

Ю.В. КОТОВ

КАК РИСУЕТ МАШИНА



МОСКВА "НАУКА"
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
1988

ББК 22.18
К73
УДК 519.6

Котов Ю.В. Как рисует машина. — М.: Наука. Гл. ред. физ.-мат. лит., 1988. — 224 с. ISBN 5-02-013791-X

Дается обзор графических устройств разных типов и их возможностей, описываются принципы моделирования в ЭВМ графических и геометрических образов, составления программ, решающих геометрические задачи. Отдельные главы посвящены широко используемым в машинной графике преобразованиям координат и построению проекций.

Может быть использована студентами вузов для первичного ознакомления с вопросами машинной графики и геометрии, слушателями факультетов повышения квалификации, а также в кружковой работе школ и ПТУ.
Табл. 7 Ил. 182 Библиогр. 6 назв.

Рецензенты доктор технических наук *Н.Н. Шапошников*, кандидат физико-математических наук *А.Б. Сосинский*

1702070000-088
К 053 (02)-88 55-88

ISBN 5-02-013791-X

© Издательство "Наука".
Главная редакция
физико-математической
литературы, 1988

ОГЛАВЛЕНИЕ

Предисловие	5
Введение	7
Глава 1. СЕМЬЯ ГРАФИЧЕСКИХ УСТРОЙСТВ	15
1.1. Виды изображений	15
1.2. Предшественники графопостроителей и чертежных машин. Самописцы	18
1.3. Современный графопостроитель	23
1.4. Полуавтоматическое чтение чертежа	32
1.5. Мозаика в ЭВМ. Предшественники растровых устройств	34
1.6. Электронно-лучевые устройства, графические терминалы	38
1.7. Персональные графические компьютеры	42
1.8. Специальное оборудование	43
1.9. Автоматизированные рабочие места (АРМ)	44
Глава 2. НАЧИНАЯ С ПРОСТЫХ КАРТИНОК	47
2.1. Программное формирование и геометрическое описание изображений. Базовое программное обеспечение	47
2.2. Геометрия + алгебра = аналитическая геометрия?	52
2.3. Первые примеры кодирования изображений	61
2.4. Типовая деталь	73
2.5. Узоры	76
Глава 3. ФУНКЦИИ И КРИВЫЕ ЛИНИИ	81
3.1. Вычерчивание кривых линий и графиков функций, задаваемых уравнениями в явном виде	81
3.2. Построение параметрических кривых	88
3.3. Построение линий по заданным условиям	99
3.4. Составные линии	105
Глава 4. СЕМЕЙСТВА ЛИНИЙ	113
4.1. Нормали и касательные	113
4.2. Эквидистанты	115
4.3. Кинематические кривые.	118
4.4. Пропорциональные кривые	119
4.5. Кривые влияния	121

Глава 5. ПРЕОБРАЗОВАНИЯ КООРДИНАТ	124
5.1. Аффинные преобразования двух координат	130
5.2. Дополнительные сведения о языке описания геометрической информации и задание на нем преобразований координат	139
5.3. Полярные координаты	143
Глава 6. ТРЕХМЕРНОЕ ПРОСТРАНСТВО	149
6.1. Двумерные фигуры в трехмерном пространстве.	149
6.2. Пространству – пространственные линии	151
6.3. Кубики и другие фигуры, из которых можно что-то смастерить	159
Глава 7. ТРЕХМЕРНЫЕ ПРЕОБРАЗОВАНИЯ КООРДИНАТ	164
7.1. Аффинные преобразования в пространстве	164
7.2. Нелинейные координаты в пространстве	169
Глава 8. ПОСТРОЕНИЕ ПРОЕКЦИЙ ПРОСТРАНСТВЕННЫХ ОБРАЗОВ	170
8.1. Построение в машинной графике простых аксонометрических проекций	173
8.2. Построение простых перспективных проекций	179
8.3. Специальные перспективные проекции	184
8.4. Контроль видимости элементов изображения	188
Глава 9. ПОВЕРХНОСТИ	194
9.1. Семейство линий в пространстве, образующее поверхность	195
9.2. Функция двух переменных, описывающая поверхность	198
9.3. Другие способы построения поверхностей и операции с ними	201
Глава 10. НЕМНОГО О ГРАФИЧЕСКОМ ДИАЛОГЕ И ПОЛУТОНОВЫХ ИЗОБРАЖЕНИЯХ	207
10.1. Графический диалог на микрокомпьютере	212
10.2. Полутоновые изображения	215
Заключение	219
Список литературы	220

ПРЕДИСЛОВИЕ

Десятилетие назад машинная графика и родственная ей машинная (вычислительная) геометрия считались частью системного программирования для ЭВМ или одним из разделов САПР (систем автоматизированного проектирования). Сейчас это уже вполне самостоятельные, со своими проблемами и спецификой области деятельности. Это и новые эффективные технические средства для проектировщиков, конструкторов и исследователей; это и программные системы и машинные языки; это и новые научные и учебные дисциплины, родившиеся на базе синтеза таких предшественников, как аналитическая, прикладная и начертательная геометрии, программирование для ЭВМ, методы вычислительной математики, приборостроение. Отдельные положения машинной графики и геометрии и более полные их курсы изучаются теперь во многих технических и строительных вузах. Скоро в том или ином виде с ними будут знакомиться, очевидно, и школьники. Машинная графика и геометрия представлены сейчас целым рядом направлений и разнообразными применениями. Некоторые из них отталкиваются от задачи автоматизации вычерчивания технической документации, другие – от проблем оперативного взаимодействия человека и машины, от задач численной обработки, расшифровки и передачи изображений и т.д. Свои подходы к машинной графике у геодезистов и картографов, у полиграфистов и астрономов, у специалистов по космической связи и конструкторов электронной аппаратуры, у кинематографистов и разработчиков "компьютерных игр". Разнообразие методов и приложений машинной геометрии и графики часто недооценивают даже специалисты. Во всяком случае, не все направления нашли должное отражение в известной нам литературе.

Конечно, у разных направлений и приложений машинной геометрии и графики есть некий общий фундамент и сходные черты, которые, наверное, и имеет смысл изучать в первую очередь. Понять сущность предмета, может

быть, удастся лучше, если в процессе знакомства с ним обратить внимание на последовательные этапы его развития.

При подготовке настоящей книги были сделаны попытки учесть эти требования, но, конечно, она не претендует на освещение всех направлений машинной графики. За основу взято, можно сказать, "традиционное" направление, сформировавшееся раньше других и ориентированное в основном на программирование вычерчивания изображений в виде последовательностей линий. Немного затрагивается и диалоговая графика. В качестве примера программной графической системы, со ссылками на которую ведется изложение, использована система Алграф, разработанная для ЭВМ серии ЕС и, частично, для ПЭВМ БК-0010. Эта система используется в учебном процессе ряда вузов и развивается под влиянием учебных задач; система также передается в некоторые производственные организации. Хотя система имеет определенные особенности, в книге в основном делается акцент на типичные для современной машинной графики и геометрии способы, приемы и геометрическую основу задач, так, чтобы пользователи или разработчики других систем и прикладных программ могли извлечь что-то полезное для себя. Все же основная цель книги — в не очень сухой и строгой форме дать практическое руководство для начального освоения машинной графики тем, кто имеет доступ к системе Алграф.

ВВЕДЕНИЕ

Одно из замечательных достижений человеческого гения в последние десятилетия – быстрое развитие электроники и вычислительной техники и создание на их основе сложных математических и информационных систем, обладающих зачатками специфического математизированного "мышления". Электроника и вычислительная техника используются в различных областях человеческой деятельности – от планирования и управления народным хозяйством до решения школьных задачек и каждодневных бытовых подсчетов. Если раньше ЭВМ были дорогими и сложными сооружениями, доступными только узкому кругу специалистов, – сейчас благодаря программируемым калькуляторам и персональным компьютерам с информатикой и программированием столкнутся сотни тысяч или даже миллионы людей.

Первоначально ЭВМ создавались для вычислений. Постепенно сфера применения вычислительных, математических методов расширилась и оказалось, что ЭВМ неплохо справляются с задачами несколько иного характера – с обработкой текстов, с хранением и поиском информации, с управлением станками, цехами и целыми производствами. Особенно интересно, когда техника привлекается к творческой, "интеллектуальной" деятельности – например, к проектированию новых машин, зданий, к работе над кинофильмами, музыкальными произведениями. Для новых разнообразных задач одних лишь электронных "мозгов" недостаточно, и вычислительные машины стали снабжать "глазами", "руками", а теперь даже и подобием человеческого голоса. "Глаза" нужны, в частности, чтобы машина могла воспринимать схемы, чертежи, фотоснимки, а иногда и натурные объекты. "Руки" – для наглядного изображения результатов расчетов, изготовления нужных людям чертежей и схем. Здесь речь идет о вводе в ЭВМ и выводе из нее графической информации, изображений.

Пословица "Лучше один раз увидеть, чем сто раз услышать" отражает исключительную для человека роль зрительного восприятия, визуальной и, в частности, графической информации. Емкость и быстрота восприятия зрительных образов весьма велики; для уяснения особенностей какого-либо

процесса иногда достаточно нескольких секунд, в течение которых мы рассматриваем график функции, диаграмму или другое наглядное изображение. На расшифровку же и сравнение тысяч чисел, которыми могут быть представлены те же объекты, понадобятся десятки минут и часы. Итак, машина должна наглядно представлять нам результаты расчетов, различные зависимости и процессы.

С помощью ЭВМ сейчас решаются многие задачи геометрического характера, в машине синтезируются простые и сложные геометрические образы — поверхности, тела, структуры. Математики задолго до появления вычислительных машин пользовались, с одной стороны, графической и геометрической интерпретацией различных математических понятий, а с другой стороны — численным описанием геометрических объектов и аналитическими методами решения геометрических задач. С распространением ЭВМ эта численная, аналитическая геометрия переживает, можно сказать, "вторую молодость". Машина наглядно изображает такие сложные геометрические объекты, которые раньше математики не брались изображать. Геометрия — одна из основных наук; в том или ином виде она связана и с физикой, и с химией, и с космонавтикой, и со многими другими областями человеческой деятельности. Без быстрого и точного решения геометрических задач невозможно успешное освоение космоса, конструирование сложных машин, строительство больших инженерных сооружений. Сфера применения электронной, машинной геометрии чрезвычайно широка!

Возьмем теперь такую разновидность человеческой деятельности, как подготовка и изготовление различных чертежей, схем и других изображений. Какой чертежник не мечтал, чтобы его кропотливую и нередко рутинную работу быстро и точно выполнял автомат? Известно, что сотни и тысячи чертежей, изготавливаемых в процессе машиностроительного и строительного проектирования, при подготовке производства, в значительной степени состоят из типовых, повторяющихся частей, изображений стандартных деталей и знаков. В иных случаях — при изготовлении точных карт, при подготовке рисунков для их последующего типографского воспроизведения — точности человеческой руки не всегда бывает достаточно несмотря на наличие линеек, циркулей и других средств "малой механизации". Автоматизация чертежно-графических работ, таким образом, может рассматриваться как одна из насущных задач, эффективное решение которой возможно с помощью вычислительной техники.

Чертежи и схемы — только частный вид графической информации. В технике, научных исследованиях, в области связи активно используются фотоснимки и другие полутонные и цветные изображения. Если такие изображения перевести в числовую, математическую форму, открываются возможности различной их трансформации, контроля качества и исправления изъянов методами вычислительной математики, высококачественной передачи таких изображений по каналам связи. Автоматизированная обработка изображений — так иногда называют эту проблему.

Решение подобных задач — это лишь часть общего дела. Большого эффекта можно добиться при комплексной автоматизации различных этапов общего процесса, протекающего на производстве, в проектной организации или научно-исследовательском учреждении. Наиболее интересен и показателен в этом отношении процесс проектирования машины или объекта строительства. С одной стороны, он, как правило, многогранен, включает в себя решение разнообразных задач, в том числе геометрических и графических.

С другой стороны, наряду с этапами технического характера, выполняемыми по заранее разработанным правилам и алгоритмам, здесь есть этапы безусловно творческие, где ведущую роль играют интеллект, изобретательность человека. Может ли здесь использоваться вычислительная техника и в какой роли? Как привлечь машину к решению творческих задач? В настоящее время большинство исследователей считают, что надо стремиться к оптимальному сочетанию возможностей человека и машины. Современная ЭВМ — автомат, она быстро и точно выполняет то, что в нее заложено человеком, успешно решает типовые задачи. Это мощный инструмент и для технической, и для творческой работы. Ценность же инструмента во многом зависит от удобства его использования. Как добиться наиболее удобного, наглядного и быстрого общения человека и машины? Раньше использование ЭВМ существенно сдерживалось из-за сложности и специфичности программирования и управления машиной. Переходя к общению с ЭВМ на уровне графических изображений — схем, фигур, графиков, чертежей, можно значительно повысить эффективность использования вычислительной техники.

Инженер, проектировщик, архитектор работают, имея перед глазами графическую или объемную модель создаваемого объекта. Чертеж называют языком техники. Поэтому понятен тот интерес к машинной графике, который сейчас наблюдается во многих странах, и активные разработки в этой области. Проблем здесь много. Требуются специальное оборудование и новые приемы работы (например, использование "светового пера"). Конечно, графический диалог нужен не только для автоматизированного проектирования, но здесь он наиболее естественно вытекает из сути работы.

Для эффективного и качественного проектирования объектов разного рода создаются *системы автоматизированного проектирования (САПР)*. Они включают средства ввода, вывода и обработки графической информации. Иногда необходимо замерять объемные объекты — модели, макеты, натурные образцы, и информацию об их форме вводить в ЭВМ, а также в автоматическом режиме воспроизводить пространственные геометрические формы. Системы, позволяющие это делать, можно назвать *системами объемного геометрического отображения*.

Когда объект спроектирован, его надо... изготовить! Однако обычно речь идет не о единичном образце машины или изделия, а о выпуске сотен, тысяч изделий. Для этого необходимо наладить новое производство, подо-

брать станки, изготовить технологическую оснастку, подготовить программы для роботов и станков с числовым программным управлением, т.е. после проектирования самого изделия надо проектировать технологию его изготовления. Для этого разрабатывают и внедряют автоматизированные системы технологической подготовки производства (АСТПП). Здесь тоже без графики и чертежей не обойтись. Простой пример – подготовка программы для станка, который должен фрезеровать штамп. Крупный штамп обрабатывается не один день. А вдруг в программе допущена ошибка? Проверить программу можно с помощью графического изображения, на котором условными линиями будут показаны последовательные движения инструмента. Работу же самого станка, автоматически изготовляющего сложную деталь (штамп, пресс-форму), можно рассматривать как вывод из ЭВМ объемной информации.

В различных областях науки и техники проектирование и конструирование имеют свою специфику, которая накладывает определенные требования и на системы автоматизированного проектирования. Сравним такие области, как картографию, проектирование электронных схем, проектирование кораблей, производство мультфильмов. Ясно, что здесь и объекты проектирования, и конструкторская документация, и машинные изображения значительно различаются. Для архитектуры, прикладного искусства, технической эстетики существенно, что учитывают не только законы экономики, технологии и технической целесообразности, но и законы красоты. Поэтому автоматизированные системы должны обеспечить достоверное и наглядное представление проектируемых объектов, легкое и удобное изменение ракурса, под которым отображается объект, возможность быстрого изменения формы, пропорций объекта в желаемом для художника (архитектора) направлении. Порой здесь желательны цветные тоновые изображения, объемные модели и макеты.

Разнообразие графической информации и областей человеческой деятельности, где она используется, приводит к выводу, что *машинная геометрия и графика* – явления разнообразные и многоплановые. Это и новые области человеческой деятельности, и новые научные дисциплины, и оборудование, и работающие в компьютерах программы (рис. 1). Основными задачами машинной графики являются ввод в ЭВМ (считывание) графической информации и вывод ее из ЭВМ (формирование изображений), а также определенного рода переработка информации в компьютере. Основные задачи машинной геометрии или, как говорят, автоматизированного геометрического моделирования и конструирования – синтез в ЭВМ и анализ геометрических объектов, решение задач геометрического характера. Машинная графика и геометрия базируются на таких дисциплинах, как различные разделы геометрии (аналитическая, начертательная, дифференциальная, проективная, прикладная), программирование для ЭВМ, элементы вычислительной математики, понятия о формальных языках, применимых в общении человека и машины.

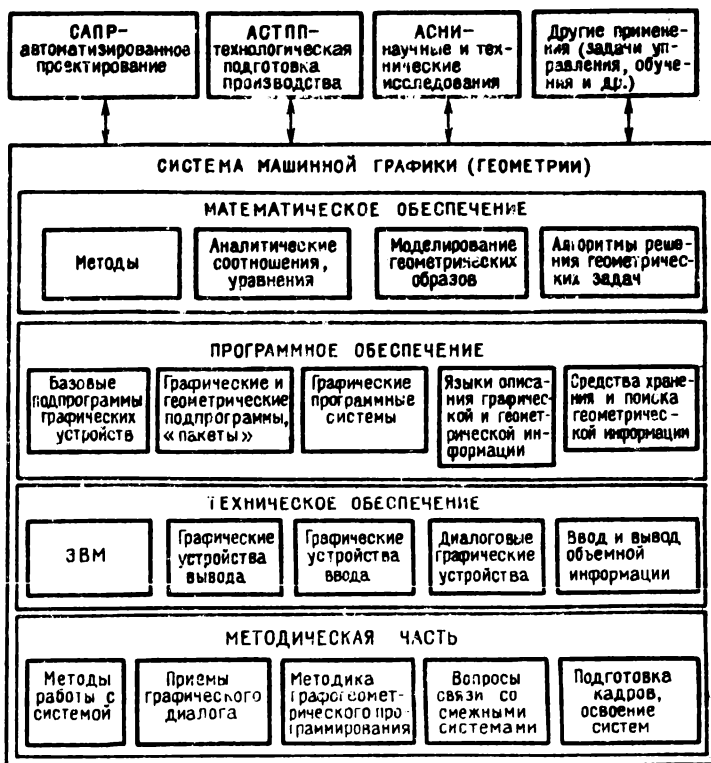


Рис. 1. Разделы и составные части системы машинной графики

Как программно-математические системы, "живущие и работающие" в электронной среде, машинная графика и геометрия складываются из множества программ и их частей (подпрограмм, моделей), хранящихся в библиотеках на магнитных дисках или лентах и вызываемых в нужные моменты, в том числе "под управлением" других программ -- проектных, конструкторских, ориентированных на решение конкретных задач. Так, системы автоматизированного проектирования состоят из многих частей -- подсистем, нацеленных на решение задач разного плана -- конструктивных, экономических, технологических и т.п. Машинная графика и геометрия -- важные, существенные подсистемы в системах автоматизированного проектирования. В конкретных случаях графика и геометрия могут быть объединены в одну программную систему (подсистему).

Разнообразие требований к машинной графике и геометрии и самих графогеометрических систем иногда недооценивается программистами, инженерами-электронщиками. Полагают, что почти все задачи можно ре-

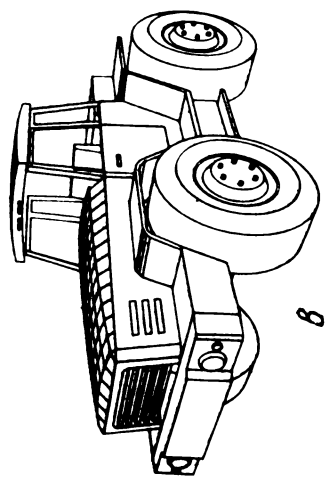
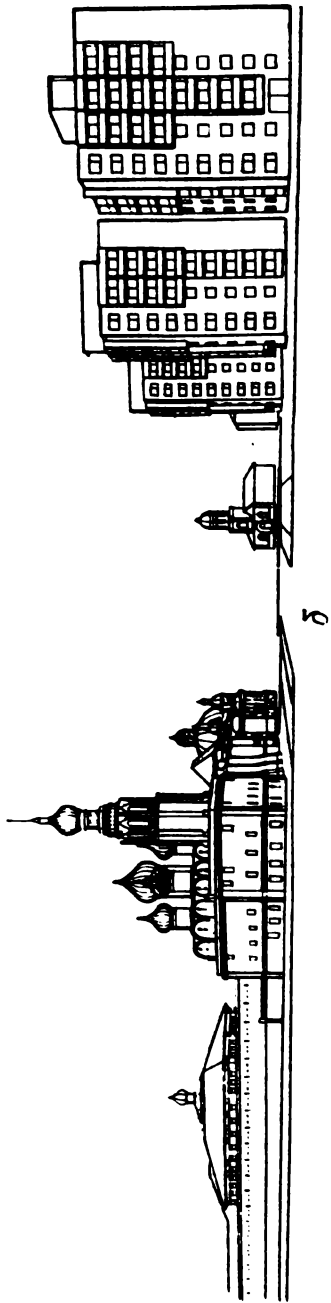


Рис. 2. Примеры машинных изображений: *а* – чертеж типовой машиностроительной детали, *б* – перспектива архитектурных объектов, *в* – изображение машины в процессе художественного конструирования

шать с помощью одной, мощной и совершенной системы (серийно выпускают далеко не все нужные типы графических устройств и т.д.). С другой стороны, в различных графических и геометрических системах много общего. Почти все чертежи и схемы содержат прямые и кривые линии окружности, надписи из букв и цифр. Общей основой почти для всех применений машинной геометрии и графики являются двумерные и трехмерные координатные системы и их преобразования. Подобный геометрический фундамент, общие уравнения, алгоритмы, методы решения задач составляют методическое и математическое "ядро", на которое дальше уже наслаиваются специфические и иногда оригинальные алгоритмы и программы. Таким образом, сама графическая или геометрическая программная система неоднородна, обычно состоит из ряда "слоев" или иерархических уровней. На нижнем уровне — так называемое *базовое обеспечение* используемых в системе графических устройств, осуществляющее передачу информации этим устройствам или прием информации от них, управляющее работой этих устройств. Ясно, что такие программы тесно связаны с конструкцией и возможностями графических устройств, но почти не зависят от области применения машинной графики, характеристик проектируемых объектов. Более сложные программы (первый или второй уровни) уже не зависят от графических устройств или зависят от них лишь частично. Если используемый набор таких программ уже определяется областью применения (например, зависит от того, нужна ли в данной области "трехмерная" машинная графика, описывающая пространственные объекты), то все же программы сохраняют свою универсальность, общий характер. "Высшие уровни" обычно представлены специализированными программами, согласованными с особенностями решаемых задач, с характеристиками проектируемых объектов. Так, в картографии это программы обработки цифровых моделей местности, построения линий-горизонталей, а в системе проектирования автомобильных кузовов — программы синтеза эстетически совершенных поверхностей, спроектированных художниками, и программы вывода управляющей информации о них для станков с ЧПУ.

Если не считать новых моделей малых персональных компьютеров и специализированных "графических ЭВМ", для решения графических и геометрических задач используются универсальные цифровые машины, в основной комплект которых не входят графические устройства, способные воспроизводить изображения или считывать их. Графические устройства поставляются отдельно или могут быть в комплекте так называемых автоматизированных рабочих мест (АРМ), содержащих ЭВМ и наборы графических устройств. Таким образом, существенной частью графической системы можно считать техническое обеспечение — ЭВМ и графические устройства. Из рис. 1 видно, что другой частью общей системы машинной графики является методическая. Сюда относятся кадровые вопросы, обучение специалистов, разработка эффективных методов и приемов работы с системой и др.

На рис. 2 приведены примеры машинных изображений из разных областей.

1.1. Виды изображений

Любой школьник отличит фотоснимок от карандашного рисунка, а рисунок – от точного чертежа. Нам надо разделить изображения на типы или классы в зависимости не от их художественных достоинств или тематики, а от того, какими техническими средствами их можно воспроизвести или “прочитать”.

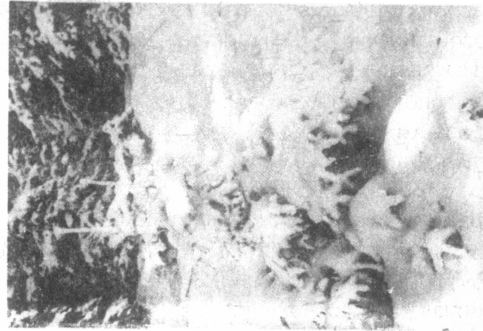
Самые простые типы изображений – штриховые, или линейчатые. Они состоят из отдельных линий – отрезков, дуг, которые можно последовательно наносить на бумагу или другой материал каким-либо инструментом. Электронный луч на экране электронно-лучевой трубки также может высвечивать линии изображения последовательно, хотя этот процесс и будет повторяться много раз в секунду, чтобы изображение оставалось видимым. Назовем этот способ получения изображений *траекторным*.

Из штриховых изображений наиболее просты изображения с одной толщиной штриха, с одной яркостью линии (рис. 3,а); несколько сложнее изображения с несколькими толщинами (яркостями) линий (рис. 3,б). Этот тип изображений – основной для различных чертежей, графиков и схем, и в первую очередь на него мы будем ориентироваться в дальнейшем изложении. На гравюрах, в рисунках пером мы встречаем штрихи с переменной толщиной. Можно сказать, что это тоже отдельный “подвид” изображений. Пока, правда, автоматически получаемых изображений с такими линиями мы не встречали. На рис. 3,в условно показаны изображения, на которых не только линии имеют разную толщину, но и предполагается разный цвет.

В среднем ряду рис. 3,г, д показаны примеры “аппликационных” изображений – изображений, составленных из областей – “пятен” разных тонов или цветов. В простейшем случае на белом фоне – черные области разной формы. К изображениям этого типа можно отнести обложки и титульные листы книг с крупными надписями, некоторые плакаты, эмблемы, декоративные рисунки на тканях и обоях, кадры некоторых мультфильмов. Теоретически “аппликационные” изображения можно воспроизвести тонким пером путем частой штриховки, когда последовательно нанесенные



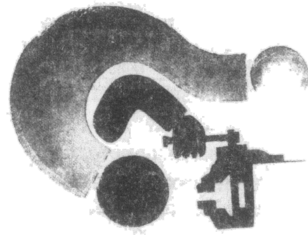
в



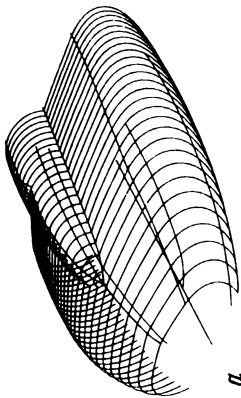
ж



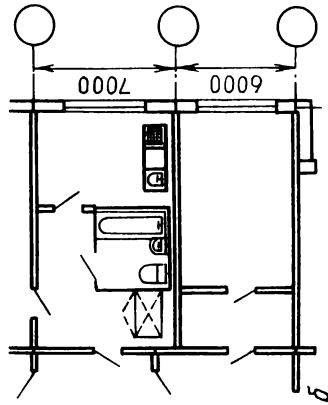
з



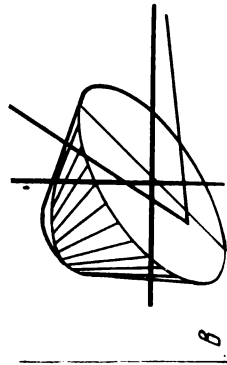
и



а



б



в

Рис. 3. Разновидности изображений

штрихи сливаются друг с другом. Но это трудоемкий и неэффективный способ, поэтому применяют специальные графические устройства.

Наконец, наиболее сложные для изготовления и чтения изображения — полутоновые, в том числе цветные (рис. 3, е, ж). Здесь каждая малая точка изображения может иметь свой тон или же — и тон, и цвет. От точки к точке тон и цвет могут изменяться постепенно и, где нужно, резко. К таким изображениям принадлежат фотоснимки и кадры большинства кинофильмов, некоторые иллюстрации в книгах и журналах, картины и акварельные рисунки. Одним тонким пером воспроизвести такое изображение практически невозможно. Мы знаем, однако, что техника, в том числе электронная техника телевидения, неплохо справляется с передачей и воспроизведением таких изображений. Можно надеяться, что с помощью ЭВМ и подключенных к ней устройств такие изображения можно будет получать и считывать. Обычно для этого используется *растровый способ* воспроизведения или считывания. Если рассмотреть через увеличительное стекло иллюстрацию в газете или журнале, как правило, будет виден *растр* — мелкие точки, пятнышки, полоски разной величины и цвета. На рис. 4 показана увеличенная часть газетной иллюстрации. Так и для моделирования изображения в ЭВМ оно раскладывается на множество точек, пятнышек, расположенных рядами и столбцами, — в своего рода "мозайку". В ЭВМ тон

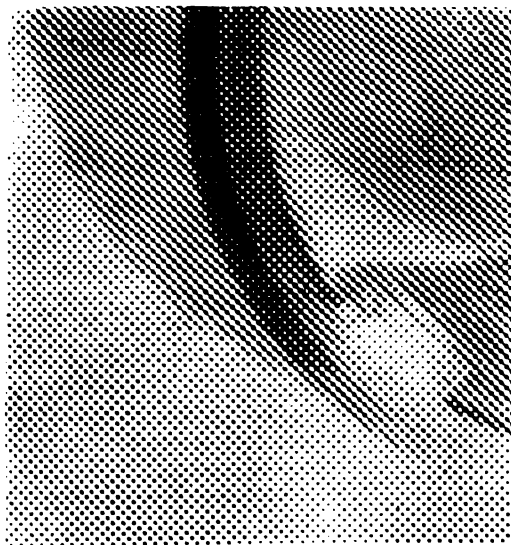


Рис. 4. Часть газетной иллюстрации в увеличенном виде

(яркость) или цвет каждой такой точки можно описать числом или тремя числами (для цветного изображения). Множество таких чисел образуют математическую модель растрового изображения. Технические устройства должны воспроизводить изображения, нанося множество точек разной интенсивности или цвета.

Здесь приведена классификация изображений с точки зрения "вида" самого изображения. Но есть и другие критерии классификации, например:

– изображения могут быть статическими (неподвижными) и динамическими, т.е. изменяющимися во времени (как в кино и телевидении);

– изображения могут существовать на постоянном носителе (бумаге, пленке и т.д.) или на каком-либо экране, и тогда они исчезают при выключении воспроизводящего их аппарата;

– изображения могут быть плоскими или стереоскопическими, дающими эффект объемности, и т.д.

Таким образом, классифицировать изображения можно по различным критериям (попробуйте это сделать для таких изображений, как телевизионное, инженерный чертеж, стереоскопический диапозитив и др.). Из этого можно заключить, что и графические устройства, работающие совместно с вычислительными машинами, должны быть разными. Действительно, появляются все более эффективные устройства новых марок и конструкций, например лазерные или струйные. Все же пока в области прикладной математики и автоматизированного проектирования основным типом изображений являются штриховые, получаемые на бумаге путем автоматического черчения или видимые на экране электронно-лучевых устройств.

1.2. Предшественники графопостроителей и чертежных машин. Самописцы

Идея механического или автоматического изготовления изображений уходит в глубь веков. С одной стороны, можно проследить, как совершенствовались чертежные инструменты – циркули, линейки, перья, карандаши. Появлялись новые приборы, например пантографы, с помощью которых изображения можно было перерисовывать в измененном масштабе (рис. 5), эллипсографы для черчения эллипсов и их дуг (рис. 6), чертежные приборы типа "кульман", различные трафареты и т.д. В картографии, где требуется особо высокая точность нанесения деталей и измерения, пришли к механическим координатографам – чертежным столам, снабженным направляющими с инструментом и зубчатыми рейками или винтовыми передачами с рукоятками и циферблатами, с помощью которых можно точно устанавливать инструмент. Такой координатограф можно считать прототипом механической части современной чертежной машины. Иногда такие машины и называют *автоматическими координатографами*.

В истории техники существовали приборы, где инструмент "сам" двигался, нанося изображение. В 16, 17 веках самым известным "автоматом"

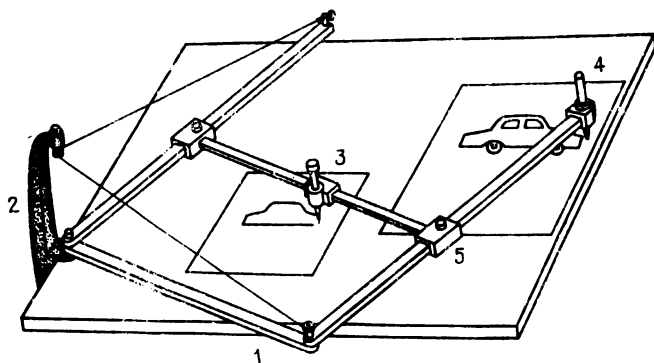


Рис. 5. Схема пантографа: 1 – рычаги, 2 – кронштейн, 3, 4 – карандаш и щуп (можно взаимно менять местами), 5 – шарниры на перемещаемых ползунах

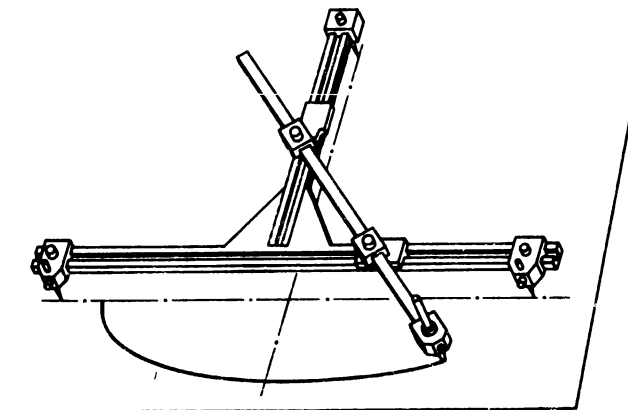


Рис. 6. Одна из конструкций эллипсографа (с перпендикулярными направляющими)

были часы; на базе часовых механизмов умельцы-изобретатели создавали различные движущиеся приборы, игрушки. В 1774 году швейцарские часовщики Пьер и Анри Дро показали на выставке в Париже трех механических кукол – рисовальщика, писца и музыкантшу. Фигуры этих сидящих людей были одеты в настоящую одежду, при работе двигали головой, ”дышали”. ”Писец” время от времени макал гусиное перо в чернильницу и на подложенном листе бумаги выписывал слова (рис. 7); в заключение он посыпал лист бумаги песком для просушивания чернил. Другая кукла рисовала фигурки, портреты королей. Конечно, устройство кукол было чисто механи-

ческим, привод — от пружины, но интересно, что в "писце" имелось устройство, которое сейчас назвали бы генератором символов, — кулачковый механизм, обеспечивающий вычерчивание различных букв. Эта кукла была "программируемой", так как мастер мог, устанавливая специальные зубцы на диске, довольно быстро программировать вычерчивание новых надписей. Эти куклы вспоминаются, когда на современных выставках мы



а



б

*A sa ville natale
Jaquet Droz*

Рис. 7. Механический писец, изготовленный братьями Дро (а), и образцы его письма и рисунка, сделанного механическим рисовальщиком (б)

видим роботов в виде механических (но уже с электроникой) людей, играющих на музыкальных инструментах, рисующих, перекладывающих предметы.

В 19 — 20 веках появились различные самопишущие приборы — термографы, барографы, компасы, регистрирующие в отсутствие человека измеряемые величины в виде графиков. Сначала они были механическими, с пружинным заводом (рис. 8) и перематывающейся бумажной лентой или поворачивающимся бумажным диском, на которых колеблющееся острие пера оставляло линию. Позже такие приборы стали снабжаться электроприводом с усилителями, электроникой и получили способность регистрировать очень быстрые процессы и сигналы малой величины.

От таких самописцев — только шаг до двухкоординатных построителей графиков.

Такой прибор может иметь плоский планшетик, на который накладывается лист бумаги (рис. 9). Перо установлено на поперечной каретке и с помощью тросиков может перемещаться в двух направлениях (движение каретки соответствует горизонтальной оси x , движение пера по направляющим каретки — оси y). Перемещение осуществляется двумя

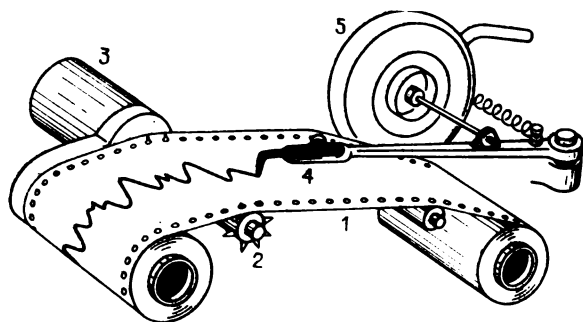


Рис. 8. Схема самопишущего прибора (барографа): 1 – перфорированная бумажная лента, 2 – ведущий барабан, 3 – привод, 4 – трубчатое перо на качающемся рычаге, 5 – датчик давления (или другой величины)

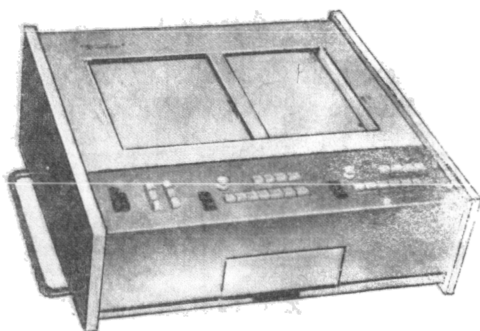


Рис. 9. Двухкоординатный самописец – вольтметр

быстродействующими двигателями, вращающими через зубчатую передачу барабанчики с тросами. Рядом с двигателями установлены потенциометры-сопротивления, посылающие электронной схеме сигнал обратной связи, величина которого зависит от положения пера (см. схему на рис. 10). На вход прибора подаются два напряжения, регистрируемые прибором. Величины этих напряжений схема все время сравнивает с напряжениями от потенциометров обратной связи. Если обнаруживается рассогласование, на соответствующий двигатель через усилитель подается ток. Под действием этого тока двигатель перемещает перо так, чтобы равновесие было восстановлено. Перо можно включать и выключать (приподнимать его острие над бумагой) с помощью дополнительного электромагнита.

Такой самописец может регистрировать не только взаимозависимость двух подаваемых от каких-то других приборов напряжений. Если его надлежащим образом подключить к ЭВМ, которая по программе посылала бы ему изменяющиеся нужным образом напряжения – получится прибор, подобный графопостроителю. Автор книги в 1972 году работал с таким построителем производства ГДР "Endim-200" и вычислительной маши-

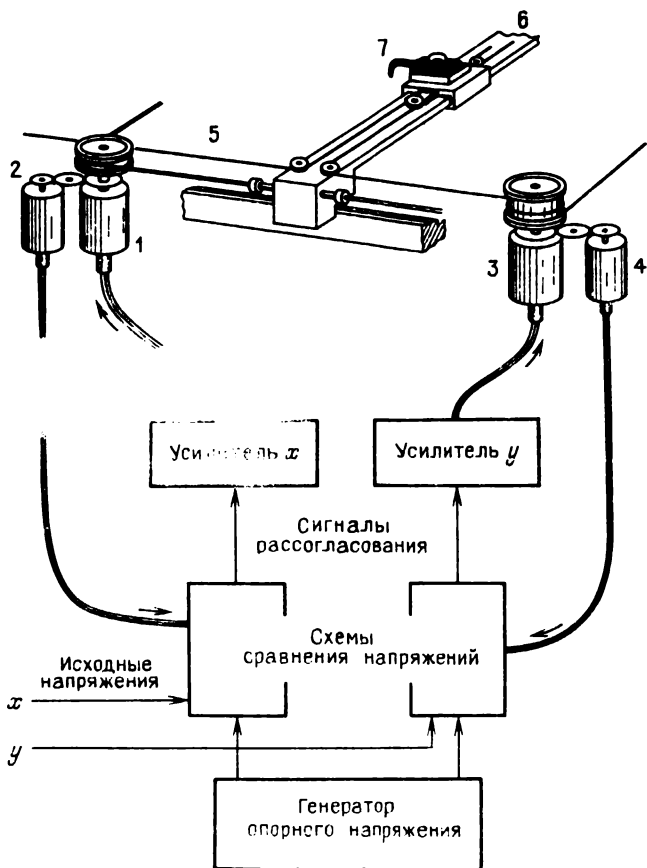


Рис. 10. Схема устройства аналогового самописца: 1 – двигатель для координаты x , 2 – датчик положения для координаты x , 3 – двигатель для координаты y , 4 – датчик положения для координаты y , 5 – приводные тросы, 6 – каретка, 7 – перо на рабочей головке

ной "Днепр". В наши дни отечественными строителями такого типа комплектуется малая машина "Искра-226". Но...

"Но" заключается в том, что регистрация положения пера и его перемещение осуществляются с помощью *аналоговых*, т.е. непрерывных сигналов (напряжений). Поэтому точность работы прибора не может быть очень большой, а для обычного черчения требуется весьма высокая относительная точность. При длине чертежа в метр линии должны стыковаться с точностью 0,1 мм. Это значит – относительная погрешность должна быть 1/10000. Аналоговая схема такой точности дать не может.

1.3. Современный графопостроитель

Графопостроители, или автоматические чертежные машины, — один из основных типов графических устройств, обеспечивающих выполнение штриховых изображений на бумаге, кальке, пленке и других материалах. Часто построители имеют цифровое управление, что позволяет получать очень точные изображения и облегчает стыковку с цифровыми ЭВМ. Построители выпускаются самых разных размеров — от миниатюрных настольных до больших чертежных установок, на которых в натуральную величину чертят кузова автомобилей и детали самолетов. Большинство построителей имеют плоский планшет, к которому крепится бумага или другой материал, но есть и модели, чертящие на бумажной ленте, сматываемой с рулона.

Конструкцию и принцип действия графопостроителя полезно знать и программисту, и математику, и конструктору, работающему с автоматизированными системами, — такие математические понятия, как координаты, интерполяция, округление чисел и другие здесь реализуются с помощью электронных схем и механики.

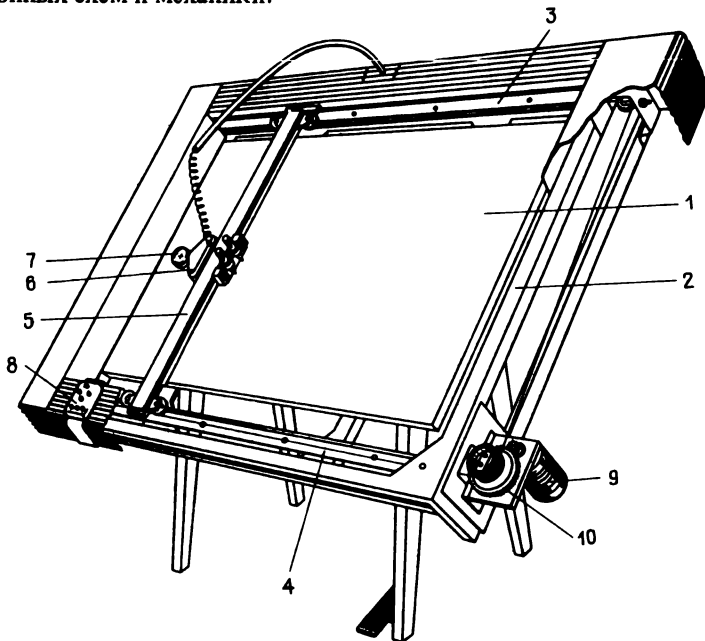


Рис. 11. Конструкция электромеханического графического устройства: 1 — планшет для крепления бумаги (стол), 2 — усиливающая рама стола, 3 — верхняя продольная направляющая, 4 — нижняя продольная направляющая, 5 — каретка с поперечными направляющими, 6 — чертежная головка с электромагнитами и перьями, 7 — дополнительное устройство с визиром для графического ввода, 8 — автономный пульт управления, 9 — один из шаговых двигателей, 10 — зубчатый редуктор с трюсовым барабаном

На рис. 11 изображена схема типового планшетного построителя с так называемыми шаговыми двигателями и тросовым приводом. Сверху и снизу планшета расположены продольные направляющие, как бы овеещающие в металле координатную ось x . По ним на роликах перемещается каретка-мостик с поперечными направляющими. Движение рабочей головки с перьями по этим направляющим соответствует координате y . Ясно, что совместное движение каретки и головки по каретке дает возможность прочертить на бумаге линию любого направления. Черчение производится шариковыми стержнями, трубчатыми перьями типа "рапидограф", фломастерами, а иногда и специальными резцами. Инструмент включается и выключается с помощью электромагнитных приспособлений. Головка содержит, как правило, 3–4 гнезда для инструментов, куда можно установить перья, дающие линии разной толщины или цвета. Чтобы бумага при черчении не сдвигалась и на ней не было "пузырей", она прижимается к планшету вакуумом (для этого в планшете делаются сотни маленьких отверстий, через которых компрессором откачивается воздух) или с помощью электростатического поля (в этом случае в планшет вмонтирован электропроводный слой, на который подается высокое напряжение).

Как показано на схеме, привод каретки и головки с перьями осуществляется с помощью системы тросиков, натянутых так, чтобы один трос перемещал перья только в направлении координаты x , а другой – только в направлении координаты y . Тросы намотаны на два барабана, которые вращаются от двух электромоторов. Эти двигатели – не обычные. У обычного двигателя неизвестно, сколько оборотов сделала его ось, и он не может повернуть свою ось на заданный угол, меньший одного оборота. Как в этом случае обеспечить точное перемещение пера? Шаговый же двигатель поворачивает ось небольшими "шагами". Один шаг может соответствовать части полного оборота, например $1/16$ или $1/240$ части! На рис. 12 показано устройство такого двигателя. Двигатель имеет зубчатый ротор и зубчатые наконечники трех (или больше) пар полюсов статора. Когда в одну пару обмоток подан ток, ближайшие зубцы ротора притягиваются к зубцам соответствующих полюсов статора. Зубцы других наконечников оказываются смещенными относительно зубцов ротора на треть шага в одну и в другую сторону. Если в первой паре обмоток ток выключить, а в другой включить, ротор повернется на *элементарный шаг* – треть расстояния между зубцами. В усовершенствованных конструкциях шаг может быть еще меньше, так как можно одновременно подавать ток в две пары обмоток, постепенно уменьшая ток в одной паре и увеличивая в другой. Есть шаговые двигатели несколько иной конструкции, где зубцы ротора намагничены. (Школьники или студенты вполне могут сделать в своем кружке модель графопостроителя, используя вместо шаговых двигателей даже храповики с электромагнитами.) Современные графопостроители при скорости черчения порядка 1 м/с и величине элементарного шага $0,01$ мм за секунду должны выполнять сто тысяч шагов.

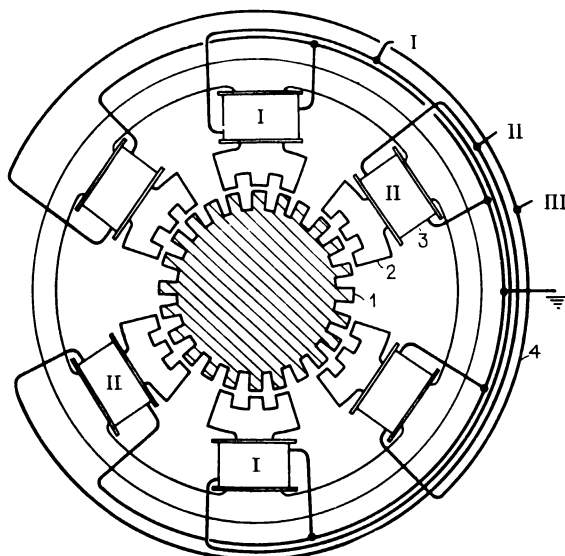


Рис. 12. Схема типового шагового двигателя: 1 – зубчатый ротор, 2 – три пары наконечников статора, 3 – обмотки, 4 – условно показанные проводники; I, II, III – пары обмоток

Если разграфить мысленно поле чертежа на клетки с частотой, соответствующей элементарному шагу (на пере), окажется, что, управляя двигателями, можно поставить перо в один из узлов этой сетки. "За один раз", переключив обмотки у одного или сразу у двух двигателей, перо можно переместить в один из восьми соседних узлов (рис. 13). Не правда ли, перо "ходит" подобно шахматному королю? Из таких отдельных ходов можно набрать траекторию, близкую к прямой или какой-то кривой линии. Программист-пользователь, конечно, не думает об отдельных ходах. Эти ходы вычисляет стандартная программа в ЭВМ (тогда говорят, что построителем она управляет в "инкрементальном" режиме) или электронная

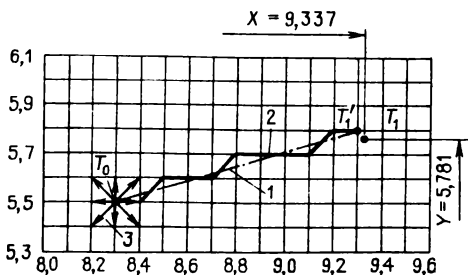


Рис. 13. Схема линейной интерполяции: 1 – теоретическая траектория, 2 – траектория, набранная из элементарных шагов, 3 – возможные элементарные шаги пера от точки T_0

схема самого устройства. Процесс этот называют *интерполяцией*. Если из последовательных ходов набирается траектория, близкая к отрезку прямой – говорят о *линейной интерполяции*. Если таким образом может аппроксимироваться дуга окружности – интерполяция *круговая*. Большинство современных строителей оборудовано линейными интерполяторами, т.е. при задании опорных точек чертежа происходит соединение их отрезками; некоторые строители имеют линейно-круговые интерполяторы и могут соответственно чертить и отрезки, и дуги окружностей (как задается дуга, будет сказано ниже).

Попробуем построить алгоритм линейной интерполяции. Его блок-схема показана на рис. 14. Здесь надо дать краткие пояснения. Кружками или овалами помечаются начальные, конечные или промежуточные "пункты" алгоритмов. В прямоугольниках записываются различные вычисления, арифметические выражения. Вызовы подпрограмм, обращения к отдельно описанным блокам алгоритмов записываются в двойных прямоугольниках, ромбы используются для проверки выполнения условий (условный переход). Условия, которые соответствуют тому или иному маршруту, помечаются около отходящих от ромба линий словами "да" или "нет", неравенствами " > 0 ", " ≤ 0 " и др. В параллелограммах с наклонными боковыми сторонами указываются операции вывода информации на внешние устройства или чтения с них. Арифметические и логические действия, уравнения в блок-схемах записывают иногда в алгебраическом виде, а иногда в виде, приближенном к языкам программирования. Для многих программистов-профессионалов язык программирования стал привычнее алгебраической записи. Дальше часто будут использоваться записи, по форме приближенные к алгоритмическому языку фортран. Несколько букв или букв и цифр подряд будут обозначать одну величину, название переменной, подпрограммы и др. Арифметические действия будут обозначаться соответствующими знаками (умножение – звездочками). Аргументы функций (в том числе тригонометрических) заключаются в скобки. Круглые скобки можно "вкладывать" друг в друга. Важно также, что в программировании и описаниях алгоритмов знак равенства означает операцию присваивания, т.е. указание, что величине, записанной слева от равенства, присваивается значение, соответствующее правой части. Запись $X = X + 1$ означает, что значение переменной X надо увеличить на единицу. В некоторых алгоритмических языках вместо знака равенства используются два подряд поставленных знака ":=".

Из блок-схемы алгоритма видно, что сначала вычисляются приращения координат для движения от отработанной (где в данный момент находится перо) точки к следующей заданной; в зависимости от того, какое из приращений по абсолютной величине больше, в работу вступает одна из двух "параллельных" ветвей алгоритма. В каждой ветви в цикле значение ведущей координаты изменяется каждый раз на величину элементарного шага (взятого с нужным знаком). Вычисленное (теоретическое) значение вто-

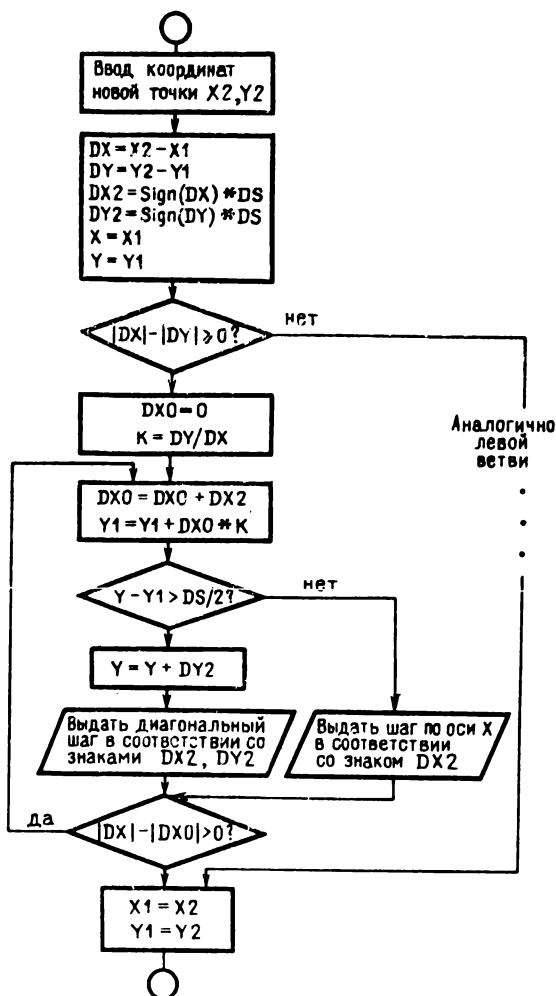


Рис. 14. Блок-схема алгоритма линейной интерполяции. При входе в алгоритм величины X_1, Y_1 содержат координаты предыдущей точки, DS – шаг элементарного перемещения

рой координаты сравнивается с ее значением, соответствующим положению пера на построителе. Если эти значения расходятся на величину, большую, чем полшага, на построитель дается команда выполнить "диагональный" шаг, иначе выполняется шаг в направлении ведущей координаты. Алгоритм можно составить более рационально, но здесь он приведен для наглядности. На базе подобного алгоритма может работать как программа в ЭВМ, так и электронная схема интерполятора.

Для вычерчивания дуги окружности можно было бы принять за основу ее уравнение. Но для реализации в виде электронной схемы этот путь оказывается слишком сложным. Придумали другой способ. Если дуга начинается от той точки, где перо уже стоит, построителю передаются координаты еще двух точек: конца дуги и ее центра. Иногда координаты заменяются их приращениями относительно отработанной точки. Кроме того, тем или иным способом задается направление обхода дуги (по или против часовой стрелки). Дальнейшие действия выполняются с помощью так называемой оценочной функции. В упрощенной постановке вопроса оценочная функция связана с радиусом дуги, или с величиной, равной сумме квадратов приращений координат от текущей точки до центра дуги (с квадратом радиуса).

Эта величина может быть подсчитана для начальной точки дуги; затем в зависимости от заданного направления обхода и квадранта, в котором находится точка, для элементарного шага выбираются знаки возможных перемещений по осям x и y . Возможные при этом варианты элементарных ходов сравниваются на предмет лучшего соответствия оценочной функции и выбирается нужный ход (рис. 15). Каждый раз проверяется удаленность текущей точки от заданной конечной точки дуги, чтобы в нужный момент завершить вычерчивание дуги. Задание дуги этим способом, вообще говоря, *переопределенное*. Конечная точка может быть задана так, что от центра дуги она будет находиться не на том же расстоянии, что начальная. В этом случае программа или электронная схема может "зациклиться" и без конца чертить одну и ту же окружность.

Кроме линейного или линейно-кругового интерполятора и усилителей, питающих обмотки двигателей, блок управления графопостроителем содержит цепи для приема информации от ЭВМ или для считывания ее с магнитной ленты, перфоленты (рис. 16). Конечно, построитель, даже быстродействующий, работает значительно медленнее современной ЭВМ, и поэтому, если ЭВМ решает только одну задачу, непосредственное подключение построителя к ней невыгодно: процессор машины оказывается недогруженным. Выгоднее быстро записать информацию о чертеже на промежуточный носитель, а затем выдавать чертежи. Если же ЭВМ работает в режиме мультипрограммирования, нередко оказывается проще и целесообразней управлять построителем по кабелю (каналу).

Блок управления построителем может содержать ряд дополнительных частей: генератор символов (дает возможность вычерчивать буквы, цифры и знаки стандартного (обычно упрощенного) начертания), генератор штриховых и штрихпунктирных линий (позволяет чертить такие линии, не вычисляя в машине конечных точек каждого штриха). Буферная память повышает эффективность использования построителя, особенно когда он непосредственно соединен с ЭВМ: ведь после передачи крупной порции информации машина на больший промежуток времени может переключиться на решение другой задачи.

Рис. 15. Схема круговой интерполяции

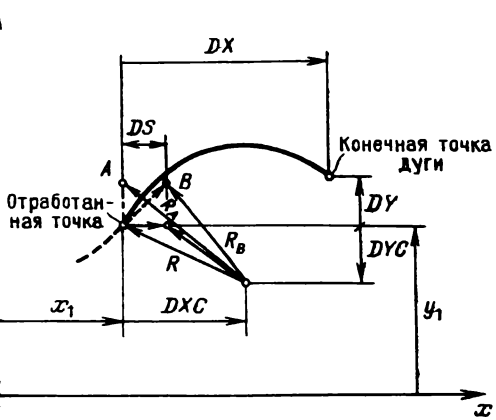
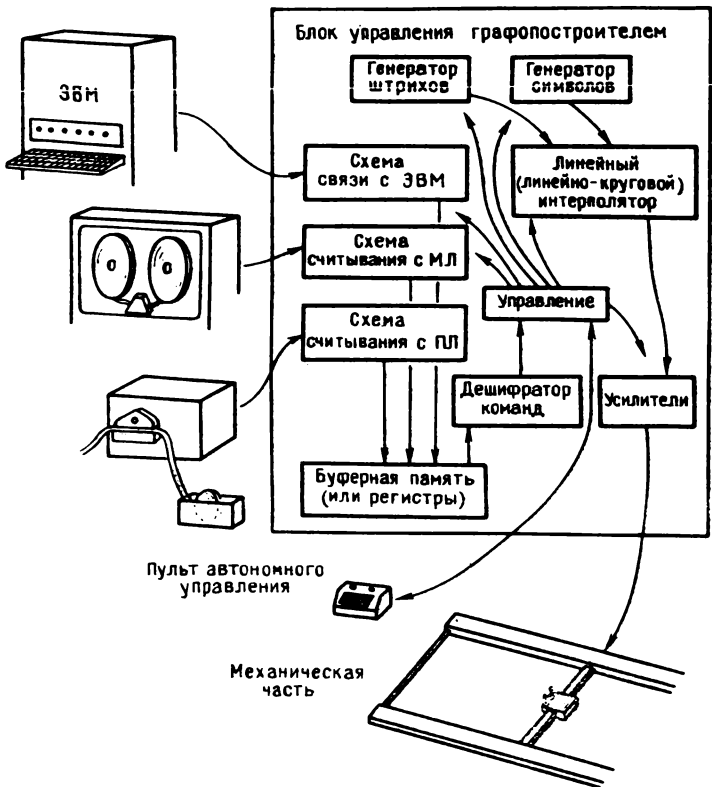


Рис. 16. Основные части блока управления графопостроителем



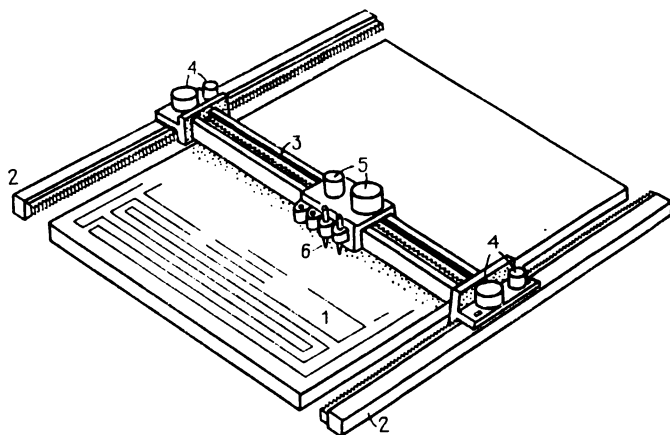


Рис. 17. Упрощенная схема привода графопостроителя: 1 – планшет с фигурным металлизированным слоем под стеклянной поверхностью для создания электростатического поля, 2 – продольные направляющие с зубчатыми рейками привода, 3 – каретка с поперечными направляющими и зубчатой рейкой, 4 – электродвигатели и датчики положения для координаты x , 5 – электродвигатель и датчик положения для координаты y , 6 – перо в электромагнитных устройствах

Сложные блоки управления позволяют изменять масштаб изображения, чертить его с поворотом, регулировать скорость черчения и т.д. В новых моделях в качестве блоков управления иногда используются программируемые процессоры или миниЭВМ.

В качестве примера типовой конструкции графопостроителя с шаговыми двигателями можно привести отечественные построители АП-7251 и ЕС-7051. Скорость черчения у них – до 50 мм/с, элементарный шаг – 0,05 мм и 0,025 мм, размер листа бумаги может достигать 1200 X 1500 мм. Одновременно могут устанавливаться три трубчатых пера, переключаемых по программе. Привод от двигателей тросовый. Первый из названных построителей имеет линейный интерполятор, второй – линейно-круговой, а также генератор штрихпунктира и генератор символов. У обеих моделей есть буферная память. Построители либо непосредственно подключаются к ЭВМ, либо работают от перфоленты.

Тросовый привод хорош тем, что дает малую инерцию движущихся частей, и это позволяет быстро изменять направление движения пера, останавливать его и снова заставлять двигаться. Некоторые современные модели позволяют вычерчивать (а не печатать) за секунду 5 – 7 символов. Недостаток – тросовый привод не может обеспечить высокой точности черчения, особенно на больших чертежных машинах. Для точных больших построителей применяют жесткий привод – с зубчатыми рейками или ходовыми винтами (рис. 17). Правда, при этом возрастает инерционность движущихся частей и мелкие детали чертежа приходится вычерчивать с пониженными скоростями. Перо "разгоняется" и "тормозит", как трамвай или автобус.

Вместо шаговых двигателей здесь чаще используют двигатели плавного вращения, но рядом с ними устанавливают датчики обратной связи — цифровые или импульсные, которые посылают электронной управляющей схеме сигналы всякий раз, как перо переместится на величину элементарного шага (по каждой из координат). Блок управления в зависимости от этого и от других обстоятельств решает, какое напряжение на какой из двигателей подать (напряжение может не только "разгонять" двигатель, но и "тормозить" его).

Из строителей этого типа в нашей стране известны "Дигиграфы" чехословацкого производства. Это устройства с реечным приводом и датчиками обратной связи, оборудованные линейно-круговыми интерполяторами, генераторами символов и штрихпунктира, программной регулировкой скорости. Если старая модель, ЕС-7054, развивала скорость черчения до 50 мм/с, а маленькие детали обрабатывала существенно медленнее, то новые модели, например 1208/3,5G (ЕС-7907), имеют максимальную скорость черчения до 400 мм/с или даже до 700 мм/с при быстром разгоне и торможении и точности 0,02 мм. Блок управления выполнен на базе программируемого процессора и позволяет выполнять дополнительные операции, облегчающие работу с машиной. Этот построитель показан на рис. 18.

