

CAD-CAM (R20A2106)

COURSE FILE

II B. Tech I Semester

(2022-2023)

Prepared By

Mr. Yugender, Assco. Prof

Department of Aeronautical Engineering



**MALLA REDDY COLLEGE OF ENGINEERING &
TECHNOLOGY**

(Autonomous Institution – UGC, Govt. of India)

Affiliated to JNTU, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2015
Certified)

Maisammaguda, Dhulapally (Post Via. Kompally), Secunderabad – 500100, Telangana State, India.

MRCET VISION

- To become a model institution in the fields of Engineering, Technology and Management.
- To have a perfect synchronization of the ideologies of MRCET with challenging demands of International Pioneering Organizations.

MRCET MISSION

To establish a pedestal for the integral innovation, team spirit, originality and competence in the students, expose them to face the global challenges and become pioneers of Indian vision of modern society.

MRCET QUALITY POLICY

- To pursue continual improvement of teaching learning process of Undergraduate and Post Graduate programs in Engineering & Management vigorously.
- To provide state of art infrastructure and expertise to impart the quality education

PROGRAM OUTCOMES

(PO's)

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design / development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multi disciplinary environments.
12. **Life- long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

DEPARTMENT OF AERONAUTICAL ENGINEERING

VISION

Department of Aeronautical Engineering aims to be indispensable source in Aeronautical Engineering which has a zeal to provide the value driven platform for the students to acquire knowledge and empower themselves to shoulder higher responsibility in building a strong nation.

MISSION

The primary mission of the department is to promote engineering education and research. To strive consistently to provide quality education, keeping in pace with time and technology. Department passions to integrate the intellectual, spiritual, ethical and social development of the students for shaping them into dynamic engineers.

QUALITY POLICY STATEMENT

Impart up-to-date knowledge to the students in Aeronautical area to make them quality engineers. Make the students experience the applications on quality equipment and tools. Provide systems, resources and training opportunities to achieve continuous improvement. Maintain global standards in education, training and services.

PROGRAM EDUCATIONAL OBJECTIVES – Aeronautical Engineering

1. **PEO1 (PROFESSIONALISM & CITIZENSHIP):** To create and sustain a community of learning in which students acquire knowledge and learn to apply it professionally with due consideration for ethical, ecological and economic issues.
2. **PEO2 (TECHNICAL ACCOMPLISHMENTS):** To provide knowledge based services to satisfy the needs of society and the industry by providing hands on experience in various technologies in core field.
3. **PEO3 (INVENTION, INNOVATION AND CREATIVITY):** To make the students to design, experiment, analyze, and interpret in the core field with the help of other multi disciplinary concepts wherever applicable.
4. **PEO4 (PROFESSIONAL DEVELOPMENT):** To educate the students to disseminate research findings with good soft skills and become a successful entrepreneur.
5. **PEO5 (HUMAN RESOURCE DEVELOPMENT):** To graduate the students in building national capabilities in technology, education and research

PROGRAM SPECIFIC OUTCOMES – Aeronautical Engineering

1. To mould students to become a professional with all necessary skills, personality and sound knowledge in basic and advance technological areas.
2. To promote understanding of concepts and develop ability in design manufacture and maintenance of aircraft, aerospace vehicles and associated equipment and develop application capability of the concepts sciences to engineering design and processes.
3. Understanding the current scenario in the field of aeronautics and acquire ability to apply knowledge of engineering, science and mathematics to design and conduct experiments in the field of Aeronautical Engineering.
4. To develop leadership skills in our students necessary to shape the social, intellectual, business and technical worlds.

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

	L	T/P/D	C
IV Year B. Tech, ANE-I Sem	4	1/-/-	3
CAD-CAM			

Objective: To Introduce To The Students About Various Components Of The Airport And Its Operations

UNIT-I

Introduction: Computers in Industrial Manufacturing, Product cycle, CAD / CAM Hardware, Basic structure.

Computer Graphics: Raster scan graphics coordinate system, database structure for graphics modeling, transformation of geometry, 3D transformations, clipping.

UNIT-II

Geometric modeling: Requirements, geometric models, geometric construction models, curve representation methods, surface representation methods, modeling facilities desired.

UNIT-III

Numerical control: NC, NC modes, NC elements, NC machine tools, structure of CNC machine tools, features of Machining center, turning center, CNC Part Programming fundamentals, manual part programming methods, Computer Aided Part Programming.

UNIT-IV

Group Technology: Part family, coding and classification, production flow analysis, advantages and limitations, Computer Aided Processes Planning, Retrieval type and Generative type.

UNIT-V

Computer Aided Quality Control: Terminology in quality control, the computer in QC, contact inspection methods, noncontact inspection methods-optical, noncontact inspection methodsnonoptical, computer aided testing, integration of CAQC with CAD/CAM.

Computer integrated manufacturing systems: Types of Manufacturing systems, Machine tools and related equipment, material handling systems, computer control systems, human labor in the manufacturing systems, CIMS benefits.

TEXT BOOK

1. CAD / CAM Theory and Practice / Ibrahim Zeid / TMH Publishers
2. CAD / CAM / A Zimmers & P. Groover / PE / PHI Publishers
3. Automation, Production systems & Computer integrated Manufacturing / Groover / Pearson Education

REFERENCES

1. CAD / CAM / CIM / Radhakrishnan and Subramanian / New Age Publishers
2. Principles of Computer Aided Design and Manufacturing / Farid Amirouche / Pearson Edu
3. CAD/CAM: Concepts and Applications / Alavala / PHI Publishers
4. Computer Numerical Control Concepts and programming / Warren S Seames / Thomson Publishers

Outcomes: The learning outcomes are assessed through the assignment and various practical performed modeling, drafting, computer aided quality control and computer integrated manufacturing systems.

UNIT 1

CAD/CAM

CAD/CAM is a term which means computer-aided design and computer- aided manufacturing. It is the technology concerned with the use of digital computers to perform certain functions in design and production. This technology is moving in the direction of greater integration of design and manufacturing, two activities which have traditionally been treated as distinct and separate functions in a production firm. Ultimately, CAD/CAM will provide the technology base for the computer-integrated factory of the future.

Computer-aided design (CAD) can be defined as the use of computer systems to assist in the creation, modification, analysis, or optimization of a design. The computer systems consist of the hardware and software to perform the specialized design functions required by the particular user firm. The CAD hardware typically includes the computer, one or more graphics display terminals, keyboards, and other peripheral equipment. The CAD software consists of the computer programs to implement computer graphics on the system plus application programs to facilitate the engineering functions of the user company. Examples of these application programs include stress-strain analysis of components, dynamic response of mechanisms, heat-transfer calculations, and numerical control part programming. The collection of application programs will vary from one user firm to the next because their product lines, manufacturing processes, and customer markets are different. These factors give rise to differences in CAD system requirements.

Computers in Industrial Manufacturing

Computer-aided manufacturing (CAM) can be defined as the use of computer systems to plan, manage, and control the operations of a manufacturing plant through either direct or indirect computer interface with the plant's production resources. As indicated by the definition, the applications of computer-aided manufacturing fall into two broad categories:

1. ***Computer monitoring and control.*** These are the direct applications in which the computer is connected directly to the manufacturing process for the purpose of monitoring or controlling the process.
2. ***Manufacturing support applications.*** These are the indirect applications in

which the computer is used in support of the production operations in the plant, but there is no direct interface between the computer and the manufacturing process.

The distinction between the two categories is fundamental to an understanding of computer-aided manufacturing. It seems appropriate to elaborate on our brief definitions of the two types.

Computer monitoring and control can be separated into monitoring applications and control applications. Computer process monitoring involves a direct computer interface with the manufacturing process for the purpose of observing the process and associated equipment and collecting data from the process. The computer is not used to control the operation directly. The control of the process remains in the hands of human operators, who may be guided by the information compiled by the computer.

Computer process control goes one step further than monitoring by not only observing the process but also controlling it based on the observations. The distinction between monitoring and control is displayed in Figure. With computer monitoring the flow of data between the process and the computer is in one direction only, from the process to the computer. In control, the computer interface allows for a two-way flow of data. Signals are transmitted from the process to the computer, just as in the case of computer monitoring. In addition, the computer issues command signals directly to the manufacturing process based on control algorithms contained in its software.

In addition to the applications involving a direct computer-process interface for the purpose of process monitoring and control, computer-aided manufacturing also includes indirect applications in which the computer serves a support role in the manufacturing operations of the plant. In these applications, the computer is not linked directly to the manufacturing process

Instead, the computer is used "off-line" to provide plans, schedules, forecasts, instructions, and information by which the firm's production resources can be managed more effectively. The form of the relationship between the computer and the process is represented symbolically in Figure. Dashed lines are used to indicate that the communication and control link is an off-line connection, with human beings often required to consummate the interface. Some examples of CAM for manufacturing support that are discussed in subsequent chapters of this book include:

Numerical control part programming by computers. Control programs are prepared for automated machine tools.

Computer-automated process planning. The computer prepares a listing of the operation sequence required to process a particular product or component.

Computer-generate work standards. The computer determines the time standard for a particular production operation.

Production scheduling. The computer determines an appropriate schedule for meeting production requirements.

Material requirements planning. The computer is used to determine when to order raw materials and purchased components and how many should be ordered to achieve the production schedule.

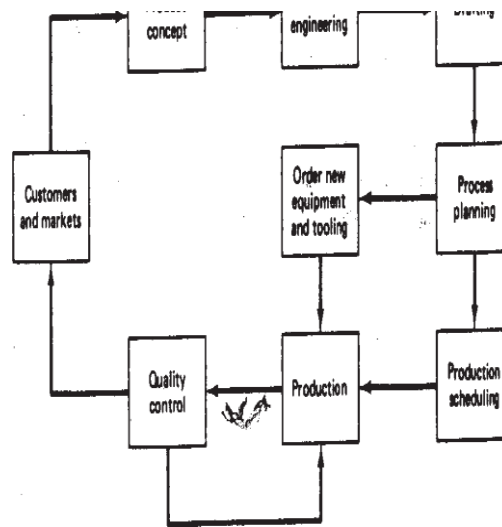
Shop floor control. In this CAM application, data are collected from the factory to determine progress of the various production shop orders. In all of these examples, human beings are presently required in the application either to provide input to the computer programs or to interpret the computer output and implement the required action.

THE PRODUCT CYCLE AND CAD/CAM

For the reader to appreciate the scope of CAD/CAM in the operations of a manufacturing firm, it is appropriate to examine the various activities and functions that must be accomplished in the design and manufacture of a product. We will refer to these activities and functions as the product cycle.

A diagram showing the various steps in the product cycle is presented in Figure. The cycle is driven by customers and markets which demand the product. It is realistic to think of these as a large collection of diverse industrial and consumer markets rather than one monolithic market. Depending on the particular customer group, there will be differences in the way the product cycle is activated. In some cases, the design functions are performed by the customer and the product is manufactured by a different firm. In other cases, design and manufacturing is accomplished by the same firm. Whatever the case, the product cycle begins with a concept, an idea for a product. This concept is cultivated, refined, analyzed, improved, and translated into a plan for the product through the design engineering process. The plan is documented by drafting a set of engineering drawings showing how the product is made and providing a set of specifications indicating how the product should perform.

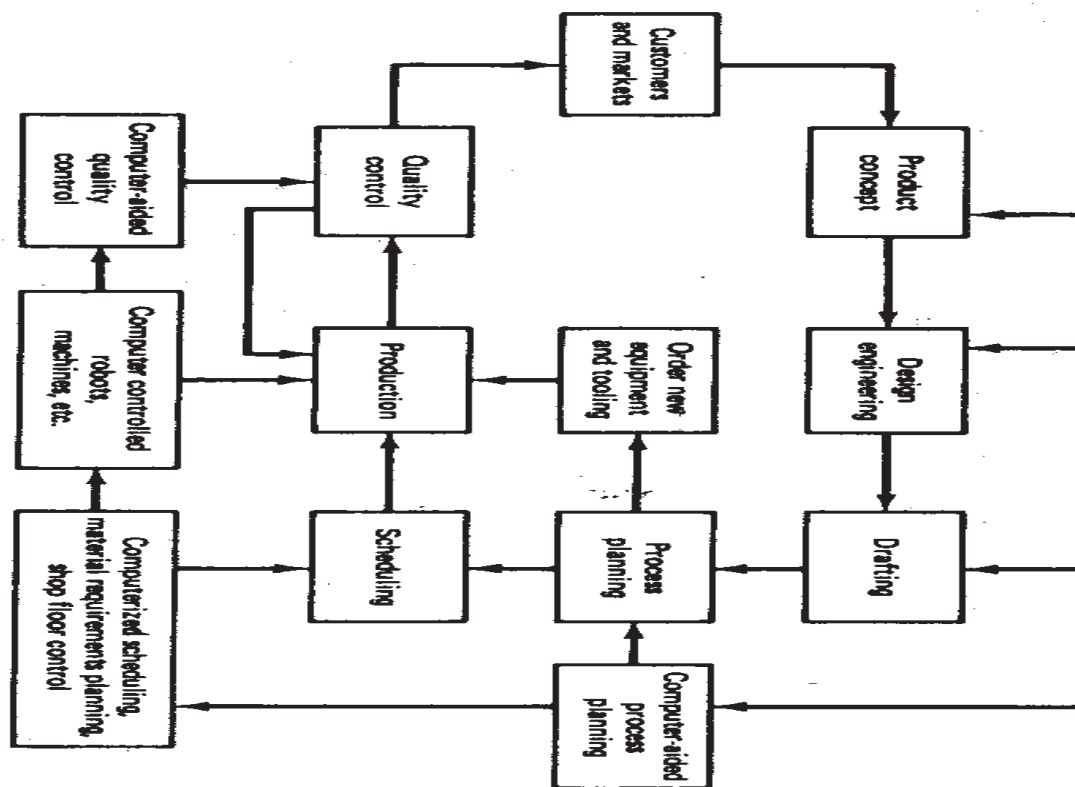
Except for engineering changes which typically follow the product throughout its life cycle, this completes the design activities in Figure. The next activities involve the manufacture of the product. A process plan is formulated which specifies the sequence of production operations required to make the product. New equipment and tools must sometimes be acquired to produce the new product. Scheduling provides a plan that commits the company to the manufacture of certain quantities of the product by certain dates. Once all of these plans are formulated, the product goes into production, followed by quality testing, and delivery to the customer.



Product cycle (design and manufacturing).

The impact of CAD/CAM is manifest in all of the different activities in the product cycle, as indicated in Figure. Computer-aided design and automated drafting are utilized in the conceptualization, design, and documentation of the product. Computers are used in process planning and scheduling to perform these functions more efficiently. Computers are used in production to monitor and control the manufacturing operations. In quality control, computers are used to perform inspections and performance tests on the product and its components.

As illustrated in Figure, CAD/CAM is overlaid on virtually all of the activities and functions of the product cycle. In the design and production operations of a modern manufacturing firm, the computer has become a pervasive, useful, and indispensable tool. It is strategically important and competitively imperative that manufacturing firms and the people who are employed by them understand CAD/CAM.



Product cycle revised with CAD/CAM overlaid.

AUTOMATION AND CAD/CAM

Automation is defined as the technology concerned with the application of complex mechanical, electronic, and computer-based systems in the operation and control of production. It is the purpose of this section to establish the relationship between CAD/CAM and automation.

As indicated in previous Section, there are differences in the way the product cycle is implemented for different firms involved in production. Production activity can be divided into four main categories:

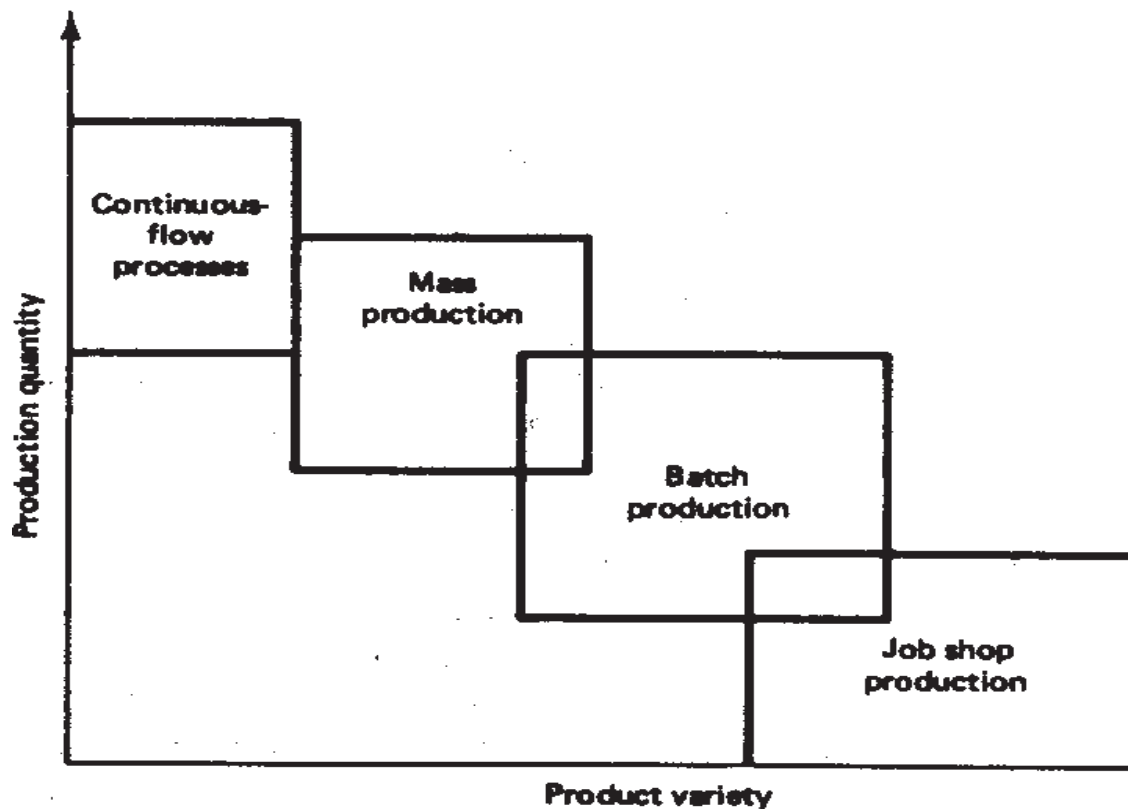
1. Continuous-flow processes
2. Mass production of discrete products
3. Batch production
4. Job shop production

The definitions of the four types are given in Table. The relationships among the four types in terms of product variety and production quantities can be conceptualized as shown in Figure. There is some overlapping of the categories as the figure indicates. Table provides a list of some of the notable achievements in automation technology for each of the four production types.

One fact that stands out from Table is the importance of computer technology in automation. Most of the automated production systems implemented today make use of computers. This connection between the digital computer and manufacturing automation may seem perfectly logical to the reader. However, this logical connection has not always existed. For one thing, automation technology

TABLE Four Types of Production

Category	Description
1. Continuous-flow processes	Continuous dedicated production of large amounts of bulk product. Examples include continuous chemical plants and oil refineries
2. Mass production of discrete products	Dedicated production of large quantities of one product (with perhaps limited model variations). Examples include automobiles, appliances, and engine blocks.
3. Batch production	Production of medium lot sizes of the same product or component. The lots may be produced once or repeated periodically. Examples include books, clothing, and certain industrial machinery.
4. Job shop production	Production of low quantities, often one of a kind, of specialized products. The products are often customized and technologically complex. Examples include prototypes, aircraft, machine tools, and other equipment.



Four production types related to quantity and product variation

TABLE Automation Achievements for the Four Types of Production

Category	Automation achievements
1. Continuous-flow processes	<p>Flow process from beginning to end</p> <p>Sensor technology available to measure important process variables</p> <p>Use of sophisticated control and optimization strategies</p> <p>Fully computer-automated plants</p>
2. Mass production of discrete products	<p>Automated transfer machines</p> <p>Dial indexing machines</p> <p>Partially and fully automated assembly lines</p> <p>Industrial robots for spot welding, parts handling, machine loading, spray painting, etc.</p> <p>Automated materials handling systems Computer production monitoring</p>

3. Batch production	Numerical control (NC), direct numerical control (DNC), computer numerical control (CNC) Adaptive control machining Robots for arc welding, parts handling, etc. Computer-integrated manufacturing systems
4. Job shop production	Numerical control, computer numerical control

FUNDAMENTALS OF CAD

INTRODUCTION

The computer has grown to become essential in the operations of business, government, the military, engineering, and research. It has also demonstrated itself, especially in recent years, to be a very powerful tool in design and manufacturing. In this and the following two chapters, we consider the application of computer technology to the design of a product. This section provides an overview of computer-aided design.

The CAD system defined

As defined in previous section, computer-aided design involves any type of design activity which makes use of the computer to develop, analyze, or modify an engineering design. Modern CAD systems (also often called CAD/CAM systems) are based on interactive computer graphics (ICG). Interactive computer graphics denotes a user-oriented system in which the computer is employed to create, transform, and display data in the form of pictures or symbols. The user in the computer graphics design system is the designer, who communicates data and commands to the computer through any of several input devices. The computer communicates with the user via a cathode ray tube (CRT). The designer creates an image on the CRT screen by entering commands to call the desired software sub-routines stored in the computer. In most systems, the image is constructed out of basic geometric elements- points, lines, circles, and so on. It can be modified according to the commands of the designer- enlarged, reduced in size, moved to another location on the screen, rotated, and other transformations. Through these various manipulations, the required details of the image are formulated.

The typical ICG system is a combination of hardware and software. The hardware includes a central processing unit, one or more workstations (including the graphics display terminals), and peripheral devices such as printers. Plotters and drafting

equipment. Some of this hardware is shown in Figure. The software consists of the computer programs needed to implement graphics processing on the system. The software would also typically include additional specialized application programs to accomplish the particular engineering functions required by the user company.

It is important to note the fact that the ICG system is one component of a computer-aided design system. As illustrated in Figure, the other major component is the human designer. Interactive computer graphics is a tool used by the designer to solve a design problem. In effect, the ICG system magnifies the powers of the designer. This has been referred to as the synergistic effect. The designer performs the portion of the design process that is most suitable to human intellectual skills (conceptualization, independent thinking); the computer performs the task: best suited to its capabilities (speed of calculations, visual display, storage of large quantities of data), and the resulting system exceeds the sum of its components.

There are several fundamental reasons for implementing a computer-aided design system.

1. To increase the productivity of the designer. This is accomplished by helping the designer to the product and its component subassemblies and parts; and by reducing the time required in synthesizing, analyzing, and documenting the design. This productivity improvement translates not only into lower design cost but also into shorter project completion times.
2. To improve the quality of design. A CAD system permits a more thorough engineering analysis and a larger number of design alternatives can be investigated. Design errors are also reduced through the greater accuracy provided by the system. These factors lead to a better design.
3. To improve communications. Use of a CAD system provides better engineering drawings, more standardization in the drawings, better documentation of the design, fewer drawing errors and greater legibility.
4. To create a database for manufacturing. In the process of creating the documentation for the product design (geometries and dimensions of the product and its components, material specifications for components, bill of materials, etc.), much of the required database to manufacture the product is also created.

THE DESIGN PROCESS

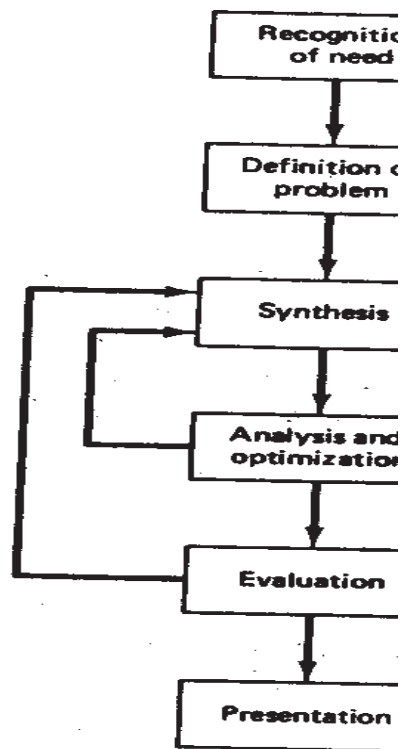
Before examining the several facets of computer-aided design, let us first consider the general design process. The process of designing something is characterized by Shigley as an iterative procedure, which consists of six identifiable steps or phases:-

1. Recognition of need
2. Definition of problem
3. Synthesis
4. Analysis and optimization
5. Evaluation
6. Presentation

Recognition of need involves the realization by someone that a problem exists for which some corrective action should be taken. This might be the identification of some defect in a current machine design by an engineer or the perception of a new product marketing opportunity by a salesperson. Definition of the problem involves a thorough specification of the item to be designed. This specification includes physical and functional characteristics, cost, quality, and operating performance.

Synthesis and analysis are closely related and highly interactive in the design process. A certain component or subsystem of the overall system is conceptualized by the designer, subjected to analysis, improved through this analysis procedure, and redesigned. The process is repeated until the design has been optimized within the constraints imposed on the designer. The components and subsystems are synthesized into the final overall system in a similar interactive manner.

Evaluation is concerned with measuring the design against the specifications established in the problem definition phase. This evaluation often requires the fabrication and testing of a prototype model to assess operating performance, quality, reliability, and other criteria. The final phase in the design process is the presentation of the design. This includes documentation of the design by means of drawings, material specifications, assembly lists, and so on. Essentially, the documentation requires that a design database be created. Figure illustrates the basic steps in the design process, indicating its iterative nature.



The general design process as defined by Shigley.

Engineering design has traditionally been accomplished on drawing boards, with the design being documented in the form of a detailed engineering drawing. Mechanical design includes the drawing of the complete product as well as its components and subassemblies, and the tools and fixtures required to manufacture the product. Electrical design is concerned with the preparation of circuit diagrams, specification of electronic components, and so on. Similar manual documentation is required in other engineering design fields (structural design, aircraft design, chemical engineering design, etc.). In each engineering discipline, the approach has traditionally been to synthesize a preliminary design manually and then to subject that design to some form of analysis. The analysis may involve sophisticated engineering calculations or it may involve a very subjective judgment of the aesthete appeal possessed by the design. The analysis procedure identifies certain improvements that can be made in the design. As stated previously, the process is iterative. Each iteration yields an improvement in the design. The trouble with this iterative process is that it is time consuming. Many engineering labor hours are required to complete the design project.

THE APPLICATION OF COMPUTERS FOR DESIGN

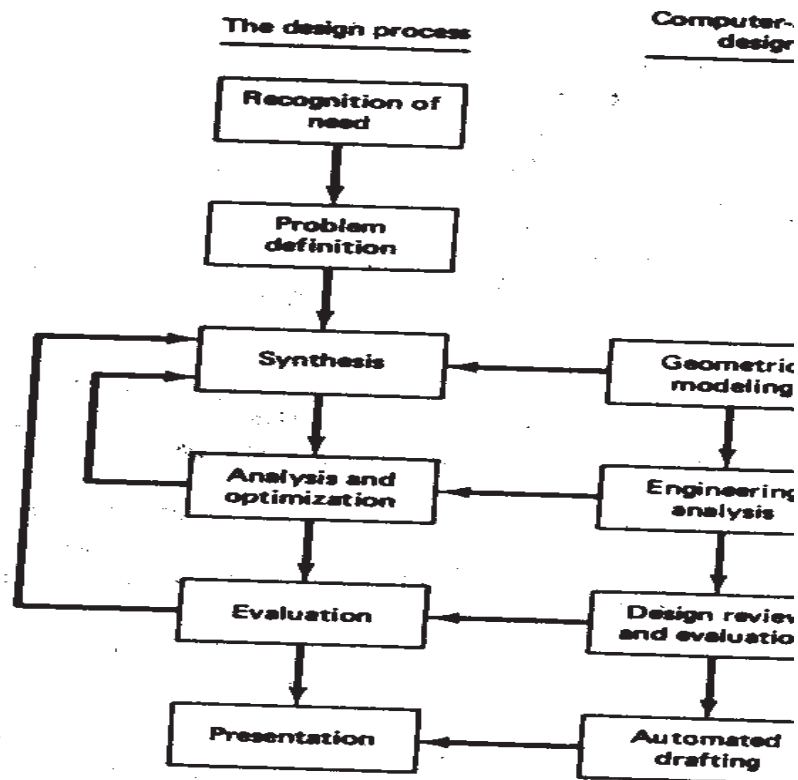
The various design-related tasks which are performed by a modern computer-aided design-system can be grouped into four functional areas:

1. Geometric modeling
2. Engineering analysis
3. Design review and evaluation
4. Automated drafting

These four areas correspond to the final four phases in Shigley's general design process, illustrated in Figure. Geometric modeling corresponds to the synthesis phase in which the physical design project takes form on the ICG system. Engineering analysis corresponds to phase 4, dealing with analysis and optimization. Design review and evaluation is the fifth step in the general design procedure. Automated drafting involves a procedure for converting the design image data residing in computer memory into a hard-copy document. It represents an important method for presentation (phase 6) of the design. The following four sections explore each of these four CAD functions.

Geometric modeling

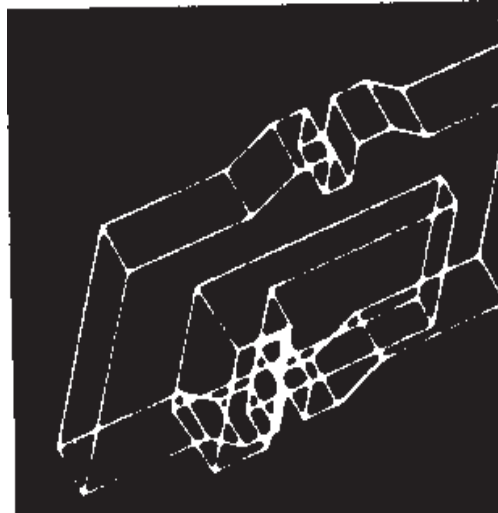
In computer-aided design, geometric modeling is concerned with the computer-compatible mathematical description of the geometry of an object. The mathematical description allows the image of the object to be displayed and manipulated on a graphics terminal through signals from the CPU of the CAD system. The software that provides geometric modeling capabilities must be designed for efficient use both by the computer and the human designer.



To use geometric modeling, the designer constructs the graphical image of the object on the CRT screen of the ICG system by inputting three types of commands to the computer. The first type of command generates basic geometric elements such as points, lines, and circles. The second command type is used to accomplish scaling, rotating, or other transformations of these elements. The third type of command causes the various elements to be joined into the desired shape of the object being created on the ICG system. During the geometric modeling process, the computer converts the commands into a mathematical model, stores it in the computer data files, and displays it as an image on the CRT screen. The model can subsequently be called from the data files for review, analysis, or alteration.

There are several different methods of representing the object in geometric modeling. The basic form uses wire frames to represent the object. In this form, the object is displayed by interconnecting lines as shown in Figure. Wire frame geometric modeling is classified into three types depending on the capabilities of the ICG system. The three types are:

1. 2D. Two-dimensional representation is used for a flat object.
2. 2½D. This goes somewhat beyond the 2D capability by permitting a three-dimensional object to be represented as long as it has no side-wall details.
3. 3D. This allows for full three-dimensional modeling of a more complex geometry.



Example of wire-frame drawing of a part.

Even three-dimensional wire-frame representations of an object are sometimes inadequate for complicated shapes. Wire-frame models can be enhanced by several different methods. Figure shows the same object shown in the previous figure but with two possible improvements. The first uses dashed lines to portray the rear edges of the object, those which would be invisible from the front. The second enhancement removes the hidden lines completely, thus providing a less cluttered picture of the object for the viewer. Some CAD systems have an automatic "hidden- line removal feature," while other systems require the user to identify the lines that are to be removed from view. Another enhancement of the wire-frame model involves providing a surface representation which makes the object appear solid to the viewer. However, the object is still stored in the computer as a wire-frame model.

The most advanced method of geometric modeling is solid modeling in three dimensions. This method, illustrated in Figure, typically uses solid geometry shapes called primitives to construct the object.



Solid model of yoke part as displayed on a computer graphics system.

(Courtesy of Computervision Corp.)

Another feature of some CAD systems is color graphics capability. By means of colour, it is possible to display more information on the graphics screen. Colored images help to clarify components in an assembly, or highlight dimensions, or a host of other purposes.

Engineering analysis

In the formulation of nearly any engineering design project, some type of analysis is required. The analysis may involve stress-strain calculations, heat-transfer computations, or the use of differential equations to describe the dynamic behavior of the system being designed. The computer can be used to aid in this analysis work. It is often necessary that specific programs be developed internally by the engineering analysis group to solve a particular design problem. In other situations, commercially available general-purpose programs can be used to perform the engineering analysis.

Turnkey CAD/CAM systems often include or can be interfaced to engineering analysis software which can be called to operate on the current design model.

We discuss two important examples of this type: Analysis of mass properties Finite-element analysis

The analysis of mass properties is the analysis feature of a CAD system that has probably the widest application. It provides properties of a solid object being analyzed, such as the surface area, weight, volume, center of gravity, and moment of inertia. For a plane surface (or a cross section of a solid object) the corresponding computations include the perimeter, area, and inertia properties.

Probably the most powerful analysis feature of a CAD system is the finite-element method. With this technique, the object is divided into a large number of finite elements (usually rectangular or triangular shapes) which form an interconnecting network of concentrated nodes. By using a computer with significant computational capabilities, the entire Object can be analyzed for stress-strain, heat transfer, and other characteristics by calculating the behavior of each node. By determining the interrelating behaviors of all the nodes in the system, the behavior of the entire object can be assessed.

