

Приведенные материалы свидетельствуют, что использование визуализации в MATLAB формы исходных тел и пересекающихся поверхностей значительно упрощает решение рассмотренных задач и дает реальное представление об их пространственной структуре. С учетом того, что студенты IT специальностей изучают выскоуровневую графику MATLAB в первом семестре, а с кратными интегралами знакомятся во втором семестре, предлагаемый подход понашему мнению целесообразно использовать в учебном процессе.

Список использованных источников:

1. Дубовец В. Д. Построение графических моделей в среде MATLAB: учеб.-метод. Пособие / В. Д. Дубовец, В. А. Столер, В. Ф. Бондаренко. – Минск: БГУИР, 2015. – 70 с.
2. Карпилова О. М. Кратные интегралы (задачи и упражнения): метод. Указания / сост. О. М. Карпилова. – Самара: Изд-во Самар. Гос. Аэрокосм. ун-та, 2008. – 44 с.

ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ ГОСТ 19.701-90 В КУРСЕ ИКГ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Дурдыев М.

Рожнова Н.Г. – преподаватель

В ГОСТе 19.701-90 «Схемы алгоритмов, программ, данных и систем» определены символы, предназначенные для использования в документации по обработке данных, и приведено руководство по условным обозначениям для применения их в:

- 1) схемах данных;
- 2) схемах программ;
- 3) схемах работы системы;
- 4) схемах взаимодействия программ;
- 5) схемах ресурсов системы.

Схемы данных отображают путь данных при решении задач и определяют этапы обработки, а также различные применяемые носители данных. Схемы программ отображают последовательность операций в программе. Схемы работы системы отображают управление операциями и поток данных в системе. Схемы взаимодействия программ отображают путь активаций программ и взаимодействий с соответствующими данными. Каждая программа в схеме взаимодействия программ показывается только один раз (в схеме работы системы программа может изображаться более чем в одном потоке управления). Схемы ресурсов системы отображают конфигурацию блоков данных и обрабатывающих блоков, которая требуется для решения задачи или набора задач.

Схемы алгоритмов, программ, данных и систем состоят из имеющих заданное значение символов, краткого пояснительного текста и соединяющих линий.

Схемы могут использоваться на различных уровнях детализации, причем число уровней зависит от размеров и сложности задачи обработки данных. Уровень детализации должен быть таким, чтобы различные части и взаимосвязь между ними были понятны в целом.

Схема программы состоит из:

- 1) символов процесса, указывающих фактические операции обработки данных (включая символы, определяющие путь, которого следует придерживаться с учетом логических условий);
- 2) линейных символов, указывающих поток управления;
- 3) специальных символов, используемых для облегчения написания и чтения схемы.

Внутри символов необходимо делать поясняющую запись.

В пункте 4.1.4 ГОСТа говорится:

«Минимальное количество текста, необходимого для понимания функции данного символа, следует помещать внутри данного символа. Текст для чтения должен записываться слева направо и сверху вниз независимо от направления потока». Так как объем текста, помещаемого внутри символа, превышает его размеры, то будет использован символ комментария. Символ «Комментарий» используют для добавления описательных комментариев или пояснительных записей в целях объяснения или примечаний. Пунктирные линии в символе комментария связаны с соответствующим символом или могут обводить группу символов. Текст комментариев или примечаний должен быть помещен около ограничивающей фигуры.

Все символы соединяются соединительными линиями в соответствии с пунктом 4.2.1: «Потоки данных или потоки управления в схемах показываются линиями. Направление потока слева направо и сверху вниз считается стандартным».

В случаях, когда необходимо внести большую ясность в схему (например, при соединениях), на линиях используются стрелки. Если поток имеет направление, отличное от стандартного, стрелки должны указывать это направление.

Рассмотрим схему алгоритма решения задачи по построению линии пересечения сферы с призмой (рис. 1). Условия задания: ввод исходных данных – с клавиатуры; вывод результатов – печатная копия.

Задание выполняется в программе VISIO.

Первое действие алгоритма – построение исходной фигуры. Данное действие программы является процессом, следовательно устанавливается символ «Предопределенный процесс».

Символ «Предопределенный процесс» отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, модуле).

Следующее действие программы – определение характерных точек. Данное действие является также предопределенным.

Затем программа должна определить точки на линии пересечения сферы с призмой. Так как для построения необходимо определить точки в нескольких секущих λ_n , то дальнейшие действия программы выполняются в цикле. Данные действия являются процессом.

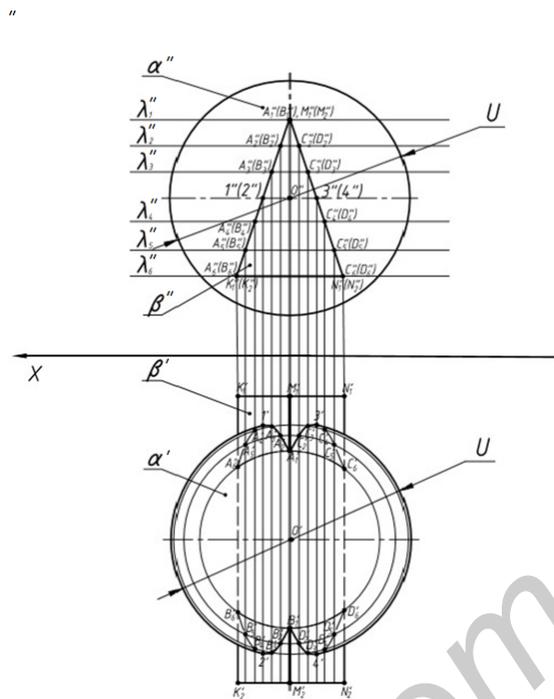


Рис.1- Пример геометрической задачи

Символ «Процесс» отображает функцию обработки данных любого вида (выполнение определенной операции или группы операций, приводящее к изменению значения, формы или размещения информации или к определению, по которому из нескольких направлений потока следует двигаться).

