. Преобразования сдвига, поворота и масштабирования

Преобразования сдвига, поворота и масштабирования многоугольников выполняются в общей системе координат плоскости, т.е. в прямоугольной декартовой системе. Для реализации этих представлений в комплекте SP имеются четыре подпрограммы.

Подпрограмма **SP25Ø1** предназначена для сдвига многоугольника на вектор.

Обращение:

CALL SP25Ø1(A,N,X,Y)

- где A - массив набора ребер, описывающих многоугольник;
- N - длина массива A в словах;
- X, Y - параметры сдвига - координаты вектора.

Подпрограмма **SP25Ø2** предназначена для масштабирования многоугольника.

Обращение:

CALL SP25Ø2(A,N,X,Y)

где A = массив набора ребер, описывающих многоугольник;
 N = длина массива A в словах;
 X, Y = параметры масштабирования.

Подпрограмма SP2503 предназначена для поворота многоугольника вокруг заданной точки на заданный угол.

Обращение:

CALL SP2503(A,N,X,Y,RC,RS)

где A = массив набора ребер, описывающих многоугольник;
 N = длина массива A в словах;
 X, Y = координаты точки — центра вращения;
 RC, RS = параметры, определяющие угол поворота: $RC=\cos(\alpha)$,
 $RS=\sin(\alpha)$, где α — угол поворота. Вращение против часовой стрелки.

Подпрограмма SP2504 предназначена для сложного преобразования многоугольника.

Обращение:

CALL SP2504(A,N,X1,Y1,X2,Y2,X3,Y3,X4,Y4)

где A = массив набора ребер, описывающих многоугольник;
 N = длина массива A в словах;
 X1, Y1, X2, Y2 = координаты двух точек на плоскости, определяющие исходный отрезок;
 X3, Y3, X4, Y4 = координаты двух точек на плоскости, определяющие результирующий отрезок.

Это преобразование выполняется следующим образом:

- многоугольник сдвигается на вектор $(X3-X1, Y3-Y1)$;
- многоугольник поворачивается относительно точки $(X3, Y3)$ на угол, равный углу между векторами $(X2-X1, Y2-Y1)$ и $(X4-X3, Y4-Y3)$, чтобы исходный и результирующий отрезки совпали по направлению;
- многоугольник масштабируется таким образом, чтобы точки $(X2, Y2)$ и $(X4, Y4)$ совпали.

Другими словами, многоугольник и исходный отрезок $(X1, Y1, X2, Y2)$ рассматриваются как единое целое, и данное преобразование обеспечивает переход исходного отрезка в результирующий. Ограничение: $(X1, Y1) \neq (X2, Y2)$ и $(X3, Y3) \neq (X4, Y4)$. Если это условие не выполняется, то подпрограмма работает впустую, т.е. ничего не делает.

Следует отметить, что описанные подпрограммы преобразований не учитывают структуру массива A, а рассматривают его как массив координат точек на плоскости и, следовательно, могут быть использованы для преобразований, в частности, ломаных.

2.2.10.3. Теоретико-множественные операции над многоугольниками

Четыре подпрограммы, реализующие теоретико-множественные операции над многоугольниками, работают, используя три базовые подпрограммы **SP2401**, **SP2402**, **SP2403**, которые фактически реализуют основной алгоритм. Эти подпрограммы описаны в разделе служебных, но могут быть использованы самостоятельно.

Во всех описываемых в этом пункте подпрограммах параметры одни и те же:

- A=A(N)** - массив набора ребер, описывающих первый многоугольник;
- N** - число ребер в массиве A;
- B=B(M)** - массив набора ребер, описывающих второй многоугольник;
- M** - число ребер в массиве B;
- C=C(L)** - массив, в который будет помещено описание многоугольника, являющееся результатом теоретико-множественной операции над двумя исходными;
- L** - длина массива C, предоставляемая пользователем.

Если результат операции не помещается в массив **C(L)**, операция не выполняется и на выходе из подпрограмм L будет < 0; иначе в L будет помещено значение длины массива, описывающего результирующий многоугольник.

Подпрограмма **SP2410** предназначена для пересечения двух многоугольников.

Обращение:

CALL SP2410(A,N,B,M,C,L)

Подпрограмма **SP2411** предназначена для объединения двух многоугольников.

Обращение:

CALL SP2411(A,N,B,M,C,L)

Подпрограмма **SP2412** предназначена для получения разности двух многоугольников.

