SPECIAL INDIAN EDITION

Second Edition

CAD/CAM

Theory and Practice

Ibrahim Zeid
R Sivasubramanian

Tata McGraw Hill

Special Indian Edition 2009

Published in India by arrangement with the McGraw-Hill Companies, Inc., New York

Sales Territories: India, Pakistan, Nepal, Bangladesh, Sri Lanka and Bhutan

CAD/CAM: Theory and Practice

Fifth reprint 2010
RAZCRRYZRABLR

ISBN (13): 978-0-07-015134-5
ISBN (10): 0-07-015134-2

Managing Director: *Ajay Shukla*

General Manager: Publishing—SEM & Tech Ed: *Vibha Mahajan*
Manager Sponsoring: *Shalini Jha*
Sr. Editorial Executive: *Surabhi Shukla*
Asst. Development Editor: *Devshree Lohchab*
Executive—Editorial Services: *Sohini Mukherjee*
Senior Production Manager: *P L Pandita*

General Manager: Marketing—Higher Education: *Michael J Cruz*
Sr. Product Manager: SEM & Tech Ed: *Biju Ganesan*

General Manager—Production: *Rajender P Ghansela*
Asst. General Manager—Production: *B L Dogra*

Typeset at Mukesh Technologies Pvt. Ltd., #10, 100 Feet Road, Ellapillaichavadi, Pondicherry 605 005 and printed at Adarsh Printers, C-50/51, Naveen Shahdara, Delhi 110 032

Cover: SDR

The McGraw-Hill Companies

Contents

PART-II 133

GEOMETRIC MODELING

Preface to the Revised Edition

Computer, today, is a very versatile and powerful tool in the hands of design engineers—a tool to design, draft, model, analyze, simulate, optimize and manufacture. The use of computer is not only helpful to handle design tasks but also in the manufacture of good quality products. Developments in the field of CAD/CAM, FMS and CIM are taking place continuously.

The response to the previous edition of the book was indeed encouraging. Considering the suggestions from the readers of the book and the latest technical advancements in this particular field, I feel delighted to present this book in its revised form. The need of this presentation was felt because information pertaining to this area of current technology is available in a very scattered form. This book, in a concise and compact form, gives the related fundamental concepts and up-to-date information on CAD/CAM. This edition of the book aims to present CAD/CAM such that the application aspect is given prime importance. Moreover, many of the CAM activities also needed to be included in the text. Hence, I have touched upon some of the manufacturing activities such as FMS, Group Technology, Computer Aided Process Planning and Computer Aided Quality Control. The emphasis in this book is on design and manufacture based on personal computers as these are popular today.

This book is intended to be used as a textbook by the undergraduate and postgraduate engineering students and teachers and as a reference by the practicing engineers, engineers employed in R& D establishments. Numerous solved problems from various Technical Universities are presented in order to make the book a self-learning tool.

The major modifications carried out in this revised edition are as follows:

- Topics on output devices such as pen plotters, hardcopy unit, electrostatic plotters and computer output-to-microfilm unit are included.
- Computer Aided Process Planning (CAPP), types of CAPP and advantages of CAPP have been added.
- Line sketches of output devices, RAM and other hardware devices are shown in respective chapters.
- Computer Aided Quality Control (CAQC), its advantages and limitations, various techniques of CAQC with simple sketches have been included.

I also wish to extend my acknowledgement and record my sincere thanks to the authors and publishers whose works have been referred to while preparing the textbook.

I am immensely thankful to Tata McGraw Hill Education for bringing out the book in the present form with an appealing look. I am highly indebted to the learned professors and faculty members of research and technical institutes for their magnificent response to this book.

I would like to thank the management, the principal Dr R Prabhakar of Coimbatore Institute of Technology, Coimbatore, for the encouragement and facilities provided. Thanks are also due to my Head of the Mechanical Engineering Department, Dr V Selladurai, and other colleagues of the Department who have contributed to this work in several ways.

Finally, I would thank the various reviewers who took out time to review the book. Their names are given below.

P M Pandey	*Indian Institute of Technology Delhi, Delhi*
V K Jain	*Indian Institute of Technology Kanpur* *Kanpur, Uttar Pradesh*
S K Sharma	*IT BHU, Varanasi, Uttar Pradesh*
S S Roy	*NIT Durgapur, West Bengal*
Sujeet Kumar Mishra	*Birla Institute of Technology, Mesra, Ranchi, Jharkhand*
K P Karunakaran	*Indian Institute of Technology Bombay* *Mumbai, Maharashtra*
H K Raval	*SVNIT Surat, Gujarat*
R Manu	*NIT Calicut, Kerala*
N H Jayadas	*School of Engineering,* *Cochin University of Science and Technology* *Cochin, Kerala*
K V K Gangadharan	*NIT, Surathkal, Karnataka*
A Venugopal	*NIT Warangal, Andhra Pradesh*
P Bangaru Babu	*NIT Warangal, Andhra Pradesh*
K Mallikarjuan Rao	*JNTU College of Engineering* *Kakinada, Andhra Pradesh*

I solicit comments and suggestions for improving this book. No effort will be spared to set the mistakes right and to include improvements.

R Sivasubramanian

PART I

OVERVIEW OF CAD/CAM SYSTEMS

1

Introduction

1.1 CAD/CAM CONTENTS AND TOOLS

In engineering practice, CAD/CAM has been utilized in different ways by different people. Some utilize it to produce drawings and document designs. Others may employ it as a visual tool by generating shaded images and animated displays. A third group may perform engineering analysis of some sort on geometric models such as finite element analysis. A fourth group may use it to perform process planning and generate NC part programs. In order to establish the scope and definition of CAD/CAM in an engineering environment and identify existing and future related tools, a study of a typical product cycle is necessary. Figure 1.1 shows a flowchart of such a cycle.

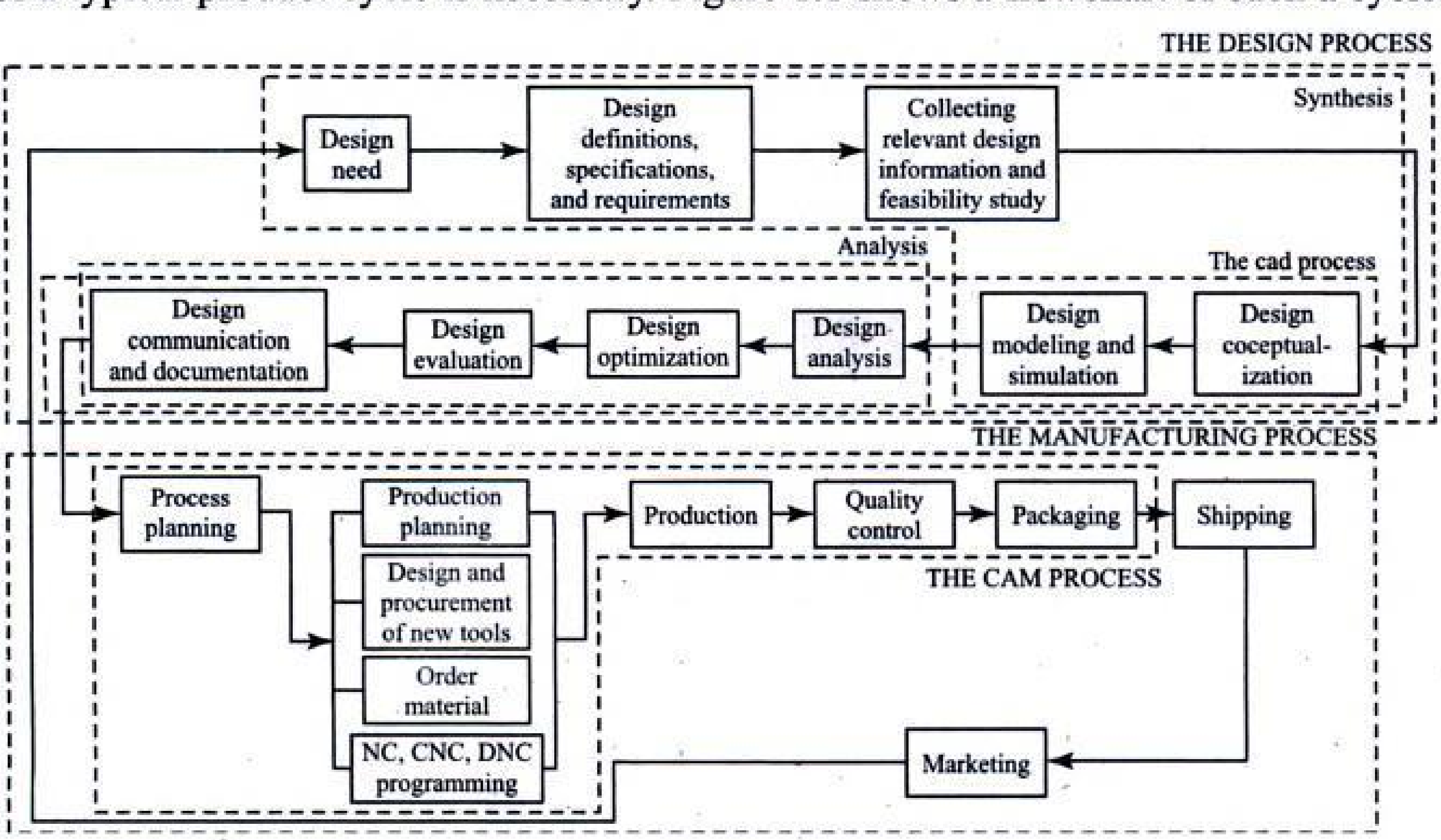


Fig. 1.1 *Typical Product Cycle.*

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

evaluations are hard to identify, they may include the proper sizing of the model after the analysis is performed to ensure engineering practices such as gradual change in dimensioning and avoidance of stress concentrations. Adding tolerances, performing tolerance analysis, generating a bill of materials and investigating the effect of manufacturing on the design by utilizing NC packages are also valuable tools that are available to designers.