В нужных случаях применяют большие чертежные установки со столами длиной по 5 — 6 метров. Здесь вместо бумаги часто используется безусадочная и более прочная синтетическая пленка. Кроме черчения тушью или специальной краской применяют гравирование иглами и специальными резцами. Для гравирования выпускается пленка, покрытая специальной краской. Прочерпанные по слою краски прозрачные линии имеют четкие края и гораздо лучше (по сравнению с начерченными) воспроизводятся

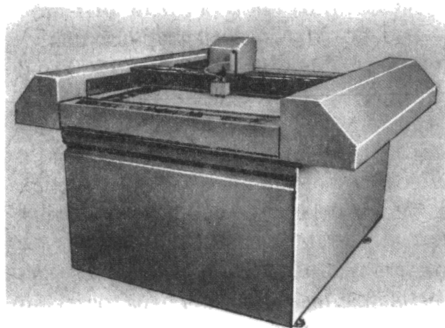


Рис. 18. Графопостроитель "Дигиграф ЕС-7907"

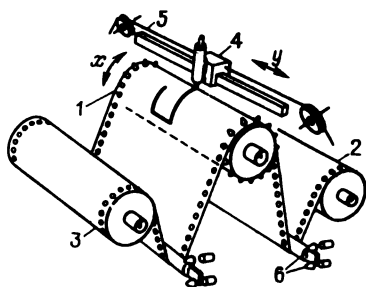


Рис. 19. Схема рулонного построителя: 1 — рабочий барабан, 2 — подающий барабан с бумагой, 3 — приемный барабан (может отсутствовать), 4 — рабочая головка с пером (или несколькими перьями), 5 — трос привода головки, 6 — система подмотки бумаги (с фотоэлементами и управляемыми двигателями)

типографскими и фотографическими способами. Для получения изображений в виде областей используют двуслойную пленку, где тонкий черный или цветной слой липким составом приклеен к более толстой прозрачной основе. Резец процарапывает верхний слой по контурам областей, а затем человек-оператор пинцетом удаляет ненужные части пленки. Есть машины, вычерчивающие изображения лучом света на фотоматериале. Помимо линий они с помощью миниатюрного проектора, смонтированного на рабочей головке, могут впечатывать в чертеж различные знаки, буквы и цифры. Еще одно из дополнительных приспособлений – маленькая телекамера, установленная на рабочей головке и в увеличенном виде передающая изображение части чертежа на телевизионный экран. В сочетании с удобными рукоятками перемещения рабочей головки это позволяет точно установить перо на большом планшете, а также в нужных случаях использовать построитель для считывания с чертежа отдельных точек и линий или просто проверки их координат.

Одна из разновидностей графопостроителей – *рулонные*. Бумажная лента, снабженная по краям перфорацией, при работе сматывается с рулона и может перематываться в обе стороны по рабочему барабанчику (это движение по координате x). Перья же движутся по рейке поперек бумажного листа (координата y). Такие устройства работают быстро, занимают немного места, но, конечно, не могут дать особо точных чертежей, да и выбор сортов бумаги ограничен. Отечественной промышленностью выпускаются устройства марок ЕС-7052, ЕС-7053, рассчитанные на разную ширину бумаги, и некоторые другие модели. Схема устройства показана на рис. 19.

Глядя на сравнительно медленные построители предыдущего поколения, многие сомневались в эффективности машинной графики вообще. Действительно, видимый факт, что "механическая рука" чертит медленнее обычной человеческой, действовал разочаровывающе. Зато теперь работа построителей передовых фирм на выставках приводит зрителей в восхищение. Реально одна из новых моделей построителей выполняет чертеж, содержащий мелкие детали, в 20 раз быстрее (!), чем построитель старого выпуска.

1.4. Полуавтоматическое чтение чертежа

Поговорим о считывании графической информации с чертежа, эскиза, фотоснимка или другого изображения и передаче ее в ЭВМ. Конечно, хотелось бы, чтобы решалась эта задача автоматически – положил чертеж, нажал кнопку – и готово! Ниже мы увидим, что и приборы для этого есть. Но, как правило, в ЭВМ нам нужна не просто копия изображения, а информация с определенным смыслом и содержанием, которая послужила бы пищей для дальнейшей работы программ. Если информация была введена в целом, "без разбора", – возникает иногда очень сложная проблема *распознавания образов*. Человек за полуавтоматическим прибором не только предварительно отсортировывает нужную информацию, но может чисто геометрически

кие данные дополнять командами, кодами и названиями, вводимыми с алфавитно-цифровой клавиатуры или с помощью клеток специальной таблицы, наложенной вместе с исходным изображением на планшет устройства.

Кодировочные устройства этого типа имеют обычно планшет и визир в виде увеличительного стекла с перекрестием либо щуп в виде карандаша, проводом соединенного с блоком управления (рис. 20). Если визир установлен на направляющих, датчики его положения каким-либо образом связаны с этими направляющими. Более современные конструкции имеют свободно перемещаемые визиры и щупы; в толщу планшета в этом случае вмонтированы ряды продольных и поперечных проводников, а визир (щуп) имеет индуктивный контур в виде катушки. Электронная схема периодически (при нажатии кнопки на визире) пропускает по проводникам импульсы, а контур воспринимает импульсы, поступившие к нему от ближайших проводников. Этим способом определяется значение координаты x и, затем, координаты y . В реальных устройствах в катушку передаются импульсы от нескольких ближайших проводников. Сравнивая величины или фазы этих импульсов, схема может уточнить значение координаты. Хорошие модели обеспечивают точность считывания 0,1 мм, хотя проводники в планшете расположены гораздо реже. Как было сказано, многие устройства дополняются алфавитно-цифровой клавиатурой. Если такой клавиатуры нет, ее изображение можно нарисовать на листе бумаги и наклеить на планшет: дело программиста — составить программу, которая бы отли-

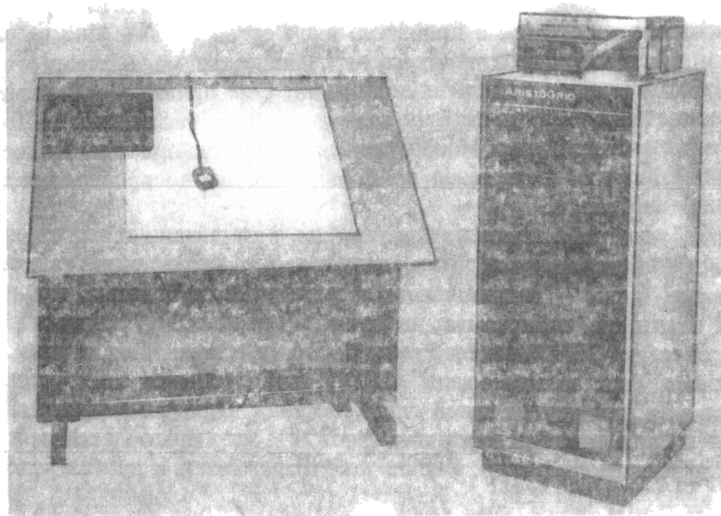


Рис. 20. Кодировочный планшет

чала указания визиром клеток клавиатуры от указания точек считываемого изображения.

Координаты точек считываются с помощью визира или шупа элементарно; их значения записываются на магнитную ленту, перфоленту либо посылаются сразу в ЭВМ. Другие фигуры, например окружности, их дуги, эллипсы, прямоугольники, могут быть заданы каждой несколькими определяющими точками. Этот вопрос подробнее будет обсуждаться ниже; пока можно пояснить, что окружность однозначно задается положением ее центра и какой-либо точки, принадлежащей ей. Некоторые устройства позволяют также считывать кривые линии, проводя визиром вдоль них.

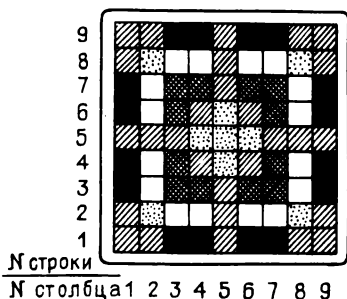
Любопытно, что помимо планшетов с вмонтированными проводниками выпускались различные экзотические конструкции, например. . . акустический прибор с искрой! В этом устройстве шуп содержал маленький разрядник наподобие свечи зажигания в автомобильном моторе. При нажатии на корпус шупа проскакивала маленькая искра, звук разряда ловили два микрофона, в виде узких ленточек смонтированные вдоль двух сторон планшета. По времени запаздывания сигнала (вызванного временем хода звука) схема определяла расстояния от микрофонов до шупа.

1.5. Мозаика в ЭВМ. Предшественники растровых устройств

Примером растрового изображения может служить детская мозаика из маленьких квадратиков или шестиугольников. Изображение можно описать последовательностью чисел или кодов, подобно тому как это сделано на рис. 21. Но в детской мозаике число строк и столбцов невелико, поэтому получить четкое изображение произвольной формы нельзя. Многие знают способ вышивки "болгарским крестом". Вышивают маленькими крестиками, расположенными рядами и столбцами. Еще пример – обычный ковер. Здесь цветных узелков-петелек так много, что с расстояния 1–2 м они сливаются в сплошной узор. Можно привести и другие примеры мозаичных изображений, но нам интереснее приборы, которые воспроизводят такие изображения на бумаге.

Когда на телеграфе появились телетайпы, печатающие текст строками, как обычная пишущая машинка, некоторые телеграфисты увлекались чем-то вроде игры – составлением и передачей грубоватых, но иногда забавных картинок, набираемых из точек, "светлых" и "жирных" букв. Позже, уже когда ЭВМ были снабжены быстродействующими печатающими устройствами (так называемыми АЦПУ), для проверки этих устройств, для рекламы и для других целей стали составлять программы, печатающие из знаков и букв подобия картинок (рис. 22). Этот способ и сейчас применяется для вывода графиков, планов территории, схем и других условных изображений.

Еще позже небольшие компьютеры стали снабжаться печатающими устройствами, по мощности промежуточными между печатной машинкой



Численная матрица

2	2	4	4	2	4	4	2	2
2	1	0	0	2	0	0	1	2
4	0	3	3	2	3	3	0	4
4	0	3	2	1	2	3	0	4
2	2	2	1	1	1	2	2	2
4	0	3	2	1	2	3	0	4
4	0	3	3	2	3	3	0	4
2	1	0	0	2	0	0	1	2
2	2	4	4	2	4	4	2	2

Рис. 21. Кодировка рисунка мозаики



Рис. 22. Часть изображения, выведенного из ЭВМ на печатающем устройстве

и "большим" АЦПУ. Здесь буквы и цифры набираются из нескольких рядов мелких точек. Печатающая головка содержит до десятка штырьков-иглолок, образующих вертикальный ряд. При работе она пробегает вдоль строки, в нужные моменты ударяя штырьками по бумаге. Такие устройства нетрудно переделать в графические, изменить придется разве что электронную схему. Если штырьки расположены на расстоянии 0,2–0,3 мм друг от друга, линии получаются несколько шероховатыми, но для многих целей такое качество уже приемлемо.

Для повышения быстродействия "точечных" устройств "бегающую" головку заменяют неподвижным рядом электродов, расположенных поперек бумажной ленты. Вместо механических ударов через красящую ленту электроды воздействуют на бумагу электрически. Бумага пропитывается специальным составом. Это может быть электрохимическая бумага, чернею-

шая или синеющая в точках, где на нее воздействовали электроды, или бумага электростатическая, некоторое время сохраняющая заряд, переданный ей электродами. Продвигаясь дальше, бумага в электростатическом устройстве попадает в контакт с жидкой краской (точнее – суспензией). Частицы краски прилипают к бумаге в наэлектризованных точках, тем или иным образом закрепляются на ней – и изображение готово (рис. 23).

Скорость работы электростатических, электрохимических и им подобных устройств не зависит от сложности, насыщенности изображения. Насыщенные изображения на таких устройствах могут быть получены в десятки

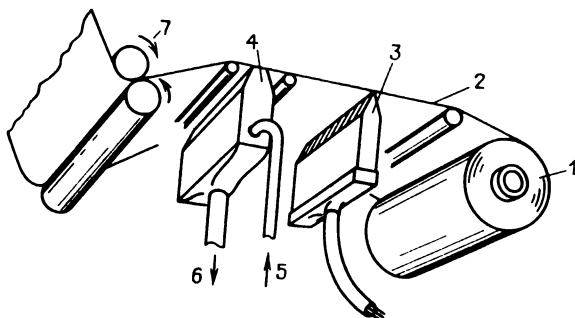


Рис. 23. Схема электростатического устройства: 1 – рулон с электростатической бумагой, 2 – бумажная лента, 3 – линейка с электродами, 4 – ванночка с краской – суспензией, 5 – подача краски, 6 – сток лишней краски, 7 – подача бумаги

раз быстрее, чем даже на новых моделях графопостроителей. Вообще говоря, механические штырьковые, электрохимические, электростатические устройства – не полутоновые. Точки имеют нерегулируемую яркость. Поскольку точек довольно много, их можно группировать в тех или иных местах плотнее или реже, тем самым имитируя тон – при соответствующем снижении, разумеется, четкости (рис. 24).

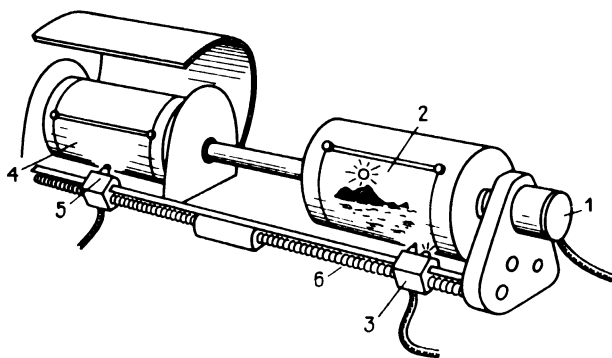
Недавно появились “струйные” устройства с миниатюрными форсунками, выбрызгивающими на бумагу в нужных местах краску. Изображения получаются цветными и яркими, но пока с невысокой разрешающей способностью.

Еще один предшественник графических устройств, подключаемых к ЭВМ, – фототелеграф. Он известен давно. По фототелеграфу передают и цветные картинки, и очень сложные изображения с мелкими деталями – например, отски газетных страниц. На рис. 25 показана схема передающей (сканирующей) и приемной (воспроизводящей) частей типичного фототелеграфного аппарата. Обе части имеют вращающиеся барабанчики. На один из них накладывается передаваемый рисунок, на приемный барабан другого аппарата – фотопленка или фотобумага. Барабанчики начинают вращаться, и у поверхности передаваемого изображения, вдоль оси барабана, движутся маленькая лампочка и фотоэлемент с линзами. Фотоэлемент “просматривает” все изображение по спирали. В каждый момент времени сила тока в фотоэлементе зависит от яркости проходящего под ним элемента изображения. Этот ток, будучи усилен, передается по линиям, а в прием-

Рис. 24. Рисунок с имитацией тона, выведенный на растровом устройстве (часть)



Рис. 25. Схема фототелеграфного аппарата или фотозаписывающего и сканирующего графического устройства: 1 – двигатель, редуктор, 2 – исходное изображение, 3 – лампа, объектив и фотоэлемент, 4 – фотопленка в закрывающейся кассете, 5 – безынерционная лампа и объектив, 6 – ходовой винт перемещения рабочих головок



ном аппарате лампочка, синхронно движущаяся относительно фоточувствительного материала, вспышками, то бледнее, то ярче и также по спирали экспонирует изображение.

Чтобы из такого аппарата получить считывающее и воспроизводящее графические устройства, к обеим частям аппарата вместо телеграфной линии надо подключить ЭВМ. Ток, поступающий от фотоэлемента, надо

разбить на уровни и превратить в последовательность чисел. Строка изображения разбивается тем самым на множество элементов.

Устройства этого типа позволяют читать и воспроизводить изображения с большой точностью. Ширина строки может составлять всего 5 мк. Обрабатываются полутоновые и, на специальной оборудованных аппаратах со светофильтрами, цветные изображения. На аппаратах этого типа воспроизводятся снимки, переданные космическими аппаратами из межпланетного пространства, с других планет. Процесс считывания изображения на сканирующем аппарате – автоматический. Правда, как раньше говорилось, прочитанная растровая модель не всегда поддается расшифровке, но некоторые операции по преобразованию изображений и на этой модели выполнить легко (например, замену цветов, превращение негатива в позитив, деформации всего изображения и др.).

1.6. Электронно-лучевые устройства, графические терминалы

У графических терминалов с электронно-лучевыми трубками есть два предшественника. Это электронно-лучевые осциллографы и телевизоры.

Осциллографы с электронно-лучевыми трубками известны почти с того же времени, что и радиолампы; да и первые, еще ламповые ЭВМ налаживались и проверялись с помощью осциллографов. В большинстве случаев осциллограф работает как двухкоординатный вольтметр – луч его в горизонтальном и вертикальном направлениях отклоняется соответствующими напряжениями, как и перо аналогового самописца. След луча на экране быстро гаснет, и потому процесс черчения лучом повторяется много раз в секунду, на экране при этом мы видим неподвижные или изменяющиеся (бегающие) линии. Впрочем, есть осциллографы с запоминающими трубками, где лучу достаточно один раз пробежать по экрану, чтобы вслед ему высветилась и не гасла, пока не дадут приказа, линия.

В телевизоре луч движется не так, как в осциллографе, а пробегает последовательные строки, в нужные моменты меняя свою яркость. Этот процесс повторяется несколько десятков раз в секунду. Можно сказать, что изображение здесь формируется растровым способом, хотя яркость луча, пока он идет вдоль одной строки, управляется еще аналоговым сигналом.

Таким образом, осциллографы можно считать прообразами "штриховых", или, как говорят, "векторных дисплеев", а телевизоры – "растровых дисплеев". Графический дисплей, или терминал, – это не только экран, на который из ЭВМ выводится картинка. Электронный блок управления содержит различные цепи, расшифровывающие поданные команды и численную информацию; у векторных терминалов изображение, как у графо-построителя, формируется из отрезков (а иногда и кривых линий); отрезки получают путем цифровой интерполяции или с помощью цифроаналоговой схемы – *генератора векторов*. Для периодического обновления изображения блок управления осуществляет *регенерацию*; чтобы меньше

загружать ЭВМ, терминалы обычно снабжаются *буферной памятью*, куда из ЭВМ и поступает численная модель изображения. Помимо графической картинки на экран тем или иным способом могут выводиться надписи, текст, цифры. В большинстве случаев это делается с помощью *генератора символов*.

Графический терминал – устройство не только вывода информации, но и ввода ее в ЭВМ, устройство для диалога человека и машины. Для этого терминалы снабжаются различными дополнительными приспособлениями. Во-первых, это обычная алфавитно-цифровая клавиатура и дополнительные кнопки управляющей функциональной клавиатуры. С помощью алфавитно-цифровой клавиатуры на экран заносят надписи, которые затем поступают в ЭВМ, или редактируют надписи, выведенные по программе. Специальная метка – “курсор”, перемещаемая дополнительными кнопками, позволяет выбирать место, куда нужно вводить новые символы. Функциональные кнопки посылают в ЭВМ сигналы прерывания, а программа в ЭВМ реагирует на это тем или иным способом. Например, при нажатии одной из кнопок изображение с экрана может быть стерто, при нажатии другой – показано в измененном масштабе и т.д.

Оригинальное изобретение “световое перо” является по виду “пером”, а по устройству это скорее “глаз”; оно содержит фотозлемент в своем корпусе или по световоду из стеклянных нитей передает свет в корпус терминала. Направленное на экран перо воспринимает световой сигнал в момент, когда электронный луч высветил какую-либо деталь изображения перед острием пера. Этот сигнал электронная схема фиксирует и опознает, какую деталь мы указали. А если мы хотим “нарисовать” что-либо на еще темном экране? Можно привести два способа. Один такой: при нажатии на кнопку или корпус пера электронная схема генерирует на экране растр – т.е. луч пробегает по экрану строками; экран как бы вспыхивает на момент. В некотором месте некоторой строки перо воспринимает сигнал; обработав его, схема определяет положение пера. Другой способ заключается в том, что на экран дополнительно выводится *маркер* – группа точек или маленьких штрихов (рис. 26). Перо наводит на маркер, и тут начинает

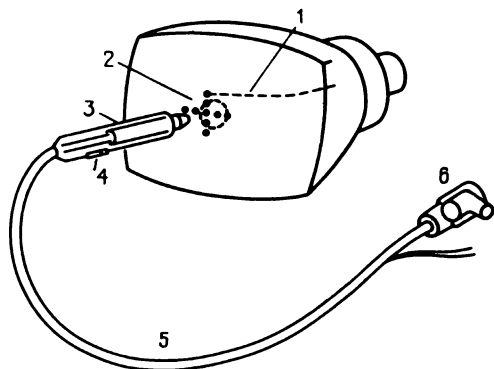


Рис. 26. Маркер на экране терминала и “световое перо”: 1 – электронный луч, 2 – высвеченные точки маркера (зона, воспринимаемая “световым пером”, отмечена штриховой линией), 3 – “световое перо”, 4 – микровыключатель в корпусе пера, 5 – световод и провода, 6 – светочувствительный элемент

работать система слежения. При не очень быстром перемещении пера маркер движется за ним, как будто перо его притягивает. На самом же деле схема все время следит, какие точки маркера засвечивают перо, а какие — нет. В нейтральном положении перо должно засвечивать центральные точки. При смещении пера оно засвечивается какими-то боковыми точками. Схема реагирует на это, смещая маркер в нужную сторону. Координаты центра маркера в нужные моменты передаются в программу и могут быть использованы.

В дополнение к световому перу или вместо него некоторые модели терминалов снабжаются наклоняемыми или шаровыми рукоятками, с помощью которых маркер можно перемещать по экрану в разных направлениях; другое приспособление — небольшой планшет и визир, устроенные так же, как описанные выше кодировочные устройства. Перемещение визира вызывает синхронные движения маркера на экране. Дополнительное преимущество: исходное изображение можно положить на планшет и "перекопировать" визиром на экран, а затем ввести в память вычислительной машины.

Некоторые модели терминалов снабжаются собственными процессорами, позволяющими ряд действий совершать без обращения к основной вычислительной машине. Основой графического диалога и эффективного использования технических возможностей является программа; с примерами таких программ читатель познакомится ниже.

В разных странах выпускается множество моделей графических терминалов (рис. 27, а); для отечественных машин серии ЕС предназначены устройства ЕС-7064 (рис. 27, б), ЕС-7905. Это векторные устройства с буферной памятью и генераторами символов, со световыми перьями. Новая модель имеет также планшетик с визиром. Для машин серии СМ выпускались и выпускаются различные устройства, в том числе упрощенные, не имеющие даже своей буферной памяти: изображение считывается непосредственно из памяти ЭВМ. Другие терминалы (например, "Графит") — "интеллектуальные". Они имеют собственные программируемые процессоры, т.е. мини-ЭВМ; устройства этого класса обеспечивают не только генерацию на экране прямолинейных отрезков (векторов), но и окружностей и их дуг.

В последнее время в мировой практике стали шире применяться растровые терминалы — как упрощенные "точечные", не рассчитанные на показ полутоновых изображений, так и высококачественные полутоновые и цветные. В основном качество растрового экрана определяется числом строк и столбцов, из которых формируется картинка. Читатель может ориентироваться на число строк развертки в наших телевизорах — их 625. Так что экраны с числом строк 250-300 дают изображение невысокой разрешающей способности, "крупнозернистое". Для САПР, для сложных изображений они мало пригодны. В высококачественных установках сейчас число строк составляет 1024, соответственно такое же и число столбцов.

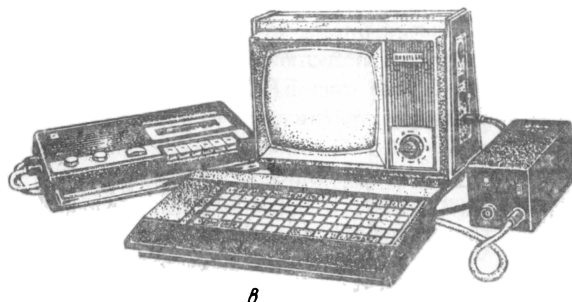
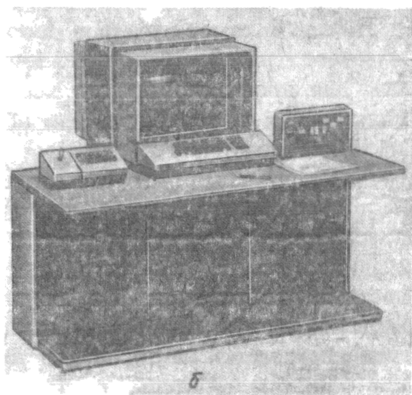
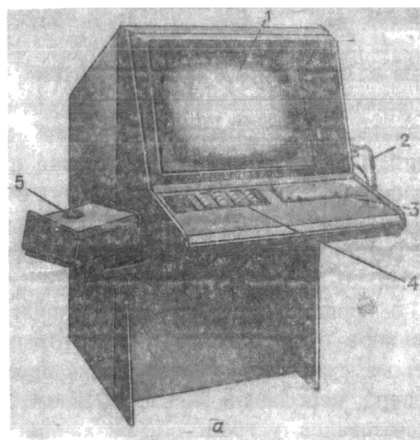


Рис. 27. Графические терминалы: *а* – общий вид: 1 – электронно-лучевой экран, 2 – "световое перо", 3 – алфавитно-цифровая клавиатура, 4 – функциональная клавиатура, 5 – шаровая рукоятка для перемещения маркера; *б* – электронный-лучевой графический терминал ЕС-7064; *в* – персональный бытовой компьютер БК-0010, подключенный к телевизору и магнитофону

Иногда в качестве растровых дисплеев используют переделанные или даже серийные бытовые телевизоры. Особенно это подходит для персональных компьютеров.

Дополнительно к растровому терминалу для ввода информации в ЭВМ применяют переоборудованные малогабаритные телекамеры; для считывания схем, чертежей, фотоснимков они укрепляются на штативах.

1.7. Персональные графические компьютеры

Пожалуй, самое заметное достижение электроники в последние годы — массовое распространение миниатюрных персональных компьютеров, как для профессионального использования, так и "бытовых", — для использования в хозяйстве, для обучения и даже детских игр. Многие из таких компьютеров обладают графическими возможностями: на экране дисплея или телевизора, к которому ПЭВМ подключается, высвечиваются не только тексты и числа, но и графические изображения. Экраны обычно работают в растровом режиме; дешевые бытовые компьютеры имеют сравнительно грубый растр (пока!), но нередко способны формировать цветные изображения. Как дополнительное оборудование к некоторым моделям можно приобрести наклоняемые рукоятки, управляющие перемещением метки на экране, световое перо, небольшой электромеханический графопостроитель, игольчатое печатающее устройство. Небольшие, но пока довольно дорогие профессиональные модели имеют быстродействие и объем памяти больше, чем большие ЭВМ десяти — пятнадцатилетней давности выпуска. Здесь видна некоторая непропорциональность развития: многочисленные периферийные устройства — быстродействующую печать, шкафы с магнитными лентами и т.д., оказалось гораздо труднее миниатюризовать, чем электронные схемы. Простейшие же бытовые, учебные и "игровые" компьютеры стоят примерно столько же, сколько телевизор, к которому они подключаются. Первым массовым бытовым компьютером в нашей стране стал БК-0010 (рис. 27, в). Этот компьютер подключается к бытовому телевизору и кассетному магнитофону, содержит оперативную память объемом 32 Кбайта, обычно поровну разделяемую на хранение программ пользователя и численной модели изображения на экране (при решении неграфических задач на экране можно оставить четыре строки текста и отвести для программ пользователя 28 Кбайт). Помимо этого, в постоянной памяти на микросхемах хранится операционная система, базовые подпрограммы ввода-вывода информации, интерпретатор с языка фокал. Язык этот сходен с распространенным для ПЭВМ языком бейсик, но проще последнего. Имеет он, к сожалению, и определенные ограничения, затрудняющие решение некоторых задач. Позже БК-0010 стали снабжаться бейсиком. Специалисты могут составлять и быстродействующие машинные программы для этого компьютера. Изображение на экране (включая и изображения букв, цифр) строится растровым способом, растр содержит 256 строк по 256 или 512 точек в строке; если используется 256 точек в строке, в цветном варианте можно получать три цвета: красный, синий, желто-зеленый; в черно-белом — два уровня яркости. На магнитную ленту под различными именами можно записывать как программы, так и данные. Дополнительный разъем под оригинальным названием "ПОРТ ввода-вывода" позволяет соединять машинку с другими устройствами, например игольчатой печатью, бытовыми приборами.

Эти компьютеры оказались достаточно удобными для организации учебного процесса в вузе, особенно если иметь в виду такие "графические" дисциплины, как начертательную геометрию, черчение. Но, конечно, не весь круг учебных и тем более производственных задач может решаться на таких маленьких машинах; нужна и более мощная техника. В нашей стране приступили к выпуску более мощных ПЭВМ, снабжаемых гибкими и жесткими дисками и дополнительным оборудованием.

1.8. Специальное оборудование

Для специальных работ изобретают и выпускают все новые и новые машины и приборы. Так, для считывания формы и контроля размеров пространственных объектов – объемных макетов, моделей, натуральных деталей – используют измерительные станки разных размеров. Шуп такого станка можно перемещать по крайней мере в трех направлениях соответственно трем координатным осям. Координаты измеряемых точек регистрируются на бумажной или магнитной ленте либо поступают непосредственно в ЭВМ. Есть измерительные устройства, где движением шупа в процессе измерений управляет подключенная миниЭВМ; появлялись в печати также сообщения об устройствах с бесконтактными щупами, с лазерными лучами, посредством которых производятся измерения.

Для измерения крупных объектов – зданий, мостов, земной поверхности – уже давно применяется техника фотограмметрии – как наземной, так и аэрофотограмметрии. В ее основе лежит фотосъемка стереопар и серий снимков. Эти снимки затем просматриваются и измеряются в специальных измерительных приборах, по сути дела, представляющих собой стереоскопы с дополнительными приспособлениями. Современная фотограмметрия становится автоматизированной и смыкается с машинной графикой; новые модели измерительных стереоприборов снабжаются микропроцессорами или подключаются к вычислительным машинам. Есть немало общего с машинной графикой и в других специальных областях. Так, в полиграфии расширяется использование автоматизированного фотонаборного оборудования, заменяющего устаревшие линотипы с расплавленным металлом и ручные "кассеты" с маленькими кубиками-литерами. Специальное оборудование применяется для раскроя тканей и листов металла и вырезки из них заготовок.

Наконец, теснейшим образом связаны с машинной геометрией, а иногда и с графикой станки с числовым программным управлением и робототехника. Геометры могут рассматривать станки с ЧПУ как устройства для вывода из ЭВМ и "овеществления" объемной геометрической информации. Для производственников и технологов, работающих с такими станками, машинная геометрия становится необходимой, как только усложняется форма изготавливаемых деталей. То, что раньше делалось с помощью множества специальных приемов и приспособлений, а иногда и "на ощупь", теперь становится на четкую математическую основу.

Представьте себе фигурную панель автомобильного кузова. Чтобы штамп для ее изготовления отфрезеровать на станке с ЧПУ, форму панели и штампа необходимо смоделировать в ЭВМ, и математика здесь будет отнюдь не простая!

1.9. Автоматизированные рабочие места (АРМ)

Итак, если раньше конструкторы, технологи, архитекторы работали с бумагой, карандашом, линейками и чертежными досками, теперь к этим традиционным средствам добавляются вычислительные машины, экраны, графопостроители и другое специальное оборудование. Как в этих условиях обеспечить не просто возможность работы с таким оборудованием, но работы гораздо более эффективной, чем раньше? На предыдущем этапе развития вычислительной техники создавались вычислительные центры "общего пользования" со своими операторами, инженерами и, конечно, начальниками. Во многих случаях эти начальники старались установить порядки, удобные не столько для заказчиков и решаемых задач, сколько для самих вычислительных центров и ЭВМ. Математики и инженеры выстраивались в очереди у столов или окошек, где "принимали задания" и "выдавали результаты", а сроки решения сложных задач растягивались на месяцы. Насколько хорошо работали ЭВМ за закрытыми дверями — заказчикам было неизвестно. Для решения определенных задач по готовым программам такое "операторское обслуживание" (но, конечно, хорошее) останется эффективным и на будущее. Но годится ли этот режим для творческих работ, автоматизированного проектирования? Ясно, что нет. Оборудование должно быть "наготове" и использоваться тогда, когда нужно. Решить этот вопрос можно двумя путями. Первый — отдать ЭВМ и графическое оборудование в распоряжение пользователя. В этом случае оборудование будет "недогруженным", как "недогружены" логарифмические линейки, рейсфедеры и карманные калькуляторы, которыми мы пользуемся лишь по мере необходимости. На это можно пойти, только если ЭВМ и подключенные к ней устройства будут относительно недорогими и малогабаритными. И такой путь вполне реален.

Другой путь — одновременная работа на большой машине нескольких пользователей. Сейчас почти все крупные ЭВМ могут работать в мультипрограммном режиме. Машины серии ЕС нередко оборудуются десятком-другим терминальных станций (пока в основном с алфавитно-цифровыми дисплеями). В принципе, можно организовать и одновременную работу нескольких графических устройств, подключенных к одной ЭВМ, надо только иметь необходимый запас мощности процессора и объема оперативной памяти; иначе пользователи будут бесполезно терять время у терминалов наподобие того, как теряем его иногда все мы, пытаясь дозвониться в справочную службу по телефону.

Оба пути — использования индивидуальных рабочих мест с персональными машинами и коллективного использования больших машин —

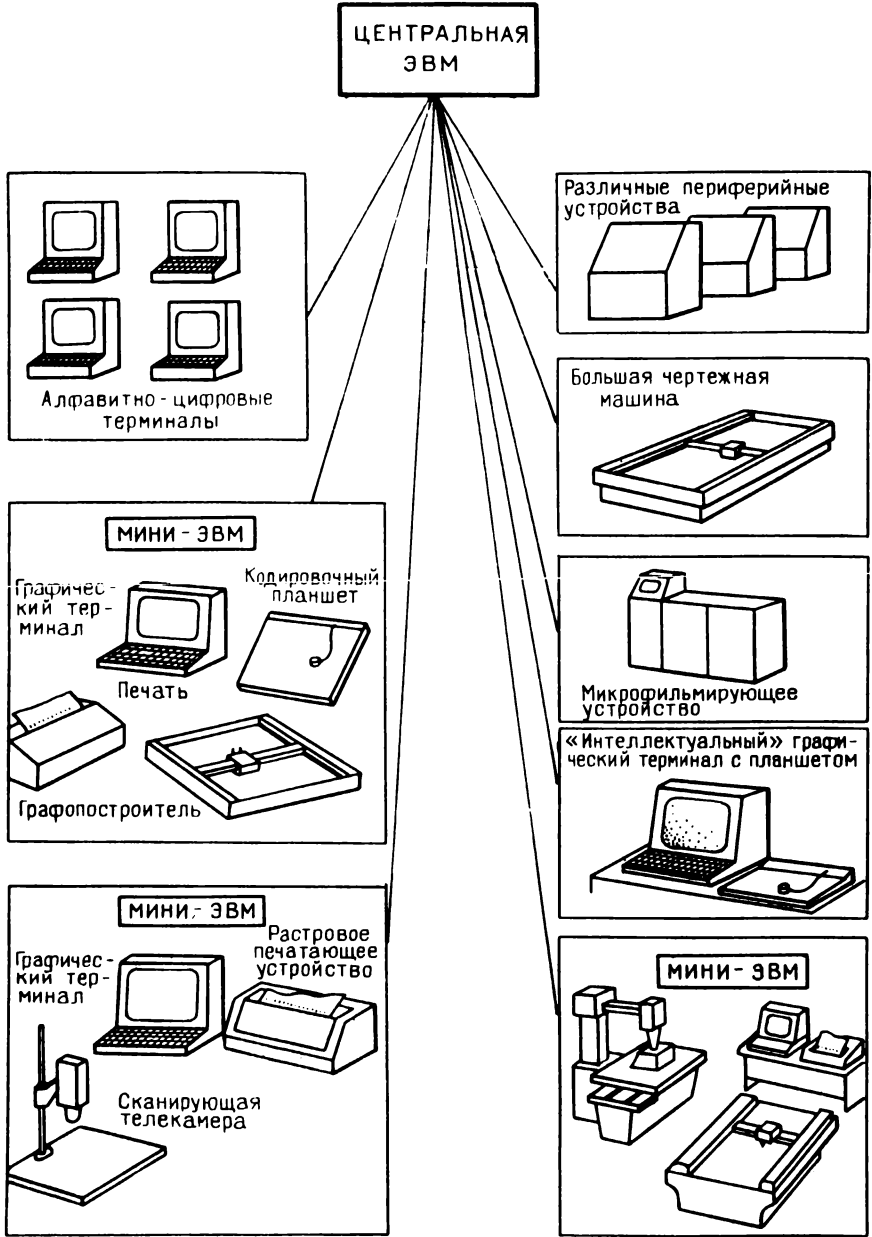


Рис. 28. Пример состава аппаратной части автоматизированной системы коллективного пользования

можно объединить, если компактные рабочие места с малыми компьютерами соединить с большой (центральной) ЭВМ (рис. 28). В этом случае простые задачи пользователи решают на своих малых машинах, не загружая зря центральную; для решения сложных вопросов или обращения к общему хранилищу информации вызывается большая ЭВМ, которая, вообще говоря, может даже находиться в другом городе!

В будущем предполагается создание больших информационно-вычислительных сетей, наподобие современных телефонных, которыми будут пользоваться тысячи и десятки тысяч абонентов.

В некоторых случаях могут оказаться эффективными и машины средней мощности, наподобие выпускаемых машин серии СМ, мощность которых, постепенно повышается и которые, уже могут обслуживать несколько пользователей в режиме мультипрограммирования. Для создания систем, эффективных, в разнообразных производственных условиях надо обеспечить легкость подключения различных графических устройств к разным маркам ЭВМ, а также и легкость соединения ЭВМ друг с другом.

НАЧИНАЯ С ПРОСТЫХ КАРТИНОК

**2.1. Программное формирование
и геометрическое описание изображений.
Базовое программное обеспечение**

Чтобы с помощью ЭВМ и графического устройства получить изображение, пишут программу. Программы обычно составляют на таких алгоритмических языках, как фортран, паскаль, ПЛ-1, бейсик. В программе можно вычислять точки, куда следует передвигать перо построителя, вычислять необходимые размеры и параметры геометрических фигур и т.д. Результатом работы программы должны быть команды для графического устройства, записанные на магнитную ленту или перфоленту либо передаваемые по электрическому кабелю. Этот общий принцип можно назвать *программным формированием изображений*. Правда, не всегда он применяется в непосредственном виде. Во-первых, если не считать учебных, исследовательских и специальных задач, составлять для каждого чертежа новую программу — слишком долго и дорого. Стремятся составлять по возможности универсальные программы — такие, которые могли бы давать целые серии изображений того или иного типа. К примеру, одна программа вполне может изображать болты самых разнообразных размеров и пропорций, другая — подшипники, третья — втулки. Для получения конкретного изображения программе тем или иным способом надо дополнительно передать исходные данные в виде чисел или специальных кодов, это проще, чем каждый раз разрабатывать новую программу. Во-вторых, программистам-пользователям в работе очень помогает модульный принцип современного программирования, т.е. то, что в новых программах можно использовать ранее написанные части — подпрограммы, модули, хранящиеся в библиотеках на магнитных дисках или лентах. Это, в частности, освобождает от необходимости налаживать непосредственную связь программ пользователей с графическими устройствами. Такие связь и кодировка команд для устройств обычно специфичны, требуют представления информации в особом виде, и тут приходится обращаться к программированию на языках, более близких к машинным командам (ассемблер, автокод и др.). Предприятия-изготовители графических устройств учитывают нужды пользователей и дают впридачу к устройствам комплекты *базовых подпрограмм*. Таких подпрограмм может быть немало, и функции у них простые: начертить отрезок, окружность или

дугу, сменить номер используемого пера, перейти на штриховые или штрихпунктирные линии, вывести текстовую или цифровую надпись, изменить масштаб изображения и др. Эти подпрограммы увязаны, с одной стороны, с конкретными конструкциями графических устройств, с другой стороны — к ним легко и удобно обращаться из программ и подпрограмм пользователя. С помощью одной строки можно обеспечить нанесение на чертеж отрезка, дуги или другой фигуры (ниже мы увидим, как это выглядит). Для отечественных графических устройств серии ЕС базовые подпрограммы включены в комплект операционной системы ОС-ЕС.

Базовые подпрограммы для различных графических устройств различны; пользователям хотелось бы, чтобы их программы не надо было изменять при переходе на новое графическое устройство. Для этого названия базовых подпрограмм и обращения к ним должны быть однотипными, либо дополнительно следует разработать некоторый набор модулей — "переходников", посредством которых унифицируется обращение к различным графическим устройствам. Один из таких международных стандартов, намечаемый к использованию и в нашей стране, называется ГКС (графический корневой сегмент).

Значительно больше возможностей представляют пользователям крупные графические системы и "пакеты графических подпрограмм". Здесь шире ассортимент типовых фигур и способов их задания, больше возможности различных деформаций и преобразований, предусматриваются операции геометрического конструирования, векторной алгебры, подсчета геометрических характеристик и т.д. Системы "трехмерной" графики позволяют в различных проекциях отображать пространственные объекты. Различные системы ориентируются на разные области применения и включают дополнительные разделы, например — разделы формирования и отображения специальных криволинейных поверхностей, раздел отображения скалярных и векторных полей, раздел "раскраски" для полутоновых цветных изображений и т.д. Конечно, есть различия между системами, используемыми на персональных и на больших компьютерах. Диалоговая машинная графика отличается от "пакетной", рассчитанной на использование графопостроителей. В итоге — в разных странах в настоящее время эксплуатируются сотни графических систем. Из отечественных разработок известны: комплекс графических подпрограмм Графор, система геометрического моделирования ФАП-КФ, графическая система Алграф, пакеты Графал, Униграф, Аниграф, Граф-СМ, Смог, Грас и др. В этой книге для объяснений и примеров мы будем использовать систему Алграф; во-первых, она достаточно универсальна и позволяет продемонстрировать многие аспекты машинной графики, во-вторых, используется в учебной работе в таких вузах, как МАДИ, МАРХИ, МИРЭА, МТИЛП и др., автор принимал участие в ее разработке и хорошо ее знает.

Не все графические и геометрические системы рассчитаны на описанный выше принцип программного формирования изображений, но подавляю-

щее большинство систем в программном отношении представлены библиотеками подпрограмм и программных секций. В некоторых системах и пакетах предприняты попытки расширить функции используемого языка и тем создать больше удобств в отношении задания геометрических образов, их поиска и манипулирования с ними. И все же не во всех случаях программный принцип создания изображений оптимален. Трансляция и редактирование программы, подготовленной на языке типа фортран, на машине средней мощности делятся минуты полторы и более; лишь после этого программа начинает выполняться. Этим путем быстрого графического диалога не получишь. Да и нужно ли составлять новые машинные программы, если задачей является кодировка заданных изображений, содержащих известные элементы, с целью их последующего вычерчивания? Здесь не нужны сложные вычисления и логические операции, задача стоит не столько расчетная, сколько информационная. В основном числами и условными кодами надо описать геометрические (и графические) элементы. Таким образом, мы пришли к языку *описания геометрической и графической информации*. Поскольку постановка задачи достаточно конкретна, можно надеяться, что для несложных изображений язык будет простым и кратким.

В дополнение к такому языку нужны программные средства его обработки. Это может быть программа типа диспетчера (монитора), которая расшифровывает информацию, записанную на языке, и в нужные моменты обращается к различным модулям графической системы (режимы, подобные этому, называются *интерпретацией*). Текст на геометрически-ориентированном языке может также переводиться (транслироваться) в какую-либо другую форму, более удобную для машинной обработки. В некоторых системах специальная программа — *плетранслятор* переводит записи с геометрически-ориентированного языка в форму, стандартную для распространенного языка типа фортран или ПЛ-1; затем программы обрабатываются уже стандартными средствами. Хотя такая многоступенчатая обработка имеет свои преимущества, в отношении затрат машинных ресурсов она весьма расточительна и для диалоговых режимов недостаточно оперативна.

В состав упомянутой системы Алграф как дополнительное средство входит язык описания геометрической информации алграф-Р и средства его обработки. В применении к простым изображениям язык легко осваивают школьники 5 — 6 классов, в то же время у него есть и алгоритмические качества: можно организовывать циклы, условные переходы, вызовы блоков информации (наподобие вызовов подпрограмм в общем программировании). При необходимости геометрические образы можно описывать не только числами, но и вводить параметры-переменные и даже задавать некоторые вычисления. Ниже этот язык будет использоваться для того, чтобы проиллюстрировать ряд понятий машинной графики.

Обработка текстов, написанных на языке алграф-Р, двухступенчатая: сначала специальный транслятор (подпрограмма с названием READAL)

переводит описание во внутримашинную форму численно-кодового массива (ЧКМ), затем в действие вступает монитор-интерпретатор (подпрограмма KTRAN), выполняющий указанные действия и по мере необходимости обращающийся к различным подпрограммам графической системы. Тем не менее обработка небольших текстов, включая срабатывание всех уровней графического обеспечения, происходит достаточно быстро; после набора текста на экране графического терминала и нажатия кнопки "ввод" изображение появляется через 3 – 5 с.

Упрощенный вариант "мини-системы" для ПЭВМ БК-0010 позволяет использовать сокращенную версию этого же языка; здесь первая фаза обработки выполняется одновременно с набором текста на клавиатуре машинки; затем в действие также включается программа типа интерпретатора. Более сложные изображения могут быть сформированы по фрагментам, тексты которых набираются в двух-трех верхних строках экрана, а графические элементы постепенно накапливаются в его средней части. В оперативную память этой маленькой машинки сразу нельзя поместить подпрограммы обработки всех типов линий, фигур и тел, предусмотренных языком. Потому подготовлены варианты наборов подпрограмм для решения задач по разным темам.

Общность геометрических алгоритмов и языка для маленьких и больших машин дает возможность на начальных этапах обучения использовать дешевые ПЭВМ как тренажеры, как модель "настоящей" большой машины и освободить дорогостоящую технику для решения более сложных задач.

Если автоматизированные графические системы используются как техническое средство обучения или для развлекательно-познавательной игры, общение человека с машиной должно быть простым. Для конкретных задач такое общение можно свести к ответам типа "да" и "нет" на выставляемые машиной вопросы, к вводу с клавиатуры нескольких чисел и другим несложным операциям. В их ряду – указание визиром или световым пером нужного символа или нужной надписи, светящихся на экране либо нарисованных в таблице, наложенной на кодировочный планшет. Из этих предпосылок развилась *техника графических клавиатур*, "световых кнопок" и "меню". Удобство и гибкость метода состоит в том, что такие клавиатуры и кнопки программируются и рисуются в соответствии с конкретными задачами. Обучающая или проектирующая диалоговая программа в нужные моменты выводит на экран терминала вопросы, советы, подсказки, примеры изображений и различные таблицы – "меню". Указывая клетки этих таблиц, можно выбирать нужные команды, графические элементы и т.д.

Способ запросов и ответов широко распространен на маленьких ПЭВМ; языки типа бейсик или фокал включают простые для использования операторы вывода на экран запроса и принятия численного или словесного ответа. Кнопки обычной алфавитно-цифровой клавиатуры в нужных случаях можно "перепрограммировать" и использовать как функциональные,

В ↑	П ↓	СПЛОШ	ПУНК1	ПУНК2
НПЕРА	НПЕРА1	ЛИН	ОТРЕЗ	ОК6
ОК18	ОК19	ОК20	ОКТ	ОК2
ЭЛЛ	ЭЛЛ1	ПРМК1	ПРМУГ4	ПРМУГ3
КРИВ	КРИВ2	КРИВ С	ПОЛ	ПОЛ1
ШТР	а			

PV	PG	ЗЕМ	ВКЛ	ПЕРК
R	R1	R2	R3	RП
C	C1	C2	C3	C4
D	D1	F	DS	D2
T	TN	T2	T3	T4

б

Рис. 29. Примеры графических клавиатур

а для облегчения работы наложить на клавиатуру "маску" с новыми обозначениями. Одна из кнопок теперь может, например, обозначать приказ отображения отрезка, другая – отображения окружности и т.д.

На рис. 29 показаны части графических клавиатур для кодировочных планшетов, предназначенные для считывания изображений, содержащих типовые элементы того или иного рода. Теперь не надо обмерять типовые части по штрихам и точкам. Достаточно дотронуться визиром до клетки таблицы, а затем до нужной точки исходного эскиза – и можно считать, что типовой элемент "вмонтирован". Правда, иногда для указания ориентации элемента или его размеров требуется задать две-три привязочные точки, но и это несложно.

Помимо таких графических действий при работе с дисплеями и кодировочными планшетами, как мы знаем, "рисуют" новые графические элементы, "стирают" или изменяют другие. Конечно, все такие операции в ЭВМ обрабатываются программно. Но для пользователя это уже не составление, в обычном смысле, новых программ. Это – *графическое взаимодействие* с машиной, дополняемое вводом команд и каких-либо чисел. Такому методу общения с машиной сулят большое будущее, особенно когда имеют в виду компьютеры пятого поколения. Читатель легко поймет, что базовое обеспечение для таких режимов должно включать в себя не только средства вывода на экран различных графических элементов, символов и надписей, но и средства "расшифровки" информации, поступающей от графического устройства. В частности, если пользователь указал визиром клетку графической клавиатуры, программы в ЭВМ по координатам этой точки должны определить, что за клетка указана и что в связи с этим программа должна выполнить.

2.2. Геометрия + алгебра = аналитическая геометрия?

Различные фигуры и геометрические действия моделируются в ЭВМ числами и кодами, геометрические задачи решаются путем вычислений. Вообще говоря, "дружба" алгебры, геометрии и вычислений началась задолго до появления ЭВМ. Люди использовали счеты, логарифмические линейки, миллиметровую бумагу и прочие средства "малой механизации". Еще в 1637 году знаменитый французский ученый Рене Декарт издал свой труд "Геометрия", в котором, можно сказать, заложил основы *аналитической геометрии*. С тех пор *декартовы системы координат* – основа численного моделирования геометрических образов. Как известно, ни теория относительности, ни искривление пространства, ни новые физические концепции не могут опровергнуть сути декартовых координат, а могут лишь уточнить геометрию мира и очертить границы применимости обычных координат. Знакомство с координатами x , y начинается еще в школе, когда строится график простейшей функции. За единичными исключениями все графические устройства работают на базе прямоугольных декартовых координат. Даже номер столбца и номер строки для точки растрового изображения можно считать специфической разновидностью прямоугольных координат. Геометры и конструкторы в нужных случаях пользуются и другими координатными системами, например полярными координатами, сферическими, логарифмическими и др. Перед выводом информации на графические устройства значения координат могут быть пересчитаны по специальным формулам.

Итак, хорошо известно, что точка на плоскости определяется двумя, а в пространстве – тремя координатами. А как другие геометрические образы? Такие фигуры, как треугольник, многоугольник, ломаная сами определяются несколькими точками, и с ними все ясно. Другие же фигуры можно задавать не только координатами, но и величинами, так или иначе с ними связанными. Из школьного курса известно, что прямую на плос-

кости можно задать двумя числами – угловым коэффициентом и длиной отрезка, отсекаемого ею на оси ординат. Последнее число – координата у для той точки прямой, которая имеет нулевую координату x . Подумайте, как связать со значениями координат угловой коэффициент?

Окружность задается тремя числами – координатами центра и радиусом. Для задания эллипса, кроме центра, надо определить величины двух его полуосей и еще направление одной из осей, например – в виде угла, под которым она наклонена к оси x . Для пользователя важно, что один и тот же геометрический образ можно определить различными способами. Например, прямую можно задать, указав две принадлежащие ей точки, окружность – координатами трех ее точек и т.д.

Среди различных способов задания геометрических образов выделяются "самые хорошие" – те, для которых количество параметров минимально. Для окружности минимальное число параметров – три, для прямой – два, а чтобы задать ограниченный отрезок, меньше чем четырьмя числами не обойдешься! Минимальное количество параметров, задающих образ, называется *параметрическим числом* этого образа.

Составляя программы и алгоритмы машинной графики и геометрии, разрабатывая способы кодирования объектов, надо знать параметрическое число геометрического образа. В табл. 1 приведены параметрические числа для некоторых геометрических образов.

Т а б л и ц а 1

Параметрические числа для некоторых геометрических образов

Объект	Размерность пространства	Параметрическое число	Объект	Размерность пространства	Параметрическое число
Прямая на плоскости	2	2	Прямоугольник частного положения	2	4
Прямая в пространстве	3	4	Прямоугольник общего положения	2	5
Отрезок на плоскости	2	4	Эллипс частного положения	2	4
Отрезок в пространстве	3	6	Эллипс общего положения	2	5
Отрезок на плоскости, проводимый от фиксированной точки	2	2	Эллипсоид	3	9
Окружность	2	3	Линия второго порядка	2	5
Окружность в пространстве	3	6	Поверхность второго порядка	3	9
Сфера	3	4	Линия полинома степени n	2	$n + 1$
Дуга окружности (на плоскости)	2	5			

Помимо "настоящих" численных параметров в задании объекта или геометрической операции могут участвовать "логические параметры". В программе не обязательно использовать для них переменные логического типа; можно ограничиться целыми числами 0 и 1 или же устанавливать параметр по знаку числа. Логические параметры не влияют на параметрическое число объекта, и, переходя от одного способа задания объекта к другому, их нередко удается ликвидировать. Так, точка на окружности может быть задана значением одной из ее координат, например координаты x . Но дополнительно надо указать, на верхней или нижней полуокружности она находится. Для этого достаточно логического параметра. Можно перейти к другому способу задания точки, где логического параметра не требуется: задавать точку соответствующим ей центральным углом.

В машинной графике существенно, с какого конца вычерчивается отрезок, дуга или ломаная линия и во многих случаях геометрические элементы должны задаваться с учетом направления их вычерчивания. Указание направления на линиях или положительной и отрицательной стороны на поверхностях нужно и для некоторых специальных целей. Например, если поверхность ограничивает твердое сплошное тело, то ясно, что у нее лишь одна сторона может быть видима. На рис. 30 изображена фигура, состоящая из отрезков и дуг окружностей — наподобие ремня, охватывающего невидимые шкивы. Каждую их окружностей-шкивов можно задать тремя числами, а дальше работает специальная подпрограмма, которая найдет касательные прямые к окружностям, выделит граничные точки отрезков

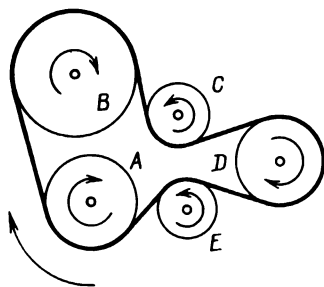


Рис. 30. Построение составной линии из дуг окружности и отрезков. Линия чертится в направлении, указанном стрелкой. Радиусы окружностей C , E заданы положительными, окружностей A , B , D — отрицательными

и дуг и т.д. Некоторые шкивы могут быть охвачены ремнем с одной или с другой стороны, поэтому, чтобы не ошибиться, представим себе, что ремень и шкивы вращаются. Каждый из шкивов может вращаться против или по часовой стрелке. В математике вращение против часовой стрелки считается положительным, по стрелке — отрицательным. Эти дополнительные данные — направления вращения или, как говорят, *направления обхода* окружностей надо дополнительно сообщить подпрограмме. Чтобы не вводить лишнего числа, присваивают радиусу окружности (в прин-

ципе, положительной величине) знак плюс или минус, который и определяет на ней направление обхода.

Дуги, кривые во многих случаях определяют точками, через которые они должны проходить. Но иногда в точке дополнительно хотят задать направление, в котором должна проходить линия. Можно было бы для этого воспользоваться касательной, но на ней надо еще определить одно из двух возможных направлений. Поэтому вместо касательных прямых используются касательные векторы, или *векторы направления*. Вектор

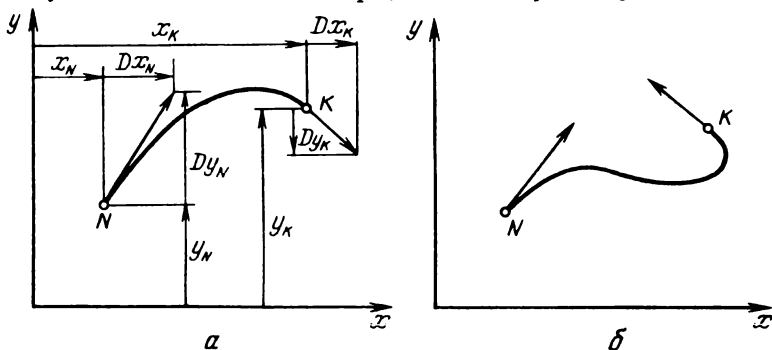


Рис. 31. Задание дуги кривой линии координатами конечных точек (N – начальная, K – конечная точки) и векторами направления в них; показаны два варианта для противоположных направлений конечного вектора

(на плоскости) можно задать двумя его проекциями – компонентами; касательные векторы обычно имеют произвольную длину и в принципе для задания такого вектора на плоскости было бы достаточно одного числа (угла его наклона к оси x). Проекциями предпочитают пользоваться из соображений точности и удобства. На рис. 31, *a* показаны дуги, построенные по двум конечным точкам и касательным векторам в них. Этот набор – граничные точки и касательные векторы в них – одна из типичных *конфигураций исходных данных*. На рис. 31, *б* показано, какая кривая может получиться, если направление одного из векторов изменить на противоположное. Активно используются в машинной графике и такие понятия, как нормали к линиям и поверхностям, кривизна и др. Нормаль, как известно, – прямая, проведенная через точку линии под прямым углом к касательной в этой точке. Нормальный вектор для точки плоской линии можно получить, повернув на прямой угол в положительном направлении касательный вектор. Таким образом, у линии можно будет выделить две “стороны”: “положительную”, с которой будут располагаться нормальные векторы, и “отрицательную”. При смене направления, выбранного на линии, все касательные и нормальные векторы меняются на противоположные; взаимно поменяются и стороны линии. По этой методике замкнутая или неограниченная линия делит плоскость на положительные

и отрицательные области. Проверьте, следуя нашим объяснениям, какой знак следует присвоить точкам круга, ограниченного окружностью положительного обхода? Затруднения возникают для самопересекающихся линий (рис. 32).

Линии, проведенные на поверхностях, делят на части эти поверхности, в свою очередь поверхности делят на части трехмерное пространство. Для чего заниматься такими специальными вопросами? Это нужно для решения простых и часто встречающихся задач: для того чтобы разобраться,

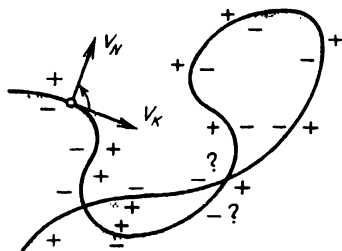


Рис. 32. Построение нормального вектора V_N путем вращения касательного V_K и деление плоскости на "положительные" и "отрицательные" области (показана неопределенность, возникающая для самопересекающихся линий)

какие части чертежа надо заштриховать; для определения принадлежности точки телу сложной детали; для выделения кусков поверхностей, которые вкуче ограничивают тело детали и т.д.

Кривые линии характеризуются также гладкостью и кривизной. Под гладкостью понимают отсутствие точек излома или возврата (рис. 33). Понятно, что, например, ломаная линия — негладкая. Кривизна — степень изогнутости кривой. Маленький кусочек кривой можно уподобить дуге окружности. Радиус этой окружности — это радиус кривизны дуги в данной точке, а обратная величина — значение кривизны (рис. 34). Кривизне и радиусу кривизны в ряде случаев присваивают знак (например, из соображений, в положительную или отрицательную сторону вращается ка-

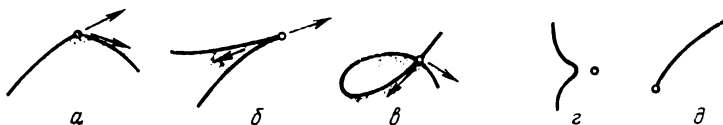


Рис. 33. Особые точки на кривой: а — точка излома; б — точка возврата; в — точка самопересечения; з — изолированная точка; д — точка прекращения

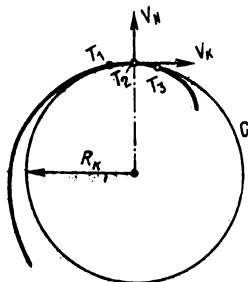


Рис. 34. Построение соприкасающейся окружности для некоторой точки кривой и определение радиуса кривизны в этой точке. R_K — радиус кривизны, V_N — нормальный вектор, V_K — касательный вектор, C — соприкасающаяся окружность; T_1, T_2, T_3 — три сближающиеся точки для ее построения

сательный вектор при движении по кривой точки в заданном направлении). Если построить две *плавно сопряженные* дуги окружностей разных радиусов, т.е. дуги, имеющие в точке стыка общую касательную, кривизна получившейся составной линии в этой точке будет меняться скачком. Глаз, вообще говоря, может замечать такие скачки, и линия в целом выглядит недостаточно плавной. Оказываются вредными скачки кривизны при проектировании трубопроводов, корпусов транспортных средств, профилей кулачков и в других случаях, связанных с механикой и аэродинамикой. Итак, иногда требуются не просто гладкие линии, но и такие, кривизна

Рис. 35. Построение нормального вектора и нормальных сечений в точке кривой поверхности. V_N – нормальный вектор, P – касательная плоскость, P_{N_1}, P_{N_2} – плоскости нормальных сечений, l_1, l_2 – нормальные сечения; m_1, m_2 – касательные к поверхности прямые

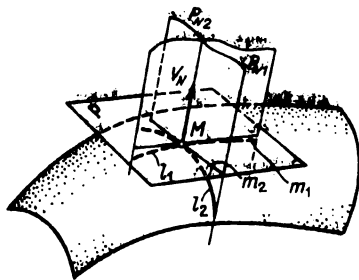
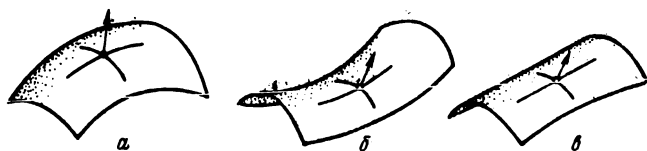


Рис. 36. Участки поверхностей с эллиптическими (а), гиперболическими (б) и параболическими (в) точками



вдоль которых изменяется постепенно, без скачков. С точки зрения дифференциальной геометрии это приводит к требованию, чтобы значения первой и второй производных изменялись непрерывно, т.е. были непрерывными функциями.

Сложнее вопросы гладкости и кривизны одной поверхности. Одного числа уже недостаточно, чтобы охарактеризовать кривизну поверхности в некоторой точке. Вообще говоря, нужны хотя бы три числа. Мы здесь не можем вдаваться в подробные объяснения; кое-что вкратце поясним с помощью рис. 35. В точке поверхности здесь построены касательная плоскость и нормаль к этой плоскости. Ближе лежащую часть поверхности можно рассекать плоскостями, проходящими через нормаль (нормальные сечения). Как правило, какое-либо из этих сечений имеет максимальное значение кривизны, а некоторое другое – минимальное значение (так называемые главные кривизны). Если обе эти кривизны одного знака, поверхность на данном участке подобна участку сферы или эллипсоида; она двояковыпуклая или, если ее рассматривать с другой стороны – двояковогнутая (рис. 36). Есть еще понятие *гауссовой кривизны*, равной произведению главных кривизн. Для таких точек гауссова кривизна положительна, а точки называют *эллиптическими*. Если главные кривизны разных знаков – гауссова кривизна отрицательная, а точки называются *гиперболическими*. Здесь поверхность имеет выпукло-вогнутый вид, напоподобие седла. Наконец, одна из главных кривизн может быть нулевой, гауссова кривизна тоже нулевая, точка называется *параболической*, а поверхность похожа на участок цилиндра. Сложные поверхности можно разделить на участки с эллиптическими и гиперболическими точками; параболические точки обычно выстраиваются в линии, отделяющие эллиптические и гиперболические участки друг от друга.