Some CAD systems have the capability to define automatically the nodes and the network structure for the given object. The user simply defines certain parameters for the finite-element model, and the CAD system proceeds with the computations.

The output of the finite-element analysis is often best presented by the system in graphical format on the CRT screen for easy visualization by the user. For example, in stress-strain analysis of an object, the output may be shown in the form of a deflected shape superimposed over the unstressed object. This is illustrated in Figure. Color graphics can also be used to accentuate the comparison before and after deflection of the object. This is illustrated in Figure for the same image as that shown in Figure. If the finite-element analysis indicates behavior of the design which is undesirable, the designer can modify the shape and recompute the finite-element analysis for the revised design.

Design review and evaluation

Checking the accuracy of the design can be accomplished conveniently on the graphics terminal. Semiautomatic dimensioning and tolerancing routines which assign size specifications to surfaces indicated by the user help to reduce the possibility of dimensioning errors. The designer can zoom in on part design details and magnify the image on the graphics screen for close scrutiny.

A procedure called layering is often helpful in design review. For example, a

good application of layering involves overlaying the geometric image of the final shape of the machined part on top of the image of the rough casting. This ensures that sufficient material is available on the casting to accomplish the final machined dimensions. This procedure can be performed in stages to check each successive step in the processing of the part.

Another related procedure for design review is interference checking. This involves the analysis of an assembled structure in which there is a risk that the components of the assembly may occupy the same space. This risk occurs in the design of large chemical plants, air-separation cold boxes, and other complicated piping structures.

One of the most interesting evaluation features available on some computer-aided design systems is kinematics. The available kinematics packages provide the capability to animate the motion of simple designed mechanisms such as hinged components and linkages. This capability enhances the designer's visualization of the operation of the mechanism and helps to ensure against interference with other components. Without graphical kinematics on a CAD system, designers must often resort to the use of pin-and-cardboard models to represent the mechanism. commercial software packages are available to perform kinematic analysis. Among these are programs such as ADAMS (Automatic Dynamic Analysis of Mechanical Systems), developed at the University of Michigan. This type of program can be very useful to the designer in constructing the required mechanism to accomplish a specified motion and/or force.

Automated drafting

Automated drafting involves the creation of hard-copy engineering drawings directly from the CAD data base. In some early computer-aided design departments, automation of the drafting process represented the principal justification for investing in the CAD system. Indeed, CAD systems can increase productivity in the drafting function by roughly five times over manual drafting.

These features include automatic dimensioning, generation of crosshatched areas, scaling of the drawing, and the capability to develop sectional views and enlarged views of particular part details. The ability to rotate the part or to perform other transformations of the image (e.g., oblique, isometric, or perspective views), as illustrated in Figure, can be of significant assistance in drafting. Most CAD systems are capable of generating as

many as six views of the part. Engineering drawings can be made to adhere to company drafting standards by programming the standards into the CAD system. Figure shows an engineering drawing with four views displayed. This drawing was produced automatically by a CAD system. Note how much the isometric view promotes a higher level of understanding of the object for the user than the three orthographic views.

Parts classification and coding

In addition to the four CAD functions described above, another feature of the CAD data base is that it can be used to develop a parts classification and coding system. Parts classification and coding involves the grouping of similar part designs into classes, and relating the similarities by means of a coding scheme. Designers can use the classification and coding system to retrieve existing part designs rather than always redesigning new parts.

CREATING THE MANUFACTURING DATA BASE

Another important reason for using a CAD system is that it offers the opportunity to develop the data base needed to manufacture the product. In the conventional manufacturing cycle practiced for so many years in industry, engineering drawings were prepared by design draftsmen and then used by manufacturing engineers to develop the process plan (i.e., the "route sheets"). The activities involved in designing the product were separated from the activities associated with process planning. Essentially, a two-step procedure was employed. This was both time consuming and involved duplication of effort by design and manufacturing personnel. In an integrated CAD/CAM system, a direct link is established between product design and manufacturing: It is the goal of CAD/CAM not only to automate certain phases of design and certain phases of manufacturing, but also to automate the transition from design to manufacturing. Computer-based systems have been developed which create much of the data and documentation required to plan and manage the manufacturing operations for the product.

The manufacturing data base is an integrated CAD/CAM data base. It includes all the data on the product generated during design (geometry data, bill of materials and parts lists, material specifications, etc.) as well as additional data required for manufacturing much of which is based on the product design. Figure shows how the CAD/CAM data base is related to design and manufacturing in a typical production-oriented company.

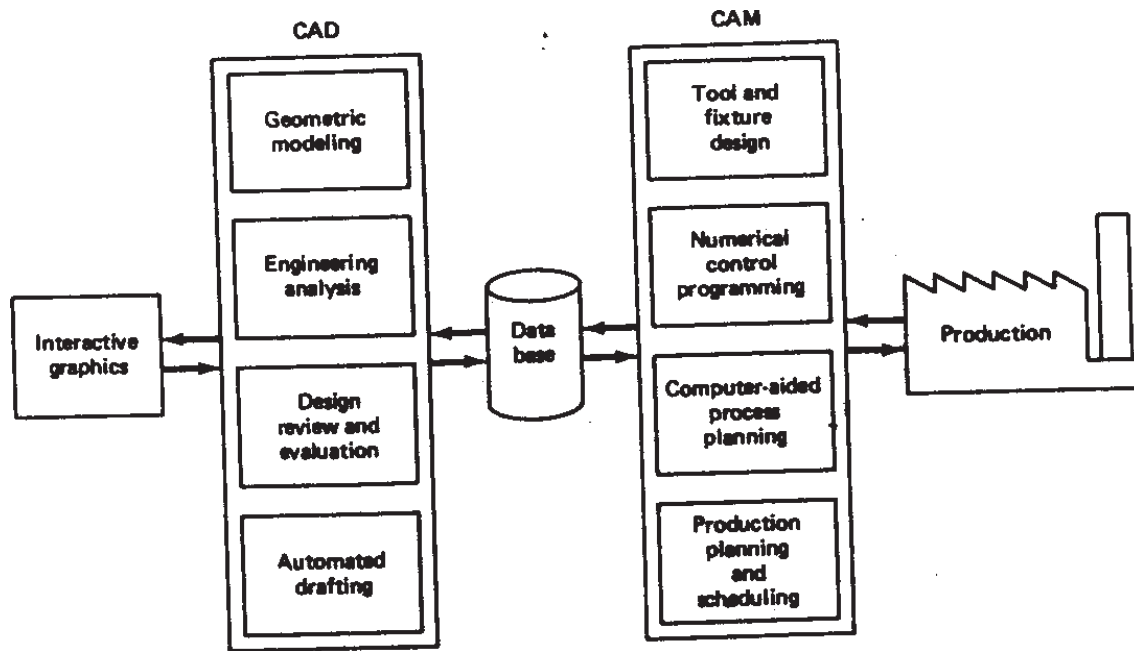


FIGURE Desirable relationship of CAD/CAM data base to CAD and CAM.

BENEFITS OF COMPUTER-AIDED DESIGN

There are many benefits of computer-aided design, only some of which can be easily measured. Some of the benefits are intangible, reflected in improved work quality, more pertinent and usable information, and improved control, all of which are difficult to quantify. Other benefits are tangible, but the savings from them show up far downstream in the production process, so that it is difficult to assign a dollar figure to them in the design phase. Some of the benefits that derive from implementing CAD/CAM can be directly measured. Table provides a checklist of potential benefits of an integrated CAD/CAM system. In the subsections that follow, we elaborate on some of these advantages.

Productivity improvement in design

Increased productivity translates into a more competitive position for the firm because it will reduce staff requirements on a given project. This leads to lower costs in addition to improving response time on projects with tight schedules.

Surveying some of the larger CAD/CAM vendors, one finds that the Productivity improvement ratio for a designer/draftsman is usually given as a range, typically from a low end of 3: 1 to a high end in excess of 10: 1 (often far in excess of that figure). There are individual cases in which productivity has been increased by a factor of 100, but it

would be inaccurate to represent that figure as typical.

TABLE Potential Benefits That May Result from implementing CAD as Part of an Integrated CAD/CAM System.

1. Improved engineering productivity
2. Shorter lead times
3. Reduced engineering personnel requirements
4. Customer modifications are easier to make
5. Faster response to requests for quotations
6. Avoidance of subcontracting to meet schedules
7. Minimized transcription errors
8. Improved accuracy of design
9. In analysis, easier recognition of component interactions
10. Provides better functional analysis to reduce prototype testing
11. Assistance in preparation of documentation
12. Designs have more standardization
13. Better designs provided
14. Improved productivity in tool design
15. Better knowledge of costs provided
16. Reduced training time for routine drafting tasks and NC part programming
17. Fewer errors in NC part programming
18. Provides the potential for using more existing parts and tooling
19. Helps ensure designs are appropriate to existing manufacturing techniques
20. Saves materials and machining time by optimization algorithms
21. Provides operational results on the status of work in progress
22. Makes the management of design personnel on projects more effective
23. Assistance in inspection of complicated parts
24. Better communication interfaces and greater understanding among engineers, designers, drafters, management, and different project groups.

Productivity improvement in computer-aided design as compared to the traditional design process is dependent on such factors as:

Complexity of the engineering drawing
Level of detail required in the drawing
Degree of repetitiveness in the designed parts
Degree of symmetry in the parts
Extensiveness of library of commonly used entities

As each of these factors is increased, the productivity advantage of CAD will tend to increase

Shorter lead times

Interactive computer-aided design is inherently faster than the traditional design. It also speeds up the task of preparing reports and lists (e.g., the assembly lists) which are normally accomplished manually. Accordingly, it is possible with a CAD system to produce a finished set of component drawings and the associated reports in a relatively short time. Shorter lead times in design translate into shorter elapsed time between receipt of a customer order and delivery of the final product. The enhanced productivity of designers working with CAD systems will tend to reduce the prominence of design, engineering analysis, and drafting as critical time elements in the overall manufacturing lead time.

Design analysis

The design analysis routines available in a CAD system help to consolidate the design process into a more logical work pattern. Rather than having a back- and- forth exchange between design and analysis groups, the same person can perform the analysis while remaining at a CAD workstation. This helps to improve the concentration of designers, since they are interacting with their designs in a real-time sense. Because of this analysis capability, designs can be created which are closer to optimum. There is a time saving to be derived from the computerized analysis routines, both in designer time and in elapsed time. This saving results from the rapid response of the design analysis and from the time no longer lost while the design finds its way from the designer's drawing board to the design analyst's queue and back again.

Fewer design errors

Interactive CAD systems provide an intrinsic capability for avoiding design, drafting, and documentation errors. Data entry, transposition, and extension errors that

occur quite naturally during manual data compilation for preparation of a bill of materials are virtually eliminated. One key reason for such accuracy is simply that

No manual handling of information is required once the initial drawing has been developed. Errors are further avoided because interactive CAD systems perform time-consuming repetitive duties such as multiple symbol placement, and sorts by area and by like item, at high speeds with consistent and accurate results. Still more errors can be avoided because a CAD system, with its interactive capabilities, can be programmed to question input that may be erroneous. For example, the system might question a tolerance of 0.00002 in. It is likely that the user specified too many zeros. The success of this checking would depend on the ability of the CAD system designers to determine what input is likely to be incorrect and hence, what to question.

Greater accuracy in design calculations

There is also a high level of dimensional control, far beyond the levels of accuracy attainable manually. Mathematical accuracy is often to 14 significant decimal places. The accuracy delivered by interactive CAD systems in three-dimensional curved space designs is so far behind that provided by manual calculation methods that there is no real comparison.

Computer-based accuracy pays off in many ways. Parts are labeled by the same recognizable nomenclature and number throughout all drawings. In some CAD systems, a change entered on a single item can appear throughout the entire documentation package, effecting the change on all drawings which utilize that part. The accuracy also shows up in the form of more accurate material and cost estimates and tighter procurement scheduling. These items are especially important in such cases as long-lead-time material purchases.

Standardization of design, drafting, and documentation procedures

The single data base and operating system is common to all workstations in the CAD system: Consequently, the system provides a natural standard for design/drafting procedure -With interactive computer-aided design, drawings are "standardized" as they are drawn; there is no confusion as to proper procedures because the entire format is "built into" the system program.

Drawings are more understandable

Interactive CAD is equally adept at creating and maintaining isometrics and

oblique drawings as well as the simpler orthographies. All drawings can be generated and updated with equal ease. Thus an up-to-date version of any drawing type can always be made available.

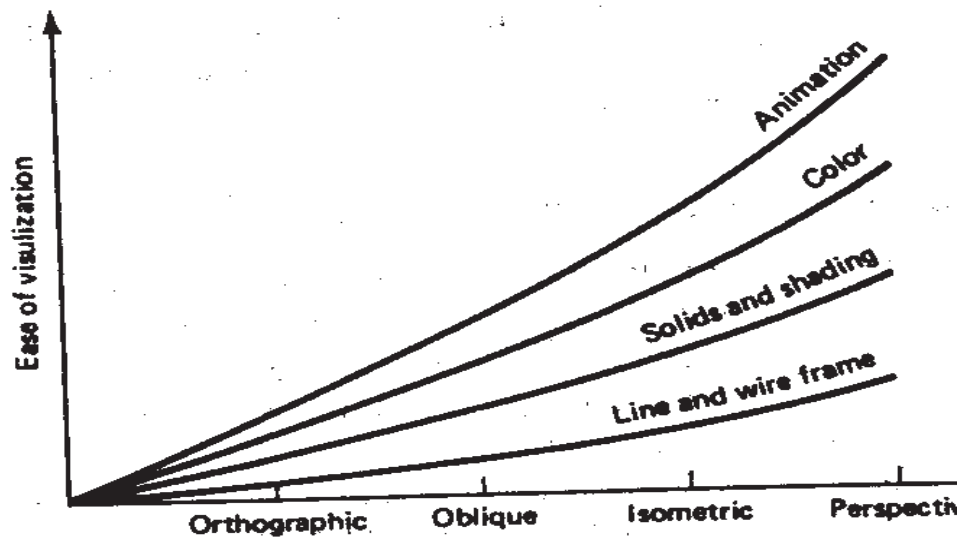


FIGURE Improvement in visualization of images for various drawing types and computer graphics features.

In general, ease of visualization of a drawing relates directly to the projection used. Orthographic views are less comprehensible than isometrics. An isometric view is usually less understandable than a perspective view. Most actual construction drawings are "line drawings." The addition of shading increases comprehension. Different colors further enhance understanding. Finally, animation of the images on the CRT screen allows for even greater visualization capability. The various relationships are illustrated in Figure..

Improved procedures for engineering changes

Control and implementation of engineering changes is significantly improved with computer-aided design. Original drawings and reports are stored in the data base of the CAD system. This makes them more accessible than documents kept in a drawing vault. They can be quickly checked against new information. Since data storage is extremely compact, historical information from previous drawings can be easily retained in the system's data base, for easy comparison with current design/drafting needs.

Benefits in manufacturing

The benefits of computer-aided design carry over into manufacturing. As

indicated previously, the same CAD/CAM data base is used for manufacturing planning and control, as well as for design. These manufacturing benefits are found in the following areas:

Tool and fixture design for manufacturing Numerical control part programming

Computer-aided process planning

Assembly lists (generated by CAD) for production Computer-aided inspection

Robotics planning Group technology

Shorter manufacturing lead times through better scheduling

These benefits are derived largely from the CAD/CAM data base, whose initial framework is established during computer-aided design. We will discuss the many facets of computer-aided manufacturing in later chapters. In the remainder of this chapter, let us explore several applications that utilize computer graphics technology to solve various problems in engineering and related fields.

HARDWARE IN COMPUTER-AIDED DESIGN

INTRODUCTION

Hardware components for computer-aided design are available in a variety of sizes, configurations, and capabilities. Hence it is possible to select a CAD system that meets the particular computational and graphics requirements of the user firm. Engineering firms that are not involved in production would choose a system exclusively for drafting and design-related functions. Manufacturing firms would choose a system to be part of a company-wide CAD/CAM system. Of course, the CAD hardware is of little value without the supporting software for the system, and we shall discuss the software for computer-aided design in the following chapter.

A modern computer-aided design system is based on interactive computer graphics (ICG). However, the scope of computer-aided design includes other computer systems as well. For example, computerized design has also been accomplished in a batch mode, rather than interactively. Batch design means that data are supplied to the system (a deck of computer cards is traditionally used for this purpose) and then the system proceeds to develop the details of the design. The disadvantage of the batch operation is that there is a time lag between when the data are submitted and when the

answer is received back as output. With interactive graphics, the system provides an immediate response to inputs by the user. The user and the system are in direct communication with each other, the user entering commands and responding to questions generated by the system.

Computer-aided design also includes nongraphic applications of the computer in design work. These consist of engineering results which are best displayed in other than graphical form. Nongraphic hardware (e.g., line printers) can be employed to create rough images on a piece of paper by appropriate combinations of characters and symbols. However, the resulting pictures, while they may create interesting wall posters, are not suitable for design purposes.

The hardware we discuss in this chapter is restricted to CAD systems that utilize interactive computer graphics. Typically, a stand-alone CAD system would include the following hardware components:

One or more design workstations. These would consist of: A graphics terminal Operator input devices.

One or more plotters and other output devices Central processing unit (CPU) Secondary storage

These hardware components would be arranged in a configuration as illustrated in Figure. The following sections discuss these various hardware components and the alternatives and options that can be obtained in each category.

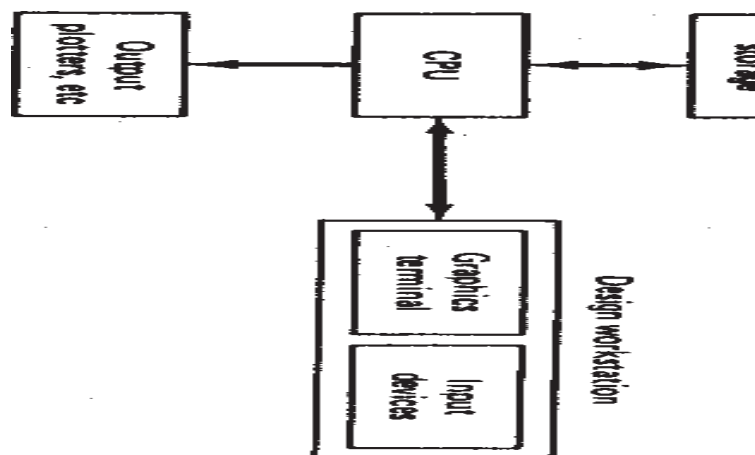


FIGURE Typical configuration of hardware components in a stand-alone CAD system. There would likely be more than one design workstation.

THE DESIGN WORKSTATION

The CAD workstation is the system interface with the outside world. It represents a significant factor in determining how convenient and efficient it is for a designer to use the CAD system. The workstation must accomplish five functions:

1. It must interface with the central processing unit.
2. It must generate a steady graphic image for the user.
3. It must provide digital descriptions of the graphic image.
4. It must translate computer commands into operating functions.
5. It must facilitate communication between the user and the system]

The use of interactive graphics has been found to be the best approach to accomplish these functions. A typical interactive graphics workstation would consist of the following hardware Components:

A graphics terminal Operator input devices

A graphics design workstation showing these components is illustrated in Figure.

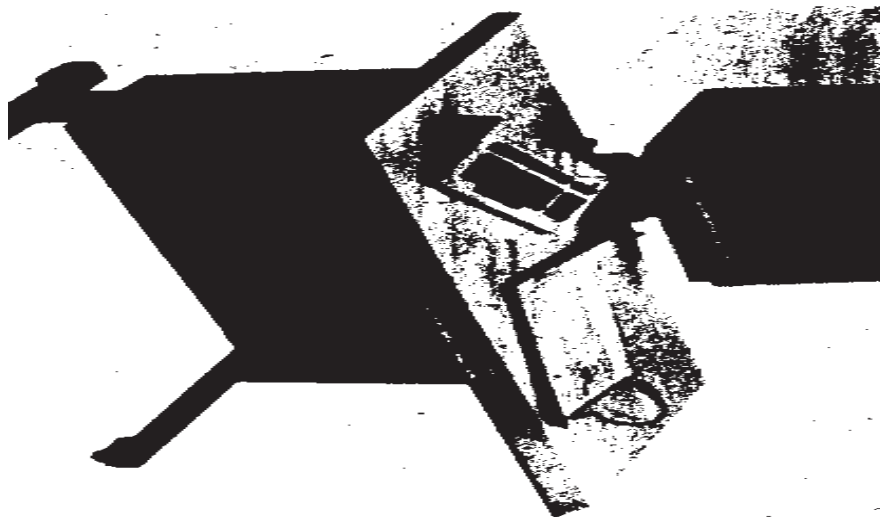


FIGURE Interactive graphics design workstation showing graphics terminal and two input devices: alphanumeric keyboard and electronic tablet and pen.

THE GRAPHICS TERMINAL

There are various technological approaches which have been applied to the development of graphics terminals. The technology continues to evolve as CAD system manufactures attempt to improve their products and reduce their costs. In this section we present a discussion of the current technology in interactive computer graphics terminals.

Image generation in computer graphics

Nearly all computer graphics terminals available today use the cathode ray tube (CRT) as the display device. Television sets use a form of the same device as the picture tube. The operation of the CRT is illustrated in Figure. A heated cathode emits a high-speed electron beam onto a phosphor-coated glass screen. The electrons energize the phosphor coating, causing it to glow at the points where the beam makes contact. By focusing the electron beam, changing its intensity, and controlling its point of contact against the phosphor coating through the use of a deflector system, the beam can be made to generate a picture on the CRT screen.

There are two basic techniques used in current computer graphics terminals for generating the image on the CRT screen. They are:

1. Stroke writing
2. Raster scan

Other names for the stroke-writing technique include line drawing, random position, vector writing, stroke writing, and directed beam. Other names for the raster scan technique include digital TV and scan graphics.

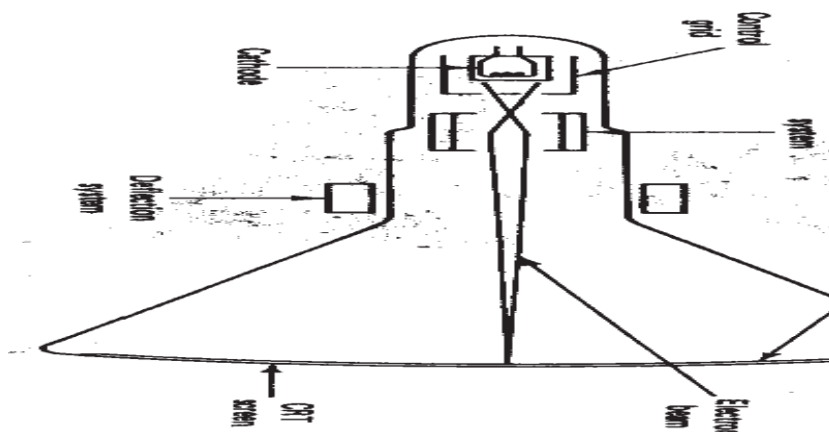


FIGURE Diagram of cathode ray tube (CRT).

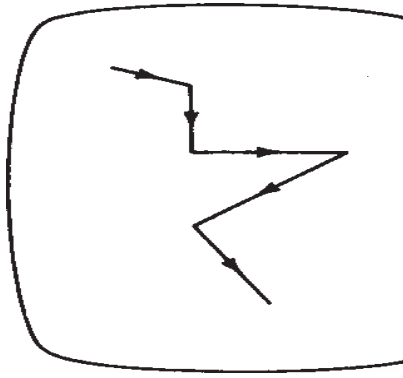


FIGURE Stroke writing for generating images in computer graphics.

The stroke-writing system uses an electron beam which operates like a pencil to create a line image on the CRT screen. The image is constructed out of a sequence of straight-line segments. Each line segment is drawn on the screen by directing the beam to move from one point on the screen to the next, where each point is defined by its x and y coordinates. The process is portrayed in Figure. Although the procedure results in images composed of only straight lines, smooth curves can be approximated by making the connecting line segments short enough.

In the raster scan approach, the viewing screen is divided into a large number of discrete phosphor picture elements, called pixels. The matrix of pixels constitutes the raster. The number of separate pixels in the raster display might typically range from 256×256 (a total of over 65,000) to 1024×1024 (a total of over 1,000,000 points). Each pixel on the screen can be made to glow with a different brightness. Color screens provide for the pixels to have different colors as well as brightness. During operation, an electron beam creates the image by sweeping along a horizontal line on the screen from left to right and energizing the pixels in that line during the sweep. When the sweep of one line is completed, the electron beam moves to the next line below and proceeds in a fixed pattern as indicated in Figure. After sweeping the entire screen the process is repeated at a rate of 30 to 60 entire scans of the screen per second.

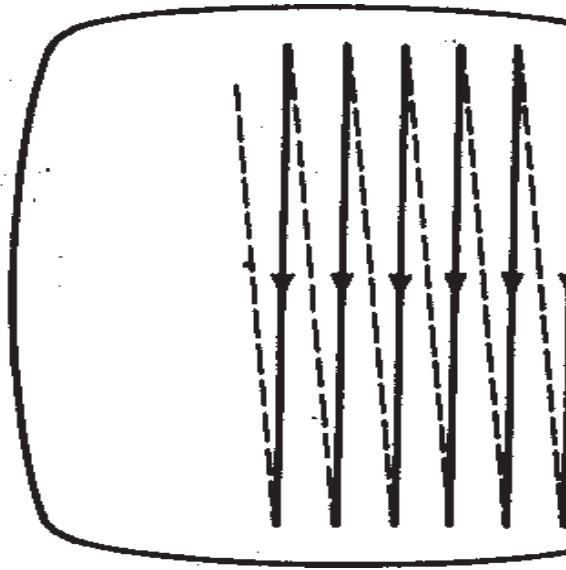


FIGURE Raster scan approach for generating images in computer graphics.

Graphics terminals for computer-aided design

The two approaches described above are used in the overwhelming majority of current-day CAD graphics terminals. There are also a variety of other technical factors which result in different types of graphics terminals. These factors include the type of phosphor coating on the screen, whether color is required, the pixel density, and the amount of computer memory available to generate the picture. We will discuss three types of graphics terminals, which seem to be the most important today in commercially available CAD systems. The three types are:

1. Directed-beam refresh
2. Direct-view storage tube (DVST)
3. Raster scan (digital TV)

The following paragraphs describe the three basic types. We then discuss some of the possible enhancements, such as color and animation.

DIRECTED-BEAM REFRESH. The directed-beam refresh terminal utilizes the stroke-writing approach to generate the image on the CRT screen. The term “refresh” in the name refers to the fact that the image must be regenerated many times per second in order to avoid noticeable flicker of the image. The phosphor elements on the screen surface are capable of maintaining their brightness for only a short time (sometimes measured in microseconds). In order for the image to be continued, these picture tubes

must be refreshed by causing the directed beam to retrace the image repeatedly. On densely filled screens (very detailed line images or many characters of text), it is difficult to avoid flickering of the image with this process. On the other hand, there are several advantages associated with the directed- beam refresh systems. Because the image is being continually refreshed, selective erasure and alteration of the image is readily accomplished. It is also possible to provide animation of the image with a refresh tube.

The directed-beam refresh system is the oldest of the modem graphics display technologies. Other names sometimes used to identify this system include vector refresh and stroke-writing refresh. Early refresh tubes were very expensive. but the steadily decreasing cost of solid-state circuitry has brought the price of these graphics systems down to a level which is competitive with other types.

DIRECT-VIEW STORAGE TUBE (DVST). DVST terminals also use the stroke-writing approach to generate the image on the CRT screen. The term storage tube refers to the ability of the screen to retain the image which has been projected against it, thus avoiding the need to rewrite the image which has been projected against it, thus avoiding the need to rewrite the image constantly. What makes this possible is the use of an electron flood gun directed at the phosphor coated screen which keeps the phosphor elements illuminated once they have been energized by the stroke-writing electron beam. The resulting image on the CRT screen is flicker- free. Lines may be readily added to the image without concern over their effect on image density or refresh rates. However, the penalty associated with the storage tube is that individual lines cannot be selectively removed from the image.

Storage tubes have historically been the lowest-cost terminals and are capable of displaying large amounts of data, either graphical or textual. Because of these features, there are probably more storage tube terminals in service in industry at the time of this writing than any other graphics display terminal. The principal disadvantage of a storage CRT is that selective erasure is not possible. Instead, if the user wants to change the picture, the change will not be manifested on the screen until the entire picture is regenerated. Other disadvantages include its lack of color capability, the inability to use a light pen as a data entry, and its lack of animation capability.

RASTER SCAN TERMINALS.

Raster scan terminals operate by causing an electron beam to trace a zigzag pattern across the viewing screen, as described earlier. The operation is similar to that of a commercial television set. The difference is that a TV set uses analog signals originally generated by a video camera to construct the image on the CRT screen, while the raster scan ICG terminal uses digital signals generated by a computer. For this reason, the raster scan terminals used in computer graphics are sometimes called digital TVs.

The introduction of the raster scan graphics terminal using a refresh tube had been limited by the cost of computer memory. For example, the simplest and lowest-cost terminal in this category uses only two beam intensity levels, on or off. This means that each pixel in the viewing screen is either illuminated or dark. A picture tube with 256 lines of resolution and 256 addressable points per line to form the image would require 256×256 or over 65,000 bits of storage. Each bit of memory contains the on/off status of the corresponding pixel on the CRT screen. This memory is called the frame buffer or refresh buffer. The picture quality can be improved in two ways: by increasing the pixel density or adding a gray scale (or color). Increasing pixel density for the same size screen means adding more lines of resolution and more addressable points per line. A 1024×1024 raster screen would require more than 1 million bits of storage in the frame buffer. A gray scale is accomplished by expanding the number of intensity levels which can be displayed on each pixel. This requires additional bits for each pixel to store the intensity level. Two bits are required for four levels, three bits for eight levels, and so forth. Five or six bits would be needed to achieve an approximation of a continuous gray scale. For a color display, three times as many bits are required to get various intensity levels for each of the three primary colors: red, blue, and green. (We discuss color in the following section.) A raster scan graphics terminal with high resolution and gray scale can require a very large capacity refresh buffer. Until recent developments in memory technology, the cost of this storage capacity was prohibitive for a terminal with good picture quality. The capability to achieve color and animation was not possible except for very low resolution levels.

TABLE Comparison of Graphics Terminal Features

	Directed-beam refresh	DVST	Raster scan
Image generation	Stroke writing	Stroke writing	Raster scan
Picture quality	Excellent	Excellent	Moderate to good
Data content	Limited	High	High
Selective erase	Yes	No	Yes
Gray scale	Yes	No	Yes
Color capability	Moderate	No	Yes
Animation capability	Yes	No	Moderate

It is now possible to manufacture digital TV systems for interactive computer graphics at prices which are competitive with the other two types. The advantages of the present raster scan terminals include the feasibility to use low-cost TV monitors, color capability, and the capability for animation of the image. These features, plus the continuing improvements being made in raster scan technology, make it the fastest-growing segment of the graphics display market.

The typical color CRT uses three electron beams and a triad of color dots on the phosphor screen to provide each of the three colors, red, green, and blue. By combining the three colors at different intensity levels, a variety of colors can be created on the screen. It is more difficult to fabricate a stroke-writing tube which is precise enough for color because of the technical problem of getting the three beams to converge properly against the screen.

The raster scan approach has superior color graphics capabilities because of the developments which have been made over the years in the color television industry. Color raster scan terminals with 1024×1024 resolution are commercially available for computer graphics. The problem in the raster terminals is the memory requirements of the refresh buffer. Each pixel on the viewing screen may require up to 24 bits of memory in the refresh buffer in order to display the full range of color tones. When multiplied by

the number of pixels in the display screen, this translates into a very large storage buffer.

The capability for animation in computer graphics is limited to display methods in which the image can be quickly redrawn. This limitation excludes the storage tube terminals. Both the directed-beam refresh and the raster scan systems are capable of animation. However, this capability is not automatically acquired with these systems. It must be accomplished by means of a powerful and fast CPU interfaced to the graphics terminal to process the large volumes of data required for animated images. In computer-aided design, animation would be a powerful feature in applications where kinematic simulation is required. The analysis of linkage mechanisms and other mechanical behavior would be examples. In computer-aided manufacturing, the planning of a robotic work cycle would be improved through the use of an animated image of the robot simulating the motion of the arm during the cycle. The popular video games marketed by Atari and other manufacturers for use with home TV sets are primitive examples of animation in computer graphics. Animation in these TV games is made possible by sacrificing the quality of the picture. This keeps the price of these games within an affordable range.

OPERATOR INPUT DEVICES

Operator input devices are provided at the graphics workstation to facilitate convenient communication between the user and the system. Workstations generally have several types of input devices to allow the operator to select the various preprogrammed input functions. These functions permit the operator to create or modify an image on the CRT screen or to enter alphanumeric data into the system. This results in a complete part on the CRT screen as well as complete geometric description of the part in the CAD data base.

Different CAG system vendors offer different types of operator input devices. These devices can be divided into three general categories:

1. Cursor control devices
2. Digitizers
3. Alphanumeric and other keyboard terminals

Of the three, cursor control devices and digitizers are both used for graphical

interaction with the system. Keyboard terminals are used as input devices for commands and numerical data.

There are two basic types of graphical interaction accomplished by means of cursor control and digitizing:

Creating and positioning new items on the CRT screen

Pointing at or otherwise identifying locations on the screen, usually associated with existing images

Ideally, a graphical input device should lend itself to both of these functions. However, this is difficult to accomplish with a single unit and that is why most workstations have several different input devices.