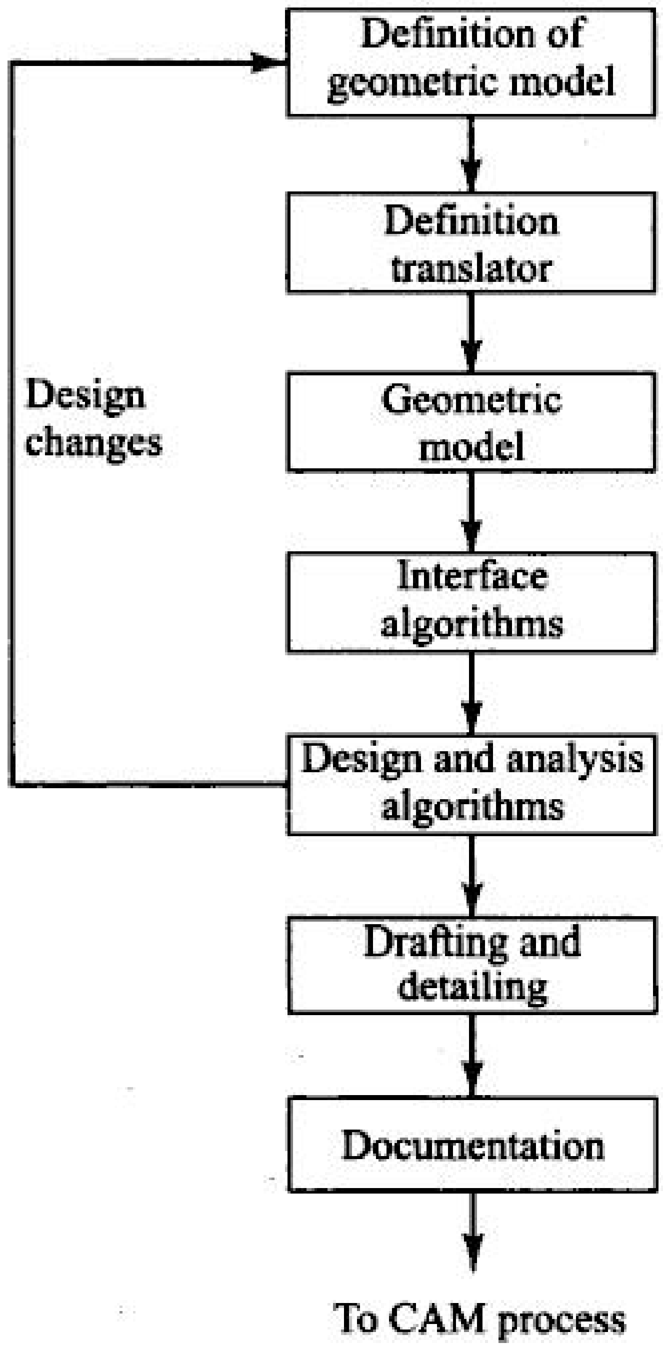


Fig. 1.2 *Implementation of a Typical CAD Process on a CAD/CAM System*

Table 1.1 *CAD Tools required to Support the Design Process*

Design phase	*Required CAD tool(s)*
Design conceptualization	Geometric modeling techniques; graphics aids, manipulations and visualization
Design modeling and simulation	Same as above; animation; assemblies; special modeling packages
Design analysis	Analysis packages; customized programs and packages
Design optimization	Customized applications; structural optimization
Design evaluation	Dimensioning; tolerances; bill of materials; NC
Design communication and documentation	Drafting and detailing; shaded images

fact that it provides unique and unambiguous geometric representations of solids which, in turn, help automate and/or support design and manufacturing applications. Major solid modeling systems now exist such as GMSolid (General Motors), Romulus (ShapeData), PADL-2 (University of Rochester), SynthaVision-based (Applicon) and Solidesign (Computervision). The hardware has kept pace with the software and applications developments. In addition to developments of special computer hardware, improved displays, almost real-time simulation hardware and microcomputer-based and workstation-based CAD/CAM systems have been emerging rapidly into the market. Most recent systems are as capable as most of the mainframe-based systems that appeared a decade ago.

While the CAD/CAM field has come a long way in four decades thus far, its future certainly holds many challenges. Extrapolating this existing history reveals that the decade of the 1990s and beyond will represent the age where the fruits of the current research efforts in integrating and automating design and manufacturing applications will mature. It is anticipated that new design and manufacturing algorithms and capabilities will become available. These applications will be supported by better and faster computing hardware and efficient networking and communication software.

1.3 CAD/CAM MARKET TRENDS[1]

The CAD/CAM market has always been in a state of flux since it began. New hardware configurations and software concepts are continuously developed. Chapters 2 and 3 cover the details and definitions of existing concepts. It is most likely that the market will continue to change rapidly over the next decade. The emergence of microcomputers and engineering workstations have contributed to the decline in price which make CAD/CAM systems more affordable by small businesses. In constant 1985 dollars, the U.S. market for CAD/CAM systems is expected to grow from about $3 billion in 1985 to $8 billion by 1992. In current dollar units, the 1992 sales figures may approach $12 billion, assuming an average inflation rate of 5.2 percent per year for the period 1985-1992. The average yearly growth rate in real (constant dollar) terms is expected to be 15 percent per year for the period; that is, the real growth is to slow down from an initial 21 percent per year in 1985 to about 10 percent per year in 1992. In current dollars, the average yearly growth is 21 percent per year. Growth is expected to decline over the seven-year span, 1985–1992, from 27 percent per year in 1985 to 17 percent per year in 1992.

In the next few years, it is expected that purchases of large numbers of personal computers and workstations will be made. Traditional turnkey systems will continue to be sold but not at the rate seen in the past. These will be aimed at the project group which works together, at the drafting and drawing archival environment and

[1] This section includes excerpts from the Frost and Sullivan Report 1564.

Table 1.4 *CAD/CAM Market, United States—sales by Application Type*

	1984	*1986*	*1988*	*1990*	*1992*	*Growth per year 1985–1992, %*
Sales by application type, 1985 $ millions						
Mechanical	1030	1426	1891	2390	2898	13.3
Electronics	502	827	1234	1688	2149	18.7
Electrical	100	133	167	197	223	9.9
AEC	404	621	891	1197	1519	17.2
Mapping	160	205	246	278	303	7.6
Technical illustration	141	207	285	369	452	14.9
Other	146	207	275	343	407	12.9
Total	2484	3626	4990	6460	7951	14.9
Percent shares						
Mechanical	41.5	39.3	37.9	37.0	36.5	– 1.4
Electronics	20.2	22.8	24.7	26.1	27.0	3.3
Electrical	4.0	3.7	3.3	3.0	2.8	– 4.4
AEC	16.3	17.1	17.9	18.5	19.1	1.9
Mapping	6.4	5.7	4.9	4.3	3.8	– 6.4
Technical illustration	5.7	5.7	5.7	5.7	5.7	0.0
Other	5.9	5.7	5.5	5.3	5.1	– 1.8
Total	100.0	100.0	100.0	100.0	100.0	
Growth rates						
Mechanical		18.0	14.3	11.8	9.4	– 8.3
Electronics		27.7	20.6	15.9	11.6	– 12.2
Electrical		15.3	10.9	8.1	6.0	– 12.2
AEC		23.9	18.7	15.1	11.6	– 9.9
Mapping		13.0	8.5	5.7	4.1	– 15.7
Technical illustration		21.1	16.3	13.0	9.9	– 10.1
Other		19.0	14.2	10.9	8.2	– 11.7
Total		20.9	16.3	13.1	10.1	– 9.8

Source: Frost and Sullivan Report 1564.

1.4 DEFINITION OF CAD/CAM TOOLS

In Sec. 1.1 we have defined CAD and CAM as subsets of the design and manufacturing processes respectively. Tables 1.1 and 1.2 list some CAD and CAM tools. In this section, we define CAD, CAM and CAD/CAM tools. These definitions are based on practical and industrial use of the CAD/CAM technology. The definitions are broad enough to encompass many of the details that readers may wish to add.

Employing their constituents, CAD tools can be defined as the intersection of three sets: geometric modeling, computer graphics and the design tools.

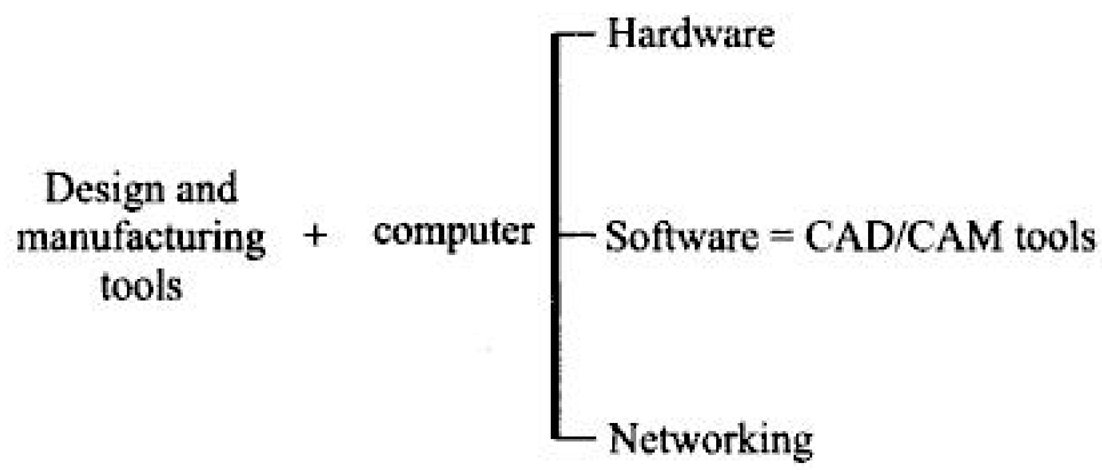


Fig. 1.11 *Definition of CAD/CAM Tools based on their Implementation in an Engineering Environment*

surfaces. Therefore, understanding the utilization and implementation of the CAD/CAM technology in an industrial environment helps to close the gap between creating the technology, managing it, using it and more importantly learning it. Figure 1.12 shows, in a general sense, how a typical CAD/CAM system is utilized in a typical industrial environment. The figure shows the major components or packages that exist. The detailed capabilities and functions of each package as well as the various types of existing user interface are what makes these systems look entirely different. As a matter of fact, practical experience has proven that learning one system is sufficient to learn another one at a much faster pace. This faster pace is attributed to dealing with the same functions. All the user has to do is to adjust to the user interface and the management hierarchy of the new system. One might conclude that learning the generic basic concepts behind these systems does not only speed up the training curve of perspective users but it also helps them utilize the technology productively.