Теперь коснемся двух принципиально различных способов представления геометрических объектов в ЭВМ. Тот и другой способы – это *численные*

модели, дополненные соответствующими кодами или указаниями, к какому классу объект относится. Один способ – это так называемые *аналитические модели* (АМД). Аналитическая модель – набор чисел и, если надо, логических параметров, которые играют роль коэффициентов и других величин в уравнениях, аналитических соотношениях, задающих объект данного типа (сюда не относятся конечно, переменные-координаты). Поскольку один объект может быть описан различными уравнениями, и аналитические модели, в принципе, могут быть различны. Для удобства выбирают основную форму АМД, соответствующую по возможности простому и удобному заданию объекта. Для окружности основная форма аналитической модели – координаты центра и радиус. С использованием этих величин может быть написано неявное уравнение окружности

$$(x - x_C)^2 + (y - y_C)^2 - R^2 = 0.$$

Окружность, как и многие другие объекты, может быть задана также параметрическими уравнениями, где кроме координат есть еще одна переменная величина – параметр (*генерирующий* или *производящий параметр*). В данном случае это центральный угол α , определяющий положение точки на окружности:

$$x = x_C + R \cos \alpha,$$

$$y = y_C + R \sin \alpha.$$

Параметрическое задание образов широко применяется в машинной геометрии и графике. Уже для окружности виден ряд преимуществ такого задания. Всю окружность можно описать последовательными близко расположенными точками, изменяя генерирующий параметр α от 0 до 2π (если считать в радианах). Если значение α каждый раз наращивать на одну и ту же величину, расстояния между последовательными точками будут одинаковы. Представьте себе, как надо будет изменять координату-аргумент x , если взять за основу уравнение, явно разрешенное относительно координаты y ?

Для ограниченных образов – таких, как дуга окружности или дуга другой кривой – в АМД входят и параметры, выделяющие нужную часть. Для окружности это могут быть начальное и конечное значения углового параметра α ; правда, в этом случае дополнительно следует указывать направление обхода дуги – хотя бы с помощью знака, присваиваемого радиусу.

Второй способ численного моделирования геометрических объектов – это так называемые *координатные модели* (КМД). В простых случаях координатные модели – это наборы точек, принадлежащих объектам, точки же в свою очередь задаются координатами. Если иметь в виду кривые и ломаные линии – точки предполагаются расположенными в модели в том же порядке, что и на линии; такие КМД назовем упорядоченными. Упорядочение точек для поверхностей – более сложная задача, и она может решаться различным образом. В большинстве случаев точки, представляющие

поверхность, последовательно размещаются вдоль линий, проведенных на поверхности.

У координатных моделей могут быть разновидности. Так, вместо координат точки могут быть заданы разностями координат или, другими словами, проекциями соединяющих их векторов (координатно-разностные модели). Помимо координат (или их разностей) в каждой точке могут быть указаны дополнительные характеристики, например: проекции касательных векторов (для поверхностей – проекции нормальных векторов), значения каких-либо параметров (расширенные координатные модели). Координатные модели могут быть дополнены кодами, управляющими командами. Например, при описании нескольких ломаных или кривых линий заранее определенные коды, включенные в модель между координатами последовательных точек, могут означать конец одной линии и начало следующей. Код другого вида может означать конец всей модели. В зависимости от требуемой точности и других факторов координатные модели могут иметь различный объем. Ясно, что чем больше точек задано на кривой линии, тем точнее ее можно воспроизвести.

Многие графические устройства не имеют круговых интерполяторов, а имеют только линейные. Если это устройства траекторного типа, то все элементы изображения представляются фактически в виде координатных моделей, дополненных управляющими командами. Последовательно задаваемые в них "опорные точки" соединяются отрезками прямыми; таким образом, окружности, их дуги и другие кривые линии представляются в виде ломаных с мелкими звеньями. Вообще говоря, подобная картина наблюдается и у станков с программным управлением. Но здесь вопросы точности могут играть большую роль, чем при построении изображений, и для определения, на каком расстоянии можно брать точки в КМД, нередко используются специальные алгоритмы. Таким образом, общепринято, что кривые линии аппроксимируются ломаными: ниже мы подробнее объясним, что и кривые поверхности аппроксимируются наборами более простых поверхностей. Такая аппроксимация нужна не только при вводе или выводе графической и геометрической информации, но нередко используется и в геометрическом конструировании, при решении геометрических задач. Описанный принцип позволяет заменять разнообразные и сложные в аналитическом отношении объекты множествами простейших объектов (отрезки, кусочки плоскостей и др.). После такого "переворота" с различными объектами можно производить операции по одним и тем же алгоритмам. Приведем задачу определения пересечений кривых линий. Для аналитически сложных кривых решать эту задачу "прямыми вычислениями" не удастся. Известно, что даже для определения корней многочленов при степенях, превышающих четыре, приходится использовать приближенные и численные способы. Если разработан один алгоритм, определяющий пересечения двух ломаных линий, с его помощью приближенно можно определить пересечения линий, описываемых любыми управ-

нениями. Алгоритм же пересечения ломаных сводится к многократному повторению алгоритма пересечения двух отрезков (заметим, что два отрезка, принадлежащих двум ломаным, могут или пересекаться, или не пересекаться).

Для некоторых алгоритмов последовательные точки моделей предполагаются соединенными не отрезками, а дугами каких-либо простых кривых, например парабол. Это позволяет при сохранении требуемого уровня точности снизить число опорных точек в моделях.

Среди координатных моделей для объектов определенных типов можно выделить также *минимальные*. Это модели с минимальным количеством

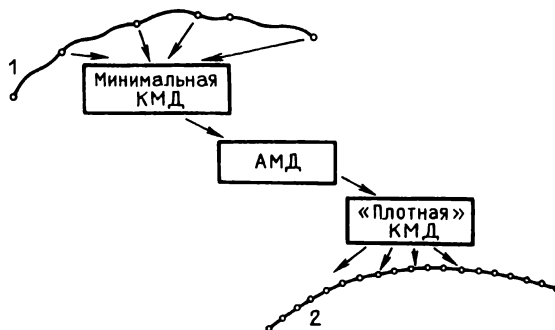


Рис. 37. Процесс построения математической кривой линии по снятым с исходной кривой данным: 1 – исходная кривая и выбранные на ней опорные точки, 2 – результирующая математическая кривая, построенная по "плотной" КМД

точек, способные однозначно определить объект данного типа. Так, для окружности минимальная координатная модель содержит три точки. От такой КМД с помощью соответствующих вычислений можно перейти к аналитической модели окружности (или другого объекта). Затем же, для дальнейших геометрических операций или для графического вывода объекта по его аналитической модели может быть синтезирована "плотная" координатная модель с большим количеством точек. Такая схема действий показана на рис. 37. В практике автоматизированного проектирования можно предположить ситуации, когда вручную нанесенные на эскиз кривые линии, например, контуры кузова автомобиля, измеряются по точкам. Таких точек может быть относительно немного, но, попав в машину, они служат исходными данными для построения проходящих через них математических кривых, близких по форме к исходным, но плавных и гладких, т.е. более совершенных, чем исходные линии. Затем на базе математически сглаженных кривых могут быть построены и закономерные, красивые (в том же смысле) поверхности.

Ниже будут рассматриваться и случаи, когда исходные объекты задаются *приближенными* координатными моделями. Предполагается, что в связи

с погрешностями измерений и другими факторами точки этих координатных моделей несколько смещены относительно из "правильного" положения. Здесь возникает задача *аппроксимации* – поиска такой линии или поверхности, которые бы проходили как можно ближе к заданным точкам.

Круг вопросов, решаемых с помощью машинной геометрии и графики, достаточно широк. Что-то в машинную геометрию и графику переходит от их "ручных" предшественников – начертательной и аналитической геометрии, но появляются и новые проблемы; начинают активно использоваться новые типы геометрических объектов, в том числе новые типы линий, поверхностей, которые ручными методами воспроизводить было трудно или невозможно.

2.3. Первые примеры кодирования изображений

Здесь читатель познакомится с кодированием простейших изображений. Постараемся параллельно описывать два способа задания изображения для ЭВМ – посредством составления программы на языке общего программирования, в качестве которого используется обычный фортран, и посредством записи кодировки на геометрически-ориентированном информационном языке алграф-Р. В обоих случаях выполнение программы в ЭВМ должно привести к одному результату – появлению на графопостроителе или экране графического терминала изображения. Для определенности в последующих объяснениях будем ориентироваться на построитель и его перо, предполагая, что практически программа (кодировка) не изменится при смене графического устройства. Параллельно познакомимся с численным моделированием таких простейших геометрических элементов, как отрезки, ломаные, окружности, их дуги, с процессом программного управления движениями перьев на построителе (или луча на экране); познакомимся с простейшими правилами записи информации на упомянутом языке алграф-Р.

Текст геометрического описания на этом языке в простых случаях состоит из слов-кодов и чисел, записываемых подряд, через запятые. Коды – заранее определенные слова, которые обрабатывающая система "понимает". В основном эти слова состоят из русских букв и цифр (первый символ в слове – всегда буква), означают типы задаваемых геометрических элементов (совместно со способами их определения), а также команды, управляющие системой и графическим устройством, названия геометрических операций и др. Числа записываются в обычном десятичном виде, перед отрицательными ставится знак минус, дробные десятичные разряды отделяются от целой части или незначащего нуля точкой (что типично и для других языков программирования). Текст пишется строками произвольной длины, но не более 72 символов (включая точки и запятые) в строке. Завершается текст знаком & (амперсанд). Вот пример текста, задающего одну окружность: "ОКР, 20, 50, 30, СТОП, &". Слово ОКР здесь – тип геометрического элемента, слово СТОП – управляющий

код, означающий конец чертежа, числа – геометрические параметры, задающие положение центра окружности и ее радиус. Некоторые коды бывают сложными, т.е. после основного кода (слова) в этих случаях в скобках записывается вторичный код – число или также какое-либо слово, а иногда и два числа (слова) через запятые. Например, код "ОКР (12)" означает двенадцатый вариант задания дуги окружности. Кроме слов-кодов в этом языке используются слова-метки. Эти слова могут содержать от одного до четырех символов. После метки ставится двоеточие, и таким образом метка помечает в тексте описания какое-либо место, число, код, группу чисел. В нужных случаях на метку ссылаются, записывая ее в скобках в качестве вторичного кода (уже без двоеточия). Метками также помечаются ячейки для хранения переменных значений, а в усложненном варианте языка соответствующие метки могут использоваться в арифметических выражениях примерно так же, как названия переменных в языках общего программирования.

В табл. 2 приведены примеры операторов на фортране и соответствующих управляющих кодов на языке алграф-Р, а в табл. 3 – коды некоторых геометрических элементов (наряду с операторами, задающими те же элементы на фортране).

Т а б л и ц а 2

Управляющие операторы и коды

Назначение оператора (кода)	Оператор на фортране	Код на языке алграф-Р
Включение пера	CALL PERO	П
Выключение пера	CALL VP	В
Смена номера используемого пера (можно задавать $n = 0 1 2 3$; если задать $n = 4$, дальнейшая смена перьев блокируется)	CALL NPERA (n)	НПЕРА (n),
Переход на черчение "основным" пером	CALL NPERA (0)	НПЕРА
Переход на черчение "тонким" пером	CALL NPERA (1)	НПЕРА 1
Переход на штриховые линии (аппаратная генерация)	CALL PUNC (1)	ПУНК1
Переход на штрих-пунктирные линии (аппаратная генерация)	CALL PUNC (2)	ПУНК2
Переход на сплошные линии	CALL PUNC (0)	СПЛОШ ПУНК (5),
Переход на штриховые линии (программная генерация, А и В – длины штрихов и пропусков)	CALL PUNC (5, А, В)	А, В, . . .
Переход на штрихпунктирные линии (программная генерация, А, В, С, D – длины четырех последовательных элементов линии – штрихов и пропусков)	CALL PUNC (6, А, В, С, D)	ПУНК (6), А, В, С, D,
Окончание черчения	CALL STP	СТОП

Рис. 38. Исходный эскиз (а) и его машинное изображение: частичное (б) и законченное (в)

Итак, поставлена задача — закодировать (на фортране и языке алграф-Р) несложное изображение в соответствии с эскизом, показанным на рис. 38, а.

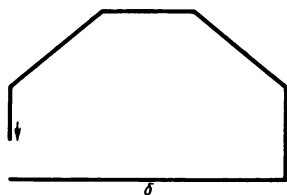
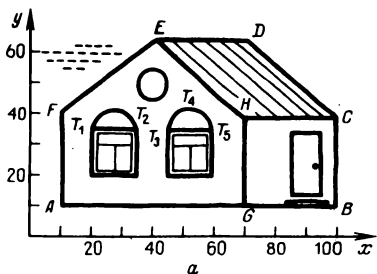
В начале работы графической программной системы ее надо "настроить". При работе с языком алграф-Р это делает обрабатывающая система, но при составлении программы на фортране для этого вызывается *иницилирующая подпрограмма*. Это типично для различных графических систем. В наших примерах мы будем пользоваться оператором CALL SBROS.

После срабатывания этой подпрограммы система будет "готова" чертить различные фигуры "основным" пером (если на графическом устройстве их несколько) сплошными линиями; линейные размеры будут восприниматься в миллиметрах, угловые — в градусах (в других системах могут быть приняты другие единицы измерения, например сантиметры, радианы). Начальное положение основного пера на построителе принимается за начало *координатной системы чертежа* (КСЧ). Перо обычно перед пуском построителя подводят к нужной точке с помощью кнопок автономного управления. Немного иначе определяется положение координатной системы на экране терминала. Иницилирующую подпрограмму в этом случае вызывают оператором

CALL SBROSE (XMIN, XMAX, YMIN, YMAX, NMAS)

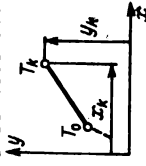
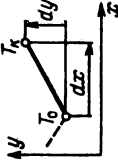
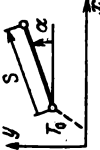
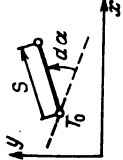
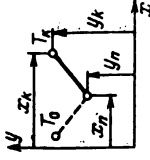
В качестве первых четырех параметров записываются минимальные и максимальные значения координат в координатной системе пользователя, отображаемые на рабочем поле экрана, в соответствии с чем определяются и масштаб изображения, и положение начала координатной системы на экране. Параметр NMAS задает (в байтах) длину области вывода в памяти ЭВМ, где перед пересылкой на терминал будет формироваться модель изображения.

Теперь можно задавать отрезки, ломаные и другие фигуры. Простейшее действие построителя (и луча векторного терминала) — это перемещение по прямой, от обработанной точки (где до того находилось перо) до вновь



Т а б л и ц а 3

Задачи геометрических элементов на плоскости

Геометрический элемент	Параметры, задающие элемент	Оператор на фортране	Код на языке алгефр-Р	Схема
Отрезок, проводимый от отработанной точки, или холостой ход (без управления пером)	Координаты конечной точки отрезка (или холостого хода)	CALL LINX (XK, YK)	ЛИН, xk, yk, ...	
" "	Проекция отрезка на координатные оси	CALL LINR (DX, DY)	ЛИНР, dx, dy, ...	
" "	Длина отрезка и угол его наклона к оси x	CALL LINS1 (S, ALF)	ЛИНС1, s, alpha, ...	
" "	Длина отрезка и угол его наклона к предыдущему вектору направления	CALL LINS2 (S, DALF)	ЛИНС2, s, dalpha, ...	
Отрезок как целая фигура (с управлением пером)	Координаты начальной и конечной точек	CALL OTREZ (XN, *YN, XK, YK)	ОТРЕЗ, xn, yn, xk, yk, ...	

Целая окружность (с управлением пером)	Координаты центра и радиус	CALL OK (XC, YC, R)	ОКР, xc, yc, r,	
Координаты центра и какой-либо точки окружности	Координаты центра и какой-либо точки окружности	CALL ОК6 (XC, YC, *XT, YT)	ОКР6, xc, yc, xt, yt	
Дуга окружности	Координаты центра, радиус; начальный центральный угол и угловая величина дуги со знаком	CALL ОК1 (XC, YC, R, *ALF, DA)	ОКР1, xc, yc, r, α, dα,	
Дуга от отработанной точки, касательная к предыдущей траектории	Координаты конечной точки	CALL ОК2 (XK, YK)	ОКР2, xk, yk, ...	
Дуга от отработанной точки	Компоненты вектора направления в начальной точке и координаты конечной точки	CALL ОК3 (DXN, DYN, *XK, YK)	ОКР3, dxn, dyn, xk, yk, ...	

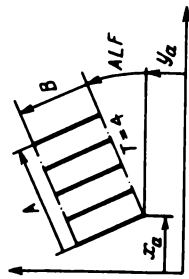
8 Таблица 3 (окончание)

Геометрический элемент	Параметры, задающие элемент	Оператор на фортране	Код на языке алграф-Р	Схема
Дуга окружности	Координаты начальной, промежуточной и конечной точек дуги	CALL OKT (XN, YN, *XT, YT, XK, YK)	ОКРТ, xp, yp, xt, yt, xk, yk,	
Эллипс частного положения	Координаты центра и длины двух полуосей (первая параллельна оси x)	CALL EL (XC, YC, *RX, RY)	ЭЛЛ, xc, yc, rx, ry, ...	
Эллипс	Координаты центра, длины полуосей, угол наклона на первой полуоси к оси x	CALL ELI (XC, YC, *RX, RY, ALF)	ЭЛЛ, xc, yc, rx, ry, α ,	
Прямоугольник со сторонами, параллельными координатным осям	Координаты одной вершины и длины отходящих от нее сторон со знаками	CALL PRM (XA, YA, *A, B)	ПРМК, xa, ya, a, b, ...	
Прямоугольник	Координаты вершины, длины отходящих от нее сторон, угол наклона первой стороны к оси x	CALL PRMUG (3, XA, *YA, A, B, ALF)	ПРМУГ, xa, ya, a, b, α , ...	

Серия равных равно-
расположенных штри-
хов, вписанных в пря-
моугольник

Координаты угла опи-
санного прямоугольни-
ка, длины его сторон,
число штрихов, умень-
шенное на 1 (штрихи
параллельны второй сто-
роне), угол наклона пер-
вой стороны

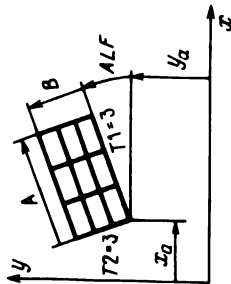
CALL SLA (1, XA,
*YA, A, B, T, ALF)



Прямоугольная сетка

Координаты угла и дли-
ны сторон внешнего
прямоугольника, число
делений первой и второй
сторон, угол наклона пер-
вой стороны к оси X

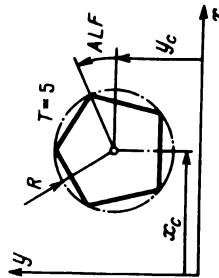
CALL SLA (3, XA,
*YA, A, B, T1, T2, ALF)



Правильный много-
угольник

Координаты центра, ра-
диус описанной окруж-
ности, угол наклона ра-
диуса, проведенного в
начальную вершину фи-
гуры, число сторон

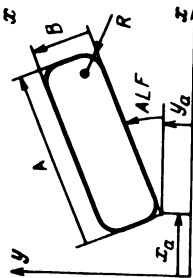
CALL SIB (12, XC,
*YC, R, ALF, T)



Прямоугольник со
скругленными углами

Координаты вершины
описанного прямоуголь-
ника, длины двух его
сторон, угол наклона
первой стороны, радиус
скругления

CALL SLB (7, XA,
*YA, A, B, ALF, R)



указанной точки. Другими словами, это линейная интерполяция, о которой говорилось выше. Если при этом перо было включено (луч "зажжен") – на изображении появится отрезок; в противном случае будет холостой ход. Новая точка может быть задана различными способами. Самый простой – ее координатами; другой способ – задать проекции вектора перемещения (использовать разности координат, т.е. относительные координаты). Еще один способ – задать длину вычерчиваемого отрезка (или холостого хода) и угол, под которым отрезок наклонен к оси x . Для таких различных способов задания отрезка на фортране вызываются различные подпрограммы (см. табл. 3). Численные параметры при их вызове определяются константами, переменными или арифметическими выражениями, выступающими в роли фактических параметров. На языке алграф-Р из таблицы выбирается код геометрического элемента и записывается в текст; он будет считаться действующим, пока его не сменит другой код геометрического элемента. После этого можно записывать подряд по два числа, каждая пара будет означать следующее перемещение. Нужно понимать, что задаются отрезки не общего положения, а частного вида (от отработанных точек). Действительно, для вычерчивания отдельно расположенного отрезка понадобятся два движения – холостой ход к его началу и черчение самого отрезка. Кроме того, описанные коды и подпрограммы сами не включают и не выключают пера; в нужные места программы или текста геометрического описания нужно включать соответствующие команды или коды из табл. 2.

Кодировать изображение (рис. 38) начнем с холостого хода пера в точку А, там перо следует включить и последовательно вычертить отрезки АВ, ВС, CD. Все эти перемещения задаются координатами конечных точек. В итоге начало программы будет такое:

```
CALL SBROS  
CALL LINA(10., 10.)  
CALL PERO  
CALL LINA (100., 10.)  
CALL LINA (100., 40.)  
CALL LINA (70., 65.)
```

На языке алграф-Р соответственно имеем

```
ЛИН, 10, 10, П, 100, 10, 100, 40, 70, 65,
```

Следующие отрезки зададим разностями координат и третьим из способов – длиной и углом наклона, после чего выключим перо:

```
CALL LINR (-30., 0.)  
CALL LINR (-30., -25.)  
CALL LINS1 (30., -90.)  
CALL VP
```

или: ЛИНР, -30, 0, -30, -25, ЛИНС1, 30, -90, В,

Когда чертятся ломаные линии, естественно задавать каждый отрезок от отработанной точки. В других случаях удобнее задавать отрезки полностью, каждый – четырьмя числами, координатами начальной и конечной точек, тем более что система будет сама включать перо в начальной точке и выключать в конечной точке каждого отрезка; надо лишь убедиться, что перед черчением первого отрезка перо выключено. Зададим таким способом отрезки EH, HG и HC (координаты точки H в данном случае придется повторить трижды):

CALL OTREZ (40., 65., 70., 40.)

CALL OTREZ (70., 40., 70., 10.)

CALL OTREZ (70., 40., 100., 40.)

или: ОТПЕЗ, 40, 65, 70, 40, 70, 40, 70, 10, 70, 40, 100, 40,

Дальше зададим целые фигуры – окружности и прямоугольники. Здесь система также самостоятельно включает и выключает перо. Для вычерчивания окружности проще задать ее аналитическую модель координатами центра и радиусом; для вычерчивания прямоугольников частного положения, у которых стороны параллельны координатным осям, задаются координаты одного из углов и длины двух сторон; стороны задаются "со знаками" в зависимости от их направления (первой задается длина стороны, параллельной оси x):

CALL ОК (40., 50., 5.)

CALL ОК (92.5, 23., 0.5)

CALL PRM (20., 20., 15., 15.)

или: ОКР, 40, 50, 5, 92.5, 23, 0.5, ПРМК, 20, 20, 15, 15,

Теперь поясним задание дуг окружностей. В нашем примере их всего две и их зададим разными способами. Кодирование первой (между точками T_1 и T_2) будет соответствовать ее аналитической модели в форме, принятой в системе Алграф. Первые три параметра – координаты центра и радиус опорной окружности, дальше идут центральный угол начальной точки дуги и угловая величина дуги со знаком, который определяется требуемым направлением обхода.

Напоминаем, что система Алграф "настроена" на градусную меру углов, хотя в нужный момент ее можно "перенастроить" на радианную или любую другую меру. Напомним также, что положительным направлением обхода считается направление против часовой стрелки. В нашем случае, поскольку дуга составляет половину окружности и должна чертиться от точки T_1 к точке T_2 по часовой стрелке, угловая длина ее будет составлять -180° . После вычерчивания дуги перо необходимо выключить специальной командой, так как в системе принято, что при черчении дуг перо автоматически включается в их начальных точках (если не включено было до того), но после завершения черчения само не выключается. Действительно, нередко

дуга продолжается другой дугой или отрезком. Итак:

CALL OK1(27.5, 35., 7.5, 180., -180.)

CALL VP

или: ОКР1, 27.5, 35, 7.5, 180, -180, В,

Другую дугу задаем координатами трех ее точек – начальной, некоторой промежуточной (не обязательно такой, которая делит дугу пополам) и конечной:

CALL ОКТ (45., 35., 52.5, 42.5, 60., 35.)

CALL VP

или: ОКРТ, 45, 35, 52.5, 42.5, 60, 35, В,

В системе Алграф более 15 различных способов задания дуг окружностей, что позволяет в конкретных случаях выбирать наиболее удобные. Все эти способы одинаково применимы как для аппаратной круговой интерполяции дуг (если такую возможность обеспечивает графическое устройство), так и для их программного построения. Когда дуги, как и целые окружности, чертятся по рассчитываемым точкам как ломаные с мелкими звеньями. Правда, круговым интерполятором не всегда можно пользоваться, так как при преобразовании координат (при построении проекций) окружности могут превращаться в эллипсы и другие фигуры, вычертить которые интерполятор неспособен. Поэтому в графической системе предусмотрено средство, "разрешающее" или "запрещающее" круговую интерполяцию. При вычерчивании окружностей и дуг по точкам частоту их расположения можно также регулировать в зависимости от требуемой точности чертежа и других факторов. Различные режимы, переменные, управляющие работой системы, значения координат и другие величины хранятся в блоке общих переменных с именем BENKOT. В своей программе или подпрограмме пользователь может описать этот блок и тем самым получить доступ к этим величинам:

COMMON/BENKOT/XOR, YOR, ZOR, XOS, YOS, ZOS, XPR, YPR, ZPR,
*XD, YD, AMATO(14), AMATI(14), AMAT2(14), IND(40), AAA(40), DXOR,
*DYOR, DZOR, ZPRI, RABOT(40)

Здесь XOR, YOR, ZOR, в частности – координаты последней отработанной точки чертежа в той системе, которая используется в данный момент; XD, YD – координаты отработанной точки в координатной системе чертежа, массивы IND и AAA содержат различные управляющие индексы и рабочие переменные. В частности, для включения и выключения пера вместо специальных подпрограмм можно использовать индекс IND(1), которому присваиваются значения 0 (перо выключить) и 1 (включить). Для "разрешения" круговой интерполяции следует задать IND(22)=1; приращение центрального угла дуги для перехода к расчету ее следующей точки (в радианах) хранится в элементе AAA(9), значение которого можно изменять.

Теперь, по ходу нашего примера, познакомимся с переключением инструмента. Предыдущие линии "по умолчанию" чертились "основным" пе-

ром (в системе основное перо имеет нулевой номер). Чтобы последующие линии чертить более тонким пером № 1 (нужные перья в соответствующие гнезда устанавливаются оператором) необходимо вызвать подпрограмму NPERA или вписать в кодировку специальный код (НПЕРА, НПЕРА1).

Можно изменять тип проводимых линий — со сплошных на штриховые или штрихпунктирные и обратно. Причем, как явствует из табл. 2, здесь также можно пользоваться аппаратной и программной генерациями штрихов. В обоих случаях после "включения" нужного типа линий пользователь не заботится о периодических выключениях пера для образования отдельных штрихов; выключение пера для холостых ходов осуществляется как обычно. Программная генерация штрихов бывает удобна и тем, что пользователь может задавать различные нужные ему длины штрихов и пропусков.

Завершить (логически) графическую программу надо с помощью завершающей подпрограммы STP или (на языке алграф-Р) кода СТОП. При этом завершается вывод информации на промежуточный носитель, а перо приводится в исходную точку чертежа (начало координатной системы). Такой возврат оказывается удобным, когда сложный чертеж выполняется "по частям", несколькими программами. При работе с графическим терминалом вместо завершающей подпрограммы вызывается подпрограмма EKРАН (или другая подобная ей), которая передает накопленную в памяти ЭВМ модель изображения терминалу и включает в нем регенерацию; на экране возникает изображение, программа же в ЭВМ при этом обычно останавливается и "ждет", пока пользователь просмотрит изображение и выполнит какое-то действие на терминале. Приведем текст программы для нашего примера и текст геометрического описания на языке алграф-Р. Сравните их!

```
CALL SBROS
CALL LINA (10., 10.)
CALL PERO
CALL LINA (100., 10.)
CALL LINA (100., 40.)
CALL LINA (70., 65.)
CALL LINR (-30., 0.)
CALL LINR (-30., -25.)
CALL LINSI (30., -90.)
CALL VP
CALL OTREZ (40., 65., 70., 40.)
CALL OTREZ (70., 40., 70., 10.)
CALL OTREZ (70., 40., 100., 40.)
CALL OK (40., 50., 5.)
CALL OK (92.5, 23., 0.5)
```

```

CALL PRM (20., 20., 15., 15.)
CALL PRM (45., 20., 15., 15.)
CALL PRM (85., 13., 10., 22.)
CALL PRM (84., 10., 12., 2.)
CALL OKI (27.5, 35., 7.5, 180., -180.)
CALL VP
CALL OKT (45., 35., 52.5, 42.5, 60., 35.)
CALL VP
CALL NPERA (I)
CALL PRM (21., 21., 13., 13.)
CALL PRM (46., 21., 13., 13.)
CALL OTREZ (21., 30., 34., 30.)
CALL OTREZ (27.5, 21., 27.5, 30.)
CALL OTREZ (46., 30., 59., 30.)
CALL OTREZ (52.5, 21., 52.5, 30.)
CALL PUNC (5, 3., 2.)
CALL OTREZ (10., 55., 20., 55.)
CALL OTREZ (5., 57., 25., 57.)
CALL OTREZ (2., 61., 30., 61.)
CALL OTREZ (15., 64., 27., 64.)
CALL STP
END

```

Текст описания:

ЛИН, 10, 10, П, 100, 10, 100, 40, 70, 65, ЛИНР, -30, 0, -30, -25, ЛИНС1, 30, -90, В, ОТРЕЗ, 40, 65, 70, 40, 70, 40, 70, 10, 70, 40, 100, 40, ОКР, 40, 50, 5, 92.5, 23.,5, ПРМК, 20, 20, 15, 15, 45, 20, 15, 15, 85, 13, 10, 22, 84, 10, 12, 2, ОКР1, 27.5, 35, 7.5, 180, -180, В, ОКРТ, 45, 35, 52.5, 42.5, 60, 35, В, НПЕРА1, ПРМК, 21, 21, 13, 13, 46, 21, 13, 13, ОТРЕЗ, 21, 30, 34, 30, 27.5, 21, 27.5, 30, 46, 30, 59, 30, 52.5, 21, 52.5, 30, ПУНК (5), 3, 2, 10, 55, 20, 55, 5, 57, 25, 57, 2, 61, 30, 61, 15, 64, 37, 64, СТОП, &

Программа на фортране составлена самым простым способом и потому получилась довольно длинной. Ее можно несколько сократить. Во-первых, для черчения двух одинаковых окон можно организовать цикл (подумайте, как). Во-вторых, для черчения ломаных в системе есть специальные подпрограммы. С их помощью наружный контур домика можно описать короче. Одна из подпрограмм, LOM, вызывается так: CALL LOM (A, NT, IM). Здесь A – имя массива, где располагаются геометрические параметры ломаной линии, NT (целочисленный параметр) задает количество опорных точек, IM – индекс модификации, он указывает способ задания каждого звена ломаной, ее размерность (ломаная на плоскости или в пространстве) и

является ли ломаная замкнутой. Для задания двумерных ломаных координатами опорных точек задается $IM = 2$, для трехмерных ломаных — $IM = 3$; замкнутые ломаные этих же типов (без повторного задания первой опорной точки) указываются значениями $IM = 102$ и $IM = 103$. Если каждое звено ломаной задается приращениями координат, для незамкнутых ломаных значения индекса будут 4 и 5, для замкнутых — 104 и 105 и т.д. Подпрограмма сама включает перо в начальной точке ломаной и выключает после вычерчивания. Параметр NT можно задавать "с запасом", но тогда описание в массиве A должно завершаться числом-кодом 10 000. (Предполагается, что столь больших значений координат не встретится.) Другой код (10001.), проставленный между параметрами последовательных точек (звеньев) в массиве A, означает временное выключение пера, т.е. холостой ход. С помощью кодов при одном обращении к подпрограмме можно расчертить множество ломаных (и кривых линий, представленных как ломаные с мелкими звеньями). Геометрические данные в массиве A можно загрузить разными способами, например через оператор DATA. Но для не очень сложных ломаных иногда удобнее подпрограмма LMNE, для которой не надо заводить отдельного массива; первым параметром при обращении к ней является индекс модификации IM, в качестве фактических параметров последовательно записываются геометрические данные, например координаты опорных точек. Можно использовать переменные, арифметические выражения (чего нельзя делать в операторе DATA). В нашем примере вместо девяти операторов для описания контура домика можно использовать один такой:

```
CALL LMNE (102, 10., 10., 100., 10., 100., 40., 70., 65., 40., 65., 10., 40.)
```

2.4. Типовая деталь

Написав программу вычерчивания типовой детали, можно использовать ее сотни раз — это и автоматизация, и эффективность. Здесь имеются в виду изображения типовых деталей — болтов, валов, подшипников, шестерен и т.д. Это могут быть рабочие чертежи в рамках, с пояснениями, размерами и дополнительными надписями, а могут быть фрагменты (типовые части), заготовленные для вычерчивания в сборочные чертежи. Принцип один: программа или подпрограмма чертит варианты изображения. Для каждого конкретного варианта она "настраивается" с помощью нескольких *независимых параметров*, которые читаются с перфокарт или передаются ей другим способом. Внутри программы (подпрограммы) помимо констант используются переменные и арифметические выражения. Различные размеры и координаты вычисляются с помощью независимых параметров. При этом в разных изображениях, формируемых программой, могут различаться не только размеры, но и топология чертежа — количество линий, их относительное расположение. Одна программа вполне может чертить болты и с шестигранной головкой, и с круглой, и с потайной. В число независимых па-

раметров могут быть для этого включены логические или целочисленные параметры, указывающие подтип детали, изображения.

С одной стороны, чем "уже", конкретнее программа, тем легче ею пользоваться, так как ей меньше надо задавать исходных параметров. С другой стороны, более универсальная программа будет чаще использоваться. Что лучше – зависит от конкретной ситуации.

Продемонстрируем эту методику на примере упрощенного изображения болта. Согласно рис. 39, а, пусть начало координатной системы размещается в точке, где ось болта пересекается с основанием головки. Как изобра-

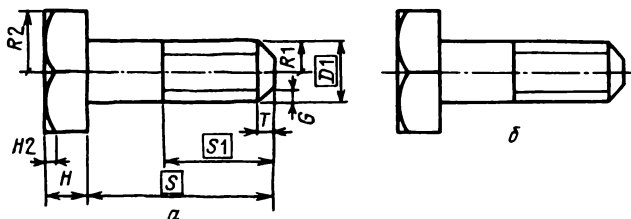


Рис. 39. Исходный эскиз (а) и машинное изображение детали (б)

жать такой болт в разных местах чертежа и в разной ориентации, не изменяя программы (подпрограммы), будет показано ниже. Независимые параметры на эскизе обозначены в квадратиках, значения остальных параметров подсчитываются по простым формулам. Составим на фортране программу, которая с перфокарты прочитает независимые параметры D1, S, S1, G, затем вычислит другие параметры и размеры, после чего с вызова иницирующей подпрограммы SBROS перейдет к графической части. Подпрограммы в основном те же, которые обсуждались выше, но геометрические параметры заданы не константами. Варианты машинных изображений показаны на рис. 39, текст программы приводится ниже.

```

READ 99, D1,S,S1,G
99 FORMAT (4F10.0)
R1=D1/2.
R2=D1
H=D1*0.7
H2=H*0.2
T=G*1.5
X1=S-S1
X2=S-T
X3=H-H2
CALL SBROS
CALL PRM (-H,-R2,H,R2*2.)
CALL OKT (-X3,-R2,-H,-R1,-X3,0.)

```

```

CALL OKT (-X3.,0.,-H,R1,-X3,R2)
CALL VP
CALL OTREZ (-X3,0.,0.,0.)
CALL LMNE(2,0.,R1,X2,R1,S,R1-T,S,T-R1,X2,-R1,0.,-R1,10001.,X1,-R1,
*X1,R1,10001.,X2,R1,X2,-R1)
CALL NPERA (1)
CALL OTREZ (X1,R1-G,X2,R1-G)
CALL OTREZ (X2,G-R1,X1,G-R1)
CALL PUNC (2)
CALL OTREZ (-H-5.,0.,S+5.,0.)
CALL STP
END

```

Составить геометрическое описание можно и на информационном языке типа алграф-Р, хотя информация приобретает при этом явные черты вычислительной программы. Польза от этого может быть существенная: вместо хранения в базе данных или библиотеке сотен и тысяч описаний конкретных деталей "жесткой" формы, может храниться значительно меньшее число описаний разных типов деталей.

Чтобы привести пример описания на языке алграф-Р, поясним, как на нем задаются арифметические выражения. Признаком такого выражения является знак равенства. Если слева от него стоит имя метки (без двоеточия) — результат выражения должен быть послан по адресу, указанному меткой. Если равенство стоит сразу после разделительной запятой, выражение будет воспринято как очередной геометрический параметр, например как координата следующей точки или радиус окружности. Сами выражения примерно такие же, как в фортране; имена меток играют роль переменных, скобки изменяют последовательность действий, есть даже ряд функций (тригонометрические, извлечение корня и др.). Есть и некоторые отличия. Так, если скобки не использовать, все арифметические действия будут выполняться слева направо, без различия их "старшинства". Помеченные метками элементы, играющие роль переменных, обычно располагаются после оператора СТОП.

Для изображения тех же болтов, которые мы формировали с помощью программы на фортране, текст описания может быть таким:

```

R1=D1/2, T=G*1.5, H=D1*0.7,X1=S-S1,
ПМК,=-H,=-D1,=H,=D1+D1,
OKT,=-H*0.8,=-D1,=-H,=-R1,=-H*0.8,0,
=-H*0.8,0,=-H,=R1,=-H*0.8,=D1,B,
OTREZ,=-H*0.8,0,0,0,ЛИН,0,=R1,П,=S-T,=R1,
=S,=R1-T,=S,=T-R1,=S-T=-R1,0,=-R1,B,
=X1,=-R1,П,=X1,=R1.B,=S-T,=R1,=S-T,=-R1,B,

```

НПЕРА 1, =X1,=R1-G,=S-T,=R1-G,=S-T,=G-R1,=X1,=G-R1,
 ПУНК 2,=-H-5,0,=S+5,0,СТОП,
 D1: ?????, S: ?????,S1: ?????,G: ?????,
 R1:, T:, H:, X1:,
 &

Будем считать, что, увидев этот текст на экране терминала, оператор заметит знаки вопросов, стоящие после меток, на конкретные числовые значения и нажмет нужную кнопку для получения изображения.

2.5. Узоры

Для проверки работы графических устройств и демонстрации их возможностей (особенно когда устройства экспонируются на выставках) составляют программы вычерчивания каких-либо красивых узоров из пересекающихся линий, с "муаровым эффектом" и т.д. Такие программы обычно несложны — важны выдумка, декоративный эффект. Машинные узоры используют для рекламы, в качестве заставок в книгах и журналах, декоративных рисунков, наносимых на ткань, посуду и т.д. В некоторых странах организуются выставки художественной машинной графики.

На рис. 40 изображены звездообразные фигуры. Присмотревшись, можно обнаружить, что они составлены из одного повторяющегося в разных направлениях и размерах мотива — "угла, заштрихованного отрезками": задаются три точки, они соединяются двумя отрезками; каждый из отрезков надо разделить на равное число одинаковых частей и затем точки с одинаковыми номерами соединить по порядку другими отрезками. Множество этих отрезков, пересекаясь, образуют криволинейную видимую границу — *огibaющую семейства* *) (ее, вообще говоря, можно найти математически, и это будет дуга параболы). Для одного такого "угла" можно подготовить подпрограмму, а в главной программе вызвать ее нужное число раз, передавая в качестве фактических параметров координаты трех точек угла и число делений каждой из сторон. Вот текст подпрограммы:

```

SUBROUTINE UGOL (X1,Y1,X2,Y2,X3,Y3,N)
  N1=N+1
  DO 1 K=1,N1
    A = ( K -1.)/N
  1 CALL OTREZ (X1 +(X2-X1)*A, Y1 +(Y2 -Y1)*A,
    * X2 +(X3-X2)*A, Y2 +(Y3 -Y2)*A)
  RETURN
END

```

*) В общем случае огibaющей семейства прямых называют линию, в каждой своей точке касающуюся одной из прямых семейства.

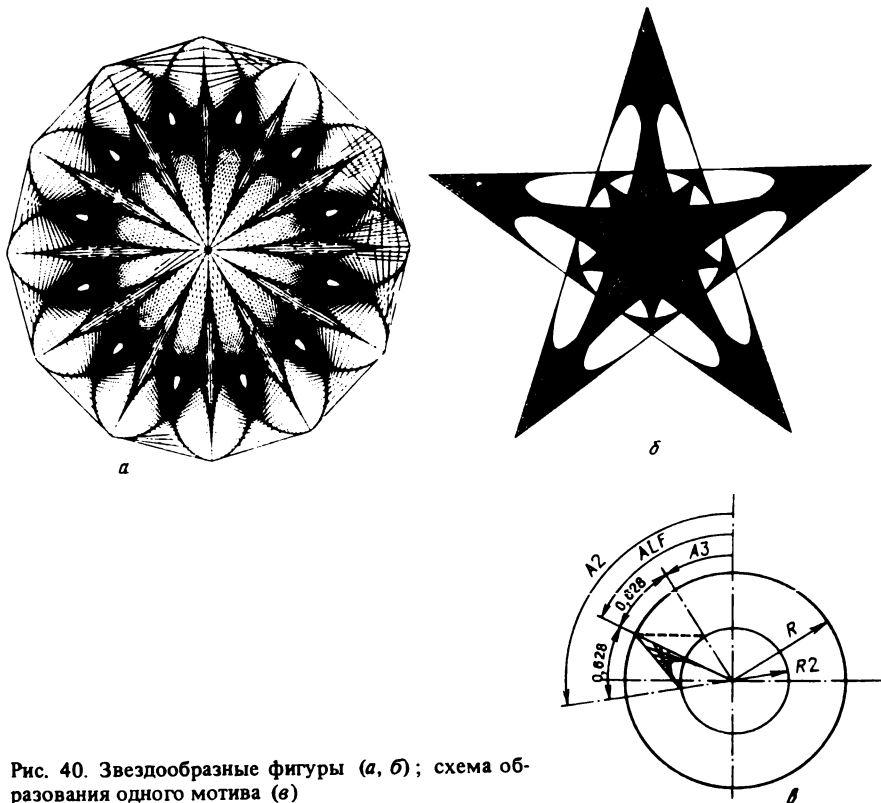


Рис. 40. Звездообразные фигуры (а, б); схема образования одного мотива (в)

Подпрограмма чертит отрезки, включая и стороны заданного угла, поэтому количество отрезков на один больше числа делений сторон угла. Подумайте, как усовершенствовать программу, чтобы холостые ходы при черчении последовательных отрезков были небольшими.

Приведем фрагмент главной программы, в котором чертятся "углы", образующие пятиконечную звезду в соответствии со схемой, показанной на рис. 40, в:

```

DATA R/120./, R2/45./
DO 1 J=1,5
AL=6.2832*(J-1)/5
A2=AL+0.6283
A3=AL-0.6283
CALL UGOL (0.,0., R*SIN(AL),R*COS(AL),R2*SIN(A2),(R2*COS(A2),25)
1 CALL UGOL (0.,0., R*SIN(AL), R*COS (AL), R*SIN(A3), R2
* COS(A3), 25). . .

```

Приведем другой прием, где также возникает иллюзия кривых линий. Берем правильный (или неправильный) многоугольник – треугольник, квадрат, ромб и т.д. Задаемся некоторым отношением $A = \gamma/p$ и делим каждую сторону многоугольника в этом отношении (p – длина стороны, γ – длина ее части, отделяемой искомой точкой). Поскольку в программе мы используем координаты точек, вспоминаем, что и каждая из проекций стороны разделится в том же отношении (так же мы поступали и в подпрограмме UGOL). Полученные на сторонах точки соединяем отрезками – и получается новый многоугольник, вписанный в предыдущий. Эту операцию

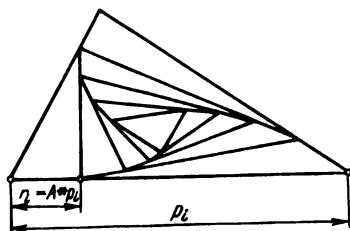


Рис. 41. Схема построения вписанных многоугольников

можно повторять "рекурсивно" много раз, получая постепенно уменьшающиеся многоугольники (рис. 41). Их углы визуально образуют закручивающиеся (спиральные) линии. Из таких ячеек можно получить занятные узоры.

На рис. 42 мы видим с "муаровым эффектом". Эффект возникает при пересечении двух семейств прямых или кривых линий, когда эти линии пересекаются под небольшими углами. Близкие друг к другу точки пересечения последовательных пар линий выстраиваются в изогнутые полосы разной ширины. Эти полосы бледнее соседних участков чертежа, потому что краска от одной линии частично или полностью ложится на краску от другой линии. Муаровыми узорами пользуются при изучении деформаций материала. Допустим, на деталь нанесен рисунок из частых параллельных штрихов, он фотографируется на одну пленку до деформации и после нее. Если при деформации линии штрихов несколько сместились и искривились – на пленке возникнет муаровый узор, и по нему можно судить о величине и направлении деформации (подумайте, как).

В простейшем случае для получения муарового узора можно начертить два семейства расходящихся отрезков (рис. 43). Текст программы очень прост:

```
CALL SBROS
DO 1 K=1, 51
Y1=(K-26)*2.
Y2=(K-26)*3.
CALL OTREZ (-100., Y1, 100., Y2)
```

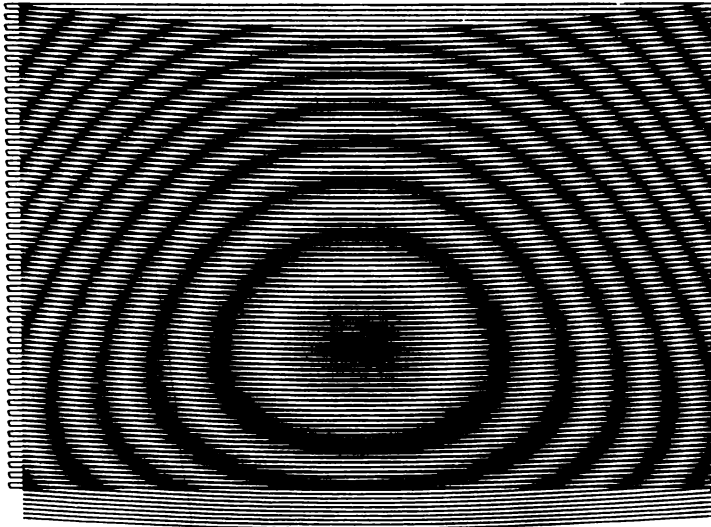


Рис. 42. Узор с "муаровым эффектом"

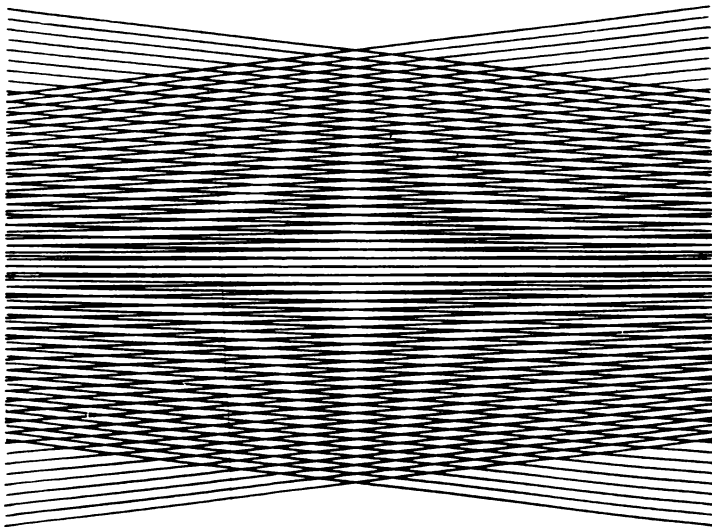


Рис. 43. Изображение расходящихся отрезков

```

1 CALL OTREZ (100., Y1., -100., Y2)
  CALL STP
  END

```

Любопытные фигуры можно образовать и из окружностей. Вот пример программы, вычерчивающей семейство окружностей:

```

DATA R/ 30. / , N/ 30/
CALL SBROS
N1=N-1
DO 1 K=1, N1
  ALF=6.283 * K/N + 1.5707
1 CALL OK6 (R * COS (ALF), R * SIN(ALF), 0.,R)
  CALL STP
  END

```

На рис. 44 виден результат – фигура, похожая на яблоко. Как образовано это семейство? Берется базовая окружность (подумайте, где она проходит), делится на равные части, и полученные точки принимаются за центры вычерчиваемых окружностей. Кроме того, все эти окружности проходят через одну из точек, взятую на опорной окружности. Огибающая полученного семейства окружностей – так называемая кардиоида – линия, похожая на сердце. Подумайте, что будет, если общую точку взять не на опорной окружности.

Ниже читатель познакомится с другими приемами образования машинных узоров и орнаментов.

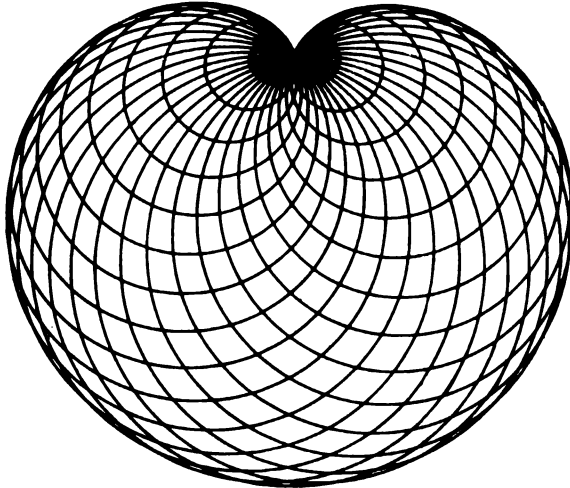


Рис. 44. Кардиоида из пересекающихся окружностей

ФУНКЦИИ И КРИВЫЕ ЛИНИИ**3.1. Вычерчивание кривых линий и графиков функций, задаваемых уравнениями в явном виде**

Автоматическое получение графиков функций было едва ли не первым практическим применением машинной графики. Ведь и слово "графопостроитель" буквально значит "построитель графиков". В физике, химии, экономике, многих других отраслях используется огромное множество функций и зависимостей, описывающих некоторые процессы, и очень часто их графическое представление оказывается полезным. Обычно графики функций строят в декартовых координатах, но иногда предпочитают полярные, логарифмические и некоторые другие. В этом разделе будут использоваться обычные прямоугольные координаты.

Графики функции обычно представляют собой кривые линии.

Поставим обратный вопрос: всякая ли кривая линия будет графиком некоторой функции? Стоит ли считать функцией окружность, если это контур детали или граница отверстия в ней? Функции обычно рассматривают однозначные, т.е. одному значению аргумента соответствует одно значение функции. Но этим свойством не обладает даже окружность. С другой стороны, и окружность, и различные другие линии имеют свое аналитическое выражение, их уравнения записываются в различных формах. Нетрудно записать уравнение, например, окружности и в явном виде, разрешив его относительно любой из координат. Однако неоднозначность решения здесь проявится в виде необходимости выбора того или иного знака перед радикалом. Конечно, окружность или другую неоднозначную линию можно разбить на "однозначные" части и считать, что эти части описаны разными функциями, но уже упоминалось, что для конструктивных целей предпочитают пользоваться параметрическими уравнениями. В такой постановке, вообще говоря, самые различные линии можно считать "функциями" параметра-аргумента, а аргумент сопоставить со временем. Но мы в данном пункте займемся все же "обычными" функциями и за аргумент будем принимать координату x . Графопостроители и другие графические устройства вычерчивают различные кривые обычно как ломаные с мелкими звеньями (круговые интерполяторы в этих случаях используются редко).

Если линия строится с помощью явного уравнения, координате-аргументу придают постепенно увеличивающиеся или уменьшающиеся значения и всякий раз подсчитывают значение координаты-функции; найденные точки остаются соединить отрезками. Правда, иногда для аргумента или функции вводят масштабные множители, чтобы график получился приемлемой величины. Вопрос, на каком расстоянии брать точки, чтобы их было не слишком много, а с другой стороны – чтобы нужная точность была выдержана, решается разными способами. Есть, например, программа, которая позволяет сначала просчитать большое количество точек, а затем отбросить некоторые так, чтобы получившаяся ломаная с более крупными звеньями удалялась от отброшенных точек не больше чем на заданную величину. В примерах, которые будут рассмотрены ниже, шаг по координате-аргументу будет назначать программист.

При нанесении на чертеж нескольких кривых их иногда помечают так, чтобы было легче отличить их друг от друга. Для этого можно пользоваться разными перьями, линиями разных типов (сплошные, штриховые и др.). Иногда линии помечают какими-либо значками. В нескольких последующих примерах будут начерчены линии сплошные и штриховые, с различными длинами штрихов.

Еще два небольших замечания. В качестве исходных данных для получения конкретной дуги линии определенного типа будем задавать аналитическую модель. В ее составе – начальное и конечное значения координаты-аргумента (x). Не оговаривая, которое из них больше, приращение (шаг) для аргумента мы всегда задаем положительным; программа же должна определять, с каким знаком следует брать этот шаг, чтобы линия могла чертиться как в направлении возрастания, так и убывания значений x . Кроме того, линия должна заканчиваться точно в заданной конечной точке, хотя эта точка может и не соответствовать очередной точке, получаемой прибавлением шага к значению аргумента.

С учетом этих замечаний составим алгоритм черчения какой-либо пространственной кривой, хотя бы синусоиды, а также подпрограмму на фортране. Уравнение синусоиды берем не в простейшем виде, а "вписываем" в него линейные преобразования аргумента и функции, чтобы линию можно было чертить "на различной высоте" и в разном масштабе:

$$Y = Y_0 + A * \text{SIN} ((X - X_0) * 6.2832/S)$$

Здесь X_0 и Y_0 – координаты нулевой точки (это не обязательно начальная точка вычерчиваемой дуги), A – амплитуда, S – длина волны. Схему алгоритма и подпрограмму располагаем "рядом", чтобы легче было видеть их соответствие (рис. 45).

Для сравнения напомним фрагмент геометрического описания на языке алгебра-Р, хотя не все особенности этой записи будут пока понятны:

$$\text{НАЧ:} M = \text{SIGN} (X_2 - X_1), X = X_1, K = 0, M1: Y = \text{SIN} (X - X_0 * 6.283/S) * A + Y_0, \text{ЛИН, } X:, Y:, P, \text{УСЛБ} (K, M2), X = M * DX + X, P = X_2 -$$

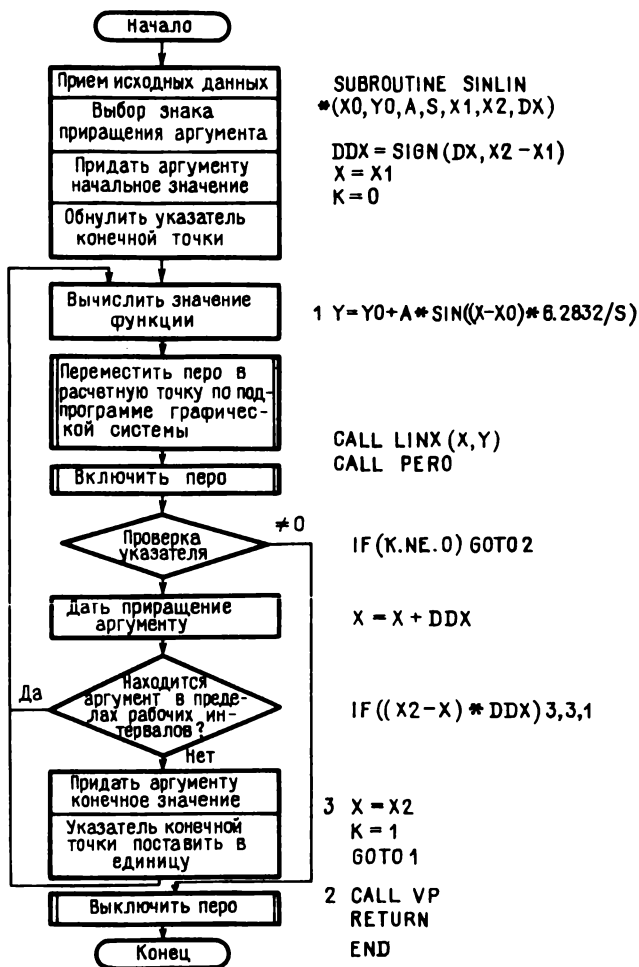


Рис. 45

X* M,УСЛБ (P,M1), K=1, X=X2,ИДТИ (M1), M2: В, КОН:, ХО:, УО:, А:, S:, X1:, X2:, DX:, K:, P: M:,

Подпрограмма SINLIN может вызываться с различными значениями параметров, как в приводимой ниже программе, по которой получен рис. 46:

CALL SBROS

CALL SINLIN(10., 30., 25., 180., 10., 100., 1.)

CALL PUNC(5, 8., 1.)

CALL SINLIN(-80., 30., 25., 180., 10., 100., 1.)

```

CALL PUNC (5,5., 1.)
CALL SINLIN (10., 30., 10., 60., 10., 180., 1.)
CALL PUNC (5,3., 1.)
CALL SINLIN (10.-15/4., 30., 7., 15., 10., 180., 1.)
CALL STP
END

```

По образцу этой подпрограммы можно написать много других. Фактически для новой функции придется изменить одну строку да заголовок подпрограммы. Можно попробовать упростить работу по подготовке черчения новых типов линий. Составим одну "постоянную" подпрограмму

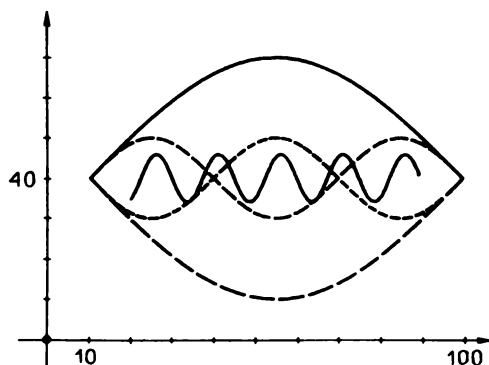


Рис. 46. Машинное изображение синусоид

черчения линий, заданных явными уравнениями, но пусть она сама обращается к той или иной подпрограмме-функции, имя которой ей сообщается в операторе вызова. Подпрограмма-функция должна только вычислить значение функции для заданного ей значения аргумента и потому может состоять всего из двух-трех строк. Чтобы была однотипность в передаче параметров различных линий аналитической модели, будем их записывать в один массив; имя массива передается подпрограмме черчения линии и далее транзитом — подпрограмме-функции. Подобная подпрограмма черчения линии в системе Алграф вызывается оператором

```
CALL CLINE (FU, X1, X2, DX, AMD, IM)
```

Здесь FU — имя подпрограммы-функции, которое в вызывающей программе должно быть объявлено как EXTERNAL; X1 и X2 — начальное и конечное значения координаты-аргумента, DX — приращение аргумента (для расчета каждой следующей точки), AMD — имя массива, содержащее аналитическую модель линии и передаваемое подпрограмме-функции; индекс IM задается нулевым, если за аргумент принимается x , и равным единице, если аргументом служит координата y . Эта подпрограмма дополнительно реагирует на значения функции, являющиеся числами-кодами (10001). Если функция "выдает" такое значение, считается, что правильного значения опреде-

лить не удалось; подпрограмма CLINE будет считать соответствующий участок линии отсутствующими. Все подпрограммы-функции, вызываемые в CLINE должны иметь заголовки одного типа (хотя и разные названия): FUNCTION FU(X, AMD), где X – значение аргумента, AMD – имя массива, содержащего аналитическую модель линии. Вот текст подпрограммы-функции для линий полиномов:

```

FUNCTION POLFU(X, AMD)
  DIMENSION AMD (1)
  K=AMD(1)
  POLFU=AMD(2)
  DO 1 M=1, K
1 POLFU=POLFU*X+AMD (M+2)
  RETURN
  END

```

Здесь первый элемент массива AMD – степень полинома (в виде действительного числа), следующие элементы – коэффициенты полинома, начиная со "старшего" и кончая свободным членом. Значение функции здесь подсчитывается по схеме Горнера. На рис. 47 показаны линии, полученные по этой подпрограмме. Попробуйте определить, какие аналитические модели были заданы для них.

Задавая коэффициенты полинома, мы не всегда заранее знаем, как пройдет линия. Но можно составить подпрограмму-функцию, которая в качестве аналитической модели "потребляет" более удобные в конструктивном отношении параметры. Вот подпрограмма-функция для вычисления точки дуги полинома 3-й степени:

```

FUNCTION PO3FU (X, A)
  DIMENSION A(6)
  XX=X-A(1)
  DX=A(4)-A(1)
  DY=A(5)-A(2)
  DK=A(3)+A(6)
  EK=DY/DX
  F=(3.*EK-DK-A(3))/DX
  G=(DK-EK-EK)/DX**2
  PO3FU=A(2)+XX*(A(3)
  * +XX*(F+XX+G))
  RETURN
  END

```

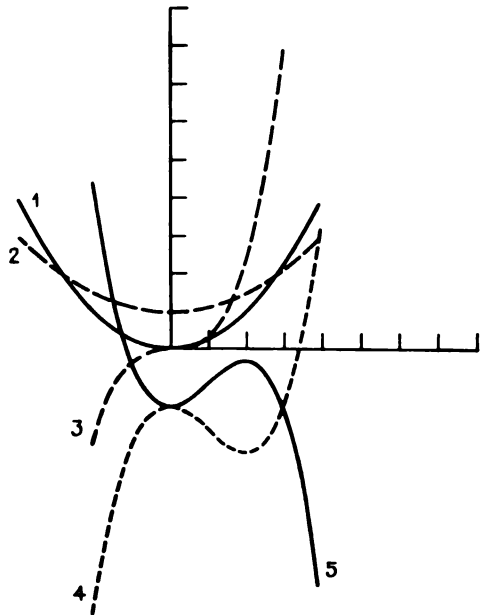


Рис. 47. Изображение линий полиномов

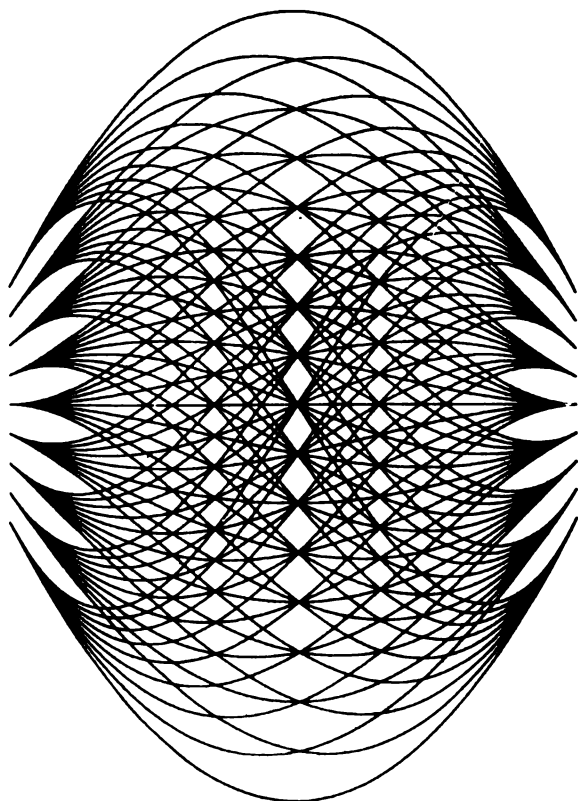


Рис. 48. Орнамент из дуг полиномов 3-й степени

В массиве А здесь заданы три параметра для первой и три параметра для второй определяющих точек дуги. Параметры такие – координаты точки и угловой коэффициент касательной прямой, иными словами – значение первой производной функции. Определяющие точки могут быть начальной и конечной точками вычерчиваемой дуги, хотя это и не обязательно (для подпрограммы CLINE координаты граничных точек дуги задаются отдельно). Любители математических кроссвордов могут по тексту приведенной подпрограммы-функции проверить, пройдет ли дуга через заданные точки и в нужном направлении!