Cursor control

The cursor normally takes the form of a bright spot on the CRT screen that, indicates where lettering or drawing will occur. The computer is capable of reading the current position of the cursor. Hence the user's capability to control the cursor position allows locational data to be entered into the CAD system data base. A typical example would be for the user to locate the cursor to identify the starting point of a line. Another, more sophisticated case, would be for the user to position the cursor to select an item from a menu of functions displayed on the screen. For instance, the screen might be divided into two sections, one of which is an array of blocks which correspond to operator input functions. The user simply moves the cursor to the desired block to execute the particular function.

There are a variety of cursor control devices which have been employed in CAD systems. These include:

Thumbwheels

Direction keys on a keyboard terminal Joysticks

Tracker ball Light pen

Electronic tablet/pen

The first four items in the list provide control over the cursor without any direct physical contact of the screen by the user. The last two devices in the list require the user to control the cursor by touching the screen (or some other flat surface which is related to

the screen) with a pen-type device.

The thumbwheel device uses two thumbwheels, one to control the horizontal position of the cursor, the other to control the vertical position. This type of device is often mounted as an integral part of the CRT terminal. The cursor in this arrangement is often represented by the intersection of a vertical line and a horizontal line displayed on the CRT screen. The two lines are like crosshairs in a gunsight which span the height and width of the screen.

Direction keys on the keyboard are another basic form of cursor control used not only for graphics terminals but also for CRT terminals without graphics capabilities. Four keys are used for each of the four directions in which the cursor can be moved (right or left, and up or down).

The joystick apparatus is pictured in Figure. It consists of a box with a vertical toggle stick that can be pushed in any direction to cause the cursor to be moved in that direction. The joystick gets its name from the control stick that was used 10 old airplanes.

The tracker ball is pictured in Figure. Its operation is similar to that of the joystick except that an operator-controlled ball is rotated to move the cursor in the desired direction on the screen.

The light pen is a pointing device in which the computer seeks to identify the

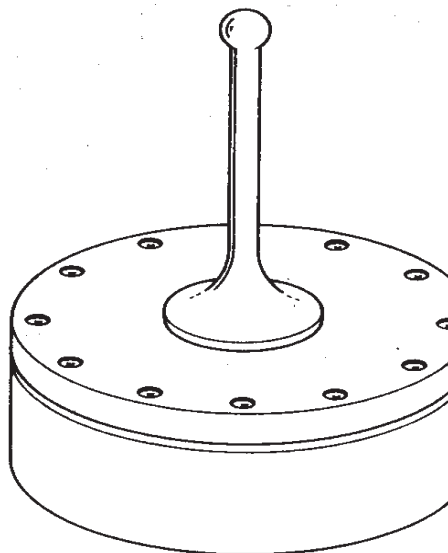


FIGURE Joystick input device for interactive computer graphics

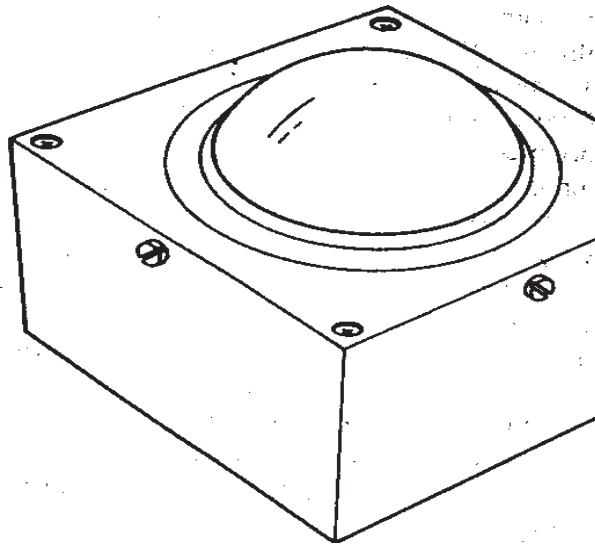


FIGURE Tracker ball input device for interactive computer graphics.

position where the light pen is in contact with the screen. Contrary to what its name suggests, the light pen does not project light. Instead, it is a detector of light on the CRT screen and uses a photodiode, phototransistor, or some other form of light sensor. The light pen can be utilized with a refresh-type CRT but not with a storage tube. This is because the image on the refresh tube is being generated in time sequence. The time sequence is so short that the image appears continuous to the human eye. However, the computer is capable of discerning the time sequence and it coordinates this timing with the position of the pen against the screen. In essence, the system is performing as an optical tracking loop to locate the cursor or to execute some other input function. The tablet and pen in computer graphics describes an electronically sensitive tablet used in conjunction with an electronic stylus. The tablet is a flat surface, separate from the CRT screen, on which the user draws with the penlike stylus to input instructions or to control the cursor

It should be noted that thumbwheels, direction keys, joysticks, and tracker balls are generally limited in their functions to cursor control. The light pen and tablet/pen are

typically used for other input functions as well as cursor control. Some of these functions are:

Selecting from a function menu

Drawing on the screen or making strokes on the screen or tablet which indicate what image is to be drawn

Selecting a portion of the screen for enlargement of an existing image

Digitizers

The digitizer is an operator input device which consists of a large, smooth board (the appearance is similar to a mechanical drawing board) and an electronic tracking device which can be moved over the surface to follow existing lines. It is a common technique in CAD systems for taking x, y coordinates from a paper drawing. The electronic tracking device contains a switch for the user to record the desired x and y coordinate positions. The coordinates can be entered into the computer memory or stored on an off-line storage medium such as magnetic tape. High-resolution digitizers, typically with a large board (e.g., 42 in by 60 in.) can provide resolution and accuracy on the order of 0.001 in. It should be mentioned that the electronic tablet and pen, previously discussed as a cursor control device, can be considered to be a small, low-resolution digitizer.

Not all CAD systems would include a digitizer as part of its core of operator input devices. It would be inadequate, for example, in three-dimensional mechanical design work since the digitizer is limited to two dimensions. For two-dimensional drawings, drafters can readily adapt to the digitizer because it is similar to their drafting boards. It can be tilted, raised, or lowered to assume a comfortable position for the drafter.

The digitizer can be used to digitize line drawings. The user can input data from a rough schematic or large layout drawing and edit the drawings to the desired level of accuracy and detail. The digitizer can also be used to freehand a new design with subsequent editing to finalize the drawing.

Keyboard terminals

Several forms of keyboard terminals are available as CAD input devices. The most familiar type is the alphanumeric terminal which is available with nearly all interactive graphics systems. The alphanumeric terminal can be either a CRT or a hard

copy terminal, which prints on paper. For graphics, the CRT has the advantage because of its faster speed, the ability to easily edit, and the avoidance of large volumes of paper. On the other hand, a permanent record is sometimes desirable and this is most easily created with a hard-copy terminal. Many CAD systems use the graphics screen to display the alphanumeric data, but there is an advantage in having a separate CRT terminal so that the alphanumeric messages can be created without disturbing or overwriting the image on the graphics screen.

The alphanumeric terminal is used to enter commands, functions, and supplemental data to the CAD system. This information is displayed for verification on the CRT or typed on paper. The system also communicates back to the user in a similar manner. Menu listings, program listings, error messages, and so forth, can be displayed by the computer as part of the interactive procedure.

These function keyboards are provided to eliminate extensive typing of commands, or calculate coordinate positions, and other functions. The number of function keys varies from about 8 to 80. The particular function corresponding with each button is generally under computer control so that the button function can be changed as the user proceeds from one phase of the design to the next. In this way the number of alternative functions can easily exceed the number of buttons on the keyboard.

Also, lighted buttons are used on the keyboards to indicate which functions are possible in the current phase of design activity. A menu of the various function alternatives is typically displayed on the CRT screen for the user to select the desired function.

PLOTTERS AND OTHER OUTPUT DEVICES

There are various types of output devices used in conjunction with a computer-aided design system. These output devices include:

Pen plotters Hard-copy units,

Electrostatic plotters Computer-output-to-microfilm (COM) units We discuss these devices in the following sections.

Pen plotters

The accuracy and quality of the hard-copy plot produced by a pen plotter is

considerably greater than the apparent accuracy and quality of the corresponding image on the CRT screen. In the case of the CRT image, the quality of the picture is degraded because of lack of resolution and because of losses in the digital-to-analog conversion through: the display generators. On the other hand, a high-precision pen plotter is capable of achieving a hard-copy drawing whose accuracy is nearly consistent with the digital definitions in the CAD data base.

The pen plotter uses a mechanical ink pen (either wet ink or ballpoint) to write on paper through relative movement of the pen and paper. There are two basic types of pen plotters currently in use:

Drum plotters Flat-bed plotters

Hard-copy unit

A hard-copy unit is a machine that can make copies from the same image data layed on the CRT screen. The image on the screen can be duplicated in a matter of seconds. The copies can be used as records of intermediate steps in the design process or when rough hard copies of the screen are needed quickly. The hard copies produced from these units are not suitable as final drawings because the accuracy and quality of the reproduction is not nearly as good as the output of a pen plotter.

Most hard-copy units are dry silver copiers that use light-sensitive paper exposed through a narrow CRT window inside the copier. The window is typically 8½ in. (216 mm), corresponding to the width of the paper, by about ½ in. (12 mm) wide. The paper is exposed by moving it past the window and coordinating the CRT beam to gradually transfer the image. A heated roller inside the copier is used to develop the exposed paper. The size of the paper is usually limited on these hard-copy units to 8½ by 11 in. Another drawback is that the dry silver copies will darken with time when they are left exposed to normal light.

Electrostatic plotters

Hard-copy units are relatively fast but their accuracy and resolution are poor. Pen plotters are highly accurate but plotting time can take many minutes (up to a half-hour or longer for complicated drawings). The electrostatic plotter offers a compromise between these two types in terms of speed and accuracy. It is almost as fast as the hard-copy unit and almost as accurate as the pen plotter.

The electrostatic copier consists of a series of wire styli mounted on a bar which spans the width of the charge-sensitive paper. The styli have a density of up to 200 per linear inch. The paper is gradually moved past the bar and certain styli are activated to place dots on the paper. By coordinating the generation of the dots with the paper travel, the image is progressively transferred from the data base into hard-copy form. The dots overlap each other slightly to achieve continuity. For example, a series of adjacent dots gives the appearance of a continuous line.

A limitation of the electrostatic plotter is that the data must be in the raster format (i.e., in the same format used to drive the raster-type CRT) in order to be readily converted into hard copy using the electrostatic method. If the data are not in raster format, some type of conversion is required to change them into the required format. The conversion mechanism is usually based on a combination of software and hardware.

An advantage of the electrostatic plotter which is shared with the drum-type pen plotter is that the length of the paper is virtually unlimited. Typical plotting widths might be up to 6 ft (1.83 m). Another advantage is that the electrostatic plotter can be utilized as a high-speed line printer, capable of up to 1200 lines of text per minute.

THE CENTRAL PROCESSING UNIT

The CPU operates as the central "brain" of the computer-aided design system. It is typically a minicomputer. It executes all the mathematical computations needed to accomplish graphics and other functions, and it directs the various activities within the system.

COMPUTER GRAPHICS SOFTWARE AND DATA BASE

INTRODUCTION

The graphics software is the collection of programs written to make it convenient for a user to operate the computer graphics system. It includes Programmes to generate images on the CRT screen, to manipulate the images, and to accomplish various types of interaction between the user and the system. In addition to the graphics software, there may be additional programs for implementing certain specialized functions related to CAD/CAM. These include design analysis programs(e.g., finite-element analysis and kinematic simulation) and Manufacturing planning programs (e.g., automated process planning and numerical control part programming). This chapter deals mainly with the graphics software.

The graphics software for a particular computer graphics system is very much a function of the type of hardware used in the system. The software must be written specifically for the type of CRT and the types of input devices used in the system. The details of the software for a stroke-writing CRT would be different than for a raster scan CRT. The differences between a storage tube and a refresh tube would also influence the graphics software. Although these differences in software may be invisible to the user to some extent, they are important considerations in the design of an interactive computer graphics system.

Newman and Spoull list six “ground rules” that should be considered in designing graphics software:

Simplicity. The graphics software should be easy to use.

1. Consistency . The package should operate in a consistent and predictable way to the user.
2. Completeness. There should be no inconvenient omissions in the set of graphics functions.
3. Robustness. The graphics system should be tolerant of minor instances of misuse by the operator.
4. Performance. Within limitations imposed by the system hardware, the performance should be exploited as much as possible by software. Graphics programs should be efficient and speed of response should be fast and consistent.
5. Economy. Graphics programs should not be so large or expensive as to make their use prohibitive.

THE SOFTWARE CONFIGURATION OF A GRAPHICS SYSTEM

In the operation of the graphics system by the user, a variety of activities take place, which can be divided into three categories:

1. Interact with the graphics terminal to create and alter images on the screen
2. Construct a model of something physical out of the images on the screen. The models are sometimes called application models.
3. Enter the model into computer memory and/or secondary storage.

In working with the graphics system the user performs these various activities in combination rather than sequentially. The user constructs a physical model and inputs it to memory by interactively describing images to the system. This is done without any thought about whether the activity falls into category 1, 2, or 3.

The reason for separating these activities in this fashion is that they correspond to the general configuration of the software package used with the interactive computer graphics (ICG) system. The graphics software can be divided into three modules according to a conceptual model suggested by Foley and Van Dam:

1. The graphics package (Foley and Van Dam called this the graphics system)
2. The application program
3. The application data base

This software configuration is illustrated in Figure. The central module is the application program. It controls the storage of data into and retrieves data out of the application data base. The application program is driven by the user through the graphics package.

The application program is implemented by the user to construct the model of a physical entity whose image is to be viewed on the graphics-screen. Application programs are written for particular problem areas. Problem areas in engineering design would include architecture, construction, mechanical components, electronics, chemical engineering, and aerospace engineering. Problem areas other than design would include flight simulators, graphical display of data, mathematical analysis, and even artwork. In each case, the application software is developed to deal with images and conventions which are appropriate for that field.

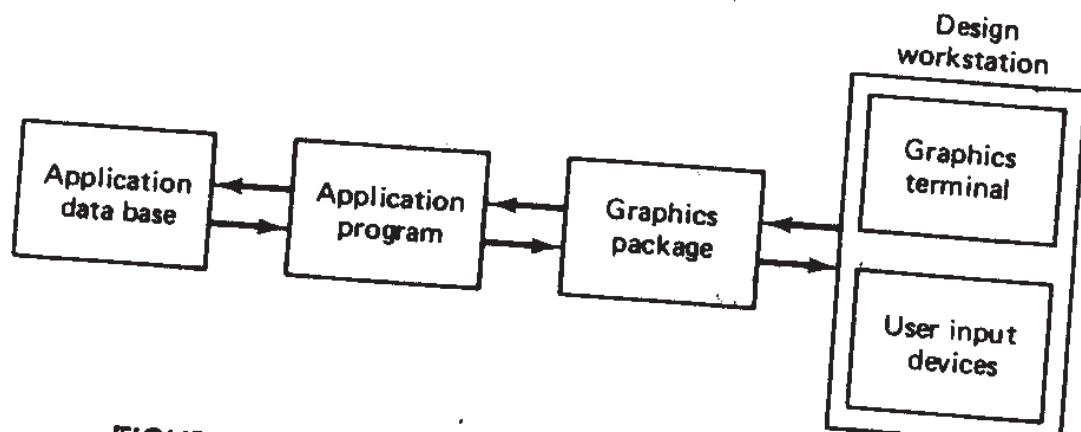


FIGURE 1

The graphics package is the software support between the user and the graphics terminal. It manages the graphical interaction between the user and the system. It also serves as the interface between the user and the application software. The graphics package consists of input subroutines and output subroutines. The input routines accept input commands and data from the user and forward them to the application program. The output subroutines control the display terminal (or other output device) and converts the application models into two-dimensional or three- dimensional graphical pictures.

The third module in the ICG software is the data base. The data base contains mathematical, numerical, and logical definitions of the application models, such as electronic circuits, mechanical components, automobile bodies, and so forth. It also includes alphanumeric information associated with the models, such as bills of materials, mass properties, and other data. The contents of the data base can be readily displayed on the CRT or plotted out in hard-copy form.

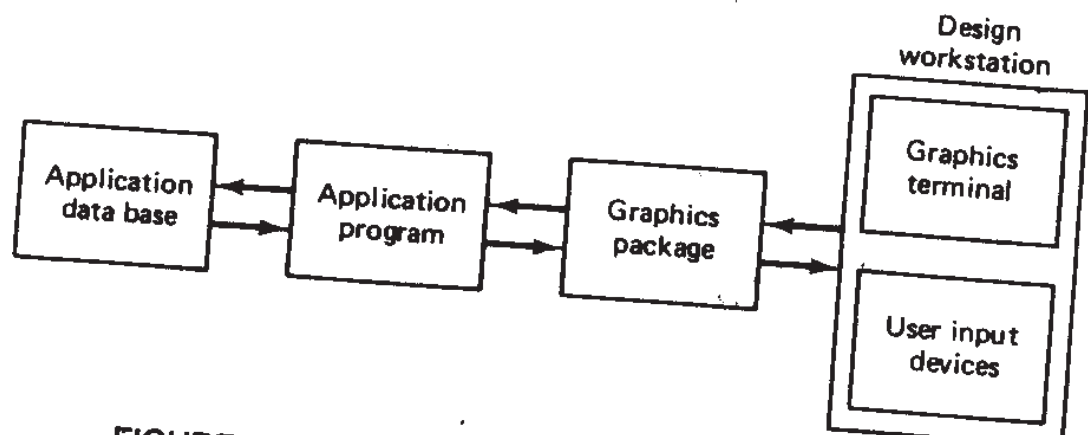


FIGURE Model of graphics software configuration.

FUNCTIONS OF A GRAPHICS PACKAGE

To fulfill its role in the software configuration, the graphics package must perform a variety of different functions. these functions can be grouped into function sets. Each set accomplishes a certain kind of interaction between the user and the system. Some of the common function sets are:

Generation of graphic elements Transformations

Display control and windowing functions Segmenting functions

TRANSFORMATIONS

Many of the editing features involve transformations of the graphics elements or cells composed of elements or even the entire model. In this section we discuss the mathematics of these transformations. Two-dimensional transformations are considered first to illustrate concepts. Then we deal with three dimensions.

Two-dimensional transformations

To locate a point in a two-axis cartesian system, the x and y coordinates are specified. These coordinates can be treated together as a 1x1 matrix: (x,y). For example, the matrix (2, 5) would be interpreted to be a point which is 2 units from the origin in the x-direction and 5 units from the origin in the y-direction.

This method of representation can be conveniently extended to define a line as a 2 x 2 matrix by giving the x and y coordinates of the two end points of the line. The notation would be

$$L = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}$$

Using the rules of matrix algebra, a point or line (or other geometric element represented in matrix notation) can be operated on by a transformation matrix to yield a new element.

There are several common transformations used in computer graphics. We will discuss three transformations: translation, scaling, and rotation.

TRANSLATION. Translation involves moving the element from one location to another. In the case of a point, the operation would be

$$x' = x + m, \qquad y' = y + n$$

where x', y' = coordinates of the

translated point x, y = coordinates

of the original point

m, n = movements in the x and y directions,

respectively In matrix notation this can be

represented as

$$(x', y') = (x, y) + T$$

where

$T = (m, n)$, the translation matrix

Any geometric element can be translated in space by applying Eq. to each point that defines the element. For a line, the transformation matrix would be applied to its two end points.

SCALING. Scaling of an element is used to enlarge it or reduce its size. The scaling need not necessarily be done equally in the x and y directions. For example, a circle could be transformed into an ellipse by using unequal x and y scaling factors.

The points of an element can be scaled by the scaling matrix as follows: $(x', y') = (x, y)$

where

$$s = \begin{bmatrix} m & 0 \\ 0 & n \end{bmatrix} \quad \text{the scaling matrix}$$

This would produce an alteration in the size of the element by the factor m in the x-direction and by the factor n in the y direction. It also has the effect of repositioning the element with respect to the cartesian system origin. If the scaling factors are less than 1, the size of the element is reduced and it is moved closer to the origin. If the scaling factors are larger than 1, the element is enlarged and removed farther from the origin.

ROTATION. In this transformation, the points of an object are rotated about the origin by an angle θ . For a positive angle, this rotation is in the counterclockwise direction. This accomplishes rotation of the object by the same angle, but it also moves the object. In matrix notation, the procedure would be as follows:

$$(x', y') = (x, y)R$$

where

$$R = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

the rotation matrix

EXAMPLE

As an illustration of these transformations in two dimensions, consider the line defined by

$$L = \begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix}$$

Let us suppose that it is desired to translate the line in space by 2 units in the x direction and 3 units in the y direction. This would involve adding 2 to the current x value and 3 to the current y value of the end points defining the line. That is,

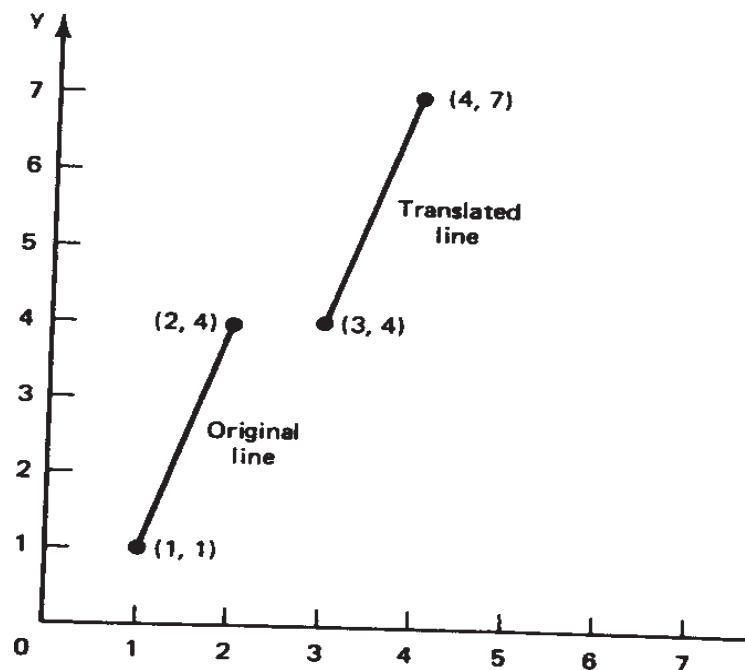


FIGURE.

Results of translation in Example

$$2 \begin{bmatrix} 1 & 1 \\ 4 & \end{bmatrix} + 2 \begin{bmatrix} 2 & 3 \\ 3 & \end{bmatrix} = 4 \begin{bmatrix} 3 & 4 \\ 7 & \end{bmatrix}$$

The new line would have end points at (3, 4) and (4, 7). The effect of the transformation is illustrated in Figure.

EXAMPLE

For the same original line as in Example 6.1, let us apply the scaling factor of 2 to the line. The scaling matrix for the 2 x 2 line definition would therefore be

$$T = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

The resulting line would be determined by Eq. as follows:

$$\frac{1}{2} \begin{bmatrix} 1 & 1 \\ 4 & - \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 2 & 2 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 2 & 4 \\ 8 & \end{bmatrix}$$

The new line is pictured in Figure.

EXAMPLE

We will again use our same line and rotate the line about the origin by 30°. Equation would be used to determine the transformed line where the rotation matrix would be:

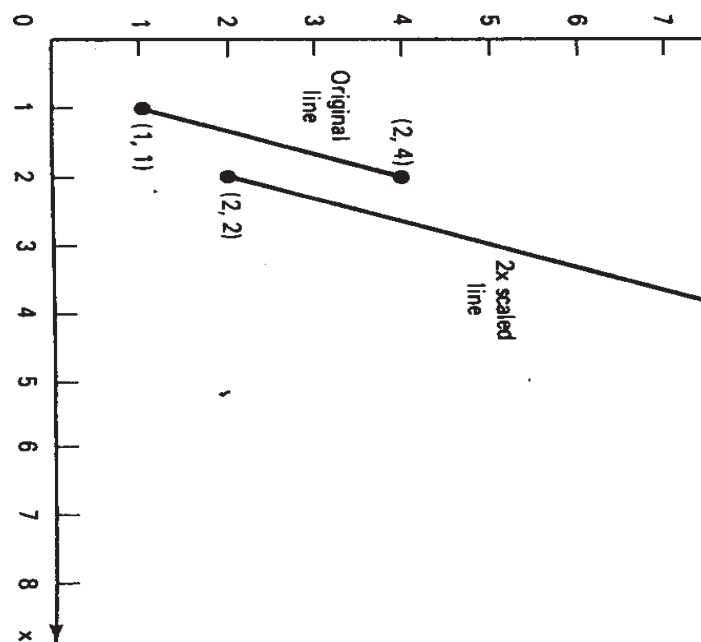


Figure Results of scaling in Example .

$$R = \begin{bmatrix} \cos 30 & \sin 30 \\ \sin 30 & \cos 30 \end{bmatrix} = \begin{bmatrix} 0.866 & 0.500 \\ 0.500 & 0.866 \end{bmatrix}$$

The new line would be defined as:

$$\begin{bmatrix} 1 & 1 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 0.866 & 0.500 \\ -0.500 & 0.866 \end{bmatrix} = \begin{bmatrix} 0.366 & 1.366 \\ -0.268 & 4.464 \end{bmatrix}$$

The effect of applying the rotation matrix to the line is shown in Figure.

Three-dimensional transformations

Transformations by matrix methods can be extended to three-dimensional space. We consider the same three general categories defined in the preceding section. The same general procedures are applied to use these transformations that were defined for the three cases by Eqs. TRANSLATION. The translation matrix for a point defined in three dimensions would be and would be applied by adding the increments m, n, and p to the respective coordinates of each of the points defining the three-dimensional geometry element.

$$T = (m, n, p)$$

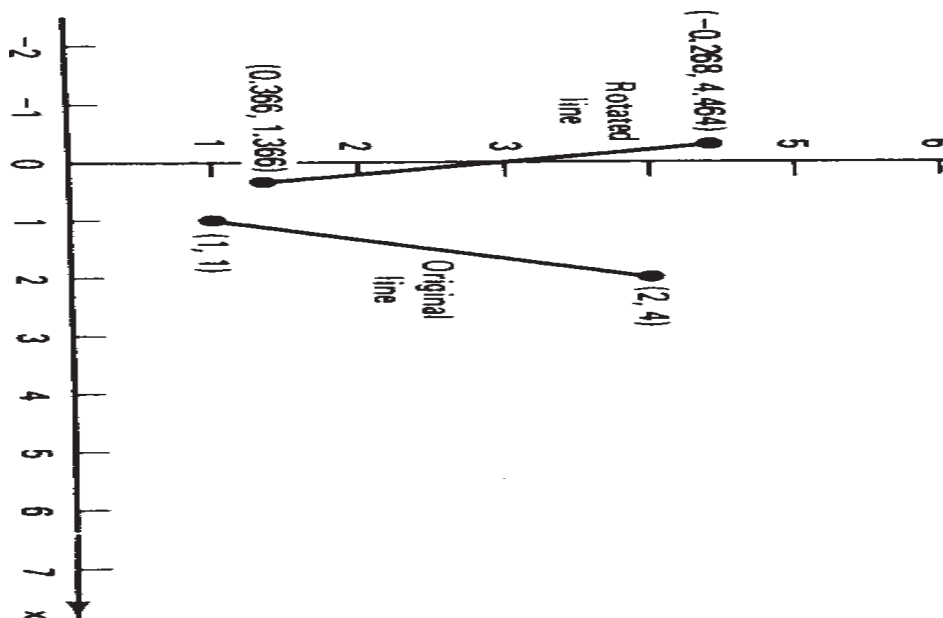


FIGURE Results of rotation in Example

SCALING. The scaling transformation is given by

$$m \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & p \end{bmatrix} S = \begin{bmatrix} 0 & n & 0 \end{bmatrix}$$

For equal values of m, n, and p, the scaling is linear.

ROTATION. Rotation in three dimensions can be defined for each of the axes.

Rotation about the z axis by an angle θ is accomplished by the matrix

$$R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation about the y axis by the angle θ is accomplished similarly.

$$R_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

Rotation about the x axis by the angle θ is done with an analogous transformation matrix.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

Concatenation

The previous single transformations can be combined as a sequence of transformations. This is called concatenation, and the combined transformations are called concatenated transformations.

During the editing process when a graphic model is being developed, the use of concatenated transformations is quite common. It would be unusual that only a single transformation would be needed to accomplish a desired manipulation of the image. Two examples of where combinations of transformations would be required would be: -

Rotation of the element about an arbitrary point in the element

Magnifying the element but maintaining the location of one of its points in the same location

In the first case, the sequence of transformations would be' translation to the origin, then rotation about the origin, then translation back to the original location. In the second case, the element would be scaled (magnified) followed by a translation to locate the desired point as needed:-

The objective of concatenation is to accomplish a series of image manipulations as a single-transformation. This allows the concatenated transformation to be defined more concisely and the computation can generally be accomplished more efficiently.

Determining the concatenation of a sequence of single transformations can be fairly straightforward if the transformations are expressed in matrix form as we have done. For example. if we wanted to scale a point by the factor of 2 in a two dimensional system and then rotate it about the origin by 45° , the concatenation would simply be the product of the two transformation matrices. It is important that the order of matrix multiplication be the same as the order in which the transformations are to be carried out. Concatenation of a series of transformations becomes more complicated when a translation is involved, and we will not consider this case.

EXAMPLE

Let us consider the example cited in the text in which a point was to be scaled by a factor of 2 and rotated by 45° . Suppose that the point under consideration was (3, 1). This might be one of several points defining a geometric element. For purposes of illustration let us first accomplish the two transformations sequentially. First, consider the scaling.

$$(x', y') = (x, y)S$$

$$\begin{matrix} (x', y') = \\ (3, 1) \end{matrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = (6, 2)$$

Next, the rotation can be

performed. $(x'', y'') =$

$$(x', y')R$$

$$= (6, 2) \begin{bmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}$$

$$= (2.828, 5.657)$$

The same result can be accomplished by concatenating the two separate transformation matrices. The product of the two matrices would be

$$SR = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}$$

$$= \begin{bmatrix} 1.414 & 1.414 \\ -1.414 & 1.414 \end{bmatrix}$$

Now, applying this concatenated transformation matrix to the original point, we have

$$\begin{bmatrix} 1.414 & 1.414 \\ -1.414 & 1.414 \end{bmatrix} (x'', y'') = (3, 1) \begin{bmatrix} 1.414 & 1.414 \\ -1.414 & 1.414 \end{bmatrix}$$

$$= (2.828, 5.657)$$

WIRE-FRAME VERSUS SOLID MODELING

The importance of three-dimensional geometry

Early CAD systems were basically automated drafting board systems which displayed a two-dimensional representation of the object being designed. Operators (e.g., the designer or drafter) could use these graphics systems to develop the line drawing the way they wanted it and then obtain a very high quality paper plot of the drawing. By using these systems, the drafting process could be accomplished in less time, and the productivity of the designers could be improved.

However, there was a fundamental shortcoming of these early systems. Although they were able to reproduce high-quality engineering drawings efficiently and quickly, these systems stored in their data files a two-dimensional record of the drawings. The drawings were usually of three-dimensional objects and it was left to the human beings who read these drawings to interpret the three-dimensional shape from the two-dimensional representation. The early CAD systems were not capable of interpreting the three-dimensionality of the object. It was left to the user of the system to make certain that the two-dimensional representation was correct (e.g., hidden lines removed or dashed, etc.), as stored in the data files.

More recent computer-aided design systems possess the capability to define objects in three dimensions. This is a powerful feature because it allows the designer to develop a full three-dimensional model of an object in the computer rather than a two-dimensional illustration. The computer can then generate the orthogonal views, perspective drawings, and close-ups of details in the object.

The importance of this three-dimensional capability in interactive computer graphics should not be underestimated.

Wire-Frame models

Most current day graphics systems use a form of modeling called wire- frame modeling. In the construction of the wire-frame model the edges of the objects are shown as lines. For objects in which there are curved surfaces, contour lines can be added; as shown in Figure, to indicate the contour. The image assumes the appearance of a frame constructed out of wire - hence the name “wire frame” model.

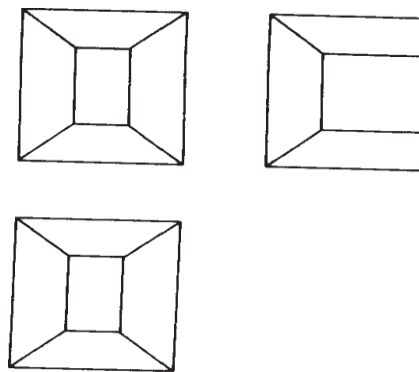


FIGURE Orthographic views of three-dimensional object without hidden- line removal.

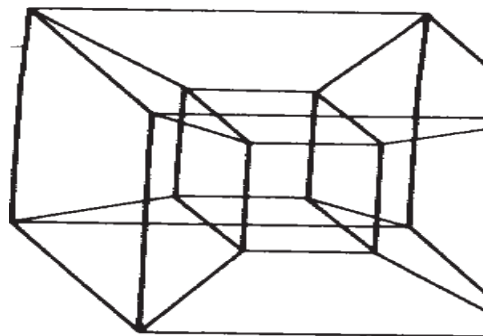


FIGURE Perspective view of three-dimensional object of Figure without hidden line removal

There are limitations to the models which use the wire-frame approach to form the image. These limitations are, of course, especially pronounced in the case of three-dimensional objects. In many cases, wire-frame models are quite adequate for two-dimensional representation. The most conspicuous limitation is that all of the lines that define the edges (and contoured surfaces) of the model are shown in the image. Many three-dimensional wire-frame systems in use today do not possess an automatic hidden-line removal feature. Consequently, the lines that indicate the edges at the rear of the model show right through the foreground surfaces. This can cause the image to be somewhat confusing to the viewer, and in some cases the image might be interpretable in several different ways. This interpretation problem can be alleviated to some extent through human intervention in removing the hidden background lines in the image.

