

УДК 515(076.5)

Решение комплексных задач по начертательной геометрии средствами графической системы AutoCAD

Вуколов К.А., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы автоматического управления»*

Научный руководитель: Суфляева Н.Е., доцент, к.т.н.

Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана

bauman@bmstu.ru

Введение.

Начертательная геометрия в ряду общеинженерных дисциплин занимает особое место – она является лучшим средством развития у человека пространственного воображения, без которого немислимо никакое инженерное творчество. Начертательная геометрия обеспечивает теоретическую базу для составления чертежа, являющегося интернациональным языком технически грамотных людей.

На протяжении более чем двухсотлетнего периода преподавания начертательной геометрии в технических вузах задачи с пространственными фигурами решались по их проекционным изображениям на плоскости. Для облегчения понимания учащимися формы трехмерных фигур и их взаимного расположения в пространстве применялись наглядные пособия в виде макетов, плакатов с аксонометрическими изображениями и другие имеющиеся в прежние времена средства визуализации.

В настоящее время для обучения графическим дисциплинам мощным подспорьем стали современные системы автоматизированного проектирования (САПР), дающие возможность в интерактивном режиме выполнять любые трехмерные построения, получая реалистичное отображение событий в пространстве.

Внедрение компьютерных технологий в учебный процесс позволяет применить САПР в обучении начертательной геометрии и поднять преподавание этой традиционной дисциплины на новый современный уровень.

Цель работы.

Целью данной работы является решение пространственных задач по начертательной геометрии средствами графической системы AutoCAD, а также поиск

преимуществ программного способа решения задач перед ручным способом, а также автоматизация получения проекций фигур с помощью языка программирования AutoLISP.

Содержание работы.

Данная работа включает в себя решение задач по начертательной геометрии с помощью среды AutoCAD и программу по автоматизации проецирования набора прямых на основные плоскости проекций с помощью встроенного в среду языка программирования AutoLISP.

Язык AutoLISP.

Графический язык программирования AutoLISP является расширением языка программирования LISP. LISP- это язык высокого уровня, ориентированный на обработку списков, который выбран в качестве базового потому, что графические примитивы (начиная с точки), блоки, наборы примитивов и блоков удобно представляются в виде списков.

В составе системы AutoCAD поставляется интерпретатор языка AutoLISP. Если при генерации AutoCADa интерпретатор AutoLISPa был подключен, то он загружается в оперативную память после запуска графического редактора ACAD и доступен в течение всего сеанса работы с ACAD.

Таким образом, графический редактор ACAD и интерпретатор языка AutoLISP представляют собой единую систему: любая функция AutoLISPa может быть вызвана из графического редактора, и любая команда редактора может быть использована в программе на AutoLISPe. Возможности применения AutoLISPa весьма широки и разнообразны.

Характерные классы применения языка AutoLISP.

1. Программирование чертежей с параметризацией.

Создаётся программа, позволяющая при каждом обращении к ней формировать новый чертёж, отличающийся от предыдущих чертежей, построенных этой же программой, размерами, а также, возможно, и топологией: могут появиться новые элементы обогащения, сечения, измениться текстовая часть чертежа и т.д. Время получения чертежа с помощью такой программы может быть в десятки раз меньше времени, необходимого для его создания с помощью редактора ACAD, и, что не менее

важно, получить чертёж сможет любой конструктор, даже мало знакомый с командами ACAD.

2. Создание и использование графических баз данных.

Если накоплено большое количество чертёжных файлов, программ на AutoLISPе, соответствующих чертёжным фрагментам, деталям, узлам, то их можно в некотором смысле считать графической базой данных. Программы на AutoLISPе в сочетании с пользовательскими меню могут организовывать просмотр, поиск, подключение к объектам их частей и т.п. Тогда работа конструктора в системе AutoCAD будет сводиться к поиску нужных объектов (сборочных единиц, деталей) или частей чертежа, обращению к соответствующим LISP-программам и ответам на вопросы этих программ. Есть ещё одно очень важное обстоятельство: хранение графических данных в виде набора программ на AutoLISPе даёт возможность в десятки и сотни раз сократить требуемую память на магнитном диске по сравнению с памятью, необходимой для хранения чертёжных файлов ACAD, так как, во-первых, одна программа позволяет получить не один, а множество чертежей, во-вторых, текст программы на AutoLISPе занимает на порядок меньше памяти, чем файл, который может быть получен в результате работы этой программы.

