

В.М.Голубев

ПРОСТАВЛЕНИЕ РАЗМЕРНОЙ И ТЕХНОЛОГИЧЕСКОЙ ИНФОРМАЦИИ НА МАШИНОСТРЭИТЕЛЬНЫХ ЧЕРТЕЖАХ

I. Введение

Перед автором настоящей статьи стояла задача разработать пакет программ, в рамках языка ГРАФИТ [1], позволяющий получать машиностроительные чертежи, полностью оформленные по стандартам ЕСКД. В статье обсуждается ряд проблем, встающих при машинной формализации ГОСТов, приводятся некоторые методы и приемы, которые могут оказаться полезными для разработки аналогичного программного обеспечения.

2. Проблемы машинной формализации ГОСТов

Основные трудности при разработке блока оформления машиностроительных чертежей вызвала та часть [2], в которой описываются правила нанесения размерной и технологической информации на чертеж. К ним относятся правила нанесения линейных, угловых, дуговых, диаметральных и радиальных размеров, уклонов и конусностей, обозначения шероховатостей, покрытий, швов сварных и неразъемных соединений и т.п. В дальнейшем, для краткости, будем называть все эти понятия одним словом: размеры. Именно размеры являются наиболее трудными объектами для формализации. Основная причина этого – нефор-

мальный характер самих ГОСТов, ориентированных на человека. Так, например, в ГОСТах описаны правила нанесения линейных размеров, но определение того, что такое линейный размер, отсутствует. Поэтому приходится начинать формализацию с формальных определений размеров.

3. Формальные определения размеров

В этом пункте будет проведен анализ размеров с точки зрения их изображения (графики) и функционального назначения. Будут выделены базовые элементы (БЭ), из которых собственно состоят размеры, и показано, как синтезируются размеры из БЭ с учетом целого ряда закономерностей (отношений). В конце приведем ряд предложений по формализации этих отношений.

3.1. Анализ размеров

Анализ [2], а также машиностроительных чертежей показывает, что, несмотря на сложную переменную структуру, все размеры состоят из небольшого числа следующих элементов, которые в дальнейшем будут называться базовыми элементами (БЭ):

- 1 - размерная линия - РЛ (отрезок прямой или дуга окружности);
- 2 - ограничитель размерной линии - ОГР (стрелка, засечка, точка и т.п.);
- 3 - продолжение размерной линии - ПРЛ (отрезок прямой или дуга окружности);
- 4 - выносная линия - ВЛ (отрезок прямой);
- 5 - продолжение выносной линии - ПВЛ (отрезок прямой);
- 6 - полка - П (отрезок прямой);
- 7 - размерная надпись - РН (набор заглавных, строчных, подстрочных, надстрочных и специальных - знаки диаметра, квадрата и т.п. - символов).

На рис. I. приведено несколько примеров размеров (базовые элементы обозначены соответствующими цифрами).

Заметим, что такое представление еще сохраняет функциональное назначение размера. Если же посмотреть на БЭ как просто на графические изображения, то все они состоят из та-

ких минимальных (атомарных) элементов (АЭ), как

- отрезки прямых (БЭ № I - 6);
- дуги окружностей (БЭ № I, 2, 3);
- символы (БЭ № 2, 7).

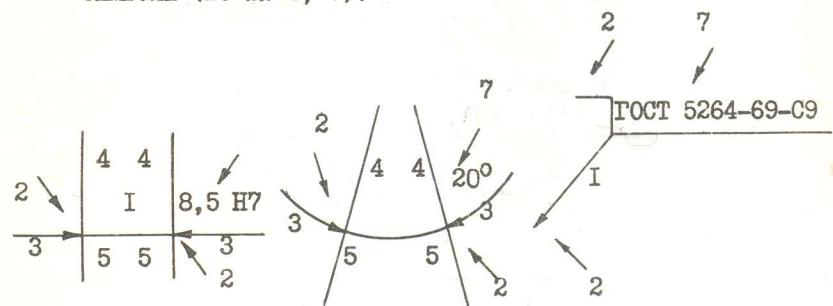


Рис. 1

Представление размера в виде набора АЭ уже не несет в себе функционального значения (по такому представлению практически невозможно восстановить, например, тип размера), но очень удобно для вывода на графическое устройство. Каждый АЭ можно описать набором чисел: отрезок – $(x, y, \varphi, 1)$ (рис. 2), дуга окружности – $(x, y, \varphi, r, d\varphi)$, символ – (x, y, H, K, n) , где H – размер шрифта, K – тип шрифта (широкий – обычный, с наклоном – без наклона), n – номер символа. Конечно, возможны и другие наборы числовых параметров, описывающих БЭ.

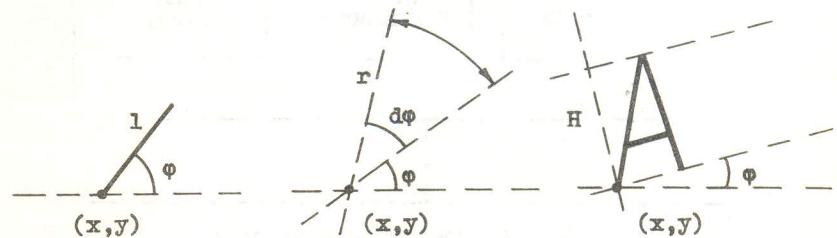


Рис. 2

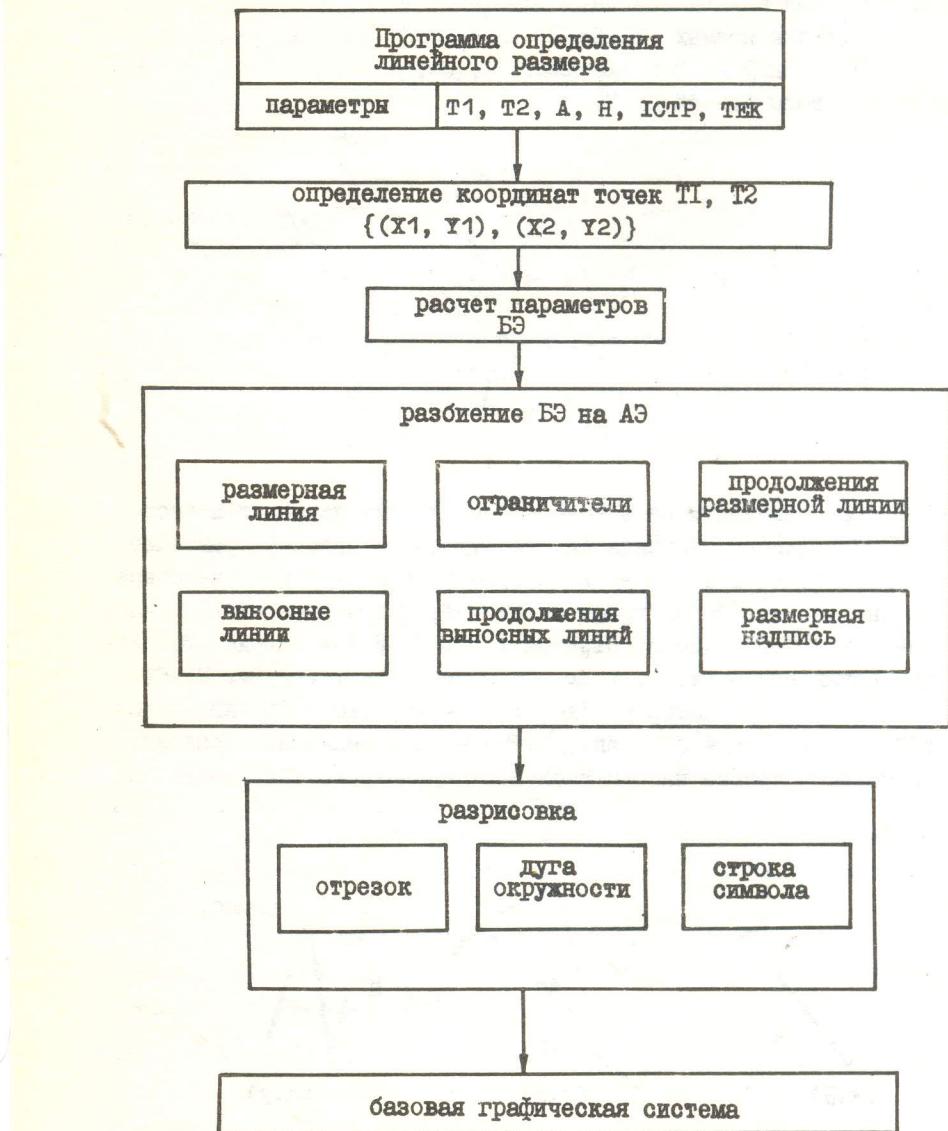


Рис. 3

3.2. Синтез размеров из БЭ

Все базовые элементы могут находиться только в некоторых определенных отношениях друг с другом. Эти отношения будем называть базовыми отношениями (БО). Например, в линейных размерах выносные линии всегда параллельны друг другу, а в угловых - радиальные, начало продолжения размерной линии всегда совпадает с одним из концов размерной линии и т.п. Так же как и атомарные, базовые элементы можно представить в виде набора числовых параметров. Например, ограничитель размерной линии - $\{x, y, \phi, t\}$, размерная надпись - $\{x, y, \phi, \text{строка символов}\}$, где $\{x, y\}$ - координаты начала БЭ, ϕ - угол наклона БЭ, t - тип ограничителя.

Конечно, базовых отношений больше, чем БЭ, но все их, в принципе, можно выписать в виде некоторых математических соотношений между параметрами БЭ.