There are also limitations with the wire-frame models in the way many CAD systems define the model in their data bases. For example, there might be ambiguity in the case of a surface definition as to which side of the surface is solid. This type of limitation prevents the computer system from achieving a comprehensive and unambiguous definition of the object.

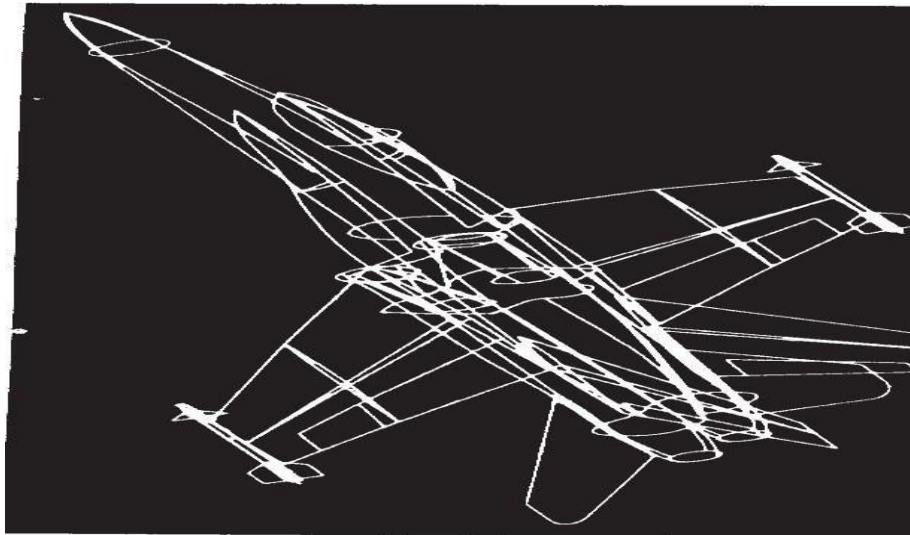


FIGURE Wireframe model of F/A-18 fighter aircraft, showing primary control curves.

Solid models

An improvement over wire-frame models, both in terms of realism to the user and definition to the computer, is the solid modeling approach. In this approach, the models are displayed as solid objects to the viewer, with very little risk of misinterpretation. When color is added to the image, the resulting picture becomes strikingly realistic. It is anticipated that graphics systems with this capability will find a wide range of applications outside computer-aided design and manufacturing. These applications will include color illustrations in magazines and technical publications, animation in movie films, and training simulators (e.g., aircraft pilot training).

There are two factors which promote future widespread use of solid modelers (i.e., graphics systems with the capability for solid modeling). The first is the increasing awareness among users of the limitations of wire-frame systems. As powerful as today's wire-frame-based CAD systems have become, solid model systems represent a dramatic improvement in graphics technology. The second reason is the continuing development of computer hardware and software which make solid modeling possible. Solid modelers require a great deal of computational power, in terms of both speed and memory, in order to operate. The advent of powerful, low-cost minicomputers has supplied the needed capacity to meet this requirement. Developments in software will provide application programs which take advantage of the opportunities offered by solid modelers. Among the possibilities are more highly automated model building and design systems, more complete three-dimensional engineering analysis of the models, including interference checking, automated manufacturing planning, and more realistic production simulation models.

Two basic approaches to the problem of solid modeling have been developed:

1. Constructive solid geometry (CSG or C-rep), also called the building-block approach
2. Boundary representation (B-rep)

The CSG systems allow the user to build the model out of solid graphic primitives, such as rectangular blocks, cubes, spheres, cylinders, and pyramids. This building-block approach is similar to the methods described in Section 6.4 except that a solid three-dimensional representation of the object is produced. The most common method of structuring the solid model in the graphics data base is to use Boolean operations,

described in the preceding section and pictured in Figure.

The boundary representation approach requires the user to draw the outline or boundary of the object on the CRT screen using an electronic tablet and pen or analogous procedure. The user would sketch the various views of the object (front, side, and top, more views if needed), drawing interconnecting lines among the views to establish their relationship. Various transformations and other specialized editing procedures are used to refine the model to the desired shape. The general scheme is illustrated in Figure.

The two approaches have their relative advantages and disadvantages. The C-rep systems usually have a significant procedural advantage in the initial formulation of the model. It is relatively easy to construct a precise solid model out of regular solid primitives by adding, subtracting, and intersecting the components. The building-block approach also results in a more compact file of the model in the database.

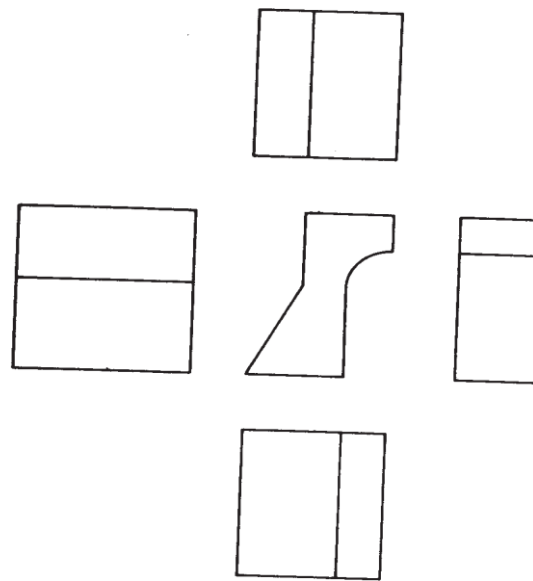


FIGURE Input views of the types required for boundary representation (B-rep).

On the other hand, B-rep systems have their relative advantages. One of them becomes evident when unusual shapes are encountered that would not be included within the available repertoire of the CSG systems. This kind of situation is exemplified by aircraft fuselage and wing shapes and by automobile body styling. Such shapes would be quite difficult to develop with the building-block approach, but the boundary representation method is very feasible for this sort of problem.

Another point of comparison between the two approaches is the difference in the way the model is stored in the data base for the two systems. The CSG approach stores the model by a combination of data and logical procedures.

(the Boolean model). This generally requires less storage but more computation to reproduce the model and its image. By contrast, the B-rep system stores an explicit definition of the model boundaries. This requires more storage space but does not necessitate nearly the same computation effort to reconstruct the image. A related benefit of the B-rep systems is that it is relatively simple to convert back and forth between a boundary representation and a corresponding wire-frame model. The reason is that the model's boundary definition is similar to the wire-frame definition, which facilitates conversion of one form to the other. This makes the newer solid B-rep systems compatible with existing CAD systems out in the field.

Because of the relative benefits and weaknesses of the two approaches, hybrid systems have been developed which combine the CSG and B-rep approaches. With these systems, users have the capability to construct the geometric model by either approach, whichever is more appropriate to the particular problem.

UNIT 2

Geometric modeling can be defined as computer compatible and mathematical representation of the geometry. If you really look at this particular definition, there are two aspects one is computer compatibility this is a must, second is a mathematical representation of geometry. If these two can be fulfilled in any definition I can call it is a geometric modeling.

Now one can always give a mathematical definition which is not computer compatible or I can have a representation which is used purely for visual representation of the geometric object. So if I look at these representations which are like which fulfills only one of the requirements then that is not what we are looking for. We are looking for something which can fulfill both. For example if I look at a mathematical definition to geometry, we study in our schools and colleges, a complete course on solid geometry or coordinate geometry which is basically concerned with mathematical representation of geometric entities like line, circle, conics also surface entities like cylindrical surface, conical surface or a spherical surface.

So we study about how these are mathematically represented in our coordinate geometric course or three dimensional solid geometric course. So these representations are mathematical but they are not really computer friendly that means what I intend to do with this geometry cannot be fulfilled by these definitions directly.

So I am looking for a better definition where I am able to use the mathematical representation also for doing some calculations which are related to cad cam or also to use this representation for visually displaying on let's say a computer screen. And if I go by other definition that is I can always give or I can always visually represent a three dimensional picture on computer screen without going into mathematical definition also, it's also possible. So, even that is not enough, so we are looking for something which is a combination of this thing.

Functions of Geometric Modeling

• **D e s i g n a n a l y s i s :**

- Evaluation of areas and volumes.
 - Evaluation of mass and inertia properties.
 - Interference checking in assemblies.
 - Analysis of tolerance build-up in assemblies.
 - Analysis of kinematics
 - mechanics, robotics.
 - Automatic mesh generation for finite element analysis.
-
- Geometric Models:
 - Two Dimensional and three dimensional.
 - The line model,
 - The surface model, and
 - The solid or volume model.

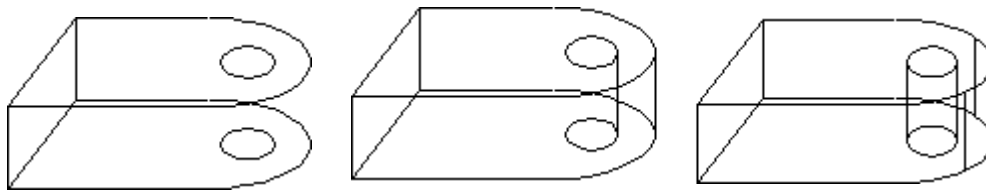
Curve representation

Wireframe models

There are three types of geometric models, wireframes, surfaces and solids. Typically, a wireframe model consists of a finite set of points (vertices), connected in pairs by straight lines (edges), or arcs, circles, conics, and curves, so that the three-dimensional form of a solid object can be visualized.

The major advantage of wireframe modeling is its simplicity to construct. It does not require as much computer time and memory as does surface or solid modeling. Wireframe modeling is considered a natural extension of traditional methods of drafting. Consequently, it does not require extensive training of users; nor does it demand the use of unusual terminology as surfaces and solids. Wireframe models form the basis for surface models. Most existing surface algorithms require wireframe entities to generate surfaces.

The disadvantages of wireframe models are manifolds. Primarily, wireframe models are usually ambiguous representations of real objects and rely heavily on human interpretation. A wireframe model of a box offers a typical example where the model may represent more than one object depending on which face(s) is assumed to exist. Models of complex designs having many edges become very confusing and perhaps even impossible to interpret. Moreover, as shown in Figure, it is often difficult to display objects with curve surfaces using wireframe



Parametric curves

Curve representation

Curves can be described mathematically by nonparametric or parametric equations. Nonparametric equations can be explicit or implicit. For a nonparametric curve, the coordinates y and z of a point on the curve are expressed as two separate functions of the third coordinate x as the independent variable [see Equation (1)]. This curve representation is known as the nonparametric explicit form. If the coordinates x , y and z are related together by two functions [see Equation (2)], a nonparametric implicit form results.

$$\mathbf{P} = [x \ y \ z]^T = [x \ f(x) \ g(x)]^T \quad (1)$$

$$\begin{aligned} F(x, y, z) &= 0 \\ G(x, y, z) &= 0 \end{aligned} \quad (2)$$

There are three problems with describing curves using nonparametric equations:

- (1) If the slope of a curve at a point is vertical or near vertical, its value becomes infinity or very large, a difficult condition to deal with both computationally and programming-wise. Other ill-defined mathematical conditions may result.
- (2) Shapes of most engineering objects are intrinsically independent of any coordinate system. What determines the shape of an object is the relationship between its data points themselves and not between these points and some arbitrary coordinate system.
- (3) If the curve is to be displayed as a series of point or straight-line segments, the computations involved could be extensive.

Parametric representation allows closed and multiple-valued functions to be easily defined and replaces the use of slopes with that of tangent vectors, as will be introduced shortly.

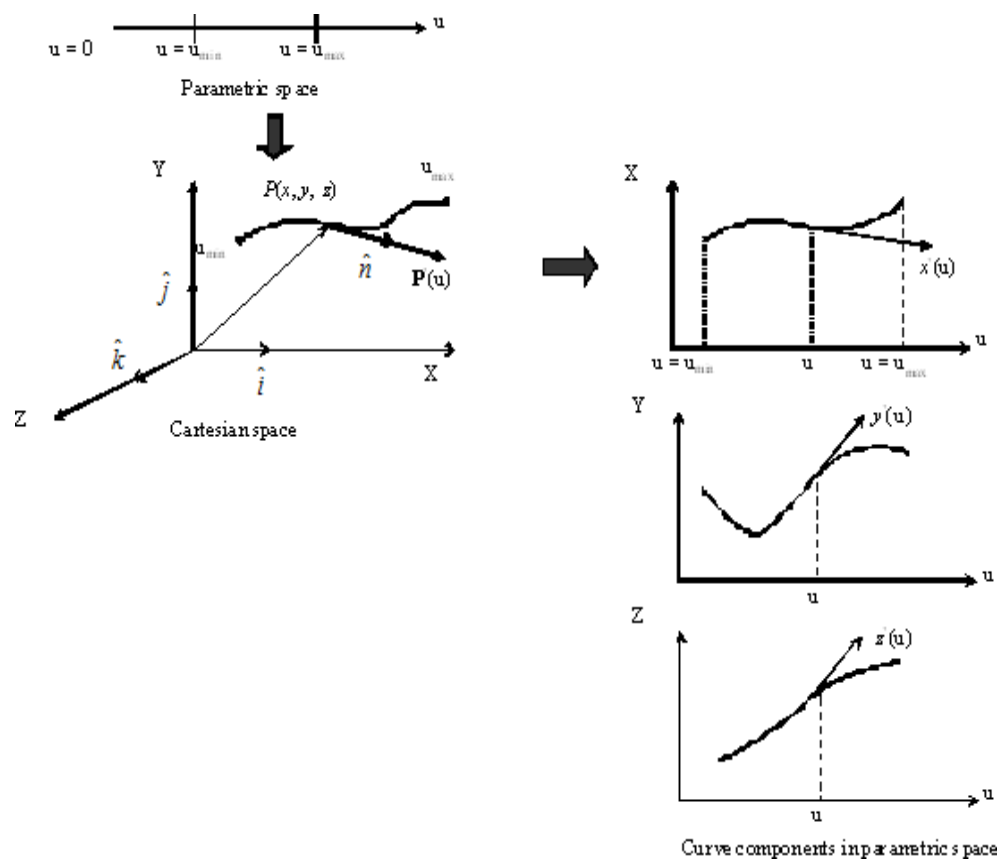
In parametric form, each point on a curve is expressed as a function of a parameter u . The parametric equation for a three-dimensional curve in space takes the following vector form:

$$\mathbf{P}(u) = [x \ y \ z]^T = [x(u) \ y(u) \ z(u)]^T, \quad u_{\min} \leq u \leq u_{\max} \quad (3)$$

Equation (3) implies that the coordinates of a point on the curve are the components of its position vector. It is a one-to-one mapping from the parametric space (Euclidean space E^1 in u values) to the Cartesian space (E^3 in x, y, z values), as shown in Figure 2.

Parametric representation of a three-dimensional curve

Parametric representation of a three-dimensional curve



Parametric representation of a three-dimensional curve

The tangent vector is defined as vector

$$\mathbf{P}'(u) = \frac{d\mathbf{P}(u)}{du} \quad (4)$$

The components of the tangent vector in the parametric space as

$$\mathbf{P}'(u) = [x' \quad y' \quad z']^T = [x'(u) \quad y'(u) \quad z'(u)]^T, \quad u_{min} \leq u \leq u_{max} \quad (5)$$

where $x'(u)$, $y'(u)$, and $z'(u)$ are the first parametric derivatives (with respect to u) of the position vector components $x(u)$, $y(u)$, and $z(u)$ respectively. The slopes of the curve are given by the ratios of the components of the tangent vector:

$$\begin{aligned} \frac{dy}{dx} &= \frac{dy/du}{dx/du} = \frac{y'}{x'} \\ \frac{dz}{dy} &= \frac{z'}{y'} \quad \text{and} \quad \frac{dx}{dz} = \frac{x'}{z'} \end{aligned} \quad (6)$$

Parametric representation of analytic curves

There are two categories of curves that can be represented parametrically: analytic curves and synthetic curves. Analytic curves are defined as those that can be described by analytic equations such as lines, circles, and conics. Synthetic curves are the ones that are described by a set of data points (control points) such as splines and Bezier curves.

Lines and circles are often expressed in analytic equations. They can also be expressed using parametric representation:

Lines

The parametric equation of the line becomes

$$\mathbf{P} = \mathbf{P}_1 + u(\mathbf{P}_2 - \mathbf{P}_1), \quad 0 \leq u \leq 1 \quad (7)$$

In scalar form, this equation can be written as

$$\left. \begin{aligned} x &= x_1 + u(x_2 - x_1) \\ y &= y_1 + u(y_2 - y_1) \\ z &= z_1 + u(z_2 - z_1) \end{aligned} \right\} \quad 0 \leq u \leq 1 \quad (8)$$

Note that Equation (7), a line is expressed as a synthetic curve.

◇ Assignment 2

Give the coordinates of two end points of a line. Write a program in Matlab to draw the x , y , z components in parametric space (hint: use Equation (8)) and the isometric view of this line in Cartesian space (hint: use the transformation matrix discussed in the previous lecture, or use the Matlab built-in command for isometric view), as in Figure 2.

Show your Matlab program too. ◇

Circles

Circles and circular arcs are among the most common entities used in wireframe modeling. Regardless of the user input information to create a circle, such information is always converted into a radius and center by the software.

The parametric equation of a circle can be written as

$$\left. \begin{aligned} x &= x_c + R \cos u \\ y &= y_c + R \sin u \\ z &= z_c \end{aligned} \right\}, \quad 0 \leq u \leq 2\pi \quad (9)$$

For display purposes, Equation (9) can be used to generate points on the circle circumference by incrementing u from 0 to 360 degrees. These points are in turn

connected with line segments to display the circles. However, this is an inefficient way due to computing the trigonometric functions in the equation for each point. A less computational method is to write Equation (9) in an incremental form.

$$\begin{aligned}
 x_n &= x_c + R \cos u \\
 y_n &= y_c + R \sin u \\
 x_{n+1} &= x_c + R \cos(u + \Delta u) \\
 y_{n+1} &= y_c + R \sin(u + \Delta u) \\
 z_{n+1} &= z_n
 \end{aligned} \tag{10}$$

Expanding the ~~Equation~~ ~~Equation~~ gives

$$\begin{aligned}
 x_{n+1} &= x_c + (x_n - x_c) \cos \Delta u - (y_n - y_c) \sin \Delta u \\
 y_{n+1} &= y_c + (y_n - y_c) \cos \Delta u + (x_n - x_c) \sin \Delta u \\
 z_{n+1} &= z_n
 \end{aligned} \tag{11}$$

Thus, the circle can start from an arbitrary point and successive points with equal spacing can be calculated recursively. The increments ~~cos~~ Δu and ~~sin~~ Δu have to be calculated only once, which eliminates computation of trigonometric functions for each point.

◇ Assignment 3

Derive Equation (9) to Equation (11) in more details. Decide a value ~~of~~ Δu and calculate ~~cos~~ Δu and ~~sin~~ Δu . Write a Matlab program to generate the points of a circle using Equation (11), then connect the points to draw a circle. Try 3 different ~~and~~ Δu

show the figures generated by your Matlab program. Show your Matlab program too. ◇

Synthetic curves

The need for synthetic curves

The need for synthetic curves in design arises on two occasions: when a curve is represented by a collection of measured data points and when an existing curve must change to meet new design requirements. Analytic curves are usually not sufficient to meet geometric design requirements of mechanical parts. Synthetic curves provide designers with greater flexibility and control of a curve shape by changing the positions of the control points. Products such as car bodies, ship hulls, airplane fuselage and wings, propeller blades, shoe insoles, and bottles are a few examples that require free-form, or synthetic, curves and surfaces. Considering that most data of objects are available in a discrete form, mainly key points, the curve equation should be able to accept points and/or tangent values as input from the designer.

Splines draw their name from the traditional draftsman's tool called "French curves or splines." In drafting terminology, a spline is a flexible strip used to produce a smooth curve

through a designated set of points. Several small weights are distributed along the length of the strip to hold it in position on the drafting table. The term spline curve referred to a curve drawn in this manner.

We specify a spline curve by giving a set of coordinate positions, call control points, which indicate the general shape of the curve. These control points are then fitted with piecewise continuous parametric polynomial functions in one of the two ways. When polynomial sections are fitted so that the curve passes through each control point, the resulting curve is said to interpolate the set of control points. On the other hand, when the polynomials are fitted to the general control point path without necessarily passing through any control point, the resulting curve is said to approximate the set of control points.

Interpolation curves are commonly used to digitize drawings or to specify animation paths. Approximation curves are primarily used as design tools to structure object surfaces.

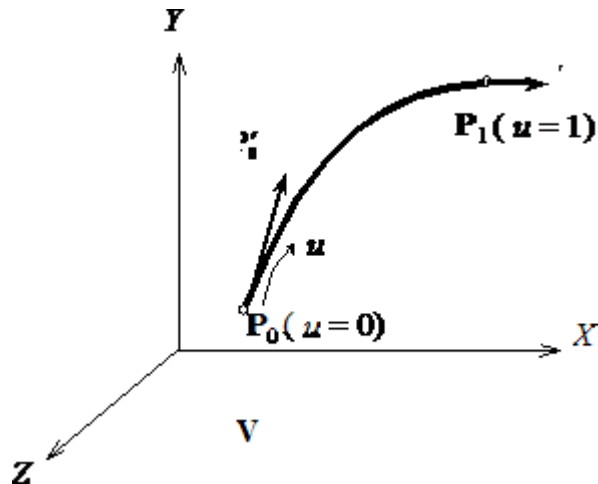
Mathematically, synthetic curves represent a curve-fitting problem to construct a smooth curve that passes through given data points. Zero-order continuity (C^0) yields a position continuous curve. First (C^1)- and second (C^2)-order continuities imply slope and curvature continuous curves respectively. A C^1 curve is the minimum acceptable curve for engineering design. A cubic polynomial is the minimum-order polynomial that can guarantee the generation of C^0 , C^1 or C^2 curves.

Higher-order polynomials are not commonly used in CAD systems because they tend to oscillate about control points, are computationally inconvenient, and are uneconomical of storing curve and surface representations in the computer.

Also, the designer may prefer to control the shape of the curve locally instead of globally by changing the control point(s). If changing a control point results in changing the curve locally in the vicinity of that point, local control of the curve is achieved; otherwise global control results.

Hermite Cubic Splines

The Hermite form of a cubic spline is determined by defining positions and tangent vectors at the data points, as shown in Figure



Hermite cubic spline curve

The parametric cubic spline curve (or cubic spline for short) connects two data (end) points and utilizes a cubic equation. Therefore, four conditions are required to determine the coefficients of the equation. The parametric equation of a cubic spline segment is given by

$$\mathbf{P}(u) = \sum_{i=0}^3 \mathbf{C}_i u^i, \quad 0 \leq u \leq 1 \quad (12)$$

where u is the parameter and C_i are the polynomial (also called algebraic) coefficients. In scalar form this equation is written as

$$\begin{aligned} x(u) &= C_{3x}u^3 + C_{2x}u^2 + C_{1x}u + C_{0x} \\ y(u) &= C_{3y}u^3 + C_{2y}u^2 + C_{1y}u + C_{0y} \\ z(u) &= C_{3z}u^3 + C_{2z}u^2 + C_{1z}u + C_{0z} \end{aligned} \quad (13)$$

In an expanded vector form, Equation (12) can be written as

$$\mathbf{P}(u) = \mathbf{C}_3 u^3 + \mathbf{C}_2 u^2 + \mathbf{C}_1 u + \mathbf{C}_0 \quad (14)$$

This equation can also be written in a matrix form as

$$\mathbf{P}(u) = \mathbf{U}^T \mathbf{C} \quad (15)$$

where $\mathbf{U} = [u^3 \quad u^2 \quad u \quad 1]^T$ and $\mathbf{C} = [C_3 \quad C_2 \quad C_1 \quad C_0]^T$, and C is called the coefficients vector. The tangent vector is

$$\mathbf{P}'(u) = \sum_{i=0}^3 \mathbf{C}_i u^{i-1}, \quad 0 \leq u \leq 1 \quad (16)$$

Now the problem is how to relate the parametric equations to the designers' input, namely, the two end points and tangent vectors. Applying the boundary conditions (P_0, P'_0 at $u=0$ and P_1, P'_1 at $u=1$), Equations (12) and (16) give

$$\begin{aligned} \mathbf{P}_0 &= \mathbf{C}_0 \\ \mathbf{P}'_0 &= \mathbf{C}_1 \\ \mathbf{P}_1 &= \mathbf{C}_3 + \mathbf{C}_2 + \mathbf{C}_1 + \mathbf{C}_0 \\ \mathbf{P}'_1 &= 3\mathbf{C}_3 + 2\mathbf{C}_2 + \mathbf{C}_1 \end{aligned} \quad (17)$$

$$\begin{aligned} \mathbf{C}_0 &= \mathbf{P}_0 \\ \mathbf{C}_1 &= \mathbf{P}'_0 \\ \mathbf{C}_2 &= 3(\mathbf{P}_1 - \mathbf{P}_0) - 2(\mathbf{P}'_0 + \mathbf{P}'_1) \\ \mathbf{C}_3 &= 2(\mathbf{P}_0 - \mathbf{P}_1) + \mathbf{P}'_0 + \mathbf{P}'_1 \end{aligned} \quad (18)$$

Substituting Equation (18) into Equation (14) and rearranging gives

$$\mathbf{P}(u) = (2u^3 - 3u^2 + 1)\mathbf{P}_0 + (-2u^3 + 3u^2)\mathbf{P}_1 + (u^3 - 2u^2 + u)\mathbf{P}'_0 + (u^3 - u^2)\mathbf{P}'_1 \quad 0 \leq u \leq 1 \quad (19)$$

$$\mathbf{P}(u) = \mathbf{U}^T [\mathbf{M}_H] \mathbf{V}, \quad 0 \leq u \leq 1 \quad (20)$$

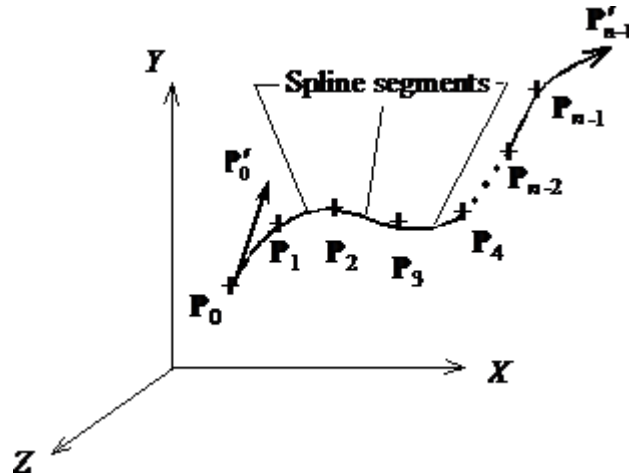
where $[\mathbf{M}_H]$ is the Hermite matrix and \mathbf{V} is the geometry (or boundary conditions) vector. Both are given by

$$[\mathbf{M}_H] = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (21)$$

$$\mathbf{V} = [\mathbf{P}_0 \quad \mathbf{P}_1 \quad \mathbf{P}'_0 \quad \mathbf{P}'_1]^T$$

Blending Cubic Spline Segments

Equation (19) is for one cubic spline segment. It can be generalized for any two adjacent spline segments of a spline curve that are to fit a given number of data points. This introduces the problem of blending or joining cubic spline segments which can be stated as follows: given a set of n points P_0, P_1, \dots, P_{n-1} and the two end tangent vectors P'_0 and P'_{n-1} (as shown in below Figure), connect the points with a cubic spline curve.



Hermite cubic spline curve

Let's start from connecting two curves. To connect two Hermite spline curves to form a C^2 continuous curve, the second derivative at the end of the first curve must be equal to the second derivative at the beginning of the first curve. Thus we have

$$\mathbf{P}'(\mu_1 = 1) = \mathbf{P}'(\mu_2 = 0) \quad (23)$$

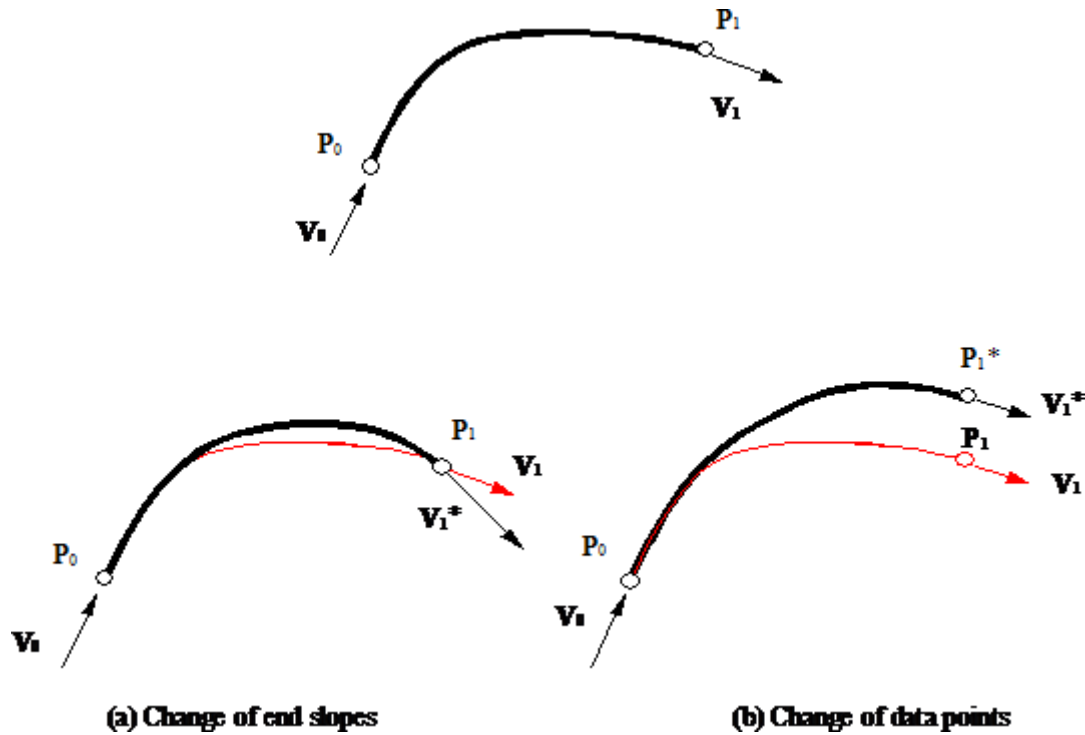
Using this relation, we can further derive the tangent vector at the end of the second curve, which is also the tangent vector at the beginning of the second curve:

$$\mathbf{P}'_1 = -\frac{1}{4}(3\mathbf{P}_0 + \mathbf{P}'_0 - 3\mathbf{P}_2 + \mathbf{P}'_2) \quad (24)$$

Using this information, we can construct a C^2 continuous curve that passes through the three given data points by blending two Hermite curves together. For N given data points, Equation (23) and (24) have to be used iteratively to obtain the tangent vector at the intermediate data points.

The use of the cubic splines in design applications is not very popular compared to Bezier or B-spline curves. The control of the curve is not very obvious from the input data due to its global control characteristics. As end slope changes the entire shape of the spline, which does not provide the intuitive feel required for design. In addition, the order of the curve is always constant (cubic) regardless of the number of data points.

shown in Figure, changing the position of a data point or an



Control of cubic spline curve

4 Lagrangian Interpolation Method

Often we have to display a curve interpolating through several points. Beside blending Hermite cubic spline segments, this can also be done by using the Lagrangian interpolation method. Equation (25) is the Lagrangian four-point form. It is easy to verify that at $u=0, 1/3, 2/3, 1$, the curve interpolates through P_0, P_1, P_2 , and P_3 .

$$\begin{aligned}
 \mathbf{P}(u) = & \frac{1}{2}(1-u)(2-3u)(1-3u)\mathbf{P}_0 \\
 & + \frac{9u}{2}(1-u)(2-3u)\mathbf{P}_1 \\
 & + \frac{9u}{2}(1-u)(3u-1)\mathbf{P}_2 \\
 & + \frac{u}{2}(2-3u)(1-3u)\mathbf{P}_3, \quad 0 \leq u \leq 1
 \end{aligned} \tag{25}$$

Surface Representation

As mentioned above, the high complexity of the mathematical description of various surfaces complicates seriously the meshing procedure with surface constraint. Therefore the proper surface representation is essential for the efficiency of the method. To make the algorithm applicable to a wide variety of surfaces with a reasonable level of complexity the family of tensor product polynomial surfaces¹(e.g., rational Bezier surface, B-spline surface, NURBS, etc.) is considered in the presented approach. This brings in several advantages. Since these surfaces are widely used in CAD and modeling systems a natural and consistent interface (at least in terms of geometric description of the surface) between the modeler and the mesh generator is ensured. All these surfaces enable relatively simple evaluation of surface normal and gradients at discrete locations on the surface. And finally, the parametric space of these surfaces allows for some procedures to be performed more efficiently than in 3D.