На рис. 48 с помощью этой подпрограммы, подпрограммы CLINE и специально написанной главной программы вычерчен целый орнамент из таких дуг. Принцип построения орнамента ясен из рисунка, можно только уточнить, что угловые коэффициенты касательных в определяющих (и одновременно граничных) точках дуг линейно зависят от координат у этих точек.

Попробуйте расшифровать, почему на рисунке получились вертикальные прямые из "узелков" – точек пересечения дуг, и как они расположены.

Описанные выше программы и подпрограммы действуют в соответствии с концепцией "вычисли и сразу выдавай". Подсчитанная точка сразу передается графической системе и выводится на промежуточный носитель или на графическое устройство. Таким путем можно генерировать тысячи и десятки тысяч точек, не переполняя памяти ЭВМ. Но иногда требуется другая стратегия: вычисляя точки кривой, накопить их координаты в специальном массиве, чтобы затем использовать полученную координатную модель для различных целей; на чертеж ее можно вывести, например, с помощью подпрограммы LOM. Приведем текст подпрограммы CLINE2, которая накапливает КМД в двумерном массиве XY; первый индекс массива принимает значения 1 и 2 соответственно номеру координаты. В качестве выходного параметра также выдается число накопленных в массиве точек.

```
SUBROUTINE CLINE2 (FU, X1, X2, DX, AMD, XY, N)
  DIMENSION XY(2, 1000), AMD(1)
  DDX=SIGN(DX, X2-X1)
  X=X1
  K=0
  N=1
1 XY(1, N)=X
  XY(2, N)=FU(X, AMD)
  N=N+1
  IF (K.NE.0) GOTO 2
  X=X+DDX
  IF ((X2-X1) * DDX) 3,3,1
3 X=X2
  K=1
  GOTO 1
2 N=N-1
  RETURN
END
```

Из накопленной координатной модели можно выделить минимальное и максимальное значения координаты y и по ним подсчитать такой масштабный множитель для этой координаты, чтобы график уложился в выделенные ему пределы.

Если различные участки кривых имеют сильно различающиеся наклоны, для улучшения расположения опорных точек на линии иногда усложняют подпрограммы черчения линий, вводят автоматическую регулировку шага – приращения координаты-аргумента. Это приращение можно выбирать в зависимости от значения производной на данном участке (чем больше по абсолютному значению производная, тем меньше шаг). Чтобы не вычислять для каждой точки производную, в качестве ее приближенного значения можно принять разность координат y для двух последних вычис-

ленных точек, деленную на разность координат x , и использовать соотношение $DX_N = DS * DX / \sqrt{DX^2 + DY^2}$, где DS – расчетное расстояние между точками линии. При резком изменении поведения линии этот алгоритм может дать недостаточно хороший результат. Впрочем, иногда шагом управляют еще проще: после расчета очередной точки анализируют разность координат y этой точки и предыдущей. Если эта разность слишком велика, шаг уменьшают и точку пересчитывают заново. В более сложных алгоритмах при определении шага учитывают направление и кривизну линии, ведь на сильно изогнутых участках для хорошего отображения линии точки должны располагаться чаще.

3.2. Построение параметрических кривых

Выше мы упоминали, что в двумерном пространстве линии могут быть заданы параметрически – двумя уравнениями, явно выражающими обе координаты через значение некоторого производящего параметра:

$$x = f(t), \quad y = \varphi(t).$$

Параметрические линии по форме могут быть гораздо более разнообразными, чем описываемые практически приемлемыми явными уравнениями. Они могут быть самопересекающимися, содержать точки с вертикальными касательными, не имеющими ограничений по "многозначности". Одну и ту же линию, в принципе, можно параметрически описать по-разному. Среди различных способов "параметризации" можно выделить несколько характерных. Например, параметризация может быть такой, чтобы при равномерном изменении параметра текущая точка с постоянной скоростью двигалась по кривой. Линию, заданную явным уравнением, формально превратить в параметрически заданную несложно: достаточно ввести уравнение $x = t$, а в явном уравнении аргумент x заменить на параметр t . Обратная задача решается не всегда просто: надо из системы двух уравнений, описывающих линию, исключить параметр t , в этом случае обычно сначала получается неявное уравнение кривой. Некоторые задачи с параметрическими линиями решаются сложнее, чем с обычными. Так, если задана одна из координат точки, принадлежащей линии, определить другую координату бывает непросто. Предварительно из одного параметрического уравнения надо определить значение параметра для этой точки, а затем уже можно воспользоваться другим уравнением.

Чтобы параметрическое представление линии не показалось слишком сложным, заметим, что линия, в принципе, может быть задана, например, двумя неявными уравнениями, содержащими (каждое) две координаты и параметр:

$$F(x, y, t) = 0,$$

$$G(x, y, t) = 0.$$

Приведем пространственную интерпретацию двумерной параметрической линии (рис. 49). Если параметр t "превратить" в третью координату, два

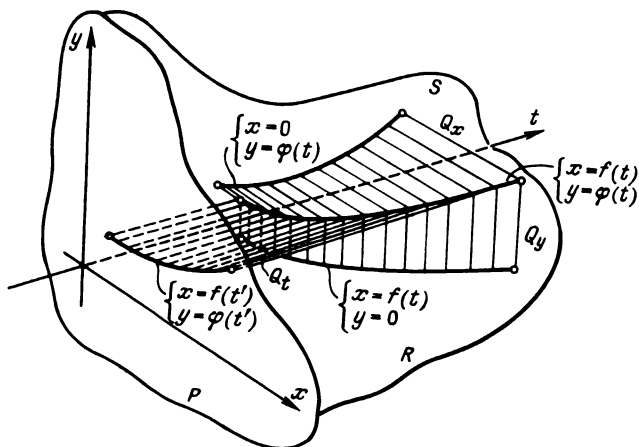


Рис. 49. Пространственная схема образования параметрической линии: P – плоскость чертежа, R, S – вспомогательные плоскости в искусственно введенном трехмерном пространстве, Q_x, Q_y – линейчатые проецирующие поверхности, пересечение которых образует пространственную линию, Q_t – проецирующая поверхность, образующая на плоскости P параметрическую линию (t – уже параметр, а не координата)

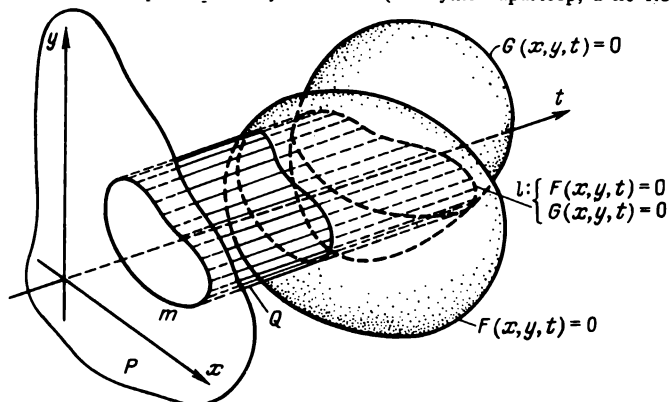


Рис. 50. Образование параметрической линии с помощью двух пересекающихся поверхностей: l – линия пересечения поверхностей в искусственно введенном пространстве, Q – проецирующая поверхность, m – проекция линии пересечения на плоскость чертежа (параметрическая кривая)

наших уравнения опишут пространственную кривую, а нужная нам двумерная линия – это проекция пространственной кривой на плоскость $t = \text{const}$ (или $t = 0$). Вышеприведенное задание линии в пространственной модели соответствует двум поверхностям, заданным неявными уравнениями. Поверхности пересекаются, порождая пространственную кривую (рис. 50), которая в двумерном пространстве имеет проекцию на плоскость $t = \text{const}$.

Параметрическая линия определена полнее, чем обычная. Ведь каждая ее точка кроме координат "снабжена" значением параметра. Этим путем мы можем различать даже совпадающие, т.е. наложенные друг на друга, линии. Если параметр сопоставить с временем, линию можно считать траекторией движения точки (хотя бы конца пера). Отношение длины дуги к разности значений параметра на ее концах будет скоростью (средней); скорость в точке линии можно записать, пользуясь терминами дифференциальной геометрии: $v = ds/dt$. Величины $v_x = dx/dt$ и $v_y = dy/dt$ – проекции вектора скорости на координатные оси. Таким образом, за касательные векторы здесь можно принимать векторы такой условной (или реальной) скорости.

Запрограммировать вычерчивание параметрической линии, в принципе, не сложнее, чем обычной. Возьмем к примеру эллипс – сначала такой, оси которого параллельны координатным осям. Его параметрические уравнения:

$$x = x_C + R_1 \cos \alpha, \quad y = y_C + R_2 \sin \alpha.$$

От уравнений окружности они отличаются только тем, что вместо одной величины радиуса здесь используются величины двух полуосей (первая параллельна оси x). Судя по рис. 51, эллипс здесь понимается как сжатая (или растянутая) окружность. Угол α – это угол, образованный осью x и радиусом окружности, описанной вокруг эллипса (или вписанной в него), который направлен в точку, перенесенную на окружность с эллипса параллельно координатной оси. Поэтому для нас угол α – производящий параметр (иногда такой угол называют *фазовым*). Если же нас интересует центральный угол точки эллипса (т.е. угол наклона радиуса, прове-

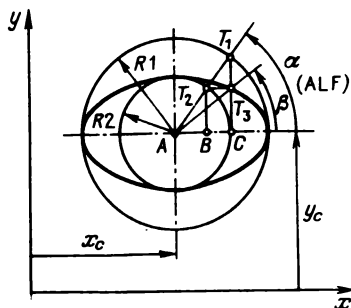


Рис. 51. Схема задания эллипса

денного в точку эллипса), можно привести тригонометрическое соотношение $\text{tg } \alpha / \text{tg } \beta = R_1/R_2$. Подпрограмма вычерчивания такого эллипса или его дуги будет очень похожа на подпрограмму SINLIN:

```

SUBROUTINE ELLIN(XC, YC, R1, R2, ALN, DAL, DDA)
  DD=SIGN(DDA, DAL)
  AL=ALN
  K=0

```

```

1 X=XC+R1 * COS (AL)
  Y=YC+R2 * SIN (AL)
  CALL LINX (X, Y)
  CALL PERO
  IF (K.NE. 0) GOTO 2
  AL=AL+DD
  IF ((ALN+DAL-AL) * DAL) 3,3,1
3 AL=ALN+DAL
  K=1
  GOTO 1
2 CALL VP
  RETURN
  END

```

Параметр AL в этой подпрограмме изменяется с постоянным приращением. Достоинством является то, что опорные точки на эллипсе располагаются "гуще" там, где кривизна линии больше.

Эллипс общего положения может быть определен примерно так же, только дополнительно надо задать угол (GAM), под которым первая полуось наклонена к оси x; уравнения усложнятся:

$$X = XC + R1 * \cos(ALF) * \cos(GAM) - R2 * \sin(ALF) * \sin(GAM)$$

$$Y = YC + R1 * \cos(ALF) * \sin(GAM) + R2 * \sin(ALF) * \cos(GAM)$$

В соответствии с этими уравнениями работает подпрограмма EL2 из системы Алграф, вызов которой имеет вид (углы задаются обычно в градусах)

```
CALL EL2(XC, YC, R1, R2, GAM, ALN, DAL)
```

По принятым правилам перо подпрограммой не выключается после черчения дуги, величина шага аргумента-параметра берется из системной переменной AAA(9). Для сокращения затрат машинного времени предварительно (не в цикле) подсчитываются величины

$$A1 = R1 * \cos(GAM); A2 = -R2 * \sin(GAM);$$

$A3 = R1 * \sin(GAM); A4 = R2 * \cos(GAM)$. Тогда уравнения эллипса будут иметь вид

$$X = XC + A1 * \cos(ALF) + A2 * \sin(ALF)$$

$$Y = YC + A3 * \cos(ALF) + A4 * \sin(ALF)$$

А что будет за линия, если в приведенных только что уравнениях величины A1, A2, A3 и A4 задать произвольно?

Это будет также эллипс общего положения. Однако как пройдут его оси и какой они будут величины — догадаться не так легко (попробуйте). Здесь есть такая особенность: четыре точки, соответствующие значениям параметра ALF (0, $\pi/2$, π , $-\pi/2$), являются концами двух сопряженных

диаметров (рис. 52). Известно, что касательные, проведенные к эллипсу в концах одного такого диаметра, будут параллельны второму и эллипс можно вписать в параллелограмм, образованный отрезками таких касательных. Подпрограмма EL7 в системе Алграф строит эллипс по координатам трех точек – центра и концов двух сопряженных полуосей:

CALL EL7 (XC, YC, X1, Y1, X2, Y2)

Подпрограмма EL8 при аналитическом вызове проводит дугу, соответствующую 90° фазового угла, между точками, заданными координатами X1, Y1, X2, Y2. Отсюда уже несложно придти к конфигурации данных, показанной ранее на рис. 31. Действительно, как видим из рис. 52, б, если

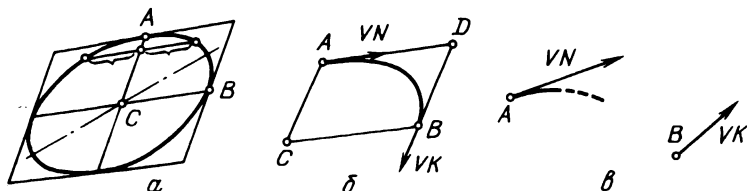


Рис. 52. Сопряженные диаметры у эллипса; а – изображение целого эллипса с сопряженными диаметрами; б – дуга эллипса, соответствующая 90° фазового угла; в – конфигурация исходных данных, не подходящая для построения дуги эллипса

от параллелограмма с вписанным в него эллипсом оставить четверть, получим дугу, вписанную в угол (ADB). По векторам направления нетрудно построить сопряженные диаметры и определить центр эллипса. В системе Алграф дуга строится от отрабтанной точки с помощью подпрограммы EL 13. Вызов ее имеет такой вид:

CALL EL13 (DXN, DYN, XK, YK, DXK, DYK)

Первые два параметра задают вектор направления в начальной точке, следующие четыре параметра – координаты конечной точки и вектор направления в ней.

Однако не все конкретные конфигурации данных позволяют с помощью этой подпрограммы построить дугу: ведь дуга эллипса никак не может иметь точки перегиба и "загибаться" может только в одну сторону. Так, на рис. 52, в приведена конфигурация данных, на которую подпрограмма отвечает "отказом". Задача читателям: сформулируйте геометрические и логические условия для конфигурации данных, обеспечивающие возможность построения дуги!

Для облегчения построения различных параметрических линий составим "промежуточную" подпрограмму CPLINE, которая наподобие ранее рассмотренной подпрограммы CLINE будет вызывать указанную подпрограмму, рассчитывающую координаты точки линии. Пусть вызов CPLINE

имеет вид

```
CALL CPLINE (FU, T1, T2, DT, AMD)
```

Вместо граничных значений координаты-аргумента здесь используются граничные значения производящего параметра. Подпрограмма FU должна иметь такой заголовок:

```
SUBROUTINE FU (T, AMD, X, Y)
```

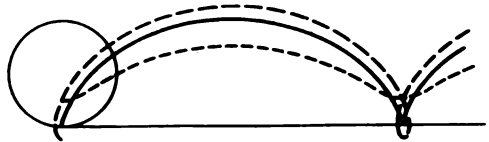
Параметр T здесь – аргумент, X и Y – результат (значения координат). С помощью этой методики рассмотрим некоторые типы параметрических кривых.

Циклоиды и трохойды. Это кривые, которые может описать точка катящейся по прямой окружности или точка, жестко скрепленная с окружностью (и находящаяся внутри или вне ее). Точка, принадлежащая самой окружности, описывает волнообразную кривую с острями (точками возврата), упирающимися в прямую, по которой окружность катится. Это – *циклоида*. Точки, расположенные внутри окружности, описывают *укороченные трохойды* – гладкие волнообразные кривые. Точки вне окружности описывают *удлиненные трохойды* с петлями. Все эти кривые показаны на рис. 53. С точки зрения механики такое движение точки можно рассматривать как сумму двух движений – вращательного и прямолинейного поступательного. Подпрограмма TRONFU, которая вызывалась для получения рис. 52, такова:

```
SUBROUTINE TRONFU (T, AMD, X, Y)
DIMENSION AMD(4)
X=AMD(1)+AMD(3)*T-AMD(4)*SIN(T)
Y=AMD(2)-AMD(4)*COS(T)+AMD(3)
RETURN
END
```

Здесь составляющие аналитической модели AMD(1) и AMD(2) – координаты "привязочной" точки линии, за которую принимается точка касания

Рис. 53. Циклоида и две трохойды, полученные с помощью подпрограммы TRONFU



окружности и прямой при нулевом значении параметра T; AMD(3) – радиус катящейся окружности и AMD(4) – расстояние от "рисующей" точки до центра окружности. При T = 0 эта точка расположена на радиусе, направленном в точку касания (или на его продолжении). Естественно, что циклоида с острями получается при AMD(3) = AMD(4).

Интересны *эпитрохоиды* и *гипотрохоиды* со своими частными случаями – *эпициклоидами* и *гипоциклоидами*. Эти линии получаются, когда

одна окружность катится по другой, неподвижной. Она может катиться по наружной стороне окружности или по внутренней ее стороне. Линия, образующаяся в первом случае, называется *эпитрохойдой*, во втором случае – *гипотрохойдой*. "Рисующая" точка может располагаться внутри или вне катящейся окружности, оставляя соответственно за собой гладкие волнообразные линии без петель либо с петлями. Если точка принадлежит окружности, получаются *эпи-* или *гипоциклоиды* – линии с острями.

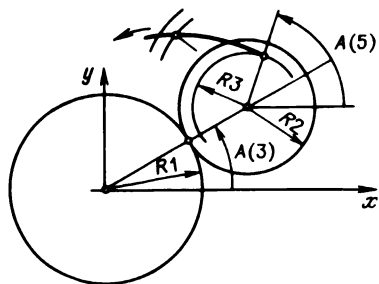


Рис. 54. Схема задания эпитрохоеид и гипотрохоеид

Для всех таких линий мы составим одну подпрограмму:

```

SUBROUTINE EPITRF (T, A, X, Y)
DIMENSION A (5)
RR=A (1) + A (2)
FI=A (3) + T
PSI=A (5) + T * (1+A (1)/A (2))
X=RR * COSG (FI) + A (4) * COSG (PSI)
Y=RR * SING (FI) + A (4) * SING (PSI)
RETURN
END

```

В этой подпрограмме тригонометрические функции SING и COSG воспринимают угловой аргумент в градусах или других величинах в соответствии с настройкой системы Алграф. Согласно рис. 54 центр неподвижной окружности мы считаем размещенным в начале координат. Параметр $A(1) = R_1$ будет радиусом этой окружности, параметр $A(2) = R_2$ – радиусом катящейся окружности, $A(4) = R_3$ определяет расстояние от рисующей точки до центра катящейся окружности. Параметры $A(3)$ и $A(5)$ угловые; они определяют в начальный момент (при $T = 0$) положение катящейся окружности и рисующей точки относительно нее.

Эпитрохоеиды и гипотрохоеиды получаются соответственно при положительных и отрицательных значениях радиуса R_2 ; эпи- и гипоциклоиды получаются, когда по абсолютной величине равны R_2 и R_3 .

Об эпициклоидах говорили еще в средние века... применительно к движению планет по небу. Когда, согласно геоцентрической системе Птолемея, Земля считалась неподвижной, получалось, что планеты весьма витиевато ходят по небосводу. Движение их складывалось из вращения по

окружностям – эпициклом, а центры эпициклов в свою очередь вращались по другим окружностям – “деферентам”. С широкой точки зрения эти рассуждения и ныне не устарели: ведь для каких-то целей может понадобиться описание движения планет в системе координат, скрепленной с Землей. Читатели могут поискать в современной технике, науке примеры движений по таким линиям.

В зависимости от соотношений параметров в аналитической модели эти линии довольно разнообразны. Во-первых, важно, является ли отношение $R2/R1$ рациональным. Если да, то линия, сделав один оборот или несколько оборотов вокруг неподвижной окружности, замкнется. Если нет – она, не замыкаясь, будет “накручиваться” неограниченно долго. Если $R1$ кратно $R2$, линия замкнется, сделав один оборот. В этих случаях можно получить волнообразные замкнутые линии, напоминающие скрученные в кольцо пружины, и т.д. Интересны частные случаи. Среди них – линия, выродившаяся в отрезок*) (определите для него параметры аналитической модели). При $R1 = R2 = R3$ получается *кардиоида*. На рис. 55 показано машинное изображение нескольких линий этого типа и значения параметров АД для них.

Параметры линий	R 1	R 2	A (3)	R 3	A (5)	T 1	TK
№ линии							
1	20	20	90	20	90	0	360
2	20	20	90	35	90	0	360
3	45	15	90	22.5	90	0	360
4	30	-10	90	10	90	0	360
5	45	15	90	7	90	0	360

На рис. 56 изображено декоративное семейство гипотрохрид, начинающееся с гипоциклоиды и полученное при различных удалениях рисующей точки от центра катящейся окружности.

Сохраняя принцип качения, можно расширить постановку задачи, заменив окружности на какие-либо другие кривые (например, на эллипсы). Аналитически такие задачи решать трудно, но численными методами и с помощью машинной графики можно исследовать множество случаев.

Гармонические колебания. Электротехника и радиотехника насквозь “пропизаны” колебаниями – синусоидальными и более сложными. Их можно видеть на экранах осциллографов. Основным геометрическим образом является синусоида и рожденные на ее основе другие кривые. Синусоида уже рассматривалась; теперь рассмотрим параметрические линии, возникающие при сложении или “суперпозиции” нескольких колебаний синусоидального характера. Аргументами тригонометрических

*) В данном случае отрезок является “двойным”; при вычерчивании на графопостроителе он будет вычерчиваться дважды – в противоположных направлениях.

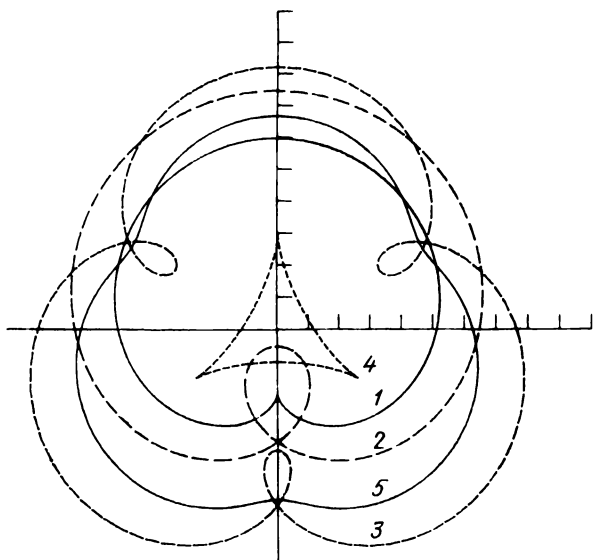


Рис. 55. Машинное изображение эпи- и гипотрохрид

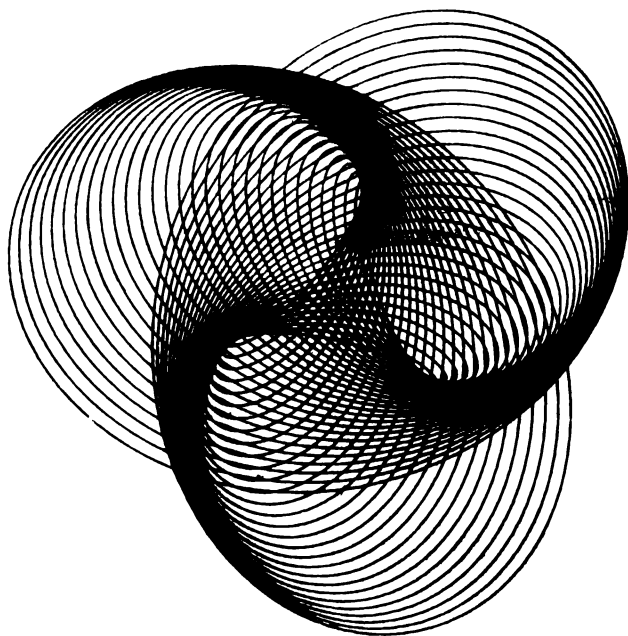


Рис. 56. Декоративное семейство гипотрохрид, включающее гипоциклоиду

функций будем считать не "геометрические", а фазовые углы, сопоставляемые со временем. Исследуем сначала картину двух синусоидальных колебаний, когда одно совершается в направлении оси x , а другое – в перпендикулярном направлении. Такая картина часто возникает на экране осциллографа – если на один вход (для горизонтальной развертки) подать одно пульсирующее напряжение, а на другой (для вертикальной развертки) – другое. Получающиеся линии называются фигурами Лиссажу. В зависимости от соотношения частот, фаз, амплитуд могут возникать частные случаи – отрезки, эллипсы, окружности. Линии могут быть замкнутыми и незамкнутыми. На рис. 57 показано несколько типичных линий этого рода.

Если нейтральное положение точки совместить с началом координатной системы, параметрические уравнения линии можно записать так:

$$X=AX * \text{SIN} (T * FX+TX)$$

$$Y=AY * \text{SIN} (T * FY+TY)$$

Величины AX и AY здесь играют роль амплитуд, T – производящий параметр, имитирующий время; FX и FY определяют периоды колебаний; по существу, это "круговые частоты". TX и TY устанавливают фазу колебаний. Подпрограмму вычисления координат точки на базе приведенных соотношений написать нетрудно. Попробуйте теперь определить параметры аналитической модели линии для таких частных случаев, как отрезок, окружность, эллипсы с осями, параллельными координатным осям, и с наклонными осями, для линии, имеющей вид положенной на бок восьмерки.

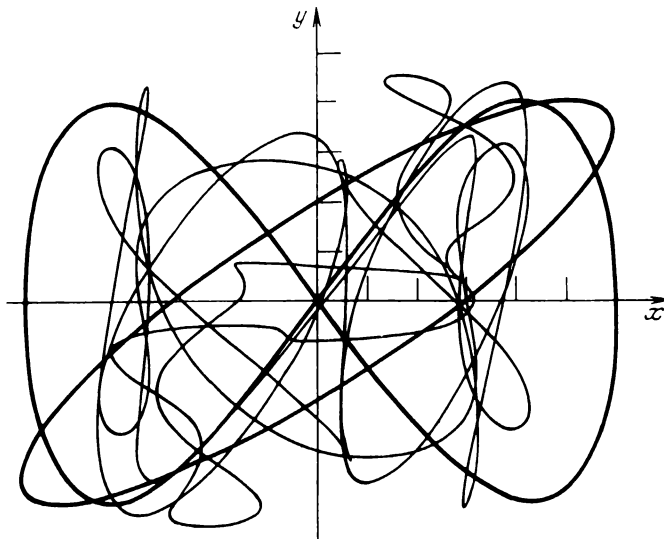


Рис. 57. Кривые Лиссажу

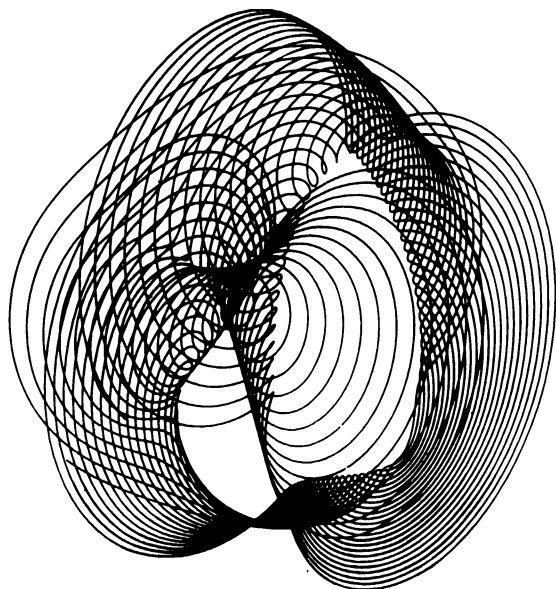


Рис. 58. Кривые, полученные сложением гармонических колебаний

Усложнить задачу можно в таком направлении. Пусть движение точки определяется сложением нескольких колебаний, каждое из которых совершается вдоль прямой, но в своем направлении, имеет свою амплитуду и частоту. Результирующие координаты точки можно подсчитать по формулам:

$$x = \sum_i A_i \sin(tF_i + t_{0_i}) \cos \varphi_i,$$

$$y = \sum_i A_i \sin(tF_i + t_{0_i}) \sin \varphi_i.$$

Индекс i определяет "номер" составляющей колебания, A_i – амплитуда, F_i – период колебаний составляющей, углы φ_i – это углы, под которыми направлены колебания (по отношению к оси x). Линии такого типа очень разнообразны, можно получить эпи- и гипотроихиды, а можно – такую сложную фигуру, как изображенная на рис. 58. Когда-то линии примерно такого типа получали с помощью механических приборов, содержащих маятники. Эти приборы играли роль "интеллектуальных игрушек" еще в 18 веке.

3.3. Построение линий по заданным условиям

В предыдущих параграфах мы получали линии, задавая коэффициенты их уравнений и другие величины, составляющие, собственно, аналитические модели линий. При этом заранее не всегда знали, как точно пройдет линия. Для графического устройства (или последующих расчетов) линии представлялись наборами точек, т.е. координатными моделями. Таким образом, реализовалась схема: аналитическая модель — координатная модель (АМД → КМД). В практике конструирования и исследований чаще встречаются ситуации, когда исходными данными для построения линий являются координаты некоторых точек, касательные векторы, значения кривизны и т.д. В этих случаях реализуется упомянутая раньше схема: исходная КМД → АМД → результирующая КМД. Исходная координатная модель может быть минимальной или приближенной и в обоих случаях расширенной (когда кроме координат заданы касательные векторы и другие параметры). Иногда для построения линий задаются специальные геометрические условия, индуцирующие решение специальных задач геометрического конструирования. Например, задано несколько прямых или кривых линий, а синтезируемая кривая должна касаться их в заданной последовательности. Другой пример — линия должна проходить на заданном расстоянии от другой, уже известной линии. В последнем случае имеем задачу построения эквидистанты.

Ниже мы рассмотрим две типичные задачи — *интерполяции* и *аппроксимации* заданного точечного ряда кривой линией определенного типа. В простом случае точечный ряд задан координатами точек — т.е., по нашей терминологии, задана исходная координатная модель. При интерполяции кривая должна точно (с возможной для машины точностью счета) проходить через заданные точки. Чтобы ее математически можно было построить, координатная модель должна содержать необходимый минимум точек. Помимо интерполяции (определяющей объект "между" заданными его элементами), возможна и экстраполяция — построение объекта вне интервала, определяемого заданными элементами. Аппроксимация — это поиск объекта заданного типа, как можно лучше приближающегося к исходным элементам. Итак, если координатная модель считается приближенной, мы ищем объект, близкий к ней. Понятия интерполяции и аппроксимации применяются не только к двумерным линиям, но и к пространственным, к поверхностям и другим объектам.

В качестве примера интерполяции точечного ряда рассмотрим задачу проведения через заданные точки линии полинома. С точки зрения "исчислительной" геометрии [2] полином степени n можно провести через $n + 1$ заданных точек. Действительно, уравнение такого полинома содержит $n + 1$ коэффициент (включая свободный член). Задача решается с помощью формирования системы линейных уравнений и ее решения каким-либо стандартным способом. Считая коэффициенты полинома неизвестными, $n + 1$ раз подставляем в уравнение координаты заданных точек и получаем систе-

му из $n + 1$ уравнения, линейного относительно искомым коэффициентов. Пусть, к примеру, степень полинома равна двум, заданы три точки. Обозначив коэффициенты через A_1, A_2, A_3 , получаем систему

$$A_1 X_1^2 + A_2 X_1 + A_3 = Y_1,$$

$$A_1 X_2^2 + A_2 X_2 + A_3 = Y_2,$$

$$A_1 X_3^2 + A_2 X_3 + A_3 = Y_3.$$

При желании решить ее можно алгебраически (если решение существует), но для систем с разным числом уравнений и неизвестных используются такие типовые подпрограммы, как SIMQ. Для ее работы надо образовать два массива – двумерный массив коэффициентов уравнений

$$\begin{bmatrix} X_1^2 & X_1 & 1 \\ X_2^2 & X_2 & 1 \\ X_3^2 & X_3 & 1 \end{bmatrix}$$

и одномерный массив из правых частей (содержащий Y_1, Y_2, Y_3). В результате работы программы машина разместит ответ в одномерном массиве (правые части Y_1, Y_2, Y_3 стираются). Программе также задается порядок системы N (число неизвестных). Исходные данные для решения системы могут быть заданы неверно (например, несколько раз будут повторены координаты одной и той же точки). В таких случаях система не решается. Решилась или не решилась система – подпрограмма сообщает об этом с помощью специальной переменной IR (индекс результата). В целом, обращение к подпрограмме имеет вид CALL SIMQ (A, B, N, IR).

Определив коэффициенты, т.е. аналитическую модель полинома, дугу его можно прочертить с помощью ранее рассмотренных подпрограмм CLINE и POLFU. Приведем главную программу и подпрограмму AMDPOL определения коэффициентов полинома. В главной программе читаются исходные данные, вызывается подпрограмма AMDPOL, вычерчивается дуга интерполирующего полинома и заданные точки помечаются маленькими кружочками. Любопытно, что порядок задания исходных точек в данном случае не имеет значения.

```

DIMENSION A(100),AM(20),RAB(400)
EXTERNAL POLFU
CALL SBROS
READ(5,9)NT,XN,XK,DX
9 FORMAT (I8,3F8.0)
NK=NT*2
READ(5,8)(A(K),K=1,NK)
8 FORMAT (10F8.0)
CALL AMDPOL(A,NT,AM(2),RAB)
AM(1)=NT-1
CALL CLINE(POLFU,XN,XK,DX,AM,0)
DO 1 K=1,NT

```

```

1 CALL OK(A(K * 2-1), A(K * 2), 1.)
  CALL SHKALA
  CALL STP
  END
  SUBROUTINE AMDPOL(A, NC, AMD, RAB)
  DIMENSION A(1), AMD(1), RAB(NC, NC)
  DO 1 J=1, NC
  C=1.
  DO 2 K=1, NC
  RAB(J, NC-K+1)=C
2 C=C+A(J*2-1)
1 AMD(J)=A(J*2)
  CALL SIMQ(RAB, AMD, NC, IR)
  IF(IR.NE.0) CALL ERRALG(199, IR, 0.)
  RETURN
  END

```

Пусть читателя не смущает, что массив A для координат исходных точек объявлен как одномерный, а массив RAB для составления матрицы коэффициентов линейных уравнений в главной программе одномерный, а в подпрограмме – двумерный. Такие “вольности” фортран разрешает. На рис. 59 приведены примеры – изображения полиномов пятой степени. Одна из линий демонстрирует явление волнообразования, характерное для полиномов высоких степеней. “Волнение” наблюдается у полинома (линия 1), хотя из шести исходных точек пять расположены на одной прямой! А за пределами интервала, на котором расположены исходные точки полиномы всегда имеют тенденцию неограниченно возрастать или убывать.

Почти не меняя описанной методики, можно построить полиномы, не только проходящие через заданные точки, но и имеющие в них заданные касательные или заданную кривизну. В целом количество всех единичных геометрических условий (прохождение через точку, заданное направление в точке, заданная кривизна в точке) должно соответствовать параметрическому числу полинома ($n + 1$, где n – степень полинома). Математический анализ и дифференциальная геометрия позволяют свести эти “геометрические” условия к значениям первой и второй производных, для полиномов соответствующие аналитические выражения легко находятся. Так что в системе линейных уравнений, которая строится

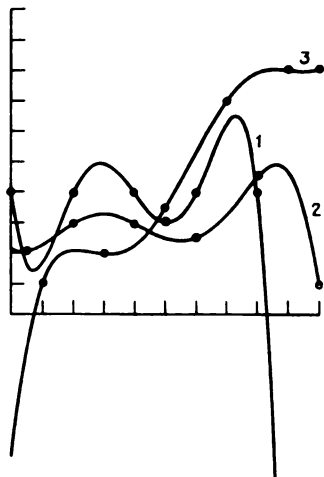


Рис. 59. Примеры линий – полиномов, построенных по точкам

для определения коэффициентов полинома, некоторые будут отличаться от показанных выше.

Сходную методiku можно применить для ряда линий иных типов, например для линий второго, третьего порядков. В табл. 1 было помечено, что параметрическое число линии второго порядка – 5. Значит, ее можно провести через 5 точек, заданных на плоскости, либо удовлетворить пяти другим единичным геометрическим условиям (например, провести линию через 3 точки с заданным наклоном в двух из них). Здесь, правда, есть свои трудности. Например, заданные точки могут "попасть" на разные ветви гиперболы или кривая пройдет через точки не в желаемом порядке.

Построение линий некоторых типов не сводится к линейным системам уравнений. В этих случаях при необходимости используют численные методы.

Через заданные точки можно проводить и параметрические линии. С формальной стороны можно, как было описано, построить линию, уравнение которой $x = f(t)$ и линию $y = \varphi(t)$. Получив два уравнения, "распоряжаемся" ими как параметрическими соотношениями для линии в плоскости xu . Но в состав исходных данных теперь входят не только координаты исходных точек, но и значения производящего параметра. Образно говоря, если линию представить как путь точки, то в исходной модели мы задаем не только "пункты назначения", но и "время прибытия".

Коснемся теперь вопроса аппроксимации. Пусть приближенная координатная модель содержит много точек; точно пройти через них линия заданного типа не может, да нам и не нужно следовать всем погрешностям в задании точек – надо как раз "сгладить" ту неровную линию, которая могла быть исходной для снятия приближенной модели. Будем искать среди линий заданного типа такую, которая проходит возможно ближе к заданным точкам. Что значит "возможно ближе"? Уточнить это понятие можно различным образом. Приведем наиболее распространенную концепцию. Если отталкиваться от понятия графика функции, отклонение линии от заданной точки естественно оценивать по разности ординат этой точки и точки линии, имеющей ту же абсциссу; как правило, возводят отклонения в квадрат и подсчитывают среднее значение квадрата отклонения. Этот подход приводит к методу *наименьших квадратов*, который имеет ряд преимуществ и среди них, в частности, то, что для определенных типов линий этот способ довольно просто реализуется аналитически и программно.

Для применения этого способа уравнение кривой соответствующим образом преобразуется, дифференцируется и образуется система линейных уравнений, решение которой дает значения коэффициентов в уравнении, обеспечивающие минимально возможную величину суммы квадратов отклонений.

На рис. 60 показаны линии полиномов третьей степени, построенные этим способом по заданным наборам точек (помеченных кружочками).

Рис. 60. Полиномы, аппроксимирующие заданные точечные ряды

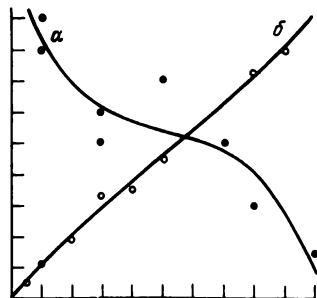
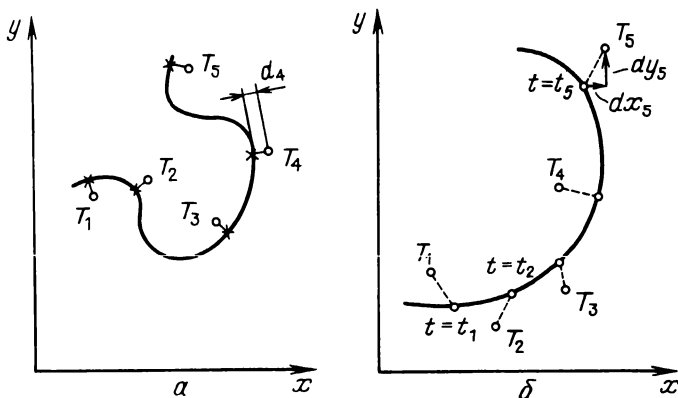


Рис. 61. Определение отклонений точек для параметрической линии (а) и линии сложной формы (б)



Применение метода наименьших квадратов достаточно разнообразно. Например, исходные точки можно задавать "с весами", и они тогда, в зависимости от веса, то сильнее, то слабее влияют на форму линии. Используется этот способ и для построения параметрических линий, и для образования аппроксимирующих поверхностей. Для параметрических линий, которые могут быть замкнутыми и самопересекающимися, отклонения от заданных точек определяются уже не по вертикали (т.е. по разности ординат для заданной точки и точки кривой с той же абсциссой), а в направлении нормалей, опущенных из точек на линию (рис. 61, а), либо за отклонения принимаются проекции на координатные оси отрезков, соединяющих исходные точки с точками линии, имеющими определенные значения параметра (так, на рис. 61, б для точки T_5 на линии выбирается точка со значением параметра t_5 и рассматриваются отклонения dx_5, dy_5).

Помимо задач интерполяции и аппроксимации возникает множество других задач построения линий по тем или иным геометрическим условиям. Нередко такие построения соответствуют концепциям *геометрического конструирования*. Последовательные операции типа "построить перпендикуляр", "провести касательную", "найти точки пересечения двух окружностей"

тей” позволяют решать и сложные задачи. В начертательной геометрии, в черчении такие операции выполняются графически, с помощью линеек, циркулей и карандашей. В автоматизированных системах – методом вычислений и преобразования численных моделей. Здесь машинная геометрия ближе к аналитической геометрии, чем к начертательной. Допустим, для определения натуральной величины отрезка, изображенного в двух проекциях и произвольно расположенного в пространстве (рис. 62, а), надо выполнить несколько графических операций. В ЭВМ моделировать последовательность таких операций неразумно: проще воспользоваться формулой расстояния между двумя точками $S = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$. Нередко для построения какой-либо фигуры достаточно применить несложные выкладки, тригонометрические функции или, к примеру, теорему Пифагора. Так, для определения координаты точки R на рис. 62, б в треугольнике PQR следует вычислить длину стороны PQ: $S_{PQ} = \sqrt{R^2 - (A/2)^2}$; затем отрезок TR определяется как разность величин R и S_{PQ} . В случае же, показанном на рис. 62, в, надо определять точки пересечения двух окружностей. Конечно, нужные аналитические выражения может найти и школьник, но это довольно трудоемко, поэтому в программной системе должен быть готовый модуль, решающий эту задачу. Соответствующий модуль в системе Алграф вызывается так:

CALL SCOK (A, B, IM, C, IR)

где A и B – массивы, содержащие аналитические модели двух окружностей.

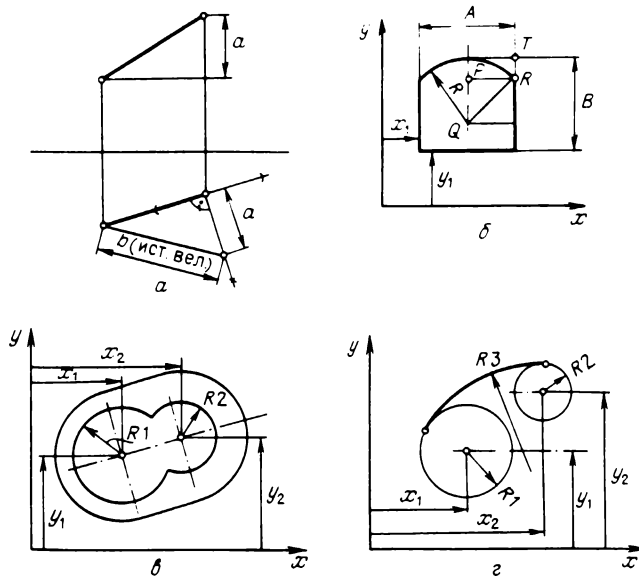


Рис. 62. Примеры операций геометрического конструирования

IM – индекс модификации (в данном случае его следует задать равным 2), C – массив для размещения результата – координат двух точек пересечения и одного дополнительного числа, IR – индекс результата, принимающий нулевое значение, если пересечение найдено, и ненулевое – если окружности не пересекаются.

На рис. 62,2 изображена несколько более сложная задача – надо построить сопряжение двух заданных окружностей дугой заданного радиуса. Условия касания окружностей и известный радиус – вот геометрические условия для построения новой окружности и затем выделения на ней дуги. Графически задача решается довольно просто – сводится к построению заделки дугами с радиусами $R_3 \pm R_1$ и $R_3 \pm R_2$. Аналитические выкладки более трудоемки. Здесь также помогает готовая подпрограмма:

CALL KASOOA (A, B, R3, IM, C, IR)

Здесь A и B – имена массивов, содержащих аналитические модели исходных окружностей (точнее – дуг, так как знак пятого параметра в каждой аналитической модели определяет направление обхода по окружности); радиус сопрягающей дуги R3 также задается со знаком, указывающим направление обхода. При этом возможны два варианта построения дуги, плавно и с учетом указанных направлений обхода сопрягающей исходные окружности. Окончательный вариант выбирается значением индекса IM (0 или 1). Массив C содержит результат – аналитическую модель сопрягающей дуги (эта аналитическая модель соответствует способу задания дуги для подпрограммы OK1); дополнительно в массив C записывается еще один, контрольный параметр. Параметр IR имеет тот же смысл, что для подпрограммы SCOK.

3.4. Составные линии

В названии "составные линии" заложен простой смысл – линия составляется из отдельных кусков – дуг и, возможно, отрезков. Однородные составные линии состоят из дуг одного типа. Простейшим примером составной линии может служить ломаная. Линии, гладкие и с изломами, составленные из отрезков и дуг окружностей, распространены в технике и других областях очень широко. Линия такого типа, причем гладкая, изображена на рис. 30. Поищите подобные линии вокруг себя! Сформируем следующую задачу: приближенно намеченную гладкую кривую линию произвольной сложности описать математически. Поскольку эта линия может иметь любую форму, подобрать к ней аппроксимирующий полином или другую линию заранее заданного типа вряд ли представится возможным (особенно если учесть трудности, возникающие при работе с аналитическими сложными кривыми). Можно формировать кривую по частям. Предположим, что каждая часть по форме достаточно проста и что точная форма каждой части несущественна. В этом случае в состав исходных данных входят координаты граничных точек каждой части; другими словами – между

каждыми двумя последовательными точками исходной координатной модели проводится отдельная дуга. Кроме координат, в этих опорных точках могут быть заданы касательные векторы и другие параметры (см. рис. 31). Для построения общей гладкой линии дуги должны стыковаться так, чтобы в точках стыка направления касательных векторов совпадали, т.е. значения первых производных были равны. (Стыковка первого порядка гладкости. Подразумевается, что сами дуги также гладкие.)

Гладкость первого порядка обеспечивается автоматически при задании в опорных точках координат и векторов направления; каждая из дуг тогда строится по конфигурации данных, ранее показанной на рис. 31. Гладкость первого порядка могут иметь и линии, составленные из отрезков и дуг окружностей, и многие другие линии. Выше мы говорили, однако, что в некоторых случаях требуется гладкость линии второго порядка, а это значит, что смежные дуги в точках стыка должны иметь одинаковые кривизны.

Составные кривые можно строить из дуг разного типа, в том числе из дуг линий второго порядка. Составные линии из таких дуг могут иметь точки перегиба, но только в местах сопряжения смежных дуг. Обычно предпочитают брать такие дуги, которые могут иметь точки перегиба, что освобождает от необходимости дополнительного контроля за исходными данными.

Простейшая линия с точкой перегиба – кубическая парабола. Такие дуги или дуги полиномов четвертой, пятой степени используют для построения составных линий, аппроксимирующих графики функций и экспериментальных зависимостей и в ряде других случаев. О дуге кубической параболы, построенной по координатам граничных точек и значениям производных в них, уже говорилось, и из них даже был сформирован узор, показанный на рис. 47. Производные, как известно, определяют наклоны касательных, и для дуг полиномов (графиков однозначных функций) "прямое" или "обратное" направление движения вдоль касательной определяется значениями абсцисс начальной и конечной точек. Приведем ниже подпрограмму, которая в свою очередь вызывает подпрограмму CLINE, а посленая с помощью функции PO3FU вычерчивает дугу. Вычерчивание нескольких таких дуг в цикле дает составную кривую первого порядка гладкости.

```
SUBROUTINE MLINE (A, NT, FU, DX)
  DIMENSION A (3, NT)
  NL=NT-1
  DO 1 J=1, NL
1  CALL CLINE(FU, A(1, J), A(1, J+1), DX, A(1, J), 0)
  RETURN
END
```

Массив А содержит расширенную исходную координатную модель – для каждой точки в ней указаны координаты и угловой коэффициент касательной. У последовательно заданных точек координаты x должны либо монотонно возрастать, либо убывать. Параметр NT указывает число точек в

координатной модели, FU — это переменная, с помощью которой подпрограмме CLINE передается имя функции POZFU (в принципе, это может быть и другая функция), DX — шаг по координате x для получения "плотной" (с частым расположением точек) выходной координатной модели. Приведем также простую главную программу, читающую исходные данные с перфокарт и вызывающую подпрограмму черчения составной линии:

```

DIMENSION A(3, 100)
EXTERNAL POZFU
READ 1, NT
READ 2, ((A(J, K), J=1, 3), K=1, NT)
1 FORMAT (I5)
2 FORMAT (12F5.0)
CALL SBROS
CALL MLINE(A, NT, POZFU, 1.)
CALL SHKALA
CALL STP
END

```

На рис. 63 показана линия этого типа, построенная по следующей координатной модели (параметры здесь искусственно отделены запятыми):

0.,10.,0.,20.,40.,0.5,40.,45.,-0.2,80.,20.,0.1,100.,25.,0.

Во многих случаях составная линия имеет "произвольную" форму, может содержать точки с "вертикальными" касательными, иногда — точки самопересечения. Обычные полиномы для таких линий не подходят, поэтому либо каждую дугу программа строит в своей локальной координатной системе (о которой пользователь не заботится), либо

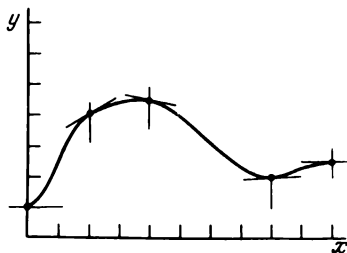


Рис. 63. Составная линия из дуг полиномов 3-й степени

используют параметрические полиномы. Линия параметрического полинома уже третьей степени может иметь точку возврата или самопересечения; параметрическое число*) такой линии — 8, что дает основания рассчитывать на разнообразие формы. Дугу такого полинома можно построить по "стандартной" конфигурации данных (координаты и касательные векторы в граничных точках), остается возможность дополнитель-

*) Параметрическое число равно общему числу коэффициентов в уравнениях.

ного регулирования формы дуги. Компоненты касательных векторов здесь являются значениями производных от соответствующих координат по параметру. Если рассматривать физическую модель построения линии, то параметр может быть временем, и тогда длина касательного вектора будет отображать скорость движения точки по линии. Эта скорость влияет на "гибкость" линии (способность линии к искривлению). При малых длинах векторов дуга вблизи граничных точек имеет обычно увеличенную кривизну, а в средней части — небольшую; при больших длинах наоборот, кривизна у граничных точек мала, а экстремум ее приходится на среднюю часть дуги; при дальнейшем удлинении векторов в средней части образуется особая точка или петля. Для программ, в которых для начальной точки дуги значение параметра берется нулевым, а для конечной — единичным, сравнительно равномерное распределение кривизны по длине дуги можно получить, задавая векторы, соизмеримые по длине с расстоянием между граничными точками.

В системе Алграф одну дугу этого типа можно построить оператором `CALL POLIN3 (A, NA, IM)` а составную линию — оператором `CALL MPPOL3 (A, NT, IM)`

Здесь *A* — массив, содержащий исходные геометрические данные, *IM* — индекс модификации, *NT* — число точек в исходной координатной модели составной линии, а *NA* указывает номер элемента массива *A*, где расположен первый геометрический параметр для начальной точки дуги. Для подпрограммы `POLIN3` мы не будем обсуждать все модификации. При задании в исходном массиве координат и компонент касательных векторов в начальной и конечной точках используются модификации 3 и 4. Первая из них реагирует на длину векторов, вторая нормирует векторы по длине и строит кривые с возможно равномерным распределением кривизны. На рис. 64 с помощью модификации 3 проведены дуги *B1*, *B2*, *B3*, а с помощью модификации — 4 дуга *G*, причем исходные данные для нее были те же, что для дуги *B3*. В качестве маленького парадокса показана дуга *E*, начинающаяся и кончающаяся в одной точке.

Подпрограмма `MPPOL3` при задании в опорных точках координат и векторов вызывается в модификациях 2 и 3 (при модификации 3 векторы нормируются). Линии *D1* и *D2* проведены соответственно этими способами (обратите внимание на петлю, возникшую на первой из линий). Но эта же подпрограмма строит составные кривые при задании только координат опорных точек; при этом модификация 0 соответствует незамкнутой, а модификация 1 — замкнутой линии. Направления линии в опорных точках определяются особым способом (связанным с построением вспомогательных парабол второй степени). Для возможно равномерного распределения кривизны расстояния между опорными точками должны быть примерно одинаковыми. Рис. 64 на примере линии *C* показывает, что на коротких дугах могут возникнуть экстремумы кривизны.

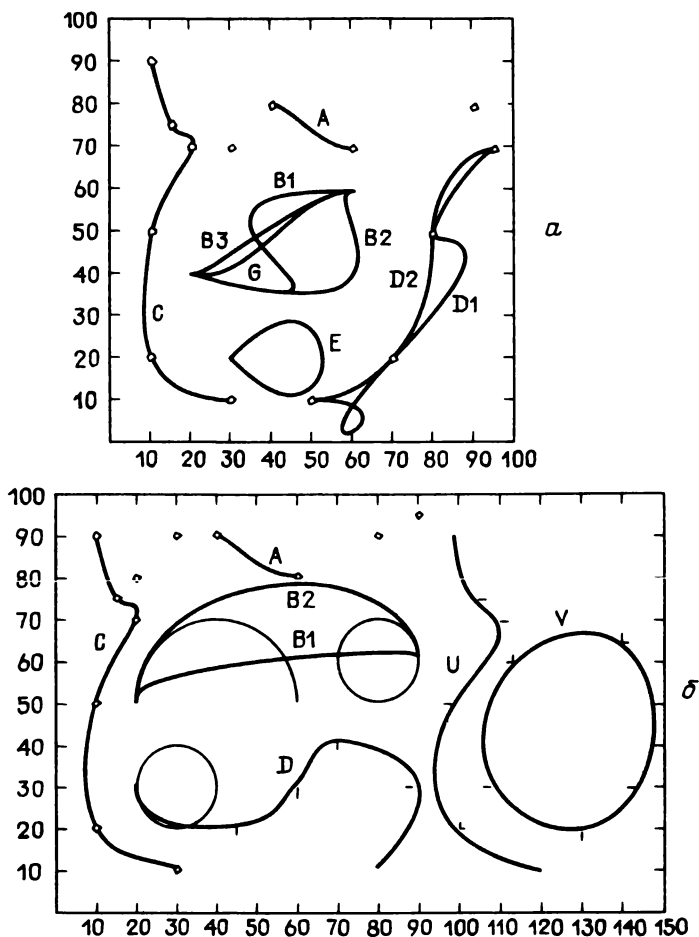


Рис. 64. Дуги параметрических полиномов и составные линии из них: *а* – на основе полиномов 3-й степени; *б* – на основе полиномов 5-й степени

Приводим программу для рис. 63:

```

COMMON/BENKOT/ACOM(11), AMAT(42), IND(40), AAA(84)
DIMENSION A(40), B(40), C(40), D(40), E(40)
CALL SBROS
DATA A/30., 70., 40., 80., 60., 70., 90., 80./
CALL POLIN3 (A, 3,0)
CALL VP
CALL POLIN3 (B, 1, 3)
CALL VP
DATA B/20., 40., 200., -50., 60., 60., 200., 0./

```

```

B(7) = 30
CALL POLIN3(B, 1, 3)
CALL VP
B(3) = 10
B(4) = -2.5
CALL POLIN3(B, 1, 3)
CALL VP
CALL POLIN3(B, 1, 4)
CALL VP
CALL POLIN3(E, 1, 3)
CALL VP
DATA E/30., 20., 90., 90., 30., 20., -90., 90./
CALL MPPOL3(C, 6, 0)
DATA C/30., 10., 10., 20., 10., 50., 20., 70., 15., 75., 10., 90./
CALL MPPOL3(D, 4, 2)
DATA D/50., 10., 100., 0., 70., 20., 100., 100., 80., 50., 0., 10.,
* 95., 70., 10., 0./
CALL MPPOL3(D, 4, 3)
DO 1 J=1, 7, 2
1 CALL OK(A(J), A(J+1), 0.7)
DO 2 J=1, 11, 2
2 CALL OK(C(J), C(J+1), 0.7)
DO 3 J=1, 13, 4
3 CALL OK(D(J), D(J+1), 0.7)
CALL STP
END

```

В более ответственных случаях составные линии можно строить из дуг параметрических полиномов пятой степени. Здесь обеспечивается гладкость сопряжения второго порядка. Одна из версий подпрограммы MPOL5, работающей совместно с подпрограммой POLIN5 черчения одной дуги, компенсирует неравномерность расстояний между опорными точками. Специальные модификации подпрограмм POLIN5 и MPOL5 позволяют задавать в конечных точках дуг не только координаты и касательные векторы, но и радиусы кривизны.

Составную второго порядка гладкости можно образовать и из дуг обыкновенных полиномов третьей степени. Как это может быть? Ведь четырех параметров дуги простого полинома едва хватает на то, чтобы построить ее по координатам и направлениям в конечных точках. Дело здесь в том, что направления в опорных точках заранее не определяются, кривизна в точках сопряжения тоже не задается. Вместо этого требуется только равенство кривизн смежных дуг и соответствие направлений (аналитически эти условия сводятся к равенству первых и вторых производных). В итоге для построения в аналитической модели составной линии параметров достаточно; кроме координат во всех опорных точках удается

еще задать по одному дополнительному параметру в начальной и конечной точках линии. Это может быть нулевая кривизна или заданные направления. Сейчас принято линии этого и некоторых других типов называть иностранным словом "сплайн". Любопытно, что аналитическое описание такой линии напоминает описание формы гибкой рейки, закрепленной в нескольких точках (рис. 65). Рейка стремится распрямиться и принять форму с минимально возможной кривизной для всех точек линии. Такая рейка имеет достаточно плавную форму. Гибкие рейки используют даже вместо лекал при ручном вычерчивании кораблей, самолетов, автомобильных кузовов. В ЭВМ построение такой линии сводится к системе

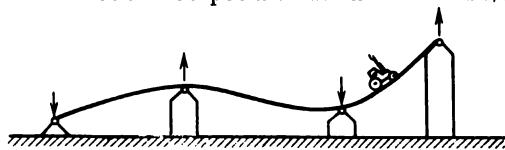


Рис. 65. Упругая рейка, закрепленная в нескольких точках

линейных уравнений, причем обычно число уравнений в четыре раза превышает число дуг (т.е. равно общему количеству определяемых коэффициентов). Различные уравнения соответствуют различным геометрическим условиям построения дуг — условиям прохождения дуг через опорные точки, условиям равенства первых производных для смежных дуг в точках сопряжения, условиям равенства вторых производных, так называемым краевым условиям, дополнительно задаваемым в первой и последних точках. Система решается стандартной подпрограммой или специальными более экономными способами. Большое число уравнений в системе для линий, содержащих значительное количество дуг, является недостатком этого метода.

В автоматизированном проектировании чаще применяются параметрические линии этого рода. Каждая из "составных" функций*), дающих значения одной и другой координаты, строится как описано выше. При этом дополнительно возникает вопрос о выборе значений производящего параметра в опорных точках. Обычно в качестве параметра принимается текущая длина ломаной, проведенной через эти точки.

В системе Алграф двумерные параметрические линии этого типа вычерчиваются подпрограммой PLNM3:

CALL PLNM3(A, NT, B, NB, IM)

Параметры A, NT, IM имеют тот же смысл, что раньше. B — это рабочий массив, передаваемый из вызывающей программы и используемый для составления матрицы системы уравнений и хранения коэффициентов дуг. Для контроля переполнения в переменной NB задается длина массива

*) Под составной функцией подразумевается множество функций, описывающих все дуги составной линии.

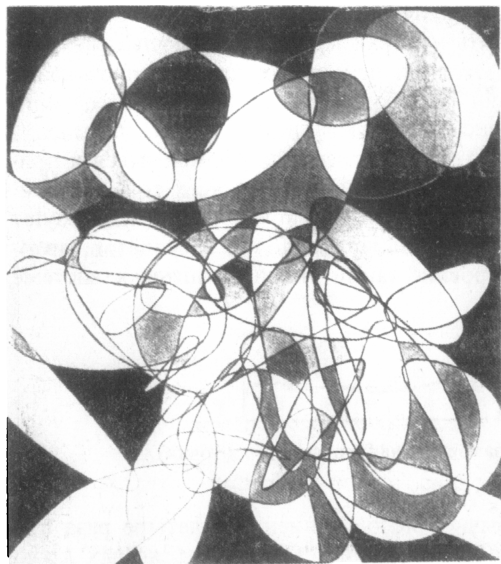


Рис. 66. Стохастический рисунок из замкнутых линий типа сплайн



Рис. 67. Гладкие составные линии в художественном проектировании

ва. Из модификаций работы программы отметим нулевую (построение незамкнутой линии) и первую (замкнутая кривая). Незамкнутая кривая в конечных точках имеет нулевую кривизну.

На рис. 66 показан рисунок из замкнутых линий этого типа, которые строились по трем или четырем точкам, выбранным с помощью подпрограммы – генератора случайных чисел (например, RANDU из пакета научных подпрограмм на фортране). Раскраска получившихся областей выбиралась из нескольких цветов (или тонов) тоже случайным образом, однако при наложении областей друг на друга цвет (тон) определялся по специальному алгоритму.

На рис. 67, а–д показаны контуры автомобильных кузовов. Здесь также есть составные кривые, построенные по разным алгоритмам.