После того как выделено множество базовых элементов и множество базовых отношений между ними, можно дать формальное определение размера любого типа. Для этого нужно

- указать, из каких БЭ может состоять размер;
- определить, в каких пределах могут изменяться параметры БЭ;
- перечислить, каким из базовых отношений должны удовлетворять БЭ размера.

Заметим, что такая схема предоставляет полную свободу разработчику программного обеспечения в выборе БЭ и БО для каждого типа размера. (Например, он может включить, а может и не включить в радиальный размер БЭ типа "продолжение размерной линии" на том основании, что такой БЭ в радиальных размерах встречается довольно редко)

3.3. О способах задания БЭ

Для того чтобы размер был полностью определен, должны быть определены значения всех параметров всех его БЭ. Часть параметров может задать сам пользователь в явном виде, указав их значения; часть может вычислить система, используя БО; некоторые параметры могут быть вычислены, если учесть, что все размеры всегда проводятся к геометрическим объектам чертежа. Так, например, линейный размер - к отрезку, диаметральный - к окружности, дуговой - к дуге окружности и т.д.

Если значение некоторого параметра не задано явным образом пользователем и его нельзя получить с помощью базовых отношений или ссылок на геометрические объекты, то система должна дать этому параметру некоторое стандартное значение по умолчанию, либо вычислить это значение каким-то стандартным образом. Например, длины продолжений выносных линий обычно неизменны в пределах чертежа, и их можно не задавать каждый раз заново при определении очередного размера. Другой пример: рекомендация ГОСТа проводить линейные размеры со стрелками внутрь, если позволяет длина размерной линии, в противном случае — наружу. (Заметим, что эту рекомендацию нельзя причислить к списку БО, поскольку кроме размерных стрелок в линейных размерах могут быть и другие ограничителя.) В этом случае пользователь может либо явно указать типы ограничителей, либо это сделает за него система, в соответствии с вышеупомянутой рекомендацией.

4. Пример программы определения линейных размеров

В этом пункте приводится пример реализации вышеизложенной схемы для разработки программы определения линейных размеров.

Линейный размер состоит из размерной линии, ограничителей размерной линии, выносных линий, продолжений выносных линий, продолжений размерной линии и размерной надписи. Обращение к соответствующей программе ППП ГРАФИТ имеет следующий вид:

$R = RL2T(T1, T2, A, H, ICTP, TEK),$

где R — имя линейного размера, T1, T2 — имена начальной и конечной точек отрезка, к которому проводится размер, A — параметр, задающий углы наклона размерной и выносных линий к оси ОХ, H — длина одной из выносных линий, ICTP — параметр, задающий типы ограничителей, TEK — параметр, задающий длину одного из продолжений размерной линии. С помощью фактических параметров программы **RL2T** пользователь задает значения лишь некоторых параметров базовых элементов (всего таких параметров свыше сорока). Остальные параметры определяются либо ис-

ходя из ссылок на точки T1, T2, либо по умолчанию. По умолчанию, например, может определяться строка символов размерной надписи (исходя из длины размерной линии). При желании пользователь может явно задать размерную надпись на размере. Для этого он должен после обращения к программе RL2T обратиться к оператору определения размерных надписей, указав имя размера, расположение размерной надписи на размере и задав строку символов, в которой закодирована размерная надпись. На рис. 3 приведена блок-схема программы RL2T.

Для простоты на этой блок-схеме не отражен ряд операторов программы, реализующих занесение информации о размере в общую модель чертежа, и т.п.

Автор благодарен В.Г.Сиротину, В.Ю.Шилкову, В.И.Ниронову, а также всем сотрудникам лаборатории машинной графики ВЦ СО АН СССР за помощь в работе.

Л и т е р а т у р а

1. ГОСТ 2. 301-68-ГОСТ 2. 317-69. Общие правила выполнения чертежей. Введ. 01.01.71. - Переизд. июнь 1978.
2. Сиротин В.Г. Язык ГРАФИТ для описания, редактирования, хранения и визуализации моделей двумерных геометрических объектов и их чертежей. - Настоящий сб., с.63-91.

В.М.Годубев, Ю.А.Савров, В.Г.Сиротин,
В.П.Токарев, Б.С.Чурюмов, Н.С.Шунга

НЕКОТОРЫЕ ПРАКТИЧЕСКИЕ АСПЕКТЫ ПРИМЕНЕНИЯ
СРЕДСТВ МАШИННОЙ ГРАФИКИ
В СИСТЕМЕ АВТОМАТИЗАЦИИ ТЕХНОЛОГИЧЕСКОЙ
ПОДГОТОВКИ ПРОИЗВОДСТВА

В настоящей статье анализируется опыт разработки и первоначальной эксплуатации экспериментальной версии системы автоматизированного проектирования режущего инструмента (САПР РИ). Приводятся рекомендации по некоторым вопросам создания подобного рода систем, а также некоторые соображения о том, каким образом следует развивать и внедрять средства машинной графики в подразделениях, ответственных за технологическую подготовку производства (ТПП).

В настоящее время машинная графика в ТПП применяется в основном в подсистемах получения и контроля управляющих программ для станков с ЧПУ. Новым моментом в описываемой системе является то, что она нацелена на решение другой важной задачи ТПП – задачи подготовки инструмента.

Перечислим требования, предъявляемые к САПР РИ при разработке ее технического проекта:

- модельность системы (т.е. в дальнейшем она должна послужить в качестве образца для изготовления аналогичных систем);
- модульность системы (система состоит из модулей двух классов – общесистемных, которые могут быть использованы при

изготовлении других подобных систем, и предметно-ориентированных);

- проектирование ведется по схеме, аналогичной схеме ручного проектирования;

- диалог конструктор - ЭВМ протекает в терминах конструктора-практика.

Описываемая версия системы разрабатывалась для проектирования расточных резцов. Система функционирует на аппаратной конфигурации АРМ-Р (СМ-3): БЭСМ-6.

Для разработки и реализации САПР РИ был организован коллектив сотрудников разных учреждений - специалистов по системному программированию, технологии и конструкторов-практиков.

Специальные алгоритмы по расчету прочности резцов и его параметров были составлены специалистами научно-технологического института. Эти алгоритмы были проверены практиками-инструментальщиками на конкретных данных, ими же были сделаны некоторые упрощения и изменения в алгоритмах на основе их собственного опыта.

Схема функционирования САПР РИ показана на рис. 1. Результат работы системы - фрагмент чертежа одного из спроектированных инструментов - приведен на рис. 2.

Как уже отмечалось выше, проектирование инструмента с помощью САПР РИ проходит те же этапы, что и при ручном способе проектирования. Более того, создаваемые на различных этапах модели имеют свои аналоги - документы, используемые при ручном проектировании. Так, М1 (см. рис. 1) эквивалентна чертежу обрабатываемой детали, М2 вместе с М3 - технологической карте. Справочные данные Д1 - это просто размещенные в памяти ЭВМ данные из технологических справочников. Технолог в явном виде не составляет документ, эквивалентный М4, но для себя он всегда выписывает параметры проектируемого инструмента.

Диалог пользователя с технологом является диалогом, инициируемым ЭВМ. Технолог на различных этапах проектирования инструмента выполняет операции одного из трех типов:

- ответ на "простой" вопрос,

- ответ на "составной" вопрос - заполнение нужных позиций в некоторой таблице,

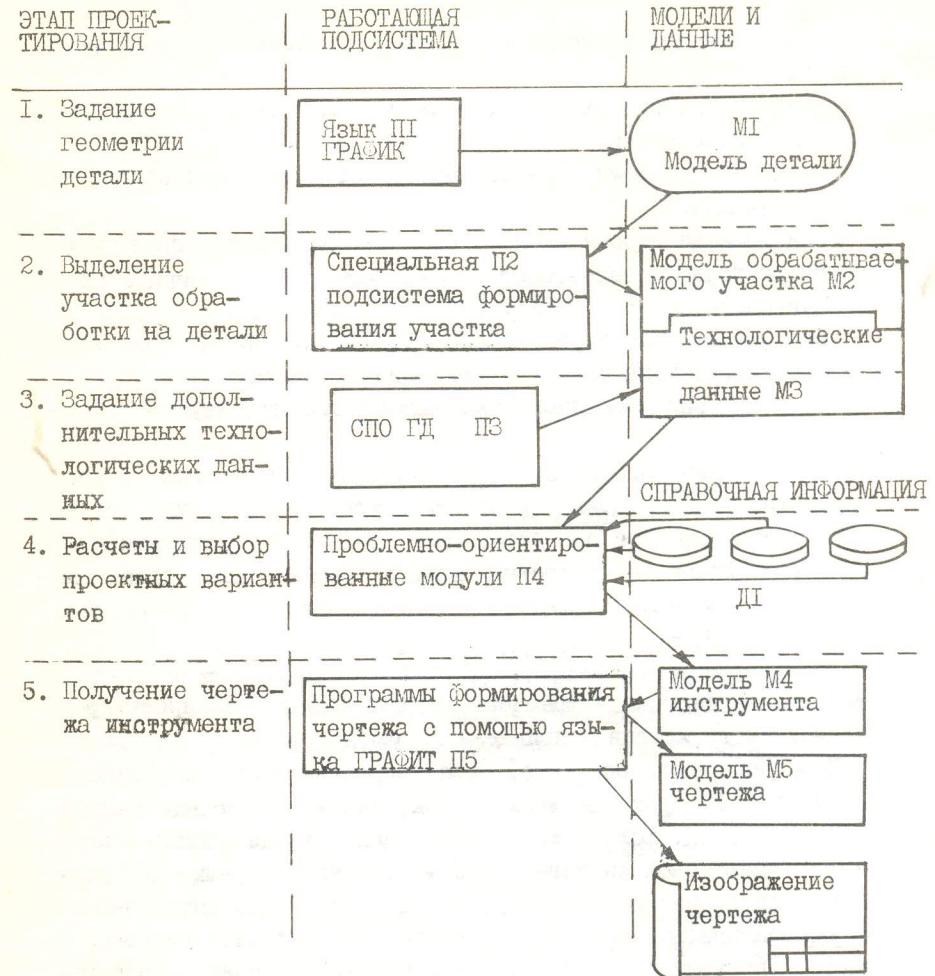
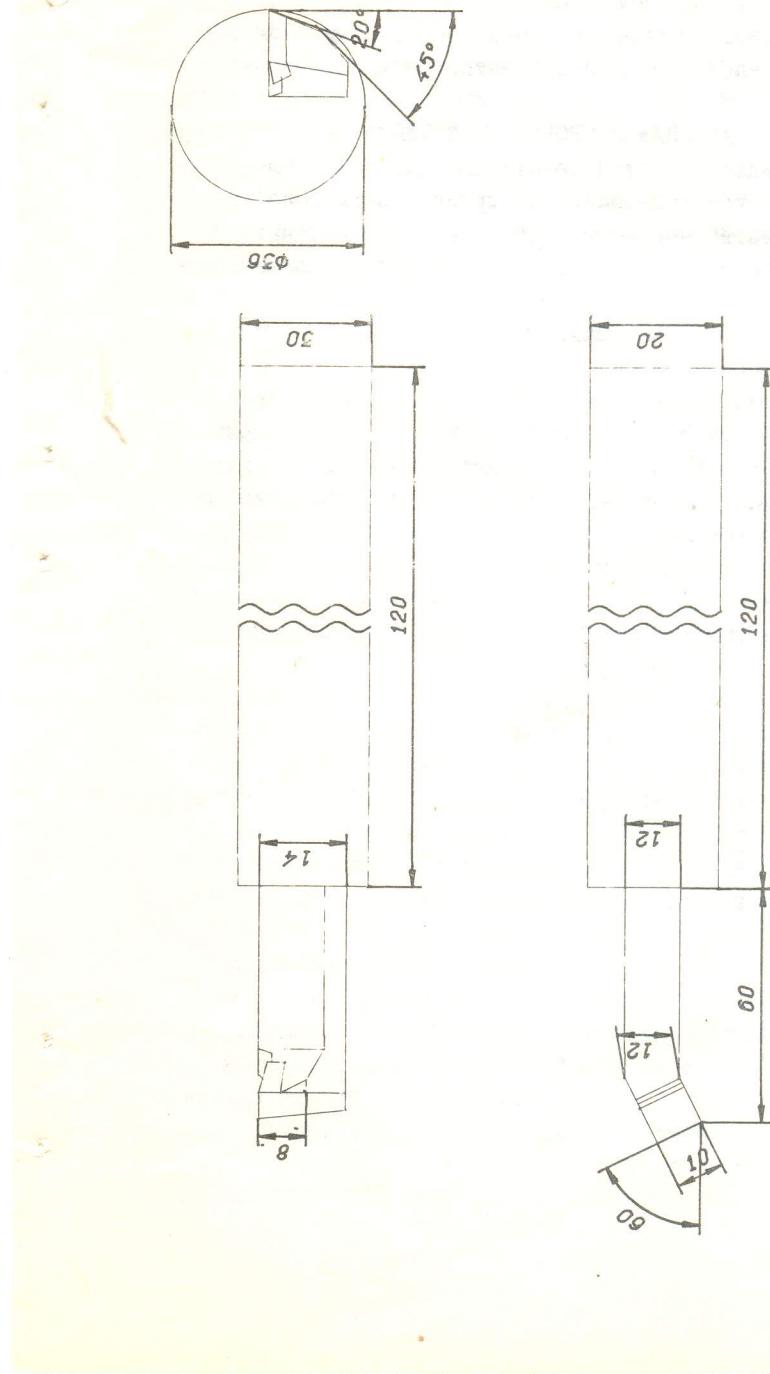


Рис. I

Fig. 2 (Instrument support pedestal)



– выбор одного из ~~нескольких~~ вариантов.

В качестве "простого" вопроса можно привести следующую строку, которая на определенном шаге проектирования высвечивается на экране:

ВВЕДИ МАТЕРИАЛ И ПРОЧНОСТЬ ДЕТАЛИ

В ответ пользователь должен ввести два числа. Из смысловых соображений иногда целесообразно группировать подобного рода запросы. Естественной формой представления сложных запросов (анкет) является таблица, в которой нужно заполнить некоторые графы.

Выбор варианта при интерактивном проектировании осуществляется следующим образом.

На экране дисплея появляется схематическое изображение объектов нескольких типов (например, несколько головок резца). С помощью светового пера пользователь выбирает нужный ему объект указанием на него. При изготовлении новых систем аналогичного назначения могут быть использованы уже разработанные базовые компоненты ПП и ПЗ.

Кроме того, может быть использована и часть специальных программ из П4, например, программы расчета державок, которые одинаковы для всех типов резцов.

Проедем сравнительный анализ различных фаз ручного и машинного проектирования на примере расточных резцов.

Затраты времени конструктора на различных этапах проектирования при ручном проектировании и проектировании с помощью САПР РИ сравниваются на рис. 3. – заштрихованная нижняя часть отражает временные затраты при ручном проектировании, а верхняя часть – при машинном.

Как видно из рис. 3, временные затраты на ручное вычерчивание эскиза детали и описание ее формы с помощью языка ГРАФИТ примерно одинаковы (этап I). Комплексность в решении этого вопроса позволит полностью исключить временные затраты технолога на этом этапе, если чертежи будут поступать на завод из конструкторского бюро в виде данных на машинных носителях.

Рассмотрим этапы II – У. Наиболее выгодно САПР РИ показы-

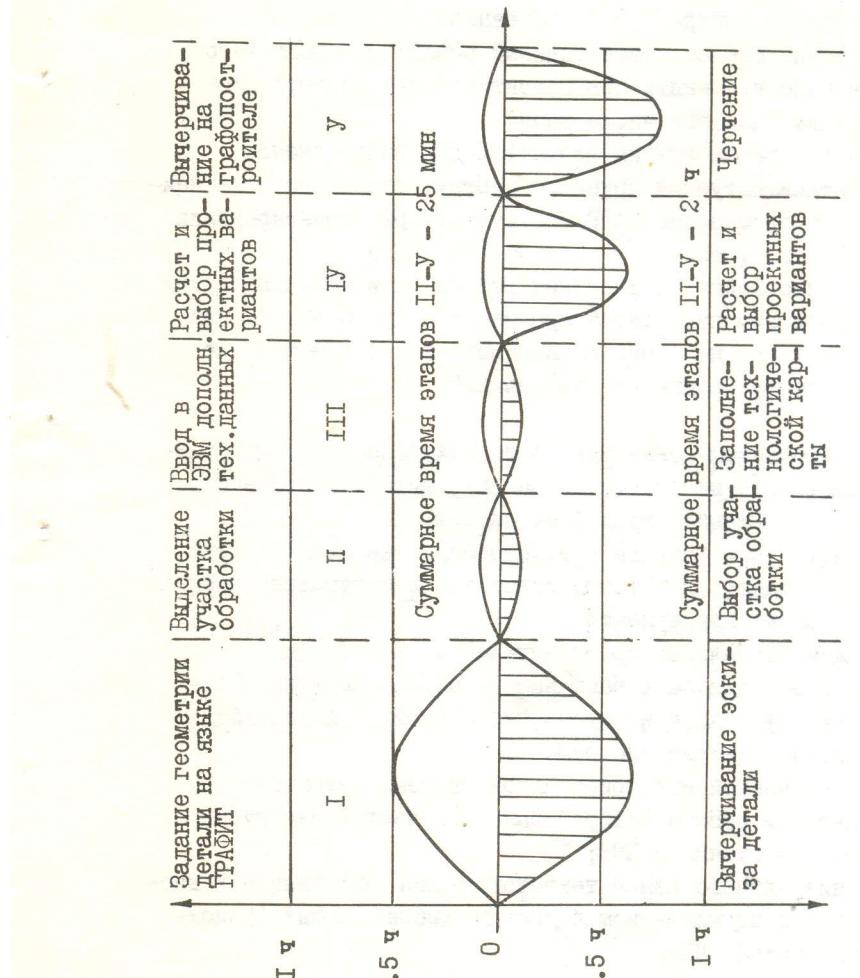


Рис. 3

вает себя на этапах IY - Y, так как ЭВМ быстрее производит арифметические вычисления, чем человек, а граffопостроители вычерчивают чертеж быстрее, чем чертежник.

Отсюда можно сделать естественный вывод, что подобного рода системы надо применять при проектировании инструментов

- требующих большого числа вычислений;
- сложных с точки зрения геометрии для вычерчивания.

В конкретной ситуации крупного инструментального цеха наиболее выгодно применение САПР Инструмент при проектировании фасонных резцов, фрез.

Начинать необходимо с фасонных резцов, при проектировании которых технологам приходится производить много вычислений, но чертежи которых не очень сложно получать с помощью уже разработанного программного обеспечения.

Важным источником повышения производительности труда технологов может быть не только применение автоматизированных систем, но и улучшение самих этих систем.