In the presented study, only the set of rational Bezier surfaces has been implemented. Each surface to be discredited is bounded by four rational Bezier curves from which at most two (but not the two adjacent ones) may be collapsed into a single point. This allows for modeling of degenerated surfaces often used as transition patches. This simple data structure, which can be easily extracted from modeler allowing to handle rational Bezier entities, may be extended in terms of additional curves (again, reducible to a single point) fixed to the surface, for example to specify the mesh density source or to constrain the tri

Modeling Facilities Desired:

The geometric modeling features.

The editing or manipulation features

The display control features

The drafting features

The programming facilities

The analysis features

The connecting features angulation.

UNIT 3

INTRODUCTION TO CAD

CAD (Computer Aided Design) is the use of computer software to design and document a product's design process. Engineering drawing entails the use of graphical symbols such as points, lines, curves, planes and shapes. Essentially, it gives detailed description about any component in a graphical form.

CAD software enables

- Efficiency in the quality of design
- Increase in the Engineer's productivity
- Improve record keeping through better documentation and communication
- Today, the use of CAD has penetrated almost all industries. From aerospace, electronics to manufacturing, CAD is used in all industry verticals. Since CAD encourages creativity and speeds up productivity, it is becoming more and more useful as an important tool for visualization before actually implementing a manufacturing process.

Types of CAD Software

Since its introduction in late 1960's, CAD software has improved by leaps and bounds. A broad classification of CAD is:

- 2D CAD
- 3D CAD
- 3D Wireframe and Surface Modeling
- Solid Modeling

HISTORY OF AUTOCAD2016

AutoCAD is a software application developed by Autodesk that enables computer-aided design (CAD) and drafting.

The first version released in AutoCAD was AutoCAD Version 1.0 in the year 1982. It has totally 32 versions since it was introduced in the year 1982 and the latest version is AutoCAD2017.

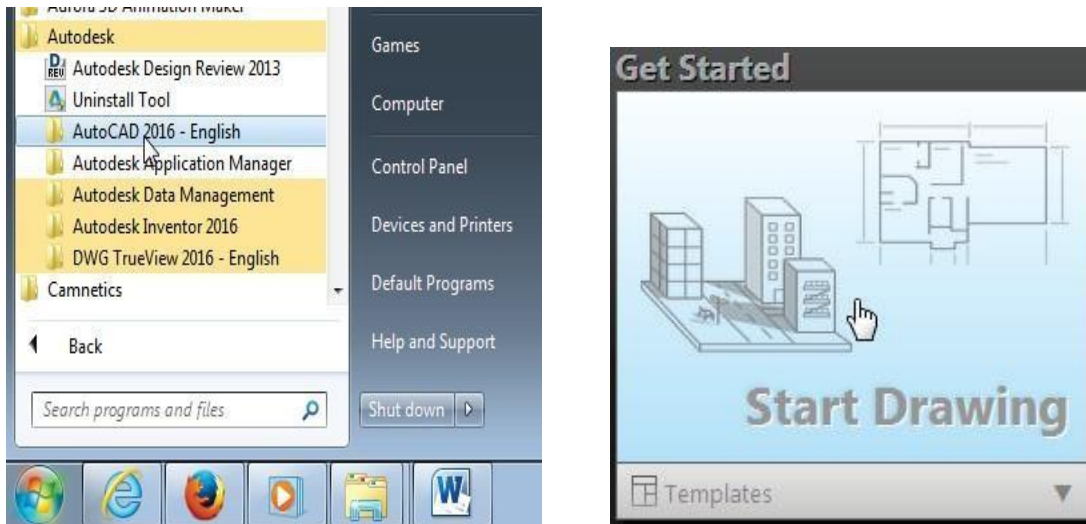
OPENING A NEW FILE IN AUTOCAD2016

Double Click The Autocad2016 Icon On Desktop

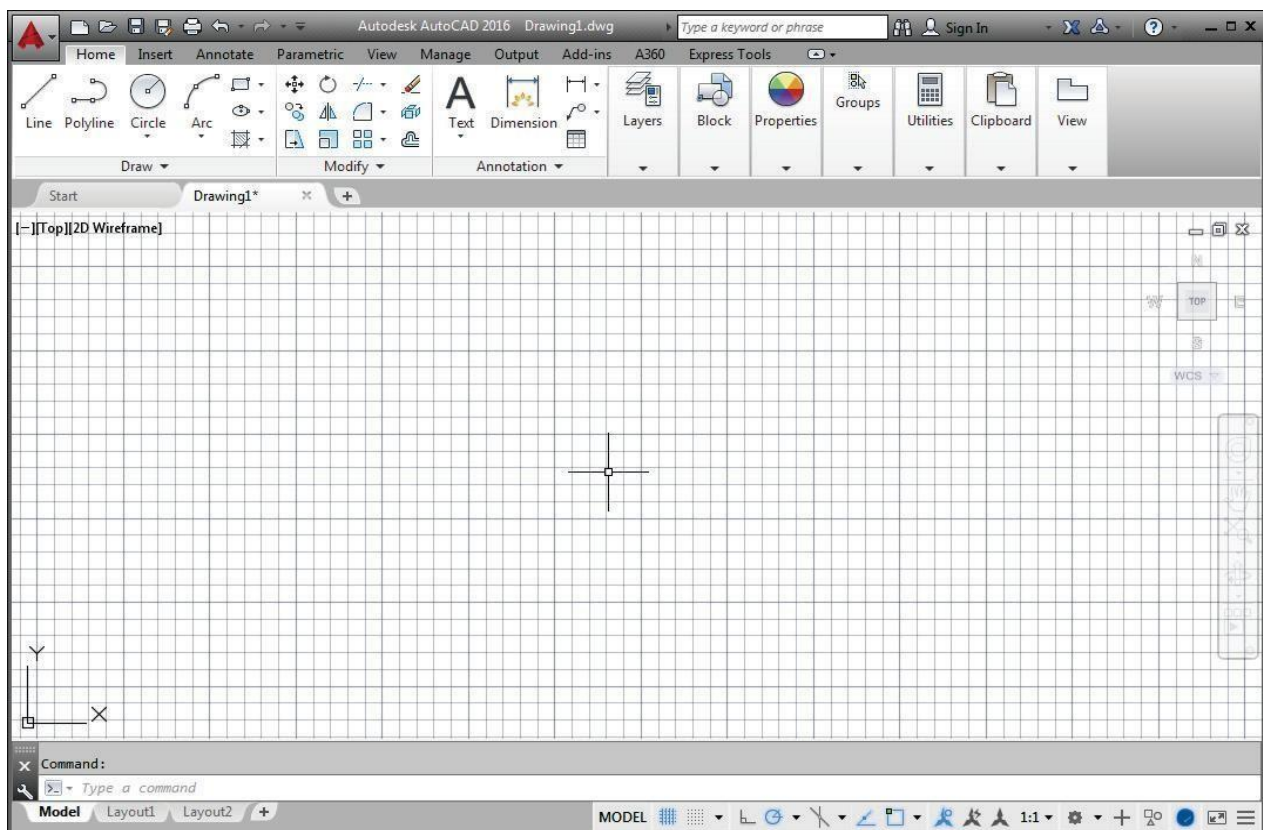
(Or)

Go To Start Button And Select The Autocad2016 Application.

Then Click “Start Drawing” To Start A New Drawing



Once the program is loaded the **AutoCAD 2016 main** drawing screen will appear on the screen.

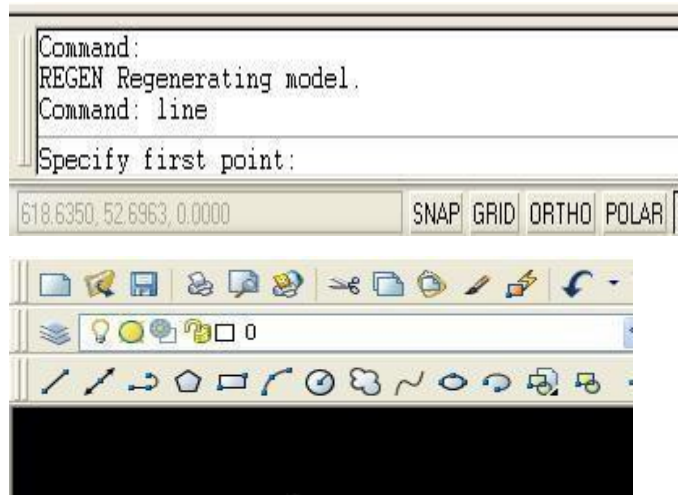


- Extension for autocad files is .dwg
- The files can be saved by clicking the “save” button available in “file” menubar or by pressing “ctrl+s”.
- Existing files can be opened by clicking the “open” button or by pressing “ctrl+o”.
- The default file name is “drawing1.dwg”.
- We can change the file name according to our requirement.
- The autocad menu bar contains multiple pull-down menus where all of the autocad commands can be accessed. Note that many of the menu items listed in the pull-down menus can also be accessed through the quick access toolbar and/or ribbon panels.

How do we give a command?

- Command line
- Toolbars
(view/Toolbars)
- Drop-down menus

You can pick any one that you are comfortable with



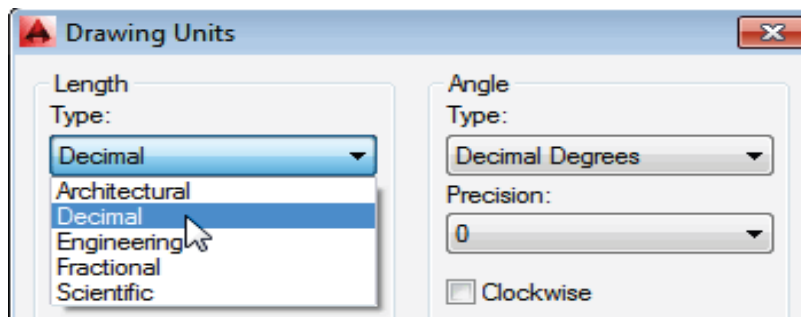
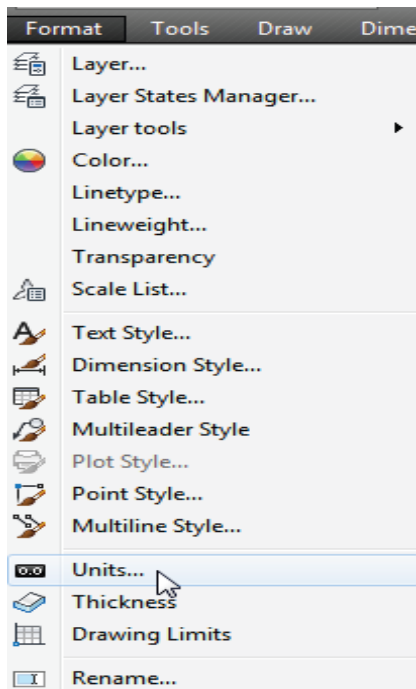
Drawing Units Setup

Every object we construct in a CAD system is measured in units.

In the Menu Bar select: [Format] → [Units]

2. Click on the *Length Type* option to display the different types of length units available.

Confirm the *Length. Type* is **Decimal**.



.On your own, examine the other settings that are available.

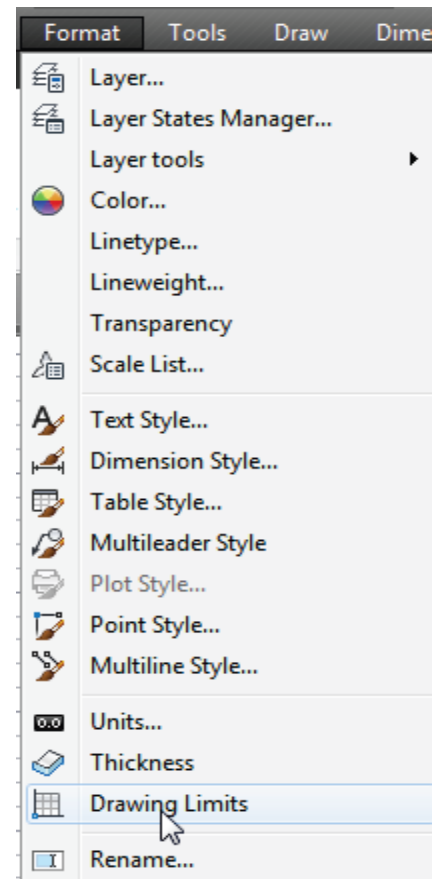
. In the *Drawing Units* dialog box, set the *Length Type* to *Decimal*. This will set the measurement to the default *English units, inches*.

Set the *Precision* to **two digits** after the decimal point.

Pick **OK** to exit the *Drawing Units* dialog box.

Drawing Area Setup

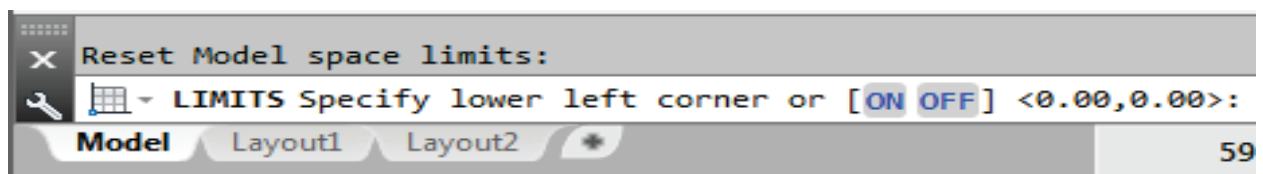
- ⦿ Setting the Drawing Limits controls the extents of the display of the *grid*. *It also serves as a visual reference that marks the working area*. It can also be used to prevent construction outside the grid limits and as a plot option that defines an area to be plotted/printed.



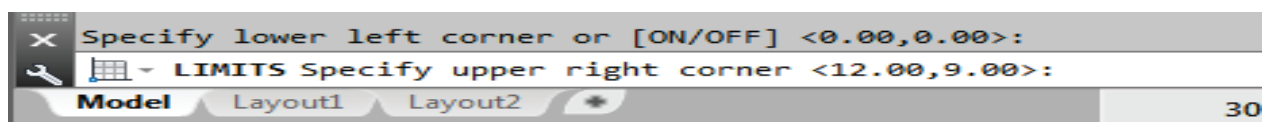
In the *Menu Bar* select:

[Format] → [Drawing Limits]

2. In the command prompt area, the message “*Reset Model Space Limits: Specify lower left corner or [On/Off]<0.00,0.00>:*” is displayed. Press the **ENTER** key *once* to accept the default coordinates **<0.00,0.00>**.



In the command prompt area, the message “*Specify upper right corner<12.00,9.00>:*” is displayed. Press the **ENTER** key *again* to accept the default coordinates **<12.00,9.00>**.



IMPORTANCE OF LAYERS

- ☉ The Layer command is a very important tool of AutoCAD. It lets you create layers to separate your drawing objects.

- Layers are used to group, change the visibility and to control objects.

Each layer you create has its own associated properties that are unique to that layer

- A list of things you can do using layers:
 - ▢ Control objects by locking and unlocking layers so they are not accidentally modified.
 - ▢ Allow layers to be plotted or not.
 - ▢ Create as many layers as you want.
 - ▢ Group layers by colors, line types, names, etc... Also, you can create a group filter inside the layer manager that can be saved within the drawings so other sharing users can use it later.
 - ▢ Isolate layers and objects with the same properties in the drawing space. For example, you want to show only walls and hide all doors.
 - ▢ Change the visibility of drawing layouts.
 - ▢ Turn on or off, freeze or thaw layers in model space or in the viewport layouts in paper space.
 - ▢ Globally change object properties.

Steps to create a layer

1. Type LAYER in the command line or select from the layer toolbar.
2. By default AutoCAD will name the layer “layer1”.
3. Double click and rename it.
4. In the right column you can assign color, line type to the layer, freeze or thaw it.
5. Select the object in the drawing to which the layer will be assigned.

Click on the layer icon bar and select the appropriate layer




















FREQUENTLY USED TOOLBARS

The following are some of the frequently used toolbars in autocad2016.

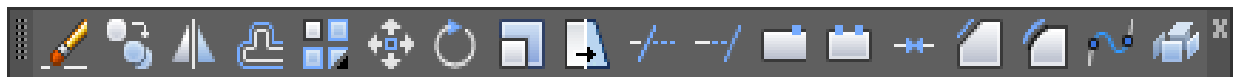
- Draw toolbar
- Modify toolbar
- Layer properties toolbar
- Dimension toolbar

DRAW TOOLBAR



-  Line
-  Construction line
-  Poly line
-  Polygon
-  Rectangle
-  Arc
-  Circle
-  Revision cloud
-  Spline
-  Ellipse
-  Elliptical arc
-  Insert block
-  Make block
-  Point
-  Hatch
-  Gradient
-  Region
-  Table
-  Multiline text

Modify toolbar



- ☒ Erase
- ☒ Copy
- ☒ Mirror
- ☒ Offset
- ☒ Array

- Move
- Rotate
- Scale
- Stretch
- Trim
- Extend
- Break at point
- Break
- Join
- Chamfer
- Fillet
- Blend curves

Explode

Numerical control defined

Numerical control can be defined as a form of programmable automation in which the process is controlled by numbers, letters, and symbol. In NC, the numbers form a program of instructions designed for a particular work part or job. When the job changes, the program of instructions is changed. This capability to change the program for each new job is what gives NC its flexibility. It is much easier to write new programs than to make major changes in the production equipment.

NC technology has been applied to a wide variety of operations, including drafting, assembly, inspection, sheet metal press working, and spot welding. However, numerical control finds its principal applications in metal machining processes. The machined work parts are designed in various sizes and shapes, and most machined parts produced in industry today are made in small to medium-size batches. To produce each part, a sequence of drilling operations may be required, or a series of turning or milling operations. The suitability of NC for these kinds of jobs is the reason for the tremendous growth of numerical control in the metal-working industry over the last 25 years.

BASIC COMPONENTS OF AN NC SYSTEM

An operational numerical control system consists of the following three basic components:

1. Program of instructions
2. Controller unit, also called a machine control unit (MCU)
3. Machine tool or other controlled process

The general relationship among the three components is illustrated in Figure. The program of instructions serves as the input to the controller unit, which in turn commands the machine tool or other process to be controlled. We will discuss the three components in the sections below.

Program of instructions

The program of instructions is the detailed step-by-step set of directions which tell the machine tool what to do. It is coded in numerical or symbolic form on some type of input medium that can be interpreted by the controller unit. The most common input medium today is 1-in.-wide punched tape. Over the years, other forms of input media have been used, including punched cards, magnetic tape, and even 35- mm motion

picture film.

There are two other methods of input to the NC system which should be mentioned. The first is by manual entry of instructional data to the controller unit. This method is called manual data input, abbreviated MDI, and is appropriate only for relatively simple jobs where the order will not be repeated. The second other method of input is by means

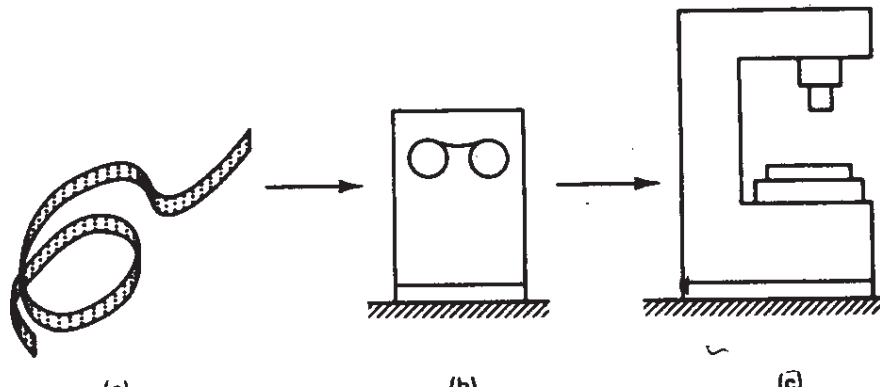


FIGURE Three basic components of a numerical control system: (a) program of instruction; (b) controller unit; (c) machine tool.

of a direct link with a computer. This is called direct numerical control, or DNC,.

The program of instructions is prepared by someone called a part programmer. The programmer's job is to provide a set of detailed instructions by which the sequence of processing steps is to be performed. For a machining operation, the processing steps involve the relative movement between the cutting tool and the workpiece.

Controller unit

The second basic component of the NC system is the controller unit. This

consists of the electronics and hardware that read and interpret the program of instructions and convert it into mechanical actions of the machine tool. The typical elements of a conventional NC controller unit include the tape reader, a data buffer signal out-put channels to the machine tool, feedback channels from the machine tool, and the sequence controls to coordinate the overall operation of the foregoing elements. It should be noted that nearly all modern NC systems today are sold with a microcomputer as the controller unit. This type of NC is called computer numerical control (CNC).

The tape reader is an electromechanical device for winding and reading the punched tape containing the program of instructions. The data contained on the tape are read into the data buffer. The purpose of this device is to store the input instructions in logical blocks of information. A block of information usually represents one complete step in the sequence of processing elements. For example, one block may be the data required to move the machine table to a certain position and drill a hole at that location.

The signal output channels are connected to the servomotors and other controls in the machine tool. Through these channels, the instructions are sent to the machine tool from the controller unit. To make certain that the instructions have been properly executed by the machine, feedback data are sent back to the controller via the feedback channels. The most important function of this return loop is to assure that the table and work part have been properly located with respect to the tool.

Sequence controls coordinate the activities of the other elements of the controller unit. The tape reader is actuated to read data into the buffer from the tape, signals are sent to and from the machine tool, and so on. These types of operations must be synchronized and this is the function of the sequence controls.

Another element NC system, which may be physically part of the controller unit or part of the machine tool, is the control panel. The control panel or control console contains the dials and switches by which the machine operator runs the NC system. It may also contain data displays to provide information to the operator. Although the NC system is an automatic system, the human operator is still needed to turn the machine on and off, to change tools (some NC systems have automatic tool changers), to load and unload the machine, and to perform various other duties. To be able to discharge these duties, the operator must be able to control the system, and this is done through the control panel.

Machine tool or other controlled process

The third basic component of an NC system is the machine tool or other controlled process. It is the part of the NC system which performs useful work. In the most common example of an NC system, one designed to perform machining operations, the machine tool consists of the workable and spindle as well as the motors and controls necessary to drive them. It also includes the cutting tools, work fixtures, and other auxiliary equipment needed in the machining operation.

NC machines range in complexity from simple tape-controlled drill presses to highly sophisticated and versatile machining centers. The NC machining center was first introduced in the late 1950s. It is a multifunction machine which incorporates several timesaving features into a single piece of automated production equipment. First, a machining center is capable of performing a variety of different operations: drilling, tapping, reaming, milling, and boring. Second, it has the capacity to change tools automatically under tape command. A variety of machining operations means that a variety of cutting tools are required. The tools are kept in a tool drum or other holding device. When the tape calls a particular tool, the drum rotates to position the tool for insertion into the spindle. The automatic tool changer then grasps the tool and places it into the spindle chuck. A third capability of the NC machining center is work piece positioning. The machine table can orient the job so that it can be machined on several surfaces, as required. Finally, a fourth feature possessed by some machining centers is the presence of two tables or pallets on which the work piece can be fixtured. While the machining sequence is being performed on one work part, the operator can be unloading the previously completed piece, and loading the next one. This improves machine tool utilization because the machine does not have to stand idle during loading and unloading of the work parts.

THE NC PROCEDURE

To utilize numerical control in manufacturing, the following steps must be accomplished.

- 1 *Process Planning.* The engineering drawing of the workpart must be interpreted in terms of the manufacturing processes to be used. this step is referred to as process planning and it is concerned with the preparation of a route sheet. The route sheet is a listing of the sequence of operations which must be performed on the workpart. It is called a route sheet because it also lists the machines through which the part must be

routed in order to accomplish the sequence of operations. We assume that some of the operations will be performed on one or more NC machines.

2. *Part programming.* A part programmer plans the process for the portions of the job to be accomplished by NC. Part programmers are knowledgeable about the machining process and they have been trained to program for numerical control. They are responsible for planning the sequence of machining steps to be performed by NC and to document these in a special format. There are two ways to program for NC:

Manual part programming

Computer-assisted part programming

In manual programming, the machining instructions are prepared on a form called a part program manuscript. The manuscript is a listing of the relative cutter/work piece positions which must be followed to machine the part. In computer-assisted part programming, much of the tedious computational work required in manual part programming is transferred to the computer. This is especially appropriate for complex work piece geometries and jobs with many machining steps. Use of the computer in these situations results in significant savings in part programming time.

4. *Tape preparation.* A punched tape is prepared from the part programmer's NC process plan. In manual part programming, the punched tape is prepared directly from the part program manuscript on a typewriter like device equipped with tape punching capability. In computer-assisted part programming, the computer interprets the list of part programming instructions, performs the necessary calculations to convert this into a detailed set of machine tool motion commands, and then controls a tape punch device to prepare the tape for the specific NC machine.

5. *Tape verification.* After the punched tape has been prepared, a method is usually provided for checking the accuracy of the tape. Some times the tape is checked by running it through a computer program which plots the various tool movements (or table movements) on paper. In this way, major errors in the tape can be discovered. The "acid test" of the tape involves trying it out on the machine tool to make the part. A foam or plastic material is sometimes used for this tryout. Programming errors are not uncommon, and it may require about three attempts before the tape is correct and ready to use.

6. *Production.* The final step in the NC procedure to use the NC tape in

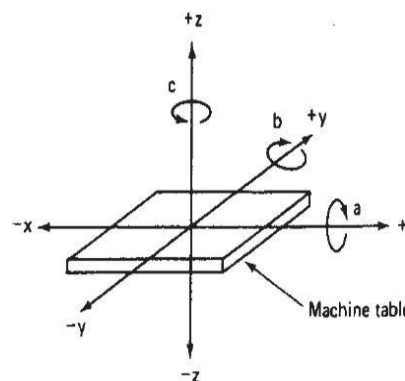
production. This involves ordering the raw workparts specifying and preparing the tooling and any special fixturing that may be required, and setting up The NC machine tool for the job. The machine tool operator's function during production is to load the raw workpart in the machine and establish the starting position of the cutting tool relative to the workpiece. The NC system then takes over and machines the part according to the instructions on tape. When the part is completed, the operator removes it from the machine and loads the next part.

NC COORDINATE SYSTEMS

In order for the part programmer to plan the sequence of positions and movements of the cutting tool relative to the workpiece, it is necessary to establish a standard axis system by which the relative positions can be specified. Using an NC drill press as an example, the drill spindle is in a fixed vertical position, and the table is moved and controlled relative to the spindle. However, to make things easier for the programmer, we adopt the viewpoint that the workpiece is stationary while the drill bit is moved relative to it. Accordingly, the coordinate system of axes is established with respect to the machine table.

Two axes, x and y , are defined in the plane of the table, as shown in Figure . The z axis is perpendicular to this plane and movement in the z direction is controlled by the vertical motion of the spindle. The positive and negative directions of motion of tool relative to table along these axes are as shown in Figure 7. A. NC drill presses are classified as either two-axis or three-axis machines, depending on whether or not they have the capability to control the z axis.

A numerical control milling machine and similar machine tools (boring mill. for example) use an axis system similar to that of the drill press. However, in addition to the three linear



axes, these machines may possess the capacity to control

FIGURE NC machine tool axis system for milling and drilling operations.

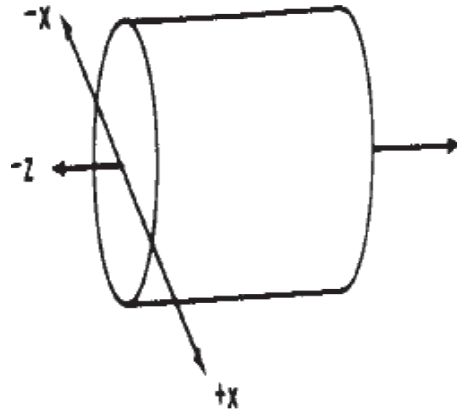


FIGURE NC machine tool axis system for turning operation.

one or more rotational axes. Three rotational axes are defined in NC: the a, b, and c axes. These axes specify angles about the x, y, and z axes, respectively. To distinguish positive from negative angular motions, the "right-hand rule" can be used. Using the right hand with the thumb pointing in the positive linear axis direction (x, y, or z), the fingers of the hand are curled to point in the positive rotational direction.

For turning operations, two axes are normally all that are required to command the movement of the tool relative to the rotating work piece. The z axis is the axis of rotation of the work part, and x axis defines the radial location of the cutting tool. This arrangement is illustrated in Figure.

The purpose of the coordinate system is to provide a means of locating the tool in relation to the work piece. Depending on the NC machine, the part programmer may have several different options available for specifying this location.

Fixed zero and floating zero

The programmer must determine the position of the tool relative to the origin (zero point) of the coordinate system. NC machines have either of two methods for specifying the zero point. The first possibility is for the machine to have a fixed zero. In this case, the origin is always located at the same position on the machine. Usually, that position is the southwest corner (lower left-hand corner) of the table and all tool locations will be defined by positive x and y coordinates.

The second and more common feature on modern NC machines allows the machine operator to set the zero point at any position on the machine table. This feature is called floating zero. The part programmer is the one who decides where the zero point should be located. The decision is based on part programming convenience. For example, the work part may be symmetrical and the zero point should be established at the center of symmetry.

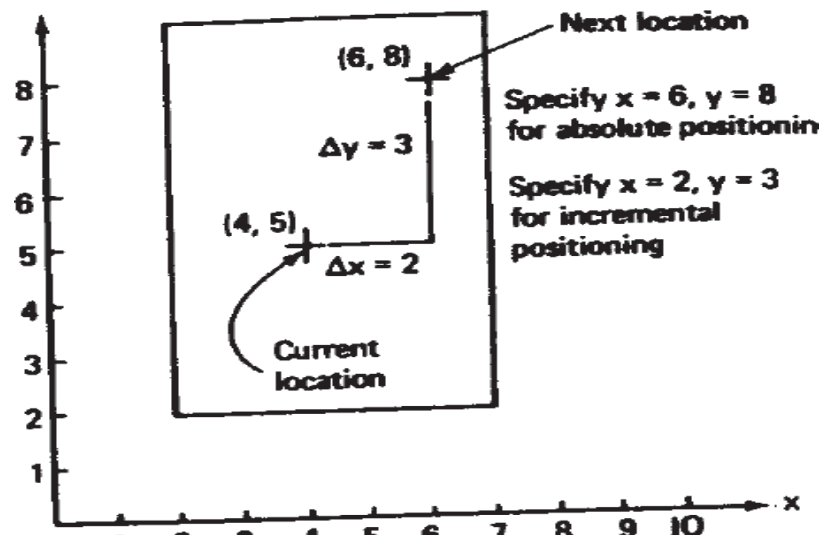


FIGURE Absolute versus incremental positioning

The location of the zero point is communicated to the machine operator. At the beginning of the job, the operator moves the tool under manual control to some "target point" on the table. The target point is some convenient place on the work piece or table for the operator to position the tool. For example, it might be a predrilled hole in the work piece. The target point has been referenced to the zero point by the part programmer. In fact, the programmer may have selected the target point as the zero point for tool positioning. When the tool has been positioned at the target point, the machine operator presses a "zero" button on the machine tool console, which tells the machine where the origin is located for subsequent tool movements.

Absolute positioning and incremental positioning

Another option sometimes available to the part programmer is to use either an absolute system of tool positioning or an incremental system. Absolute positioning means that the tool locations are always defined in relation to the zero point. If a hole is to be drilled at a

spot that is 8 in. above the x axis and 6in. to the right of the y axis, the coordinate location of the hole would be specified as $x = +6.000$ and $y = +8.000$. By contrast, incremental positioning means that the next tool location must be defined with reference to the previous tool location. If in our drilling example, suppose that the previous hole had been drilled at an absolute position of $x = +4.000$ and $y = +5.000$. Accordingly, the incremental position instructions would be specified as $x = +2.000$ and $y = +3.000$ in order to move the drill to the desired spot. Figure illustrates the difference between absolute and incremental positioning.

NC MOTION CONTROL SYSTEMS

In order to accomplish the machining process, the cutting tool and workpiece must be moved relative to each other. In NC, there are three basic types of motion control systems: -

1. Point-to-point
2. Straight cut
3. Contouring

Point-to-point systems represent the lowest level of motion control between the tool and workpiece. Contouring represents the highest level of control.

Point-to-point NC

Point-to-point (PTP) is also sometimes called a positioning system. In PTP, the objective of the machine tool control system is to move the cutting tool to a predefined location. The speed or path by which this movement is accomplished is not important in point-to-point NC. Once the tool reaches the desired location, the machining operation is performed at that position.

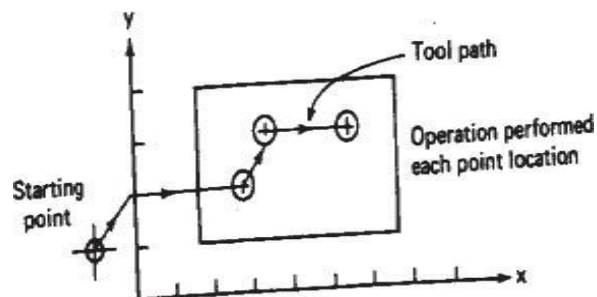
NC drill presses are a good example of PTP systems. The spindle must first be positioned at a particular location on the work piece. This is done under PTP control. Then the drilling of the hole is performed at the location, and so forth. Since no cutting is performed between holes, there is no need for controlling the relative motion of the tool and work piece between hole locations. Figure illustrates the point-to-point type of control.

Positioning systems are the simplest machine tool control systems and are therefore the least expensive of the three types. However,

for certain processes, such as drilling operations and spot welding, PIP is perfectly suited to the task and any higher level of control would be unnecessary.