The principal packages available consist of geometric modeling and graphics, design, manufacturing and programming software. The three available types of modeling are wireframes, surfaces and solid modeling. The underlying theories of these modeling types are presented in Chapters. 4, 5 and 6 respectively. A wide variety of geometric entities or items are accessible by the designer under each modeling technique. Graphics encompass such functions as geometric transformations, drafting and documentation, shading, coloring and layering. The design applications package includes mass property calculations, finite element modeling and analysis, tolerance stack analysis, mechanisms modeling and interference checking. If a design or manufacturing application is encountered where the system's standard software can not be utilized, a customized software may be developed using the programming language provided. These languages are typically either system dependent or independent. Once the design is complete, drafting and documentation are performed on the model database. The model is now ready for CAM applications such as process planning, tool path generation and verification, inspection and assembly.

This text is written with the above industrial look at CAD/CAM in mind. A coherent realization of the current CAD/CAM tools and their relationships to one another form an essential core to the learning process. Thus learning the basis of existing tools enhances both the utilization of current systems and the development of new design and manufacturing applications.

solids. Chapters 4, 5 and 6 cover these techniques respectively in a consistent fashion. Each chapter deals first with the various geometric entities provided by each modeling technique. This is followed by the manipulations' functions such as displaying, trimming, intersection and projection. Examples are provided throughout the chapters. In addition, there is a design and applications section in each chapter which combines the material in each chapter into relevant engineering problems. Chapter 7 ends this part by covering the exchange of CAD/CAM databases among various CAD/CAM systems.

Part III is a logical flow of Part II. It treats the basic graphics concepts which when applied to geometric models result in the versatile and basic visual tools CAD/CAM systems provide. Chapter 8 introduces the concepts of homogeneous transformation. Chapter 9 covers algorithms for hiding surfaces and solids, shading and coloring.

While the first three parts of the book deal with the fundamentals of CAD/CAM, the last three parts harness the acquired knowledge and cover the related CAD and CAM tools and applications. Part IV deals with the interactive tools that are typically available to designers on all today's CAD/CAM systems. These tools facilitate the creation and management of CAD/CAM databases.

Chapter 10 presents the available graphics aids. Some of these aids such as geometric modifiers and grids help minimize the amount of calculations to create a geometric model. Others such as names and groups help manage and manipulate the model. This chapter also introduces the concept of layering which is important for managing graphics of large and extensive models. Chapter 11 deals with manipulating graphics of geometric models for creation, visualization and calculation purposes. Chapter 12 covers animation procedures. Examples of how to employ animation in engineering studies are included. In many engineering problems, there is a need to customize the available tools. In the case of the CAD/CAM subject, it is desired to customize CAD/CAM systems to meet particular needs and increase productivity. Examples and problems are covered in the chapter.

Part V describes the most widely available CAD applications on CAD/CAM systems. The basic concepts of each application, its implementation into software and its utilization in design problems are covered. Chapter 13 deals with finite element modeling and analysis and discusses the issues related to interfacing modeling and analysis. Chapter 14 discusses how design projects with CAE focus can be developed to reflect the capabilities of CAD/CAM systems.

Part VI discusses the issue of integrating CAD and CAM databases. Chapter 15 describes the available prominent CAM applications such as process planning and part programming. The chapter also discusses the influence of CAD on CAM.

PROBLEMS

Part 1: Theory

Short Answer Questions/Objective Type Questions

1.1 What is CAD/CAM?
1.2 What is process planning? What is its importance in product development?

CAD/CAM hardware has progressed steadily from slow specialized systems to fast standard ones. In the 1970s and early 1980s, the majority of commercially available CAD/CAM systems were based on 16-bit word minicomputers. It was typical then to find turnkey vendors who designed and manufactured their own hardware to run their software. As a result, users of such systems were faced with the problem of dealing with nonstandard operating systems which most often resulted in system isolation. In addition, interfacing and networking these systems with other computers were not feasible. Other vendors have envisioned the importance of standard hardware and have consequently adapted it to fit the architecture and configuration of their systems. Bundled systems were common at that time. A bundled system is a packaged hardware and software that is sold and maintained by a single vendor. It is also known as a turnkey system. With the constantly increasing demand in performance and diversity of utilizations, CAD/CAM systems have migrated to 32-bit word minicomputers to provide the accuracy and support calculation intensity required by applications such as finite element analysis and solid modeling. These systems are commonly unbundled and offer users the flexibility to choose the optimum hardware and software configurations to meet their needs.

The majority of today's CAD/CAM systems utilizes open hardware architecture and standard operating systems. Open hardware architecture implies that CAD/CAM vendors no longer design and manufacture their own hardware platforms. Instead the CAD/CAM industry relies upon the giant general-purpose computer companies and smaller firms that specialize in engineering workstations. Thus users can network the CAD/CAM systems to other computer systems as well as hardwire them to various manufacturing cells and facilities. They can also run third party software to augment the analysis capabilities typically provided by CAD/CAM vendors. With the advancements in IC (Integrated Circuit), PC (Printed Circuit) and VLSI (Very Large Scale Integration) technology, current CAD/CAM systems are based on the workstation concept. Such a concept provides both single-user and timesharing environments. These advancements have resulted in reducing the developing and manufacturing costs and time of chips. It has therefore become feasible to develop firmware by embedding calculations and graphics-intensive algorithms into chips to speed up their executions instead of developing conventional software.

The microcomputer (PC)-based CAD systems have been developing remarkably in the past few years. The conventional problems of memory size, processing speed and memory-accessing speed seem to be going away. Similarly, peripheral storage has been enhanced by developing high-capacity fixed (hard) disks with high access speeds. User-interaction techniques have also been developing rapidly.

CAD/CAM systems based on either the microcomputer or workstation concept represent a distinct philosophy or trend in hardware technology which is based on a distributed (stand-alone) but networked (linked) environment. Workstations can be linked together as well as to mainframes dedicated to numerical computations. Other processors may exist in the network to control other types/ of hardware such as file and print servers. These distributed systems are able to perform major graphics functions locally at the workstations and operations that require more power are sent to the mainframe. The communication between devices in this distributed design and manufacturing environment becomes an important part of the system configuration and design.

The dynamics and rapid changes in the hardware technology have created an absorption problem at the user's part. There are always various types and configurations of CAD/CAM systems to choose from. To choose and implement a system in an industrial or educational environment requires the development of a set of guidelines that must address both hardware and software requirements. Section 2.3 discusses how to choose a system. A key factor in a system evaluation is the capabilities and integration of its software which influence the productivity rate directly.

Managing a CAD/CAM system (as other computer systems) covers both hardware and software. CAD/CAM managers are typically responsible for day-to-day operation and maintenance of the system, developing training programs for users and keeping informed of the latest hardware and software trends. Day-to-day operation involves developing a sign-up scheme because there are never enough terminals for unlimited access, reporting hardware problems and software bugs to the vendor, dealing with users' problems and questions and keeping a record of system utilization. Developing a training program for users is an important function of the manager. Another function of a manager is to keep informed of the latest products and trends. This typically requires attending vendor and/or users' group meetings, as well as trade shows/exhibits and conferences. Thus the manager is always in a position to recommend future upgrades of the existing system.

The main objective of this chapter is to provide generic descriptions of CAD/CAM hardware and its related terminology. We mainly cover the types of architectures of CAD/CAM systems, input devices and output devices.