Обращение:

CALL SP2412(A,N,B,M,C,L)

На выходе получается результат вычитания второго многоугольника из первого.

Подпрограмма **SP2413** предназначена для специального вычитания многоугольника.

Обращение:

CALL SP2413(A,N,B,M,C,L)

В результате работы этой подпрограммы получается набор ребер первого многоугольника или их частей, не лежащих внутри второго, т.е. второй многоугольник является как бы экраном. Результирующий набор ребер не является многоугольником.

Следует отметить, что во всех этих подпрограммах перед выполнением операции происходит дискретизация исходных многоугольников с помощью подпрограммы

SP2403, обращение к которой имеется в **SP2402**. Эти подпрограммы описаны в п. 2.2.14.

2.2.11. Многогранники в пространстве

2.2.11.1. Общие сведения

Многогранник в комплекте **SP** задается своей ориентированной границей, т.е. набором граней. Каждая грань представляет собой плоский многоугольник, но заданный в пространстве, т.е. составляющие его ребра являются пространственными отрезками, каждый из которых определяется шестью числами – координатами его концов. Ориентация грани, т.е. вектор нормали к ней, определяется как нормированный вектор площади грани, вычисляемый по ее направленной границе. Таким образом, ориентация грани индуцируется ориентацией ребер, ее образующих. В комплекте **SP** предполагается, что индуцированные ориентации граний, задающих многогранник, направлены локально внутрь многогранника.

Далее будут описаны способы представления многогранника в памяти ЭВМ и подпрограммы комплекта **SP**, обеспечивающие построение и обработку многогранников.

2.2.11.2. Структура многогранника

2.2.11.2.1. Структура грани

Грань описывается одномерным массивом, содержащим координаты концов ее ребер. Каждое ребро задается шестеркой чисел. Грань предполагается заданной в правосторонней прямоугольной декартовой системе координат. Все ребра грани считаются направленными. Если поместить грань в плоскость XY так, чтобы ее "внутренность" соответствовала правостороннему обходу ребер, нормаль к ней будет $(0, 0, -1)$.

В подпрограммах грань задается двумя способами:

- 1) массивом координат концов ребер А и его общей длиной в словах **N**, т.е. парой **A, N** (число ребер будет $N/6$);
- 2) массивом **A**, первое слово которого содержит значение **N + 1**, а начиная со второго слова идет описание ребер грани, т.е. координаты концов $N/6$ ребер.

Общая длина массива А во втором случае равна **A(1)**.

В некоторых служебных подпрограммах грань задается массивом описания ее ребер **A** и числом ребер **K**. При описании подпрограмм вид задания грани конкретизируется.

2.2.11.2.2. Структура набора граний

В подпрограммах набор граний задается двумя способами:

- 1) массивом **A**, содержащим описание граний, входящих в набор (каждая грань описывается по способу 2 (см. п. 2.2.9.2.3)), и количеством граний в наборе, т.е. парой **A, N**;
- 2) массивом **A**, первое слово которого содержит общую длину массива, второе – число граний в наборе, а с третьего слова начинается последовательное описание граний набора, каждое по способу 2 (см. п. 2.2.9.2.3).

Такое задание многогранника не накладывает ограничений на его замкнутость, т.е. многогранная поверхность, не обязательно замкнутая, может быть задана таким же образом. Это свойство используется в комплекте **SP**, поскольку большинство подпрограмм комплекта работают именно с набором граней, не предполагая, что они образуют замкнутую поверхность. Иногда задание способа 1 (см. п. 2.2.9.2.3) модифицируется, и вместо количества граней задается общая длина массива A.

2.2.11.3. Построение многогранников

В силу того, что многогранник задается неупорядоченным набором граней, возможно его поэтапное построение. Для этого, в частности, могут быть использованы подпрограммы работы с массивами **SP0001** + **SP0003**, которые описываются в служебных подпрограммах. Далее описываются подпрограммы, предназначенные для построения как отдельных граней, так и наборов пространственных граней.

Подпрограмма **SP2301** предназначена для преобразования массива двумерных точек в массив пространственных точек.