В первом символе процесса описываются начальные условия входа в цикл ($i=1$).

Последующие символы:

- построение секущей λ_n ;
- построение прямой линии из точки пересечения секущей λ_n с линией пересечения сферы с призмой к окружности;
- построение точек A_i, B_i, C_i, D_i ;
- условие для перехода к следующей секущей.

Следующий устанавливаемый символ – «Решение». Символ «Решение» отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа. Соответствующие результаты вычисления могут быть записаны по соседству с линиями, отображающими эти пути.

Внутри символа записывается условие необходимое для выхода из цикла.

В ветвь «Нет» (условие не выполнено) устанавливается символ «Соединитель».

Символ «Соединитель» отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте. Соответствующие символы-соединители должны содержать одно и то же уникальное обозначение. Второй символ «Соединитель» устанавливается в место, с которого цикл должен повториться.

Если условие выполнено – то цикл прерывается, и выполняется вторая ветвь: строится искомая линия пересечения. Устанавливается символ «Процесс».

Далее необходимо визуально проконтролировать результаты работы программы. Соответственно устанавливается символ «Дисплей».

Символ «Дисплей» отображает данные, представленные в человекочитаемой форме на носителе в виде отображающего устройства (экран для визуального наблюдения, индикаторы ввода информации).

По условию задания необходимо создать печатную копию результатов работы программы. Соответственно устанавливается символ «Бумажная лента». Символ «Бумажная лента» отображает данные, представленные на носителе в виде бумажной ленты.

На этом работа программы завершается – соответственно устанавливается символ «Терминатор», и записывается в него «Остановка».

Далее необходимо пронумеровать все символы в соответствии с пунктом 4.1.5. и скорректировать размер алгоритма принимая во внимание пункт 4.1.2. ГОСТа. «... В схемах может использоваться идентификатор символов. Это связанный с данным символом идентификатор, который определяет символ для использования в справочных целях в других элементах документации (например, в листинге программы). Идентификатор символа должен располагаться слева над символом». «4.1.2. Символы в схеме должны быть расположены равномерно. Следует придерживаться разумной длины соединений и минимального числа длинных линий». На этом выполнение задания завершается.

Данный пример не описывает в полном объеме ГОСТ19.701-90, а лишь позволяет познакомиться с его практическим применением на примере решения конкретной геометрической задачи.

Также следует отметить особенности применения некоторых символов описанных в данном ГОСТе:

- к запоминающим устройствам с прямым доступом относятся: накопители на жестких или гибких магнитных дисках; оперативные (ОЗУ) или постоянные (ПЗУ) запоминающие устройства компьютера;
- к запоминающим устройствам с последовательным доступом относятся: полупроводниковые (так называемые «Flash – накопители»), ленточные (так называемые «кассеты»), оптические (CD, DVD и др.) носители информации.

Список использованных источников: A_0, B_0, \dots, E_0

1. ГОСТ 19.701-90 (Исо 5807-85) Схемы алгоритмов, программ, данных и систем.

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ КРАШ-ТЕСТОВ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Скороход А.А., Коршунов Р.А.

Павлович В.В. – преподаватель

Современное компьютерное моделирование давно сделало шаг вперед в своем развитии, позволяя с феноменальной точностью воссоздать любое явление или процесс, будь то построение дома или краш-тест, моделирование которого является основной темой нашего доклада.



Рис.1 - Моделирование легкового автомобиля для краш-тестов

Краш-тест — это прежде всего испытание дорожных автомобилей на безопасность.

Появившись в 30-е года 20 века как необходимость решения проблемы безопасности пассажиров автомобиля, сегодня успешное прохождение краш-теста - это важная характеристика безопасности автомобиля, без которой его (автомобиль) никто на дорогу не выпустит. С каждым годом стоимость производства автомобилей неумолимо растет, вследствие чего производство прототипов автомобилей становится достаточно затратным. Для примера, стоимость современной экспериментальной модели автомобиля составляет сумму до 2 млн. долларов, стоимость одного краш-теста - от 150 до 200 тысяч долларов. И все бы ничего, но для успешного запуска производства автомобиля необходимо провести несколько сотен краш-тестов. Не нужно быть математиком для понимания того, что это очень невыгодно производителям. Поэтому с развитием компьютерных технологий и средств компьютерного моделирования в процессе создания автомобиля появилась новая фаза - компьютерное моделирование краш-теста. Его стоимость не превышает 7 тысяч долларов, а высокая степень точности позволяет быстро определить слабые стороны конструкции автомобиля и внести изменения в параметры модели. Если система сложна, а требуется проследить за каждым ее элементом, то на помощь могут прийти компьютерные имитационные модели. На компьютере можно воспроизвести последовательность временных событий, а потом обработать большой объем информации. На компьютере с помощью моделирования можно разбить несчислимое количество автомобилей, можно рассмотреть несколько вариантов конструкций, чтобы определить, какая подходит, а какая никуда не годится. Таким образом, можно подобрать наиболее оптимальную конструкцию с точки зрения безопасности кузова. Один из методов краш-теста (фронтальный) изображен на рис.2, подробнее обо всех методах, можно увидеть презентации.