Для качественного сдвига в этой области необходимо решение следующих задач, к которым советские и зарубежные специалисты пока только приступают:

- нахождение аналогов среди деталей по их чертежу.
Эта задача позволила бы в базе данных со связью типа [деталь, инструмент для ее изготовления] сразу находить подходящий для данной детали инструмент;
- создание единой интегрированной системы автоматизации ТПП, результатом работы которой был бы и чертеж инструмента, и программа для станка с ЧПУ;
- создание единого банка технологической информации в масштабах завода с подключением к другим банкам данных (родственных предприятий, КБ).

С.А.Упольников

АЛГОРИТМЫ КОНСТРУИРОВАНИЯ МОДЕЛЕЙ ТРЕХМЕРНЫХ ОБЪЕКТОВ НА ЭВМ

В лаборатории машинной графики ВЦ СО АН СССР, в рамках создания интерактивной графической системы общего назначения, в настоящее время разрабатывается пакет прикладных программ **SPACE**, предназначенный для решения трехмерных графических задач на ЭВМ.

Пакет **SPACE** ориентируется на решение широкого круга задач, возникающих, в частности, при автоматизации процессов моделирования, конструирования и расчета трехмерных конструкций на ЭВМ в рамках САПР.

Статья посвящена описанию базовой модели объекта, принятой в пакете **SPACE**, и ряда алгоритмов конструирования и расчета некоторых характеристик этих моделей. Предварительно дается краткая спецификация задач, на решение которых ориентируется пакет.

I. Классификация задач

Под трехмерными графическими задачами мы понимаем классические задачи переработки информации с помощью ЭВМ (т.е. ввод, хранение, обработка и вывод), но "имеющие дело" с трехмерной геометрической информацией и графическими устройствами ЭВМ. Специфика этих задач определяется, с одной стороны, тем, что

современные ЭВМ практически не приспособлены для хранения и обработки геометрической информации в адекватном для ее описания или восприятия человеком виде, с другой стороны, восприятие и описание сложных геометрических форм само по себе очень трудоемко и требует автоматизации.

Таким образом, главной задачей пакета **SPACE** является обеспечение пользователя набором специальных программных средств, позволяющих ему решать прикладные задачи на ЭВМ, по возможности с минимальными затратами на преодоление указанной выше специфики трехмерной геометрической информации.

Сразу отметим, что мы не рассматриваем здесь задачи непосредственного вывода информации на графические устройства, ввода информации в режиме диалога с графических устройств и хранения информации на внешних носителях. Предполагается, что эти задачи решаются разрабатываемыми независимо от пакета **SPACE** подсистемами, а в пакете предусматриваются средства обеспечения интерфейса с ними.

I.I. Построение модели трехмерного объекта и ее представление в ЭВМ – наиболее существенная проблема, поскольку то, как решаются задачи, в нее входящие, определяет, по сути дела, основные возможности пакета **SPACE** как в отношении описания разнообразия исходных форм, так и в отношении реализации основных операций над объектами. Здесь мы выделяем следующие задачи:

- обеспечение возможности описания геометрической модели объекта двумя способами: совокупностью граничных поверхностей и набором базовых элементов для конструирования с помощью теоретико-множественных операций;
- задание и ввод в ЭВМ элементарных объектов в рамках выбранной модели, а также контроль их корректности;
- построение или компоновка объекта из уже определенных ранее;
- представление объекта в памяти ЭВМ в удобном виде для хранения или последующей обработки, в частности, преобразование представлений;
- хранение атрибутов подобъектов в представлении объекта и возможность их использования;

- использование библиотеки типовых и конструктивных элементов.

I.2. Задачи, относящиеся к средствам редактирования модели объекта и структурных операций над ней, являются, в некотором смысле, внутренними, поскольку их необходимо решать практически при любой работе с моделями объектов или их представлениями. Выделим следующие задачи:

- поиск элементов объекта по его представлению и их корректировка;
- изменение параметров описания элементов без изменения общей структуры объекта;
- замена или удаление элементов объекта в рамках используемой модели;
- выбор и присвоение атрибутов подобъектам.

I.3. Задачи, связанные с вычислением физических и геометрических характеристик трехмерного объекта по его модели носят вспомогательный характер и ориентированы на использование пакета в САПР. В этой группе естественно выделяются следующие задачи:

- вычисление физических характеристик объектов, таких, как длины, площади, объемы, центры и моменты инерции;
- вычисление геометрических характеристик, т.е. позиционных соотношений между объектами, а также размеров и расстояний между ними.

I.4. Задачи построения и визуализации плоских моделей исходного трехмерного объекта являются наиболее важными, поскольку дают возможность человеку оценить результат его работы по моделированию объекта и получить модель документа, т.е. рисунка или чертежа. Здесь выделяются следующие задачи:

- выбор графического образа объекта для более адекватного его восприятия человеком;
- построение плоских объектов по исходным моделям, т.е. проекций, разрезов, сечений;
- построение специальных плоских графических объектов исходных объектов с удалением невидимых линий или поверхностей;

- визуализация плоских моделей или "передача" их другим специализированным пакетам, например, для построения чертежей.

Описанные выше постановки задач являются весьма общими и конкретизируются, в частности, при решении вопросов о

- выборе внутренней или базовой модели объекта;
- способе представления модели в памяти ЭВМ.

Ниже будет описана базовая модель объекта, принятая в пакете **SPACE**, и ряд алгоритмов конструирования и расчета этих моделей. Следует подчеркнуть, что выбором базовой или внутренней модели объекта существенно определяются возможности использования различных способов описания исходных форм, а также методы представления и алгоритмы решения многих из описанных выше задач.

Относительно представления модели объекта в памяти ЭВМ заметим следующее. Считаем, что модель объекта представляется в виде: конечного набора точек в R^3 , заданных своими координатами в единой прямоугольной декартовой системе координат, - координатная информация; некоторой иерархической структурой над ними, заданной в виде набора списков, - структурная информация; множества различного типа атрибутов (векторы, числа, тексты), связанных с элементами структуры, - атрибутная информация. Ясно, что аффинные преобразования, такие как поворот, сдвиг и масштабирование, реализуются простым пересчетом координатной и геометрической информации. Более детальное описание представления модели объекта в этой статье не рассматривается.

2. Базовая модель объекта

В качестве основной внутренней модели объектов в пакете **SPACE** выбрана модель полиэдров. Такие объекты, как точки, ломаные, многоугольники, кусочно-линейные поверхности и многогранники, являются полиэдрами. В частности, многоугольники и многогранники с внутренними "перегородками" также являются полиэдрами. Привлекательность этой модели определяется следующими причинами:

- универсальность, так как в рамках этой модели можно построить аппроксимацию практически любых объектов, с любой степенью точности и даже с учетом внутренней неоднородности ("перегородками");
- полиэдры просты в описании, и их представление адекватно представлению объекта человеком во многих случаях;
- алгоритмы решения многих задач для полиэдров проще, чем для других моделей;
- графический образ является подмножеством исходной модели, т.е. "каркас" уже есть;
- полиэдры очень удобны для представления в ЭВМ, поскольку определяются конечным набором точек и некоторой структурой над ними, в частности, это дает возможность очень естественно включать атрибутные (не геометрические) характеристики объектов в структуру их модели, что существенно для моделирования реальных объектов.

Понятие полиэдра возникло в комбинаторной топологии [I] и базируется на понятии симплекса. Неформально можно сказать, что симплексами в R^3 являются точки, отрезки ненулевой длины, треугольники ненулевой площади и тетраэдры ненулевого объема, т.е. некоторые элементарные куски пространства с кусочно-линейной границей. Оказалось, что представление многообразий в виде конечного набора правильно примыкающих друг к другу элементарных кусков является существенно более удобным и мощным средством для изучения их топологических и геометрических свойств, чем представление в виде систем аналитических уравнений и неравенств. Поскольку наиболее общей геометрической моделью реальных объектов является многообразие в R^3 , то выбор полиэдров в качестве основной внутренней модели объектов нам кажется очень естественным [I].

Рассмотрим векторное евклидово пространство R^3 и точечные множества в нем. Будем считать, что в R^3 задан ортогональный базис и все точки и вектора определяются в этом базисе. Ниже предлагается формальное построение моделей объекта и операций над ними, которое является теоретическим обоснованием рассматриваемых в следующем пункте алгоритмов.

2.1. Система точек $x_0, \dots, x_k \in R^k$ ($k \leq 3$) называется независимой, если система векторов $x_i - x_0$ линейно независима [I].

2.2. Множество точек x из R^3 , представимых в виде
$$x = \sum_{i=0}^k \alpha_i x_i, \quad \text{где } x_0, \dots, x_k \text{ - независимы, } \alpha_i \in R, \sum_{i=0}^k \alpha_i = 1$$
 и α_i - неотрицательные действительные числа, называется симплексом размерности k [I]. Точки x_0, \dots, x_k называют вершинами симплекса и пишут: $C^k = (x_0, \dots, x_k)$.

2.3. Пусть $C^k = (x_0, \dots, x_k)$ - симплекс. Тогда симплекс, образованный подмножеством из $\{x_0, \dots, x_k\}$, называется гранью симплекса, или граничным симплексом C^k . Если подмножество истинно, т.е. не совпадает со всем множеством, то грань называется истинной [I].

2.4. Два симплекса расположены правильно, если они либо не пересекаются, либо их пересечение является гранью каждого из них. Ясно, что грани одного симплекса расположены правильно [I].

2.5. Конечная совокупность симплексов называется комплексом, если в нее входят, наряду с каждым симплексом, все его грани и каждые два симплекса этой совокупности расположены правильно [I]. Размерностью комплекса называют максимальную из размерностей входящих в него симплексов. Вершинами комплекса называют его 0-мерные симплексы-точки.