Обращение:

CALL **SP2301(A,N,B,M,R,I)**

где **A=A(N)** - массив, содержащий координаты двумерных точек, т.е. координаты $N/2$ точек;
N - длина массива A;
B=B(M) - массив, в который будут помещены координаты $M/2$ пространственных точек, полученных из заданных $N/2$ точек;
M - общая длина результирующего массива B, она насчитывается в подпрограмме и равна $(N/2) * 3$;
R - значение добавляемой координаты, у всех точек одно и то же;
I - признак:
 I = 1 - точка X, Y преобразуется в R, X, Y;
 I = 2 - точка X, Y преобразуется в X, R, Y;
 I = 3 - точка X, Y преобразуется в X, Y, R.

Эта подпрограмма может быть использована для построения грани по многоугольнику, т.е. двумерному объекту.

Подпрограмма **SP3012** предназначена для преобразования ломаной в набор ребер, возможно, с изменением ориентации.

Обращение:

CALL **SP3012(A,N,B,M,I)**

где **A=A(N)** - массив, содержащий координаты точек ломаной, т.е. координаты $N/|I|$ точек ($I=\pm 2, \pm 3$);
N - длина массива A;
B=B(M) - массив, в который будут помещены координаты концов ребер, являющихся звеньями ломаной;
M - общая длина массива B, она насчитывается, $M=2 * (N/|I|)$;
I - признак изменения ориентации и указатель размерности точек;

$I = 2$ - двумерную ломаную перевести в многоугольник, т.е.
в набор ребер, без изменения ориентации;

$I = -2$ - то же, что и предыдущее, но с изменением ориентации;

$I = 3$ - трехмерную ломаную перевести в грань, т.е. в набор
ребер, без изменения ориентации;

$I = -3$ - то же, что и предыдущее, но с изменением ориентации.

Эта подпрограмма может быть использована для построения грани по замкнутой
пространственной ломаной.

Подпрограмма SP3013 предназначена для изменения ориентации грани.

Обращение:

CALL SP3013(A, N, B)

где $A=A(N)$ - массив ребер, $N/6$ ребер;

N - длина массива A ;

B - массив, в который будут помещены ребра из A с изменен-
ной ориентацией.

Подпрограмма SP3014 предназначена для определения вектора нормали к
грани.

Обращение:

CALL SP3014(A, N, X, Y, Z)

где $A=A(N)$ - массив ребер, задающих грань, $N/6$ ребер;

N - длина массива A ;

X, Y, Z - вычисляемые координаты вектора нормали. Вектор нормали
вычисляется по формуле:

$$(X, Y, Z) = \sum_i [V_{i1} * V_{i2}]$$

где V_{i1} и V_{i2} - векторы (точки) концов i -го
ребра, а $[*]$ - векторное произведение. Сумма берется
по всем ребрам из A . Результирующий вектор не норми-
руется.

Далее описываются подпрограммы, предназначенные для построения наборов
граней, являющихся поверхностью и телом вращения обобщенного цилиндра и обоб-
щенного конуса. Все описываемые здесь подпрограммы основаны на подпрограммах
SP3020, SP3021, SP3024.

Подпрограмма SP3022 предназначена для построения поверхности, являю-
щейся смещением заданного набора ребер на заданный вектор.

Обращение:

CALL SP3022(A, N, B, M, V, I)

где $A=A(N)$ - массив пространственных ребер, $N/6$ ребер, лежащих
в одной плоскости;

N - длина массива A ;

$B=B(M)$ - массив, в который будет помещено описание набора M
граней;

M — общая длина массива B ;

$V=V(3)$ — вектор смещения, он не должен быть параллелен плоскости ребер и иметь нулевую длину;

I — признак:

$I = 1$ — ориентацию не менять;

$I = -1$ — ориентацию менять на противоположную, как показано на рис. 16.

Длина массива B вычисляется по формуле $M=N * 4 + N/6$.

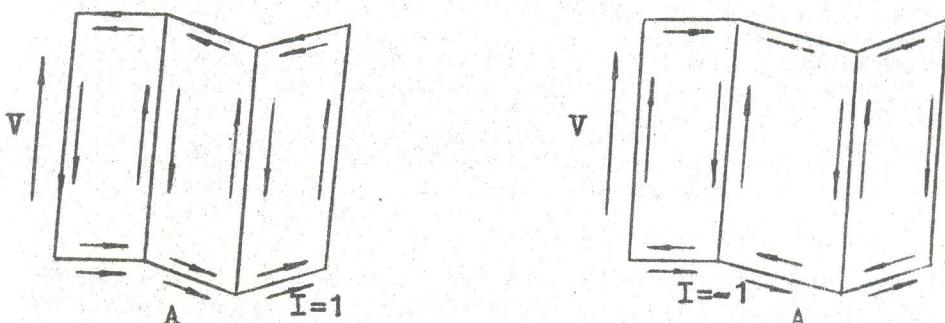


Рис. 16. Построение поверхности смещения набора ребер по вектору

Подпрограмма **SP3023** предназначена для построения поверхности по набору ребер и точке.