2.2 TYPES OF SYSTEMS[1]

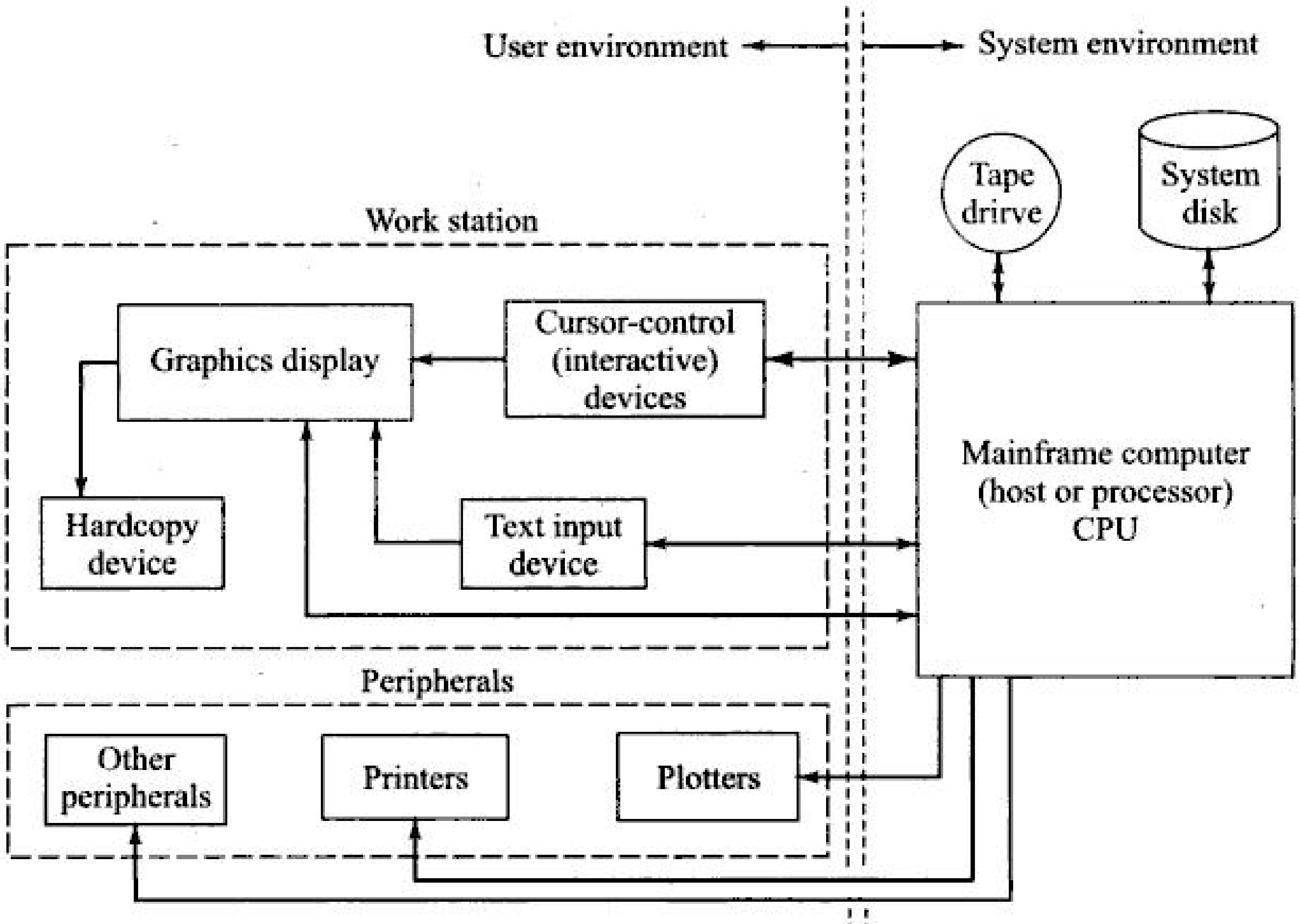
The types of CAD/CAM systems available to end users are quite diverse. Within each type, there exist various configurations and options. A generic classification of these systems and a general description of each type and its features are beneficial to understand hardware trends. The classification criteria are subjective and can vary significantly from one occasion to another. One might classify systems on the basis of their hardware configuration and performance. A second person might choose software capabilities as a criterion or systems may be categorized on the basis of their main geometric modeling technique, whether it is wireframe, surface, or solid modeling.

In this section, CAD/CAM systems are classified based on their hardware. More specifically, the type of host computer that drives the system is the major factor in the classification. While various configurations and peripherals may exist within each type, this is less important here. Some of the types described below are of less use than others due to the rapid changes in the hardware technology or due to the newness of the hardware concept itself. Nevertheless, it is interesting and informative to be aware of the development of this technology.

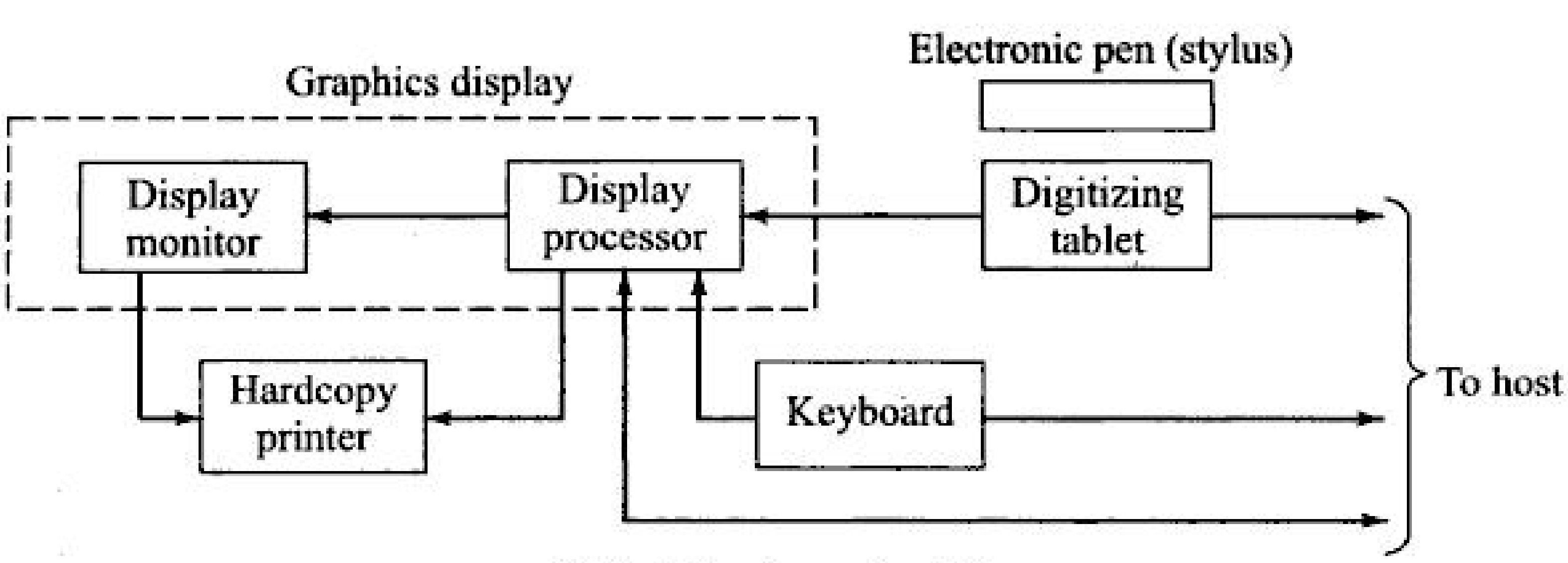
[1] This section includes excerpts from the Frost and Sullivan report 1564.

2.2.1 Mainframe-Based Systems

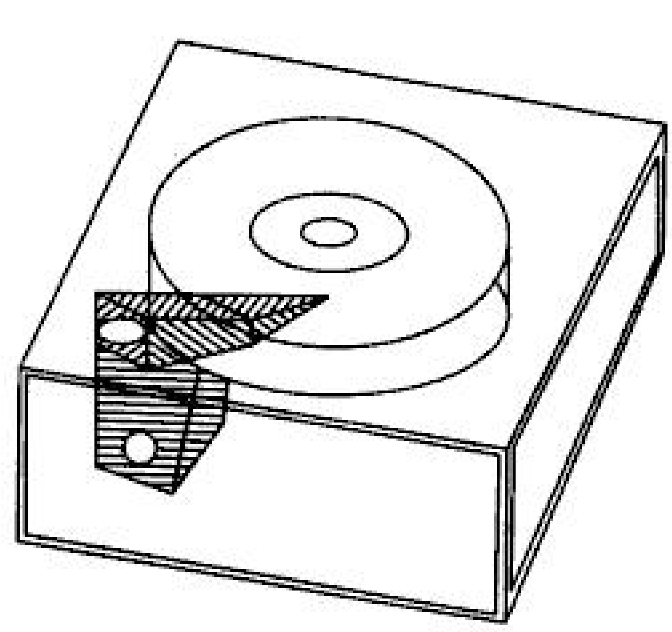
At one point all CAD/CAM systems had been mainframe-based since that was the only type of computer available. A typical mainframe-based CAD/CAM system