3. Анализ и (или) автоматическое преобразование изображений.

Программа на AutoLISPе может воспринимать чертёж на экране, построенный с помощью графического редактора и обсчитывать его. Программа также может быстро осуществить преобразование изображения, на которое при работе в графическом редакторе пришлось бы затратить значительное время, например:

- заменить все вставки одного типа на вставки другого типа из какого-либо чертёжного файла;
- перенести все объекты с одного слоя на другой;
- повернуть все блоки на заданный угол - каждый относительно своей базовой точки и т.п.

Программа по автоматизации проецирования прямых на основные плоскости проекций.

При решении задач по начертательной геометрии возникла необходимость в каждом из чертежей строить проекции фигуры на основные плоскости проекций. В связи

с этим возникла идея автоматизировать этот процесс для уменьшения времени работы над чертежом. Для этого был изучен язык AutoLISP и написана соответствующая программа.

Принцип работы программы:

Для построения проекции точки на любую из плоскостей проекций достаточно знать три координаты этой точки. При этом проекция получается при обнулении соответствующей координаты. Например, имея точку $P(x, y, z)$ можно получить ее проекцию на плоскость (XY) при обнулении координаты z : $P'(x, y, 0)$.

Для получения проекции отрезка достаточно иметь проекции начала и конца отрезка, а значит, достаточно иметь координаты начала и конца отрезка.

Для получения координат начала и конца отрезка в языке AutoLISP нужно получить список свойств соответствующего примитива.

Так как программа направлена на работу с набором отрезков, то в программу поступает набор примитивов:

```
(setq lineList (ssget))
```

Данная строка программы предлагает пользователю выбрать несколько примитивов (функция `ssget`), и после нажатия клавиши `Enter` список примитивов помещается в переменную `lineList`.

Далее необходимо получить из списка каждый отдельный примитив. Для этого была использована функция `ssname`, которая в качестве первого параметра принимает список примитивов, а в качестве второго – индекс нужного примитива в списке (индексирование начинается с 0):

```
(ssname lineList i)
```

Здесь в переменной `i` хранится индекс примитива. Для последовательного доступа к каждому примитиву в списке был использован цикл `while`, вызванный следующим образом:

```
(setq i 0)
(while (< i (sslength lineList))
  (setq lineObj (ssname lineList i))
  ;Здесь производится работа с примитивом
  (setq i (+ i 1))
)
```

В данном коде до цикла переменной `i` присваивается 0 (начало индексирования), затем пока не закончился список (функция `sslength` позволяет получить длину списка

примитивов) производится работа с примитивами. Для начала примитив помещается в переменную lineObj функцией ssname.

Для дальнейшей работы с примитивом необходимо получить список его свойств. Для этого в языке AutoLISP предусмотрена функция entget. Функция возвращает следующий список (пример):

```
((-1 . <Entity name: 7ffff705bc0>)
  (0 . "LINE")
  (330 . <Entity name: 7ffff7039f0>)
  (5 . "234")
  (100 . "AcDbEntity")
  (67 . 0)
  (410 . "Model")
  (8 . "0")
  (100 . "AcDbLine")
  (10 0.0 0.0 0.0)
  (11 200.0 200.0 0.0)
  (210 0.0 0.0 1.0)
)
```

Здесь можно увидеть, что в списке хранятся различные свойства примитива, такие как ссылки, имя и так далее. Нас же интересуют координаты начала и конца отрезка. Они хранятся в полях, начинающихся на 10 и 11. Для получения координат воспользуемся функциями cdr b assoc:

```
(setq p1 (cdr (assoc 10 line)))
(setq p2 (cdr (assoc 11 line)))
```

Здесь мы в переменных p1 и p2 будут помещены координаты начала и конца отрезка. Сложность заключается в том, что координаты заданы в начальной ПСК. Если пользователь изменил ПСК, то для данной системы координаты являются неверными. Тогда следует обратиться к функции trans:

```
(setq line (entget lineObj))
(setq p1 (cdr (assoc 10 line)))
(setq p1 (trans p1 0 1))
(setq p2 (cdr (assoc 11 line)))
(setq p2 (trans p2 0 1))
```

Тогда путаницы с координатами не возникнет.