Обращение:

CALL SP3023(A, N, B, M, V, I)

где $A=A(N)$ — массив пространственных ребер, $N/6$ ребер, лежащих в одной плоскости;

N — длина массива A ;

$B(M)$ — массив, в который будет помещено описание набора M граней;

M — общая длина массива B ;

$V(3)$ — точка, она не должна лежать в плоскости ребер;

I — признак:

$I=1$ — ориентацию не менять;

$I=-1$ — ориентацию менять на противоположную, как показано на рис. 17.

Длина массива B вычисляется по формуле $M=N * 3 + N/6$.

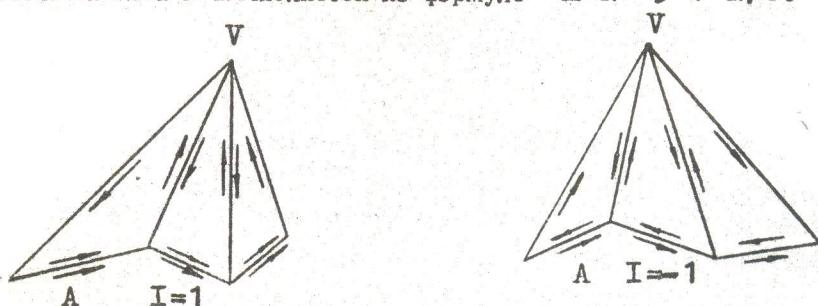


Рис. 17. Построение поверхности по набору ребер и точке

Подпрограмма SP3025 предназначена для построения поверхности вращения по набору ребер и параметрам угла.

Обращение:

CALL SP3025(A, N, B, M, T, V, RS, RC, K)

где A=A(N) - массив пространственных ребер, N/6 ребер, лежащих в одной плоскости;

N - длина массива A;

B(M) - массив, в который будет помещено описание набора граней;

M - общая длина массива B;

T=T(3), V=V(3) - точка и направляющий вектор прямой, определяющие ось вращения, которая не должна быть перпендикулярна плоскости ребер,
V - единичный вектор;

RS, RC - параметры угла приращения RS=SIN α , RC=COS α ,
где α - угол приращения, т.е. элемент вращения;

K - число элементов вращения (рис. 18).

Длина массива B вычисляется по формуле $M=4 * L * N + L * N/6$.

Это

максимальная длина массива B. Если ребра касаются оси или лежат на оси, величина M может быть меньше. Вращение осуществляется против часовой стрелки, если смотреть вдоль оси вращения против вектора V. Ориентация граней показана на рис. 18.

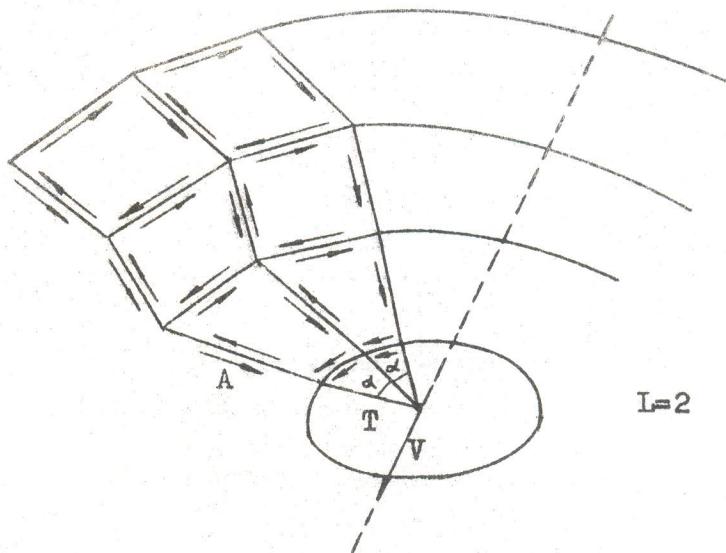


Рис. 18. Построение поверхности вращения

Подпрограмма SP3031 предназначена для построения многогранника призмы по заданной грани - основанию и вектору смещения.

Обращение:

CALL SP3031(A, N, VH, V, B, K, M)