(*a*) Overall view of the system



(*b*) Details of a work station



(*c*) Internal Hard Disk

Fig. 2.1 *Schematic Diagram of a Typical Mainframe-based CAD/CAM System*

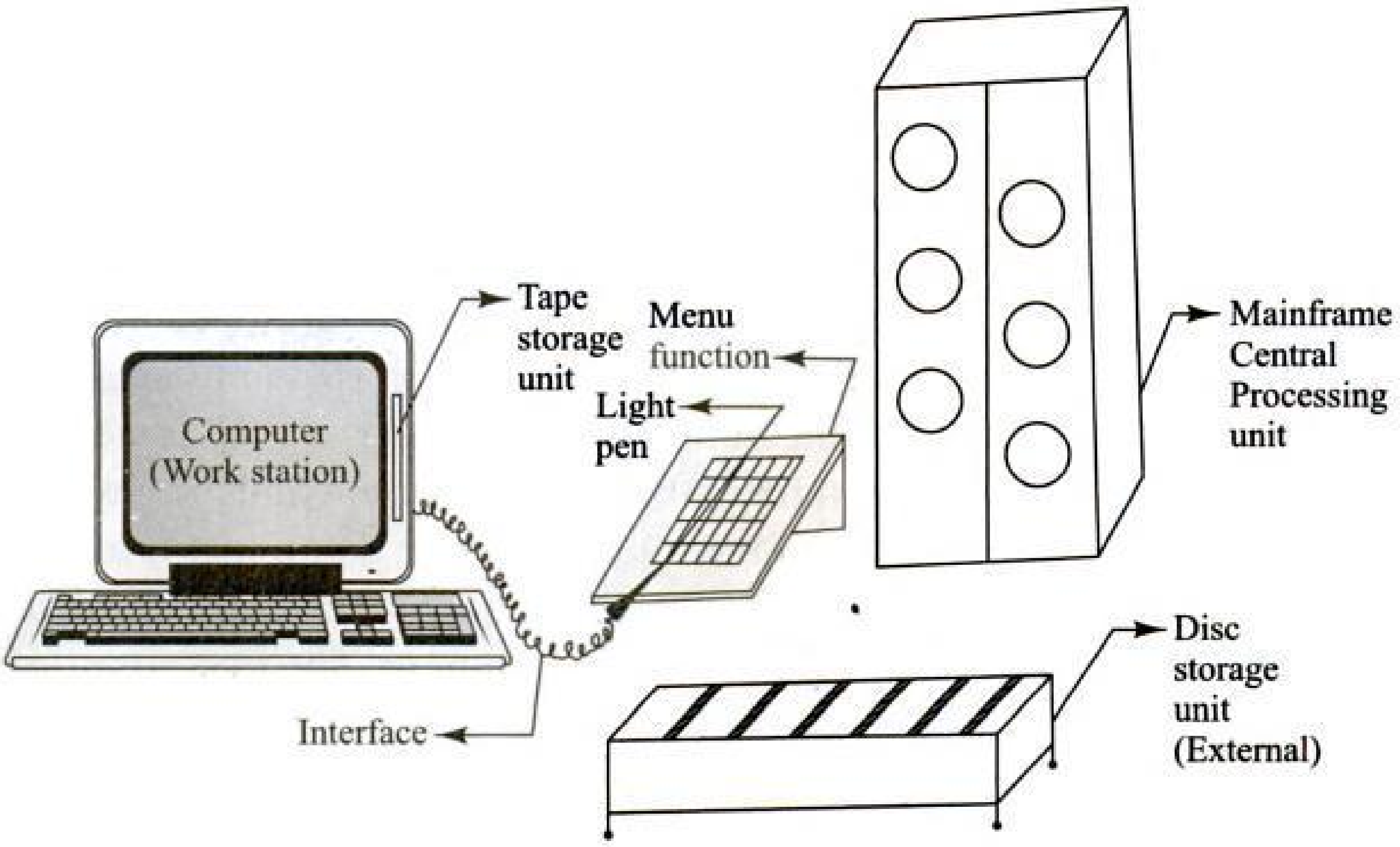


Fig. 2.2 *Mainframe-based CAD/CAM System. (Courtesy of McDonnell Douglas Information Systems Group.)*

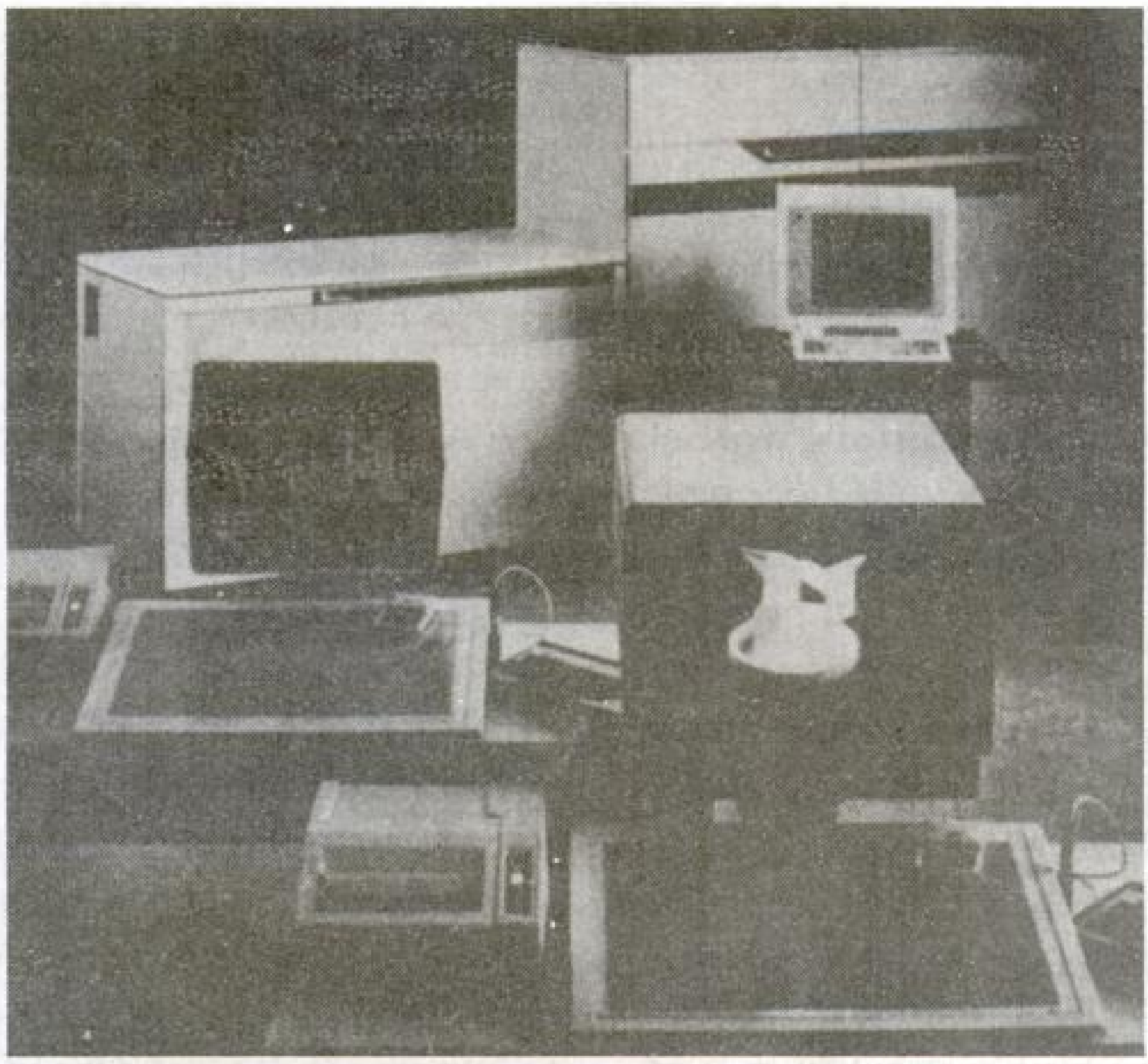


Fig. 2.3 *Mainframe-based CAD/CAM System. (Courtesy of Computervision Corp.)*

2.2.2 Minicomputer-Based Systems

The development of LSI (Large Scale Integrated) circuits and now VLSI, has changed the basic principles of computer architecture and has directly led to the

2.2.4 Workstation-Based Systems

The workstation concept underlying these systems has emerged from the simple downward evolution of well-established systems technology into a single-user or an office environment. For CAD/CAM applications, the concept offers significant advantages over the timesharing, central computing facility accessed through graphic display terminals offered by mainframe- or supermini-based systems. Among these advantages offered by workstations are their availability, portability, the ability to dedicate them to a single task without affecting other users and their consistency of time response. By supplying engineering professionals with their own dedicated computing resources, workstations have proven their effectiveness in shortening time to complete typical engineering tasks. Studies have shown that a user's train of thought is broken and productivity suffers when the system's response to a user's command is not within one second. More specifically, an IBM study of system response time showed that each one-tenth-second decrease in this time led to a four-minute reduction in time required to complete the experimental design task.

Graphics terminals attached to mainframes, minis, or PCs do not qualify as workstations. These terminals may be referred to as "work stations" (two words). A workstation (sometimes also referred to as a personal, technical, or engineering workstation) can be defined as a "work station" with its own computing power to support major software packages, multitasking capabilities demanded by increased usage and complex tasks and networking potential with other computing environments. A set of standards have been adopted since 1981 to guide the development of workstations and to differentiate them from low-cost 16-bit PCs and graphics terminals. The most important of these standards are the 32-bit architecture, Unix operating system and Ethernet local area network. However, the rapid development of graphics terminals, PCs and workstations is making the dichotomy between them difficult to sustain. One might look at the workstation concept as a trend that is placing downward pressure on the PCs and upward pressure on the high-end graphics terminals, obviously suggesting the diminishment of both.

The workstation concept seems to form the basis of the next generation of CAD/CAM systems. The style of distributed computing offered by CAD/CAM workstations is preferred over that of central computing offered by mainframe- or superminicomputer-based CAD/CAM systems. The argument for such preference is typically based on the inadequate performance of multiuser environments provided by the latter systems, the networking potential of workstations and low entry costs. Figures 2.8 and 2.9 show examples of various workstations that are currently available.

on the cost of a system in addition to the initial capital investment. Various vendors offer various types of contracts ranging from an immediate service to an on-call basis service.

2.3.1.4 Vendor Support and Service

Vendor support typically includes training, field services and technical support. Most vendors provide training courses, sometimes on-site if necessary. The timely response of the vendor response centers to customer's technical questions is important during the startup time when no in-house technical expertise is available.

2.3.2 Geometric Modeling Capabilities

2.3.2.1 Representation Techniques

The geometric modeling module of a CAD/CAM system is its heart. The applications module of the system is directly related to and limited by the various representations it supports. Wireframes, surfaces and solids are the three types of modeling available. Most commercial CAD/CAM systems provide them. However, it is important to consider the various entities supported by each representation. The integration between the various representations and the applications they support is an essential issue.

2.3.2.2 Coordinate Systems and Inputs

In order to provide the designer with the proper flexibility to generate geometric models, various types of coordi nate systems and coordinate inputs ought to be provided. A working coordinate system is an example of the former. The ability of the designer to define the proper planes of constructions is valuable to create faces that are not parallel to the standard orthogonal planes (front, top, right). Coordinate inputs can take the form of cartesian (x, y, z), cylindrical (r, θ, z) and spherical (θ, ϕ, z).

2.3.2.3 Modeling Entities

The fact that a system supports a representation scheme is not enough. It is important to know the specific entities provided by the scheme. The ease to generate, verify and edit these entities should be considered during evaluation.

2.3.2.4 Geometric Editing and Manipulation

It is essential to ensure that these geometric functions exist for the three types of representations. Editing functions include intersection, trimming and projection and manipulations include translation, rotation, copy, mirror, offset, scaling and changing attributes.

2.3.2.5 Graphics Standards Support

If geometric models' databases are to be transferred from one system to another, both systems must support exchange standards. These standards are valuable if various systems exist in one organization or if design models are shipped to outside vendors for various reasons such as tools, jigs and fixture designs, or generating NC part programs. It should be mentioned here that graphics standards may introduce inconsistencies and errors into CAD/CAM databases.