Straight-cut NC

Straight-cut control systems are capable of moving the cutting tool parallel to one of the major axes at a controlled rate suitable for machining. It is therefore appropriate for performing milling operations to fabricate workpieces of rectangular configurations. With this type of NC system it is not possible to combine movements in more than a Single axis direction. Therefore, angular cuts on the workpiece would not be



possible. An example of a straight-cut operation is shown in Figure.

FIGURE Point-to-point (positioning) NC system.

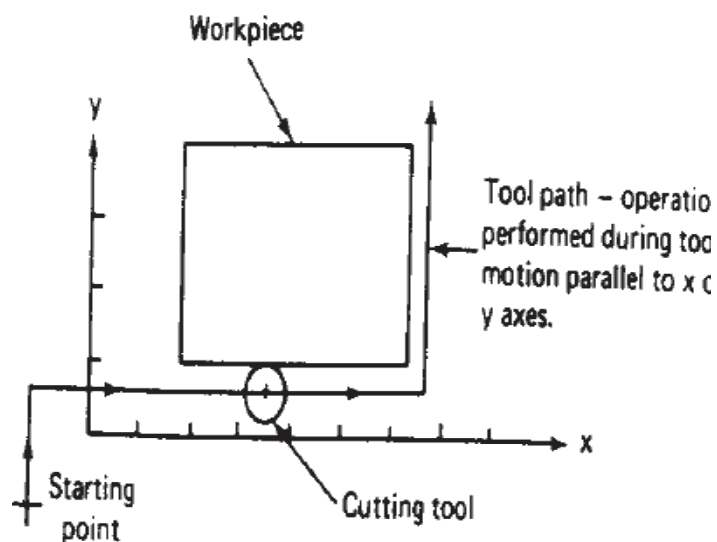


FIGURE Straight-cut system.

An NC machine capable of straight cut movements is.-also

capable of PTP movements.

Contouring NC

Contouring is the most complex, the most flexible, and the most expensive type of machine tool control. It is capable of performing both PTP and straight-cut operations. In addition, the distinguishing feature of contouring NC systems is their capacity for simultaneous control of more than one axis movement of the machine tool. The path of the cutter is continuously controlled to generate the desired geometry of the workpiece. For this reason, contouring systems are also called continuous-path NC systems. Straight or plane surfaces at any orientation, circular paths, conical shapes, or most any other mathematically definable form are possible under contouring control. Figure illustrates the versatility of continuous path NC. Milling and turning operations are common examples of the use of contouring control.

In order to machine a curved path in a numerical control contouring system, the direction of the feed rate must continuously be changed so as to follow the path. This is accomplished by breaking the curved path into very short straight-line segments that approximate the curve. Then the tool is commanded to machine each segment in succession. What results is a machined outline that closely approaches

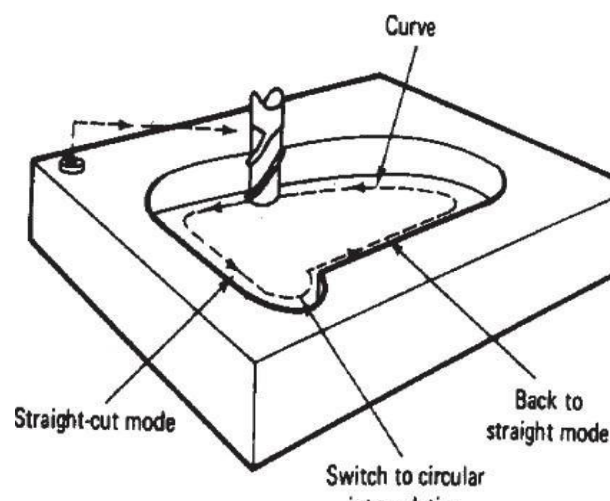


FIGURE Contouring (continuous path) NC system for two-dimensional operations.

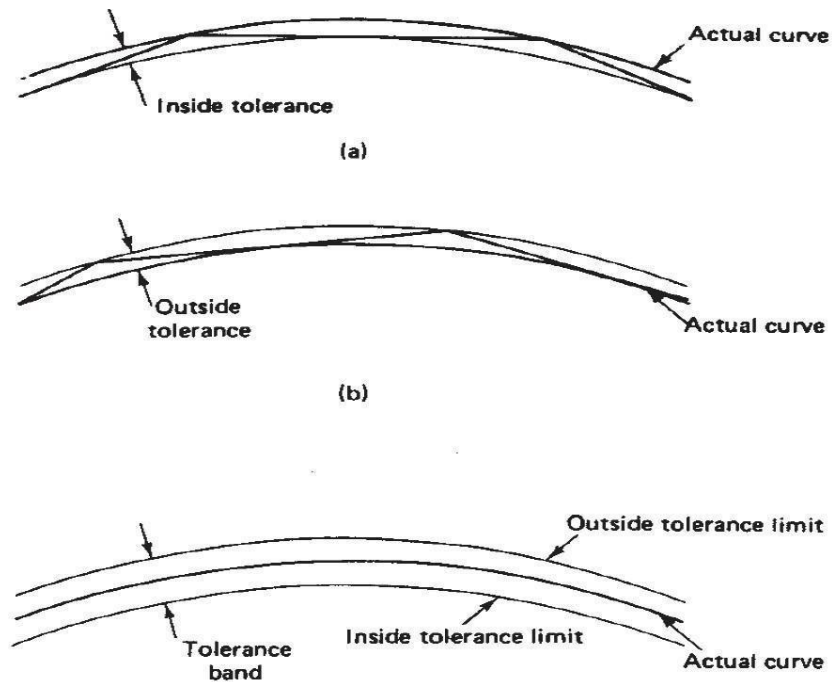


FIGURE Approximation of a curved path in NC by a series of straight-line segments.

The accuracy of the approximation is controlled by the "tolerance" between the actual curve and the maximum deviation of the straight-line segments. In (a), the tolerance is defined on the inside of the curve. It is also possible to define the tolerance on the outside of the curve, as in (b). Finally, the tolerance can be specified on both inside and outside, as shown in (c).

the desired shape. The maximum error between the two can be controlled by the length of the individual line segments, as illustrated in Figure.

APPLICATIONS OF NUMERICAL CONTROL

Numerical control systems are widely used in industry today, especially in the metalworking industry. By far the most common application of NC is for metal cutting machine tools. Within this category, numerically controlled equipment has been built to perform virtually the entire range of material removal processes, including:

Milling

Drilling and related processes

Boring

Turning Grinding Sawing

Within the machining category, NC machine tools are appropriate for certain jobs and inappropriate for others. Following are the general characteristics of production jobs in metal machining for which numerical control would be most appropriate:

1. Parts are processed frequently and in small lot sizes.
2. The part geometry is complex.
3. Many operations must be performed on the part in its processing.
4. Much metal needs to be removed.
5. Engineering design changes are likely.
6. Close tolerances must be held on the workpart.
7. It is an expensive part where mistakes in processing would be costly.
8. The parts require 100% inspection

It has been estimated that most manufactured parts are produced in lot sizes of 50 or fewer. Small-lot and batch production jobs represent the ideal situations for the application of NC. This is made possible by the capability to program the NC machine and to save that program for subsequent use in future orders. If the NC programs are long and complicated (complex part geometry, many operations, much metal removed), this makes NC all the more appropriate when compared to manual methods of production. If engineering design changes or shifts in the production schedule are likely, the use of tape control provides the flexibility needed to adapt to these changes. Finally, if quality and inspection are important issues (close tolerances, high part cost, 100% inspection required), NC would be most suitable, owing to its high accuracy and repeatability.

In order to justify that a job be processed by numerical control methods, it is not necessary that the job possess every one of these attributes. However, the more of these characteristics that are present, the more likely it is that the part is a good candidate for NC.

In addition to metal machining, numerical control has been applied to a variety of other operations. The following, although not a complete list, will give the reader an idea of the wide range of potential applications of NC:

Press working machine tools Welding machines Inspection machines Automatic drafting Assembly machines

Tube bending Flame cutting Plasma arc cutting

Laser beam processes Automated knitting machines Cloth cutting

Automatic riveting Wire-wrap machines

Advantages of NC

Following are the advantages of numerical control when it is utilized in the type of production jobs described.

1. *Reduced nonproductive time.* Numerical control has little or no effect on the basic metal, cutting (or other manufacturing) process. However; NC can increase the proportion of time the machine is engaged in the actual process. It accomplishes this by means of fewer setups, less time in setting up, reduced work piece handling time, automatic tool changes on some machines, and so on.

In a University of Michigan survey reported by Smith and Evans, a comparison was made between the machining cycle times for conventional machine tools versus the cycle times for NC machines. NC cycle times, as a percentage of their conventional counterparts, ranged from 35% for five-axis machining centers to 65% for presswork punching. The advantage for numerical control tends to increase with the more complex processes.

2. *Reduced fixturing.* NC requires fixtures which are simpler and

less costly to fabricate because the positioning is done by the NC tape rather than the jig or fixture

3. *Reduced manufacturing lead time.* Because jobs can be set up more quickly with NC and fewer setups are generally required with NC, the lead time to deliver a job to the customer is reduced.

4. *Greater manufacturing flexibility.* With numerical control it is less difficult to adapt to engineering design changes alterations of the production schedule, changeovers in jobs for rush orders, and so on.

5. *Improved quality control.* NC is ideal for complicated workparts where the chances of human mistakes are high. Numerical control produces parts with greater accuracy, reduced scrap, and lower inspection requirements. 6. *Reduced inventory.* Owing to fewer setups and shorter lead times with numerical control, the amount of inventory carried by the company is reduced.

7. *Reduced floor space requirements.* Since one NC machining center can often accomplish the production of several conventional machines, the amount of floor space required in an NC shop is usually less than in a conventional shop.

Disadvantages of NC

Along with the advantages of NC, there are several features about NC which must be considered disadvantages:

1. *Higher investment cost.* Numerical control machine tools represent a more sophisticated and complex technology. This technology costs more to buy than its non-NC counterpart. The higher cost requires manufacturing managements to use these machines more aggressively than ordinary equipment. High machine utilization is essential on order to get reasonable returns on investment. Machine shops must operate their NC machines two or three shifts per day to achieve this high machine utilization.

2. *Higher maintenance cost.* Because NC is a more complex technology and because NC machines are used harder, the maintenance problem becomes more acute. Although the reliability of NC systems has been improved over the years, maintenance costs for NC machines will

generally be higher than for conventional machine tools.

3. *Finding and/or training NC personnel.* Certain aspects of numerical control shop operations require a higher skill level than conventional operations. Part programmers and NC maintenance personnel are two skill areas where available personnel are in short supply. The problems of finding, hiring, and training these people must be considered a disadvantage to the NC shop.

The ladder logic diagram is an excellent way to represent the combinatorial 10 control problems in which the output variables are based directly on the values of inputs. As indicated by Example 9.6, it can also be used to display sequential control (timer) problems, although the diagram is somewhat more difficult to interpret and analyze for this purpose. The ladder diagram is the principal technique for setting up the control programs in PLCs.

PROGRAMMABLE LOGIC CONTROLLERS

A programmable logic controller (PLC) can be defined as a microcomputer-based controller that uses stored instructions in programmable memory to implement logic sequencing, timing, counting, and arithmetic functions through digital or analog input/output (I/O) modules, for controlling machines and processes. PLC applications are found in both the process industries and discrete manufacturing. Examples of applications in process industries include chemical processing, paper mill operations, and food production. PLCs are primarily associated with discrete manufacturing industries to control individual machines, machine cells, transfer lines, material handling equipment, and automated storage systems. Before the PLC was introduced around 1970 hard-wired controllers composed of relays, coils, counters, timers, and similar components were used to implement this type of industrial control. Today, many older pieces of equipment are being retrofitted with PLCs to replace the original hard wired controllers, often making the equipment more productive and reliable than when it was new.

In this section, we describe the components, programming, and operation of the PLC. Although its principal applications are in logic control and sequencing (discrete control), many PLCs also perform additional functions, surveyed later in the section.

Components of the PLC

A schematic diagram of a PLC is presented in Figure. The basic components of the PLC are the following: (1) processor, (2) memory unit, (3) power supply, (4) I/O module, and (5) programming device. These components are housed in a suitable cabinet designed for the industrial environment.

The processor is the central processing unit (CPU) of the programmable controller. It executes the various logic and sequencing functions by operating on the PLC inputs to determine the appropriate output signals. The CPU consists of one or more microprocessors similar to those used in PCs and other data processing equipment. The difference is that they have a real-time operating system and are programmed to facilitate I/O transactions and execute ladder logic functions. In addition, PLCs are hardened so that the CPU and other electronic components will operate in the electrically noisy environment of the factory.

Connected to the CPU is the PLC memory unit, which contains the programs of logic, sequencing, and I/O operations. It also holds data files associated with these programs, including I/O status bits, counter and timer constants, and other variable and parameter values. This memory unit is referred to as the user or application memory because its contents are entered by the user. In addition, the processor also contains the operating system memory, which directs the execution of the control program and coordinates I/O operations. The operating system is entered by the PLC manufacturer and cannot be accessed or altered by the user.

A power supply of 115 V ac is typically used to drive the PLC (some units operate on 230 V ac). The power supply converts the 115 V ac into direct current (dc) voltages of ± 5 V. These low voltages are used to operate equipment that may have much higher voltage and power ratings than the PLC itself. The power supply often includes a battery backup that switches on automatically in the event of an external power source failure.

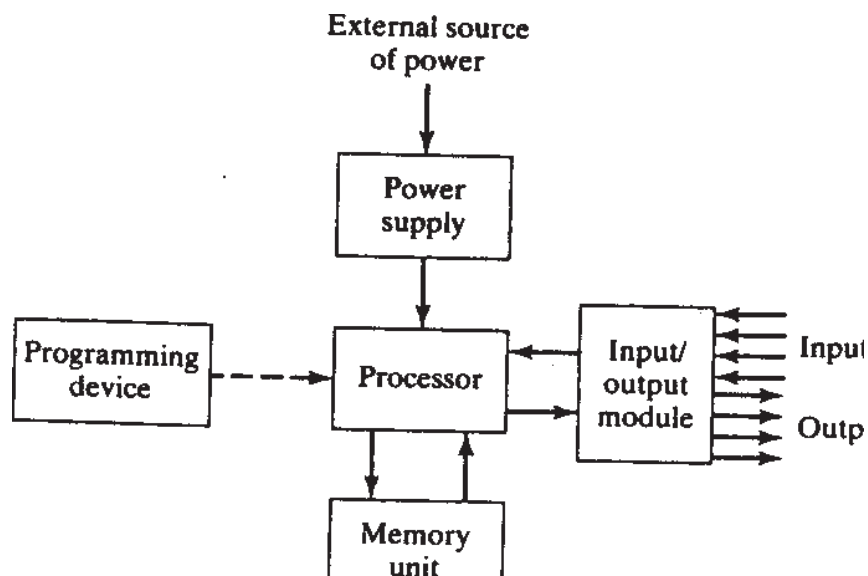


Figure Components of a PLC.

The input/output module provides the connections to the industrial equipment or process that is to be controlled. Inputs to the controller are signals from limit switches, push-buttons, sensors, and other on/off devices. Outputs from the controller are on/off signals to operate motors, valves, and other devices required to actuate the process. In addition, many PLCs are capable of accepting continuous signals from analog sensors and generating signals suitable for analog actuators. The size of a PLC is usually rated in terms of the number of its I/O terminals, as indicated in Table.

The PLC is programmed by means of a programming device. The programming device is usually detachable from the PLC cabinet so that it can be shared among different controllers. Different PLC manufacturers provide different devices, ranging from simple teach-pendant type devices, similar to those used in robotics, to special PLC

programming keyboards and displays. Personal computers can also be used to program PLCs. A PC used for this purpose sometimes remains connected to the PLC to serve a process monitoring or supervisory function and for conventional data processing applications related to the process.

PLC Operating Cycle

As far as the PLC user is concerned, the steps in the control program are executed simultaneously and continuously. In truth, a certain amount of time is required for the PLC processor to execute the user program during one cycle of operation. The typical operating cycle of the PLC, called a scan, consists of three parts: (1) input scan, (2) program scan, and (3) output scan. During the input scan, the inputs to the PLC are read by the processor and the status of these inputs is stored in memory. Next, the control program is executed during the program scan. The input values stored in memory are used in the control logic calculations to determine the values of the outputs. Finally, during the output scan, the outputs are updated to agree with the calculated values. The time to perform the scan is called the scan time, and this time depends on the number of inputs that must be read, the complexity of control functions to be performed, and the number of outputs that must be changed. Scan time also depends on the clock speed of the processor. Scan times typically vary between 1 and 25 sec.

One of the potential problems that can occur during the scan cycle is that the value of an input can change immediately after it has been sampled. Since the program uses the input value stored in memory, any output values that are dependent on that input are determined incorrectly. There is obviously a potential risk involved in this mode of operation. However, the risk is minimized because the time between updates is so short that it is unlikely that the output value being incorrect for such a short time will have a serious effect on process operation. The risk becomes most significant in processes in which the response times are very fast and where hazards can occur during the scan time. Some PLCs have special features for making "immediate" updates of output

signals when input variables are known to cycle back and forth at frequencies faster than the scan time.

TABLE Typical Classification of PLCs by Number of Input/Output Terminals

PLC Size	I/O Count
Large PLC	≤ 1024
Medium PLC	< 1024
Small PLC	< 256
Micro PLC	≤ 32
Nano PLC	≤ 16

Additional Capabilities of the PLC

The PLC has evolved to include several capabilities in addition to logic control and sequencing. Some of these additional capabilities available on many commercial PLCs include

- *Analog control.* Proportional-integral-derivative (PID) control is available on some programmable controllers. These control algorithms have traditionally been implemented using analog controllers. Today the analog control schemes are approximated using the digital computer, with either a PLC or a computer process controller.
- *Arithmetic functions.* These functions are addition, subtraction, multiplication, and division. Use of these functions permits more complex control algorithms to be developed than what is possible with conventional logic and sequencing elements.
- *Matrix functions.* Some PLCs have the capability to perform matrix operations on stored values in memory. The capability can be used to compare the actual values of a set of inputs and outputs with the values stored in the PLC memory to determine if some error has occurred.

- *Data processing and reporting.* These functions are typically associated with business applications of PCs. PLC manufacturers have found it necessary to include these PC capabilities in their controller products, as the distinction between PCS and PLCs blurs.

Programming the PLC

Programming is the means by which the user enters the control instructions to the PLC through the programming device. The most basic control instructions consist of switching, logic, sequencing, counting, and timing. Virtually all PLC programming methods provide instruction sets that include these functions. Many control applications require additional instructions to accomplish analog control of continuous processes, complex control logic, data processing and reporting, and other advanced functions not readily performed by the basic instruction set. Owing to these differences in requirements, various PLC programming languages have been developed. A standard for PLC programming was published by the International Electro technical Commission in 1992, entitled International Standard for Programmable Controllers (IEC 1131-3). This standard specifies three graphical languages and two text-based languages for programming PLCs, respectively: (1) ladder logic diagrams, (2) function block diagrams, (3) sequential functions charts, (4) instruction list, and (5) structured text. Table 9.9 lists the five languages along with the most suitable application of each. IEC 1131-3 also states that the five languages must be able to interact with each other to allow for all possible levels of control sophistication in any given application.

TABLE Features of the Five PLC Languages Specified in the IEC 1131-3 Standard Applications Best Suited for

Language	Abbreviation	Type	Applications Best Suited for
Ladder logic diagram	(LD)	Graphical	Discrete control

Function block diagram	(FBD)	Graphical	Continuous control
Sequential function chart	(SFC)	Graphical	Sequencing
Instruction list	(IL)	Textual	Discrete control
Structured text	(ST)	Textual	Complex logic, computations, etc.

Ladder Logic Diagram. The most widely used PLC programming language today involves ladder diagrams (LDs), examples of which are shown in several previous figures. Ladder diagrams are very convenient for shop personnel who are familiar with ladder and circuit diagrams but may not be familiar with computers and computer programming. To use ladder logic diagrams, they do not need to learn an entirely new programming language.

Direct entry of the ladder logic diagram into the PLC memory requires the use of a keyboard and monitor with graphics capability to display symbols representing the components and their interrelationships in the ladder logic diagram. The PLC keyboard is often designed with keys for each of the individual symbols. Programming is accomplished by inserting the appropriate components into the rungs of the ladder diagram. The components are of two basic types: contacts and coils, as described in Section 9.2. Contacts represent input switches, relay contacts, and similar elements. Coils represent loads such as motors, solenoids, relays, timers, and counters. In effect, the programmer inputs the ladder logic circuit diagram rung by rung into the PLC memory with the monitor displaying the results for verification.

Function Block Diagrams. A function block diagram (FBD) provides a means of inputting high-level instructions. Instructions are composed of operational blocks. Each block has one or more inputs and one or more outputs. Within a block, certain operations take place on the inputs to transform the signals into the desired outputs. The function blocks include operations such as timers and counters, control computations using equations (e.g., proportional-integral-derivative control), data manipulation, and data transfer to other computer-based systems. We leave further description of these function blocks to other references, such as Hughes and the operating manuals for commercially available PLC products.

Sequential Function Charts. The sequential function chart (SFC, also called the Grafcet method) graphically displays the sequential

functions of an automated system as a series of steps and transitions from one state of the system to the next. The sequential function chart is described in Boucher . It has become a standard method for documenting logic control and sequencing in much of Europe. However, its use in the United States is more limited, and we refer the reader to the cited reference for more details on the method.

Instruction List. Instruction list (IL) programming also provides a way of entering the ladder logic diagram into PLC memory. In this method, the programmer uses a low-level computer language to construct the ladder logic diagram by entering statements

TABLE Typical Low-Level Language Instruction Set for a PLC

STR	Store a new input and start a new rung of the ladder.
AND	Logical AND referenced with the previously entered element. This is interpreted as a series circuit relative to the previously entered element.
OR	Logical OR referenced with the previously entered element. This is interpreted as a parallel circuit relative to the previously entered element.
NOT	Logical NOT or inverse of entered element.
OUT	Output element for the rung of the ladder diagram.
TMR	Timer element. Requires one input signal to initiate timing sequence. Output is delayed relative to input by a duration specified by the programmer in seconds. Resetting the timer is accomplished by interrupting (stopping) the input signal.
CTR	Counter element. Requires two inputs: One is the incoming pulse train that is counted by the CTR element, the other is the reset signal indicating a restart of the counting procedure.

that specify the various components and their relationships for each rung of the ladder diagram. Let us explain this approach by introducing a hypothetical PLC instruction set. Our PLC "language" is a composite of various manufacturers' languages. It contains fewer features than most commercially available PLCs. We assume that the programming device consists of a suitable keyboard for entering the individual components on each rung of the ladder logic diagram. A monitor capable of displaying each ladder rung (and perhaps several

rungs that precede it) is useful to verify the program. The instruction set for our PLC is presented in Table with a concise explanation of each instruction.

NC PART PROGRAMMING

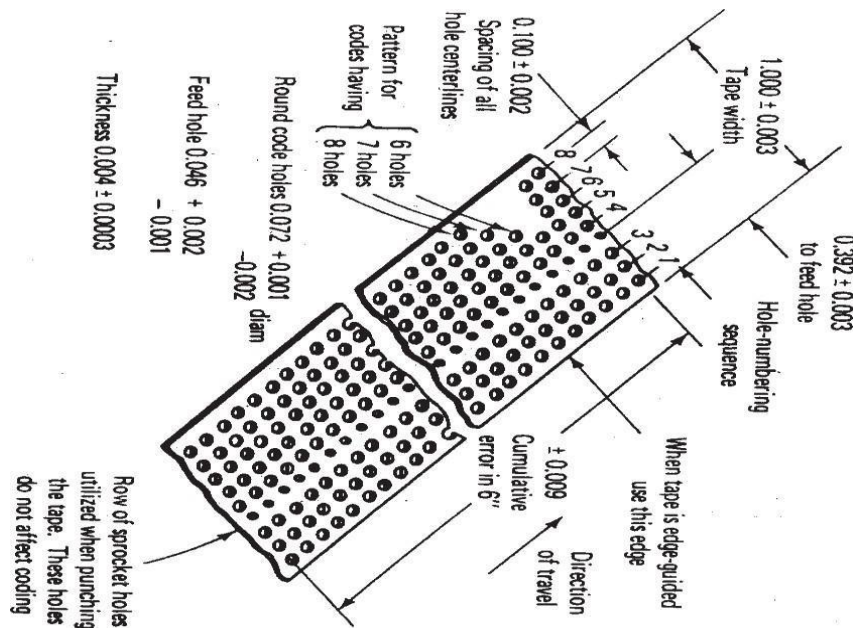
INTRODUCTION

Numerical control part programming is the procedure by which the sequence of processing steps to be performed on the NC machine is planned and documented. It involves the preparation of a punched tape (or other input medium) used to transmit the processing instructions to the machine tool. There are two methods of part programming: manual part programming and computer-assisted part programming. In this chapter we describe both of these methods, with emphasis on the latter.

It is appropriate to begin the discussion of NC part programming by examining the way in which the punched tape is coded. Coding of the punched tape is concerned with the basic symbols used to communicate a complex set of instructions to the NC machine tool. In numerical control, the punched tape must be generated whether the part programming is done manually or with the assistance of some computer package. With either method of part programming, the tape is the net result of the programming effort. In coming sections our attention will be focused on the punched tape and the structure of the basic language used by the NC system.

PUNCHED TAPE IN NC

The part program is converted into a sequence of machine tool actions by means of the input medium, which contains the program, and the controller unit, which interprets the input medium. The controller unit and the input medium must be compatible. That is, the input medium uses coded symbols which represent the part program, and the controller unit must be capable of reading those symbols. [The most common input medium is punched tape. The tape has been standardized, and that type punchers are manufactured to prepare the NC tapes, and tape readers (part of the controller unit) can be manufactured to read the tapes. The punched tape used for NC is 1 in. wide. It is standardized as shown in Figure by the electronics Industries Association (EIA), which has been responsible for many of the important standards in the NC industry.



There are two basic methods of preparing the punched tape. The first method associated with manual part programming and involves the use of a typewriter like device. Figure illustrates a modern version of this kind of equipment. The operator types directly from the part programmer's handwritten list of coded instructions. This produces a typed copy of the program as well as the punched type. The second method is used with computer-assisted part programming. By this approach, the tape is prepared directly by the computer using a device called a tape punch.

By either method of preparation, the punched tape is ready for use. During production on a conventional NC machine, the tape is fed through the tape reader once for each workpiece. It is advanced through the tape reader one instruction at a time. While the machine tool is performing one instruction, the next instruction is being read into the controller unit's data buffer. This makes the operation of the NC system more efficient. After the last instruction has been read into the controller, the tape is rewound back to the start of the program to be ready for the next workpart.

TAPE CODING AND FORMAT

NC tape coding

As shown in Figure, there are eight regular columns of holes running in the lengthwise direction of the tape. There is also a ninth column of holes between the third and fourth regular columns. However, these are smaller and are used as sprocket holes for feeding the tape.

Figure shows a hole present in nearly every position of the tape. However, the coding of the tape is provided by either the presence or absence of a hole in the various positions. Because there are two possible conditions for each position—either the presence or absence of a hole—this coding system is called the binary code. It uses the base 2 number system, which can represent any number in the more familiar base 10 or decimal system. The NC tape coding system is used to code not only numbers, but also alphabetical letters and other symbols. Eight columns provide more than enough binary digits to define any of the required symbols.

How instructions are formed

A binary digit is called a bit. It has a value of 0 or 1 depending on the absence or presence of a hole in a certain row and column position on the tape. (Columns of hole positions run lengthwise along the tape. Row positions run across the tape.) Out of a row of bits, a character is made. A character is a combination of bits, which represents a letter, number, or other symbol. A word is a collection of characters used to form part of an instruction. Typical NC words are x position, y position, cutting speed, and so on. Out of a collection of words, a block is formed. A block of words is a complete NC instruction. Using an NC drilling operation as an example, a block might contain information on the x and y coordinates of the hole location, the speed and feed at which the cut should be run, and perhaps even a specification of the cutting tool.

To separate blocks, an end-of-block (EOB) symbol is used (in the EIA standard, this is a hole in column 8). The tape reader feeds the data from the tape into the buffer in blocks. That is, it reads in a complete instruction at a time.

NC words

Following is a list of the different types of words in the formation of a block. Not every NC machine uses all the words. Also, the manner in which the words are expressed will differ between machines. By convention, the words in a block are given in the following order:

SEQUENCE NUMBER (n-words): This is used to identify the block.

PREPARATORY WORD (g-words): This word is used to prepare the controller for instructions that are to follow. For example, the word G02 is used to prepare the C controller unit for circular interpolation along an arc in the clockwise direction. The preparatory word is needed so that the controller can correctly interpret the data that follow it in the block.

COORDINATES (x-, y-, and z-words): These give the coordinate positions of the tool. In a two-axis system, only two of the words would be used. In a four- or five- axis machine, additional a-words and V or b-words would specify the angular positions.

Although different NC systems use different formats for expressing a coordinate, we will adopt the convention of expressing it in the familiar decimal form: For example, $x + 7.235$ or $y - 0.5$. Some formats do not use the decimal point in writing the coordinate. The + sign to define a positive coordinate location is optional. The negative sign is, of course, mandatory.

FEED RATE (f-word): This specifies the feed in a machining operation. Units are inches per minute (ipm) by convention.

CUTTING SPEED (s-word): This specifies the cutting speed of the process, the rate at which the spindle rotates.

TOOL SELECTION (t-word): This word would be needed only for machines with a tool turret or automatic tool changer. The t-word specifies which tool is to be used in the operation. For example, T05 might be the designation of a 1/2-in. drill bit in turret position 5 on an NC turret drill.

MISCELLANEOUS FUNCTION (m-word): The m-word is used to

specify certain miscellaneous or auxiliary functions which may be available on the machine tool. Of course, the machine must possess the function that is being called. An example would be m03 to start the spindle rotation. The miscellaneous function is the last word in the block. To identify the end of the instruction, an end-of-block (EOB) symbol is punched on the tape.

MANUAL PART PROGRAMMING

To prepare a part program using the manual method, the programmer writes the machining instructions on a special form called a part programming manuscript. The instructions must be prepared in a very precise manner because the typist prepares the NC tape directly from the manuscript. Manuscripts come in various forms, depending on the machine tool and tape format to be used. For example, the manuscript form for a two-axis point-to-point drilling machine would be different than one for a three-axis contouring machine. The manuscript is a listing of the relative tool and workpiece locations. It also includes other data, such as preparatory commands, miscellaneous instructions, and speed/feed specifications, all of which are needed to operate the machine under tape control.

Manual programming jobs can be divided into two categories: point-to-point jobs and contouring jobs. Except for complex work parts with many holes to be drilled, manual programming is ideally suited for point-to-point applications. On the other hand, except for the simplest milling and turning jobs, manual programming can become quite time consuming for applications requiring continuous-path control of the tool. Accordingly, we shall be concerned only with manual part programming for point-to-point operations. Contouring is much more appropriate for computer-assisted part programming.

The basic method of manual part programming for a point-to-point application is best demonstrated by means of an example.

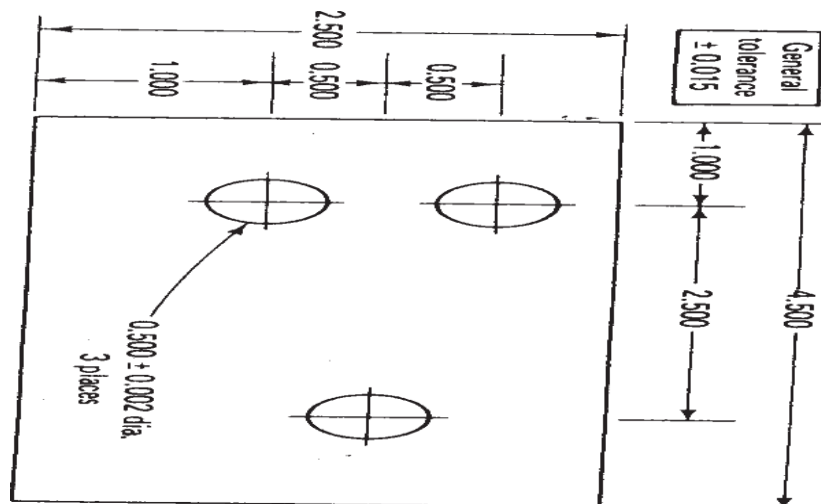
EXAMPLE

Suppose that the part to be programmed is a drilling job. The engineering drawing for the part is presented in Figure. Three holes are to be drilled at a diameter of 31 in. The close hole size tolerance requires reaming to 0.500 in. diameter.