2.6. Полиэдром называется объединение точек всех симплексов, входящих в некоторый комплекс [I]. Собственно это и является наиболее общей моделью объекта, принятой в пакете **SPACE**. Понятие комплекса будем рассматривать как некоторый способ задания объекта или полиэдра. Нетрудно видеть, что комплекс представляет собой конечное множество точек с выделенным на нем семейством подмножеств, являющихся вершинами входящих в него симплексов. Очевидно, что если разбить все выделенные подмножества на классы, в зависимости от размерностей определяемых ими симплексов, т.е. 0, 1, 2, 3, то можно представить

комплекс в виде естественной иерархической структуры над множеством 0-мерных симплексов или точек.

2.7. Комплекс размерности k (и, соответственно, полиэдр, им порожденный) будем называть однородным, если любой симплекс размерности $1 < k$, входящий в комплекс, является гранью некоторого симплекса размерности k , также входящего в этот комплекс. Нетрудно видеть, что любой комплекс размерности k представляет собой совокупность однородных комплексов размерностей 0, I, ..., k .

Мы будем рассматривать только однородные объекты (точнее, их подмножество, определенное ниже), поскольку именно они являются естественными моделями таких геометрических объектов, как ломаные и поверхности. Объединение однородных полиэдров в объект общего вида, т.е. неоднородный полиэдр, представляет собой задачу построения общей модели исходного реального объекта, учитывающей, в частности, его атрибутные (не геометрические) характеристики, и здесь не рассматривается.

В качестве естественного задания однородного объекта размерности k можно рассматривать не весь порождающий его комплекс, а только входящие в него симплексы размерности k , поскольку все остальные симплексы этого комплекса являются их гранями, т.е. если несколько сузить понятие комплекса, данное в 2.5 и, в частности, считать, что в определении 2.4 исключено совпадение симплексов, то ясно, что класс однородных объектов при этом не изменится. Мы вернемся к этому ниже.

2.8. Под многообразием, порождаемым симплексом, в дальнейшем будем понимать аффинное многообразие минимальной размерности, его содержащее. Ясно, что размерность порожденного многообразия для симплекса s^k равна k .

2.9. Однородный комплекс (и, соответственно, объект, им порождаемый) будем называть телесным, если порожденные многообразия входящих в него симплексов совпадают. Телесные объекты и операции над ними являются основными объектами изучения в данной статье. Однородные нетелесные объекты представляются

в виде совокупности телесных объектов (например, сами симплексы, их составляющие) и используются в пакете SPACE при представлении исходных форм поверхностей и кривых; мы их пока не рассматриваем.

2.10. Множество из E^k называется регулярным в E^k , если оно совпадает с замыканием своей внутренности в топологии E^k , т.е. множество A регулярно, если $A = \bar{A}$. На множестве подмножеств E^k можно ввести операцию регуляризации, т.е. $r(A) = \bar{A}$. Рассмотрим следующие операции над регулярными множествами в E^k .

$$\begin{aligned} A \cup^* B &= r(A \cup B), \\ A \cap^* B &= r(A \cap B), \\ A \setminus^* B &= r(A \setminus B); \end{aligned}$$

где \cup , \cap , \setminus - обычные теоретико-множественные операции. Будем называть операции \cup^* , \cap^* , \setminus^* регулярными теоретико-множественными операциями [2]. Нетрудно видеть, что регулярные подмножества в E^k образуют алгебру относительно этих операций [2]. 19

2.11. Рассмотрим замкнутые полупространства H из E^k , образованное аффинными многообразиями M размерности $k-1$, и множества вида $B^*(H_1, \dots, H_n)$, где B^* - регулярная булева функция от конечного числа полупространств. Тогда нетрудно видеть, что такие множества образуют подалгебру в алгебре регулярных множеств E^k .

Выделим в этой подалгебре класс ограниченных множеств. Очевидно, он замкнут относительно регулярных теоретико-множественных операций, введенных выше.

Теорема. Класс ограниченных конечно-порожденных полупространствами множеств из E^k совпадает с классом телесных объектов размерности k из E^k , определенных в 2.9.

Доказательство. Каждый телесный объект может быть представлен в виде регулярной булевой функции от полупространств, поскольку каждый симплекс S^k из E^k является пересечением $k+1$ полупространства, образованных многообразиями, порожденными его граничными симплексами, размерности $k-1$. Таким образом, в одну сторону включение доказано. Рассмотрим

ограниченное множество A , являющееся регулярной булевой функцией от конечного набора $H_i (i = 1, \dots, n)$ полупространств из \mathbb{R}^k , образованных многообразиями M_i , т.е. $A = B^*(H_1, \dots, H_n)$. По известной теореме [3], эту функцию можно привести к каноническому виду $A = \bigcup_j A_j$, где $A_j = \bigcap_{i=1}^n z_i (z_i = H_i \text{ или } \mathbb{R}^k \setminus H_i)$, причем $A_{j_1} \cap A_{j_2} = \emptyset$, если $j_1 \neq j_2$. Очевидно, что A_j — выпуклые множества с кусочно-линейной границей, легко могут быть представлены в виде конечного набора правильно расположенных симплексов. Покажем, что если A_i и A_j имеют общую границу, то она является гранью обоих элементов. Пусть M_{io} — многообразие, задающее H_{io} и $H'_{io} = \mathbb{R}^k \setminus H_{io}$, и точка $t \in M_{io}$ является граничной для A_i и A_j вместе с некоторой окрестностью в M_{io} . Рассмотрим множества \tilde{A}_i и \tilde{A}_j , полученные заменой \cap^* на \cap и членов H_{io} и H'_{io} на M_{io} в выражениях для A_i, A_j . Поскольку в выражениях для \tilde{A}_i, \tilde{A}_j остальные члены $H (k \neq io)$ остались без изменения, то либо $\tilde{A}_i \cap^* \tilde{A}_j$ (в топологии M_{io}) пусто, если некоторый член в $A_i = H_k$, а в $A_j = H'_k$, что противоречит условию существования t с окрестностью, либо соответственные члены равны, и, следовательно, теорема доказана.

Из этой теоремы непосредственно следует, что введенный нами в 2.9 класс телесных объектов размерности k из многообразия размерности k замкнут относительно регулярных (в топологии этого многообразия) теоретико-множественных операций.

2.12. Симплекс называется ориентированным, если установлен порядок задающих его вершин. Если новый порядок вершин получается четной перестановкой из исходного, то будем считать, что ориентация симплекса не меняется, если нечетной — меняется на противоположную [I]. Ориентацию будем обозначать как $c^k = \varepsilon(x_0, \dots, x_k)$, где $\varepsilon = \pm 1$.

Пусть задан ориентированный симплекс размерности k . Его граничные подсимплексы размерности $k - 1$ можно получить из исходного набора вершин вычеркиванием одной вершины. Введем индуцированную ориентацию граней следующим образом:
 $c^{k-1} = (-1)^i \varepsilon(x_0, \dots, \hat{x}_i, \dots, x_k)$, где i — индекс вычеркнутой вершины [I].

Введенное понятие ориентации симплекса тесно связано с алгебраической теорией.

финными преобразованиями многообразия, им порожденного. Действительно, если заданы два симплекса $C^k = (x_0^!, \dots, x_k^!)$ и $C^{k'} = (x_k^!, \dots, x_{k'}^!)$ и порожденные ими многообразия совпадают, то семейства векторов $\{x_i - x_0\}$ и $\{x_i^! - x_0^!\}$ можно рассматривать как два репера в этом многообразии. Тогда взаимная ориентация этих симплексов определяется знаком определителя матрицы аффинного преобразования этих реперов. Если определитель отрицательный – ориентации симплексов противоположны, если положительный – ориентации совпадают. Правило четности перестановки вершин соответствует правилу перестановки строк в определителе матрицы преобразования. Ориентация подсимплекса также связана с аффинными преобразованиями. Пусть $C^k =$

$$= (x_0, \dots, x_k) \text{ – симплекс. Точка } \tilde{x} = \frac{1}{k+1} \sum_{i=0}^k x_i \text{ является}$$

внутренней точкой симплекса. Нетрудно видеть, что вершины любой $k-1$ -мерной грани C^k и \tilde{x} образуют симплекс размерности $k \cdot C_{i'}^k$, где i' – номер вычеркнутой вершины в исходном наборе для получения $k-1$ -мерной грани. Рассмотрим реперы $\{x_i - x_0\}$ ($i = 1, \dots, k$) и $\{x_i - \tilde{x}\}$ ($i \neq i'$). Непосредственно выписав матрицу преобразования этих реперов, легко убедиться, что знак ее определителя равен $(-1)^{i'}$, т.е. правило определения индуцированной ориентации не произвольно и связано с разложением определителя на миноры. Рассмотренное понятие ориентации является существенно относительным, т.е. имеет смысл говорить только об одинаковой или разной ориентации симплексов.

Лемма. Пусть заданы два симплекса C^k и $C^{k'}$ с совпадающими порожденными многообразиями и одинаково ориентированные. И пусть у них имеется общая $k-1$ -мерная грань, причем ее индуцированная ориентация одна и та же. Тогда симплексы C^k и $C^{k'}$ имеют общую внутреннюю точку.

По условию леммы симплексы можно представить в виде $C^k = (x_0, x_1, \dots, x_k)$, $C^{k'} = (y_0, x_1, \dots, x_k)$. Из их одинаковой ориентации легко получить, что в разложении $y_0 = \sum_{i=0}^k \lambda_i x_i$, $\lambda_0 > 0$. Тогда точка $z = \frac{1-\varepsilon}{k} \sum_{i=1}^k x_i + \varepsilon y_0$ является внутренней для обоих симплексов при достаточно малом ε что легко проверяется подстановкой разложения y_0 .