2.3.3 Design Documentation

2.3.3.1 Generation of Engineering Drawings

After a geometric model is created, standard drafting practices are usually applied to it to generate the engineering drawings or the blueprints. Various views (usually top, front and right side) are generated in the proper drawing layout. Then dimensions are added, hidden lines are eliminated and/or dashed, tolerances are specified, general notes and labels are added, etc. These activities are time-consuming. To generate an engineering drawing, it typically takes as much as two to three times as long as it takes to generate the geometric model.

2.3.4 Applications

2.3.4.1 Assemblies or Model Merging

Generating assemblies and assembly drawings from individual parts is an essential process. The two issues that are worth investigation are the assembly procedure itself and the clean-up of the resulting assembly.

2.3.4.2 Design Applications

There are design packages available to perform applications such as mass property calculations, tolerance analysis, finite element modeling and analysis, injection modeling analysis and mechanism analysis and simulation. What should be evaluated are the capabilities of these packages, their integration and interfaces with geometric databases and the representation techniques they utilize. Some packages might require clumsy user input while others might lack the proper way to display results.

2.3.4.3 Manufacturing Applications

The common packages available are tool path generation and verification, NC part programming, postprocessing, computer aided process planning, group technology, CIM applications and robot simulation. It is essential to ensure that the CAD and CAM applications that are provided by the system are truly integrated.

2.3.4.4 Programming Languages Supported

It is vital to look into the various levels of programming languages a system supports. Attention should be paid to the syntax of graphics commands when they are used inside and outside the programming languages. If this syntax changes significantly between the two cases, user confusion and panic should be expected.

All of the above evaluation criteria are covered in full detail throughout the book. To gain a better understanding of any one of these criteria, the reader is advised to refer to the corresponding part, chapter, or section in the book.

2.4 INPUT DEVICES

The user of a CAD/CAM system spends much time sitting at a workstation communicating and interacting with the computer to develop a particular engineering design. As shown in Fig. 2.1, the user utilizes both input and output devices that comprise a workstation to achieve the design task. These devices are universal and mostly independent of the types of CAD/CAM systems discussed in Sec. 2.2. Software drivers are required to enable the host application program, i.e., the CAD/CAM software, to interpret the information received from input devices as well as send information to output devices. In the past, most of these drivers used to be hardware-dependent. However, greater acceptance of graphics standards, such as GKS, enables programmers to write device-independent codes. A number of input devices are available. These devices are used to input the two possible types of information: text and graphics. Text-input devices are the alphanumeric (character-oriented) keyboards. There are three classes of graphics-input devices: locating devices, digitizers and image-input devices. Locating devices, or locators, provide a position or location on the screen. These include lightpens, mice, digitizing tablets and styli, joysticks, trackballs, thumbwheels, touchscreens and touchpads. The keyboard arrow (cursor direction) keys are inadequate for most graphics applications and therefore are not considered here. Locating devices typically operate by controlling the position of a cursor on the screen. Thus, they are also referred to as cursor-control devices. The popularity of window and icon-oriented user interfaces has stimulated the further development and enhancement of locating devices. Normally, locators do not only have positioning functionality; they also provide other functions and graphical input modes such as picking and choosing, tracing and sketching. Picking or pointing means selecting a displayed item or entity on the screen. A drawing can be input to the computer by carefully tracing over its graphics entities. Line segments can be traced by simply digitizing their endpoints. Sketching, sometimes referred to as painting, involves the freehand generation of a drawing. All locating devices, with the exception of the joystick which may provide three-dimensional input, are two-dimensional input devices. Three-axis input devices have been demonstrated using acoustic and mechanical techniques. However, they remain expensive and are not employed in CAD/CAM systems.

Another class of graphics input devices, besides locating devices, is digitizer boards or tablets, or simply digitizers. They are considered as electronic drafting boards. A digitizer consists of a large synthesized electronic board with a movable stylus called the cursor. It is a two-dimensional input device with high resolution and accuracy. Typical sizes are 36×48 and 48×72 inches. Available resolution and accuracy are up to 0.001 and 0.003 inch respectively. Digitizers can be divided into three kinds relative to the mode of operation of the cursor. They are free-cursor, constrained-cursor and motor-cursor digitizers. In the first kind, the cursor is attached to the end of a flexible chord, in the second it slides along a gantry that traverses the entire digitizing board area and in the third kind the cursor motion is accomplished by motors driven by an operator-controlled joystick. Each kind has advantages and disadvantages. The first kind provides greater ease of moving the cursor. The cursor is restricted in the second kind but the digitizer can be used in an upright position. Motorized digitizers are expensive but combine the best features of the first two.

Image-input devices such as video frame grabbers and scanners comprise the third class of graphics-input devices. Electronic imaging is an area of relevance to image processing. This area may become significant to the CAD/CAM field if robot vision systems are to be driven by CAD/CAM databases. Video digitizers is

Fig. 2.12 *A Dual Screen CAD/CAM System. (Courtesy of GE (U.S.A.) Calma, Inc.)*

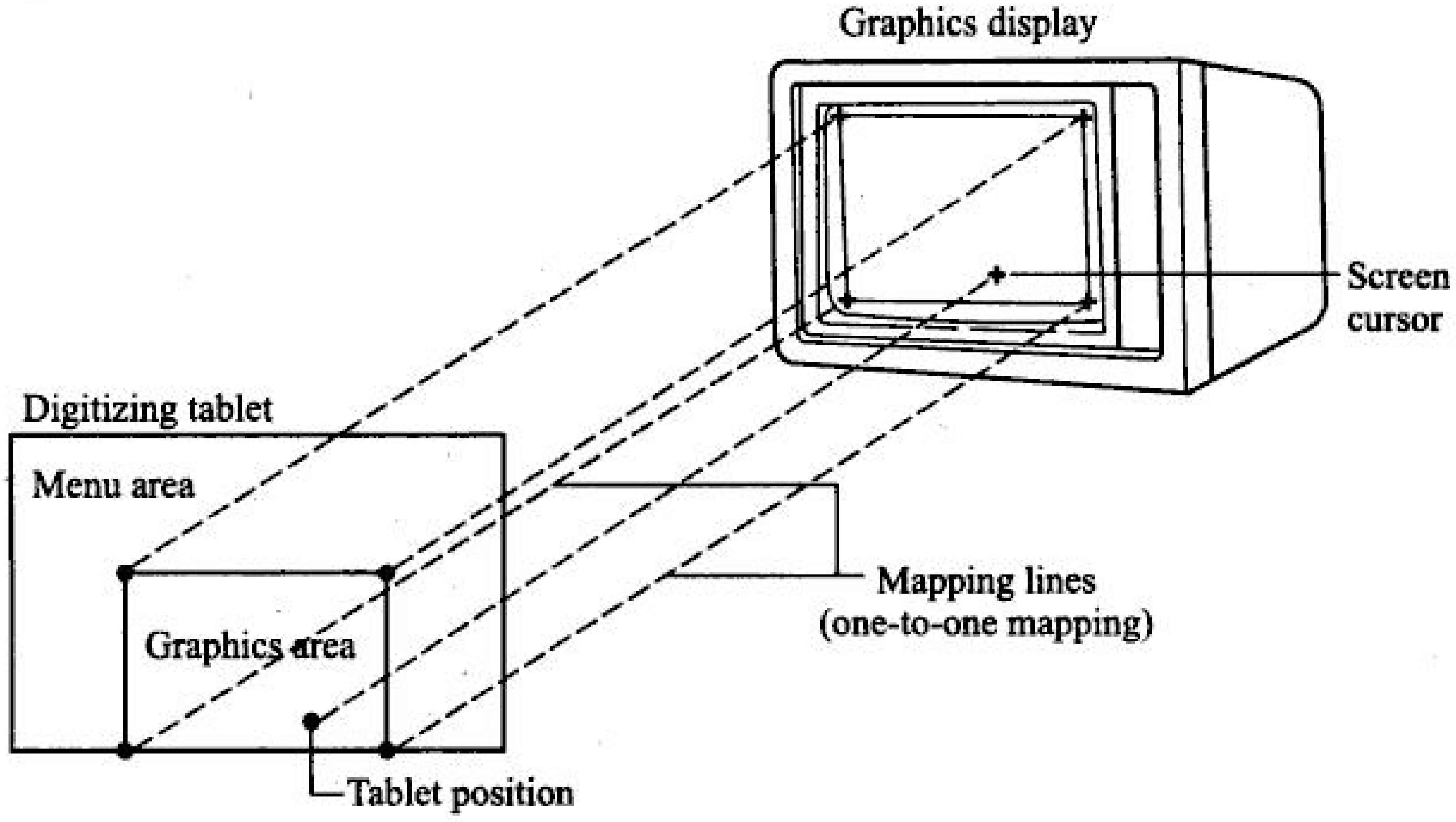
should be warned that these files are usually hard to read, especially when functions such as backspaces, control keys, or digitizes are encountered in the files.

How does the keyboard communicate with the CAD/CAM software or the main application program? How is the software interrupted to receive the keyboard input? Each keyboard or input device, in general, is connected to the computer by means of registers whose contents can be read by the computer. A keyboard has typically two registers, one to set a status bit when a key has been struck, the other to identify the key by its character code. The value of the status bit is monitored in a continuous repetitive manner by the software via a programming technique known as "polling." When the user hits a key, the status bit is set and the application program is consequently interrupted to clear the status bit, followed by reading the corresponding code of the key character. The loop is repeated every time the user strikes a key. Keyboard characters are identified by their ASCII (American Standard Code for Information Interchange) codes. ASCII codes for alphanumeric characters are 7-bit codes. Therefore, a character register has seven bits. The character code for a capital letter, say A, is different from that for a small letter, say a. ASCII codes for A and a are 1000001 and 1100001 respectively, the difference being the replacement of the first zero from the left by one which is the case for all the other characters. This is why some programs may require text input as either lower case or upper case. EBCDIC (Extended Binary Coded Decimal Interchange Codes) for alphanumeric characters are 8-bit codes and do not have this distinction (11000001 is the code for A); that is, capital letters are always used.