Множества линий одного типа, построенные по определенному алгоритму, иногда бывают весьма выразительны. В программном отношении получение семейства обычно связано с образованием цикла, в котором чертится линия и каким-то образом изменяются ее параметры или условия ее построения с тем, чтобы следующая линия была уже несколько иной. Оставаясь пока в рамках двумерной геометрии, вспомним, что в пространстве семейство линий (если считать, что количество линий бесконечно велико) образует поверхность. Ниже мы приведем несколько примеров формирования семейств.

4.1. Нормали и касательные

Если на линии взять последовательность точек и в каждой из них построить касательную к линии или нормаль, может получиться интересное семейство прямых. Рисунок может быть похож на те, что приводились в разд. 2.5. На этих рисунках отчетливо видны огибающие семейств прямых. Если семейство прямых состоит из нормалей к некоторой кривой, то для огибающей есть специальное название — *эволюта*: тогда исходную кривую называют *эвольвентой* (по отношению к своей эволюте). Если же семейство состоит из касательных к некоторой кривой, то огибающей этого семейства, конечно же, будет исходная кривая.

На рис. 68 и 69 нанесены нормали к эллипсу и к эпитрохоиде с петлей (так называемой улитке Паскаля). Попробуйте проверить с помощью циркуля, что эволюта является геометрическим местом центров кривизны линии.

Как были построены нормали, например, к эллипсу? Ведь аналитически определить их не так просто! С применением дискретно-численного способа вычислялись координаты двух близко расположенных точек эллипса и затем их разности — компоненты вектора-хорды. Приблизительно считали этот вектор касательным. Чтобы получить проекцию нормального вектора

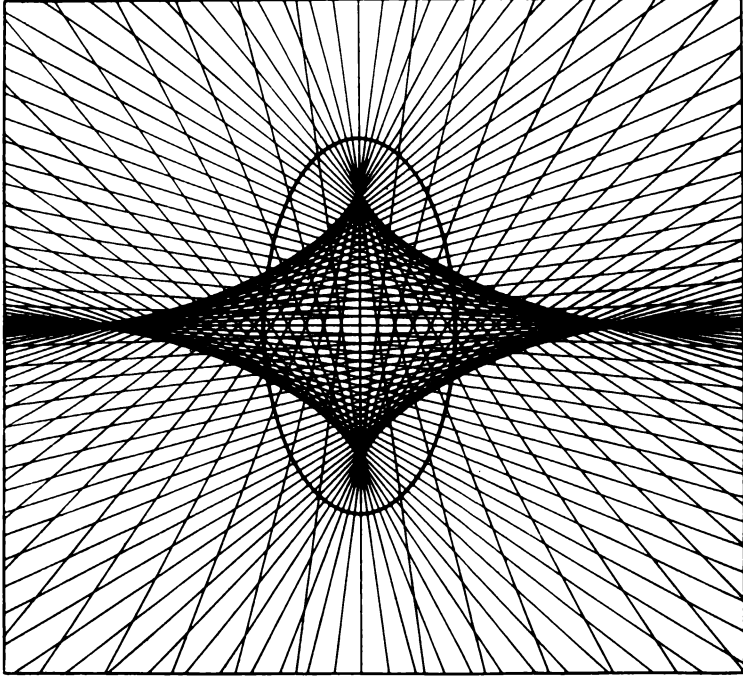


Рис. 68. Нормали, построенные к эллипсу, и образование огибающей (эволюты)

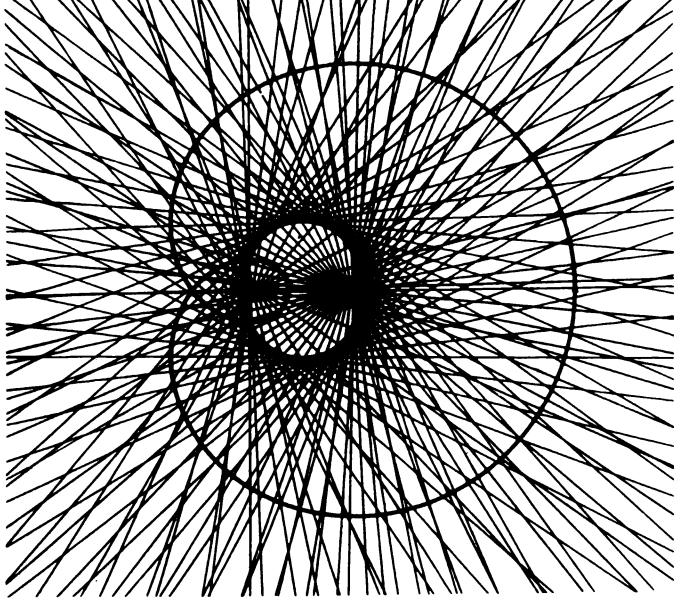


Рис. 69. Семейство нормалей к кардиоидной линии

на оси x и y , надо проекции касательного вектора поменять местами и у одной из них изменить знак. Вектор можно привести к единичной длине, разделив каждую его проекцию на квадратный корень из суммы квадратов обеих проекций. Затем легко вычислить проекции вектора любой длины, имеющего то же направление. Прибавляя величины проекций к координатам рассчитанной точки эллипса, получаем координаты начала отрезка, нормального к эллипсу. Вычтя из координат величины проекций, можно получить конец отрезка. Такую операцию по построению нормального отрезка к линии повторяем заданное число раз для последовательных точек этой линии.

4.2. Эквидистанты

Если на данном расстоянии от некоторой прямой, на всевозможных ее нормалях, по одну сторону от этой прямой, откладывать точки, то все эти точки будут лежать на прямой, параллельной данной. Аналогично можно поступать и для кривых. Проведем всевозможные нормали к данной кривой (в одну и ту же сторону от нее) и будем откладывать на них точки на фиксированном расстоянии от точек, из которых проведены нормали; получим линию, которая называется *эквидистантой* данной кривой. Практически к любой кривой можно построить множество эквидистант, соответствующих разным расстояниям и двум сторонам исходной линии (расстояние до эквидистанты можно брать со знаком: для положительных расстояний эквидистанты располагать с той стороны линии, куда направлены положительные нормальные векторы). Эквидистанты широко используются в технике; дело в том, что траектория центра окружности, катящейся по линии, как правило является эквидистантой этой линии. Исключения составляют участки, где радиус кривизны линии меньше радиуса подвижной окружности, а центр окружности расположен с той же стороны от линии, что и центры кривизны для точек линии на этом участке. Здесь у математической эквидистанты образуются особые участки (петли), по которым центр реальной окружности не сможет пройти. На рис. 70,а показана схема механизма с кулачком и роликовым толкателем. Заметьте, ролик касается кулачка не на продолжении оси толкателя! На рис. 70,б — изображение детали в виде пластины, которая вырезается на фрезерном станке. Цилиндрическая фреза режет боковой поверхностью, а движение ее центра (оси) должно происходить по эквидистанте к обрабатываемому контуру!

Аналитически определять эквидистанты к кривым (если не иметь в виду окружностей) обычно нелегко. Даже для эллипса эквидистанта — кривая восьмого порядка! Поэтому в машинной геометрии строить эквидистанты предпочитают дискретно-численным способом. Этот способ близок к описанному выше способу построения семейства нормалей. На линии берут последовательные точки, в них строят нормальные векторы заданной длины и через полученные концы этих векторов проводят ломаную, ко-

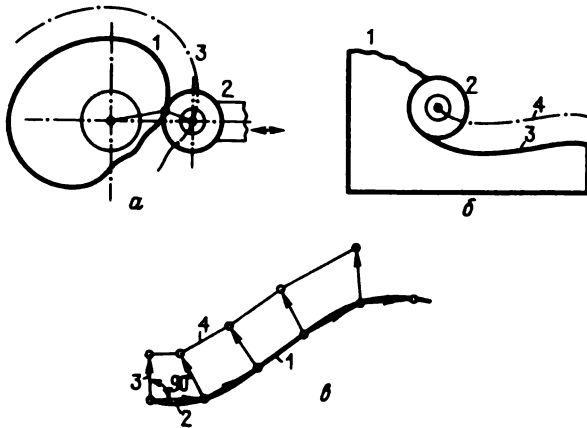


Рис. 70. Применение эквидистант: *a* – при расчете кулачкового механизма (1 – кулачок, 2 – роликовый толкатель, 3 – эквидистанта к профилю кулачка, определяющая движение центра ролика); *б* – для обработки деталей вращающейся фрезой (1 – заготовка, 2 – фреза, 3 – обрабатываемый профиль, 4 – эквидистанта к профилю, по которой движется центр фрезы); *в* – упрощенный численный способ построения эквидистант (1 – исходная кривая, 2 – вектор-хорда, аппроксимирующая касательную к кривой, 3 – вектор, нормальный к хорде, 4 – ломаная, аппроксимирующая эквидистанту)

торая и будет приближенной моделью эквидистанты. Приведем текст максимально упрощенной подпрограммы построения эквидистанты к кривой, которая уже задана достаточно плотной координатной моделью:

```

SUBROUTINE EKV(A, P)
  DIMENSION A(2, 100)
  IA=1
  1 IF(A(1, IA).GE.10000.) GOTO 10
  DX=A(1, IA+1)-A(1, IA)
  DY=A(2, IA+1)-A(2, IA)
  S=SQRT(DX**2+DY**2)
  X=A(1, IA)-DY*P/S
  Y=A(2, IA)+DX*P/S
  CALL LINX(X, Y)
  CALL PERO
  IA=IA+1
  GOTO 1
  10 CALL VP
  RETURN
  END

```

Здесь массив *A* содержит координатную модель исходной кривой, заканчивающуюся числом-кодом 10000., параметр *P* определяет, с учетом знака, расстояние до эквидистанты.

На практике алгоритмы построения эквидистант приходится усложнять. Для повышения точности эквидистанты при той же исходной координатной модели применяют круговую или параболическую интерполяцию к каждому трем последовательно взятым точкам координатной модели и построение нормали к этой интерполирующей линии в ее средней точке (рис. 71,а).

При построении эквидистанты к негладким линиям, например, к ломаным, надо считаться с особыми ситуациями, возникающими около точек излома (рис. 71,б). Когда угол, образованный точкой излома (если смотреть со стороны эквидистанты), выпуклый, на эквидистанте образуется

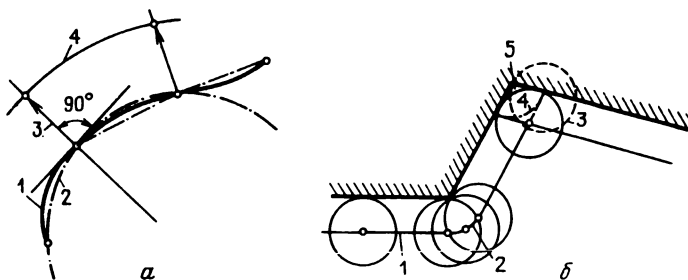


Рис. 71. а – Уточненный способ построения дуги эквидистанты (1 – исходная кривая, 2 – парабола или дуга окружности, построенная по трем точкам кривой, 3 – вектор, нормальный к дуге в средней из трех точек, 4 – эквидистанта); б – эквидистанта к негладкой линии (1 – эквидистанта, 2 – круговой участок эквидистанты, 3 – физически недостижимое положение катящейся окружности, 4 – петля теоретической эквидистанты, 5 – область, недоступная для обработки фрезой)

дуга окружности с центром в точке излома. Если угол "вогнутый" – либо образуется самопересекающийся участок эквидистанты, с дугой окружности и точки возврата, либо, если отсечь показанные штрихами части (эти части, вообще говоря, расположены к некоторым точкам исходной кривой ближе, чем определено заданным расстоянием) – эквидистанта будет также иметь точку излома. В этом случае к некоторым точкам исходной линии не будет соответствующих точек эквидистанты. Это, в частности, приводит к тому, что внутренние углы контура круглой фрезой обработать невозможно. Сходные ситуации встречаются, когда на исходной линии последовательно расположены несколько маленьких отрезков.

В основной комплект системы Алграф включена подпрограмма построения эквидистанты для негладкой линии, не производящая, однако, сложных отсечений. С ее помощью получен рис. 72. Вот текст главной программы для него:

```
DIMENSION A(40)
CALL SBROS
```

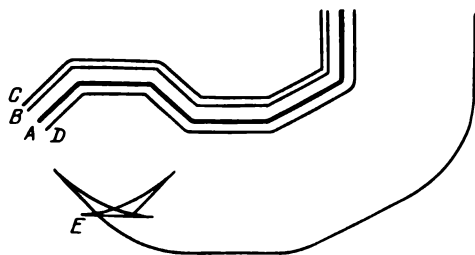


Рис. 72. Семейство эквидистант, построенных к исходной линии *A*; эквидистанта *E* содержит петли

```
DATA A/0., 0., 10., 10., 30., 10., 40., 0., 60., 0., 80., 10., 80.,
* 30., 10000./
CALL LOM(A, 99, 2)
CALL NPERA(1)
CALL EKVID(A, 4., 0)
CALL EKVID(A, 6., 1)
CALL EKVID(A, -3., 1)
CALL EKVID(A, -35., 1)
CALL STP
END
```

Последний параметр при обращении к EKVID – индекс модификации. Если он задан нулевым – эквидистанта строится в виде ломаной, если индекс единичный – в эквидистанту включаются дуги окружностей.

4.3. Кинематические кривые

Техника, механика насыщены различного рода шарнирными механизмами, трансмиссиями, кулачками и т.д. Точки движущихся частей описывают различные траектории, иногда довольно причудливые. Да и сами перемещающиеся детали нередко приходится чертить в различных положениях. Так, при проектировании автомобилей довольно сложная задача – вычертить в различных возможных положениях колеса машины, чтобы затем сконструировать колесные кожухи или вырезы в боковине кузова. Приведем простой пример условного изображения в виде плоской схемы шарнирно-рычажного механизма крепления сельскохозяйственного орудия к трактору. На рис. 73,а показана исходная схема механизма. Движение задается перемещением плунжера насоса *A*, в связи с чем изменяется длина отрезка *BC*. Положение неподвижных шарниров задано, координаты подвижных шарниров для каждой заданной длины элемента *BC* вычисляются последовательно. Обычно используется построение треугольника по трем сторонам. Так, в треугольнике *FAB* все стороны известны, вершины *F* и *A* определены. Чтобы определить положение вершины *B*, надо найти точку пересечения окружностей известных радиусов с центрами в точках *A* и *F*. Уже упоминалась типовая подпрограмма SCOK, решающая такую задачу. После определения точки *B* точку *C* нетрудно найти "по правилу рычага".

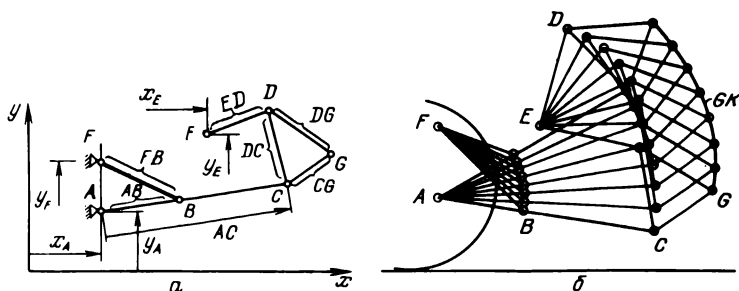


Рис. 73. Схема кинематики рычажного механизма и кинематическое образование кривой

Затем находится положение точки D (опять как пересечение двух дуг) и наконец точки G . В цикле подобные вычисления производятся для последовательных положений механизма. Каждый раз рычаги вычерчиваются с помощью известных подпрограмм. Кроме того, координаты последовательных положений шарнира G запоминаются в массиве. В конце работы точки этого массива соединяются ломаной — и кинематическая кривая GK вычерчена (рис. 73,б).

Чтобы в различных положениях показать более или менее сложные контуры деталей, чтобы показать положение различных предметов в процессе их движения, пользуются преобразованиями координат, на которых мы остановимся позднее.

4.4. Пропорциональные кривые

Линия движется, постепенно изменяясь, превращается в другую линию, затем в третью... Может быть, это движение тела змеи или изменение контура плода в процессе его роста? А возможно — изображение марсианина, который в фантастическом мультфильме превращается в земное существо (рис. 74).

Во всех этих случаях имеются как минимум две заданные линии и надо построить семейство "промежуточных", которое моделировало бы "постепенное превращение". Простейший способ построения таких кривых, которые условно назовем пропорциональными, описан ниже.

Наметим на обеих исходных линиях одинаковое количество точек. Пусть это заданные в исходных координатных моделях опорные точки либо вновь построенные точки на линиях. В последнем случае их можно найти, разделив обе линии на одинаковое число равных частей.

Теперь точки с одинаковыми номерами соединяем отрезками. Вводим параметр t , соответствующий времени или фазам превращения линии. Пусть для первой исходной линии $t = 0$, а для второй — $t = 1$. Для некоторой промежуточной линии зададим значение $t = t_i$. Теперь каждый отрез-

могут образовываться петли и особые точки. Если на исходных линиях есть изломы — на промежуточных линиях эти изломы также "отпечатываются", причем в общем случае их количество равно сумме изломов на исходных линиях. Впрочем, особым образом выбирая опорные точки на линиях, можно управлять преобразованиями так, чтобы изломы первой линии "переходили" в изломы второй линии.

Ранее показанные контуры автомобилей (см. рис. 67) были получены этим способом. Исходными фигурами здесь были контуры верхнего и нижнего автомобилей. Эти рисунки содержат, конечно, много построенных описанным способом линий. Этот же способ используется в производстве мультфильмов. В традиционной работе он называется "фазовка". Способ можно различным образом усложнять и модифицировать. Сочетая такое преобразование с преобразованиями координат, можно получить криволинейные пути точек линии в процессе ее видоизменения. С другой стороны, исходных линий может быть не две, а несколько. Тогда вместо одного параметра t вводится несколько, и они играют роль весов, присваиваемых каждой из исходных линий. Этим путем можно нарисовать, например, гибрид яблока, груши, банана и еще какого-либо фрукта. Не правда ли, заманчивая возможность для художников-конструкторов, выбирающих оптимальные решения из предложенных вариантов?

4.5. Кривые влияния

Допустим, кривая равномерно "заряжена" в каком-то смысле и ее "заряд" влияет на окружающее пространство, образуя некое "поле". Чем ближе к кривой, тем потенциал этого поля выше. Вспоминая, что говорилось о возможности разделения плоскости линией на положительные и отрицательные области, условно примем, что потенциал с двух сторон от линии имеет разный знак. Для определения величины потенциала в точке плоскости можно подсчитать расстояние от этой точки до линии. Тогда, если задаться фиксированным значением потенциала, точки плоскости с таким значением потенциала выстроятся в эквидистанту, проведенную около линии. Но ниже будет описан иной способ оценки близости точки к исходной кривой и построение "линий равного влияния"*).

Если уравнение окружности записать в неявном виде: $(x - x_c)^2 + (y - y_c)^2 - R^2 = 0$, и подставить в левую часть координаты какой-либо точки, не принадлежащей этой окружности, левая часть уравнения будет равна некоторой положительной или отрицательной величине. Пусть эта величина и определяет "степень влияния". Она будет нулевой для точек самой линии и возрастать по мере удаления от линии (при необходимости можно взять обратное значение этой величины и получить эффект падения потенциала по мере удаления от кривой). Итак, для получения линий "равного влияния" мы будем брать неявное уравнение исходной линии и в правой

*) Термин условный.

части заменять нуль на некоторую константу, определяющую "потенциал". Если такую операцию совершить с уравнением окружности, получим также окружность, причем концентрическую по отношению к предыдущей, т.е. эквидистантную. Для других кривых получаются уже, как правило, не эквидистанты. Вообще говоря, форма и математическое описание "линий влияния" зависят от того, в каком виде записано уравнение исходной кривой. Подумайте, каковы будут линии равного влияния для прямой, синусоиды, линии полинома.

Построим линии влияния для исходной кривой особого вида. Возьмем окружность и пересекающую ее прямую. Уравнения обеих линий известны.

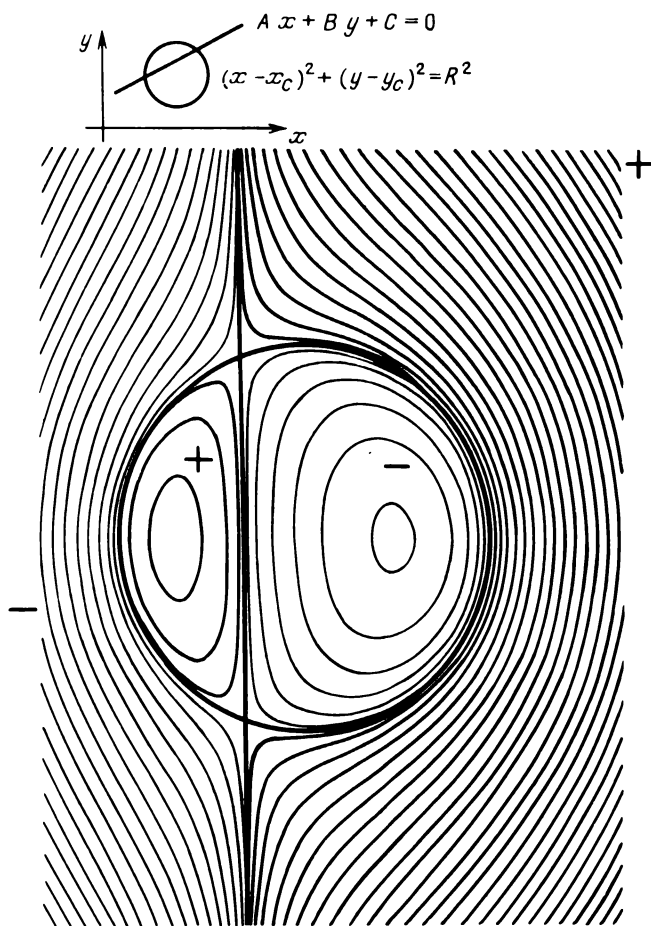


Рис. 76. Линия третьего порядка из окружности и прямой и "линии влияния" для окружности и прямой

Перемножив левые части обоих уравнений и приравняв произведение нулю, получим следующее уравнение:

$$[(x - x_c)^2 + (y - y_c)^2 - R^2] (Ax + By + C) = 0.$$

Можно произвести алгебраические преобразования и убедиться, что получилось уравнение линии третьего порядка. Тем не менее оно описывает сразу обе исходные фигуры (почему?).

Теперь в правой части уравнения заменяем нуль на значение "потенциала" D и получаем уравнение "линии влияния". Как вычертить линию, имея ее уравнение, уже обсуждалось. Перенеся параметр D в левую часть, можно получить уравнение в неявном виде, можно попытаться разрешить уравнение относительно y и затем искать корни получившегося уравнения третьей степени. Можно воспользоваться стандартными подпрограммами, которые численными способами определяют корни неявных уравнений.

На рис. 76 показано семейство "линий влияния" для нашей составной фигуры. Области, на которые разделена плоскость исходными линиями, в шахматном порядке соответствуют положительным и отрицательным значениям "потенциала" (как на шахматной доске при переходе от клетки к клетке цвет меняется на противоположный, так и здесь однократное пересечение исходной линии соответствует изменению знака).

Описанный способ может иметь некоторые неожиданные применения. Например – для построения аппроксимирующих линий с использованием метода наименьших квадратов в случаях, когда искомая линия замкнута или имеет сложную форму (см. рис. 61).

ПРЕОБРАЗОВАНИЯ КООРДИНАТ

Преобразования координат широко используются в машинной графике и геометрии. Многие из того, что при ручной работе мы, не задумываясь, делаем как само собой разумеющееся – в автоматизированных системах выполняется с помощью преобразований. Преобразования используются в разных целях: чтобы различные части объекта можно было описывать в наиболее удобных для этого координатных системах; чтобы типовые и повторяющиеся части объекта можно было располагать в разных местах и положениях на чертеже и в пространстве, в том числе с использованием циклов; чтобы без повторной кодировки можно было получать симметричные части объекта; для направленной деформации фигур, тел и их частей; для изменения масштаба чертежа, построения проекций пространственных образов и др.

С аналитической точки зрения преобразования – это пересчет значений координат по определенным формулам, с геометрической точки зрения на эту операцию можно смотреть как бы с двух позиций [4]. С "пассивной" позиции описываемой координатами объект представляется нам неизменным, но одну систему отсчета мы заменяем на другую и ищем координаты в новой системе. С "активной" позиции координатная система одна, но сам объект перемещается, деформируется. Правда, и в этом случае с объектом можно "скрепить" первичную систему и продеформировать ее таким же образом. Эта первичная координатная система может стать уже косоугольной или даже криволинейной.

Вообще говоря, в арсенале геометров много различных типов преобразований. Лишь некоторые из них оставляют прямые линии прямыми. Кроме того, преобразования могут быть однозначными и неоднозначными, взаимно-однозначными, а иногда и "вырожденными". В отдельных случаях не все точки исходной плоскости могут быть преобразованы, либо отображенные точки заполняют не всю вторичную плоскость (аналогично в пространстве). Для примера на рис. 77 показано некоторое специальное преобразование плоскости, когда в определенной ее части образуется "наложе-

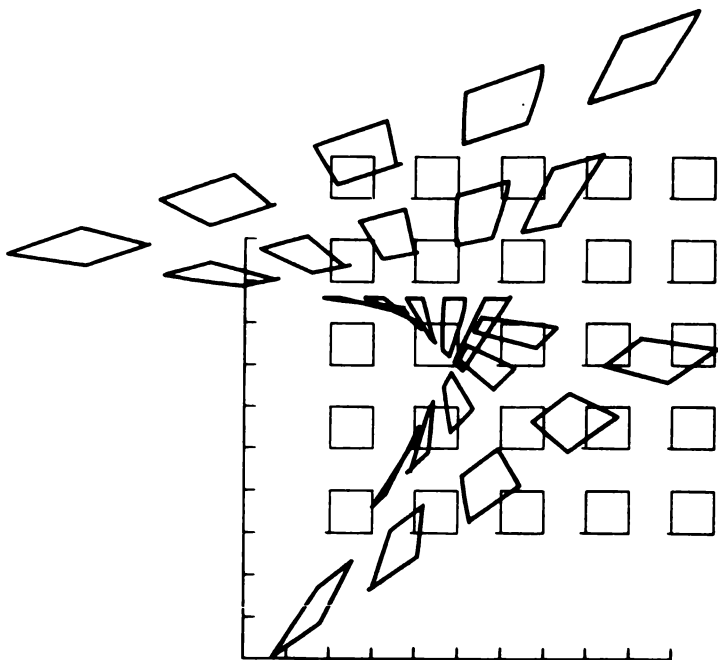


Рис. 77. Образование наложений при сложном преобразовании (на первичной плоскости изображены оси прямоугольной координатной системы и квадратная сетка)

ние”, а в другой ее части нет образов преобразованных точек первичной плоскости*). Линия, разделяющая плоскость на эти две части, играет в преобразовании особую роль.

Двумерные и трехмерные преобразования различаются по типам, соответственно разный вид имеют и их уравнения. Желательно, чтобы уравнения явно выражали значения вторичных координат:

$$x_2 = f(x_1, y_1), \quad y_2 = \varphi(x_1, y_1).$$

Для трехмерных преобразований в пространстве должно быть три уравнения.

Наиболее распространены так называемые аффинные преобразования — двумерные и трехмерные. Основные их геометрические свойства: прямые линии после преобразования остаются прямыми, параллельные прямые — параллельными, деление отрезка в некотором отношении сохраняется. При трехмерных преобразованиях плоскости также остаются плоскостями и параллельные плоскости — параллельными. За исключением вырож-

*) Исходный объект (“таблица” квадратиков) вычерчен тонкой линией.

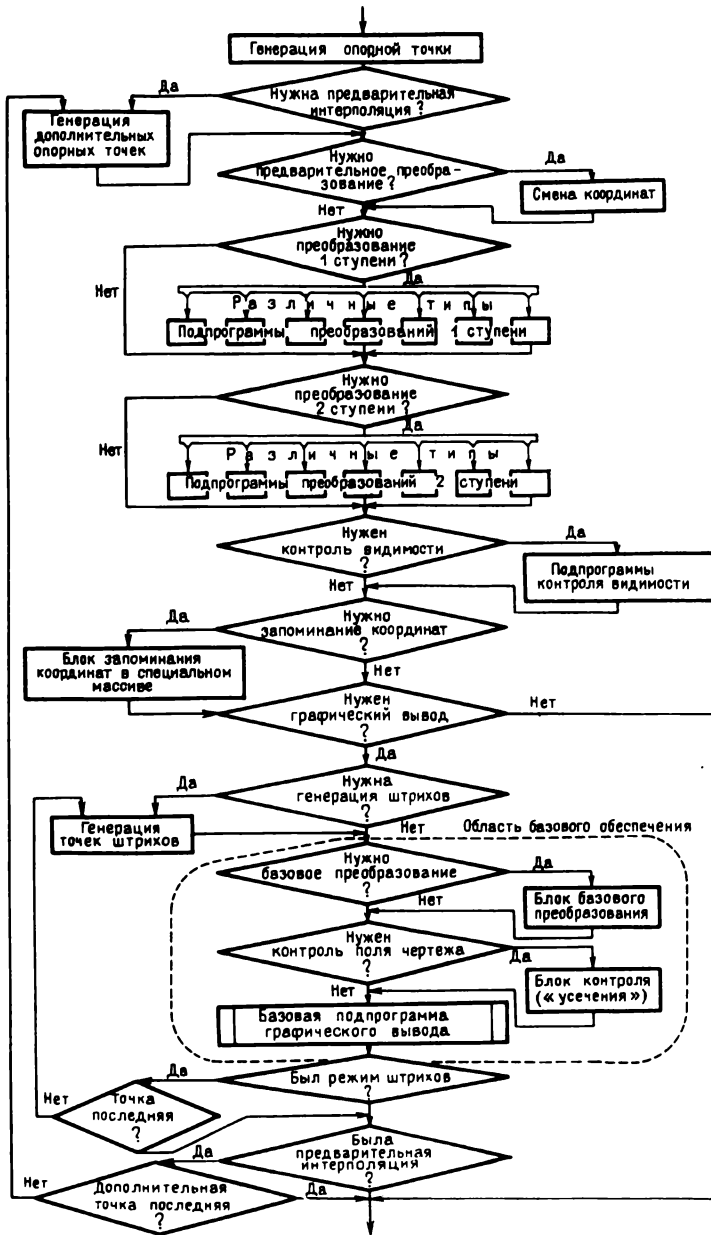


Рис. 78. Схема действия ядра графической системы

денных случаев все точки первичной плоскости (или пространства) преобразуются и заполняют соответственно все точки вторичной плоскости (пространства). Исключительная роль этих преобразований в машинной геометрии и графике объясняется наличием таких частных случаев, как перенос, поворот, подобие, симметрия...

Коснемся программных вопросов. Во-первых, последовательность действий при выполнении преобразований может быть различной, и это различие напоминает разницу между трансляцией и интерпретацией программы. Исходная координатная модель (а иногда и аналитическая модель) может преобразовываться заданным образом, порождая новую КМД (АМД); затем эта новая модель может быть как-то преобразована еще раз и т.д. Какая-либо "последняя" модель может быть выведена на графическое устройство. С другой стороны, процесс может быть организован так, что каждая точка, рожденная программами и подпрограммами генерации образов, последовательно проходит все заданные преобразования, а окончательные координаты, если надо, передаются графическому устройству (или хотя бы базовой подпрограмме).

В системе Алграф стандартным является второй из описанных процессов, хотя при необходимости можно преобразовывать и координатные модели и можно пользоваться так называемой подсистемой запоминания координат. На рис. 78 показана схема работы ядра системы при графическом выводе. Поясним отдельные блоки. Блок предварительной интерполяции появился в связи с криволинейными преобразованиями, при которых отрезки могут искривляться. Однако обычно преобразуются лишь конечные точки отрезков, а их прямолинейная форма обеспечивается графическим устройством; чтобы отрезок "с заданной точностью" искривлялся, его надо разбить на мелкие части и подвергнуть преобразованию полученные промежуточные точки. Это и делает блок предварительной интерполяции, когда переменная-индекс AAA(15) из блока BENKOT (см. разд. 2.3) задается ненулевой. Значение же этой величины принимается за шаг интерполяции (в первичной системе).

На рис. 78 показаны блоки четырех ступеней преобразования. Предварительное преобразование выполняет ограниченные функции смены координатных осей и, как будет показано ниже, используется при работе с трехмерными объектами. Оно "управляется" значением индекса IND(4) (см. табл. 4).

Преобразования первой ступени наиболее активно используются для различных целей и бывают разных типов; некоторые типы сами содержат несколько "подступеней". Тип определяется значением индекса IND(5), а коэффициенты преобразования хранятся обычно в массиве блока BENKOT AMAT1. Для дополнительной подступени отведен массив AMAT0. Предварительное преобразование и преобразование первой ступени переводят координаты из *частной координатной системы* (ЧКС) в общую, или *основную* (ОКС). В блоке BENKOT для хранения координат текущей

Таблица 4

Типы преобразований в системе Алграф

Ступень	Значение индекса и задание на языке фортран	Код на языке алграф-Р	Тип преобразования	Используемые поля
1	2	3	4	5
0	IND(4) = 0	XYZ	нет смены осей	нет
0	IND(4) = 1	XZY	смена осей Y и Z	нет
0	IND(4) = -1	ZYX	смена осей X и Z	нет
1	IND(5) = 0 CALL PREOB1(0)	ПРБ1	нет преобразования	нет
1	IND(5) = 1 CALL PREOB1(1)	ПРБЛ2	аффинное, двух координат	AMAT1
1	IND(5) = 2 CALL PREOB1(2)	ПРБЛ3	аффинное, трех координат	AMAT1
1	IND(5) = 3 CALL PREOB1(3)	ПРБ1 (3)	двойное аффинное, двух координат	сначала AMAT0 затем AMAT1
1	IND(5) = 4 CALL PREOB1(4)	ПРБ1 (4)	двойное аффинное, трех координат	то же
1	IND(5) = 5 CALL PREOB1(5)	ПРБ1 (5)	из полярной в декартову, затем аффинное, двух координат	то же
1	IND(5) = 6 CALL PREOB1(6)	ПРБ1 (6)	из цилиндрической в декартову, затем аффинное, трех координат	то же
1	IND(5) = 7 CALL PREOB1(7)	ПРБ1 (7)	двойное аффинное, трех координат	сначала AMAT1, затем AMAT0
1	IND(5) = 8 CALL PREOB1(8)	ПРБ1 (8)	из специальной полярной в декартову, затем аффинное, двух координат	сначала AMAT0, затем AMAT1
1	IND(5) = 9 CALL PREOB1(9)	ПРБ1 (9)	из спец. цилиндрической в декартову, затем аффинное, трех координат	то же
1	IND(5) = 10 CALL PREOB1(10)	ПРБ1 (10)	из сферической в декартову и аффинное, трех координат	то же
1	IND(5) = 12 CALL PREOB1 (12)	ПРБ1 (12)	по подпрограмме пользователя с именем PRBS1	?
2	IND(6) = 0 CALL PREOB2(0)	ПРБ2	нет преобразования	нет

Т а б л и ц а 4 (окончание)

1	2	3	4	5
2	IND(6) = 1 CALL PREOB2(1)	ПРБАКС	аксонометрическое	АМАТ2
2	IND(6) = 2 CALL PREOB2(2)	ПРБПЕР	перспективное (центральная линейная проекция)	АМАТ2
2	IND(6) = 3 CALL PREOB2(3)	ПРБ2 (3)	аксонометрическое короткое	АМАТ2
2	IND(6) = 4 CALL PREOB2(5)	ПРБ2 (5)	по подпрограмме пользователя с именем PRBS3	?

(последней) точки в ЧКС отведены ячейки XOR, YOR, ZOR, а для ОКС – переменные XOS, YOS, ZOS. По мере необходимости при этих преобразованиях возникают также *промежуточные координатные системы* (ПКС). Преобразование второй ступени обычно используется для построения проекций или изменения всего чертежа. Его тип указывается индексом IND(6), коэффициенты сохраняются на рабочем поле АМАТ2, а координаты из ОКС переводятся в *координатную систему чертежа* – КСЧ, для которой в блоке ВЕНКОТ отведены ячейки XD, YD. При построении проекций образуется также *пространственная система, связанная с чертежом* – ПРКС, о которой речь будет идти ниже. Наконец, ограниченные средства преобразований имеются в базовом обеспечении. При работе с терминалом они, в частности, нужны, чтобы на рабочем поле экрана можно было отображать нужную часть картинной плоскости.

Отметим возможность нестандартных преобразований, выполняемых по подпрограммам пользователя (которые готовятся по определенным правилам).

На схеме видно, что графический вывод можно заблокировать (для этого индексу IND(10) надо придать нулевое значение). Вместо него (или параллельно с ним) можно подключить запоминание координат генерируемых точек в массиве АААА блока общих переменных ЗАРКОТ:

COMMON/ЗАРКОТ/ІАМАХ, ІАТЕК, АААА (nn)

Пользователь в своей программе может объявить этот блок с нужной ему длиной nn этого массива. Контрольной переменной ІАМАХ надо придать значение, не большее nn, переменная ІАТЕК – счетчик элементов массива (указывает следующий "свободный" элемент). В начале запоминания обычно ІАТЕК = 1. Режимом запоминания "ведает" индекс IND(8). Различные его ненулевые значения соответствуют запоминанию

двух или трех координат в ЧКС, ОКС, ПРКС или КСЧ. С помощью специальных подпрограмм в нужные моменты между координатами последовательных точек можно вставлять специальные числа-коды.

На рис. 78 видно, что в нужных случаях полученные в КСЧ точки могут быть проверены на принадлежность заданному прямоугольному полю чертежа, либо элементы изображения (отрезки) проверяются на предмет их видимости при построении проекций пространственных образов. Ядро системы может также генерировать штрихи, когда задан режим программной генерации штриховых или штрихпунктирных линий.

Описанная структура операций, выполняемых ядром, довольно сложна, но оказывается полезной при работе со сложными объектами. Другие графические системы имеют обычно более простые подсистемы преобразований, но и они могут содержать две-три ступени.

5.1. Аффинные преобразования двух координат

Основные свойства аффинных преобразований уже отмечались; здесь не рассматриваются их особенности, поясним лишь вид уравнений преобразования:

$$x_2 = a_{xx}x_1 + a_{xy}y_1 + a_x,$$

$$y_2 = a_{yx}x_1 + a_{yy}y_1 + a_y.$$

На фортране с использованием переменных из блока VENKOT запись выглядит так:

С использованием переменных из блока VENKOT:

$$\text{XOS} = \text{AMAT1}(1) * \text{XOR} + \text{AMAT1}(2) * \text{YOR} + \text{AMAT1}(3)$$

$$\text{YOS} = \text{AMAT1}(4) * \text{XOR} + \text{AMAT1}(5) * \text{YOR} + \text{AMAT1}(6)$$

Шесть коэффициентов преобразования (включая свободные члены) хранятся в поле AMAT1 по строкам, но часто их записывают в виде матрицы

$$\begin{bmatrix} a_{xx} & a_{xy} & | & a_x \\ a_{yx} & a_{yy} & | & a_y \end{bmatrix}.$$

Левая квадратная часть из четырех коэффициентов — та матрица, с помощью которой можно преобразовать компоненты векторов. На рис. 79 приведены несколько схем, иллюстрирующих общий случай аффинного преобразования и распространённые частные случаи. Первая схема показывает, как можно графически интерпретировать члены матрицы. Так, свободные члены — это координаты начала первичной системы (в данном случае — ЧКС) во вторичной КС, другие элементы — проекции единичных (базисных) векторов ЧКС, замеренные в ОКС. Последняя схема показывает композицию переноса и поворота; известно, что на плоскости это дает преобразование движения, которое является ортогональным и изометрическим, т.е. сохраняет неизменными расстояния между любыми точками

Операторы двумерных аффинных преобразований

Действие оператора	Параметры	Код на языке алграф	Оператор на фортране
Задание тождественной матрицы	Нет	MTI2	CALL MATI (10)
Пересылка матрицы в рабочем поле	A, B, C, D, E, F – шесть элементов матрицы	MATP2	CALL MATL2 (A, B, C, D, E, F)
Пересылка матрицы из массива в рабочем поле	AD – метка подмассива или имя массива длиной 6 чисел	MATR (610, AD)	CALL MATR (AD, 6, 1)
Пересылка матрицы с рабочего поля в массив	- " -	MATR (611, AD)	CALL MATR (AD, 6, 11)
Перенос ЧКС на вектор, заданный в ОКС	DX, DY – компоненты вектора в ОКС	PEREN, DX, DY	CALL PEREN(DX, DY, 0)
Перенос на вектор, заданный в ЧКС	DX1, DY1 – компоненты вектора в ЧКС	PEREN1, DX1, DY1	CALL PEREN(DX1, DY1, 1)
Перенос начала ЧКС в точку	X0, Y0 – координаты начала ЧКС в ОКС	PEREN2, X0, Y0	CALL PEREN(X0, Y0, 2)
Задание переноса и повернутого положения осей ЧКС; оси ЧКС ортогональны, масштабы по осям и их относительная ориентация те же, что в ОКС	X0, Y0, ALF – координаты начала ЧКС в ОКС и угол наклона оси X ЧКС к оси X _{ос}	POB1, X0, Y0, ALF	CALL POB1 (X0, Y0, ALF)

Т а б л и ц а 5 (окончание)

Действие оператора	Параметры	Код на языке алгебра	Оператор на фортране
Дополнительное вращение осей ЧКС вокруг точки, заданной в ОКС, на заданный угол	ХТ, УТ, ВТ – координаты центра вращения и угол поворота	ПОВБ, ХТ, УТ, ВТ	CALL POVB(ХТ, УТ, ВТ)
Масштабирование по осям ЧКС	ХМ, УМ – масштабные множители для двух осей	МАСШ, ХМ, УМ	CALL MASH(ХМ, УМ, 0)
Масштабирование по осям ОКС	ХМ1, УМ1 – масштабные множители для двух осей	МАСШ1, ХМ1, УМ1	CALL MASH(ХМ1, УМ1, 1)
Умножение матриц преобразования; результат образуется в рабочем поле	AD – метка подмассива или имя массива, где хранится матрица-множитель; K = 0, если исходная матрица на рабочем поле определяет первое преобразование, и K = 2, если она определяет второе преобразование	УМТ (K, AD)	CALL UMT(AD, K)

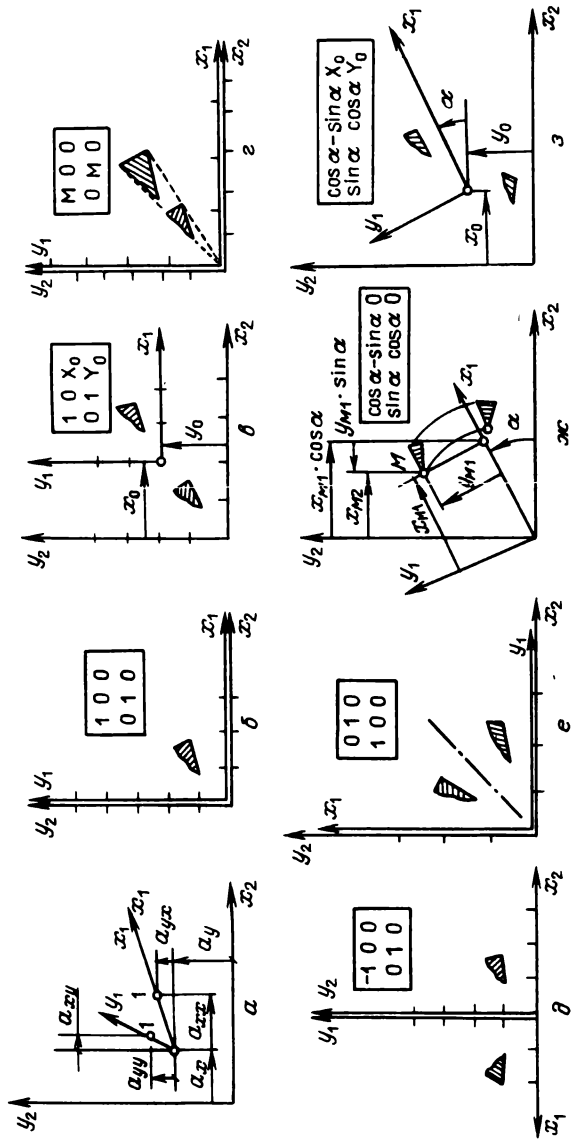


Рис. 79. Аффинные первичные системы в декартовой вторичной: а – косоугольная система, б – тождественная КС, в – преобразование параллельного переноса, г – преобразование поворота, д – симметрия относительно координатной оси, е – смена осей, ж – преобразование поворота, з – преобразование движения

объекта и взаимную перпендикулярность координатных осей. Аналитически это соответствует четырем условиям ортогональности-изометричности, наложенным на значения членов матрицы. Определитель матрицы равен единице. Есть, как известно, еще и изометрия второго рода, когда определитель равен -1 , а ориентация на плоскости меняется на противоположную ("правое" меняется на "левое"). В методике алгебра используется также понятие "ортоизометрическое преобразование в расширенном смысле". Это композиция обычного изометрического и подобного преобразований, т.е. здесь допускается равномерное изменения масштаба.

Согласно табл. 4 этот тип преобразования задается единичным значением индекса IND(5) (можно использовать специальную подпрограмму или код); кроме задания типа необходимо задать значения коэффициентов преобразования. В программе на фортране значения элементам массива AMAT1 можно придать непосредственно, но все же, если коэффициенты известны, удобнее воспользоваться подпрограммой MATL2 (см. табл. 5) или кодом MATL2, после которого через запятые записываются значения коэффициентов. Однако не всегда пользователь знает значения коэффициентов. Обычно применяют специальные операторы (подпрограммы и коды), формирующие или изменяющие желаемым образом матрицу преобразования. В табл. 5 приведены такие операторы, в том числе оператор, задающий *тождественную матрицу*, которая может быть использована для дальнейшей направленной модификации преобразований (такая матрица автоматически формируется и инициализирующими модулями системы).

Операторы *переноса* изменяют только свободные члены матрицы; таким образом, ранее заданная первичная система переносится параллельно себе с сохранением прежних направлений осей и масштабов по ним (рис. 80, а). "Новое" преобразование из ЧКС в ОКС будет переносом, только если предыдущее преобразование было переносом или тождественным преобразованием.

Операторы *масштабирования* также воздействуют на "предыдущее" преобразование; один из них (MACIII) умножает на первый заданный масштабный множитель элементы первого столбца матрицы, на второй – второго. Свободные члены не изменяются, и потому начало ЧКС остается на прежнем месте. Не меняются и направления осей ЧКС, но они как бы "растягиваются" или "сжимаются". Направления могут поменяться на противоположные, если масштабные множители отрицательны. Если "предыдущее" преобразование тождественно и оба масштабных множителя одинаковы – "новое" преобразование будет подобием. При тех же множителях и общем случае предыдущего аффинного преобразования отношением подобия будут связаны эквивалентные фигуры, полученные при "старом" и "новом" преобразованиях (рис. 80, б). Если оба множителя равны -1 , "новые" и "старые" фигуры будут симметричны относительно начала ЧКС (рис. 80, в). Симметрию относительно одной из осей ЧКС можно получить, задав один из множителей равным $+1$, а другой -1 , но только в том случае, если оси ЧКС, в их отображении на вторичную плос-

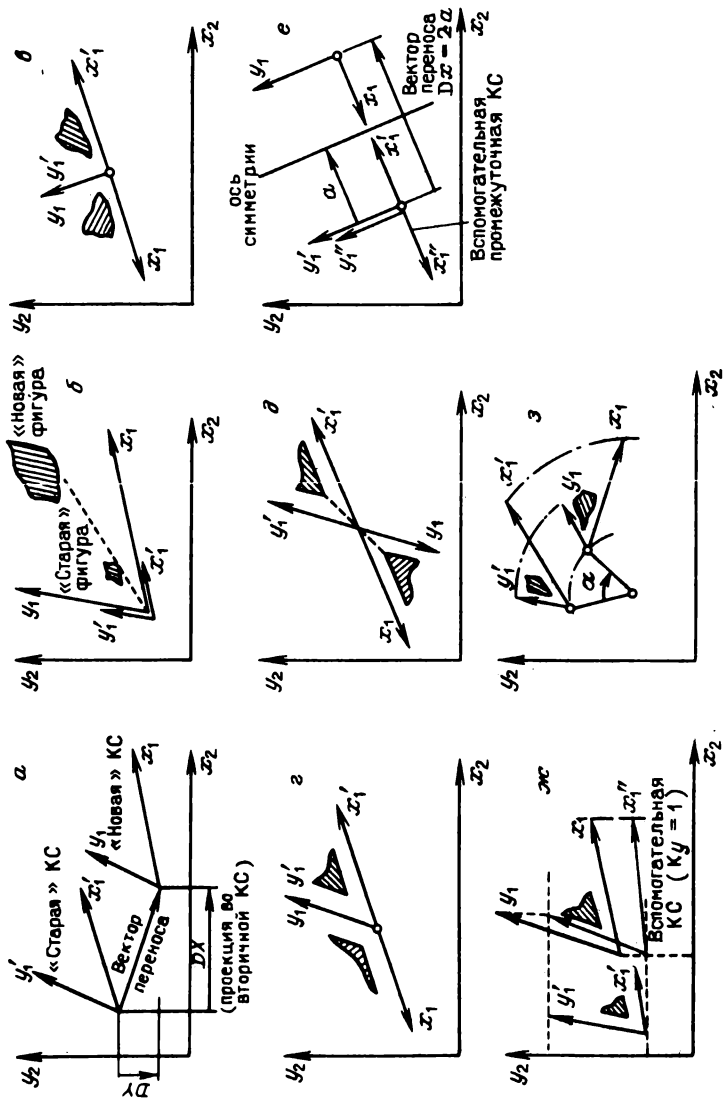


Рис. 80. Действия операторов переноса, масштабирования и поворотов в системе Алграф

кость, ортогональны (иначе получится "квазисимметрия"). Другой оператор масштабирования (МАСШ1) как бы растягивает или сжимает первичную плоскость в направлении вторичных координатных осей (рис. 80, ж). Начало ЧКС смещается, изменяются направления ее осей. В аналитическом отношении здесь на первый множитель умножаются элементы первой строки матрицы и на второй множитель – элементы второй строки. Этим способом легко получить симметрию относительно начала ОКС или ее осей.

Из вариантов *поворота* в табл. 5 приведены два. Оператор ПОВ1 заново формирует всю матрицу, игнорируя предыдущее преобразование. Новое преобразование из ЧКС в ОКС будет движением – изометрическим, ортогональным преобразованием (рис. 80, з). Оператор удобен для размещения на чертеже типовых фрагментов, стандартных деталей, закодированных в "своих" системах отсчета. Чтобы получить изображение детали в симметричном виде или с изменением масштаба, после оператора поворота сразу можно применить оператор МАСШ. Другой оператор – ПОВБ – осуществляет "дополнительное вращение". Первичную плоскость с "предыдущей" ЧКС он поворачивает как жесткое целое вокруг заданной в ОКС точки (рис. 80, з). Масштабы на осях ЧКС при этом не меняются, не меняется и угол между ними.

Известно, что двумерное аффинное преобразование можно задать тремя точками, указав их положение в первичной и вторичной системах отсчета. Для этого можно воспользоваться оператором АФФА. В программе на фортране первичные и вторичные координаты записываются в двух массивах, передаваемых подпрограмме AFFA; на графическом языке после кода записываются сначала первичные координаты точек, затем – вторичные. Точки в первичной и вторичной системах не должны быть коллинеарными и, конечно, не должны совпадать. В зависимости от их положения могут получаться как общие случаи аффинного преобразования, так и частные. Могут получаться и преобразования, обращающие ориентацию (определители матриц для которых отрицательны).

Теперь – об *умножении преобразований*. Известно, что последовательность двух или нескольких аффинных преобразований можно заменить одним таким преобразованием, если должным образом подобрать его коэффициенты. По аналогии с умножением матриц в векторной алгебре здесь также перемножаются матрицы, но (в нашей системе) расширенные. Одна из них – матрица "предыдущего" преобразования, хранящаяся в рабочем поле АМАТ1, а другая размещена в памяти ЭВМ. Результирующая матрица образуется также в поле АМАТ1. Такое умножение *некоммутативно*, т.е. результат зависит от порядка сомножителей, поэтому надо проверять, какое из исходных преобразований должно быть первым, а какое – вторым. Сформировать матрицы исходных преобразований можно, конечно, различными способами, но обычно пользуются упомянутыми выше и другими стандартными операторами совместно с операторами пересылки матриц (они тоже указаны в табл. 5).

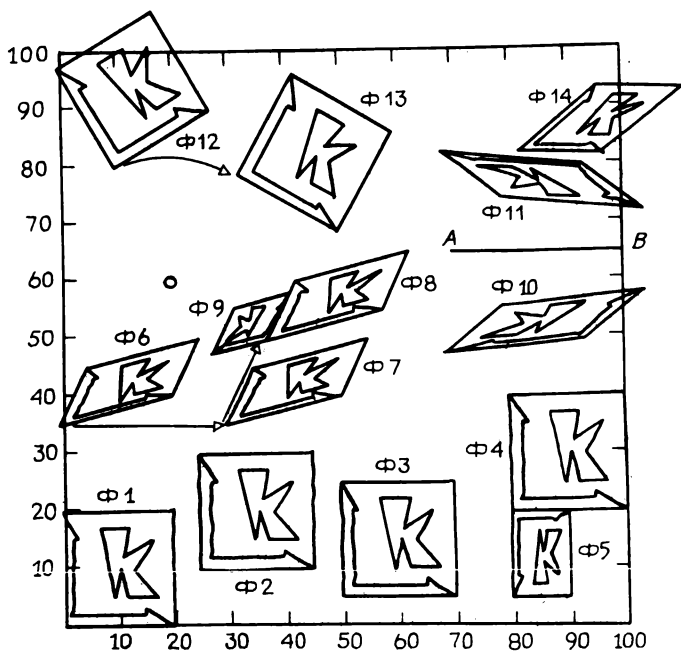


Рис. 81. Пример на аффинные преобразования

На рис. 81 приведен пример, иллюстрирующий эти преобразования. Здесь в разных "видах" изображается тестовая фигура — квадратик со сторонами длиной 20 мм и стилизованными стрелками и буквой. В ЧКС квадратик задан так, что одна его вершина инцидентна началу ЧКС, а два ребра идут по осям x и y . Это дает возможность по виду фигуры, вычерченной с каким-то преобразованием, судить о положении осей первичной системы и масштабах по ним. Оси ОКС на рисунке помечены. Ниже приводятся текст программы на фортране — и справа от нее — соответствующий текст на языке алграф-Р, который обсудим чуть позже. Простую подпрограмму KV, которая изображает тестовую фигуру, мы не приводим.

DIMENSION A(6)	
CALL SBROS	
CALL PREOB1 (1)	ПРБЛ2,
CALL KV	Б (К, В),
С ПЕРЕНОСЫ	
CALL PEREN (25., 10., 0)	Ф2 : ПЕРЕН, 25, 10, Б (К, В),
CALL KV	
CALL PEREN (25., -5., 0)	Ф3 : ПЕРЕН, 25, -5, Б (К, В),
CALL KV	

CALL PEREN (80., 20., 2)	Ф4: ПЕРЕН2, 80, 20, Б (К, В),
CALL KV	
С МАСШТАБИРОВАНИЕ	
CALL MASH (0.5, -0.75, 0)	Ф5: МАСШ, 0.5, -0.75, Б (К, В),
CALL KV	
С ПРОИЗВОЛЬНАЯ МАТРИЦА	
CALL MATL2 (1., .25, 0., .25, .5, 35.)	Ф6: МАТЛ2, 1., .25, 0., .25, .5, 35,
CALL KV	
CALL PEREN (30., 0., 0)	Б (К, В),
CALL KV	Ф7: ПЕРЕН, 30, 0, Б (К, В),
С ЗАПОМНИТЬ МАТРИЦУ	
CALL MATR (A, 6, 11)	МАТР (611, А),
CALL PEREN (0., 30., 1)	Ф8: ПЕРЕН1, 0, 30, Б (К, В),
CALL KV	
CALL MASH (-0.5, 0.8, 0)	Ф9: МАСШ, -0.5, 0.8, Б (К, В),
CALL KV	
CALL MASH (2.5, 1., 1)	Ф10: МАСШ1, 2.5.1 Б (К, В),
CALL KV	
С СИММЕТРИЯ	
CALL MASH (1., -1., 1)	Ф11: МАСШ1, 1, -1, ПЕРЕН, 0, 130,
CALL PEREN (0., 130., 0)	
CALL KV	Б (К, В),
С ПОВОРОТЫ	
CALL POV1 (10., 80., 30.)	Ф12: ПОВ1, 10, 80, 30, Б (К, В),
CALL KV	
CALL POVВ (20., 60., -60.)	Ф13: ПОВВ, 20, 60-60, Б (К, В),
CALL KV	
С УМНОЖЕНИЕ ПРЕОБРАЗОВАНИЙ	
CALL UMT (A, 2)	Ф14: УМТ (2, А), Б (К, В),
CALL KV	
С ДОП. ЛИНИИ, ЗАВЕРШЕНИЕ	
CALL PREOB1 (0)	ПРБ1, ОТРЕЗ, 70, 65, 100, 65, ОКР,
CALL OTREZ (70., 65., 100., 65.)	20, 60, 1
CALL OK (20., 60., 1.)	
CALL SHKAL	
CALL STP	СТОП,
END	А: ,,,,,,
	К: ПРМК, 20, 0, -20, 20, П, ЛИН, 15, 3,
	15, 2, 2, 2, 2, 15, 3, 15, 0, 20, В, 9.5, 5, П,
	10.5, 10, 12, 5, 17, 5, 13, 10, 16, 15, 11, 12,
	12, 17, 7, 17, 9.5, 5, В, В:,

Фигура на рис. 81 начерчена так, как если бы она была задана непосредственно в ОКС, ибо иницирующий модуль SBROS формирует тождест-

венную матрицу. Далее демонстрируется действие операторов переноса (фигуры Ф2, Ф3, Ф4), масштабирование (с изменением направления одной из осей на противоположное). Для фигуры Ф6 матрица общего вида задана непосредственно, затем демонстрируется действие различных модификаций переноса, что позволяет ощутить разницу между ними. На примере фигуры Ф11 показано, как можно получить симметрию относительно прямой, параллельной какой-либо оси ОКС (эта прямая помечена отрезком АВ), "Предыдущая" ЧКС сначала зеркально отражается от одной из осей ОКС, а затем дается перенос на удвоенное расстояние от координатной оси до оси симметрии.

Фигуры Ф12 и Ф13 демонстрируют действие операторов поворота. Заметим, что после "косых" предыдущих фигурок Ф12 снова стала конгруэнтной фигуре Ф1. Наконец, последняя из фигурок – Ф14 – получена с помощью умножения преобразований. Матрица предыдущего преобразования умножается на матрицу, образованную перед черчением фигурки Ф7 и запоминавшуюся с помощью подпрограммы MATR в массиве А. При этом первым преобразованием было предыдущее, т.е. давшее фигурку Ф13. Проверьте, так ли это, представив себе промежуточную координатную систему, оси которой совпадают с осями ЧКС, давшей фигурку Ф7. Относительно этих осей Ф14 должна быть расположена так же, как Ф13 относительно осей ОКС.

5.2. Дополнительные сведения о языке описания геометрической информации и задание на нем преобразований координат

Аналогично вызову подпрограмм в различных языках программирования в языке описания геометрической информации предусмотрен вызов блоков информации. Действительно, сложные изображения часто содержат повторяющиеся части и фрагменты, которые разумно описать однократно, а затем по мере необходимости обращаться к этим описаниям, дополнительно указывая, где и в каком положении какую из частей надо разместить. Для указания места и положения, естественно, используются преобразования координат, а обособленное описание части информации в системе Алграф называется *блоком информации*. В других системах говорят о "сегментах", "структурированных файлах" и т.п.

Блоки информации в тексте располагаются после основной части текста, завершаемой кодом СТОП. Каждый блок помечается двумя метками – начальной и конечной, причем конечная помечает пустой элемент (после двоеточия сразу ставится запятая); этот элемент будет использован в процессе работы системы. Чтобы из главной части текста вызвать блок (т.е. в нужный момент отработать его текст) записывается команда такого вида: Б(Н,К), где Н и К – начальная и конечная метки нужного блока. Таким образом, чтобы получить изображение, которое мы видим на рис. 82,а, можно подготовить такой текст с блоком, где "в приращениях"

закодировано вычерчивание крестика:

ПРМК,0,0,50,20,ЛИН,10,10,Б (Н,К) ,20,5,Б (Н,К) ,30,15,Б (Н,К) ,
40,10,Б (Н,К) ,СТОП,Н:ЛИНР,0,-2,П,0,4,В,-2,-2,П,4,0,В,ЛИН,К: , &

Заметьте, что в конце "тела" блока не случайно код геометрического элемента снова заменяется на ЛИН.

Блоков к основному тексту описания может быть несколько и располагаются они в произвольном порядке друг за другом; из одного блока можно вызывать другой, из другого третий...

Другой мощный алгоритмический прием – циклы. Конструкция цикла позволяет заданное число повторить отработку части текста описания.

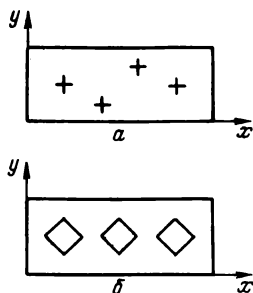


Рис. 82. Изображения, полученные с помощью блоков информации и циклов на языке геометрического описания

Эта часть, называемая *телом цикла*, размещается сразу после *заголовка цикла* и продолжается вплоть до *элемента возврата* – пустого элемента, помеченного меткой конца цикла. Заголовок цикла имеет такую структуру: ЦИКЛ(МК) , n; МК здесь – имя метки конца цикла, вместо буквы n записывается число повторений. Вот простейший пример, соответствующий рисунку 82, б:

ПРМК,0,0,50,20, ЛИН,10,5,	
ЦИКЛ(КК) , 3,	<i>заголовок цикла</i>
ЛИНР,П,5,5,-5,5,-5,-5,5,-5,В,15,0,	<i>тело цикла</i>
КК: ,	<i>элемент возврата</i>
СТОП, &	

В теле цикла можно организовать другой, "вложенный" цикл. Можно в теле цикла вызывать блоки информации и наоборот, в блоках можно образовывать циклы.

Читатели, знакомые с основами фортрана или другого языка программирования, легко установят соответствие синтаксических конструкций этих языков и взятого нами в качестве примера геометрически-ориентированного языка. Справа от программы на фортране для рис. 81 приведен соответствующий текст геометрического описания, где вместо вызовов подпрограммы KV для изображения текстовой фигуры вызывается блок

с метками К и В; текст его приведен в конце описания. Заметим, что здесь используются не приращения координат, а координаты и другие геометрические параметры, так как различное положение фигурок на чертеже обеспечивается за счет преобразований координат. Несколько запятых, представленных после метки А, резервируют место для хранения шести элементов матрицы (можно использовать и другие способы резервирования места в массиве).