Recommended speeds and feeds are as follows:

	Speed (rpm)	Speed (in./min)
0.484-in.-diameter drill	592	3.55
0.500-in.-diameter drill	382	3.82

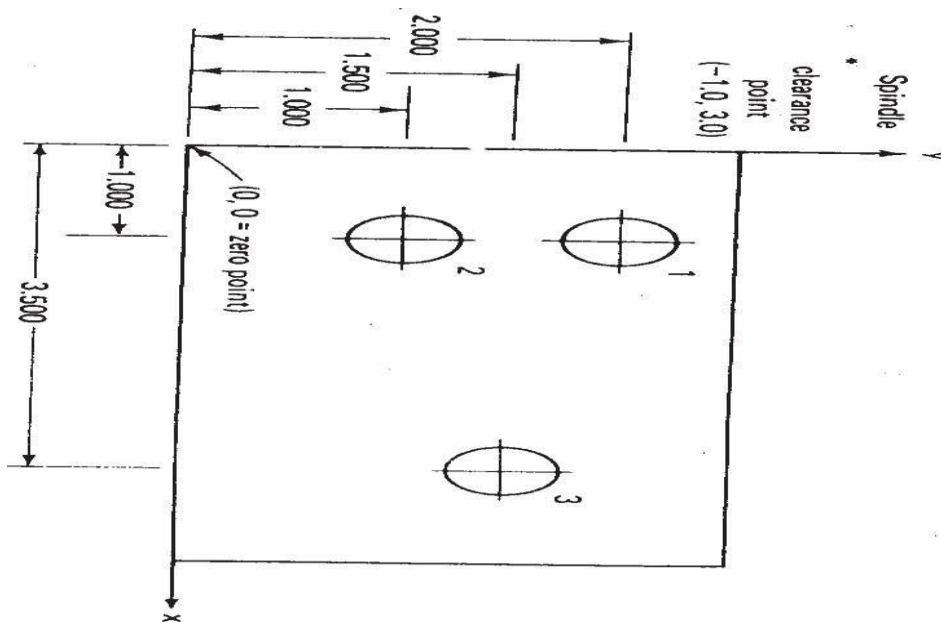
The NC drill press operates as follows. Drill bits are manually changed by the machine operator, but speeds and feeds must be programmed on the tape. The machine has the floating-zero feature and absolute positioning.



Part drawing for Example.

The first step in preparing the part program is to define the axis coordinates in relation to the work part. We assume that the outline of the part has already been machined before the drilling operation. Therefore, the operator can use one of the corners of the part as the target point. Let us define the lower left-hand corner as the target point and the origin of

our axis system. The coordinates are shown in Figure for the example part. The x and y locations of each hole can be seen in the figure. The



completed manuscript would appear as in Figure. The first line shows the x and y coordinates at the zero point. The machine operator would insert the tape and read this first block into the system. (A block of instruction corresponds generally to one line on the manuscript form.) The tool would then be positioned over the target point on the machine table. The operator would then press the zero buttons to set the machine.

Coordinate system defined for PART IN example

The next line on the manuscript is RWS, which stands for rewind-stop. This signal is coded into the tape as holes in columns 1, 2, and 4. The symbol stops the tape after it has been rewound. The last line on the tape contains the m30 word, causing the tape to be rewound at the end of the machining cycle. Other m-words used in the program are m06, which stops the machine for an operator tool change, and m13, which turns on the spindle and coolant. Note in the last line that the tool has been repositioned away from the work area to allow for changing the workpiece.

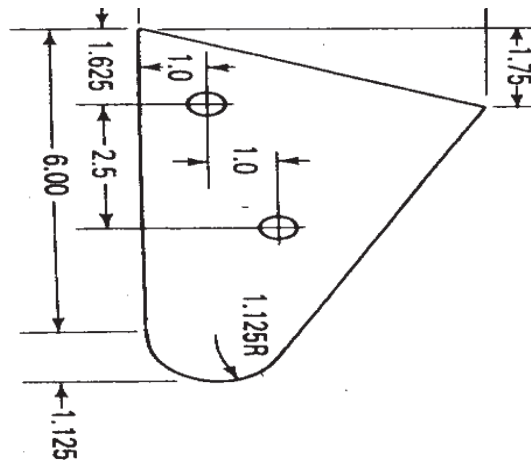
preparing the tape from the engineering drawing is followed as. The machining instructions are written in English-like statements of the NC programming language, which are then processed by the computer to prepare the tape. The computer automatically punches the tape in the proper tape format for the particular C machine.

The part programmer's responsibility in computer-assisted part programming consists of two basic steps:

1. Defining the workpart geometry
2. Specifying the operation sequence and tool path

No matter how complicated the workpart may appear, it is composed of six geometric elements. Using a relatively simple workpart to illustrate, consider the component shown in Figure. Although somewhat irregular in overall appearance, the outline of the part consists of intersecting straight lines and a partial circle. The holes in the part can be expressed in terms of the center location and radius of the hole. Nearly any component that can be conceived by a designer can be described by points, straight lines, planes, circles, cylinders, and other mathematically defined surfaces. It is the part programmer's task to enumerate the elements out of which the part is composed. Each geometric element must be identified and the dimensions and location of the element explicitly defined.

After defining the workpart geometry, the programmer must next construct the path that the cutter will follow to machine the part. This tool path specification involves a detailed step-by-step sequence of cutter moves. The moves are made among the geometry elements, which have previously been defined. The part programmer can use the various motion commands to direct the tool to machine along the workpart surfaces, to go to point locations, to drill holes at these locations, and so on. In addition to part geometry and tool motion statements, the programmer must also provide other instructions to operate the machine tool properly.



Sample workpart, like other parts, can be defined in terms of basic geometric elements, such as points, lines, and circles.

The computer's job

The computer's job in computer-assisted part programming consists of the following steps:

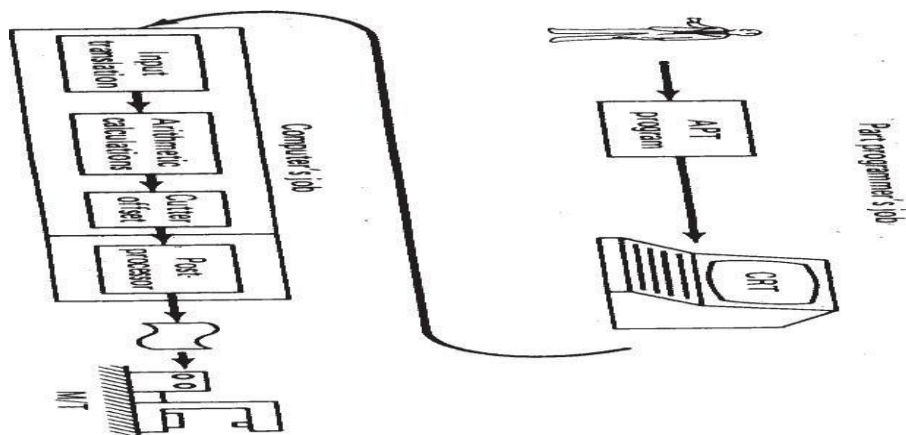
1. Input translation
2. Arithmetic calculations
3. Cutter offset computation
4. Postprocessor

The sequence of these steps and their relationships to the part programmer and the machine tool are illustrated in Figure.

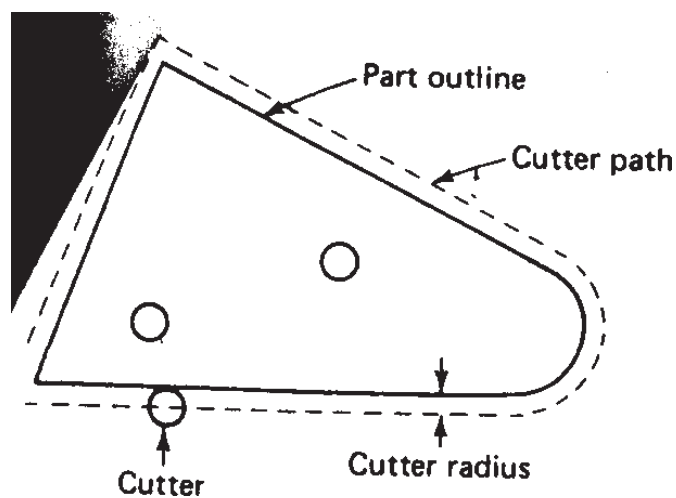
The part programmer enters the program written in the APT or other language. The input translation component converts the coded instructions contained in the program into computer-usable form, preparatory to further processing.

The arithmetic calculations unit of the system consists of a comprehensive set of subroutines for solving the mathematics required to generate the part surface. These subroutines are called by the various part programming language statements. The arithmetic unit is really the fundamental element in the part programming package. This unit frees the programmer from

the time-consuming geometry and trigonometry calculations, to concentrate on the workpart processing.



Steps in computer-assisted part programming.



Cutter offset problem in part programming for contouring.

The second task of the part programmer is that of constructing the tool path. However, the actual tool path is different from the part outline because the tool path is defined as the path taken by the center of the cutter. It is at the periphery of the cutter that machining takes place. The purpose of the cutter offset computations is to offset the tool path from the desired part surface by the radius of the cutter. This means that

the part programmer can define the exact part outline in the geometry statements. Thanks to the cutter offset calculation provided by the programming system, the programmer need not be concerned with this task. The cutter offset problem is illustrated in Figure.

As noted previously, NC machine tool systems are different. They have different features and capabilities. They use different NC tape formats. Nearly all of part programming languages, including APT, are designed to be general purpose languages, not limited to one or two machine tool types. Therefore, the main task of the computer in computer-assisted part programming is to take the general instructions and make them specific to a particular machine tool system. The unit that performs this task is called a postprocessor.

The postprocessor is a separate computer program that has been written to prepare the punched tape for a specific machine tool. The input to the postprocessor is output from the other three components: a series of cutter locations and other instructions. The output of the postprocessor is the NC tape written in the correct format for the machine on which it is to be used.

Part programming languages

NC part programming language consists of a software package (computer program) plus the special rules, conventions, and vocabulary words for using that software. Its purpose is to make it convenient for a part programmer to communicate the necessary part geometry and tool motion information to the computer so the desired part program can be prepared. The vocabulary words are typically English-like, to make the NC language easy to use.

There have probably been over 100 NC part programming languages developed since the initial MIT research on NC programming in the mid-1950s. Of the languages were developed to meet particular needs and have not survived the test of time. Today, there are several dozen NC languages still in use. Refinements and enhancements to existing languages are continually being made. The following list provides a

description of some of the important NC languages in current use.

APT (Automatically Programmed Tools). The APT language was the product of the MIT developmental work on NC programming systems. Its development began in June, 1956, and it was first used in production around 1959. Today it is the most widely used language in the United States. Although first intended as a contouring language, modern versions of APT can be used for both positioning and continuous-path programming in up to five axes. Versions of APT for particular processes include APTURN (for lathe operations), APTMIL (for milling and drilling operations), and APTPOINT (for point-to-point operations).

ADAPT (Adaptation of APT). Several part programming languages are based directly on the APT program. One of these is ADAPT, which was developed by IBM under Air Force contract. It was intended to provide many of the features of APT but to utilize a smaller computer. The full APT program requires a computing system that would have been considered large by the standards of the 1960s. This precluded its use by many small and medium-sized firms that did not have access to a large computer. ADAPT is not as powerful as APT, but it can be used to program for both positioning and contouring jobs.

EXAPT (Extended subset of APT). This was developed in Germany starting around 1964 and is based on the APT language. There are three versions:

EXAPT I- designed for positioning (drilling and also straight-cut milling), EXAPT II- designed for turning, and EXAPT III- designed for limited contouring operations. One of the important features of EXAPT is that it attempts to compute optimum feeds and speeds automatically.

UNIAPT. The UNIAPT package represents another attempt to adapt the APT language to use on smaller computers. The name derives from the developer, the United Computing Corp. of Carson, California. Their efforts have provided a limited version of APT to be implemented on minicomputers, thus allowing many smaller shops to possess computer-assisted programming capacity.

SPLIT (Sundstrand Processing Language Internally Translated).

This is a proprietary system intended for Sundstrand's machine tools. It can handle up to five- axis positioning and possesses contouring capability as well. One of the unusual features of SPLIT is that the postprocessor is built into the program. Each machine tool uses its own SPLIT package, thus obviating the need for a special postprocessor.

COMPACT II. This is a package available from Manufacturing Data Systems, Inc. (MDSI), a firm based in Ann Arbor, Michigan. The NC language is similar to SPLIT in many of its features. MDSI leases the COMPACT II system to its users on a time-sharing basis. The part programmer uses a remote terminal to feed the program into one of the MDSI computers, which in turn produces the NC tape. The COMPACT II language is one of the most widely used programming languages. MDSI has roughly 3000 client companies which use this system.

PROMPT. This is an interactive part programming language offered by Weber N/C System, Inc., of Milwaukee, Wisconsin. It is designed for use with a variety of machine tools, including lathes, machining centers, flame cutters, and punch presses.

CINTURN II. This is a high-level language developed by Cincinnati Milacron to facilitate programming of turning operations.

The most widely used NC part programming language is APT, including its derivatives (ADAPT, EXAPT, UNIAPT, etc.).

THE APT LANGUAGE

In this section we present an introduction to the APT language for computer assisted part programming. Our objectives are to demonstrate the English-like statements of this NC language and to show how they are used to command the cutting tool through its sequence of machining operations.

APT is not only an NC language; it is also the computer program that performs the calculations to generate cutter positions based on APT statements. We will not consider the internal workings of the computer program. Instead, we will concentrate on the language that the part

programmer must use.

APT is a three-dimensional system that can be used to control up to five axes. We will limit our discussion to the more familiar three axes, x, y, and z, and exclude rotational coordinates. APT can be used to control a variety of different machining operations. We will cover only drilling and milling applications. There are over 400 words in the APT vocabulary. Only a small (but important) fraction will be covered here.

There are four types of statements in the APT language:

1. *Geometry statements.* These define the geometric elements that comprise the workpart. They are also sometimes called definition statements.
2. *Motion statements.* These are used to describe the path taken by the cutting tool.
3. *Postprocessor statements.* These apply to the specific machine tool and control system. They are used to specify feeds and speeds and to actuate other features of the machine.
4. *Auxiliary statements.* These are miscellaneous statements used to identify the part, tool, tolerances, and so on.

Geometry statements

To program in APT, the workpart geometry must first be defined. The tool is subsequently directed to move to the various point locations and along surfaces of the workpart which have been defined by these geometry statements. The definition of the workpart elements must precede the motion statements.

The general form of an APT geometry statement is this: An example of such a statement is

P1 = POINT/5.0, 4.0, 0.0

This statement is made up of three sections. The first is the symbol used to identify the geometric element. A symbol can be any

combination of six or fewer alphabetic and numeric characters. At least one of the six must be an alphabetic character. Also, although it may seem obvious, the symbol cannot be one of the APT vocabulary words.

The second section of the geometry statement is an APT vocabulary word that identifies the type of geometry element. Besides POINT, other geometry elements in the APT vocabulary include LINE, PLANE and CIRCLE.

The third section of the geometry statement comprises the descriptive data that define the element precisely, completely, and uniquely. These data may include quantitative dimensional and positional data, previously defined geometry elements, and other APT words.

The punctuation used in the APT geometry statement is illustrated in the example, Eq. .The statement is written as an equation, the symbol being equated to the surface type. A slash separates the surface type from the descriptive data. Commas are used to separate the words and-numbers in the descriptive data.

There are a variety of ways to specify the different geometry elements. The appendix at the end of this chapter presents a dictionary of APT vocabulary words as well as a sampling of statements for defining the geometry elements we will be using: points, lines, circles, and planes. The reader may benefit from a few examples.

To specify a line, the easiest method is by two points through which the line passes:

$$L3 = \text{LINE}/P3, P4$$

The part programmer may find it convenient to define a new line parallel to another line which has previously been defined. One way of doing this would be

$$L4 = \text{LINE}/P5, \text{PARLEL}, L3$$

This states that the line L4 must pass through point P5 and be parallel (PARLEL) line L3.

A plane can be defined by specifying three points through

which it passes: PL1 = PLANE/P1, P4, P5

Of course, the three points must not lie along a straight line. A plane can also be defined as being parallel to another plane, similar to the previous line parallelism statement.

PL2 = PLANE/P2, PARLEL, PL1

Plane PL2 is parallel to plane PL1 and passes through point P2. A circle can be specified by its center and its radius.

C1 = CIRCLE/CENTER, P1, RADIUS, 5.0

The two APT descriptive words are used to identify the center and radius. The orientation of the circle perhaps seems undefined. By convention, it is a circle located in the x-y plane.

There are several ground rules that must be followed in formulating an APT geometry statement:

1. The coordinate data must be specified in the order x, y, z. For example, the statement

P1 = POINT/5.0, 4.0, 0.0

is interpreted by the APT program to mean a point x = 5.0, y = 4.0, and z = 0.0.

2. Any symbols used as descriptive data must have been previously defined. For example, in the statement

P2 = POINT/INTOF, L1, L2

the two lines L1 and L2 must have been previously defined. In setting up the list of geometry statements, the APT programmer must be sure to define symbols before using them in subsequent statements.

3. A symbol can be used to define only one geometry element. The same symbol cannot be used to define two different elements. For example, the following sequence would be incorrect:

P1 =

POINT/1.0,

1.0, 1.0 P1 =

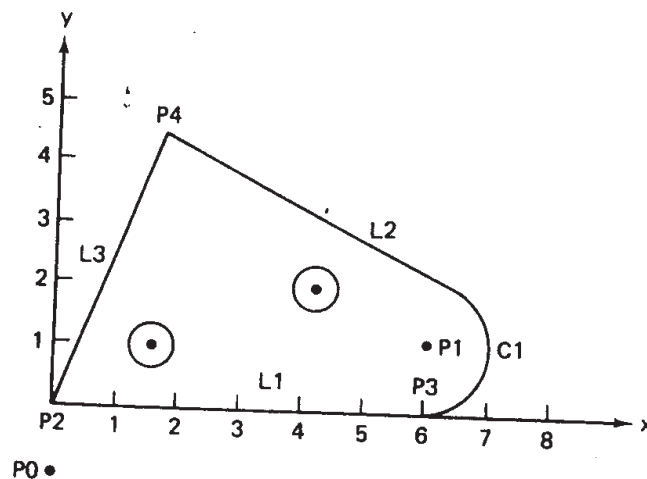
POINT/2.0,

3.0,4.0

4. Only one symbol can be used to define any given element. For example, the following two statements in the same program would render the program incorrect:

P1 = POINT/1.0, 1.0, 1.0 n = POINT/1.0, 1.0, 1.0

5. Lines defined in APT are considered to be of infinite length in both directions. Similarly, planes extend indefinitely and circles defined in APT are complete circles.



Workpart from previous figure redrawn with x-y coordinate system and geometric elements labeled

EXAMPLE

To illustrate some of these geometry statements we will define the geometry of the workpart shown in Figure. The drawing of the part is duplicated in Figure, except that we have added the coordinate axis system and labeled the various geometric elements. We also add the target point P0 to be used in subsequent motion commands.

P0 =
POINT/0, -
1.0, 0 P1 =

```

POINT/6.0,
1.125,0 P2 =
POINT/0, 0,
0
P3 = POINT/6.0, 0, 0
P4 =
POINT/1.75,
4.5, 0 L1 =
LINE/P2, P3
C1 = CIRCLE/CENTER, P1,
RADIUS, 1.125 L2 = LINE/P4,
LEFT, TANTO, C1
L3 = LINE/P2, P4
PL1 = PLANE/P2, P3, P4

```

Motion statements

APT motion statements have a general format, just as the geometry statements do. The general form of a motion statement is motion command/descriptive data. The example of a motion statement is GOTO/PI

The statement consists of two sections separated by a slash. The first section is the basic motion command, which tells the tool what to do. The second section is comprised of descriptive data, which tell the tool where to go. In the example statement above, the tool is commanded to go to point PI, which has been defined in a preceding geometry statement.

At the beginning of the motion statements, the tool must be given a starting point. This point is likely to be the target point, the location where the operator has positioned the tool at the start of the job. The part programmer keys into this starting position with the following statement:

```
FROM/TARG
```

The FROM is an APT vocabulary word which indicates that this is the initial point from which others will be referenced. In the statement above, TARG is the symbol given to the starting point. Any other APT symbol could be used to define the target point. Another way to make this statement is

```
FROM/-2.0, -2.0, 0.0
```


where the descriptive data in this case are the x, y, and z coordinates of the target point. The FROM statement occurs only at the start of the motion sequence.

It is convenient to distinguish between PTP movements and contouring movements when discussing the APT motion statements.

POINT-TO-POINT MOTIONS. There are only two basic PTP motion commands: GOTO and GODLTA. The GOTO statement instructs the tool to go to a particular point location specified in the descriptive data. Two examples would be:

GOTO/P2 GOTO/2.0, 7.0, 0.0

In the first statement, P2 is the destination of the tool point. In the second statement, the tool has been instructed to go to the location whose coordinates are $x = 2.0$, $y = 7.0$, and $z = 0$.

The GODLTA command specifies an incremental move for the tool. For example, the statement

GODLA/2.0, 7.0, 0.0

instructs the tool to move from its present position 2 in. in the x direction and 7 in. in the y direction. The z coordinate remains unchanged.

The GODLTA command is useful in drilling and related operations. The tool can be directed to a particular hole location with the GOTO statement. Then the GODLTA command would be used to drill the hole, as in the following sequence:

GOTO/P2 GODLTNO, 0, -1.5

GODLTNO, 0, +1.5

EXAMPLE

Previous example was a PTP job which was programmed manually. Let us write the APT geometry and motion statements necessary to perform the drilling portion of this job. We will set the plane defined by $z = 0$ about 1/4 in. above the part surface. The part will be

assumed to be 1/2 in. thick. The reader should refer back to Figures 8.3 and 8.4.

P1 = POINT/1.0, 2.0, 0

P2 = POINT/1.0, 1.0, 0 P3 = POINT/3.5, 1.5, 0

P0 = POINT/-1.0, 3.0, 2.0 FROM/P0

GOTO/P1 GODLTA/0, 0, -1.0

GODLTA/0, 0, + 1.0 GOTO/P2 GODLTA/0, 0, -1.0

GODLTA/0, 0, +1.0

GOTO/P3 GODLTA/0, 0, -1.0

GODLTA/0, 0, +1.0 GOTO/P0

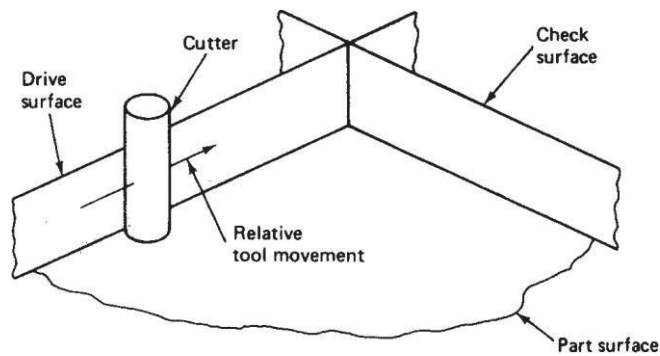
This is not a complete APT program because it does not contain the necessary auxiliary and postprocessor statements. However, the statement sequence demonstrates how geometry and motion statements can be combined to command the tool through a series of machining steps.

CONTOURING MOTIONS. Contouring commands are somewhat more complicated because the tool's position must be continuously controlled throughout the move. To accomplish this control, the tool is directed along two intersecting surfaces as shown in Figure 8.10. These surfaces have very specific names in APT:

1. *Drive surface.* This is the surface (it is pictured as a plane in Figure 8.10) that guides the side of the cutter.

2. *Part surface.* This is the surface (again shown as a plane in the figure) on which the bottom of the cutter rides. The reader should note that the "part surface" may or may not be an actual surface of the workpart. The part programmer must define this plus the drive surface for the purpose of maintaining continuous path control of the tool.

There is one additional surface that must be defined for APT contouring motions:



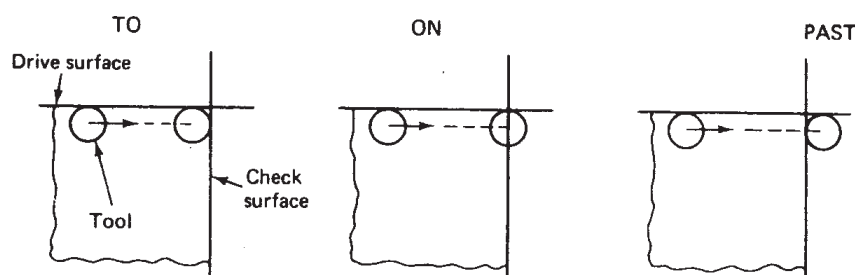
Three surfaces in APT contouring motions which guide the cutting tool.

3. *Check surface*. This is the surface that stops the movement of the tool in its current direction. In a sense, it checks the forward movement of the tool.

There are several ways in which the check surface can be used. This is determined by APT modifier words within the descriptive data of the motion statement. The three main modifier words are TO, ON, and PAST, and their use with regard to the check surface is shown in Figure. A fourth modifier word is TANTO. This is used when the drive surface is tangent to a circular check surface, as illustrated in Figure. In this case the cutter can be brought- to the point of tangency with the circle by use of the TANTO modifier word.

The APT contour motion statement commands the cutter to move along the drive and part surfaces and the movement ends when the tool is at the check surface. There are six motion command words:

GOLFT GOFWD GOUP GORGT GOBACK GODOWN

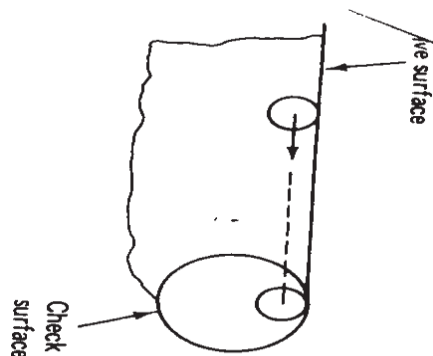


Use of APT modifier words in a motion statement: TO, ON and PAST. TO moves tool into initial contact with check surface. ON moves tool until tool center is

on check surface. PAST moves tool just beyond check surface.

Previous figure Use of APT modifier word TANTO. TANTO moves tool to point of tangency between two surfaces, at least one of which is circular.

Their interpretation is illustrated in Figure. In commanding the cutter, the programmer must keep in mind where it is coming from. As the tool reaches the new check surface, does the next movement involve a right turn or an upward turn or what? The tool is directed accordingly by one of the six motion words. In the use of these words, it is helpful for the programmer to assume the viewpoint that the workpiece remains

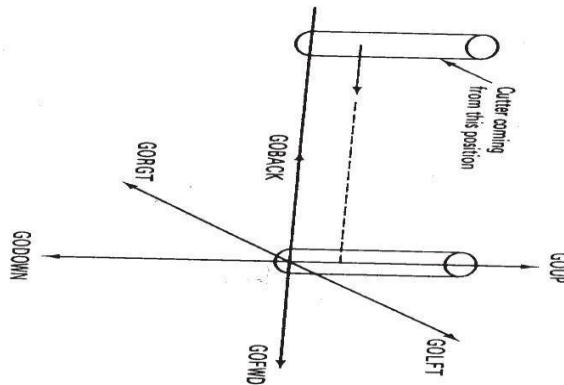


stationary and the tool is instructed to move relative to the piece. To begin the sequence of motion commands, the FROM statement, Eq. is used in the same manner, as for PTP moves. The statement following the FROM statement defines the initial drive surface, part surface, and check surface. The sequence is of the following form:

FROM/TARG

GO/TO, PL1, TO, PL2, TO, PL3

The symbol TARG represents the target point where the operator has set up the tool. The GO command instructs the tool to move to the intersection of the drive.



Use of APT motion commands.

surface (PL1), the part surface (PL2), and the check surface (PL3). The periphery of the cutter is tangent to PU and PU, and the bottom of the cutter is touching P12. This cutter location is defined by use of the modifier word TO. The three surfaces included in the GO statement must be specified in the order: drive surface first, part surface second, and check surface last.

Note that the GO/TO command is different from the GOTO command.

GOTO is used only for PTP motions. GO/TO is used to initialize the sequence of contouring motions.

After initialization, the tool is directed along its path by one of the six command words. It is not necessary to repeat the symbol of the part surface after it has been defined. For instance, consider Figure. The cutter has been directed from TARG to the intersection of surfaces PU, PL2, and PU. It is desired to move the tool along plane PL3. The following command would be used:

GORG/PL3, PAST, PL4

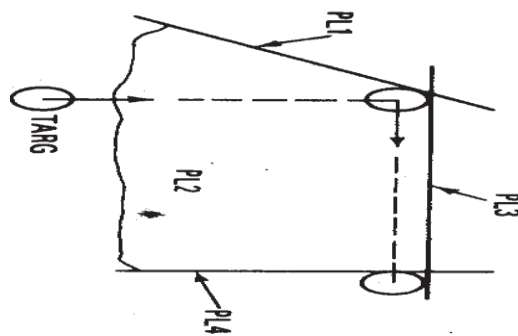
This would direct the tool to move along PL3, using it as the drive surface. The tool would continue until past surface PL4, which is

the new check surface. Although the part surface (PL2) may remain the same throughout the motion sequence, the drive surface and check surface are redefined in each new command.

Let us consider an alternative statement to the above which would accomplish the same motion but would lead to easier programming:

GORGT/U, PAST, L4

We have substituted lines L3 and L4 for planes PL3 and PL4, respectively. When looking at a part drawing, such as Figure, the sides of the part appear as lines. On the actual part, they are three-dimensional surfaces, of course. However, it would be more convenient for the part programmer to define these surfaces as lines and circles rather than planes and cylinders. Fortunately, the APT computer program allows the geometry of the part to be defined in this way. Hence the lines L3 and L4 in the foregoing motion statement are treated as the drive surface and check surface. This substitution can only be made when the part surfaces are perpendicular to the x- y plane.



Initialization of APT contouring motion sequence.

EXAMPLE

To demonstrate the use of the motion commands in a contouring sequence, we will refer back to the workpart of Example. The geometry statements were listed in this example for the part shown in Figure . Using the geometric elements shown in this figure, following is the list of motion statements to machine around the periphery of the part. The sequence begins with tool located at the target point PO.

FROM/PO

GO/TO, L1, TO,

PL1, TO, L3

GORGTL/L1,

TANTO, C1

GOFWD/C1,

PAST, L2

GOFWD/L2,

PAST, L3

GOLFT/L3,

PAST, L1

GOTO/PO

The reader may have questioned the location of the part surface (PL1) in the APT sequence. For this machining job, the part surface must be defined below the bottom plane of the workpiece in order for the cutter to machine the entire thickness of the piece. Therefore, the part surface is really not a surface of the part at all.

Example raises several other questions: How is the cutter size accounted for in the APT program? How are feeds and speeds specified? These and other questions are answered by the postprocessor and auxiliary statements.

Postprocessor statements

To write a complete part program, statements must be written that control the operation of the spindle, the feed, and other features of the machine tool. These are called postprocessor statements. Some of the common postprocessor statements that appear in the appendix at the end of the chapter are:

COOLNT/ RAPID END SPINDL/ FEDRAT/ TURRET/ MACHIN/

The postprocessor statements, and the auxiliary statements in the following section, are of two forms: either with or without the slash (/). The statements with the slash are self-contained. No additional data are needed. The APT words with the slash require descriptive data after the slash. These descriptions are given for each word in the appendix.

The FEDRAT/ statement should be explained. FEDRAT stands for feed rate and the interpretation of feed differs for different machining operations. In a drilling operation the feed is in the direction of the drill bit axis. However, in an end milling operation, typical for NC, the feed would be in a direction perpendicular to the axis of the cutter.

Auxiliary statements

The complete APT program must also contain various other statements, called auxiliary statements. These are used for cutter size definition, part identification, and so on. The following APT words used in auxiliary statements are defined in the appendix to this chapter:

CLPRNT INTOL/ CUTTER OUTTOL/ FINI PARTNO

The offset calculation of the tool path from the part outline is based on the CUTTER/definition. For example, the statement CUTTER/.500 would instruct the APT program that the cutter diameter is 0.500 in.

Therefore, the tool path must be offset from the part outline by 0.250 in.

EXAMPLE

We are now in a position to write a complete APT program. The work part of Example 8.4 will be used to illustrate the format of the APT program.

We will assume that the work piece is a low-carbon steel plate, which has previously been cut out in the rough shape of the part outline. The tool is a Y2-in.- diameter end-milling cutter. Typical cutting conditions might be recommended as follows: cutting speed = 573 rpm and feed = 2.29in.lmin.

Figure 8.15 presents the program with correct character spacing identified at the top as if it were to be keypunched onto computer cards. Modern NC programming systems utilize a CRT terminal for program

entry.

THE MACROSTATEMENT IN APT

In the preceding section we described the basic ingredients of the APT language. In the present section we describe a very powerful feature of APT, the MACRO statement. The MACRO feature is similar to a subroutine in FORTRAN and other computer programming languages. It would be used where certain motion sequences would be repeated several times within a program. The purpose in using a MACRO subroutine is to reduce the total number of statements required in the APT program, thus making the job of the part programmer easier and less time

Column				29
1	6	8	10	72
PARTNO			EXAMPLE PART	
			MACHIN/MILL, 1	
			CLPRNT	
			INTOL/.001	
			OUTTOL/.001	
			CUTTER/.500	
P0	=		POINT/0, -1.0, 0	
P1	=		POINT/6.0, 1.125, 0	
P2	=		POINT/0, 0, 0	
P3	=		POINT/6.0, 0, 0	
P4	=		POINT/1.75, 4.5, 0	
L1	=		LINE/P2, P3	
C1	=		CIRCLE/CENTER, P1, RADIUS, 1.125	
L2	=		LINE/P4, LEFT TANTO, C1	
L3	=		LINE/P2, P4	
PL1	=		PLANE/P2, P3, P4	
			SPINDL/573	
			FEDRAT/2.29	
			COOLNT/ON	
			FROM/P0	
			GO/TO, L1, TO, PL1, TO, L3	
			GORGTL1, TANTO, C1	
			GOFWD/C1, PAST, L2	
			GOFWD/L2, PAST, L3	
			GOLFT/L3, PAST, L1	
			RAPID	
			GOTO/P0	
			COOLNT/OFF	
			FINI	

APT program for Example

consuming. The MACRO subroutine is defined by a statement of the following format:

symbol = MACRO/parameter definition(s)

The rules for naming the MACRO symbol are the same as for

any other APT symbol. It must be six characters or fewer and at least one of the characters must be a letter of the alphabet. The parameter definition(s) following the slash would identify certain variables in the subroutines which might change each time the subroutine was called into use. Equation (8.6) would serve as the title and first line of a MACRO subroutine. It would be followed by the set of APT statements that comprise the subroutine. The very last statement in the set must be the APT word TERMAC. This signifies the termination of the MACRO.