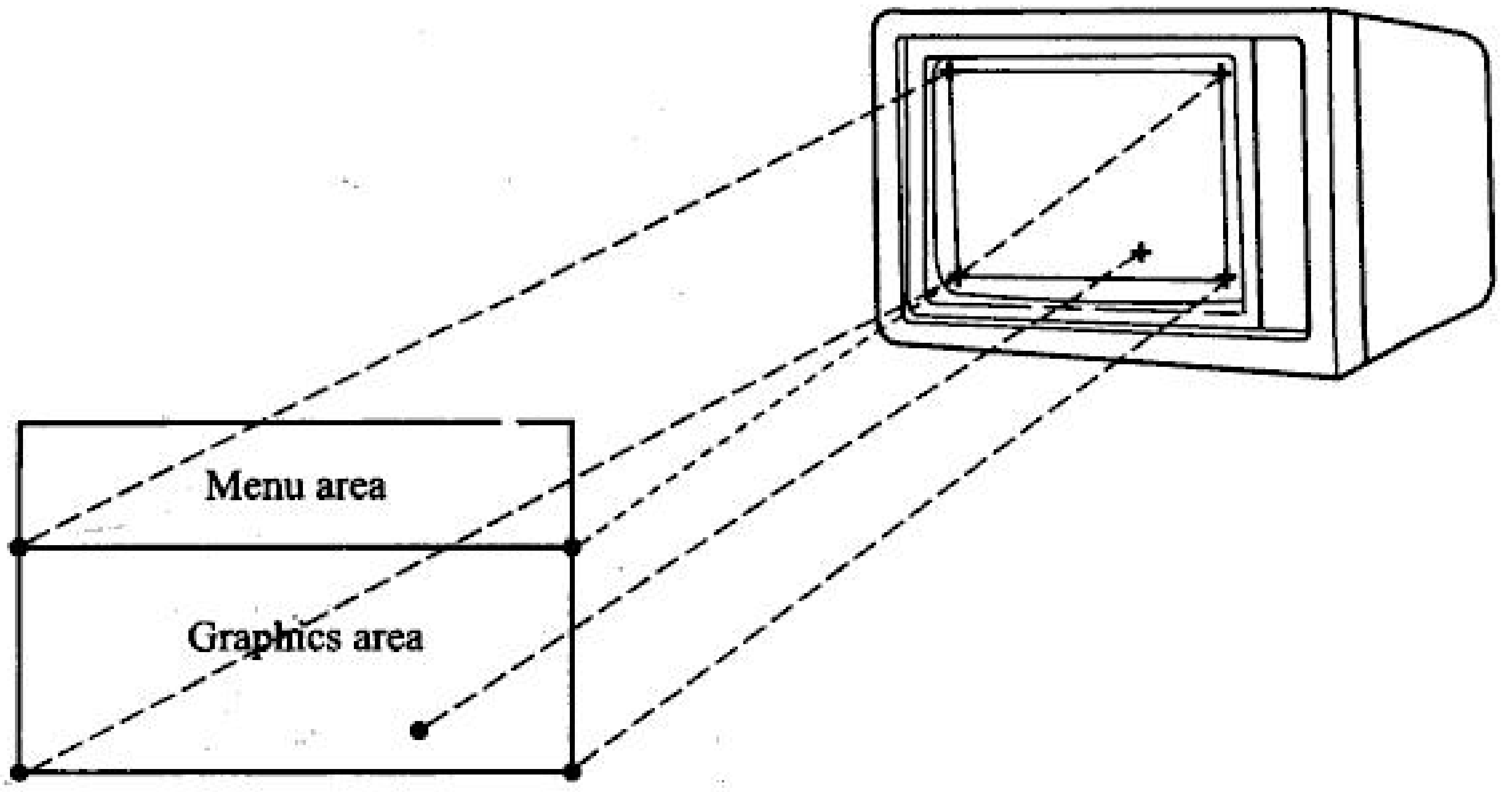
2.4.2 Lightpens

The lightpen is intrinsically a pointing or picking device that enables the user to select a displayed graphics item on a screen by directly touching its surface in the vicinity of the item. The application program processes the information generated from the touching to identify the selectable item to operate on. The lightpen, however, does not typically have hardware for tracking, positioning, or locating in comparison to a digitizing tablet and stylus. Instead, these functions are performed by utilizing the hardware capabilities of the graphics display at hand. The lightpen itself does not emit light but rather detects it from graphics items displayed on the screen.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



(*a*) Organization A (used by Computervision)



(*b*) Organization B (used by GE (U.S.A) Calma)

Fig. 2.16 *Mapping between a Tablet Graphics Area and a Display Screen*

2.4.4 Mouse Systems

The mouse was invented in the late 1960s as a location device but has only recently become fairly popular due to its convenient use with icons and pop-up and pull-down menus. Unlike the digitizing tablet, the mouse measures its relative movement from its last position, rather than where it is in relation to some fixed surface. There are two basic types of mice available: mechanical and optical. The mechanical mouse is a box with two metal wheels or rollers on the bottom whose axes are orthogonal in order to record the mouse motion in the X and Y directions. The roll of the mouse on any flat surface causes the rotation of the wheel which is encoded into digital values via potentiometers. These values may be stored, when a mouse pushbutton is depressed, in the mouse registers accessible by the application program either immediately or during the computer interrupt every refresh cycle. Using these values, the program can determine the direction and magnitude of the mouse

Table 2.2 *Comparison of input devices*

	Device						
Function	***Lightpen***	***Tablet/stylus***	***Tablet/puck***	***Mouse***	***Joystick***	***Trackball***	***Touchscreen***
Resolution	Low	High	High	Medium	Medium	Medium	Low
Response to hand movement (positioning speed	High	High	High	High	High	High	High
Locating (positioning)	Poor	Very good	Very good	Very good	Good	Good	—
Picking (pointing)	Good	Very good	Very good	Very good	Good	Good	—
Digitizing	—	Good	Very good	—	—	—	—
Traking	—	Very good	Good	—	—	—	—
Sketching	—	Very good	Good	Good	Poor	Poor	Fair
Fatigue	High	Low	Low	Medium	Low	Low	Low

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

DVST was introduced by Tektronix as an alternative and inexpensive solution. It is believed that the emergence of the DVST in that time had a significant impact on making CAD/CAM systems affordable for both users and programmers. The DVST eliminates the refresh processors completely and, consequently, the refresh buffer used with the refresh display, as shown in Fig. 2.24. It also uses a special type of phosphor that has a long-lasting glowing effect. The phosphor is embedded in a storage tube. In addition, the speed of the electron beam in the DVST is slower than in the refresh display due to elimination of the refresh cycle.

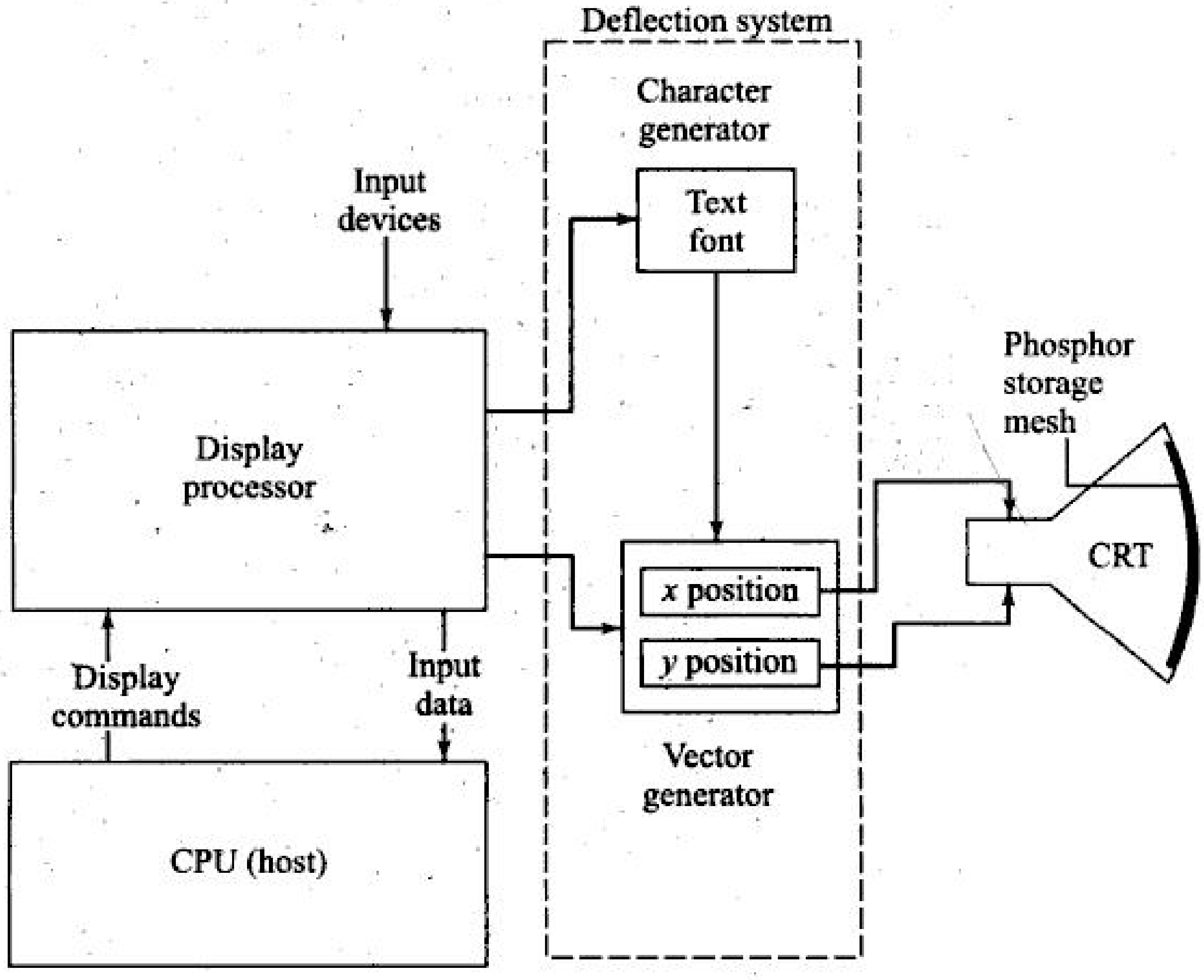


Fig. 2.24 *Direct view storage tube*

In the DVST, the picture is stored as a charge in the phosphor mesh located behind the screen's surface. Therefore, complex pictures could be drawn without flicker at high resolution. Once displayed, the picture remains on the screen until it is explicitly erased. This is why the name "storage tube" was suggested. New picture items can be added and displayed rapidly. However, if a displayed item is erased, the entire screen must be cleared and the new picture displayed (typically by using a "repaint" command) to reflect the removal of the item.