Из этой леммы непосредственно следует, что если два симплекса расположены правильно, одинаково ориентированы и имеют общую грань, то либо они совпадают, либо индуцированные ориентации общей грани противоположны.

2.13. Будем говорить, что однородный комплекс ориентирован, если каждый симплекс, входящий в него, ориентирован. Объект будем называть ориентированным, если задающий его комплекс ориентирован. На множестве однородных ориентированных комплексов введем функцию "склеивания" s , которая каждому однородному ориентированному комплексу размерности k ставит в соответствие ориентированный однородный комплекс размерности k , но у которого нет совпадающих симплексов размерности k (см. замечание в 2.7), т.е. если несколько симплексов $\epsilon c_{i_1}^k, \dots, \epsilon c_{i_s}^k$ совпадают, как множества, то им сопоставляем один симплекс с ориентацией $\epsilon = \text{SIGN}(\sum_{j=1}^s \epsilon_{i_j})$ при $\epsilon \neq 0$ или просто их все удаляем. Границей ориентированного однородного комплекса размерности k назовем ориентированный однородный комплекс размерности $k-1$, полученный из исходного следующим образом. Пусть комплекс состоит из ориентированных симплексов $\{c_j^k\}$. Тогда множество ориентированных симплексов $s(\{c_j^{k-1}\})$, где c_j^{k-1} – граничный симплекс какого-либо симплекса c_j^k , взятый с индуцированной ориентацией, и будет границей исходного ориентированного однородного объекта.

2.14. Будем считать, что телесные объекты всегда ориентированы, точнее, что симплексы максимальной размерности задающих их комплексов ориентированы одинаково. Тогда ориентированной границей телесного объекта будем называть границу порождающего его однородного ориентированного комплекса. Нетрудно видеть (в силу леммы 2.12), что граница телесного объекта совпадает с его теоретико-множественной границей. Выше была установлена связь между ориентацией симплекса и аффинными преобразованиями многообразия, им порожденного, и их связь с ориентацией граничных подсимплексов. В силу того, что ориентация многообразия на единицу большей размерности может быть задана направлением вектора нормали, то ясно, что эти два

способа задания ориентации эквивалентны с точностью до конкретного значения (+ или -). Поэтому в дальнейшем будем считать ориентацию объекта положительной, что соответствует направлению нормали к граням "внутрь" объекта. Нетрудно видеть, что для случая \mathbb{R}^3 это соответствует тому, что основная система декартовых координат правосторонняя [3].

Таким образом, полученную выше границу телесного объекта можно рассматривать как совокупность ориентированных симплексов меньшей размерности, или симплексов с векторами нормали. Если, в свою очередь, в этой совокупности выделить замкнутые объекты меньшей размерности и задать их своей ориентированной границей (например, сами граничные симплексы), то получим иерархическое представление объекта, на каждом уровне неупорядоченную совокупность ориентированных объектов меньшей размерности, заданных своей ориентированной границей. Такое представление объекта является основным в пакете SPACE. Основной особенностью такого представления является то, что оно локально, т.е. граничные подобъекты не связаны между собой, и, как будет показано ниже, это свойство существенно упрощает и делает единообразными многие алгоритмы решения задач, перечисленных в п. I, например в отличие от алгоритмов, описанных в [4]. Более того, такое представление дает возможность сводить алгоритмы решения задач с трехмерными объектами к совокупности независимых задач с плоскими объектами и т.д. Рассмотренное представление относится к классу граничных представлений объекта, и способы их моделирования на ЭВМ рассматриваются, в частности, в [5]. Следует отметить, что аналогичный подход к моделированию объектов был предложен в работе [6] и там же был развит аппарат для решения задач конструирования объектов с использованием теоретико-множественных операций. Но отличительной особенностью предлагаемого здесь подхода является, во-первых, ориентация граничных элементов представления и, во-вторых, как следствие, появляется возможность реализации регулярных операций, что, с одной стороны, расширяет класс допустимых объектов, в частности, автоматически разрешаются неопределенности при совпадении границ, с другой стороны, регулярные операции являются более естеств-

венными аналогами операций под реальными объектами, чем "чисто" теоретико-множественные. Кроме того, автоматически появляется возможность очень просто решать расчетные задачи, такие, как вычисление площадей, объемов, центров тяжести и моментов инерции, что существенно для САПР. В работе [6]. Расчетные задачи не рассматриваются вообще, а предлагаемые алгоритмы теоретико-множественных операций не всегда определены (например, для разности).

2.15. Прежде чем переходить к рассмотрению алгоритмов решения задач над описанной выше моделью объекта, подытожим кратко то, о чём говорилось в п. 2.1-2.14, и приведем основные схемы алгоритмов.

Мы рассматриваем множество телесных объектов в R^n :

$$T = \{\emptyset, T^0, T^1, \dots, T^n\},$$

где T^k – множество телесных объектов размерности k . Каждое из множеств T^k , в свою очередь, разбивается на подклассы телесных объектов размерности k , лежащие в фиксированном многообразии, и, следовательно, эквивалентные с точностью до аффинного преобразования R^n . Было доказано, что каждый такой подкласс эквивалентен классу конечно-порожденных полупространствами ограниченных объектов из этого многообразия и, следовательно, замкнут относительно регулярных операций (естественно, с предположением, что $\emptyset \in T^1$).

На множестве T введем операцию пересечения объекта с аффинным многообразием следующим образом. Пусть $t^k \in T^k \subset T$, $t^k \subset M^k$, $M^{k-1} \subset M^k$, тогда

$$t = t^k + M^{k-1} = r_{k-1}(t^k \cap M^{k-1}),$$

где r_{k-1} – операция регуляризации в топологии M^{k-1} . Нетрудно видеть, что либо $t = \emptyset$, либо $t \in T^{k-1}$. Доказательство этого факта непосредственно вытекает из разложения объекта t^k в канонический вид в соответствии с п. 2.II. Из линейной алгебры [7] известно, что пересечение двух аффинных многообразий является также аффинным многообразием, принадлежащим

двум исходным, и любое аффинное многообразие размерности k может быть представлено как пересечение подходящих $n - k$ гиперплоскостей в \mathbb{R}^n (гиперплоскость — аффинное многообразие размерности $n - 1$ в \mathbb{R}^n). Отсюда очевидно, что для любого M^{k-1} можно найти такое M^{n-1} , что $M^k \cap M^{n-1} = M^{k-1}$, и, следовательно, в вышеприведенную формулу можно подставить гиперплоскость.

Мы также рассматриваем множество однородных комплексов

$$Q = \{\emptyset, Q^0, \dots, Q^n\},$$

где Q^k — множество однородных комплексов размерности k из \mathbb{R}^n .

По определению, если $q^k \in Q^k$, то $q^k = \{c_i^k\}$, где c_i^k — правильно расположенные симплексы. Предполагается, что существует соответствие $k : T \rightarrow Q$, которое каждому объекту из T сопоставляет порождающий его комплекс, естественно, не однозначно. В соответствии с п. 2.14 будем считать, что элементы Q ориентированы. Ориентацию будем обозначать чертой над символом. Таким образом, $K : T \rightarrow Q$, причем, если $t^k \in T^k$ и $K(t^k) = \bar{q}^k$, то k -мерные симплексы из \bar{q}_k ориентированы одинаково. На множестве \bar{Q} была введена операция "склеивания" $S : \bar{Q} \rightarrow \bar{Q}$, которая совокупности ориентированных симплексов размерности k из $\bar{q}^k \in \bar{Q}$ ставит в соответствие подмножество из этой совокупности следующим образом: если c_i^k и $c_j^k \in \bar{q}^k$ такие, что $c_i^k = c_j^k$, то либо они оба выбрасываются при $\varepsilon_i + \varepsilon_j = 0$, либо оставляется один из них.

На основе S строится функция $\Gamma : \bar{Q} \rightarrow Q$. Если $\bar{q}^k \in \bar{Q}^k$, то

$$\Gamma(\bar{q}^k) = S(\{c_{ij}^{k-1}\}) = \bar{q}^{k-1} \in Q^{k-1},$$

где $\{c_{ij}^{k-1}\} (j = 1, \dots, k)$ — множество $k - 1$ -мерных граней симплекса c_i^k , взятых с индуцированной ориентацией.

Каждому элементу множества T ставим в соответствие его ориентированную границу в виде набора ориентированных симплексов. Если $t^k \in T^k$, то

$$\Gamma_{t^k} \stackrel{\text{def}}{=} \Gamma(K(t^k)).$$

3. Алгоритмы решения некоторых задач

В предыдущем пункте было приведено определение рассматриваемых объектов, их представление и определены регулярные операции над ними. Здесь мы рассмотрим кратко несколько алгоритмов реализации этих операций и решения некоторых других задач.

3.1. Введем операцию "сечения" представления объекта $t^k \in \epsilon T^k$ ориентированным многообразием \bar{M}^{k-1} , лежащим в M_{t^k} , т.е. многообразием с нормалью к нему в M_{t^k} . Эта операция вводится рекурсивно. Для $k = 1$ можно считать, что задан вектор многообразия \bar{M}_{t^1} , равный v_1 , тогда

$$R_L(\Pi_{t^1}, M^o) \stackrel{\text{def}}{=} S([\Pi_{t^1}, M^o]_1),$$

где $[\Pi_{t^1}, M^o]$ – представление точки M^o в виде ϵM_o , которое для фиксированного i есть

$$\epsilon = \begin{cases} f(L, \epsilon_i, \epsilon_o), & \text{если } t_i^o = M^o, t_i^o = \epsilon_i t_i^o, M^o = \epsilon_o M^o, \\ \epsilon_i \cdot \text{SIGN}(t_i^o - M^o, v_1), & \end{cases}$$

где $(,)$ – скалярное произведение в R^n .