Ниже приведен полный текст программы:

```
(defun C:linepr ()
  (setq lineList (ssget))
  (setq ++ 1+) ;Increment
  (setq i 0)
  (while (< i (sslength lineList))
    (setq lineObj (ssname lineList i))
    (setq line (entget lineObj))
    (setq p1 (cdr (assoc 10 line)))
    (setq p1 (trans p1 0 1))
    (setq p2 (cdr (assoc 11 line)))
    (setq p2 (trans p2 0 1))
    (command "_LINE" ".xy" p1 0 ".xy" p2 0 "");Proection on (X Y)
    (command "_LINE" ".xz" p1 0 ".xz" p2 0 "");Proection on (X Z)
    (command "_LINE" ".yz" p1 0 ".yz" p2 0 "");Proection on (Y Z)
    (if (= i 0) (progn
      (command "_LINE" p1 ".xy" p1 0 "")
      (command "_LINE" p1 ".xz" p1 0 "")
      (command "_LINE" p1 ".yz" p1 0 "")
    ))
    (command "_LINE" p2 ".xy" p2 0 "")
    (command "_LINE" p2 ".xz" p2 0 "")
    (command "_LINE" p2 ".yz" p2 0 "")
    (setq i (++ i))
  )
)
```

Несколько замечаний. При построении линии связей они могут «накладываться» друг на друга. Для обработки данной ошибки введено условие (if (= i 0) ...). Была «перегружена» операция инкрементирования для большего сходства с языком C (более привычным для многих программистов).

Пример работы программы представлен на рис. 1-2:

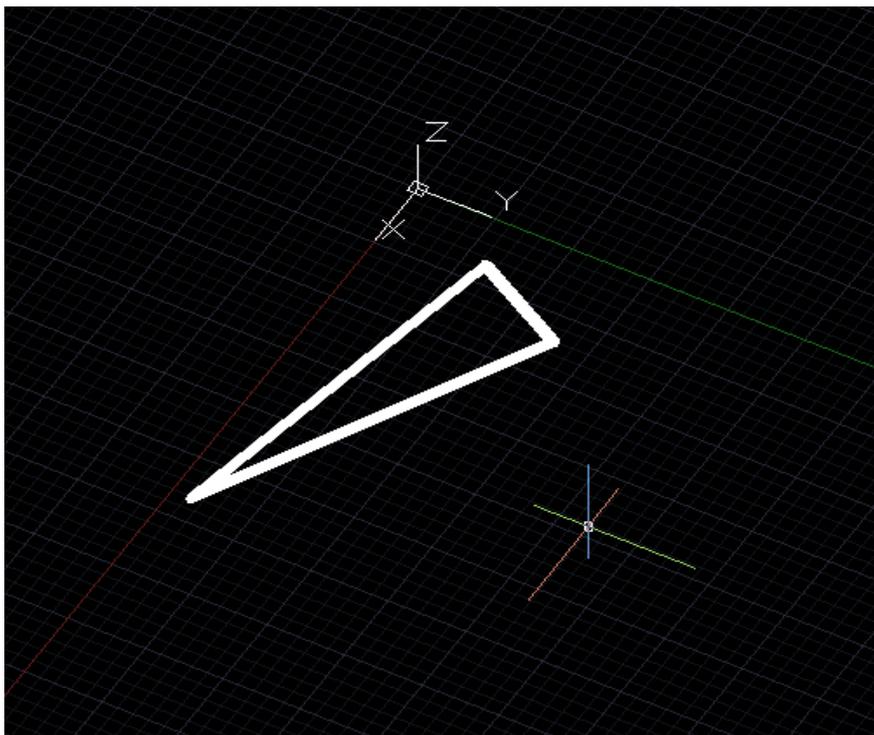


Рис. 1. Пример работы программы

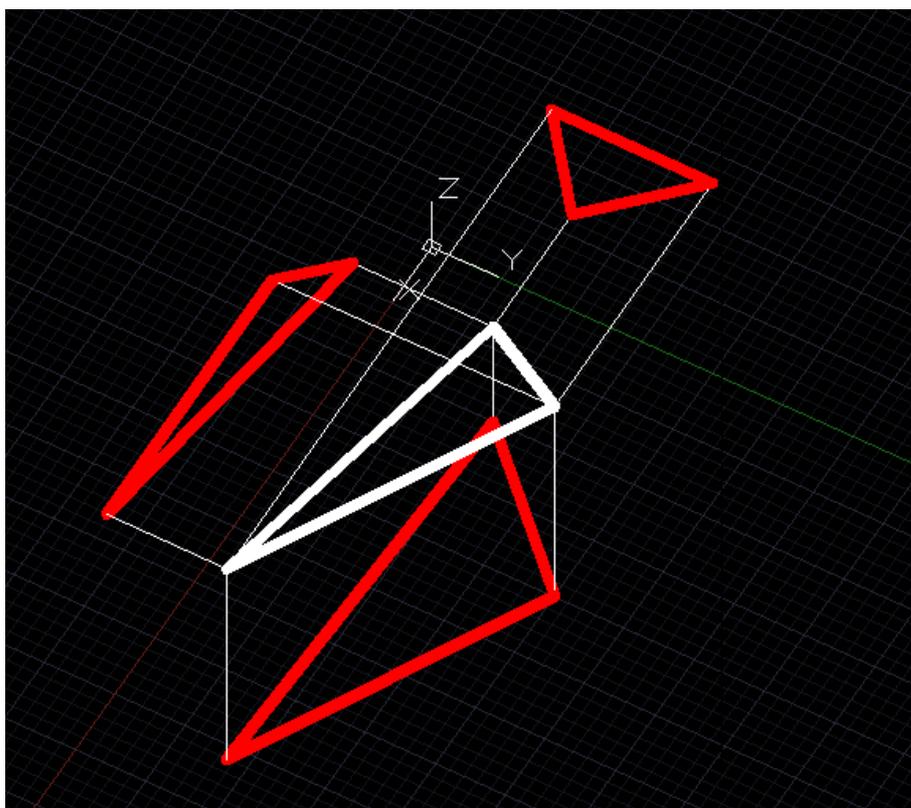


Рис. 2. Пример работы программы

Преимущества программы:

1. Ускорение процесса проецирования фигуры.

2. Автоматическое построение линий связи.
3. Снижение до 0 ошибок при проецировании по сравнению с ручным способом.

Недостатки:

1. Возможность работы только с примитивом LINE.

Данная программа была использована для решения задач по начертательной геометрии.

Решение задач по начертательной геометрии.

Протокол решения задачи.

Задача:

Построить параллелограмм ABCD при условии, что сторона BC длиной 100 мм расположена на прямой MN и длина боковой стороны AB равна 60 мм. Определить углы наклона высоты, а также плоскости параллелограмма к плоскостям проекций.

Создайте слои: Дано, h (или f в зависимости от варианта), ЛНН к П1, ЛНН к П2, Параллелограмм, Угол к П1, Угол к П2 с разными свойствами (цветом и весом линий).

1. На формате А3 в режиме ОРТО командой ОТРЕЗОК чертим 2 отрезка – направления осей X (влево) и Y (вниз);

2. Меняем точку зрения: меню ВИД – 3Dвиды = ЮЗизометрия;

Строим ось Z, используя координатные фильтры

Команда: ОТРЕЗОК

Первая точка: указать привязкой на точку пересечения отрезков

Следующая точка: **.XY** - указать привязкой на точку пересечения отрезков

Требуется Z: **100** (задать любую длину оси Z)

3. Делаем построенные оси координатными: меню СЕРВИС – Новая ПСК – 3 точки

Команда: ПСК

Новое начало координат: - указать привязкой на точку пересечения отрезков

Точка на положительном луче оси X: укажите на левый конец горизонтального отрезка

Точка на положительном луче оси Y в плоскости XY: укажите на нижний конец вертикального отрезка.