To activate the MACRO subroutine within an APT program, the following call statement would be used:

CALL/symbol, parameter specification

The symbol would be the name of the MACRO that is to be called. The parameter specification identifies the particular values of the parameters that are to be used in this execution of the MACRO subroutines.

An example will serve to illustrate the use of the MACRO statement and how the MACRO would be called by the main APT program.

EXAMPLE

We will refer back to the drilling operations of Example. In this example the GODLTA sequence was repeated in the program a total of three times, once for each hole. This represents an opportunity to use the MACRO feature in the APT system. The four point locations (P0, P1, and P3) would be defined just as they are in Example. These points would be used in the MACRO subroutine and main APT program in the following way:

DRILL = MACRO/PX

GOTO/PX GODLTA/0, 0, -1.0

GODLTA/0, 0, + 1.0 TERMAC FROM/P0

CALL/DRILL, PX = P1 CALL/DRILL, PX = P2 CALL/DRILL,

PX = P3 GOTO/PO

In this example the number of APT motion statements in the main program has been reduced from 11 down to five. (If we include the MACRO subroutine in our line count, the reduction is from 11 statements to 10.) The reader can visualize the power of the MACRO feature for a programming job in which there are a large number of holes to be drilled. The savings in the required number of APT statements would approach $66^{2/3}\%$ in this case, since one call statement replaces three motion statements in the program.

The MACRO feature has many uses in APT. They are limited primarily by the imagination of the part programmer. Some of these uses will be considered in the exercise problems at the end of the chapter. It is even possible to have a CALL/ statement within one MACRO which refers to another MACRO subroutine. This might be used for example in a matrix of holes where both the x and y positions of the holes are changed with each drilling operation.

UNIT 4

GROUP TECHNOLOGY

INTRODUCTION

Group technology (abbreviated GT) is a manufacturing philosophy in which similar parts are identified and grouped together to take advantage of their similarities in manufacturing and design. Similar parts are arranged into part families. For example, a plant producing 10,000 different part numbers may be able to group the vast majority of these parts into 50 or 60 distinct families. Each family would possess similar design and manufacturing characteristics. Hence, the processing of each member of a given family would be similar, and this results in manufacturing efficiencies. These efficiencies are achieved in the form of reduced setup times, lower in-process inventories, better scheduling, improved tool control, and the use of standardized process plans. In some plants where GT has been implemented, the production equipment is arranged into machine groups, or cells, in order to facilitate work flow and parts handling.

In product design, there are also advantages obtained by grouping parts into families. For example, a design engineer faced with the task of developing a new part design must either start from scratch or pull an existing drawing from the files and make the necessary changes to conform to the requirements of the new part.

The problem is that finding a similar design may be quite difficult and time consuming. For a large engineering department, there may be thousands of drawings in the files, with no systematic way to locate the desired drawing. As a consequence, the designer may decide that it is easier to start from scratch in developing the new part. This decision is replicated many times over in the company, thus consuming valuable time creating duplicate or near-duplicate part designs. If an effective design- retrieval system were available, this waste could be avoided by permitting the engineer to determine quickly if a similar part

already exists. A simple change in an existing design would be much less time consuming than starting from scratch. This design-retrieval system is a manifestation of the group technology principle applied to the design function. To implement such a system, some form of parts classification and coding is required.

Parts classification and coding is concerned with identifying the similarities among parts and relating these similarities to a coding system. Part similarities are of two types: design attributes (such as geometric shape and size), and manufacturing attributes (the sequence of processing steps required to make the part). While the processing steps required to manufacture a part are usually correlated with the part's design attributes, this is not always the case. Accordingly, classification and coding systems are often devised to allow for differences between a part's design and its manufacture.

Whereas a parts classification and coding system is required in a design- retrieval system, it can also be used in computer-aided process planning (CAPP). Computer-aided process planning involves the automatic generation of a process plan (or route sheet) to manufacture the part. The process routing is developed by recognizing the specific attributes of the part in question and relating these attributes to the corresponding manufacturing operations.

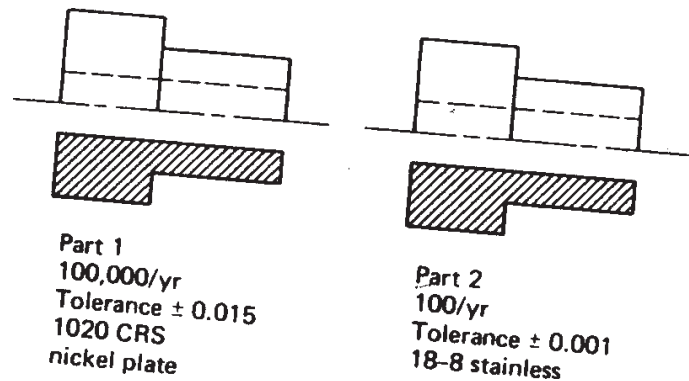
In the present chapter we develop the topics of group technology and p classification and coding. In the following chapter we present a discussion of computer-aided process planning and several related issues. Group technology and parts classification and coding are based on the concept of a part family.

PART FAMILIES

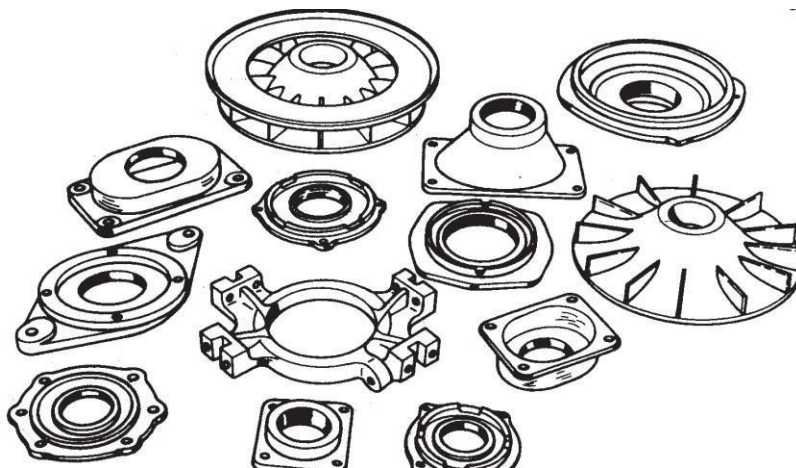
A part family is a collection of parts which are similar either because of geometric shape and size or because similar processing steps are required in their manufacture. The parts within a family are different, but their similarities are close enough to merit their identification as members of the part family. Figures show two part families. The two

parts shown in Figure are similar from design viewpoint but quite different in terms of manufacturing. The parts shown in Figure might constitute a part family in manufacturing, but their geometry characteristics do not permit them to be grouped as a design part family.

The part family concept is central to design-retrieval systems and current computer-aided process planning schemes. Another important manufacturing

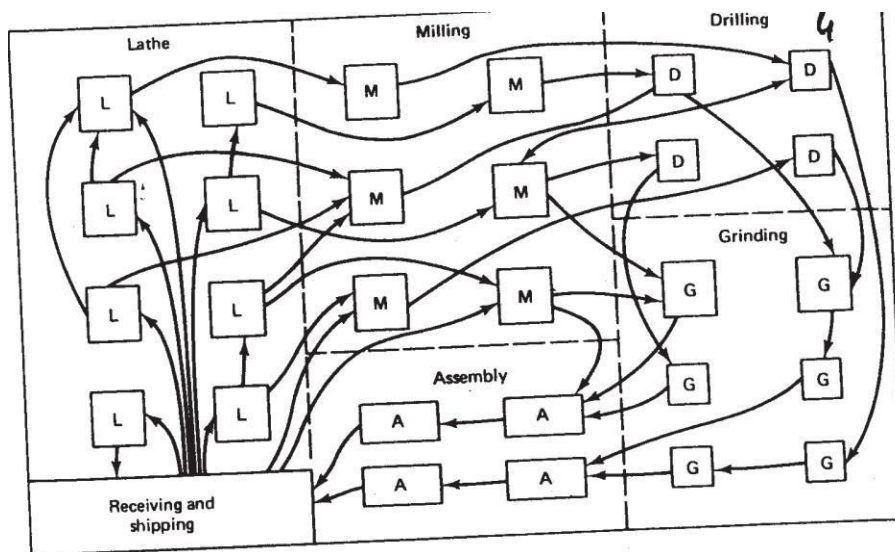


Two parts of identical shape and size but different manufacturing requirements.

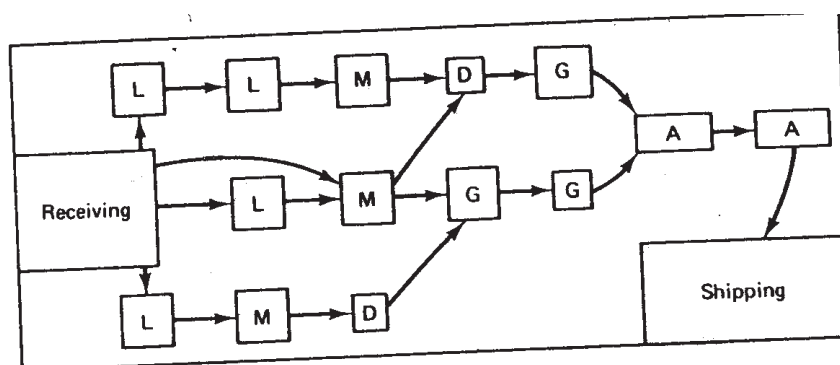


Thirteen parts with similar manufacturing process requirements but different design attributes. Advantage derived from grouping workparts into families can be explained with reference to Figures. Figure shows a process-type layout for batch production in a machine shop. The various machine tools are arranged by function. There is a lathe section,

milling machine section, drill press section, and so on. During the machining of a given part, the workpiece must be moved between sections, with perhaps the same section being visited several times. This results in a significant amount of material handling, a large in-process inventory, usually more setups than necessary, long manufacturing lead times, and high cost. Figure shows a production shop of supposedly equivalent capacity, but with the machines arranged into cells. Each cell is organized to specialize in the manufacture of a particular part family. Advantages are gained in the form of Process-type layout.



Process-type layout.



Group technology layout.

reduced workpiece handling, lower setup times, less in-process inventory, less floor space, and shorter lead times. Some of the manufacturing cells can be designed to form production flow lines, with

conveyors used to transport workparts between machines in the cell.

The biggest single obstacle in changing over to group technology from a traditional production shop is the problem of grouping parts into families. There are three general methods for solving this problem. All three methods are time consuming and involve the analysis of much data by properly trained personnel. The three methods are:

2. Production flow analysis (PFA)

3. Parts classification and coding system

The visual inspection method is the least sophisticated and least expensive method. It involves the classification of parts into families by looking at either the physical parts or photographs and arranging them into similar groupings. This method is generally considered to be the least accurate of the three.

The second method, production flow analysis, was developed by J. L. Burbidge. PFA is a method of identifying part families and associated machine tool groupings by analyzing the route sheets for parts produced in a given shop. It groups together the parts that have similar operation sequences and machine routings. The disadvantage of PFA is that it accepts the validity of existing route sheets, with no consideration given to whether these process plans are logical or consistent. The production flow analysis approach does not seem to be used much at all in the United States.

The third method, parts classification and coding, is the most time consuming and complicated of the three methods. However, it is the most frequently applied method and is generally recognized to be the most powerful of the three.

PARTS CLASSIFICATION AND CODING

This method of grouping parts into families involves an examination of the individual design and/or manufacturing attributes of each part. The attributes of the part are uniquely identified by means of a code number. This classification and coding may be carried out on the

entire list of active parts of the firm, or a sampling process may be used to establish the part families. For example, parts produced in the shop during a certain given time period could be examined to identify part family categories. The trouble with any sampling procedure is the risk that the sample may be unrepresentative of the entire population. However, this risk may be worth taking, when compared to the relatively enormous task of coding all the company's parts.

Many parts classification and coding systems have been developed throughout the world, and there are several commercially available packages being sold to industrial concerns. It should be noted that none of them has been universally adopted. One of the reasons for this is that a classification and coding system should be custom-engineered for a given company or industry. One system may be best for one company while a different system is more suited to another company.

TABLE Design and Manufacturing Part Attributes Typically Included in a Group Technology Classification System

Part design attributes

Basic external shape	Major dimensions
Basic internal shape	Minor dimensions
Length/diameter ratio	Tolerances
Material type	

S

urface finish Part function

Part manufacturing attributes

Major process	Operation sequence
Minor operations	Production time
Major dimensions	Batch size
Length/diameter ratio	Annual production
Surface finish	Fixtures needed

Machine tool

Cutting tools

Design systems versus manufacturing systems

Parts classification and coding systems divide themselves into one of three general categories:

1. Systems based on part design attributes
2. Systems based on part manufacturing attributes
3. Systems based on both design and manufacturing attributes

Systems in the first category are useful for design retrieval and to promote design standardization. Systems in the second category are used for computer-aided process planning, tool design, and other production-related functions. The third category represents an attempt to combine the functions and advantages of the other two systems into a single classification scheme. The types of design and manufacturing parts attributes typically included in classification schemes are listed in Table. It is clear that there is a certain amount of overlap between the design and manufacturing attributes of a part.

Coding system structure

A parts coding scheme consists of a sequence of symbols that identify the part's design and/or manufacturing attributes. The symbols in the code can be all numeric, all alphabetic, or a combination of both types. However, most of the common classification and coding systems use number digits only. There are basic code structures used in group technology applications:

1. Hierarchical structure
 2. Chain-type structure
 3. Hybrid structure, a combination of hierarchical and chain-type structures
- With the hierarchical structure, the interpretation of each succeeding symbol

depends on the value of the preceding symbols. Other names commonly used for this structure are monocode and tree structure. The hierarchical code provides a relatively compact structure which conveys much information about the part in a limited number of digits.

In the chain-type structure, the interpretation of each symbol in the sequence is fixed and does not depend on the value of preceding digits. Another name commonly given to this structure is polycode. The problem associated with

polycodes is that they tend to be relatively long . On the other hand, the use of a polycode allows for convenient identification of specific part attributes. This can be helpful in recognizing parts with similar processing requirements.

To illustrate the difference between the hierarchical structure and the chain type structure, consider a two-digit code, such as 15 or 25. Suppose that the first digit stands for the general part shape. The symbol 1 means round workpart and 2 means flat rectangular geometry. In a hierarchical code structure, the interpretation of the second digit would depend on the value of the first digit. If preceded by 1, the 5 might indicate some length/diameter ratio, and if preceded by 2, the 5 might be interpreted to specify some overall length. In the chain-type code structure, the symbol 5 would be interpreted the same way regardless of the value of the first digit. For example, it might indicate overall part length, or whether the part is rotational or rectangular.

Most of the commercial parts coding systems used in industry are a combination of the two pure structures. The hybrid structure is an attempt to achieve the best features of monocodes and polycodes. Hybrid codes are typically constructed as a series of short polycodes. Within each of these shorter chains, the digits are independent, but one or more symbols in the complete code number are used to classify the part population into groups, as in the hierarchical structure. This hybrid coding seems to best serve the needs of both design and production.

THREE PARTS CLASSIFICATION AND CODING SYSTEMS

When implementing a parts classification and coding system, most companies elect to purchase a commercially available package rather than develop their own. Inyong Ham recommends that the following factors be considered in selecting a parts coding and classification system:

Objective. The prospective user should first define the objective for the system. Will it be used for design retrieval or part-family manufacturing or both?

Scope and application. What departments in the company will use the system? What specific requirements do these departments have? What kinds of information must be coded? How wide a range of products must be coded? How complex are the parts, shapes, processes, tooling, and so forth?

Costs and time. The company must consider the costs of installation, training, and maintenance for their parts classification and coding system. Will there be consulting fees, and how much? How much time will be required to install the system and train the staff to operate and maintain it? How long will it be before the benefits of the system are realized?

Adapability to other systems. Can the classification and coding system be readily adapted to the existing company computer systems and data bases? Can it be readily integrated with other existing company procedures, such as process planning, NC programming, and production scheduling?

Management problems. It is important that all involved management personnel be informed and supportive of the system. Also, will there be any problems with the union? Will cooperation and support for the system be obtained from the various departments involved?

In the sections below, we review three parts classification and coding systems which are widely recognized among people familiar with GT:

- i . Opitz system
2. MICLASS system
3. CODE system

The Opitz classification system

This parts classification and coding system was developed by H. Opitz of the University of Aachen in West Germany. It represents one of the pioneering efforts in the group technology area and is perhaps the best

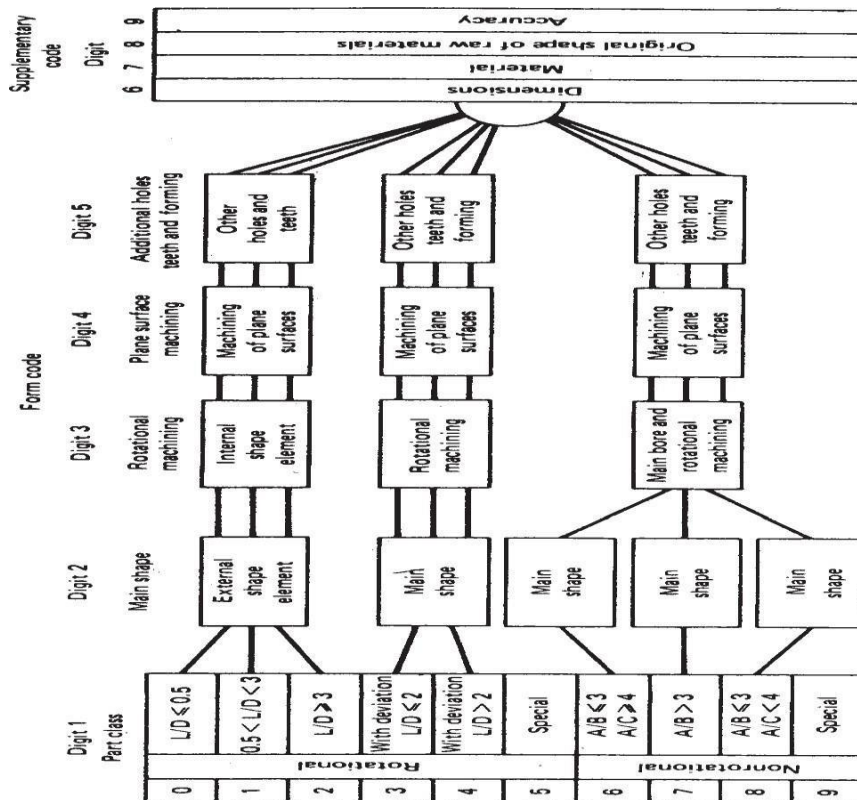
known of the classification and coding schemes.

The Opitz coding system uses the following

digit sequence: 12345 6789

ABCD

The basic code consists of nine digits, which can be extended by adding four more digits. The first nine digits are intended to convey both design and manufacturing data. The general interpretation of the nine digits is indicated in Figure. The first five digits, 12345, are called the "form code" and describe the primary design attributes of the part. The next four digits, 6789, constitute the "supplementary code."



It indicates some of the attributes that would be of use to manufacturing (dimensions, work material, starting raw workpiece shape and accuracy). The extra four digits, ABCD, are referred to as the "secondary code" and are intended to identify the production operation type and sequence. The secondary code can be designed by the firm to serve its own particular needs.

The complete coding system is too complex to provide a comprehensive description here. Opitz wrote an entire book on his system [12]. However, to obtain a general idea of how the Opitz system works, let us examine the first five digits of the code, the form code. The first digit identifies whether the part is a rotational or a non rotational part. It also describes the general shape and proportions of the part. We will limit our survey to rotational parts possessing no unusual features, those with code values 0, 1, or 2. See Figure for definitions. For this general class of workparts, the coding of the first five digits is given in Figure. An example will demonstrate the coding of a given part.

Digit 1		Digit 2		Digit 3		Digit 4		Digit 5	
Part class		External shape, external shape elements		Internal shape, internal shape elements		Plane surface machining		Auxiliary holes and gear teeth	
0	Rotational parts	L/D \leq 0.5		Smooth, no shape elements		No hole, no breakthrough		No auxiliary hole	
1		0.5 < L/D < 3		No shape elements		Surface plane and/or curved in one direction, external		Axial, not on pitch circle diameter	
2		L/D > 3		Thread		External plane surface related by graduation around a circle		Axial on pitch circle diameter	
3	Nonrotational parts	Stepped to one end or smooth		Functional groove		External groove and/or slot		Radial, not on pitch circle diameter	
4		No shape elements		No shape elements		External spline (polygon)		Axial and/or radial on PCD and/or other direction	
5		Thread		Thread		External plane surface and/or slot, external spline		Axial and/or radial on PCD and/or other directions	
6		Functional groove		Functional groove		Internal plane surface and/or slot		Spur gear teeth	
7		Functional cone		Functional cone		Internal spline (polygon)		Bevel gear teeth	
8		Operating thread		Operating thread		Internal and external polygon, groove and/or slot		Other gear teeth	
9		All others		All others		All others		All others	

The MICLASS System

MICLASS stands for Metal Institute Classification System and was developed by TNO, the Netherlands Organization for Applied Scientific Research. It was started in Europe about five years before being introduced in the United States in 1974. Today, it is marketed in the United States by the Organization for Industrial Research in Waltham, Massachusetts. The MICLASS system was developed to help automate and standardize a number of design, production, and management

functions. These include:

- Standardization of engineering drawings
- Retrieval of drawings according to classification number
- Standardization of process routing
- Automated process planning
- Selection of parts for processing on particular groups of machine tools
- Machine tool investment analysis

The MICLASS classification number can range from 12 to 30 digits. The first 12 digits are a universal code that can be applied to any part. Up to 18 additional digits can be used to code data that are specific to the particular company or industry. For example, lot size, piece time, cost data, and operation sequence might be included in the 18 supplementary digits.

The workpart attributes coded in the first 12 digits of the MICLASS number are as follows:

1st digit	Main shape
2nd and 3rd digits	Shape elements
4th digit	Position of shape elements
5th and 6th digits	Main dimensions
7th digit	Dimension ratio
8th digit	Auxiliary dimension
9th and 10th digits	Tolerance codes
11th and 12th digits	Material codes

The CODE system

The CODE system is a parts classification and coding system developed and marketed by Manufacturing Data Systems, Inc. (MDSI), of Ann Arbor, Michigan. Its most universal application is in design engineering for retrieval of part design data, but it also has applications in manufacturing process planning, purchasing, tool design, and inventory

control.

The CODE number has eight digits. For each digit there are 16 possible values (zero through 9 and A through F) which are used to describe the part's design and manufacturing characteristics. The initial digit position indicates the basic geometry of the part and is called the Major Division of the CODE system. This digit would be used to specify whether the shape was a cylinder, flat piece, block, or other. The interpretation of the remaining seven digits depends on the value of the first digit, but these remaining digits form a chain-type structure. Hence the CODE system possesses a hybrid structure.

BENEFITS OF GROUP TECHNOLOGY

Although group technology is expected to be an important principle in future production plants, it has not yet achieved the widespread application which might be expected. There are several reasons for this. First, as we have already indicated, there is the problem of rearranging the machines in the plant into GT cells. Many companies have been inhibited from adopting group technology because of the expense and disruption associated with this transition to GT machine cells. Second, there is the problem of identifying part families among the many components produced in the plant. Usually associated with this problem is the expense of parts classification and coding. Not only is this procedure expensive, but it also requires a considerable investment in time and personnel resources. Managers often feel that these limited resources can better be allocated to other projects than group technology with its uncertain future benefits. Finally, it is common for companies to encounter a general resistance among its operating personnel when changeover to a new system is contemplated.

When these problems are solved and group technology is applied, the company will typically realize benefits in the following areas:

Product design Tooling and setups Materials handling, Production and inventory control

Employee satisfaction, Process planning procedures

COMPUTER-AIDED PROCESS PLANNING

THE PLANNING FUNCTION

This chapter examines several process planning functions which can be implemented by computer systems. Process planning is concerned with determining

the sequence of individual manufacturing operations needed to produce a given part or product. The resulting operation sequence is documented on a form typically referred to as a route sheet. The route sheet is a listing of the production operations and associated machine tools for a workpart or assembly.

Closely related to process planning are the functions of determining appropriate cutting conditions for the machining operations and setting the time standards for the operations. All three functions-planning the process, determining the cutting conditions, and setting the time standards-have traditionally been carried out as tasks with a very high manual and clerical content. They are also typically routine tasks in which similar or even identical decisions are repeated over and over. Today, these kinds of decisions are being made with the aid of computers. In the first four sections of this chapter we consider the process planning function and how computers can be used to perform this function.

Traditional process planning

There are variations in the level of detail found in route sheets among different companies and industries. In the one extreme, process planning is accomplished by releasing the part print to the production shop with the instructions “make to drawing.” Most firms provide a more detailed list of steps describing each operation and identifying each work center. In any case, it is traditionally the task of the manufacturing engineers or industrial engineers in an organization to write these process plans for new part designs to be produced by the shop. The process planning procedure is very much dependent on the experience and judgment of the planner. It is the manufacturing engineer's responsibility to determine an optimal routing for each new part design. However, individual engineers each have their own opinions about what constitutes the best routing. Accordingly, there are differences among the operation sequences developed by various planners. We can illustrate rather dramatically these differences by means of an example.

In one case cited, a total of 42 different routings were developed for various sizes of a relatively simple part called an "expander sleeve." There were a total of 64 different sizes and styles, each with its own part number. The 42 routings included 20 different machine tools in the shop. The reason for this absence of process

standardization was that many different individuals had worked on the parts: 8 or 9 manufacturing engineers, 2 planners, and 25 NC part programmers. Upon analysis, it was determined that only two different routings through four machines were needed to process the 64 part numbers. It is clear that there are potentially great differences in the perceptions among process planners as to what constitutes the "optimal" method of production.

In addition to this problem of variability among planners, there are often difficulties in the conventional process planning procedure. New machine tools in the factory render old routings less than optimal. Machine breakdowns force shop personnel to use temporary routings, and these become the documented routings even after the machine is repaired. For these reasons and others, a significant proportion of the total number of process plans used in manufacturing are not optimal.

Automated process planning

Because of the problems encountered with manual process planning, attempts have been made in recent years to capture the logic, judgment, and experience required for this important function and incorporate them into computer programs. Based on the characteristics of a given part, the program automatically generates the manufacturing operation sequence. A computer-aided process planning (CAPP) system offers the potential for reducing the routine clerical work of manufacturing engineers. At the same time, it provides the opportunity to generate production routings which are rational, consistent, and perhaps even optimal. Two alternative approaches to computer-aided process planning have been developed. These are:

1. Retrieval-type CAPP systems (also called variant systems)
2. Generative CAPP systems

The two types are described in the following two sections.

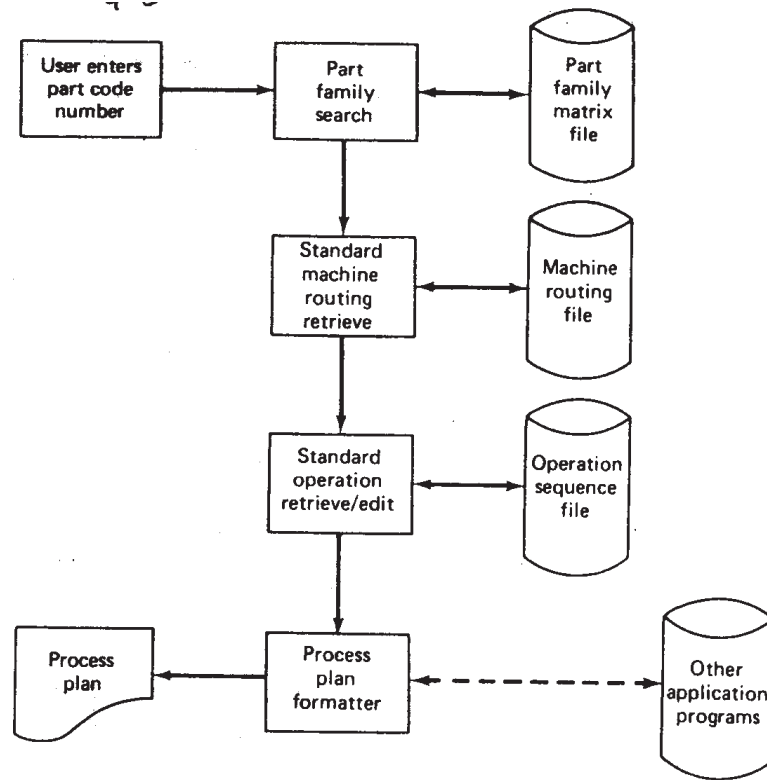
RETRIEVAL - TYPE PROCESS PLANNING SYSTEMS

Retrieval-type CAPP systems use parts classification and coding and group technology as a foundation. In this approach, the parts produced in the plant are

grouped into part families, distinguished according to their manufacturing characteristics. For each part family, a standard process plan is established. The standard process plan is stored in computer files and then retrieved for new workparts which belong to that family. Some form of parts classification and coding system is required to organize the computer files and to permit efficient retrieval of the appropriate process plan for a new workpart. For some new parts, editing of the existing process plan may be required. This is done when the manufacturing requirements of the new part are slightly different from the standard. The machine routing may be the same for the new part, but the specific operations required at each machine may be different. The complete process plan must document the operations as well as the sequence of machines through which the part must be routed. Because of the alterations that are made in the retrieved process plan, these CAPP systems are sometimes also called by the name 'variant system.'

Figure will help to explain the procedure used in a retrieval process planning system. The user would initiate the procedure by entering the part code number at a computer terminal. The CAPP program then searches the part family matrix file to determine if a match exists. If the file contains an identical code number, the standard machine routing and operation sequence are retrieved from the respective computer files for display to the user. The standard process plan is examined by the user to permit any necessary editing of the plan to make it compatible with the new part design. After editing, the process plan formatter prepares the paper document in the proper form.

If an exact match cannot be found between the code numbers in the computer file and the code number for the new part, the user may search the machine routing file and the operation sequence file for similar parts that could be used to develop the plan for the new part. Once the process plan for a new part code number has been entered, it becomes the standard process for future parts of the same classification.



Information flow in a retrieval-type computer-aided process planning system.

In Figure the machine routing file is distinguished from the operation sequence file to emphasize that the machine routing may apply to a range of different part families and code numbers. It would be easier to find a match in the machine routing file than in the operation sequence file. Some CAPP retrieval systems would use only one such file which would be a combination of operation sequence file and machine routing file.

The process plan formatter may use other application programs. These could include programs to compute machining conditions, work standards, and standard costs. Standard cost programs can be used to determine total product costs for pricing purposes.

A number of retrieval-type computer-aided process planning systems have been developed. These include MIPLAN, one of the MICLASS modules [6,20] the

CAPP system developed by Computer-Aided Manufacturing-International [1], COMCAPP V by MDSI, and systems by individual companies [10]. We will use MIPLAN as an example to illustrate these industrial systems.

GENERATIVE PROCESS PLANNING SYSTEMS

Generative process planning involves the use of the computer to create an individual process plan from scratch, automatically and without human assistance. The computer would employ a set of algorithms to progress through the various technical and logical decisions toward a final plan for manufacturing. Inputs to the system would include a comprehensive description of the workpart. This may involve the use of some form of part code number to summarize the workpart data, but does not involve the retrieval of existing standard plans. Instead, the general CAPP system synthesizes the design of the optimum process sequence, based on an analysis of part geometry, material, and other factors which would influence manufacturing decisions.

In the ideal generative process planning package, any part design could be presented to the system for creation of the optimal plan. In practice, current generative-type systems are far from universal in their applicability. They often fall short of a truly generative capability, and they are developed for a some limited range of manufacturing processes.

We will illustrate the generative process planning approach by means of a system called GENPLAN developed at Lockheed-Georgia Company.

BENEFITS OF CAPP

Whether it is a retrieval system or a generative system, computer-aided process planning offers a number of potential advantages over manually oriented process planning.

1. *Process rationalization.* Computer-automated preparation of operation routings is more likely to be consistent, logical, and optimal than its manual counterpart. The process plans will be consistent because

the same computer software is being used by all planners. We avoid the tendency for drastically different process plans from different planners, as described in Section 13.1. The process plans tend to be more logical and optimal because the company has presumably incorporated the experience and judgment of its best manufacturing people into the process planning computer software.

2. *Increased productivity of process planners.* With computer-aided process planning, there is reduced clerical effort, fewer errors are made, and the planners have immediate access to the process planning data base. These benefits translate into higher productivity of the process planners. One system was reported to increase productivity by 600% in the process planning function [10].

3. *Reduced turnaround time.* Working with the CAPP system, the process planner is able to prepare a route sheet for a new part in less time compared to manual preparation. This leads to an overall reduction in manufacturing lead time.

4. *Improved legibility.* The computer-prepared document is neater and easier to read than manually written route sheets. CAPP systems employ standard text, which facilitates interpretation of the process plan in the factory.

5. *Incorporation of other application programs.* The process planning system can be designed to operate in conjunction with other software packages to automate many of the time-consuming manufacturing support functions.