Три следующих примера показывают, как можно использовать циклы совместно с преобразованиями. На рис. 83,а для изображения фасада применен двойной цикл с переносами. Проследите за перемещением осей ЧКС в процессе вычерчивания различных окон. Проверьте, правильно ли задан наклонный вектор переноса, подготавливающий систему для вычерчивания следующего вертикального ряда (столбца) окон. Вот текст кодировки:

```
ПРМК,8,5,84,5, ОТРЕЗ,10,10,10,50,90,10,90,50,ЛИН,8,50,П,20,
60,80,60,92,50,8,50,В, МТЛ2, ПРБЛ2, ЦИКЛ(4),4,ЦИКЛ(М),2,
ПРМК,15,15,10,10,ОТРЕЗ,15,22,25,22,20,22,20,15,ПЕРЕН,0,15,
М:,ОКР,20,15,3, ПЕРЕН,20,-30,К:, СТОП,&
```

Желающие могут составить текст программы на фортране для этого рисунка.

В следующем примере операция дополнительного вращения используется для элементов, равномерно повторенных по окружности (рис. 83, б):

```
ОКР,30,30,22,30,30,20,30,30,8,30,30,5,ПРБЛ2,ЦИКЛ(К),6,ОКР1,
30,30,18,65,50,30,30,11,105,-30,30,30,18,65,,1,В,ПОВБ,30,
30,60,К:,МТЛ2, ПУНК2,ОТРЕЗ,0,30,60,30,30,0,30,60,СТОП,&
```

Чтобы замкнуть контур отверстия, приведу перо в начало ранее вычер-

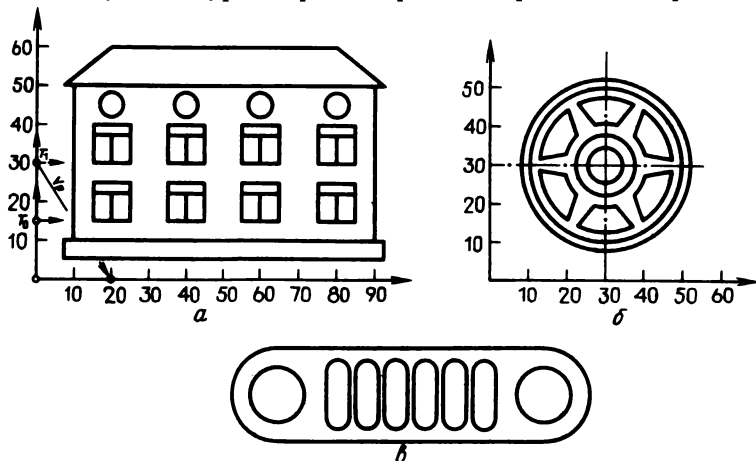


Рис. 83. Примеры сочетания преобразований координат с циклами



Рис. 84. Изображение фасадов зданий с использованием циклов и преобразований

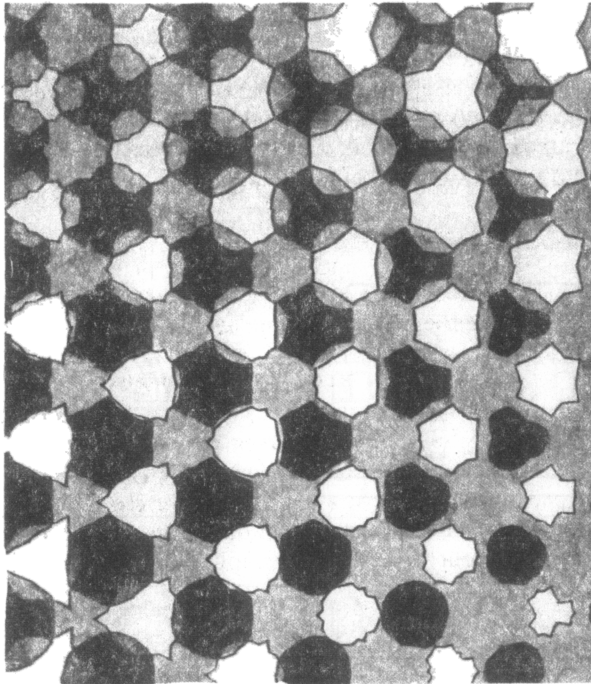


Рис. 85. Орнамент, сгенерированный машиной

ченной дуги, здесь задается фиктивная "короткая" дуга. Заметим, что для черчения осевых линий тождественная матрица восстанавливается кодом МТЛ2.

На рис. 83, в вычерчена симметричная фигура, в которой содержатся повторяющиеся элементы. Для упрощения ось u_{oc} совпадает с осью симметрии фигуры. Здесь, как и в первом примере, использован двойной цикл. Обратите внимание, что вектор переноса во внутреннем цикле задается в ЧКС, а не в ОКС (почему?):

```
ПРБЛ2,ЦИКЛ(К),2,ОТРЕЗ,0,8-35,8,ОКР2,-35,32,ОТРЕЗ,-35,32,
0,32,ОКР,-35,20,7,ЦИКЛ(М),3,ОКР1,-4,26,3,0,180,-4,14,3,180,
180,ЛИН,-1,26,В,ПЕРЕН1,-8,0,М:,ПЕРЕН1,24,0,МАСШ,-1,1К:,
СТОП,&
```

На рис. 84 изображены более сложные фасады зданий. Попробуйте для них составить программу (или геометрическое описание).

Очень подходит подобная структура с циклами и преобразованиями для получения рисунков различных орнаментов, заполняющих полосу, круг или часть плоскости. Число вложений циклов здесь может быть четыре, пять и более; может потребоваться запоминание матриц и их умножение. На рис. 85 показан орнамент с постепенно изменяющимися элементами. При сохранении четкой схемы замощения на базе равнобедренных треугольников рисунок элементарных ячеек постепенно изменяется. Как это было сделано? В четырех угловых ячейках прямоугольного поля были заданы различные рисунки, состоящие, однако, из одинакового числа штрихов; для остальных ячеек алгоритм образования рисунка напоминает алгоритм построения пропорциональных линий, обсуждавшийся выше.

5.3. Полярные координаты

Полярные координаты – наиболее распространенный пример нелинейных координат, порождающих криволинейные преобразования. Действительно, если придерживаться "активной" позиции, получится, что прямые линии объекта, "вмещенного" в полярную координатную систему, искривятся (кроме прямых, проходящих через полюс). В практике машинной графики и геометрии может потребоваться перевод координат как из полярной КС в декартову, так и обратно. Чаще объект, заданный в полярных координатах, требуется перевести в декартову систему и отобразить на графическом устройстве. Напоминаем, что в простейшем случае, когда полюс полярной КС инцидентен началу декартовой (рис. 86, а), уравнения преобразования таковы:

$$\begin{aligned}x &= \rho \cos \varphi, \\y &= \rho \sin \varphi.\end{aligned}$$

Полярный радиус ρ и полярный угол φ в графической системе будут теми же первичными координатами в ЧКС, которые заносятся подпрограмма-

ми генерации геометрического элемента в ячейки XOR и YOR. Кроме того, в системе реализовано преобразование несколько более общего вида: во-первых, оно содержит две подступени и вторая подступень – аффинное преобразование из промежуточной КС в ОКС по матрице, хранящейся на рабочем поле АМАТ1. Во-вторых, полюс полярной системы можно располагать в произвольной точке промежуточной системы, а начальный полярный радиус, от которого отсчитываются углы, можно направлять

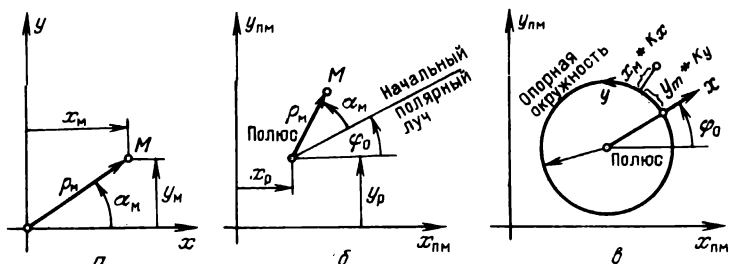


Рис. 86. Схемы задания полярных координат: а – обычных, б – в системе Алграф, в – специальных

под желаемым углом к оси $x_{пм}$ (рис. 86, б). Уравнения будут такими:

$$x = x_0 + \rho \cos(\varphi + \varphi_0),$$

$$y = y_0 + \rho \sin(\varphi + \varphi_0).$$

Коэффициенты преобразования x_0 , y_0 , φ_0 , образующие "квазиматрицу", располагаются в первых трех элементах массива АМАТ0 (после инициализации графической системы эти элементы нулевые). Еще более сложным является преобразование из *специальной полярной системы* (рис. 86, в). Здесь полюс и начало системы (где обе координаты равны нулю) – различные точки. Обе координаты носят, как говорят, лонгальный характер, т.е. первая в данном масштабе и с учетом знака соответствует расстоянию от точки до опорной окружности, которая играет здесь роль криволинейной координатной оси, вторая – также с учетом масштаба и знака – длине дуги опорной окружности от начала КС до проекции точки на эту окружность. Масштабные множители, радиус опорной окружности и другие величины составляют "квазиматрицу" преобразования.

Рассматривая преобразование с "активной" точки зрения, любопытно разобраться в его особых элементах, отображаемых областях, наложениях и других особенностях. Так, при $\rho = 0$ ($XOR = 0$) точки первичной плоскости с различными значениями φ (YOR), образующие положительный луч оси y , отобразятся в одну точку – полюс (имеется ввиду не специальная полярная система). Можно ли преобразовывать точки первичной плоскости, для которых $\rho < 0$? Программа выдаст результат. Выдаст она ре-

зультат, и если полярный угол будет соответствовать величине, превышающей "один оборот". Это значит, что первичная плоскость будет "накручиваться" на вторичную с бесчисленным множеством наложений! Прямая общего положения, "вмещенная" в полярную КС, превратится в спираль, окружности – в кривые четвертого порядка, которые могут иметь точки самопересечения (когда?). Семейство таких окружностей показано на рис. 87.

Деформировать подобным образом можно различные кривые, в программах для этого можно использовать подпрограмму CLINE, перед вызовом которой задается нужный тип преобразования и уточняются его коэффициенты. Может понадобиться, чтобы для вызываемой в CLINE подпрограммы-функции аргумент соответствовал полярному углу (второй координате), а функция – радиусу; в этом случае последний параметр в операторе вызова задают единичным. Этим способом получают изображения синусоид, "скрученных" в кольцо. Уравнение линии в полярных координатах можно записать:

$$\rho = R_0 + A \cos\left(\varphi \frac{2\pi}{S}\right), \text{ где:}$$

ρ – полярный радиус, φ – полярный угол, R_0 – радиус "осевой линии", A – амплитуда, S – длина волны в угловых единицах (в нашем случае –

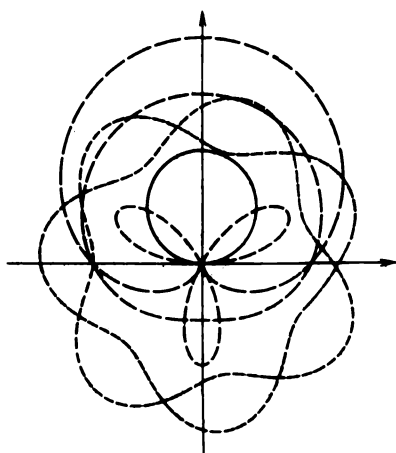
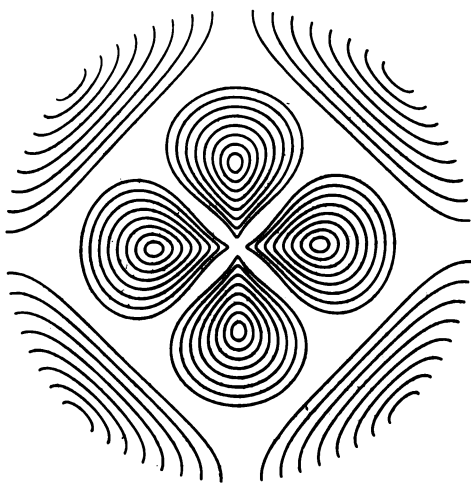


Рис. 87. Окружности, "вмещенные" в полярную систему

Рис. 88. Изображение синусоид в полярных координатах

в градусах). Для вычерченных линий параметры уравнения следующие:

№ линии	R_0	A	S	φ_{\max}
1	0	40	360	360
2	40	40	360	360
3	40	20	360	360
4	0	35	120	180
5	50	10	102	720

Среди них – двойная окружность, кардиоида, так называемый трилистник. Выполняя различные действия с тригонометрическими функциями, можно получить массу любопытных кривых. На рис. 89 приведены несколько примеров.

Уравнение линии первого семейства такое:

$$\rho = A (5 + 2 \cos(\varphi) + 3 \cos(\varphi^m) \operatorname{sign}(\cos(\varphi))) - B (\sin(18\varphi))^2 (\cos(\varphi/2))^8.$$

Параметры A , B , m изменялись для получения различных линий семейства. Отдельные части выражения здесь играют роль формообразования опорной

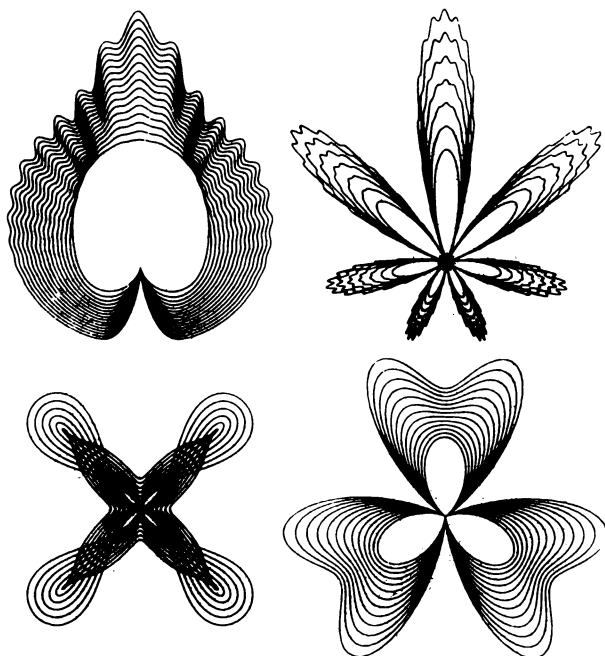


Рис. 89. Более сложные кривые, сконструированные в полярных координатах

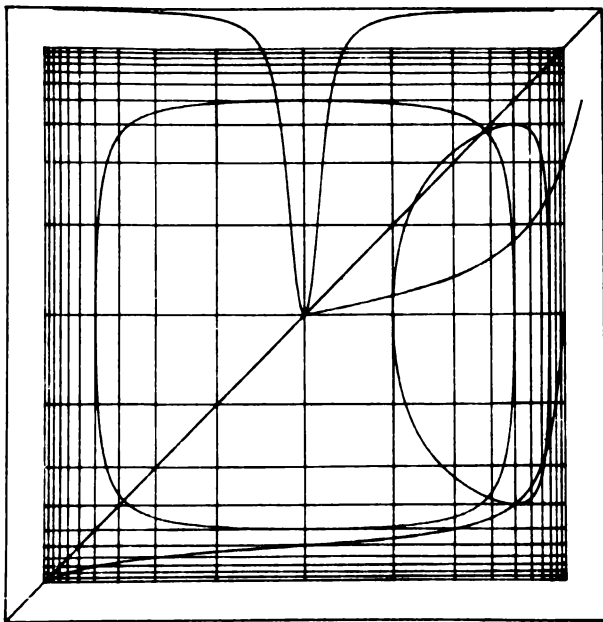


Рис. 90. Изображение объектов в арктангенциальных координатах

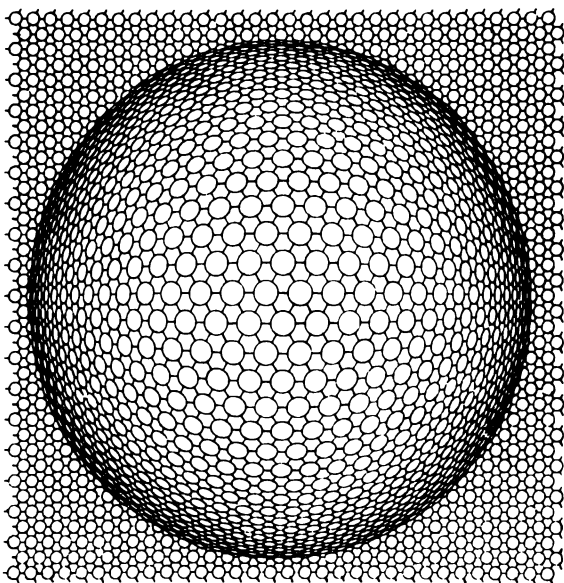


Рис. 91. Орнамент, построенный с помощью нелинейных преобразований

(несущей) кривой, мелких волн, закона изменения амплитуды этих волн. Угол φ здесь предполагается заданным в радианах.

Для второго семейства уравнение выглядит так:

$$\rho = A [\cos(\varphi) + \sqrt{3 + (\cos(\varphi))^2}] [\cos(3,5\varphi)]^2 - B [\sin(60\varphi)]^2.$$

Третье семейство строится на основе параметрических уравнений

$$\rho = A \cos(4t) + B,$$

$$\varphi = t - C \sin(4 + t).$$

Вот уравнение для линии четвертого семейства:

$$\rho = A(1 + \cos(3\varphi)) + B(\sin(3\varphi))^2.$$

Иногда оказываются полезными преобразования, вмещающие всю первичную плоскость в круг или прямоугольник заданных размеров. В этом случае график любой функции, какие бы большие значения она не принимала, можно изобразить в пределах ограниченного поля чертежа (рис. 90). На рис. 91 показан орнамент, который "сначала был" равномерным, а затем часть его подверглась криволинейному преобразованию. Это преобразование радиально-функционального растяжения перемещает точку первичной плоскости по радиусу к началу КС или от начала в зависимости от ее исходного положения. Попробуйте составить уравнения преобразования! Здесь наблюдается явление умножения (композиции) преобразований разных типов: ведь равномерный орнамент получается с помощью аффинных преобразований.

ТРЕХМЕРНОЕ ПРОСТРАНСТВО

Знакомые объекты в трехмерном пространстве становятся сложнее: точка задается уже тремя координатами, окружность общего положения надо задавать по крайней мере шестью числами (таково ее параметрическое число). Появляются качественно новые объекты – пространственные линии, поверхности, ограниченные поверхностями тела. Еще одно усложнение: необходимость наглядного изображения на плоскости (или поверхности экрана) пространственных объектов и ситуаций. Это – задача *построения проекций*. Ее мы обсудим позже, а пока будем считать, что графопостроитель тоже стал как бы трехмерным, чертит в пространстве, а мы рассматриваем фотоснимки с этих “пространственных картинок”. Сначала посмотрим, как можно в пространстве задавать плоские фигуры и фигуры объемные (пространственные).

6.1. Двумерные фигуры в трехмерном пространстве

За исключением оговоренных случаев считаем, что в пространстве введены прямоугольные декартовы координаты. Три оси могут быть расположены различным образом, но для определенности наметим некоторое их начальное положение перед наблюдателем. Пусть ось x проходит слева направо, ось y – снизу вверх и ось z – “вдоль” от наблюдателя. В различных приложениях геометрии ось z чаще направляют вертикально, к тому же наша тройка осей получилась “левой” в отличие от обычно применяемой “правой”. Такое расположение осей позволяет легче сочетать двумерную машинную графику с трехмерной. Ведь мы привыкли видеть перед собой двумерный чертеж, где ось x идет вправо, а ось y – вверх.

Если использовать трехмерные координатные системы, в наших графических программах двумерные фигуры естественным образом оказываются расположенными в плоскости $z = \text{const}$. Действительно, программы, генерирующие такие геометрические элементы, работают с двумя первыми координатами, с переменными XOR и YOR ; значение же третьей координаты

ты не используется. После действия иницизирующего модуля $ZOR = 0$. и потому двумерные фигуры будут расположены в плоскости $z = 0$; в нужный момент действия программы переменной ZOR можно придать любое другое значение (на языке алгеф-Р для этого мы записываем код Z и после него, через запятую, численное значение координаты). Фигуры после этого "попадут" в другую плоскость.

Нередко нужно расположить двумерные фигуры в других координатных плоскостях ($x = \text{const}$ и $y = \text{const}$). Формально это можно сделать с помощью преобразования координат. В алгебре можно использовать *основное* аффинное трехмерное преобразование, о котором пойдет речь ниже, а можно специально для этого предназначенную предварительную ступень, которая управляется индексом $IND(4)$ и предназначена только для *смены осей* (взаимной смены значений двух координат). При такой смене автоматически вводится промежуточная координатная система, а "основное" преобразование можно использовать дополнительно. Заметим, что в ЧКС фигуры располагаются по-прежнему в плоскостях $z = \text{const}$.

На рис. 92 приведен пример несложного машинного изображения, где несколько фигур размещены в различных координатных плоскостях. Средства построения проекции в приводимых ниже программе и тексте

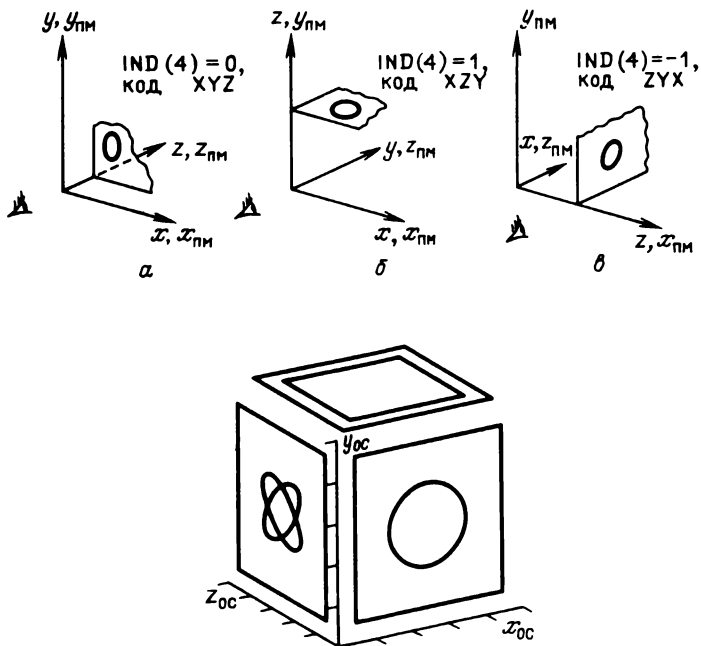


Рис. 92. Схемы, иллюстрирующие смену координатных осей (а, б, в) и машинное изображение, использующее эти операции

кодировки мы пока не обсуждаем.


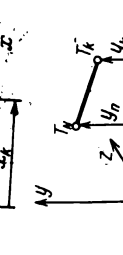

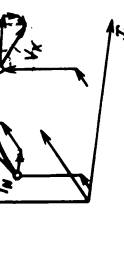
```
COMMON/BENKOT/XOR,YOR,ZOR,XYZ(50),IND(40)
CALL SBROS
IND(6)=1                ПРБАКС,
CALL AKS(0.,0.,0.,0.,30.,20.,1.) АКС,0,0,0,0,30,20,1,
ZOR=0.                  Z,0,
CALL PRM(5.,5.,40.,40.) ПРМК,5,5,40,40,
CALL OK(25.,25.,10.)   ОК,25,25,10,
IND(4)=-1              ЗУХ,ПРМК,5,5,40,40,
CALL PRM(5.,5.,40.,40.) ЭЛЛ1,25,25,10,5,45,25,25,10,5,-45,
CALL EL1(25.,25.,10.,5.,45.)
CALL EL1(25.,25.,10.,5.,-45.)
IND(4)=1                XZY,Z,50,
ZOR=50.
CALL PRM(5.,5.,40.,40.) ПРМК,5,5,40,40,10,10,30,30,
CALL PRM(10.,10.,30.,30.)
CALL STP                СТОП,&
END
```

В программе на фортране блок **BENKOT** описан сокращенно (что допускают правила языка); оператор **ZOR = 0.** включен в программу для наглядности.

Чтобы плоские фигуры расположить в плоскостях общего положения, наклонно проходящих в ОКС, надо воспользоваться трехмерными аффинными преобразованиями. Применяя нелинейные преобразования трех координат, можно располагать двумерные фигуры в координатных поверхностях этих систем; в ОКС эти фигуры будут уже не плоскими.

6.2. Пространству – пространственные линии

С учетом того что точка в пространстве задается тремя координатами, а вектор – тремя проекциями, задание в пространстве отрезков и ломаных аналогично заданию их на плоскости. В табл. 6 приводится несколько подпрограмм и кодов. Как и в двумерной графике, при задании отрезков от отработанных точек пользователь должен управлять включением и выключением пера. Качественные нововведения возникают лишь при задании отрезка его длиной и углами наклона. На схеме (рис. 93) видно, что один из углов – это угол наклона проекции отрезка на плоскость $хоу$ к оси x , а второй – угол наклона самого отрезка к этой же плоскости $хоу$ (или к параллельной ей). Те же подпрограммы черчения ломаных, которые мы упоминали раньше, используются и для трехмерных ломаных, только для каждого из способов задания значение индекса модификации на единицу больше. Например, при задании незамкнутой ломаной координатами опорных точек значение индекса – 3. Трехмерные ломаные в общем случае являются пространственными линиями, не расположенными в какой-либо одной плоскости. Пространственные кривые линии в общем случае

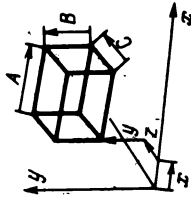
Элементы фигуры	Параметры	Операторы преобразования	Код на языке диграф-Р	Схема
Отрезок в пространстве от обработанной точки, холостой ход	Координаты конечной точки	CALL LINP *(XK, YK, ZK)	ЛИНП, xk, yk, zk	
Отрезок в пространстве с управлением пером	Координаты начальной и конечной точек	CALL OTREZP *(XN, YN, ZN, *XK, YK, ZK)	ОТРЕЗП, xp, yp, zp, xk, yk, zk, ...	
Пространственная дуга параметрического полинома 3 степени	Три координаты и три компонента вектора направления для начальной и для конечной точек (в массиве A)	CALL POL3P *(A, 1, 14)		
Пространственная составная линия из дуг полиномов	Три координаты для каждой опорной точки (на фортране - в массиве A); NT - число точек; минимально - 4 точки	CALL MPP3P *(A, NT, 10)	ПОЛИН (30), x1, y1, z1, x2, y2, z2, ..., B,	

Параллелепипед, ребра которого параллельны координатным осям; чертятся все ребра

Координаты одной из вершин и длины трех отходящих от нее ребер, со знаками

CALL PRP (0, X,
*Y, Z, A, B, C)

ПРПШ, x, y, z,
a, b, c, ...

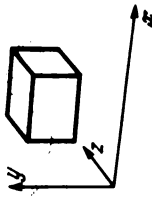


То же, чертятся ребра граней, обращенных к наблюдателю

То же

CALL PRP (10, X,
*Y, Z, A, B, C)

ПРПН, x, y, z,
a, b, c, ...

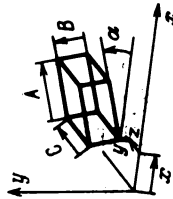


Параллелепипед, 4 ребра которого параллельны оси Z, чертятся все ребра

Координаты одной вершины, длины трех отходящих от нее ребер, угол наклона первого ребра к оси x

CALL PRP (3, X,
*Y, Z, A, B, C, ALF)

ПРПУГП, x, y, z,
z, a, b, c, α, ...



То же, чертятся ребра граней, обращенных к наблюдателю.

То же

CALL PRP (13, X,
*Y, Z, A, B, C, ALF)

ПРПУГН, ...

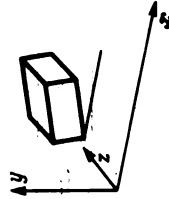
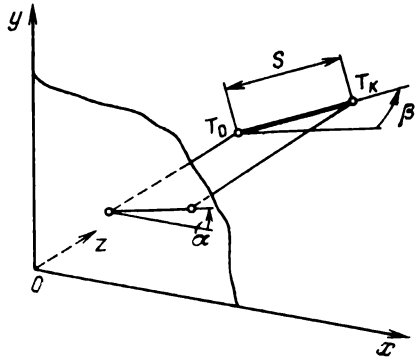


Таблица 6 (окончание)

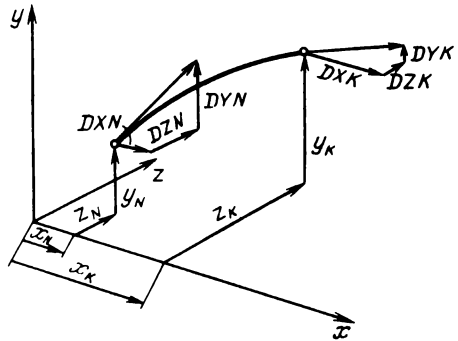
Элемент, фигура	Параметры	Оператор на фортране	Код на языке алграф-Р	Схема
<p>Призма с боковыми ребрами, параллельными оси z (чертятся все ребра)</p>	<p>Координата z, определяющая плоскость 1-го основания, высота призмы, далее по две координаты для каждой вершины одного основания (на фортране – в массиве A, NT – число вершин)</p>	<p>CALL PRIZM *(0, A, NT)</p>	<p>ПРИЗМП, z0, h, x1, y1, x2, y2, ..., B,</p>	
<p>То же, чертятся ребра граней, обращенных к наблюдателю</p>	<p>То же</p>	<p>CALL PRIZM(10, A, *NT)</p>	<p>ПРИЗМН, z0, h, x1, y1, x2, y2, ...</p>	
<p>Конус круговой, ось параллельна оси z</p>	<p>Координаты центра одного основания, радиусы оснований, высота со знаком</p>	<p>CALL KONUS2(XC, *YC, ZC, R1, R2, H)</p>	<p>КОНУС, xc, yc, zc, r1, r2, h,</p>	

Рис. 93. Один из способов задания отрезка от отработанной точки в пространстве (длиной S и углами α, β)



могут изгибаться в любых направлениях. Однако для некоторых типов дуг и составных кривых задание их не сложнее, чем для двумерных линий. Вспомня типовую конфигурацию данных, показанную на рис. 31, естественным образом распространяем ее на трехмерный случай (рис. 94). Аналогично двумерной дуге параметрического полинома, вычерчиваемой подпрограммой POLIN3, дугу пространственного полинома аналогичного типа вычерчивает подпрограмма POL3P; в частности, модификации IM=13 и IM=14 рассчитаны на задание в начальной и в конечной точках дуги координат и проекций касательных векторов; первая из модификаций "реагирует" на длины векторов, вторая нормирует их в зависимости от расстояния между точками. Пространственные составные линии из дуг того же

Рис. 94. Типовая конфигурация исходных данных для задания пространственной дуги



типа проводит подпрограмма MPPL3P. В табл. 6 указаны некоторые ее модификации. Наверное, читателю понятно, что для трехмерной пространственной линии нужны три уравнения, выражающие значения каждой из трех пространственных координат через значение параметра:

$$\begin{aligned} x &= f(t), \\ y &= \varphi(t), \\ z &= \psi(t). \end{aligned}$$

Пространственную кривую нельзя описать одним уравнением, как минимум нужны два уравнения. Каждое из них можно интерпретировать как уравнение кривой поверхности, а оба они описывают линию пересечения этих поверхностей (рис. 95, а) :

$$F(x, y, z) = 0,$$

$$G(x, y, z) = 0.$$

Возьмем теперь такой частный случай, когда в одном из уравнений нет

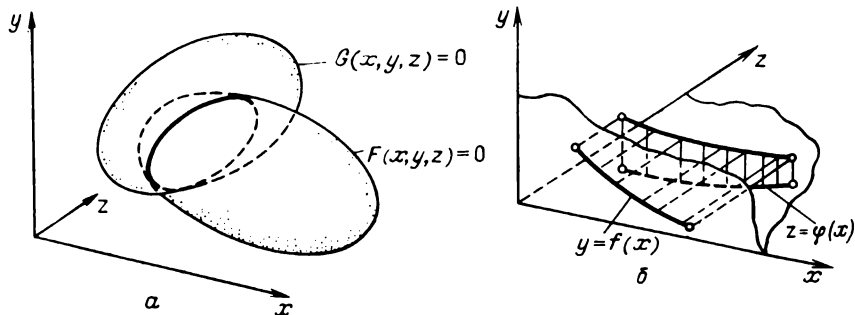


Рис. 95. Задание пространственной линии как пересечения двух поверхностей (а) и пересечения двух поверхностей в виде проецирующих цилиндров (б)

координаты z , а во втором — координаты y и, кроме того, эти уравнения явно разрешены относительно y и соответственно z :

$$y = f(x),$$

$$z = \varphi(x).$$

В пространстве эти два уравнения описывают две проецирующие цилиндрические поверхности (рис. 95, б). Если рассматривать поочередно две плоскости — $хоу$ и $хоz$, — уравнения описывают две плоские кривые. Их можно считать проекциями пространственной кривой пересечения цилиндрических поверхностей. Построить проекцию пространственной кривой на одну из координатных плоскостей (хотя бы ту же $хоу$) можно с помощью вырожденного специальным образом преобразования трех координат. Наша задача в некотором смысле обратная: по двум плоским проекциям восстановить пространственную кривую. Случай в практике проектирования нередкий. Действительно, представьте себе чертеж детали в двух видах; на виде спереди и виде сверху некоторая линия вычерчена как дуга окружности. Вопрос: что представляет собой эта линия в пространстве?

Для описанного частного случая можно составить специальную подпрограмму. Более общий способ восстановления пространственной кривой по ее проекциям носит дискретно-численный характер. Обе проекции считаются предварительно представленными в памяти машины в виде достаточно плотных координатных моделей — т.е. в виде ломаных с мелкими

звеньями. Затем подпрограмма типа имеющейся в системе Алграф SNLOM3 из двух двумерных координатных моделей строит одну трехмерную. Если в первой координатной модели представлены координаты x, y , а во второй — x, z , синтез кажется тривиально простым: из четырех координат двух точек, взятых из двух координатных моделей, надо выбрать три, и это будет точка результирующей координатной модели. Реально точки в двух исходных координатных моделях идут "несинхронно" и имеют различные значения общей координаты (y нас — x). Поэтому подпрограмма, определив начальную точку пространственной линии, продвигается в направлении оси x , каждый раз находя в той или иной координатной модели следующую близкую точку. Из соответствующей КМД берутся две координаты для точки пространственной линии, а третья координата находится путем интерполяции отрезка ломаной из альтернативной координатной модели. В итоге результирующая модель может содержать суммарное количество точек исходных моделей.

В системе Алграф для запоминания координатной модели плоских проекций можно использовать подсистему запоминания координат в массиве AAAA, которая управляется с помощью индекса IND(8); индекс IND(10), получив нулевое значение, временно блокирует черчение. На этом этапе обе проекции генерируются в координатах x и y и последовательно записываются в массив AAAA; описание каждой проекции заканчивается числом-кодом 10000.; номер элемента массива, соответствующий началу второй проекции, присваивается выбранной для этого переменной; режим запоминания отключается, система переходит в режим графического вывода и подпрограмма синтеза пространственной линии вызывается, например, таким оператором:

```
CALL SNLOM3(AAAA,AAA(IA2),F,NF)
```

где IA2 — номер элемента массива, указывающий начало второй проекции (предполагается, что первая проекция записана с начала массива), F — массив для результирующей КМД, NF — число заполненных элементов массива F. Ломаную линию после этого можно расчертить оператором LOM в модификации IM=3. В нужных случаях перед черчением можно изменить преобразование координат, это дает возможность в качестве исходных проекций использовать не только упомянутые, но и другие. Во всех случаях, правда, этот описанный способ годится только для линий, однозначных относительно координаты, принятой за аргумент.

Аналогично тому как мы синтезировали с помощью вновь составляемых подпрограмм линии специальных типов на плоскости, можно это делать и в пространстве. Займемся, например, спиральной линией, расположенной на конической поверхности. Будем строить линию кинематическим способом (см. разд. 4.3). Считаем, что ось конической поверхности параллельна оси z , а точка вращается вокруг этой оси, постепенно удаляется от нее и, кроме того, движется в направлении оси. Начальное

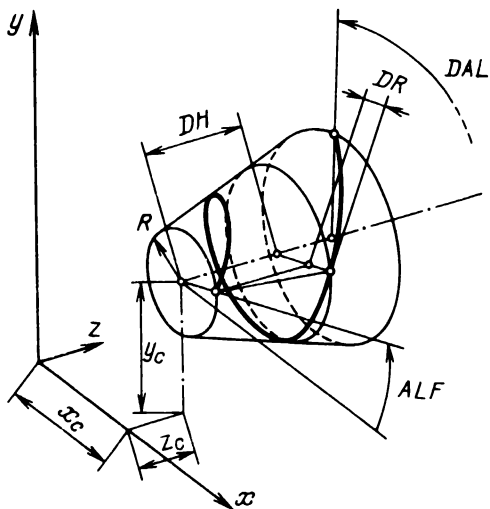


Рис. 96. Схема задания пространственной винтовой линии

положение точки (рис.96) опишем следующими параметрами: x_C, y_C, z_C – координаты точки на оси конуса, через которую проходит перпендикулярная плоскость, содержащая начальную точку линии; R – расстояние от начальной точки до оси; ALF – угол наклона радиуса, проведенного в начальную точку, к оси x . Далее задаем параметры спирали. Это DR и DH – шаги спирали в радиальном и осевом направлениях, т.е. изменение расстояния от точки до оси, когда она делает полный оборот, и смещение ее за это время в направлении оси z . Обе эти величины задаются со знаками. Наконец, надо задать угловую величину дуги спирали DAL (сравните с заданием дуги окружности в разд. 2.3). Эта величина тоже задается со знаком, который будет определять "направление закручивания" спирали. Спирали будем определять параметрическими уравнениями, принимая за параметр угол, составляемый текущим радиусом точки с осью x . Приведем текст подпрограммы, в основных чертах напоминающий ранее приводившийся текст подпрограммы SINLIN:

да она делает полный оборот, и смещение ее за это время в направлении оси z . Обе эти величины задаются со знаками. Наконец, надо задать угловую величину дуги спирали DAL (сравните с заданием дуги окружности в разд. 2.3). Эта величина тоже задается со знаком, который будет определять "направление закручивания" спирали. Спирали будем определять параметрическими уравнениями, принимая за параметр угол, составляемый текущим радиусом точки с осью x . Приведем текст подпрограммы, в основных чертах напоминающий ранее приводившийся текст подпрограммы SINLIN:

```

SUBROUTINE SPRLIN(XC,YC,ZC,R,ALF,DR,DH,DAL)
COMMON/BENKOT/XYZ(53),IND(40),AAA(40)
DDA=SIGN(AAA(9),DAL)
AL=0.
K=0
1 RR=AL/360.*DR+R
CALL LINP(XC+RR*COSG(ALF+AL),YC+RR*SING(ALF+AL),
*ZC+AL/360.*DH)
CALL PERO
AL=AL+DDA
IF(K.NE.0) GOTO 2
IF(DAL-AL)*DDA)3,3,1
3 AL=DAL
K=1
GOTO 1
2 CALL VP
RETURN
END

```

6.3. Кубики и другие фигуры, из которых можно что-то смастерить

Какие типовые фигуры, тела, поверхности в первую очередь используются в машиностроительном и архитектурно-строительном проектировании? Параллелепипеды, призмы, круговые цилиндры и конусы, торовые и сферические поверхности, поверхности эллипсоидов... Предположим, для изображения этих фигур составлены специальные подпрограммы и остается

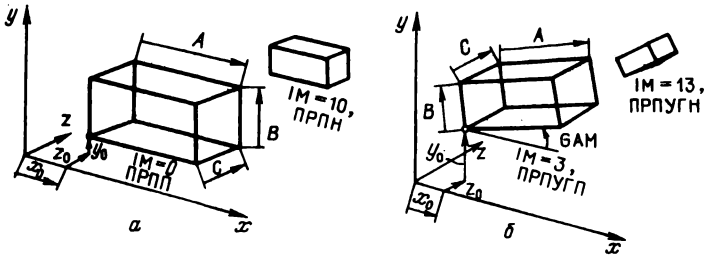


Рис. 97. Задание параллелепипедов в системе Аллграф

разобраться, какими параметрами можно задавать форму, размеры этих объектов и их положение в пространстве. Здесь мы коснемся только некоторых фигур.

На рис. 97 показаны параллелепипеды, расположенные в ЧКС частным образом: один – так, что все его ребра параллельны координатным осям (рис. 97, а), и другой, так, что четыре его ребра параллельны оси z (рис. 97, б). Параллелепипед первого типа задается шестью числами – координатами одной из вершин и длинами трех отходящих от нее сторон (вспомните задание прямоугольника). Ниже показано, как задать параллелепипед на фортране:

CALL PRP(IM,XO,YO,ZO,A,B,C), где IM = 0 или 10.

на языке аллграф-Р:

$$\left. \begin{array}{l} \text{ПРПП} \\ \text{ПРПН} \end{array} \right\} , \text{XO,YO,ZO,A,B,C},$$

Параллелепипед второго типа задается семью числами, причем первые шесть имеют тот же смысл, а седьмой параметр – угол, под которым первая из заданных сторон наклонена к оси x .

CALL PRP(IM,XO,YO,ZO,A,B,C,GAM)

$$\left. \begin{array}{l} \text{ПРПУГП} \\ \text{ПРПУГН} \end{array} \right\} , \text{XO,YO,ZO,A,B,C,GAM}, \dots$$

Подпрограмма вычерчивания параллелепипедов сама в нужные моменты включает и выключает перо.

Однако на нынешнем этапе развития техники заказчики все чаще требуют более реалистических изображений, где объекты показаны непрозрачными, а то и раскрашенными. Об особенностях создания таких изображений мы поговорим ниже, а пока отметим возможность *частичного устранения невидимых линий*. В ряд подпрограмм "встроены" алгоритмы, которые блокируют черчение ребер, принадлежащих тем граням тела (поверхности), которые обращены к наблюдателю "обратной" (внутренней) стороной.

Такой параллелепипед можно получить с помощью модификаций 10 и 13; если он на чертеже один, то будет выглядеть как непрозрачный. Правда, различные параллелепипеды, как и другие тела, будут по-прежнему "просвечивать" друг через друга.

Следующий по распространенности многогранник – призма (рис. 98). В прямоугольных системах отсчета призмы прямые, основания лежат в плоскостях $z = \text{const}$, а боковые ребра параллельны оси z . Призмы могут быть правильными и неправильными. Правильные призмы с основаниями в виде правильных многоугольников можно получить как частные случаи правильных пирамид (рис. 98, б), для этого существует специальная подпрограмма, ее вызов на фортране осуществляется так:

CALL PYRAM5(XC,YC,ZC,R1,R2,H,GAM,NT,IM)

NT здесь число боковых граней; IM=0 для "прозрачной" пирамиды и

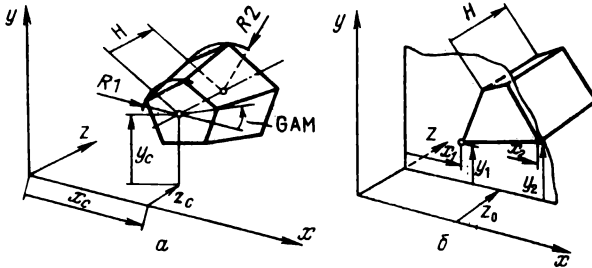


Рис. 98. Задание призм и правильных пирамид в системе Алграф

IM=10 для частичного контроля видимости; другие параметры ясны из рисунка.

Остальные призмы вычерчиваются (рис. 98, б) подпрограммой PRIZM (коды ПРИЗМН, ПРИЗМП).

CALL PRIZM(IM,A,NT)

Здесь A – массив с геометрическими параметрами, IM=0 для "прозрачной" и IM=10 для частично "непрозрачной" призмы; NT – число боковых граней.

Геометрические параметры в массиве A располагаются в таком порядке:

Z0,H,X1,Y1,X2,Y2,...,Xn,Yn

Как показано на рис. 98, при обращении к подпрограмме указывается имя массива, содержащего геометрические параметры фигуры.

Первым параметром является значение координаты z для точек одного из оснований призмы, следующим – высота фигуры со знаком (расстояние от плоскости первого основания до плоскости второго), далее идут значения координат x и y для последовательных вершин одного основания.

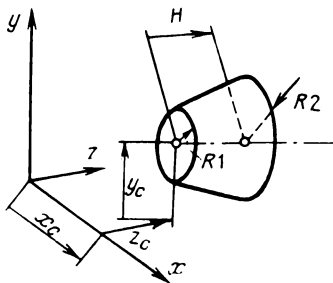


Рис. 99. Задание конусов в системе Алграф

В аналогичном порядке геометрические параметры задаются и при описании фигуры на языке алграф-Р:

$$\left. \begin{array}{l} \text{ПРИЗМН} \\ \text{ПРИЗМП} \end{array} \right\} x_0, h, x_1, y_1, x_2, y_2, \dots, B.$$

В этом случае число вершин не указывается, но после задания всех параметров в тексте записывается какой-либо код (например, B), который служит "сигналом" об окончании описания очередной призмы. При индексе модификации $IM=0$ призмы чертятся как "прозрачные", если $IM=10$, некоторая часть невидимых ребер устраняется. В последнем случае обход вершин основания при перечислении их координат должен быть положительным, т.е. совершаться в направлении от положительного луча оси x к лучу оси y . Противоположное направление обхода принимают, когда надо изобразить призматическое отверстие.

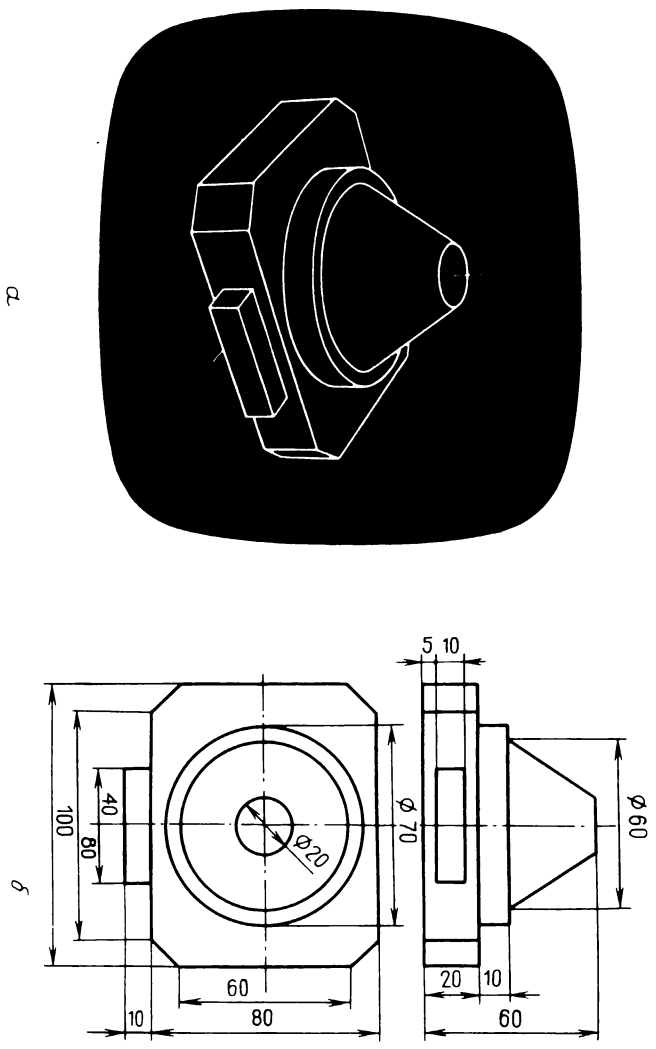
Круговые конусы (в общем случае подразумеваются усеченные конусы, в частных случаях могут быть цилиндры, конусы с вершинами) расположены в ЧКС так, что их оси параллельны оси z ЧКС (рис. 99). Смысл геометрических параметров понятен из рисунка.

На фортране: `CALL KONUS2(XC,YC,ZC,R1,R2,H)`

На языке алграф: `КОНУС,xc,yc,zc,r1,r2,h,...`

Подпрограмма в зависимости от ракурса частично устраняет участки невидимых контуров оснований и находит очерковые образующие, которые и вычерчиваются (если они есть).

Подобным же образом задаются и другие типовые фигуры. Для размещения фигур в общем положении предлагается использовать преобра-



а

Рис. 100. Изображение детали, составленное из типовых элементов

зования координат. Из фигур и дополнительных линий можно с той или иной степенью обобщенности формировать тела деталей, архитектурные композиции и т.д.

На рис. 100, *a* изображена деталь, образзанная из призмы, параллелепипеда, кругового цилиндра и конуса. Вот ее кодировка на языке алграф:

```
ПРБАКС,АКС,0,0,0,0,30,30,2,  
ПРПН,-20,5,-50,40,10,10,ХZY,ПРИЗМН,0,25,-50,-30,-40,-40,  
40,-40,50,-40,50,40,40,40,-40,40,-50,30,В, КОНУС,0,0,25,35,  
35,10,0,0,30,30,10,30,СТОП,&
```

Первую строку, в которой задается проекция, пока не обсуждаем. Код ХZY, подключающий смену координатных осей, позволяет изобразить призму и конусы с основаниями в "горизонтальных" плоскостях.

Теперь представим себе, что в призматическом объеме этой детали выфрезерован паз, а по оси конусов просверлено сквозное отверстие. Типовые фигуры и подпрограммы на все такие случаи подготовить трудно, а моделирование подобных деталей необходимо не только для их машинного расчерчивания, но и для подсчета массы, объема, расчета деталей и узлов на прочность и т.д. В разных странах уже действуют программные системы, позволяющие моделировать тела не только из более простых составляющих, как мы только что делали, но и производя операции "отсечения", "разрезания" и др.

ТРЕХМЕРНЫЕ ПРЕОБРАЗОВАНИЯ КООРДИНАТ

Как и на плоскости, преобразования в пространстве могут быть разных типов, здесь чаще всего применяются аффинные преобразования и особенно — их частные случаи, моделирующие движение в пространстве тел неизменной формы и изменение масштаба. Из нелинейных систем отсчета чаще встречаются цилиндрические и сферические.

7.1. Аффинные преобразования в пространстве

О геометрических свойствах трехмерных аффинных преобразований уже говорилось; уравнения преобразований имеют ту же структуру, что и при двух координатах, но теперь их три:

$$x_2 = a_{xx}x_1 + a_{xy}y_1 + a_{xz}z_1 + a_x,$$

$$y_2 = a_{yx}x_1 + a_{yy}y_1 + a_{yz}z_1 + a_y,$$

$$z_2 = a_{zx}x_1 + a_{zy}y_1 + a_{zz}z_1 + a_z.$$

Расширенная матрица здесь имеет 12 элементов:

$$\left[\begin{array}{ccc|c} a_{xx} & a_{xy} & a_{xz} & a_x \\ a_{yx} & a_{yy} & a_{yz} & a_y \\ a_{zx} & a_{zy} & a_{zz} & a_z \end{array} \right].$$

Из нее можно выделить 9-членную квадратную часть, используемую для преобразований векторов. Тожественная матрица имеет такой вид:

$$\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}$$

Переносы и масштабирование по осям вторичной или первичной системы в пространстве выполняются примерно так же, как на плоскости, только число параметров возрастает на один. Аналогично задается преобразование и определяющими точками, хотя их теперь должно быть четыре. С пово-

ротами в пространстве дело обстоит сложнее, чем на плоскости. Если на плоскости поворачивать объект (или всю первичную плоскость) можно, задавая центр вращения и значение одного угла поворота, то в пространстве поворот задается последовательным вращением вокруг трех координатных осей или вращением вокруг некоторой прямой, определенным образом расположенной в пространстве. Чуть позже мы это проиллюстрируем.

В системе Алграф преобразование этого типа выполняется при значении $IND(5) = 2$ или, на геометрически ориентированном языке, после кода

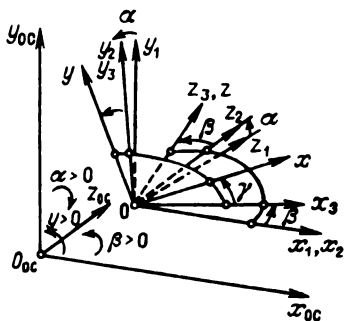


Рис. 101. Схема задания трехосного поворота

ПРБЛЗ. В табл. 7 приводятся коды и названия подпрограмм для формирования и модификации матрицы, для операций переноса, масштабирования и др. Подробнее поясним только действие операторов поворота.

Подпрограмма POV3 (код ПОВЗ), оставляя начало ЧКС в прежнем месте, заново формирует всю квадратную часть матрицы так, что преобразование из ЧКС в ОКС будет *движением*, т.е. изометрическим преобразованием с сохранением ориентации. Представим себе подвижные координатные оси (рис. 101). В первый момент по направлению (и масштабам)

Т а б л и ц а 7

Некоторые операторы трехмерных аффинных преобразований

Операция	Параметры	Оператор на фортране	Оператор на языке алграф-Р
1	2	3	4
Задание тождественного преобразования	нет	CALL MATT(11)	МТЛЗ
Засылка матрицы в рабочее поле	члены матрицы (12 чисел)	CALL MATL3 *(C1, C2, ... C12)	МАТЛЗ, с 1, с 2, ... с 12,
Перенос ЧКС на вектор, заданный в ОКС	компоненты вектора в ОКС	CALL PERENP *(DX, DY, DZ, O)	ПЕРЕНП, dx, dy, dz

Т а б л и ц а 7 (окончание)

1	2	3	4
Перенос ЧКС на вектор, заданный в ЧКС	компоненты вектора в ЧКС	CALL PERENP *(DX1, DY1, *DZ1, 1)	ПЕРП 1, dx1, dy1, dz1
Перенос начала ЧКС в точку	координаты начала ЧКС в ОКС	CALL PERENP *(XO, YO, ZO, 2)	ПЕРП2, xo, yo, zo,
Масштабирование по осям ЧКС	масштабные множители для трех осей	CALL MASHP *(XM, YM, ZM, 0)	МАСШП, xm, ym, zm,
Масштабирование по осям ОКС	— ” —	CALL MASHP *(XMI, YMI, ZMI, *1)	МАСШП1, xм1, ym1, zm1
Задание трехосного поворота (без смещения начала ЧКС)	три угла последовательных поворотов вокруг осей x_1, y_2, z_3	CALL POV3 (ALF, BET, GAM)	ПОВ3, α, β, γ ,
Дополнительное вращение ЧКС вокруг заданной в ОКС оси	координаты двух точек на оси вращения (на фортране задаются в массиве U) и угол поворота	CALL POVE (U, ALF)	ПОВЕ, x1, y1, z1, x2, y2, z2, alf, ...
Задание преобразования движения тремя определяющими точками	координаты начала ЧКС в ОКС, координаты какой-либо точки на положительном луче X ЧКС и точки, для которой $U = 0$ (на фортране задаются в массиве A)	CALL POV3 (A)	ПОВД, xo, yo, zo, x1, y1, z1, x2, y2, z2
Умножение матриц (в некотором массиве и на рабочем поле)	ADR – имя массива или метка, где хранится матрица-сомножитель; K – порядок умножения ($K = 1$ или $K = 3$)	CALL UMT (ADR, K)	УМТ (K, adr)
Пересылка матрицы из рабочего поля или в рабочее поле	ADR – имя массива или метка, куда (откуда) пересылается матрица; K – направление пересылки ($K = 0$ для пересылки в поле или $K = 1$ при пересылке из поля).	CALL MATR (ADR, 12, M) $M = K * 10 + 1$	МАТР (m, adr), где $m = 1201 + K * 10$.

они совпадают с осями ОКС; затем как жесткое целое эти оси поворачиваются вокруг оси x_1 на угол ALF (см. табл. 7) (α); в новом положении оси назовем x_2, y_2, z_2 (x_2 совпадает с x_1). Далее, осуществляется поворот на угол BET (β) вокруг оси y_2 (уже наклонившейся на угол α). Получаем оси x_3, y_3, z_3 (y_3 совпадает с y_2). Наконец, оси поворачиваются вокруг z_3 на угол GAM (γ) и получаются окончательные направления осей x, y, z (z_3 совпадает с z) нашей ЧКС. Такая операция называется *трехосным поворотом*. Из аналитической геометрии известно, что последовательно выполняемые повороты *некоммумутативны*, т.е. результат *зависит* от порядка, в котором повороты совершались. У нас порядок вращений "заложен" в подпрограмму. Если по смыслу задачи нужен другой порядок, с помощью этого же оператора можно задать последовательно матрицы *одноосных поворотов* (это делается с помощью той же подпрограммы при задании ненулевым одного из углов-параметров). Матрицы надо запомнить, а затем в нужном порядке перемножить. Можно использовать более сложную подпрограмму, где задаются номера осей, вокруг которых совершаются последовательные повороты. Вообще, любое заданное направление осей первичной КС можно получить, выполняя три вращения вокруг двух координатных осей; в этом случае для первого и третьего поворотов берется одна ось, например ось x , но в процессе преобразования она сама поворачивается. На рис. 101 можно было бы первые два поворота вокруг осей x_1 и y_2 сохранить, а третий осуществить вокруг оси x_3 , а не z_3 . Использовать такие углы предложил знаменитый Леонард Эйлер, исследовавший вращения планет и волчков.

Операция *дополнительного вращения* в пространстве выполняется подпрограммой POVE (кодом ПОВЕ). Здесь за исходное берется предыдущее

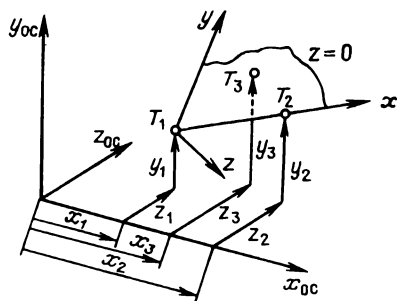


Рис. 102. Задание изометрического трехмерного преобразования точками

преобразование и все первичное пространство с осями ЧКС как жесткое целое поворачивается на заданный угол вокруг оси, заданной в ОКС. Положение оси определяется координатами двух ее точек, а порядок этих точек указывает на ось направление, в котором нужно смотреть, чтобы правильно выбрать знак угла поворота (положительные углы соответствуют вращению против часовой стрелки). Итак, после действия оператора *дополнительного вращения* преобразование из ЧКС в ОКС не обяза-

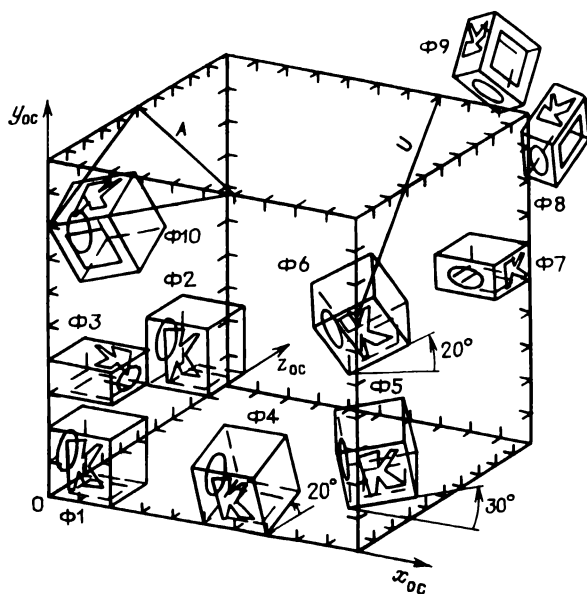


Рис. 103. Пример на трехмерные аффинные преобразования

тельно будет преобразованием движения; ведь операция вращения как бы умножается на предыдущее преобразование. В нужных случаях перед операцией вращения можно сформировать тождественную матрицу.

Порой углы поворотов нелегко замерить на исходном чертеже; проще опираться на координаты точек. Но при задании преобразования определяющими точками нет гарантии, что мы получим движение, т.е. изометрическое преобразование. Поэтому для задания преобразования движения точками разработан специальный алгоритм (подпрограмма POVД, код ПОВД). Согласно рис. 102, первая из точек, заданных в ОКС, определяет начало ЧКС (точка T_1), вторая расположена где-то на положительном луче оси x первичной системы, и расстояние от нее до начала ЧКС не имеет значения; третья точка размещается где-то в плоскости $z = 0$, точнее — в той ее полуплоскости, для которой значения координаты y положительны.

Уточнить обращения к подпрограммам можно по табл. 7.

На рис. 103 приведен пример на аффинные трехмерные преобразования. В качестве тестовой фигуры отдельно был закодирован кубик, одна из вершин которого соответствует началу ЧКС, а три ребра идут по ее осям. Грань кубика с буквой K лежит в плоскости $z_{oc} = 0$. "Большой" куб с делениями показывает расположение осей ОКС. Фигурка Ф1 здесь соответствует тождественному преобразованию, Ф2 — переносу, Ф3 — масштабированию по осям ЧКС (при этом два отрицательных множителя вызывают изменение направления осей ЧКС на противоположное). Фигуры Ф4,

Ф5 и Ф6 показывают как бы последовательные стадии в трехосном повороте; дополнительные операции переноса позволяют избежать наложения фигур друг на друга. Для фигуры Ф7 задана матрица. Фигуры Ф8 и Ф9 получены путем дополнительного вращения вокруг прямой общего положения (ось отмечена отрезком U). Фигура Ф10 задана оператором, определяющим движение точками. Эти точки – углы дополнительно нанесенного треугольника А. Приводим текст кодировки на языке алгебра-Р (желающие могут написать программу на фортране).

ПРБАКС, АКС, 0, 0, 0, 0, -30, 20, 0.7, ПРБЛЗ, МТЛЗ, Ф1: Б (КУ, Б), Ф2: ПЕРЕНП, 20, 30, 20, Б (КУ, Б), Ф3: МАСШП, -1,0.5, -1, Б (КУ, Б), Ф4: ПЕРП2, 50, 0, 0, ПОВЗ, -20, 0, 0, Б (КУ, Б), Ф5: ПЕРЕНП, 0, 0, 30, ПОВЗ, -20, 30, 0, Б (КУ, Б), Ф6: ПЕРЕНП, 0, 40, 0, ПОВЗ, -20, 30, 20, Б (КУ, Б), Ф7: МАТЛЗ, 0, 0, -1, 100, 0, 0.5, 0, 50, 1, 0, 0, 80, 80, Б (КУ, Б), Ф8: ПОВЕ, 100, 70, 0, 70, 100, 100, 40, Б (КУ, Б), Ф9: ПОВЕ, 100, 70, 0, 70, 100, 100, 40, Б (КУ, Б), Ф10: ПОВД, 0, 80, 0, 60, 100, 0, 0, 100, 50, Б (КУ, Б), ПРБ1, ОТРЕЗП, 100, 70, 0, 70, 100, 100, ЛИНП, 0, 80, 0, П, 60, 100, 0, 0, 100, 50, В, Б (ШК, АЛ), СТОП, ...

7.2. Нелинейные координаты в пространстве

Цилиндрические, сферические, проективные и различные другие координаты в пространстве используются в нужных случаях, они и соответствующие преобразования позволяют различным образом деформировать пространство. Подробно эти преобразования здесь не описываются, но на рис. 104 показана пространственная форма, образованная из двумерной линии, которая была помещена в сферическую координатную систему и "размножена" с помощью преобразований.