Нетрудно видеть, что результатом этой операции является либо представление точки M^o , либо пустое множество, причем, если M^o – граничная точка для t^1 , то, в зависимости от $f(L, \epsilon_i, \epsilon_o)$, она будет рассматриваться различным образом. Об этой функции $f = \pm 1$ будем говорить дальше.

Для произвольного k операция определяется следующим образом:

$$R_L(\bar{\Pi}_{t^k}, \bar{M}^{k-1}) \stackrel{\text{def}}{=} S([\bar{\Pi}_{t^{k-1}}, \bar{M}^{k-1}]_1),$$

где

$$[\bar{\Pi}_{t^{k-1}}, \bar{M}^{k-1}] = \begin{cases} f(L, \epsilon_i, \epsilon_o) \Pi_{t^{k-1}}, & \text{если } M_{t^{k-1}} = \bar{M}_{t^{k-1}}, \\ M_{t^{k-1}} = \epsilon_i M_{t^{k-1}}, \bar{M}^{k-1} = \epsilon_o \bar{M}^{k-1}; & \end{cases}$$

$$\left\{ \begin{array}{l} S(R_{L_1}(\Pi_{t_1^{k-1}}, M_1^{k-2})), \\ \text{где } M_1^{k-2} = M_{t_1^{k-1}} \cap M^{k-1}, \end{array} \right.$$

т.е. операция определяется рекурсивно, а $[,]$ - внутренняя часть алгоритма "сечения".

Введенная операция является аналогом операции " $+$ ", введенной ранее, но уже для представлений объектов. Параметр операции введен по следующей причине. В операции " $+$ " существует операция регуляризации, которая автоматически разрешает неопределенности при совпадении границ объектов, поскольку "имеет дело" с окрестностями. В случае граничного представления мы лишены такой возможности, и для алгоритмов введен специальный "инструмент" в виде параметров L , L_1 . Смысл его следующий: если мы рассмотрим пересечение объекта с многообразием, то возможно совпадение многообразия с многообразиями граничных элементов (элементов граничного представления), и можно либо включать этот элемент в результат пересечения, либо не включать, причем в зависимости от соотношения ориентаций многообразия и граничного элемента. Если разбить многообразие на части, лежащие внутри объекта, на его границе и вне объекта, и обозначить их соответственно M^+ , M^0 , M^- , а M^0 разбить на две части в зависимости от совпадения ориентаций $M^0 = M^{0+} \cup M^{0-}$, то возможны четыре случая "пересечения":

$$M^+, M^+ \cup M^{0+}, M^+ \cup M^{0-}, M^+ \cup M^-,$$

т.е. алгоритм $R_L (L = 1, \dots, 4)$ реализует четыре возможных случая пересечения объекта с многообразием и получает в результате корректный объект на единицу меньшей размерности. Следовательно, можно работать с ним, как с объектом размерности $k - 1$.

Этот алгоритм является базовым в пакете SPACE. Алгоритм "склеивания" S не представляет никакой сложности и связан с реализацией представления объекта на ЭВМ, а в одномерном случае эквивалентен упорядочению точек прямой. Для случая R^3 рекурсию можно опустить, модифицировав внутреннюю часть алго-

ритма "сечения", т.е. вычисления [,]. В реализации, осуществленной автором, имеет место только однократная рекурсия при переходе от R^3 к R^2 .

Одним из преимуществ того, что алгоритм – единый почти для всех операций, является сосредоточение всех вычислительных действий в одном месте, в этом алгоритме, что позволяет аккуратно проследить возможные ошибки, связанные с машинной точностью. В этом алгоритме таких мест всего два, а именно: вычисление многообразия, являющегося пересечением двух заданных, и оценка знака скалярного произведения. Поскольку структура представления объекта такова, что вершины одни и те же, то это оказалось легко проследить и производить вычисления с некоторой фиксированной точностью. При нахождении пересечения отрезка и прямой, а также отрезка и плоскости вычисляются расстояния от концевых точек отрезка до многообразия, точнее, пропорциональные им величины; и только в случае несовпадения знаков по этим величинам, предварительно округленных до нуля (если это необходимо), вычисляются координаты точек пересечения. Такой способ, во-первых, экономичнее и, во-вторых, позволяет очень просто обходить возможные машинные ошибки. Хотя он чисто эвристический, но дает хорошие результаты и впоследствии, возможно, будет обоснован.

На основе этого алгоритма очевидно решаются задачи штрихования области, получения сечений, определения принадлежности точки объекту.

3.2. Реализация регулярных теоретико-множественных операций над представлениями автоматически вытекает из описанного в п. 3.1. алгоритма и базируется на нем. Если разбить граничные элементы (объекты меньшей размерности из граничного представления) одного объекта на части, лежащие вне, внутри и на границе другого объекта, путем подходящего разбиения многообразия, порождающего каждый элемент по операции пересечения, а затем применения какой-либо регулярной операции к граничному элементу и результату пересечения, то получим набор различных элементов, которые можно сгруппировать следующим образом. Пусть у нас есть два объекта A и B , тогда выделяются

следующие группы: $\Gamma_A^+, \Gamma_A^{0+}, \Gamma_A^{0-}$, Γ_A^- и $\Gamma_B^+, \Gamma_B^{0+}, \Gamma_B^{0-}$, Γ_B^- ; обозначения очевидны. Тогда справедливы следующие соотношения:

$$\Gamma_{(A \cap B)} = \{\Gamma_A^+, \Gamma_A^{0+}, \Gamma_B^+\} = \{\Gamma_A^+, \Gamma_B^{0+}, \Gamma_B^+\},$$

$$\Gamma_{(A \cup B)} = \{\Gamma_A^-, \Gamma_A^{0+}, \Gamma_B^-\} = \{\Gamma_A^-, \Gamma_B^{0+}, \Gamma_B^-\},$$

$$\Gamma_{(A \setminus B)} = \{\Gamma_A^-, \Gamma_A^{0-}, -\Gamma_B^+\} \quad \{\Gamma_A^-, \Gamma_B^{0-}, -\Gamma_B^+\},$$

где $-\Gamma$ обозначает противоположную ориентацию.

Первое из этих соотношений доказывается на основе приведения к каноническому виду, а остальные – на основе соотношений $A \cup B = (A \setminus B) \cup A$, где $A \cap^* (A \setminus B) = \emptyset$ и $A \setminus B = A \cap^* B$, хотя дополнение, вообще говоря, не определено, но для нашего класса представлений его можно рассматривать как $-\Pi_B$.

Таким образом, алгоритм каждой из операций представляет собой последовательный процесс построения части границы одного объекта, а затем, симметрично, для другого. Результат получается также в виде граничного представления. Все алгоритмы построены по одной и той же схеме.

3.3. При использовании пакета при решении сложных задач, например, в САПР, существенное значение приобретает контроль корректности представления объекта. Мы рассматриваем корректность в двух отношениях. Во-первых, с точки зрения задания координат-вершин объекта, т.е., грубо говоря, чтобы они не "слипались" при обработке на ЭВМ. Эта задача решается более или менее просто, путем непосредственного сравнения. Во-вторых, с точки зрения "правильности" задания структуры объекта, т.е. иерархии составляющих его граничных подобъектов, а также их ориентации. Оказывается, для нашего класса объектов и их представлений справедливо следующее: представление объекта корректно, т.е. соответствует некоторому телесному объекту тогда и только тогда, когда пересечение его с многообразием, порожденным его граничным элементом, корректно. Пересечение рассматривается в смысле алгоритма из п. 3.1.

В общем случае оно конструктивно доказывается из алгоритма пересечения и разложения в канонический вид. Более того, процесс сечения объекта его же границей дает канонический вид объекта и, следовательно, триангуляцию, т.е. один из порождающих его комплексов.

3.4. Расчетные задачи, такие как вычисление объемов, площадей и центров тяжести, решаются совершенно очевидно из представления объекта. Так, например, объем объекта размерности 3 вычисляется по формуле

$$V = \frac{1}{3} \sum (v_i^2, x_i),$$

где сумма берется по всем граничным подобъектам, x_i – произвольная точка из многообразия подобъекта, v_i^2 – вектор площади. В правосторонней системе координат объем положительно ориентированного объекта положителен. Вектор площади вычисляется аналогично:

$$V^2 = \sum v_i^1 \cdot x_i \text{ и } S^2 = \frac{1}{2} |V^2|,$$

где сумма берется по его граничным подобъектам, v_i^1 – вектор длины, равный

$$V^1 = \sum_{i=1}^k \epsilon v_i,$$

где $\{\epsilon v_i\}_1^k$ – ориентированная граница одномерного объекта.

Аналогичным образом вычисляются другие интегральные характеристики, что следует непосредственно из векторного анализа [3].

3.5. Интересной задачей является задача построения проекций с удалением невидимых поверхностей или граней. Для ее решения, в рамках изложенного подхода, предлагается следующий алгоритм. Если рассмотреть алгоритм реализации теоретико-множественных операций и внести в представления объекта параметр α (например, задавать ориентацию не ± 1 , а $\pm \alpha$), а для

другого объекта β и $\alpha \neq \beta$, то легко внести в алгоритм "пересечения" представления объекта многообразием, учет их значений и, в зависимости от их соотношения, получать представление некоторого, вообще говоря, нетелесного однородного объекта, который является как бы наложением одного объекта на другой. Эта операция сама по себе очень интересна для моделирования неоднородных объектов. Если теперь рассмотреть проекции граней трехмерного объекта на плоскости и вместо фиксированного α каждой вершине сопоставить, например, расстояние от точки наблюдения, то, в силу линейности границ, легко заметить, что можно использовать тот же алгоритм наложения или проникновения. Таким образом, последовательное применение этого алгоритма ко всем проекциям граней дает изображение объекта с удалением невидимых граней в виде представления некоторого однородного полиэдра на плоскости.