4. Строим отрезок MN по заданным координатам

Команда: ОТРЕЗОК

Первая точка: 40,60,80

Следующая точка: 150,60,40 - Enter

5. Строим высоту параллелограмма $АН$ – опускаем перпендикуляр (привязка НОРМАЛЬ) из точки A на прямую MN – нашли точку H
Команда: ОТРЕЗОК
Первая точка: 95, 90, 100
Следующая точка: привязкой Нормаль указать на прямую MN
6. Узнаем длину и угол наклона к плоскости Π_1 : меню СЕРВИС – Сведения – Список ; записывает длину и угол наклона высоты на поле чертежа;
7. Делаем координатной плоскость AMN : меню СЕРВИС – Новая ПСК – 3 точки
Команда: ПСК
Новое начало координат: - указать привязкой на точку A
Точка на положительном луче оси X : укажите на основание высоты параллелограмма
Точка на положительном луче оси Y в плоскости XY : укажите на точку M или N
8. Находим точку B на прямой MN в пересечении ее с окружностью с центром в точке A радиусом, равным стороне пар-ма 60 мм
Команда: КРУГ
Центр круга: укажите привязкой на точку A
Радиус круга: 60
9. Проведите отрезок AB , соединив точку A с одной из точек пересечения окружности и прямой MN
10. Проведите отрезок из построенной точки B до длинного конца прямой MN
11. Отложите вдоль этого отрезка длину стороны $BC = 100$
Поменяйте Стиль точки меню ФОРМАТ – Отображение точек – в диалоговом окне ОТОБРАЖЕНИЕ ТОЧЕК выберите X ;
Команда: ТОЧКА, режим РАЗМЕТИТЬ
Выберите объект для разметки: укажите на прямую MN ближе к точке B
Длина сегмента: 100 – нашли точку C
12. Достройте параллелограмм с объектной привязкой УЗЕЛ, копируя сторону AB (базовая точка B) в точку C (построили сторону CD), и соединяя точки D и C .
13. В слое параллелограмм обведите его командой ПОЛИЛИНИЯ с соответствующей толщиной.
14. Чтобы узнать угол наклона фигуры к Π_2 копируем исходные отрезки осей X и Z (базовая точка – точка их пересечения) в точку H – точку основания высоты.

Делаем эту плоскость координатной (см. п. 3), выбирая за начало координат точку Н, Точка на положительном луче оси X: укажите на левый конец горизонтального отрезка, Точка на положительном луче оси Y в плоскости XY: укажите на верхний конец вертикального отрезка (бывшей оси Z).

15. Найдите проекцию высоты АН на эту координатную плоскость

Команда: ОТРЕЗОК

Первая точка: укажите привязкой на точку Н основания высоты

Следующая точка: .XY - Enter – укажите на точку А

Требуется Z: 0 – нашли проекцию А'

16. Делаем координатной плоскость АНА' (см. п. 3), выбрав за начало координат точку Н, а за положительные направления X и Y – точки А и А'.

Узнаем угол наклона фигуры к П2: меню Размеры – Угловой – укажите на стороны угла – прямые АН и А'Н.

Вывод.

Программный способ решения задач по начертательной геометрии обладает множеством преимуществ по сравнению с ручным способом, а именно:

1. Точность решения.
2. Скорость решения.
3. Востребованность специалистов, обладающих навыками программного способа решения.
4. Актуальность способа.

Список литературы

1. Гладков С.А. Программирование на языке Автолисп в системе САПР Автокад. Л.: Диалог-МИФИ, 1991. 91 с.
2. Бугрименко Г.А. АВТОЛИСП – язык графического программирования в системе AutoCAD. Л.: Машиностроение, 1992. 139 с.
3. Суфляев Н.Е. AutoCAD в инженерной графике. Л.: Изд-во МГТУ им. Н.Э. Баумана, 2011. 30 с.