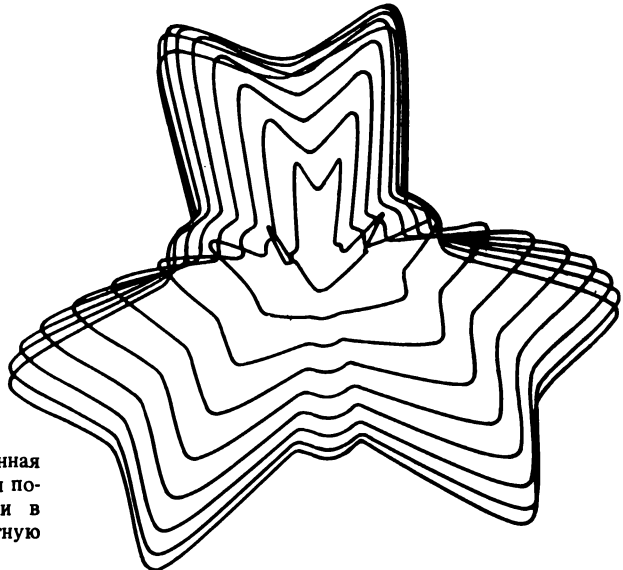


Рис. 104. Пространственная форма, полученная путем помещения плоской линии в сферическую координатную систему

ПОСТРОЕНИЕ ПРОЕКЦИЙ ПРОСТРАНСТВЕННЫХ ОБРАЗОВ

Проблема наглядного изображения на плоскости объемных тел и ситуаций так же стара, как само человечество. Еще первобытные люди оставили на стенах пещер изображения своих безусловно объемных сородичей и зверей. Любопытно, что большинство древних рисунков, в том числе принадлежащих эпохе Древнего Египта, да и средневековую – причем рисунков порой очень выразительных, показывают человека и животных или сбоку, или, как иногда говорят, анфас (рис. 105); иногда одна часть тела изображена в профиль, а другая фронтально. Очевидно, легче было постичь ортогональные виды предметов, в которых линия, силуэт более выразительны и определены, чем более реалистические, но сложные промежуточные ракурсы. Особый интерес к достоверности рисунка, правильности в передаче объемных предметов и сцен возникает в эпоху Возрождения. Альбрехт Дюрер, Леонардо да Винчи, другие художники много сделали в осмыслении процесса зрительного восприятия окружающего нас объемного мира и плоской картины, моделирующей этот мир. Искали связь между восприятием, искусством и геометрией, изобретали механические приспособления, позволяющие правильно построить рисунок (рис. 106). Был выработан основной подход к построению наглядных изображений: картина рассматривалась как своеобразное "окно", сквозь которое зритель видит пространство. "Заменяя" пространство на картину, надо "заменить" лучи света, идущие в глаз наблюдателя от натуральных предметов, на лучи, идущие от точек картины. Своего рода "идеал" – когда направления лучей после такой замены не изменяются. Конечно, зрительное восприятие как картины, так и природы – процесс сложный, и, если уточнять, в ход вступает множество дополнительных факторов – аккомодация зрачка (его фокусировка, настройка на разглядывание близких или далеких предметов), конвергенция двух глаз (непараллельность зрительных осей глаз при рассматривании близких предметов), стереоскопический эффект, существующий благодаря наличию двух глаз и позволяющий нам более объемно, полно видеть мир, однако "выдающий" плоскостность картинной поверхности, и т.д. В свое время для получения рисунков

Рис. 105. Рисунки наших предков



а



б



в

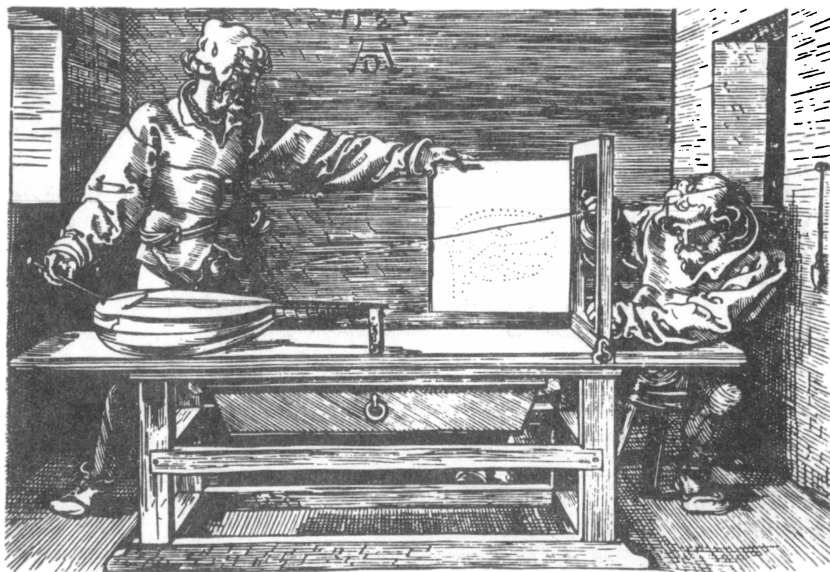


Рис. 106. Репродукция с гравюры Дюрера, на которой изображен процесс построения перспективы

применялась камера-обскура, представлявшая собой непрозрачный ящик с маленьким отверстием в передней стенке и экраном в виде матового стекла вместо задней стенки или даже специальную комнату с небольшим отверстием в ставнях, через которое в солнечный день на противоположную стену отбрасывалось хотя и перевернутое, но реалистическое изображение окружающего пейзажа. Теперь мы уже так привыкли к фотографиям, кино и телевидению, где в основном действуют те же геометрические правила построения изображений, что такие изображения кажутся нам правильными и естественными (за исключением особых случаев, о которых будет упомянуто). Общепринято такие изображения называть *перспективными*. Одно из основных свойств этих изображений – то, что более удаленные предметы изображаются в меньших масштабах. Параллельные прямые в общем случае на изображении непараллельны. В итоге геометрия изображения оказывается достаточно сложной; построить такое изображение не очень просто, а по готовому изображению трудно определить размеры тех или иных частей объекта. По этой причине в проектировании, конструировании продолжают широко использовать

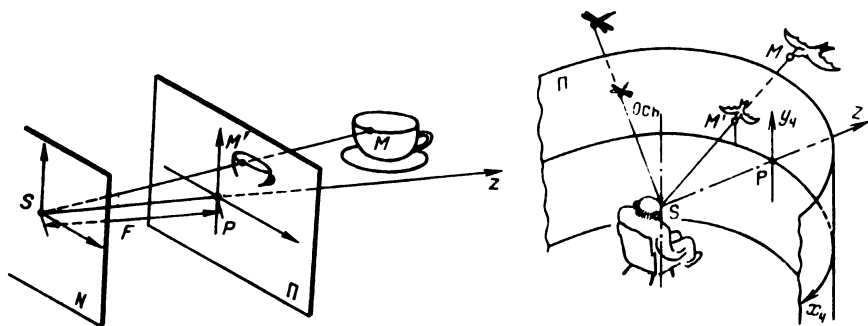


Рис. 107. Геометрическая схема построения центральной проекции на плоскость: Π – картинная плоскость, P – главная точка картины, z – главный луч зрения, S – точка зрения (центр проекции), N – нейтральная плоскость, F – главное (фокусное) расстояние

Рис. 108. Схема построения проекции на цилиндрической поверхности: z – условный "главный" луч, Π – цилиндрическая картинная поверхность

ортогональные виды объектов – вид спереди, сверху, сбоку и т.д., а также *аксонометрические проекции* – по своей геометрии более простые, чем перспективные.

Во всех описанных выше случаях картина представляется плоской (*картинная плоскость*). Изображение на нее переносится с точек натуры прямыми лучами. Эти *проецирующие лучи* либо все параллельны, и тогда мы имеем ортогональный вид или аксонометрическую проекцию, либо проходят через точку (глаз наблюдателя, или центр проецирования) и

образуют перспективную (центральную) проекцию. Схема построения таких проекций показана на рис. 107.

Когда мы хотим показать широкую панораму или окружающее нас пространство, плоская картина становится неэффективной. Можно представить себе зрителя окруженным цилиндрической (рис. 108) или даже сферической (полусферической) поверхностью, на которую нанесено изображение. Если знать геометрические правила, на плоскости можно нарисовать изображение, которое в виде развертки представит собой проекцию, затем картину можно изогнуть нужным образом, и тогда изображение не будет искажено. Заметим, что прямые линии, на развертке изогнувшись, при правильном рассматривании представятся наблюдателю прямыми. Развернуть и свернуть цилиндрическую или коническую картинную поверхность легко, со сферической поверхностью этого не сделаешь. Вместо этого можно применить вогнутый, сферический экран, на который с плоской пленки или пластинки проецируется изображение. Чтобы на экране изображение воспринималось правильным, на пленке оно должно быть построено по специальным алгоритмам. Такие разнообразные проекции называются *специальными*.

8.1. Построение в машинной графике простых аксонометрических проекций

Под аксонометрическими проекциями понимаются проекции на плоскость с помощью параллельных лучей. Среди этих проекций выделяются *ортогональные* – те, у которых проецирующие лучи перпендикулярны к плоскости изображения. Поскольку и рассматриваем мы изображения обычно взором, перпендикулярным (или почти перпендикулярным) к его поверхности, такие проекции представляются более правильными, чем косоугольные (рис. 109). Полученную таким образом проекцию можно равномерно увеличить или уменьшить (как бы применив уже двумерное преобразование подобия).

Нетрудно проверить, что ортогональные виды, в том числе и "главный" вид, можно считать частными случаями параллельных проекций – когда плоскость изображения совпадает с одной из координатных плоскостей, а проецирующие лучи параллельны одной из осей координат. Можно также видеть, что в нашей пространственной модели проецирования смещение плоскости картины параллельно себе не меняет изображения. Полагаем здесь, что проецировать на картинную плоскость можем точки, расположенные в пространстве как перед ней, так и за нею.

Как можно реализовать построение подобных проекций в графической системе? В системе Алграф "главный" вид пространственного объекта получается, если объект описан как пространственный, а преобразованием координат второй ступени (см. рис. 78) не задано. Тогда просто две первые координаты ОКС становятся координатами в системе чертежа КСЧ, а третья координата z_{oc} "теряется". Геометрически это соответствует

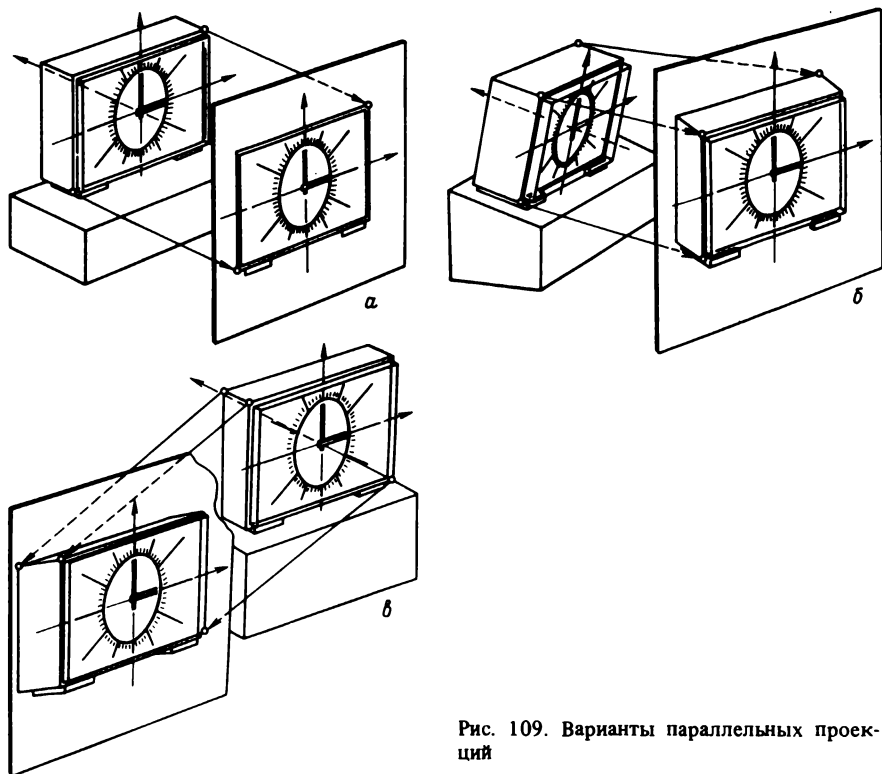


Рис. 109. Варианты параллельных проекций

проецированию объекта лучами, параллельными оси z_{OC} , на плоскость $z_{OC} = 0$ (рис. 110, а). Как получить другие проекции? Введем еще одну пространственную трехмерную систему координат (ПРКС) – такую, что первые ее две оси совпадают с осями системы чертежа, которую мы пока считаем перемещаемой в пространстве вместе с картинной плоскостью. Третья ось перпендикулярна к плоскости картины. Разместим картинную плоскость с этой ПРКС в каком-либо положении относительно объекта (рис. 110, б). Оси ОКС и ПРКС в общем случае будут непараллельны. Но преобразование из ОКС в ПРКС, если найти параметры преобразования, будет, очевидно, аффинным, причем частным его случаем – движением (если масштабы по осям ОКС и ПРКС выбраны одинаковыми). Пусть мы нашли коэффициенты преобразования. Тогда можем высчитывать координаты точек объекта в ПРКС, коль скоро они известны в ОКС. Затем первые две координаты ПРКС пересылаются в координатную систему чертежа – и получаем данные для проекции. Если масштабы по осям ПРКС взять отличными от масштабов в ОКС – проекция будет увеличенной или уменьшенной. Итак, изометрическое аффинное преобразование из ОКС

в ПРКС порождает построение ортогональной параллельной проекции. Изометрическое "в расширенном смысле"*) преобразование дает проекцию с изменением масштаба. Если взять общий случай аффинного преобразования — будет получена косоугольная параллельная проекция. Вообще говоря, при подсчете координат в ПРКС третью координату ($z_{пр}$) можно было бы не вычислять, так как конфигурация изображения от нее не зависит. Однако в некоторых подпрограммах значение этой координаты используется для определения видимости элементов объекта, поиска очерковых линий (линий видимого контура) и в других целях.

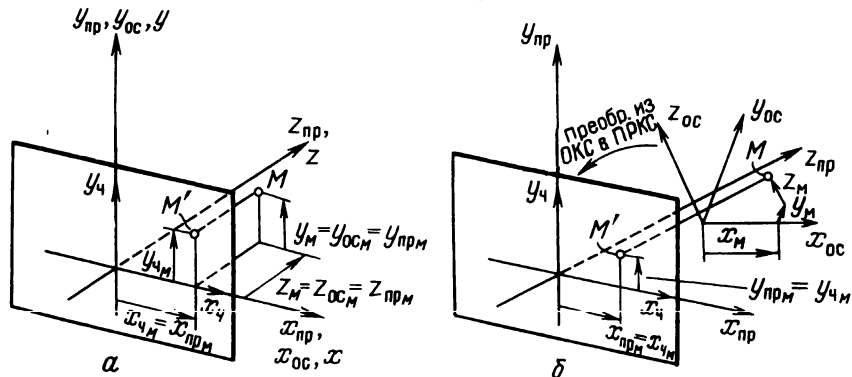


Рис. 110. Использование преобразований координат для построения параллельной проекции

В системе Алграф для перевода координат из ОКС в ПРКС и КСЧ используется вторая ступень преобразований. Коэффициенты этой ступени хранятся в рабочем поле АМАТ2; координаты берутся из ячеек ХОS, YOS, ZOS, а подсчитанные вторичные координаты засылаются в ячейки ХPR, YPR, ZPR и затем первые две из них — в ячейки XD, YD. Для косоугольной проекции и в тех случаях, когда на чертеже известны направления осей ОКС и масштабы по ним (рис. 111), элементы матрицы можно задать непосредственно (хотя бы для первых двух строк преобразования). Но подобрать значения коэффициентов для ортогональной проекции не так просто. Нужно соблюдать шесть условий ортогональности-изометричности, выражаемых уравнениями, в которых значения коэффициентов записаны в квадрате. Решить систему из шести нелинейных уравнений сложно. Поэтому матрицы для ортогональных проекций формируют с помощью специальных подпрограмм, исходные данные для которых легко снять с исходного эскиза или определить иными способами. Первый из операторов для определения матрицы использует повороты (сравните

*) Имеется в виду, что определитель матрицы $\neq 1$, но оси координат взаимно ортогональны, и масштабы по осям одинаковы.

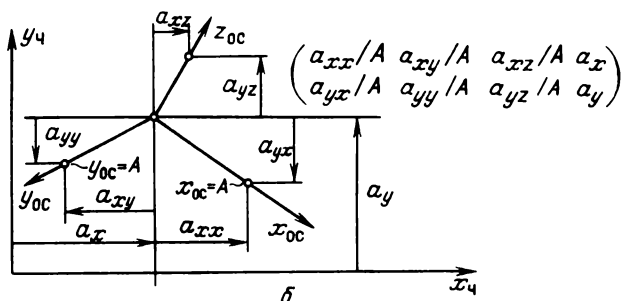
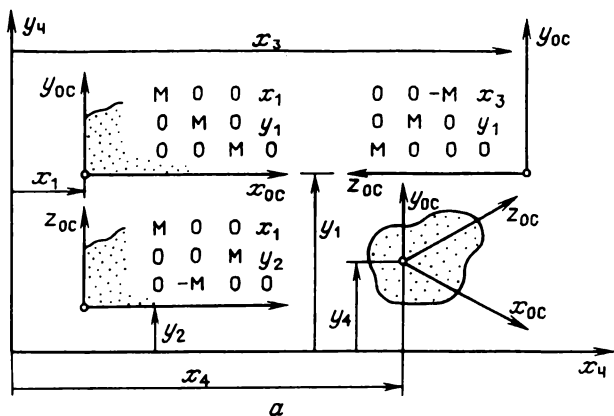


Рис. 111. Задание матриц для построения главных видов (а) и графическое определение матрицы для аксонометрической проекции (б)

с описаниями поворотов в разд. 7.1). На рис. 112 показана схема задания проекции. Считая картинную плоскость и ПРКС неподвижными, будем перемещать и поворачивать ОКС относительно них. Пусть в первый момент оси ОКС совпадают с осями ПРКС $x_{пр}, y_{пр}, z_{пр}$. Перенесем их параллельно себе на заданный вектор с компонентами x_0, y_0, z_0 (заданными в ПРКС). Получим промежуточное положение осей x_1, y_1, z_1 ; поворачиваем их вокруг вертикальной оси y_1 на угол ALF , затем вокруг оси x_2 на угол GAM и вокруг уже наклонившейся оси z_3 на угол BET (здесь используются углы Эйлера; поворот дважды совершается вокруг одноименной оси). Последняя операция – подобное преобразование всей ОКС (включая вектор переноса, заданный компонентами x_0, y_0, z_0), соответствующее заданному масштабному множителю AM . В результате получается матрица *изометрического* (в расширенном смысле) преобразования. В программе на фототране этот оператор вызывается так:

CALL AKS (X0, Y0, Z0, ALF, BET, GAM, AM)

На языке алграф-Р те же по смыслу параметры записываются через запятые после кода АКС. Обращаем внимание, что порядок записи углов поворотов в операторе не соответствует порядку выполнения поворотов.

Этот способ задания проекций достаточно удобен, когда отображаемые объекты представляются пользователю небольшими, подвижными. Параметры x_0 и y_0 позволяют располагать проекцию в желаемом месте чертежа; параметр z_0 обычно задается нулевым, но его значение видимого влияния на проекцию не оказывает. Изменяя углы Эйлера, можно

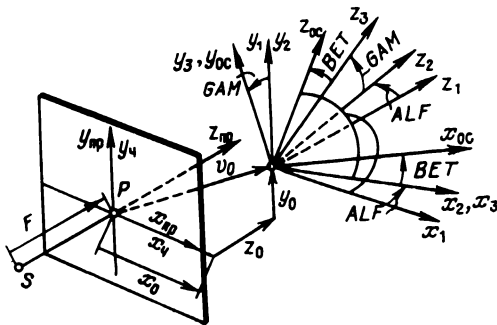


Рис. 112. Первый способ задания параметров аксонометрической (и перспективной) проекции

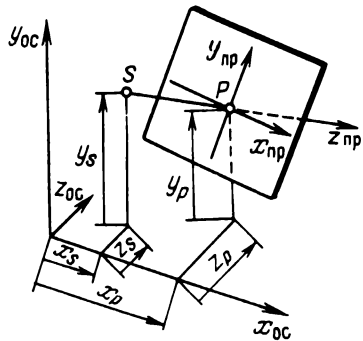


Рис. 113. Второй способ задания параметров аксонометрической проекции

придать объекту любое положение относительно наблюдателя. Однако на практике угол ALF почти всегда задают нулевым. При таком его значении прямые, в пространстве параллельные оси y_{oc} , будут параллельны оси y_c на изображении, т.е. вертикальны. Тогда угол BET обеспечивает вращение объекта перед наблюдателем "вокруг вертикали", а угол GAM — наклон проецирующих лучей к горизонтальной плоскости.

Второй способ задания проекции больше подходит для "крупных" объектов. Перефразируя поговорку о горе и Магомете, можно сказать, что если "не движется объект перед наблюдателем", то "наблюдатель ходит вокруг объекта". Как показано на рис. 113, теперь мы считаем ОКС неподвижной и в ней задаем положение ПРКС и картинной плоскости. Координатами в ОКС задаются две точки: начало будущей ПРКС (совмещенное в нашей пространственной модели с началом системы чертежа (точка P)) и какая-либо точка S на отрицательном луче оси z_{np} . Ось z_{np} в нашей схеме играет роль *главного проецирующего луча*, а начало ПРКС — *главной точки картины* (P). Подпрограмма проводит через две заданные точки главный луч, через главную точку перпендикулярно к лучу строит картинную плоскость, в ней "горизонтально", т.е. параллельно плоскости $y_{oc} = 0$, проводится ось x_{np} ; наконец, перпендикулярно к x_{np} в картинной плоскости проводится и ось y_{np} . Как и при первом способе задания проек-

ции, здесь можно использовать масштабирование. Данный способ порождает не любые проекции, а только такие, которые соответствуют нулевым значениям угла ALF (см. рис. 112). Любопытно, что проекция при этом способе задания не изменяется при перемещении главной точки и точки S вдоль неподвижного главного луча (по крайней мере пока не меняется относительный порядок этих точек на луче).

В программе на фортране задать проекцию описанным способом можно с помощью оператора

CALL AKS1 (XP, YP, ZP, XS, YS, ZS, AM)

где в качестве параметров сначала задаются координаты главной точки P , затем — точки S и наконец записывается масштабный множитель AM . В тексте на геометрически-ориентированном языке эти же параметры записываются после кода АКС1.

На рис. 114 приведен тестовый пример. Тот же кубик, что использовался для иллюстрации аффинных преобразований, изображаем в различных проекциях применительно к одной системе чертежа. Вот основная часть текста геометрического описания:

НАЧ: ПРБАКС, МАТР (1202,0), 0.5,-0.5,-2,-60,8,1.2,-1,20,0,0,0, 0,А: Б (КУ,Б),
 МАТР(1202,0),1,0,,353,-30,0,1,,353,20,0,0,0,0, В: Б (КУ,Б),
 АКС, 20,40,0,0,0,0,1, С: Б (КУ,Б), АКС,20,10,100,0,0,90,1, D: Б (КУ,Б),
 АКС,80,60,0,0,30,0,0.7, Е:Б (КУ,Б),
 АКС,80,10,0,0,30,40,0.7, F:Б (КУ,Б),
 АКС,80,-30,0,40,0,40,0.7, Q:Б (КУ,Б),
 АКС1,20,20,20,0,0,0,,7,Н:Б (КУ,Б),
 АКС1,40,20,0,10,20,-50,1,К:Б (КУ,Б), ПРБ2, Б (ШК2,АЛ2), СТОП,

Код ПРБАКС здесь задает тип преобразования второй ступени (см. табл. 4); матрицы для первых двух фигур заданы непосредственно, при-

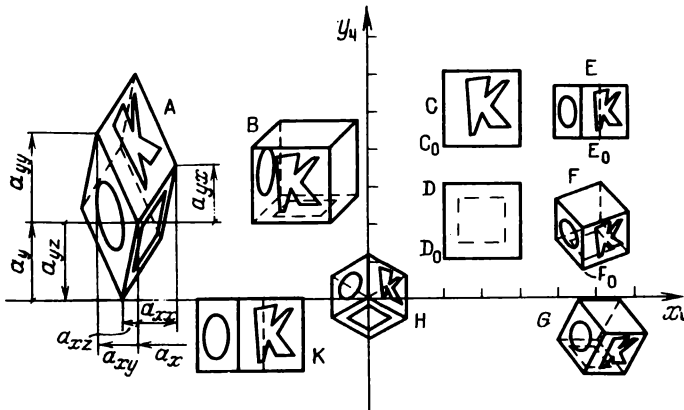


Рис. 114. Пример машинного построения аксонометрических проекций

чем на рис. 114 дополнительно помечены размеры, с помощью которых можно записать первые две строки матрицы. Учитывая, что длина ребра кубика в ЧКС 20 мм, эти строки можно записать так:

$$\begin{bmatrix} a_{xx}/20 & a_{xy}/20 & a_{xz}/20 & a_x \\ a_{yx}/20 & a_{yy}/20 & a_{yz}/20 & a_y \end{bmatrix}.$$

Фигура В вычерчена в так называемой кабинетной проекции, нередко используемой в практике конструирования; ее передняя грань изображается в "натуральном виде" (проверьте по матрице, так ли это?), а боковые ребра наклонены к оси x КСЧ под углом 45° . Фигуры С и Д моделируют "вид спереди" и "вид сверху", расположенные в проекционной связи. Фигуры Е и F показывают как бы две стадии движения кубика – с поворотом на один угол и с добавлением второго поворота; точками C_0, D_0, E_0, F_0 помечены начала ОКС. Сравните проекции F и G: обе они получены при двух ненулевых углах поворотов, но для одной использованы углы *BET* и *GAM*, а для другой – *ALF* и *GAM*. Фигуры К и Н демонстрируют второй способ задания проекции. Для одной из них главный проецирующий луч проходит в плоскости верхней грани кубика, для другой – инцидентен его пространственной диагонали.

Возвращаясь к рис. 92, 100, можно проанализировать, как для них были заданы аксонометрические проекции.

8.2. Построение простых перспективных проекций

Итак, обычная перспективная проекция – это центральная проекция на плоскость прямыми лучами, проходящими через точку – *центр проецирования*, или *точку зрения*. Один из проецирующих лучей перпендикулярен к картинной плоскости, и он называется *главным*; у нас он будет совмещен с осью $z_{пр}$ пространственной системы отсчета. Точка пересечения этого луча и картинной плоскости – *главная точка картины*, она же – начало ПРКС. Расстояния от главной точки до центра проецирования называется *главным*, или *фокусным расстоянием*. Как вы думаете, есть у этого понятия что-то общее с фокусным расстоянием объектива фотоаппарата или кинокамеры? Через центр проекции еще полезно провести плоскость, параллельную картинной. Она называется *нейтральной*. Расположенные в ней точки пространства нельзя спроецировать на картинную плоскость. Говорят, что они при такой попытке становятся несобственными, т.е. бесконечно удаленными. Точки, расположенные за нейтральной плоскостью, формально можно было бы спроецировать, но подумайте, как они отобразятся и стоит ли этим заниматься?

Из специальных курсов можно познакомиться с геометрическими свойствами перспективных проекций. В частности, прямые линии отображаются как прямые, но параллельные прямые в общем случае на картине не парал-

лельны. Параллельные прямые, расположенные в горизонтальной плоскости, изображаются сходящимися на линии горизонта.

Но нас здесь интересуют в основном аналитические способы построения таких проекций и способы определения их параметров. Будем получать проекции с помощью двух подступеней преобразования второй ступени. Сначала с помощью такого же аффинного преобразования, которое использовалось при построении аксонометрических проекций, находим координаты точек в ПРКС – системе, связанной с чертежом. Далее выполняем *каноническое* (т.е. в некотором смысле простейшее) преобразование, откуда и определяются координаты точек в КСЧ (в программах – переменные XD, YD). Из подобия треугольников, образуемых проецирующими и координатными отрезками и заштрихованных на рис. 115, можно вывести формулы канонического преобразования и записать их на фортране так:

$$\begin{aligned} ZPR1 &= ZPR + F \\ XD &= XPR * F / ZPR1 \\ YD &= YPR * F / ZPR1 \end{aligned}$$

Здесь F – фокусное расстояние, ZPR1 можно считать координатой точки в модифицированной пространственной системе, которая получается из ПРКС параллельным переносом начала в центр проекции (первые две координаты точки в этой системе те же, что в ПРКС).

Для задания перспективных проекций можно использовать операторы, близкие к тем, которыми задавались аксонометрические проекции. Один из них такой:

CALL PERSP (XO, YO, ZO, ALF, BET, GAM, AM, F)

Из новых параметров здесь добавлен только один – фокусное расстояние F. Тип преобразования задается в соответствии с табл. 4 (IND (6) = 2).

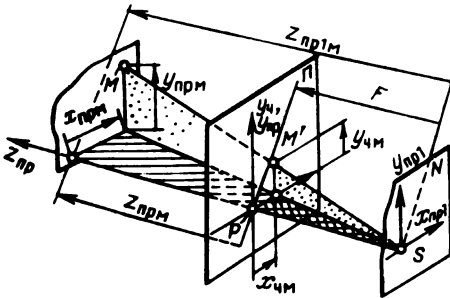


Рис. 115. Схема построения центральной (перспективной) проекции: П – картинная плоскость, N – нейтральная плоскость, P – главная точка, S – точка зрения, F – фокусное расстояние

На графически-ориентированном языке используется код ПРБПЕР, а параметры проекции задаются после кода ПЕРСП.

Второй способ задания проекции обеспечивается точно тем же оператором, который использовался при задании аксонометрических проекций (подпрограмма AKS1, код АКС1). Но точка S здесь – не любая точка на отрицательном луче оси $z_{пр}$, а центр проекции.

Приведем несколько соображений об особенностях проекций и рациональном выборе их параметров. Во-первых, известно, что более удаленные

объекты изображаются в относительно меньших масштабах, и это явление иногда называется *перспективными сокращениями*. Интенсивность перспективных сокращений зависит, при неизменном положении картинной плоскости, от заданной величины фокусного расстояния. При его уменьшении и одновременном приближении точки зрения к объекту эта интенсивность возрастает (см. рис. 116), одновременно возрастают и углы, под которыми части объекта видны при правильном (расчетном) рассматривании проекции. Возрастает и общий угол зрения — угол между лучами, проведенными в крайние точки изображений (объекта). При больших фокусных расстояниях и небольших углах зрения проекция становится похожей на аксонометрическую; при малых расстояниях и больших углах перспектива

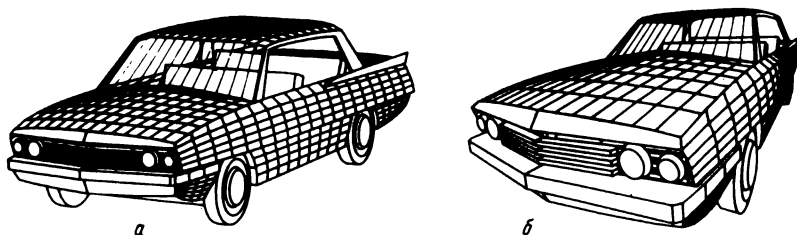


Рис. 116. Примеры изображения объекта, построенного при большом (а) и малом (б) фокусных расстояниях

становится "резкой", появляются замечаемые глазом искажения. Углы между линиями объекта искажаются, размеры по глубине как бы утрируются. Учитывая это, точку зрения стремятся выбирать так, чтобы общий угол зрения не превышал $40-60^\circ$. Впрочем, в значительной степени подобные искажения кажутся. Дело в том, что классическая схема построения проекции предполагает не только правильное нанесение изображения, а и правильное (расчетное) его рассматривание. С учетом принятых масштабов мы должны рассматривать изображение из точки, соответствующей центру проецирования. При этом, если фокусное расстояние было небольшим, мы соответственно должны приблизить глаз к картине. Углы зрения, под которыми будут видны отдельные части объекта, восстановятся. Но, естественно, боковые части картины будем рассматривать под углом (глаз можно поворачивать, но нельзя двигать головой). Кажущиеся перспективные искажения на проекции будут компенсированы, если части изображения будут рассматриваться в определенных ракурсах. Обычная ошибка — когда изображения, построенные для больших углов зрения (так называемые широкоугольные), мы пытаемся рассматривать в мелком масштабе с гораздо более далекого расстояния.

Во-вторых, главная точка картины должна, как правило, размещаться где-то в средней части изображения, ведь картину рассматривают, обычно располагая глаза где-то против ее средней части. Даже если мы смотрим

на картину, укрепленную высоко на стене, стоя сбоку от нее, глаз "исправляет" изображение. Впрочем, в отдельных случаях – при росписи стен и сводов в архитектурных сооружениях, декораций в театре – от этого правила сознательно отступают. Есть даже такое понятие: "театральная перспектива".

Проиллюстрируем перспективные проекции с помощью кубиков (рис. 117). Приводим текст геометрического описания:

ПРБПЕР, ПЕРСП, $-10, -10, 0, 0, 0, 0, 1 \cdot 5, 80$ А : Б (КУ, Б),
 ПЕРСП, $30, -10, 0, 30, 0, 0, 1, 100$, Б (КУ, Б), ПЕРСП, $-40, -10, 0, 0, 30, 0$
 $1, 300$, С : Б (КУ, Б), ПЕРСП, $-10, 30, 0, 0, 0, 45, 1, 200$, D : Б (КУ, Б),
 ПЕРСП, $30, 30, 0, 30, 0, 45, \cdot 8, 200$, Е : Б (КУ, Б), ПЕРСП, $-40, 30, 0, 0, 30,$
 $45, \cdot 8, 200$, F : Б (КУ, Б), ПЕРСП, $30, -50, 0, 0, 10, 10, 1, 40$,
 G : Б (КУ, Б), АКС $1, 20, 50, 0, 20, 50, -70, 1$, H : Б (КУ, Б), АКС $1, 50, 40, 0, 0, 100,$
 $-150, 1$, К : Б (КУ, Б), СТОП, &

Первая проекция (А) – пример так называемой фронтальной перспективы. Две грани объекта параллельны картинной плоскости и изображаются в "истинном виде" с точностью, как говорят, "до подобного преобразования". Прямые, в пространстве параллельные главному лучу, на картине имеют точку схода в главной точке картины. Фигуры В и С заданы с одним ненулевым углом поворота: в этом случае углы АЛФ и ВЕТ работают аналогично. Подумайте, почему на проекции В кубик кажется повернутым больше, чем на проекции С, ведь значения углов одинаковы? На всех трех проекциях, для которых угол ГАМ был нулевым, можем заметить, что вертикальные в пространстве прямые и изобразились как вертикальные. На проекции D это не так: если вертикальные ребра продолжить, они пересекутся

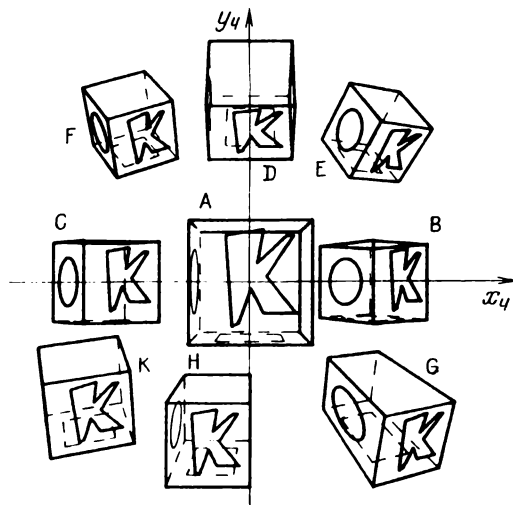


Рис. 117. Иллюстрация построения перспективных проекций

где-то на оси u_c . При использовании углов ВЕТ и GAM отображения вертикальных линий также имеют точку схода на оси u_c (проверьте по фигуре F). Этого нельзя сказать о проекции E, где ненулевыми заданы углы ALF и GAM (фигура явно накренилась набок). Для проекции G фокусное расстояние нарочно задано очень маленьким, поэтому фигура кажется искаженной.

Оператор типа PERSP имеет ту особенность, что части объекта, в пространственной схеме "попавшие" в картинную плоскость, изображаются

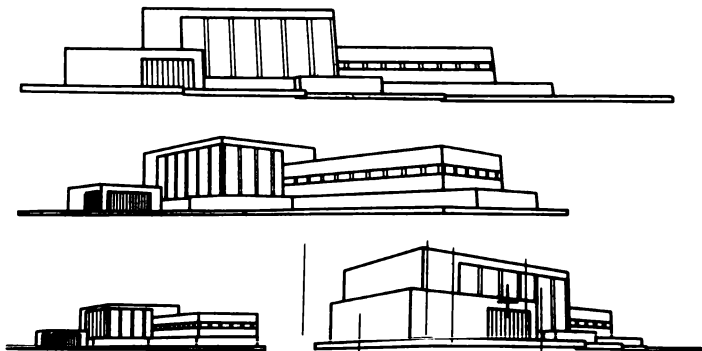


Рис. 118. Изображение архитектурной композиции с различных точек зрения

в одном и том же локальном масштабе при изменениях фокусного расстояния, т.е. при изменении расстояния от объекта до наблюдателя. Рассматривая натуру или пользуясь фотоаппаратом или кинокамерой с обычными объективами, имеющими постоянные фокусные расстояния, мы замечаем, что по мере удаления от объекта он виден как бы уменьшающимся. Поэтому кроме оператора PERSP есть и другие операторы задания проекции. Исходные параметры для них легко замерить на местности геодезическими приборами или другими способами. Координаты центра проекции задаются в пространственной ОКС. Пользуясь такими операторами, машинные проекции можно совмещать с фотоснимками, сделанными точной камерой.

При использовании любых операторов задания проекций можно изменять их параметры и составить программу, которая в цикле будет генерировать множество проекций с постепенно изменяющейся точки зрения. Если кадры с экрана терминала или с отдельных чертежей перенести на киноплёнку, получится машинный мультфильм, где объект будет показан в динамике движения. Для архитектора, художника-конструктора даже 10–20 проекций, показывающих объект в различных ракурсах, с разных расстояний, позволяют более полно и достоверно ощутить его форму, композицию. На рис. 118 приведены несколько проекций одного сооружения.

Замечание: во многих зарубежных и некоторых отечественных системах машинной графики аксонометрические и перспективные проекции строят-

ся с помощью четырехмерного пространства. Используется понятие так называемых четырех однородных координат: уравнения преобразований записываются как однородные (без свободных членов). Этот способ требует большого объема вычислений, повторяющихся для каждой точки объекта. Поэтому автор считает этот способ нерациональным, кроме того, от этого способа труднее перейти к построению специальных проекций.

8.3. Специальные перспективные проекции

Выше были упомянуты центральные проекции на цилиндрические, конические и сферические поверхности. Проекция на цилиндрическую поверхность позволяет показывать объекты с очень большими углами зрения "по горизонтали" – вплоть до круговой панорамы. Вычерчиваются проекции в виде разверток на обычных графических устройствах. В системе Алграф такую проекцию можно построить, заменив только вторую подступень преобразования второй ступени. Такое преобразование будет считаться специальным (см. табл. 4) (IND (6)=4) и выполняться по подпрограмме с именем PRBS3. В этой подпрограмме первую, аффинную подступень преобразования можно выполнить, вызвав один из стандартных модулей PLIN3. Этот модуль должен пересчитать координаты XOS, YOS, ZOS в координаты системы ПРКС XPR, YPR, ZPR. Коэффициенты этой подступени по-прежнему могут храниться в поле AMAT2. Пусть тринадцатый элемент на этом поле – значение фокусного расстояния F, которое принимается за радиус цилиндрической поверхности. Тогда для определения координат точки на поверхности чертежа, используя фортран, можно применить соотношения

$$\begin{aligned}F &= \text{AMAT2}(13) \\ZPR1 &= ZPR + F \\XD &= F * \text{ATAN2}(XPR, ZPR1) \\YD &= F * YPR/\text{SQRT}(XPR ** 2 + ZPR1 ** 2)\end{aligned}$$

Учитывая, что прямые линии на цилиндрической проекции в общем случае искривляются, надо использовать режим предварительной интерполяции, о котором выше упоминалось (см. рис. 78).

Что касается проекций на сферические поверхности – здесь трудность заключается в том, что сфера на плоскость не разворачивается, а "сферических графопостроителей" пока не выпускают. Поэтому приходится перепроецировать изображение со сферической поверхности на плоскость и вычерчивать такую плоскую "перепроекцию". Если точки пространства проецируются на поверхность сферы лучами, проходящими через центр сферы, то перепроецировать их отображения на плоскость можно множеством способов, например ортогональным проецированием параллельными лучами, способом развертки меридианов и др. Среди этих различных способов выделяется *стереографический*; здесь со сферы на плоскость точки перепроецируются прямолинейными лучами, проходящими через

полюс сферы, диаметрально противоположный тому, в котором сфера касается плоскости (рис. 119). Положительное свойство этой проекции — ее *конформность*. Это значит, что углы между пересекающимися линиями на сфере и их отображениями на плоскости равны. Если брать небольшие части изображения, где искривлением прямых можно пренебречь — получим, что отношение длин отрезков, изображенных на сфере, будет тем же и при отображении их на плоскости. А это значит, что при различных, в том числе больших углах отклонения лучей от главного проецирующего достоверно — правда, масштаб их отображения изменяется в зависимости от направления лучей. Как лучше рассматривать такую плоскую проекцию? В отличие от классической "плоской" перспективы здесь лучше каждую часть изображения рассматривать перпендикулярно к ней, т.е. при

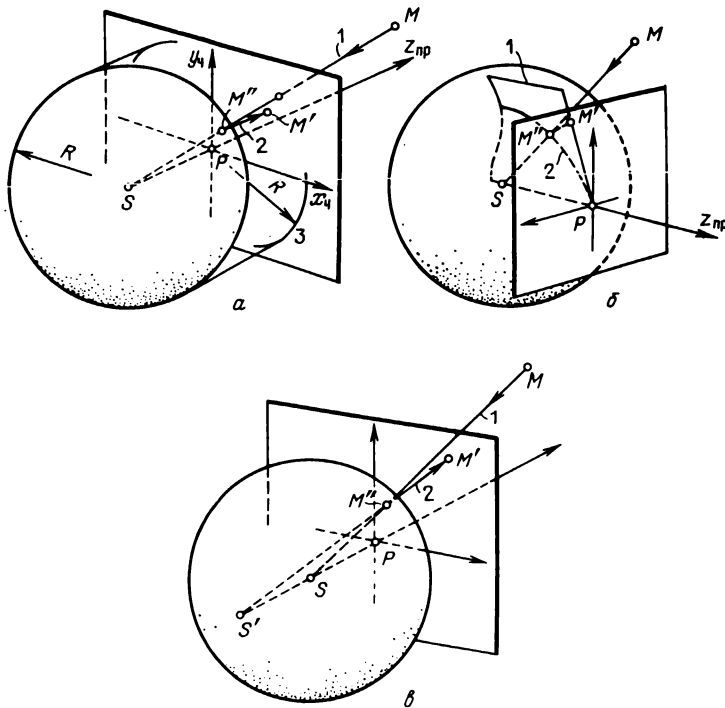


Рис. 119. Схемы квазисферических перспективных проекций: *а* — ортографический вариант перепроецирования (1 — луч центрального проецирования точки на сферу, 2 — луч перепроецирования точки со сферы на плоскость, параллельный оси, 3 — окружность радиуса R , являющаяся границей области отображения); *б* — эквилонгальный вариант (1 — плоскость дуги большого круга, 2 — дуга большого круга); *в* — стереографический вариант (1 — луч проецирования на сферу (инцидентен центру S), 2 — луч перепроецирования со сферы на плоскость (инцидентен центру S')).

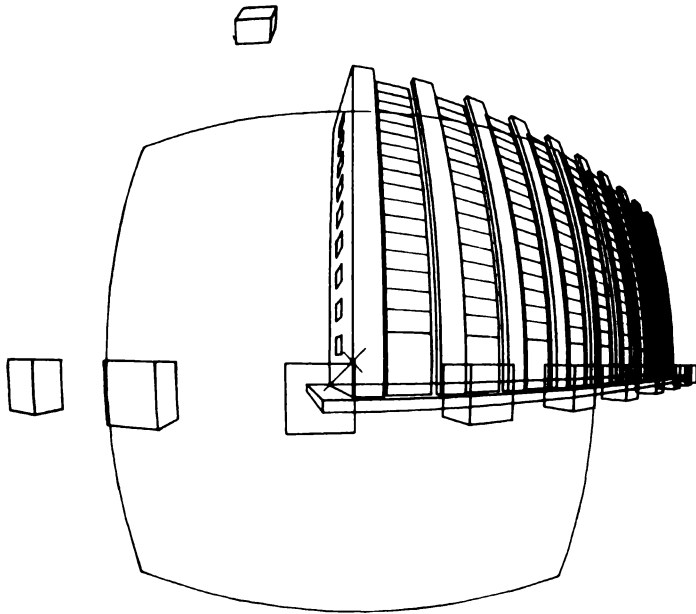


Рис. 120. Пример изображения объекта в "квазисферической" проекции. Границы четырехугольного поля (рамка) соответствуют отклонению лучей вправо, влево, вверх и вниз на 45°; проекция не вполне конформна, так как кубы на периферии несколько сплюснуты

просмотре различных частей проекции голова наблюдющего или рисунок перемещаются. Такое рассматривание можно назвать *фрагментарным*.

В программном отношении задача сходна с предыдущей. Операторы второй подступени преобразования могут быть такими:

$$\begin{aligned}
 F &= \text{AMAT2}(13) \\
 ZPR1 &= ZPR + F \\
 C &= 2 * F / (ZPR1 + \text{SQRT}(XPR ** 2 + YPR ** 2 + ZPR1 ** 2)) \\
 XD &= XPR * C \\
 YD &= YPR * C
 \end{aligned}$$

Пример "квазисферической" перспективы показан на рис. 120.

Еще один вид специальных проекций – стереоскопические. Простейший вид стереоизображения образуется с помощью стереопары – двух перспективных проекций, построенных каждая для "своего" глаза, т.е. с несколько отличных точек зрения. Для объемного восприятия стереопары надо устроить так, чтобы каждый глаз видел свою проекцию. Дости-

гается это разными способами: просмотром маленьких диапозитивов через стереоскоп с линзами; просмотром двух изображений большего размера через зеркальный стереоскоп (рис. 121); с помощью *анаглифических* изображений – т.е. двух наложенных друг на друга проекций, выполненных линиями разного цвета, которые рассматривают через очки с цветными стеклами; наконец, просмотром изображений, проецируемых на экран в поляризованном свете, через поляроидные светофильтры. Все эти способы применяются не только в машинной графике, но часто используются для просмотра фотоснимков. Изображения объектов, составляющие стереопару, для правильной передачи объема картинки должны быть очень

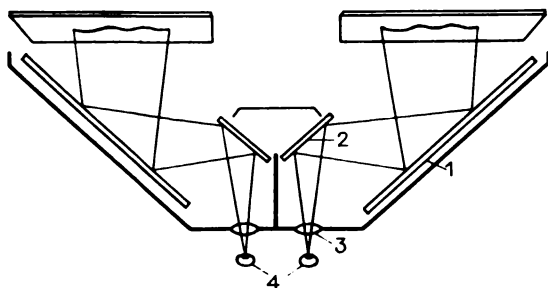


Рис. 121. Схема зеркального стереоскопа (вид сверху, в разрезе): 1 – большое стекло, 2 – малое стекло, 3 – окуляры или смотровые отверстия, 4 – глаза

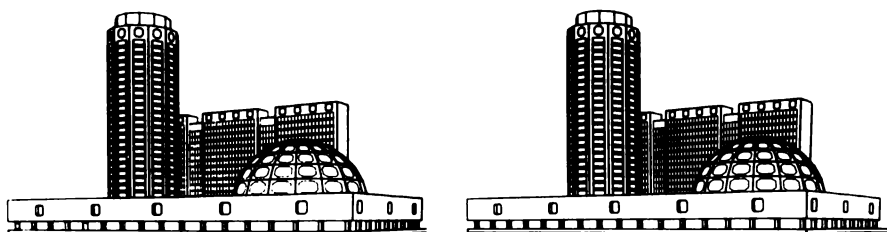


Рис. 122. Пример машинной стереопары

точными, и вручную создавать их весьма сложно. На рис. 122 показана машинная стереопара, где разница между проекциями незначительна.

Более сложные системы стереоскопии используют не две различные проекции, а целую серию их или даже подобие "непрерывной" проекции, построенной для меняющейся точки зрения и "разложенной" на узкие полосы, просматриваемые через специальный растровый экран.

8.4. Контроль видимости элементов изображения

Задача контроля видимости или, как говорят, устранения невидимых линий (и участков поверхностей для полутонсовых изображений) оказалась для машинного решения довольно сложной и требующей повышенного расхода машинного времени и оперативной памяти. Основная причина состоит в том, что в общем случае каждый из элементов смоделированного в трехмерном пространстве объекта может быть закрыт, при наблюдении с той или иной точки зрения, любым другим элементом. Поэтому при увеличении количества элементов число их взаимных сопоставлений увеличивается в квадрате. Для полутоновой растровой графики вопрос контроля видимости иногда решается проще, чем для векторной (линейчатой) графики. Последняя пока более распространена, поэтому коснемся алгоритмов, применимых для нее [6].

Линии, изображающие объемные предметы и пространственные сцены, могут иметь различный смысл. Во-первых, это могут быть тонкие нити, стержни, которые загораживают лежащие за ними части объекта лишь в пределах собственной толщины. Во-вторых, линии могут изображать рисунок, нанесенный на какую-нибудь плоскость, поверхность. Сам по себе такой рисунок тоже ничего не загораживает. В-третьих, это могут быть края, граничные линии участков поверхностей, граней, которыми часть объекта заканчивается или по которым одна грань (поверхность) сопрягается с другой. Наблюдая объект, можно видеть и такие очерковые, или как иногда говорят, контурные линии, которые для нашего глаза представляют видимую границу объекта, но на самом объекте их как таковых нет. Причины, по которым какая-либо линия может быть на проекции частично или полностью невидимой, могут быть двоякого рода: либо носитель линии (поверхность, грань, сплошное тело) повернут к наблюдателю "обратной" стороной, либо линия заслоняется более близкими к зрителю частями объекта. В связи с этим программы контроля видимости могут опираться на два критерия: критерий ориентации и критерий экранирования.

Про критерий ориентации упоминалось в разд. 6.3, где речь шла о подпрограммах черчения типовых тел — таких как параллелепипеды, призмы, конусы. Составляя новые подпрограммы или программы черчения нестандартных поверхностей и тел, пользователь может сам организовать проверку видимости по этому критерию.

Пусть объект является выпуклым многогранником или выпуклой поверхностью, которая может быть представлена набором мелких граней. Если многогранник — сплошное тело, то ясно, что любая его грань может быть видима только с одной — наружной стороны. Пусть и кусок поверхности обладает таким же свойством. Вопрос: как аналитически проверить, с какой стороны видна грань (часть поверхности) на проекции? Один из приемов такой. Нанесем на грань три пронумерованные точки, образующие треугольник — такой, чтобы при взгляде на эту грань

с внешней стороны треугольник имел положительную ориентацию (направление обхода его вершин было бы положительным). Тогда на проекции признаком того, что грань видна с этой же наружной стороны, будет положительная ориентация проекции этого треугольника. Если известны координаты точек в первичной пространственной КС, очевидно, можно найти координаты их образов в системе чертежа. Можно воспользоваться простой формулой, определяющей ориентированную площадь треугольника или сходной формулой, подсчитывающей величину векторного произведения двух сторон треугольника, принятых за векторы:

$$C = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1).$$

Знак величины C указывает, с какой стороны мы видим грань. Для правильной работы этого способа используемые преобразования координат должны сохранять ориентацию пространственных осей или по крайней мере пользователь должен знать, не меняется ли ориентация.

Другой прием заключается в том (см. рис. 123), что в какой-либо точке исследуемой грани или в точке поверхности строят нормальный к ним вектор, направленный "наружу", и определяют величину его проекции на луч зрения, направленный в его основание. Если знак проекции отрицательный — конец вектора ближе к наблюдателю, чем начало, а грань видна с наружной стороны, и наоборот. При использовании этого приема преобразования координат должны быть изометрическими (в расширенном смысле).

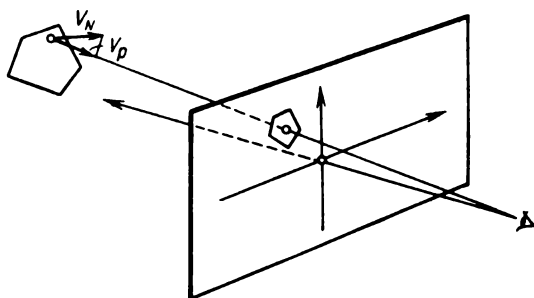


Рис. 123. Определение ориентации грани с помощью нормального вектора: V_N — вектор; V_P — проекция вектора на луч зрения

В программной системе для таких операций могут быть использованы специальные модули. Так, для определения величины проекции вектора в системе Алграф можно использовать оператор

CALL ORVEC1 (X, Y, Z, DX, DY, DZ, C)

где координаты начала вектора и его компоненты задаются в ЧКС, а выходной параметр C — величина проекции вектора на луч зрения.

Используя критерий ориентации, можно экономить машинное время, "обходя" проработку и графический вывод целых фрагментов объекта, например множества окон на целом фасаде. Однако этот метод не позволяет полностью устранять невидимые линии, необходимо также использовать контроль по экранированию. Как осуществляется этот контроль?

На первом этапе работы на изображаемом объекте надо найти все части, которые могут заслонить другие. (Это могут быть криволинейные поверхности, их отсеки и др.). Для упрощения алгоритма примем, что все кривые поверхности представлены наборами плоских граней (сравните с тем, как разные кривые линии аппроксимируются ломаными). Состыкованные друг с другом грани могут ограничивать сплошные (или даже пустые внутри) тела либо быть тонкими фигурными пластинами, так или иначе расположенными в пространстве. Граница грани, в свою очередь, — ломаная линия. Эти ломаные на самом деле плоские, но мы будем считать их заданными в пространстве. На первом этапе работы алгоритма надо в каком-либо массиве запомнить контуры всех граней (можно использовать пространственную ПРКС или систему чертежа, но дополнительно с указанием расстояния от каждой точки до наблюдателя (глубины)). Можно запоминать не все грани, а, пользуясь критерием ориентации, только те, которые на проекции обращены к наблюдателю "лицом". На втором этапе начинается черчение различных линий, в том числе и контуров граней

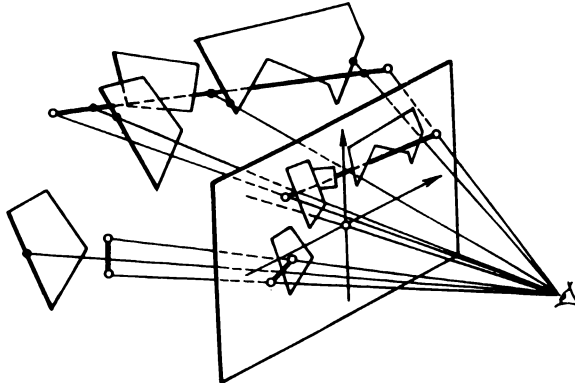


Рис. 124. Схема, показывающая экранирование отрезков в пространстве

самих экранов. Каждый отрезок предварительно переводится в КСЧ, т.е. определяется его проекция. Далее проверяется, не попал ли этот отрезок или его часть в область, занятую каким-либо экраном. Если это так, то проверяется, расположен ли отрезок ближе к наблюдателю, чем экран, или наоборот, отрезок дальше экрана и, частично либо полностью, гранью заслоняется. Как видим на рис. 124, отрезок может быть заслонен несколькими гранями, невыпуклая грань может одной своей частью заслонять отрезок, тогда как другая ее часть будет расположена за отрезком, и т.д.

Проанализировав все экраны, программа может набрать некоторое количество точек, соответствующих пересечениям проекции отрезка с контурами различных граней. Эти точки необходимо рассортировать, расположив в том относительном порядке, в каком они расположены по ходу отрезка. Таким образом отрезок оказывается поделенным этими точками на несколько частей, среди которых могут быть видимые и невидимые. Для каждой части программа должна определить "индекс" ее видимости и в зависимости от этого вычерчивать ее либо пропускать. Как показано на схеме (рис. 125, а), для одного экрана условие "попадания" точки

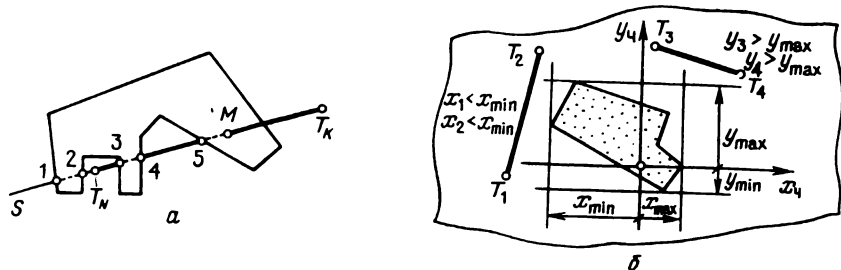


Рис. 125. а – Определение принадлежности точки области, ограниченной замкнутым контуром; б – принцип, позволяющий быстро решать вопрос экранирования для некоторых отрезков: T_N, T_K – начальная и конечная точки вычерчиваемого отрезка, M – точка, для которой исследуется экранирование, S – "обратный луч" отрезка

внутри его области определяется число пересечений луча, проведенного из точки наружу. Если это число нечетное, считается, что точка "попала внутрь". За такой луч в программе контроля видимости может быть взят луч, принадлежащий прямой, на которой лежит вычерчиваемый отрезок. Этот луч направлен от исследуемой точки против направления вычерчивания отрезка. Итак, если говорить об одном экране, точки пересечения его контура с проекцией отрезка делят отрезок поочередно на видимые и невидимые части (рассматривается случай, когда в пространстве отрезок расположен дальше от наблюдателя, чем экран). Если же проекция пересекается с несколькими экранами – алгоритм соответственно усложняется.

Из описания видно, что это алгоритм достаточно общего характера, но в то же время он требует значительных затрат машинного времени. Для ускорения его работы в системе Алграф и других системах предусмотрены дополнительные приемы, например предварительный анализ относительного положения отрезка и экрана согласно рис. 125, б.

В подсистему экранирования могут входить специальные подпрограммы формирования экранов для типовых фигур и другие средства. Так, в системе Алграф подпрограммы образования экранов для таких тел, как параллелепипеды, призмы, конусы, пирамиды, вызываются почти так же, как подпрограммы черчения этих тел, рассмотренные выше. Лишь имена подпрограмм изменяются (вместо "PRP" вызывается "ЕКPRPA", вместо "PRIZM" – "ЕКPRIZ", вместо "KONUS2" – "ЕКONS2"; правда, в

последнем случае первым параметром является индекс модификации, расширяющий возможности подпрограммы). Причем систему можно настроить так, что параллельно с запоминанием моделей экранов в другом массиве будут запоминаться ребра, очерковые линии этих тел. Чтобы после формирования всех экранов такие запоминавшиеся линии вычертить, достаточно вызвать одну подпрограмму: CALL PLADOP. Этот вызов

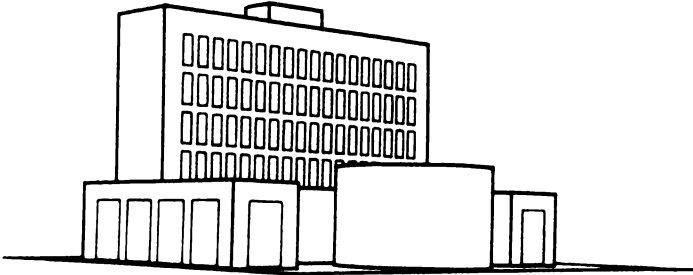


Рис. 126. Машинное изображение с устраниением невидимых линий

не мешает дальше, с контролем по накопленным экранам, чертить другие линии — различные стержни, проемы, рисунки окон и т.д.

В начале книги проводились изображения архитектурных и технических объектов, выполненные этим способом. На рис. 126 приведем изображение попроще, но сопроводим его текстом программы на фортране:

```

COMMON/BENKOT/XOR, YOR, ZOR, XYZ(50), IND(40)
*/ZAPKOT/IAMAX, IATEK, AAAA(2000)/ADP/IADP, IADM, ADP(2000)
DIMENSION A(20), B(20), C(10)
CALL SBROS
CALL VISIN6(0, -1, 1, 2000, 2000)
CALL PERSP(0., -10., 0., 0., 40., 0., 1., 300)
IND(6) = 2
IND(4) = 1
CALL EKPRIZ(10, A, 8)
DATA A/0., 20., 0., 0., 20., 0., 30., 30., 90., 30., 100., 0.,
*120., 0., 120., 60., 0., 60./
CALL EKPRPA(10, 10., 40., 20.05, 100., 20., 45.)
CALL EKPRPA(10, 50., 45., 65.1, 20., 10., 6.)
CALL EKONS2(0, 60., 0., 0., 20., 20., 25.)
CALL PLADOP
CALL NPERA(1)
IND(4) = 0
CALL ORVEC1(0., 0., 0., 0., 0., -1., V)
IF(V.GT.0.)GOTO 1
ZOR = -0.05
CALL PRM(5., 0., 10., 15.)

```



```

CALL PRM(105., 0., 10., 15.)
1 CALL ORVEC1(0., 0., 40., 0., 0., -1., V)
  IF(V.GT.0.)GOTO2
  ZOR = 39.95
  CALL SLIN20(B, 6)
  DATA B/16., 20.1, 3., 8., 5., 10., 18., 4./
2 CALL ORVEC1(0., 0., 60., 0., 0., 1., V)
  IF(V.GT.0.)GOTO3
  ZOR = 60.1
  CALL SLIN20(B, 6)
3 IND(4) = -1
  CALL ORVEC1(0., 0., 0., 0., 0., -1., V)
  IF(V.GT.0.)GOTO4
  ZOR = -0.05
  CALL SLIN20(C, 4)
  DATA C/5., 0.1, 10., 15., 13., 5./
4 CALL ORVEC1(0., 0., 120., 0., 0., 1., V)
  IF(V.GT.0.)GOTO5
  ZOR = 120.05
  CALL SLIN20(C, 4)
5 IND(4) = 1
  ZOR = -0.05
  CALL PRM(-10., -30., 140., 100.)
  CALL PRM(-15., -35., 150., 110.)
  CALL STP
  END

```

Здесь модели экранов запоминаются в массиве AAAA блока ZAPKOT, ребра типовых тел для последующего черчения – в массиве ADP блока ADOP, модуль VIZIN6 использован для инициализации настройки на нужный режим подсистемы экранирования. После образования экранов и вызова подпрограммы PLADOP вычерчиваются окна; при этом для ускорения работы программы используется проверка для фрагментов изображения по критерию ориентации.

В частных случаях алгоритм экранирования может быть более простым. В ряду зарубежных систем и в отечественном пакете программ Графор используется сравнительно простой и быстро работающий алгоритм, пригодный для изображения однозначных (относительно координат-аргументов) поверхностей, вычерчиваемых в виде семейства сечений или сетки. Поверхность в этом случае чертится начиная с более близких к наблюдателю частей, и уже вычерченные части образуют один, но постепенно наращиваемый, постепенно увеличивающийся в процессе работы программы экран, который может перекрывать линии, вычерчиваемые далее.