В соответствии с описанным подходом были реализованы алгоритм сечения и алгоритмы теоретико-множественных операций в виде набора Фортран-программ. Предложенный в работе подход можно рассматривать как попытку формализации методов конструирования, обработки и представления геометрической информации в ЭВМ. В настоящее время этот подход развивается и реализуется в пакете SPACE.

Автор благодарен А.М.Мацокину за постановку задач, постоянное внимание и обсуждение вопросов, затронутых в данной работе.

Л и т е р а т у р а

1. Понтрягин Л.С. Основы комбинаторной топологии. - М.: Наука, 1976.
2. Tilove R.B. Set membership Classification. - A Unified Approach to Geometric Intersection. - IEEE Trans. Comput., 1980, v.29, no. 10, p. 874-883.
3. Корн Г.А., Корн Т.М. Справочник по математике для научных работников и инженеров. Определения, теоремы, формулы. - М.: Наука, 1978.

4. Ротков С.И. Алгоритмы теоретико-множественных операций над образами геометрических тел. - В сб.: Автоматизация поиска новых технических решений. Горький, 1980, с.143-151.
5. Michael A. Wesley Construction and Use of Geometric Models. - In: Lect Notes Comput Sci., 1980, v. 89, p.79-136.
6. Кислюк О.С. Синтез геометрических объектов. Обобщенные многогранники: Препринт №30. - Владивосток, 1982. - 22 с.- В надзаг.: ДВНЦ АН СССР, Ин-т автоматики и процессов упр.
7. Мальцев А.И. Основы линейной алгебры. - М.: Наука, 1970.

В.П.Потапов, В.Н.Вылегжанин, Э.И.Витковский

СИСТЕМА МАШИННОЙ ГРАФИКИ ДЛЯ АВТОМАТИЗИРОВАННОГО
ПРОЕКТИРОВАНИЯ ЭЛЕМЕНТОВ ШАХТНЫХ ПОЛЕЙ

При решении задач автоматизированного проектирования элементов шахтных полей (горных выработок, их сопряжений, определений на плане элементов залегания пласта и т.п.) требуется отображать большое количество самой разнообразной и разнородной информации в графической форме. До настоящего времени автоматизация решения задач подобного рода ограничивалась в основном автоматическим вычерчиванием изолиний некоторого показателя в пределах шахтного поля. Однако в связи с появлением банков данных, разработанных для решения задач ОАСУ-уголь и САПР-уголь, возникла необходимость в достаточно универсальной графической системе, которая бы, с одной стороны, могла работать с уже имеющимися базами данных, а с другой стороны, помогала бы решать большинство задач по геометризации элементов шахтных полей, используя информацию, получаемую при предварительной обработке геологоразведочных данных.

Анализ известных графических систем (не носящих интерактивного характера) [1 - 5] позволил остановиться на двух наиболее распространенных в настоящее время: ГРАФОРе и СМОГе. При этом в качестве базовой графической системы выбрана система СМОГ, разработанная в ВЦ СО АН СССР [2], что объясняется в основном несколькими причинами: удачной системой управления графическим выводом, использующей большинство применяемых устройств графического вывода; наличием архива рисунков на магнитном диске; возможностью подключения графических

устройств без изменения среды, в которой работает система машинной графики.

Недостаточный объем функциональных программ заставил обратиться к системе ГРАФОР, которая к настоящему времени имеет большой набор программ, практически для большинства возможных приложений. Стыковка систем была осуществлена на уровне программ PLOT и MOVE, путем их переключения на нижний уровень СМОГа. Изменения в небольшой степени коснулись программ инициализации пакета и программ открытия страницы. Благодаря такому изменению пользователь может работать как с ГРАФОРом, так и со СМОГом, или с той или другой системой раздельно, однако вывод будет всегда происходить в архив СМОГ. Применяя данный подход, можно осуществить стыковку системы СМОГ с другими графическими системами, например, ПАД-ЕС [3], программы которой были использованы нами для расширения системы СМОГ-ГРАФОР.

Большая часть чертежно-графической документации в горном деле изображается с помощью изолиний, поэтому был проанализирован ряд программ их построения, как автономных, так и взятых из отдельных графических систем, которые затем были опробованы на тестовых примерах по реальным горно-геологическим данным. Исходя из результатов тестирования, предпочтение было отдано комплексу программ построения изолиний из системы СМОГ, адекватно отображающим горно-геологическую обстановку.

Одним из вопросов во многих системах машинной графики является выбор метода аппроксимации для описания различных поверхностей. Несмотря на наличие большого количества работ по данному вопросу [6 - 8], трудно дать конкретные рекомендации по решению практических задач. При описании угольных пластов задача усложняется еще и тем, что данные (в большинстве своем) собираются на редкой хаотической сетке. Нам удалось достичь хороших результатов, применяя для аппроксимации метод сплайн-функции Грина, что дало возможность решать с помощью вышеописанной системы СМОГ-ГРАФОР не только типичные задачи машинной графики, но и ряд технологических задач (например, прогноза нарушенности).

При решении задач горно-инженерной графики с использованием системы СМОГ-ГРАФОР пришлось столкнуться с рядом труднос-

тей, что, в основном, было обусловлено отсутствием структуризации графических объектов и малым объемом программ геометрических вычислений. Это послужило причиной подключения к системе СМОГ-ГРАФОР такого известного и широко распространенного пакета программ для геометрических вычислений, как ФАП-КФ [4]. Адаптация системы ФАП-КФ была выполнена через программы первого уровня ГРАФОРа и через аналогичные программы СМОГа. Несмотря на кажущуюся громоздкость полученного комплекта графических программ, их применение не вызывает никаких осложнений со стороны операционной системы, при этом удается работать на машинах с объемом ОП в 256 Кбайт. За счет использования ФАП-КФ удалось в значительной степени решить таких задач, как подсчет площадей контуров, нахождение их периметров, центров тяжестей и т.п., что необходимо для решения, например, такой важной и часто встречающейся в практике задачи, как подсчет запасов. На языке операторов ФАП-КФ эта задача решается довольно просто. Кроме этого, существование таких объектов, как геометрические комплексы и составные графические объекты, позволило создавать библиотеки типовых графических объектов, которые в дальнейшем используют при решении различных задач.

Вышеописанная система работает в комплексе с программами предварительной обработки информации [7] и с базой данных "Горно-геологические условия", спроектированной с помощью СУБД ОКА, и позволяет значительно автоматизировать процессы решения многих задач АСУ и САПР. Развитие данной системы предполагается вести как за счет замены системы СМОГ на более совершенную, обеспечивающую интерактивную и распределенную работу [9], так и путем создания специализированных пакетов для решения конкретного класса задач горно-инженерной графики с помощью инструментальных систем типа МЕМО [10], ПРИЗ [11].

В заключение авторы считают своим приятным долгом поблагодарить В.Н.Вавро и сотрудников ВЦ СО АН СССР А.М.Мацокина, В.А.Василенко, С.А.Упольникова, В.Г.Сиротина за большую помощь и советы при выполнении данной работы.

Л и т е р а т у р а

1. Баяковский Ю.М., Михайлова Т.Н., Мишакова С.Т., ГРАФОР: комплекс графических программ на форTRANе: Препринт № 41, - М., 1972. - В надзаг.: ИПМ АН СССР.
2. Математическое обеспечение граfolостроителей. СМОГ, Еуро-вень: Инструкция по программированию / Под ред. Ю.А.Кузнецова. - Новосибирск, 1976.
3. Зозулевич Д.М. Машинная графика в автоматизированном проектировании. - М.: Машиностроение, 1976.
4. Горелик А.Г. Автоматизация инженерно-графических работ с помощью ЭВМ. - Минск: Вышшая школа, 1980.
5. Принс М.Д. Машинная графика и автоматизация проектирования. - М.: Советское радио, 1975.
6. Лоран П.-Ж. Аппроксимация и оптимизация. - М.: Мир, 1975.
7. Ковалков А.В. Функции Грина и сплайн-аппроксимации в многомерных областях: Препринт № 70. - Новосибирск, 1980. - В надзаг.: ВЦ СО АН СССР.
8. Завьялов Ю.С., Квасов Б.И., Мироновиченко В.Л. Методы сплайн-функций. - М.: Наука, 1980.
9. Проблемы машинной графики: Сб. статей. - Новосибирск: Б.и., 1982.
10. Приз Е.С. Введение в систему программирования. Таллин, 1979.
- II. Пруден Ю.И., Маркуш А. Мониторная система САПР в машиностроении. - В сб.: Вычислительная техника социалистических стран. М., 1982.

МАШИННАЯ ГРАФИКА И ЕЕ ПРИЛОЖЕНИЯ

(Сборник научных трудов)

Под редакцией Александра Михайловича Мацокина

Ответственный за выпуск В.Г.Сиротин

Редактор Л.А.Карев

Обложка художника Д.К.Томашевского

Подписано в печать 19.07.83 МН 17653

Формат бумаги 60x90 1/16 Объем 10 уч.-изд.л., 10.1 п.л.
Тираж 500 экз. Цена 60 коп. Заказ № 366

Ротапrint ВЦ СО АН СССР, Новосибирск 90