Поверхность в трехмерном пространстве выражается одним явным или неявным уравнением: $z = f(x, y)$; $\Phi(x, y, z) = 0$. Если, в первом случае, поверхность однозначная относительно координат-аргументов, ее можно считать геометрическим выражением функции двух переменных. Можно описать поверхность и тремя параметрическими уравнениями (как и пространственную кривую линию). Однако каждое из этих уравнений или хотя бы одно из них должно содержать два производящих параметра:

$$x = f(t, v), \quad y = \varphi(t, v), \quad z = \psi(t, v).$$

Этот факт соответствует, как известно, тому, что поверхность содержит двумерное множество точек. Параметры t и v можно принять за своеобразные координаты (криволинейные), определяющие положение точки на поверхности; линии, получающиеся при фиксации одного параметра и изменении другого, т.е. линии $t = \text{const}$ и $v = \text{const}$, обычно называют *параметрическими*. Самый распространенный пример – линии параллелей и меридианов на поверхности глобуса (напишите три параметрических уравнения сферы, взяв в качестве параметров угол долготы и широты). Однопараметрическое семейство линий в пространстве образует поверхность. "Сплошная" поверхность может получиться, только если этих линий неограниченно много. Представьте себе образование поверхности на трехкоординатном фрезерном станке: фреза движется последовательными проходами вдоль заданных линий, расстояние между этими линиями порядка нескольких миллиметров, но – за счет вращения и собственной формы фрезы – получается "сплошная" поверхность, хотя немного "ребристая". С помощью программы можно образовать счетное количество линий; если их достаточно, вполне можно судить о форме и размерах поверхности. Поверхность можно изобразить и в виде сетки пересекающихся линий. Это могут быть продольные и поперечные сечения, два семейства параметрических линий и др. Как и для линий, для поверхностей можно создавать численные модели – аналитические и координатные.

9.1. Семейство линий в пространстве, образующее поверхность

Выше мы уже образовывали семейства линий на плоскости. Независимо от того, плоские ли сами линии или пространственные — если их разместить в пространстве, они представляют собой поверхность. Иногда удобно и естественно полагать, что неизменная либо постепенно деформирующаяся линия движется в пространстве, оставляя за собой "след"

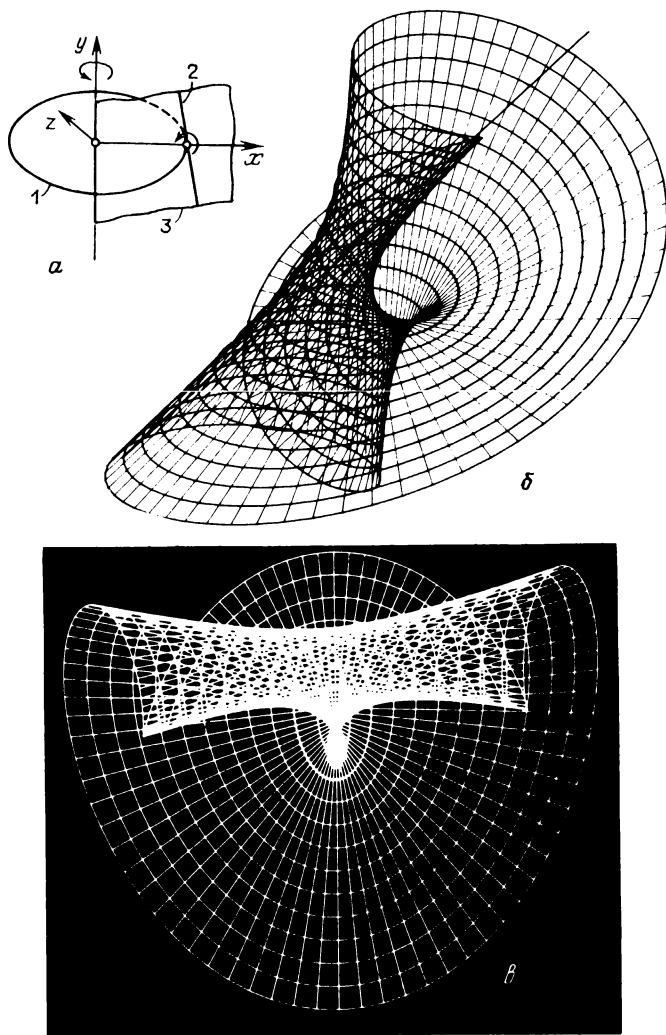


Рис. 127. Односторонняя поверхность, образованная семейством отрезков: *a* — схема образования (*1* — направляющая окружность, *2* — образующая, *3* — радиальная плоскость); *б, в* — машинные изображения

в виде поверхности. Это так называемые кинематические способы образования поверхностей.

Образование поверхности программным путем с выводом ее изображения сводится, очевидно, к вычерчиванию в цикле последовательности постепенно изменяющихся в пространстве линий. Приведем пример. В качестве *образующей* линии возьмем отрезок, который будет скользить своей средней точкой по неподвижной *направляющей* — окружности (рис. 127, *а*), оставаясь к ней перпендикулярным (в точке пересечения). Кроме того, пусть отрезок поворачивается вокруг касательной к окружности, построенной в точке пересечения. Допустим, за один обход окружности отрезок повернется на пол-оборота. Тогда его конечное положение совпадет с начальным, только верхний конец окажется внизу, а нижний наверху, т.е. отрезок опишет одностороннюю поверхность — знаменитый лист Мёбиуса. Если отрезок взять подлиннее — поверхность будет самопересекающейся. Вместо отрезка можно представить себе и неограниченную прямую.

Чтобы составить простую программу вычерчивания (в какой-то проекции) множества таких отрезков, движение отрезка по окружности будем моделировать с помощью преобразований координат; отрезок будет располагаться в плоскости $z = 0$ частной координатной системы, а вся система будет поворачиваться вокруг своей вертикальной оси (y); в программе это реализуется включением в цикл оператора поворота. Поворот самого отрезка в плоскости $z = 0$ осуществим с помощью тригонометрических функций. Ниже приведено начало программы, а на рис. 127, *б* показано машинное изображение поверхности; кроме семейства отрезков вычерчено также семейство кривых, описываемых различными точками отрезка, так что поверхность изображена в виде сети. Попробуйте дописать окончание программы

```
COMMON/BENKOT/XOR, YOR, ZOR, XYZ(50), IND(40)
CALL SBROS
CALL AKS (0., 0., 0., 0., 30., 35., 1.)
IND(6) = 1
IND(4) = 1
ZOR = 0.
CALL OK (0., 0., 30.)
IND(4) = 0
IND(5) = 2
CALL MATT(11)
DO 1 K = 1,72
A = K * 5.
CALL POV3 (0., A, 0.)
X = 90. * COSG (A/2.)
Y = 90. * SING (A/2.)
1 CALL OTREZP (X + 30., Y, 0., 30. - X, -Y, 0.)
```

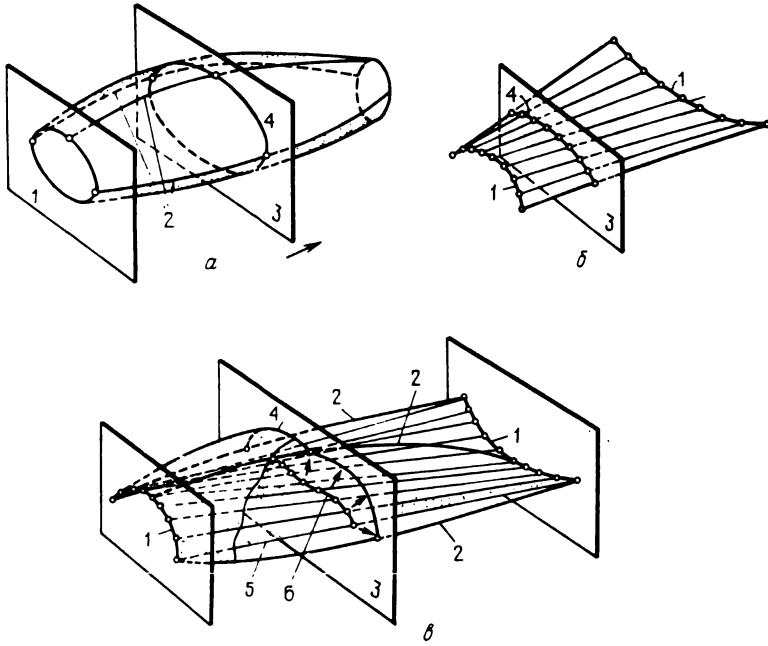


Рис. 128. Схемы образования поверхностей:
a – деформацией исходного сечения; *б* – с помощью двух исходных сечений (линейчатая поверхность); *в* – с помощью двух исходных сечений и кривых направляющих (*1* – исходные сечения, *2* – направляющие, *3* – плоскость текущего (промежуточного) сечения, *4* – промежуточное сечение, *5* – вспомогательная линейчатая поверхность, *6* – предварительно образуемое промежуточное сечение)

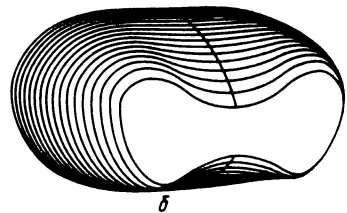
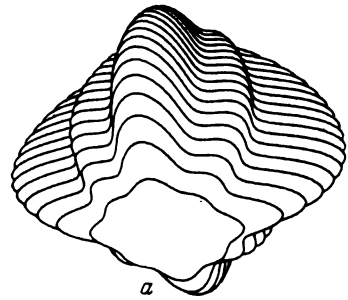


Рис. 129. Построение поверхностей как семейств сечений

Функции SING и COSG вычисляют здесь синус и косинус при задании угла в градусах, радианах или других единицах в зависимости от настройки системы.

В технике и художественном конструировании поверхности часто синтезируются как семейства плоских сечений либо пространственных образующих. При ручном построении таких семейств – в автомобилестроении, авиационной и других областях – давно уже применяются так называемые *плазово-ключевые* способы. В машинном исполнении некоторые разновидности таких поверхностей образуются с помощью аффинной или какой-либо другой деформации исходного сечения (или исходной образующей), которое движется в пространстве, "скользя" несколькими точками по заданным кривым-направляющим (рис. 128, а). В качестве исходных данных для построения куска поверхности могут быть заданы два граничных сечения разной конфигурации (рис. 128, б). Множество "промежуточных" сечений может быть построено примерно так, как в двумерной геометрии мы строили пропорциональные кривые. При этом образуется линейчатая поверхность. Чтобы она стала "выпуклой" и легла на заданные направляющие (рис. 128, в), предварительно построенные в памяти ЭВМ пропорциональные сечения деформируются с помощью аффинных или других преобразований. На рис. 129, б приводится машинное изображение такой поверхности.

9.2. Функция двух переменных, описывающая поверхность

В физике, химии, экономике, автоматизированном проектировании и других областях для наглядного изображения функций двух и даже нескольких переменных вычерчивают поверхности. Этот способ применим и для "конструктивных" целей – для моделирования поверхностей деталей, архитектурных сооружений, рельефа местности и т.д. Если поверхность описана явным уравнением, по нему, задаваясь парами значений двух координат-аргументов, нетрудно получить значения третьей координаты. Тогда в плоскости координат-аргументов можно наметить квадратную или прямоугольную сетку и подсчитать значения функции для ее узлов (рис. 130, а). В итоге поверхность будет представлена своей координатной моделью. По полученной модели можно построить семейства линий – сечений поверхности плоскостями $x = \text{const}$ и $y = \text{const}$. Обычно эти линии строятся как ломаные, а поверхность получается составленной из четырехугольных ячеек. В пределах каждой ячейки предполагается, что точная форма для нас несущественна и исходную поверхность можно заменить либо на два треугольника, либо на простую билинейную поверхность – поверхность гиперболического параболоида $z = axu + bx + cy + d$. В нужных случаях через точки координатной модели поверхности можно проводить и составные кривые линии сечений, например линии типа сплайн. Координатная модель поверхности может исполь-

зоваться не только для ее вычерчивания, но и для различных других целей – для геометрического конструирования, расчетов на прочность, подсчета геометрических характеристик и т.д. Можно, например, подсчитать площадь куска поверхности, объем части пространства под ней. Как хранить координатную модель поверхности в машине? Наиболее естественный способ – тройками координат ее точек, упорядоченными по линиям "продольных" или "поперечных" сечений. Приведем один из способов задания поверхностей. В двумерном массиве с именем, например, ZMAS,

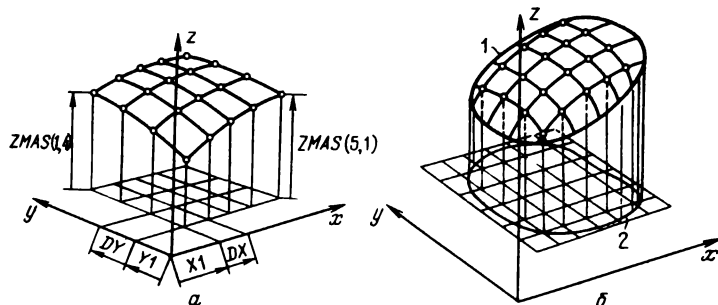


Рис. 130. Поверхность, заданная координатной моделью: а отсек, проецирующийся на плоскость координат-аргументов в прямоугольник; б отсек, ограниченный кривой линией

имеющем NX строк и NY столбцов, будем хранить только значения координат-функций z_{ij} . Координаты-аргументы для точки с индексами i, j мы всегда можем вычислить, зная параметры прямоугольной сетки, в узлах которой генерировалась модель. Если i – номер "поперечной" координатной линии ($x = \text{const}$), а j – номер "продольной" линии, то $x_{ij} = x_1 + (i - 1)d_x$, $y_{ij} = y_1 + (j - 1)d_y$, где x_1, y_1 – значения координат-аргументов для одного из углов сетки, d_x, d_y – величины шагов (взятые с учетом знаков). Таким образом, вся координатная модель представляет из себя следующую структуру данных: ZMAS, NX, NY, X1, Y1, DX, DY. Этим способом можно моделировать куски поверхностей, проецирующихся на плоскость координат-аргументов, в прямоугольники. А если нужный кусок имеет другую форму? Условимся поступать так. Опишем вокруг проекции куска прямоугольник и в нем построим опорную сетку для КМД (рис. 130, б). Однако для точек, которые лежат вне заданного куска, в массив ZMAS будем записывать числа-коды, равные 10000. или больше по величине.

Независимо от того, с помощью каких уравнений создавалась КМД описанного вида, или даже в случае если ее точки были замерены на исходном объекте, поверхность можно вычертить в заданной проекции с помощью стандартной подпрограммы, вызвав ее, например в системе Алграф так:

CALL SURF1 (ZMAS, NX, NY, X1, Y1, DX, DY, JS, IM)

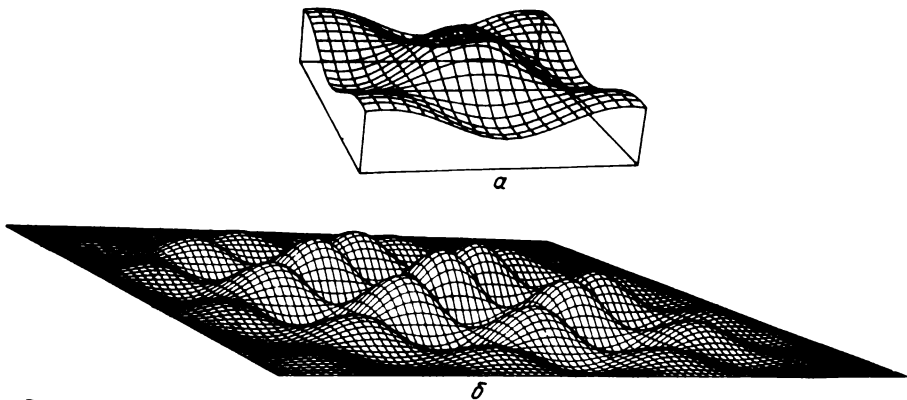


Рис. 131. Волнообразные поверхности, полученные как функции двух переменных

Первые параметры выше пояснялись, параметр JS позволяет чертить не все линии сечений, определяемые моделью, а, например, каждую вторую или третью. Стороны получившихся более крупных ячеек поверхности будут тогда уже не отрезками, а ломаными, аппроксимирующими дуги кривых. Индекс модификации IM позволяет вычерчивать линии "поперечных" или "продольных" сечений либо оба семейства их; помимо этого, можно подключать контроль видимости элементов поверхности по критерию их ориентации.

Ниже для наглядности приводится программа, где сначала синтезируется КМД поверхности волнообразной конфигурации, а затем расчерчиваются ее сечения (без какого-либо контроля):

```

COMMON/BENKOT/XOR, YOR, ZOR, XYZ(50), IND(40)
DIMENSION ZMAS(41, 41)
CALL SBROS
READ 90, X1, Y1, DX, DY, A, B, C, BET, GAM, AM
90 FORMAT (10F8.0)
DO 1 K = 1,41
DO 1 M = 1,41
X = X1 + (M - 1) * DX
Y = Y1 + (K - 1) * DY
1 ZMAS(M, K) = A * COS(X * B) * COS(Y * C)
CALL AKS(0., 0., 0., 0., BET, GAM, AM)
IND(6) = 1
IND(4) = 1
DO 2 K = 1,41
Y = Y1 + (K - 1) * DY
DO 3 M = 1,41

```



```

CALL LINP (X1 + (M - 1) * DX, Y, ZMAS (M, K))
3 CALL PERO
2 CALL VP
...
CALL STP
END

```

Желающие могут попробовать восстановить пропущенную часть программы. На рис. 131 показаны машинное изображение и изображение поверхности сходного типа, изготовленное с устранением невидимых линий по упрощенному алгоритму с "наращиваемым" экраном.

9.3. Другие способы построения поверхностей и операции с ними

Довольно широко распространены параметрически заданные поверхности. В соответствии с закономерностями их образования графически такие поверхности естественно представить семействами параметрических линий. Если вычертить два семейства параметрических линий, на поверхности образуется сеть. Эта сеть ни на одну из координатных плоскостей не будет проецироваться как прямоугольная (рис. 132, а). На одной поверхности можно ввести параметризацию различным образом. Стараясь подобрать ее так, чтобы нужный отсек поверхности описывался

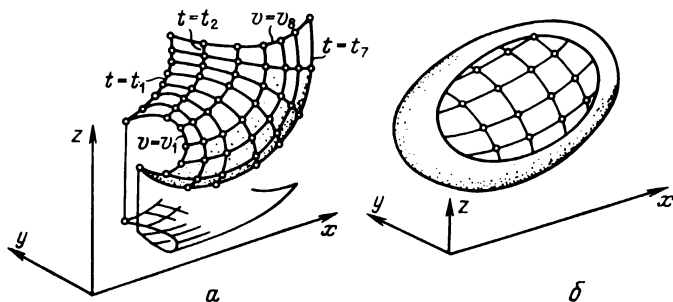


Рис. 132. Изображение параметрически заданной поверхности сетью параметрических линий: а – отсек, соответствующий прямоугольному полю параметров; б – отсек, ограниченный заданной кривой линией

на "прямоугольном поле генерирующих параметров", а это значит, что отсек будет ограничен четырьмя дугами, являющимися частями параметрических линий. Если такую параметризацию подобрать не удастся, поступить можно аналогично тому, как мы пояснили выше (для непараметрических поверхностей). Координатная модель параметрической поверхности может содержать трехмерный массив координат точек: XYZ(3, NT, NY) (записаны предельные значения индексов). Первый индекс указывает номер координаты, второй и третий – номера параметрических линий, пересечение которых дает точку.

В практике машинной геометрии подобран ряд алгоритмов, позволяющих заполнить гладкой поверхностью пространственный криволинейный четырехугольник (рис. 133).

```

COMMON/BENKOT/XYZ (53),IND(40),AAA(84)/ZAPKOT/IAMAX,IATEK,
*AAAA(1000)
DIMENSION IA(4),IB(6),A(6,2,4),B(1000)
DATA A/0.,30.,10.,4.,7.,9.,40.,100.,100.,4.,7.,9.,
*90.,30.,0.,-1.,2.,0.,100.,70.,60.,2.,1.,1.,
*0.,30.,10.,2.,-1.,1.,90.,30.,0.,4.,3.,2.,
*40.,100.,100.,2.,1.,2.,100.,70.,60.,3.,0.,-2./
CALL SBROS1
CALL AKS(0.,0.,0.,0.,-30.,20.,0.8)
IND(6)=1
CALL PRP(0,0.,0.,0.,100.,100.,100.)
IND(8)=3
IAMAX=1000
DO 1 K=1,4
IA(K)=IATEK
CALL POL3P(A(1,1,K),1,14)
CALL ZAPK(0)
1 IND(1)=0
IND(8)=0
IB(1)=1
DO 2 K=1,4
CALL LPLIN3(AAAA
*(IA(K)),B(IB(K)),N,M)
2 IB(K+1)=IB(K)+M+2
DO 3 K=1,2
CALL SUR4P2(B,B(IB(2)),B(IB(3)),B
*(IB(4)),2,20,20,1)
IND(5)=2
3 CALL MATL3(1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.)
CALL STP
END

```

Для некоторых алгоритмов граничные дуги – дуги линий определенного типа (например, полиномов третьей степени): для других могут быть линии разных, произвольных типов, но переведенные в координатные модели – т.е. наборы последовательных точек. Из таких криволинейных ячеек можно составлять более сложные *составные поверхности* аналогично тому, как раньше мы получали составные линии. Осуществить гладкую стыковку смежных кусков здесь, однако, сложнее, чем для линий, и алгоритмы приходится соответственно усложнять.

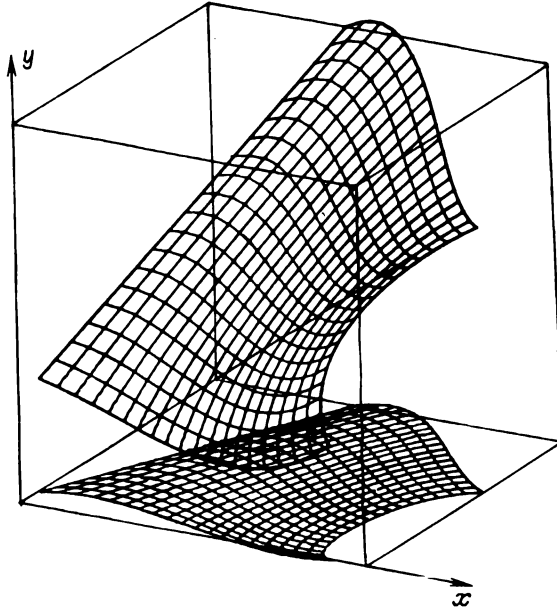


Рис. 133. Формирование отсека параметрической поверхности, затягивающей пространственный криволинейный четырехугольник (показано в аксонометрической проекции, с дополнительным вычерчиванием вторичной проекции)

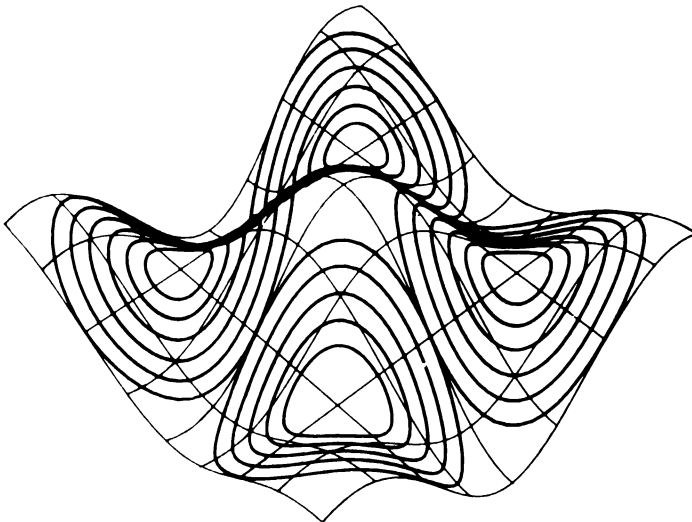


Рис. 134. Рисунок, спроецированный на криволинейную поверхность

При работе с поверхностями приходится решать довольно сложные задачи геометрического конструирования. Не проста задача построения линии пересечения двух сложных поверхностей. Типовые алгоритмы подготовлены для построения линий сечения поверхности (представленной в виде координатной модели) координатными плоскостями. Эти линии $z = z_i = \text{const}$ часто называются *линиями уровня*, или *горизонталями*. Другая задача — спроецировать на поверхность какой-либо рисунок. К примеру, на рис. 134 изображена волнообразная поверхность, на которую нанесены изогнутые фигуры. В проекции на плоскость $z = 0$ это семейства концентрических окружностей!

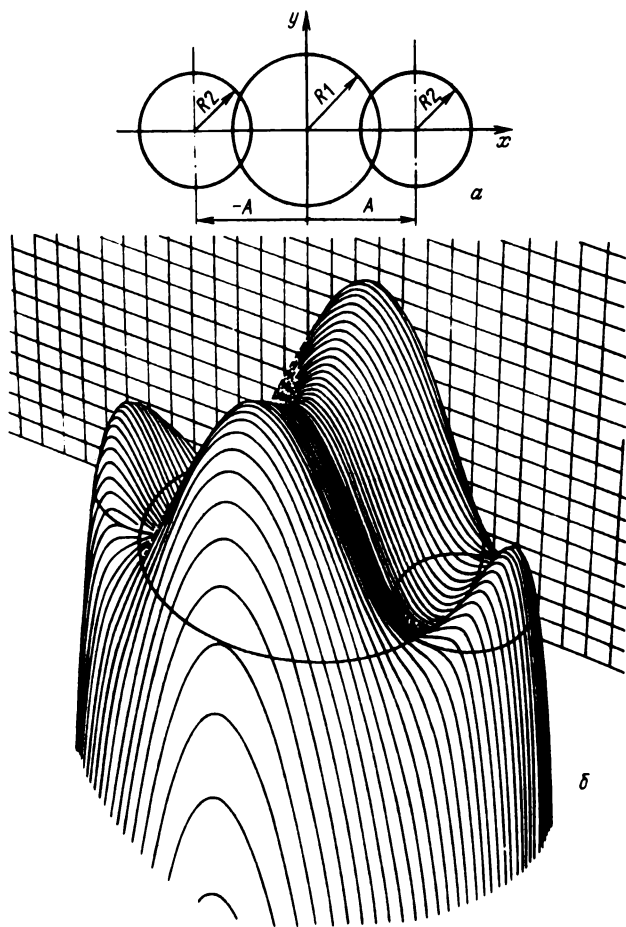


Рис. 135. Образование поверхности из семейства линий влияния: *a* — исходные данные в плоскости $z = 0$; *б* — машинный чертеж

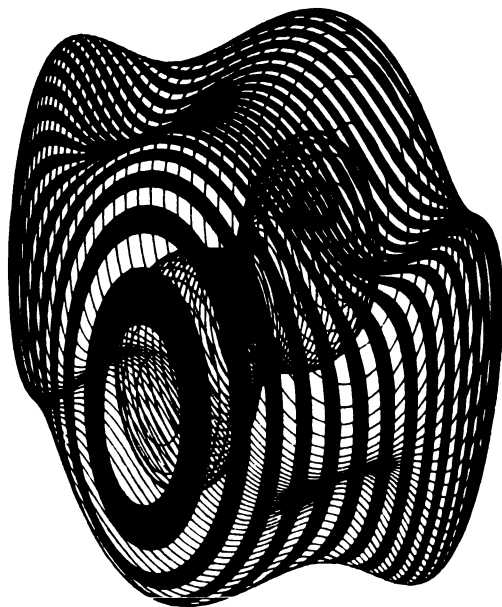


Рис. 136. Второй вариант поверхности из линий влияния

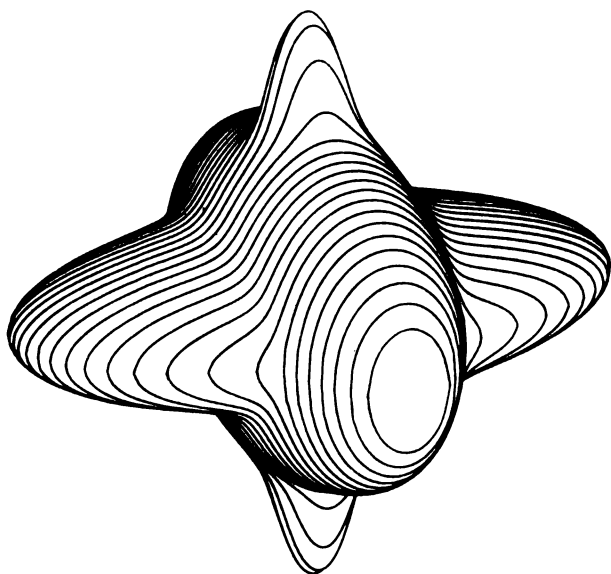


Рис. 137. Изображение "поверхности влияния"

Приведем еще пару примеров аналитического задания поверхностей. Вспомним семейства линий влияния на плоскости (см. рис. 75). Из такого семейства в пространстве можно получить поверхность, если расположить линии в различных плоскостях $z = \text{const}$ в зависимости от значения "потенциала" D . В простейшем случае $z \equiv D$. На рис. 135, б показана поверхность, образованная этим способом. Исходными для ее построения являются три попарно пересекающиеся окружности (рис. 135, а). Сама поверхность изображена не линиями "равного влияния", а линиями сечений плоскостями $y = \text{const}$ (для лучшего выявления ее формы). На рис. 136 показан другой вариант, полученный из тех же исходных элементов, но координата z подсчитывалась как функция от потенциала согласно уравнению $D = B - Z^2$, где B – константа.

Если для исходных линий можно образовать "линии влияния", то для исходных поверхностей в пространстве сходным образом получим "поверхности влияния". Сконструируем для примера сложную поверхность из нескольких более простых. Рассмотрим простую поверхность – эллипсоид. Если его оси параллельны координатным осям, уравнение несложно:

$$(x - x_0)^2/A^2 + (y - y_0)^2/B^2 + (z - z_0)^2/C^2 - 1 = 0,$$

(здесь A, B, C – длины полуосей, x_0, y_0, z_0 – координаты центра). Зададим три таких эллипсоида с общим центром, но разными соотношениями полуосей и перемножим левые части их уравнений. Получим уравнение поверхности шестого порядка, которое будет описывать три эллипсоида, "вместе". Теперь в правой части нуль заменим на константу – "потенциал" D . Это будет уже другая поверхность; если D по абсолютной величине мало, поверхность приближается к исходным трем эллипсоидам. На рис. 137 видим внешнюю часть одного из вариантов. Внутри расположена вторая часть поверхности. Попробуйте представить себе ее форму. Для построения этой поверхности многократно (для каждой расчетной точки) пришлось искать корни уравнения шестой степени. Поверхность такой сложности вручную точно изобразить практически невозможно.

**НЕМНОГО О ГРАФИЧЕСКОМ ДИАЛОГЕ
И ПОЛУТОНОВЫХ ИЗОБРАЖЕНИЯХ**

Хорошее машинное изображение можно сравнить с картиной, рисунком; их последовательность – с иллюстрированным журналом. Множество изображений на экране, быстро сменяющих друг друга или даже образующих единое динамически-изменяемое во времени изображение, напоминает диафильм или кинофильм. Однако когда мы вводим в действие средства диалога, управляющие работой программ и формированием последующих изображений, возникает явление более сложное, чем кинофильм. Это может быть, например, *управляемый кинофильм*. Мы предвидим в следующем веке (а возможно, и раньше) появление даже художественных "гибридов" кинофильмов и машинных программ. Зритель получит возможность влиять на сюжет, игру актеров. Уже сейчас появилось много развлекательных и познавательных программ-игр для бытовых (персональных) компьютеров, где смешные человечки или чудовища гоняются друг за другом, летают космические корабли и т.д.

Приведем несколько простых примеров осуществления графического диалога, чтобы раскрыть его принципы.

Выше отмечалось, что программа для вывода изображения на экран терминала строится почти так же, как для графопостроителя. Теперь покажем, как в программе можно предусмотреть управляющее воздействие пользователя. Уже говорилось, что после описания всех графических элементов, которые нужны на экране, надо вызвать подпрограмму, передающую информацию в буферную память терминала и запускающую регенерацию, например, (в системе Алграф) – с помощью оператора

CALL EKRAN (IZA)

Изображение после этого "светится", а программа "ждет". Для ее продолжения нужно нажать одну из кнопок функциональной клавиатуры, передающей в ЭВМ сигнал прерывания. Изображение на экране погаснет (впрочем, можно и оставить изображение), программа будет работать дальше, а номер нажатой кнопки будет передан программе через перемен-

ную, указанную при вызове подпрограммы. В последующих операторах программы можно предусмотреть анализ значения этой переменной, условные переходы и т.д. Приведем пример программы, где изображение (заклепки) сначала генерируется с исходными размерами согласно схеме на рис. 138, а, а при нажатии кнопок от № 2 до № 7 происходит перезадавание изображения с увеличением или уменьшением одного из размеров. Два варианта машинных изображений показаны на рис. 138, б. Нажатие

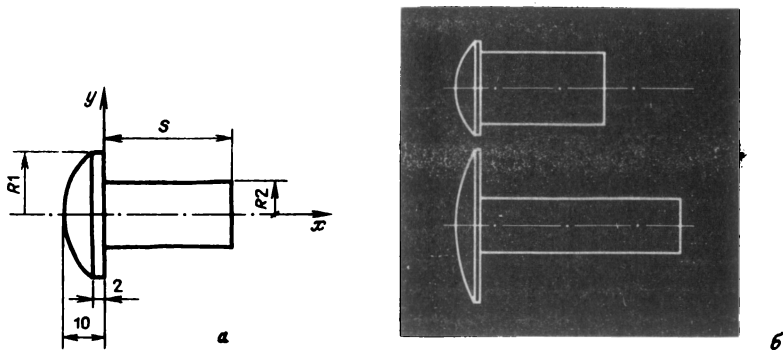


Рис. 138. Схема построения изображения детали (а) и варианты изображений на экране терминала (б)

кнопки № 1 завершает работу программы. Вот текст программы:

```

DATA R/10./, S/40./, R2/15./, DR/5./, DS/5./
CALL SBROSE (-100., -100., 100., 100., 6000)
С КОДИРОВКА ФИГУРЫ
10 CALL PRM (-2., -R2, 2., R2 * 2.)
CALL OKT (-2., -R2, -10., 0., -2., R2)
CALL VP
CALL LMNE (2, 0., -R, S, -R, S, R, 0., R)
CALL PUNC (6, 8., 2., 2., 2.)
CALL OTREZ (-12., 0., 80., 0.)
CALL PUNC (0)
С ВЫВОД НА ЭКРАН И ОЖИДАНИЕ
CALL EKRAN(K)
С ПРОВЕРКА НОМЕРА КНОПКИ И ПЕРЕХОДЫ
KK = K/2
KKK = 1 - (K - KK * 2) * 2
GOTO (1, 2, 3, 4), KK
С КОНЕЦ РАБОТЫ
1 STOP 1

```


С ИЗМЕНЕНИЯ РАЗМЕРОВ

```
2 R = R + DR * KKK  
GOTO 10  
3 S = S + DS * KKK  
GOTO 10  
4 R2 = R2 + DR * KKK  
GOTO 10  
END
```

Проверьте, правильно ли мы разделили номер кнопки на две "части".

Более сложный и разнообразный диалог можно организовать с привлечением алфавитно-цифровой клавиатуры. Для последующего ввода текста с клавиатуры терминала ЕС-7064 (или ЕС-7905) предварительно на экран с помощью стандартной подпрограммы выводится некоторый условный текст. Он может содержать какие-либо вопросы пользователю, а может состоять из символов или пробелов. С помощью другой подпрограммы в нужную позицию текста вводится еще метка — курсор. Тогда с алфавитно-цифровой клавиатуры пользователь может редактировать текст; курсор перемещается в нужные места специальными кнопками, вновь введенные символы заменяют "старые" символы или пробелы. Все эти замены фиксируются в буферной памяти терминала. Когда текст отредактирован, нажимается одна из кнопок прерывания и программа продолжает работу. С помощью подпрограммы базового обеспечения надо из буферной памяти прочесть в указанный в программе массив текстовую информацию. Так как эта информация поступает в символьном виде и ее требуется расшифровать, введенные числа необходимо перевести во внутримашинную форму. После этого введенные числа и команды можно использовать в программе по своему усмотрению. Примера такой программы мы приводить не будем, а заметим, что в принципе таким методом организована работа с терминалом и языком алграф-Р. После запуска обрабатывающей программы-монитора на экране появляются три пустые строки и метка курсора. Пользователь набирает, полностью или частично, текст будущей картинки на языке алграф-Р. Если для этого трех строк не хватит, нажатие функциональной кнопки № 8 сдвигает текст на одну строку вверх (верхняя строка сохраняется в памяти ЭВМ), снизу появляется новая пустая строка. Набрав весь текст или нужную часть его, человек нажимает функциональную кнопку № 5. Это заставляет программу прочесть весь текст, оттранслировать его и отработать, выведя на экран соответствующее изображение. Дальше перед пользователем открывается несколько возможностей: с помощью одной из функциональных кнопок он может закончить программу, с помощью другой — погасить изображение и вернуться к пустым строкам, чтобы формировать новую картинку, с помощью третьей кнопки можно, временно оставив изображение, повторно вывести ранее набранный текст для его исправления или дополнения.

Еще одна "графическая" операция – указание на экране положения точки или группы точек, для чего используется метка – "маркер", перемещаемый по экрану "световым пером". Вывести маркер в нужном месте экрана можно оператором

CALL BGTRK (NGD, X, Y)

(здесь NGD – программное имя набора графических данных, X и Y – координаты центра маркера). Затем программа переводится в состояние ожидания, а пером можно переместить маркер в нужное место. При продолжении работы программы (с гашением или без гашения картинки) координаты нового положения маркера можно определить, применив оператор

CALL RDTRK (NGD, X, Y)

После этого изображение можно обновить, пополнить новыми элементами, сгенерированными с помощью полученных координат. Световым пером можно также указывать элементы уже светящегося изображения. Для этого перо наводят на элемент, и он начинает "мигать". В этот момент надо нажать на корпус пера, прижимая его конец к стеклу экрана; срабатывает микровыключатель, и в машину подается сигнал прерывания. Переменная, передающая программе пользователя номер нажатой кнопки, будет равна 34. С помощью дополнительных подпрограмм можно определить, во-первых, координаты конца отрезка, который "мигал", а во-вторых – условную метку или номер графического элемента либо

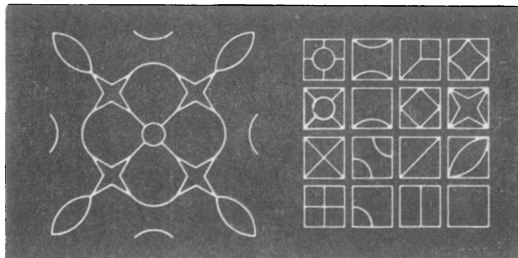


Рис. 139. Формирование на экране орнамента из типовых ячеек

фрагмента изображения, если при генерации они были помечены. С помощью этого способа можно "стереть" ненужный элемент изображения. Есть терминалы, где подобные действия могут осуществляться без обращения к центральной ЭВМ и программе. Указание частей изображений "пером" используется для обеспечения работы графических клавиатур высвеченных на экране.

На рис. 139 приведен снимок с экрана, иллюстрирующий действие одной из учебных программ. В правой части экрана – табличка из типовых

фигурок, в центральной части в начальный момент действия программы -- маленькие фигурки или точки, помечающие узлы квадратной сетки. Задача -- набрать из фигурок достаточно красивый орнамент. Для этого световым пером сначала указывают понравившуюся фигурку в табличке (она помечается наклонным штрихом). Затем в выбранной ячейке центрального поля пером указывают два диагонально расположенных угла (помеченные точками или фигурками). После этого в ячейке вспыхивает выбранная фигурка. В зависимости от того, какой из углов ячейки был указан первым, фигурка может быть расположена в одной из

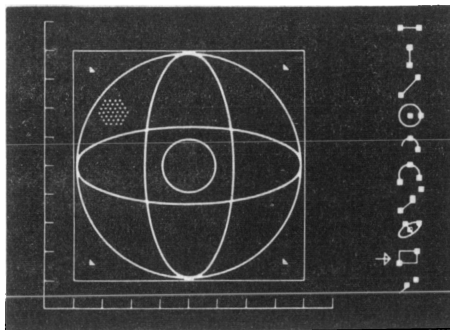


Рис. 140. Диалоговое задание геометрических фигур

четырех возможных ориентаций (без поворота или с поворотом на 90° , 180° или 270°). Аналогично заполняют и другие клетки поля. Неудачно введенную фигурку или все фигурки можно "стереть", нажав соответствующую функциональную кнопку. Когда рисунок закончен, опорные точки или фигурки, помечающие узлы сетки, также можно "стереть". Еще одна кнопка предназначена для вывода перфоленты, по которой копия рисунка может быть вычерчена на построителе.

На рис. 140 указан пример действия другой программы. Здесь изображение формируется из нескольких типов геометрических фигур, которые условно изображены в правой части экрана (эти фигуры можно менять, как бы "листая" страницы каталога). Выбранную фигуру отмечают пером и около нее появляется индицирующая стрелочка. Теперь с помощью маркера и пера в средней части экрана надо набрать нужное для данной фигуры количество опорных точек. Допустим, для генерации окружности надо указать ее центр и какую-либо точку окружности. Когда группа точек набрана, нажатие соответствующей функциональной кнопки приводит к порождению фигурки на экране -- в нужном месте, с нужными размерами. Другие функциональные кнопки позволяют стирать "плохие" фигуры, стирать все изображение, изменять его масштаб и т.д. После задания маркером каждой точки координаты ее высвечиваются в правой

нижней части экрана. За ними можно следить, а положение только что введенной точки можно подкорректировать, передвигая ее с помощью кнопок "шагами" вправо, влево, вверх или вниз.

Для учебных и производственных целей готовятся специализированные "сценарии" и затем уже программы. (Сценарий описывает последовательность действий пользователя и ЭВМ и возможные варианты действий.) Программы могут быть очень сложными, но диалог, по готовым программам, можно осуществлять простыми для пользователя средствами. Так, на рис. 141 показаны пересекающиеся в пространстве фигуры

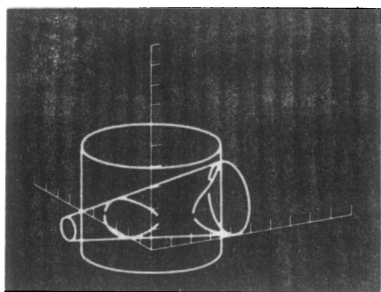


Рис. 141. Иллюстрация диалогового решения задачи на пересечение поверхностей

(два конуса, два цилиндра, конус и эллипсоид и др.). Для задания двух таких тел в ответ на два высвечиваемых программой запроса надо ввести несколько чисел — аналитические модели тел. Программа строит линию пересечения поверхностей. С помощью функциональных кнопок можно изменять ракурс и масштаб изображения на экране, тем самым просмотреть линию пересечения с разных сторон. Изменяя числовые параметры одного или другого тела и повторяя процесс, можно проследить, как при этом изменяется конфигурация линии пересечения.

10.1. Графический диалог на микрокомпьютере

"Полуграфический диалог", при котором человек вводит в машину числа, словесные команды, строки текста, а машина в ответ генерирует на экране те или иные изображения, на персональных микрокомпьютерах проходит примерно так же, как на терминалах больших ЭВМ. Но дешевые микрокомпьютеры часто не имеют таких специальных устройств для ввода графической информации или указания ее на экране, как световые перья и планшеты с визирами. В этих случаях подобие графического ввода можно организовать с помощью специальных кнопок клавиатуры, предназначенных для перемещения курсора или маркера, либо буквенно-цифровых кнопок, используемых в функциональном режиме. При однократном нажатии на кнопку маркер в виде крестика, точки и т.п.

перемещается на экране на один шаг в одном из восьми возможных направлений (как перо у графопостроителя с шаговыми двигателями, см. рис. 13). Размер шага может быть минимальным для данного экрана (соответствовать его растровым единицам) либо изменяемым с помощью соответствующих кнопок.

Более дорогие профессиональные персональные машины могут снабжаться и световыми перьями, и планшетами, и наклоняемыми рукоятками, управляющими маркером, к ним могут подключаться миниатюрные и даже большие графопостроители. Оперативность графического диалога на ПЭВМ с растровыми экранами может быть несколько снижена по сравнению с векторными терминалами больших машин потому, что на растровом экране сложнее независимо друг от друга изменять накладываемые фрагменты или "слои" изображения. Не всегда достаточна и разрешающая способность таких экранов. Однако быстрый прогресс техники дает основания рассчитывать на дальнейшее расширение возможностей ПЭВМ.

Выше упоминалась малая отечественная ПЭВМ БК-0010, подключаемая к бытовому телевизору, на экран которого можно выводить как численно-текстовую, так и графическую информацию. Базовое графическое обеспечение на языке фокал и в машинных кодах дает возможность вывести на экран светлую или темную точку, провести от предыдущей отработанной точки к следующей заданной светлый или темный отрезок. Вариант машины, дающий цветное изображение, позволяет выбирать (кроме черного) один из трех цветов точки или отрезка, "черно-белый" вариант дает линии двух уровней яркости. Язык бейсик, которым стали снабжаться эти ПЭВМ в 1987 году, позволяет дополнительно выводить дуги окружностей, целые окружности, прямоугольники, закрашенные круги, секторы и прямоугольные области, а также закрашивать области, ограниченные замкнутыми линиями — контурами.

Для ввода графической информации "с экрана" предусмотрен специальный режим, при работе в котором курсор, обычно отмечающий место, где будет расположен следующий вводимый символ, заменяется на графический маркер в виде крестика. С помощью тех же кнопок со стрелками (для перемещения курсора) в графическом режиме перемещается маркер — на одну растровую единицу экрана по горизонтали, вертикали или в диагональных направлениях. Кнопка повторения (ПОВТ) заставляет маркер "бежать" в предварительно выбранном направлении. Дополнительно управляющие кнопки (ЗАП и СТИР) заставляют маркер "рисовать" при своем перемещении светлую или темную линию. Точнее, при "рисовании" темной линией маркер работает как резинка, стирая ранее выведенные линии, по которым он проходит. Надо сказать, что нарисовать этим способом на экране нужное изображение сложнее, чем карандашом на бумаге. Другое затруднение — в том, чтобы связать полученную этим способом картинку с вычислительной программой, хранящейся в памяти ЭВМ. Сходные способы "рисования на экране" могут быть реализованы программно, без переключения ЭВМ в специальный режим. Так,

в комплект учебных программ для начального курса машинной графики, реализованный на базе системы Алграф входит программа (сходная с упомянутой выше программой для ЭВМ ЕС), позволяющая задавать типовые фигуры и линии определяющими точками. Алфавитно-цифровая клавиатура здесь работает как функциональная, для удобства на нее накладывают гибкую "маску" со специальными обозначениями. Некоторые кнопки теперь отводятся для перемещения маркера-точки, другие — для регулирования шага перемещения, третьи — для выбора типа линии или фигуры, которые должны выводиться на экран. В верхней строке экрана высвечиваются координаты последней отработанной или запоминавшейся точки, а также величина шага перемещения. С учетом этих данных учащийся перемещает точку-маркер в нужные положения и специальными кнопками запоминает эти положения в качестве *определяющих точек*. Набрав группу из необходимого количества определяющих точек, можно нажать на кнопку порождения геометрического элемента выбранного типа. Такая программа по мере накопления геометрических элементов на экране может запоминать их аналитические модели в памяти ЭВМ; затем они могут быть использованы тем или иным образом. Более сложные программы этого рода сочетают графическое задание элементов с дополнительным вводом численных параметров, указанием операций геометрического конструирования и т.д.

ПЭВМ БК-0010 снабжается программируемым разъемом для подключения различных дополнительных устройств, так называемым *портом ввода-вывода*. 16 контактов в этом разьеме отведены для вывода информации и столько же контактов — для ввода. К этому разьему, составив соответствующую программу, можно подключить и подключаемую рукоятку, которая будет управлять перемещением маркера. В зависимости от силы нажима (величины наклона) можно обеспечить перемещение маркера не только в различных направлениях, но и с разными скоростями.

Использование на ПЭВМ интерпретируемых (а не транслируемых) языков типа фокал, бейсик имеет свои преимущества, и свои недостатки. Отдельные арифметические выражения и несложные программы выполняются быстро, тогда как даже на ЭВМ серии ЕС при использовании фортрана на это требуется значительно больше времени. Но сложные программы с циклами и вызовами подпрограмм при интерпретации выполняются медленно, и это сильно сказывается на графике. Так, окружность или эллипс по программе, составленной на фокале, генерируются от 15 секунд до минуты. Программы, составленные в машинных кодах, могут работать быстрее в 10–20 раз. Вообще говоря, с позиций пользователя удобны такие алгоритмические языки, которые могли бы обрабатываться как в режиме интерпретации, так и в режиме трансляции (компиляции).

Быстрая работа требуется от игровых компьютерных программ. Эти программы нередко приходится составлять в машинных кодах и помимо этого прибегать к различным ухищрениям. Например, если по экрану

”бегают” маленький человек либо подобие какого-то чудовища, это не значит, что каждую секунду многократно перепрограммируется все поле экрана. Небольшой прямоугольный фрагментик изображения, содержащий фигурку, как ”заплата” подменяет фрагментик фоновой картинки (пейзажа, замка, подземного царства) то в одном, то в другом месте.

10.2. Полутоновые изображения

Вопросы формирования, обработки, считывания полутоновых изображений имеют свою специфику. Здесь немного коснемся этих вопросов; тем более, что, хотя и в грубоватом виде, некоторые бытовые и игровые компьютеры способны генерировать цветные и тоновые изображения. Ранее было отмечено, что растр порядка 200×200 или 300×300 точек дает как бы зернистые, с явно выраженной ступенчатостью линий и контуров изображения. При экономии оперативной памяти в малых ЭВМ на каждую точку (элемент изображения) отводятся 2–4 бита информации, что позволяет указывать в точке лишь один или несколько цветов. Изображение получается ”ступенчатым” и в отношении цвета, и в отношении яркости. Для профессиональных задач иногда используются терминалы и печатающие (графические) устройства, рассчитанные на 1024×1024 точек. Для получения цветных полутоновых изображений с плавными (где это нужно) переходами на каждую точку в памяти ЭВМ или терминала надо отвести 2–3 байта. Всего под один ”кадр” изображения тогда будет занято 2–3 Мбайта памяти. Чтобы в нужных случаях быстро сменять несколько кадров изображения или выполнять операции наподобие ”наложения” кадров друг на друга, ”маскирования”, ”сглаживания”, объем памяти надо еще в несколько раз увеличить. Да и просчитать цвет и яркость для полутонового изображения в миллионе составляющих его точек – задача весьма сложная. Поэтому понятно, что для профессиональной работы с полутоновыми цветными изображениями необходимы особо мощные ЭВМ.

Полутоновая и цветная графика в настоящее время быстро развиваются. Как и в более традиционной линейчатой (векторной) графике, здесь двумерные задачи решаются обычно проще, чем задачи построения наглядных изображений трехмерных объектов и тем более задача анализа, ”расшифровки” изображений трехмерных объектов и ”сцен”. Примером несложного алгоритма, работающего с двумерным изображением, может быть упоминавшийся алгоритм закраски (заливки) областей на плоскости, ограниченных замкнутыми линиями контуров. Различия этого алгоритма для полутоновых изображений и простейших изображений, в которых каждая точка может быть только темной или светлой, невелико. Обычно в начальный момент указывается какая-либо принадлежащая закрашиваемой области точка; программа начинает закрашивать в заданный цвет соседние точки, продвигаясь, например, вправо по строке. Но ”предыдущий” цвет каждой точки предварительно анализируется програм-

мой и процесс прекращается, как только будет обнаружена точка с кодом цвета, который соответствует линии контура. Тогда программа аналогично закрашивает строку от заданной начальной точки влево. Затем анализируется следующая вышележащая строка. Здесь уже могут встретиться ситуации, когда надо будет закрасить несколько интервалов из соседствующих точек, так что программа должна запоминать, что уже "закрашено", а к каким местам еще надо будет вернуться. Идя по строкам вверх, программа либо доходит до края изображения, либо до ситуации, когда над закрашивавшимися в предыдущей строке точками расположены только точки граничного контура. Тогда аналогично осуществляется закрашка ниже первоначально обработанной строки. Затем надо проверить, не остались ли еще участки, где возможно продолжение окрашивания. Если машина работает не слишком быстро, можно видеть, как "краска" растекается по экрану, "просачивается" через сужения контура, а в случае ошибки в задании контура иногда и заполняет весь экран (сами линии контура могут либо сохранять свой цвет, либо, по желанию программиста, "перекрашиваться" в цвет области).

Довольно просто работает целая группа алгоритмов поточечной обработки изображения (предварительно уже сформированного в памяти машины). Просматриваются последовательные точки растра, и с учетом их прежних кодов вычисляются новые значения кодов. Например, для построения негатива черно-белого изображения новый код точки можно принять равным $k_2 = k_m - k_1$, где k_1 — ее прежний код, а k_m — максимально возможное значение кода. Немного сложнее подсчет кодов для раскладки полутонового изображения по условным цветам (операция, которая часто применяется для более явного визуального выделения градаций тона). В этом случае задается несколько интервалов яркости точек исходного изображения и для каждого интервала указывается один условный цвет на результирующем изображении. Программа для каждой точки определяет, какому интервалу соответствует ее код, и заменяет ее кодом условного цвета. В таких условных цветах мы видели снимки кометы, изображения, полученные с помощью тепловизоров, и т.д. Имея в памяти ЭВМ области для моделей двух изображений, можно выполнять более сложные операции по наложению изображений или их частей друг на друга, например маскирование или сглаживание контуров и др. Операция типа маскирования нередко выполняется другими средствами на телевидении. Порой можно заметить, как за певцом или комментатором, находящимися на фоне синей (или другого цвета) стены, возникают вдруг кадры пейзажа или какой-либо стройки. Растровые изображения можно подвергать деформациям, перенося точки исходного изображения в соответствующие места результирующего изображения. Преобразование координат здесь выполняется для каждой точки исходного изображения и заключается в определении номера столбца и строки для нее в результирующем изображении. Возможно, какие-либо точки должны

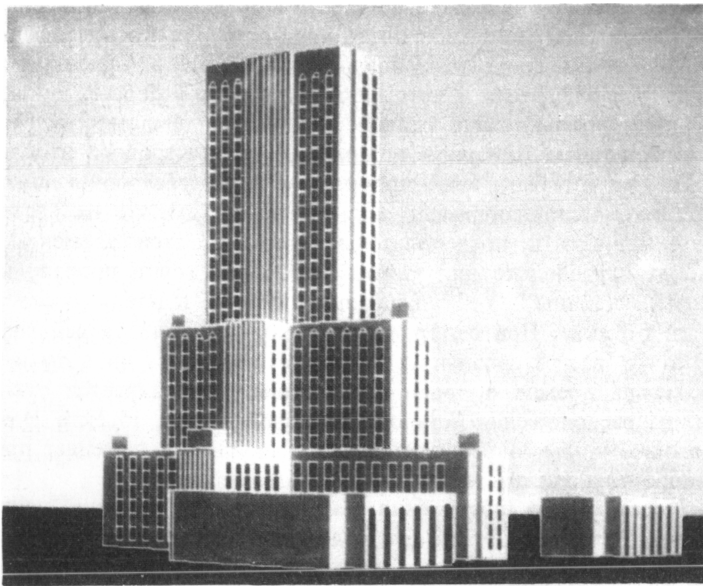


Рис. 142. Полутоновое (в оригинале цветное) изображение на экране растрового графического терминала

быть при этом отброшены (как выходящие за пределы поля результирующего изображения), а другие – добавлены.

Интересна операция "дифференцирования" изображений, когда код точки результирующего изображения зависит от того, близки ли значения кодов для соседних точек в соответствующем месте исходного изображения. В простом случае для такого анализа можно взять группу из девяти точек исходного изображения, восемь из которых "кольцом" окружают среднюю, которая в данный момент исследуется. В частных случаях этим способом из полутонового изображения можно получить штриховое (линейчатое). "Дифференцирование" этого рода полезно при изучении физических полей и в других подобных случаях. Несколько напоминает эту операцию построение линий, соединяющих точки равной плотности или одинакового цвета на исходном изображении.

Построение цветных полутоновых изображений пространственных объектов – достаточно сложная задача (рис. 142). Помимо формирования геометрии аксонометрической или перспективной проекции, здесь надо решать и задачу видимости деталей изображения (и не только линий, но и точек поверхностей). Дополнительные трудности доставляют падающие тени, блики, рефлексы, если их решено моделировать. При упрощенном решении этой задачи (без бликов, изображения фактуры

и падающих теней) поверхности объектов представляются в виде наборов плоских граней, видимые цвет и яркость для которых зависят от "натурного" цвета соответствующей поверхности и ориентации грани относительно наблюдателя и источника света. Такой подсчет не очень сложен, но дополнительно надо решить вопрос о видимости каждого участка грани. Алгоритм контроля видимости для растрового изображения, в принципе, может быть даже проще, чем для изображения линейчатого (штрихового). Основная мысль заключается в том, что на "растровой" картинной плоскости последовательно отображаются элементы, грани объекта, но параллельно для каждой растровой точки производится запоминание "глубины", т.е. удаленности от наблюдателя соответствующей точки объекта. При отображении какой-либо следующей грани для каждой точки растра анализируется, не отображалась ли в ту же точку какая-то грань прежде, и, если отображалась, сравнивается относительная глубина расположения "старой" и "новой" точек объекта. Новая точка, со своим кодом, определяющим цвет и яркость, заменяет прежнюю, если расположена ближе к наблюдателю.

Для качественного построения наглядных изображений используют принцип прослеживания обратного хода световых лучей — от глаза наблюдателя к точкам растра картины и затем — либо "в бесконечность пустоты", либо к точкам граней (поверхностей) объекта. Если луч попал на блестящую или полупрозрачную поверхность, подсчитывается направление отраженного или преломленного луча (условно считается, что свет идет в обратном направлении), прослеживание продолжается по этим лучам и т.д. Ясно, что для синтеза таких изображений с высокой разрешающей способностью необходимы особо мощные машины.

ЗАКЛЮЧЕНИЕ

Быстрый прогресс в области электроники и вычислительной техники требует и своевременного их освоения. Уже сейчас вычислительные машины и графические устройства настолько разнообразны, что играют несоизмеримые, казалось бы, роли – от развлекательных детских игрушек до огромных "фабрик информации и проектирования". Чтобы ЭВМ не остались дорогостоящими игрушками для взрослых, осваивать их надо серьезно и профессионально. Несмотря на быстрое развитие технических средств, их совершенствование, математические основы машинной геометрии и графики, принципы, закладываемые в программы, все же достаточно стабильны. Световое перо может устареть, вычислительные машины из больших шкафов могут перебраться в карман программиста или даже в корпус авторучки, но теорема Пифагора и уравнение эллипсоида не потеряют своей актуальности и в 21 веке!

СПИСОК ЛИТЕРАТУРЫ

1. Александров П.С. Курс аналитической геометрии и линейной алгебры. – М.: Наука, 1979. – 511 с.
2. Гильберт Д., Кон-Фоссен С. Наглядная геометрия. – М.: Наука, 1981. – 344 с.
3. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. – М.: Наука, 1984. – 832 с.
4. Котов Ю.В. Геометрическое конструирование и машинная графика. Ч. 1–3. МАДИ. – М.: 1981, 1983, 1987.
5. Фокс А., Прайт М. Вычислительная геометрия, применение в проектировании и на производстве. – М.: Мир, 1972. – 304 с.
6. Фолли Дж., ван Дэм А. Основы интерактивной машинной графики. Ч. 1, 2. – М.: Мир, 1985.

Юрий Владимирович Котов

КАК РИСУЕТ МАШИНА

Редактор О.И. Сухова

Художественный редактор Т.Н. Кольченко

Технические редакторы С.В. Геворкян, С.Н. Баронина

Корректоры Л.И. Назарова, Т.В. Обод, Т.А. Печко

Набор осуществлен в издательстве
на наборно-печатающих автоматах

ИБ № 32284

Сдано в набор 09.09.87. Подписано к печати 12.01.88. Т-04637

Формат 60 × 88/16. Бумага кн.-журн.

Гарнитура Пресс-Роман. Печать офсетная

Усл.печ.л. 13,72. Усл. кр.-отт. 14,46 Уч.-издл. 13,75

Тираж 140 000 экз. Тип. зак. № 377. Цена 60 коп.

Типография им. Котлякова
издательства "Финансы и статистика"

Государственного комитета СССР

по делам издательств, полиграфии и книжной торговли

195273 Ленинград, ул. Руставели 13

ИЗДАТЕЛЬСТВО "НАУКА",

ГЛАВНАЯ РЕДАКЦИЯ ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ

117071 Москва, В-71, Ленинский проспект, 15

ГОТОВИТСЯ К ПЕЧАТИ:

Смирнов А.Д. Архитектура вычислительных систем: Учеб. Аннотированный тематический план 1989 г., п. 163)

Рассматривается архитектура вычислительных машин и систем от суперЭВМ до мини- и персональных ЭВМ. Охватываются поколения машин от первого до пятого. Материал изложен в структурированной форме: от основных понятий, обзора и эволюции архитектуры и структуры вычислительных систем к логической организации и эволюции основных устройств, примерам ЭВМ и системам различных классов.

Для студентов вузов, а также для специалистов в области информатики.

Предварительные заказы на указанную книгу принимаются без ограничения всеми магазинами Книготорга и Академкниги, распространяющими физико-математическую литературу.

ИЗДАТЕЛЬСТВО "НАУКА",

ГЛАВНАЯ РЕДАКЦИЯ ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ

117071, Москва, В-71, Ленинский проспект, 15

ГОТОВИТСЯ К ПЕЧАТИ:

Авраменко В.С., Легоньков В.И., Хисамутдинов В.Р. Математическое обеспечение диалоговых вычислительных систем – (Б-чка программиста) Аннотированный тематический план 1989 г., п 131)

Рассматриваются вопросы создания автоматизированных информационных систем, анализируются их компоненты, определяется место и роль программного обеспечения в ряду других компонент. Основные разделы посвящаются программной системе "МОДИС", обеспечивающей возможность построения и эксплуатации информационных систем различного назначения в среде ОС "ДИСПАК" ЭВМ БЭСМ-6 и ОС ЕС для ЕС ЭВМ.

Для разработчиков автоматизированных информационных систем, программистов и студентов вузов.

Предварительные заказы на указанную книгу принимаются без ограничения всеми магазинами Книготорга и Академкниги, распространяющими физико-математическую литературу.

ИЗДАТЕЛЬСТВО "НАУКА",

ГЛАВНАЯ РЕДАКЦИЯ ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ

117071 Москва, В-71, Ленинский проспект, 15

ГОТОВИТСЯ К ПЕЧАТИ:

Эйрис Р. Проектирование СБИС. Метод кремниевой компиляции: Пер. с англ. Аннотированный план 1988 г., п 51)

Введение в кремниевую компиляцию – новую методику автоматического проектирования микрокристалла. Это перспективный подход к задаче проектирования СБИС – сверхбольших интегральных схем. Излагаются основные понятия и методы, необходимые для создания, понимания и эксплуатации систем для кремниевой компиляции.

Для научных работников, инженеров, аспирантов и студентов в области информатики и вычислительной техники, а также для разработчиков систем автоматизированного проектирования.

Предварительные заказы на данную книгу принимаются без ограничения всеми магазинами Книготорга и Академкниги, распространяющими физико-математическую литературу.