In addition to the lack of selective erasure, the DVST cannot provide colors, animation and use of a lightpen as an input device. Due to its main advantages of inexpensive price and high resolution, early turnkey CAD/CAM systems used storage tubes for their displays. DVST can now be found with a local intelligence and display file to provide selective erasure without the need to refresh the picture.

2.5.1.3 Raster Display

The inability of the DVST to meet the increasing demands by various CAD/CAM applications for colors, shaded images and animation motivated hardware designers to continue searching for a solution. During the late 1970s raster displays based on the standard television technology began to emerge as a viable alternative. The drop in memory price due to advances in solid states made large enough refresh buffers available to support high-resolution displays. A typical resolution of a raster display is 1280 × 1204 with a possibility to reach 4096 × 4096 as the DVST. Raster displays are very popular and nearly all recent display research and development focus on them.

In raster displays, the display screen area is divided horizontally and vertically into a matrix of small elements called picture elements or pixels, as shown in Fig. 2.25. A pixel is the smallest addressable area on a screen. An $N \times M$ resolution defines a screen with N rows and M columns. Each row defines a scan line. A rasterization process is needed in order to display either a shaded area or graphics entities. In this process, the area or entities are converted into their corresponding pixels whose intensity and color are controlled by the image display system.

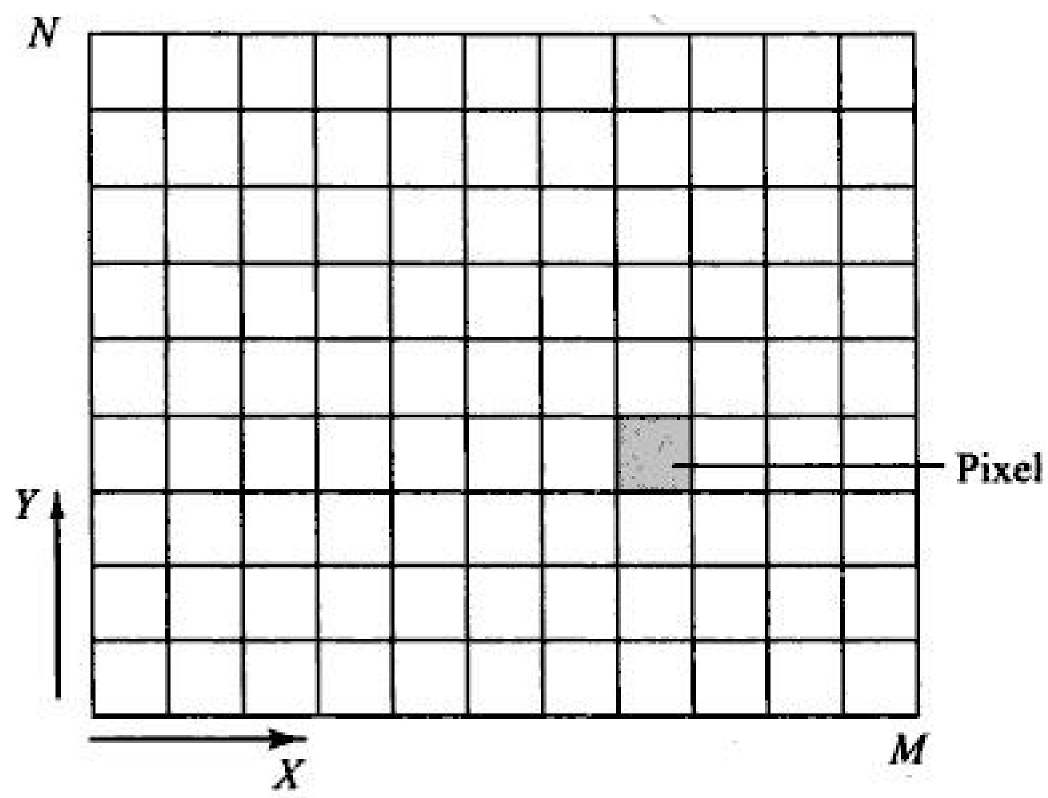


Fig. 2.25 *Typical Pixel Matrix of a Raster Display*

Figure 2.26 shows a schematic of a typical color raster display. Images (shaded areas or graphics entities) are displayed by converting geometric information into pixel values which are then converted into electron beam deflection through the display processor and the deflection system shown in the figure. If the display is monochrome, the pixel value is used to control the intensity level or the gray level on the screen. For color displays, the value is used to control the color by mapping it into a color map.

The creation of raster-format data from geometric information is known as scan conversion or rasterization. A rasterizer that forms the image-creation system shown in Fig. 2.26 is mainly a set of scan-conversion algorithms. Due to the universal need for these algorithms, the scan conversion or rasterization process is now hardware implemented and is done locally in the workstation. As an example, there are standard algorithms such as the DDA (Digital Differential Analyzer) and Bresenham's method which are used to draw a line by generating pixels to approximate the line. Similar algorithms exist to draw arcs, text and surfaces. This is why it is possible to create images with different colors and hollow areas on raster displays.

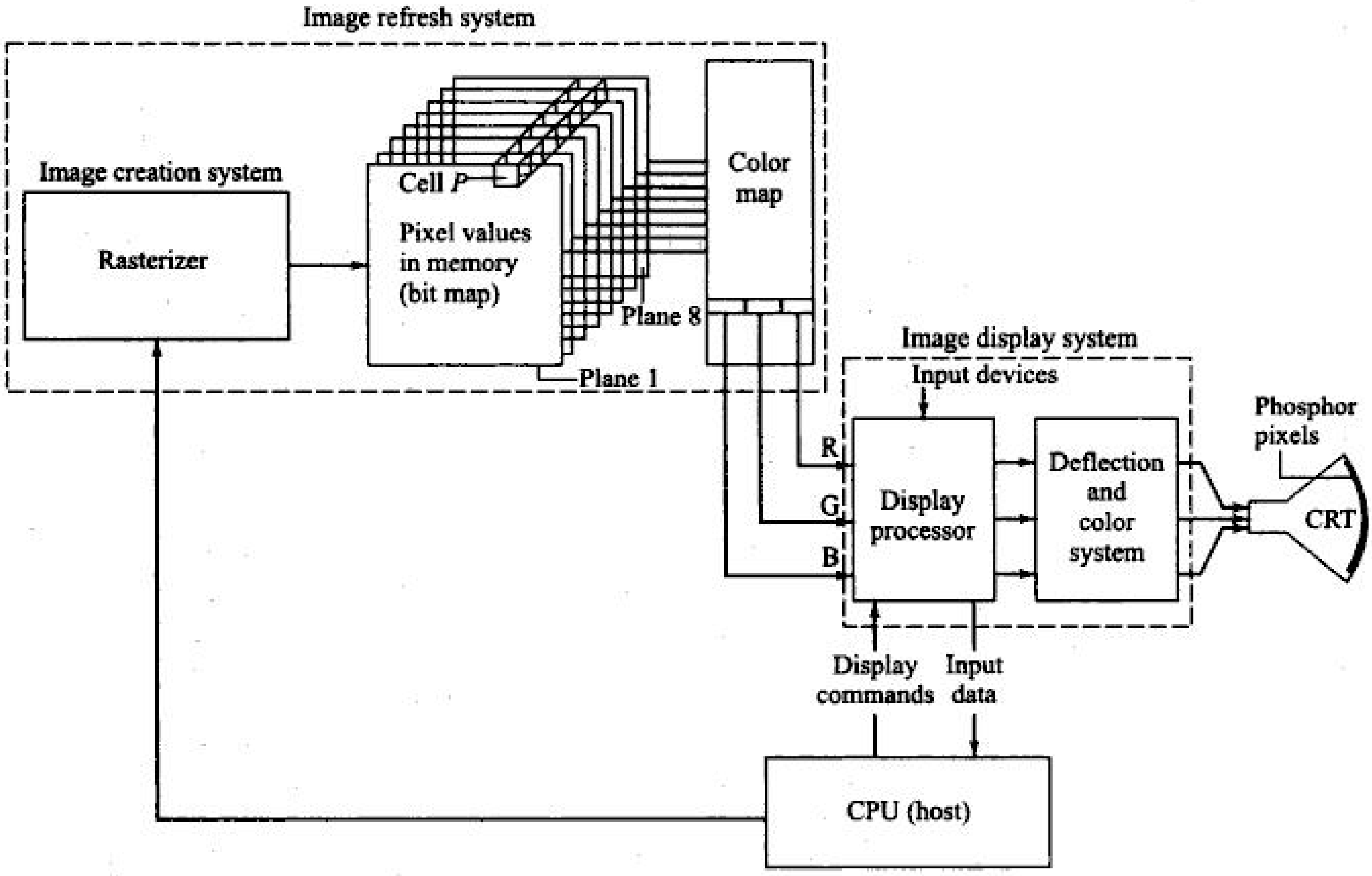
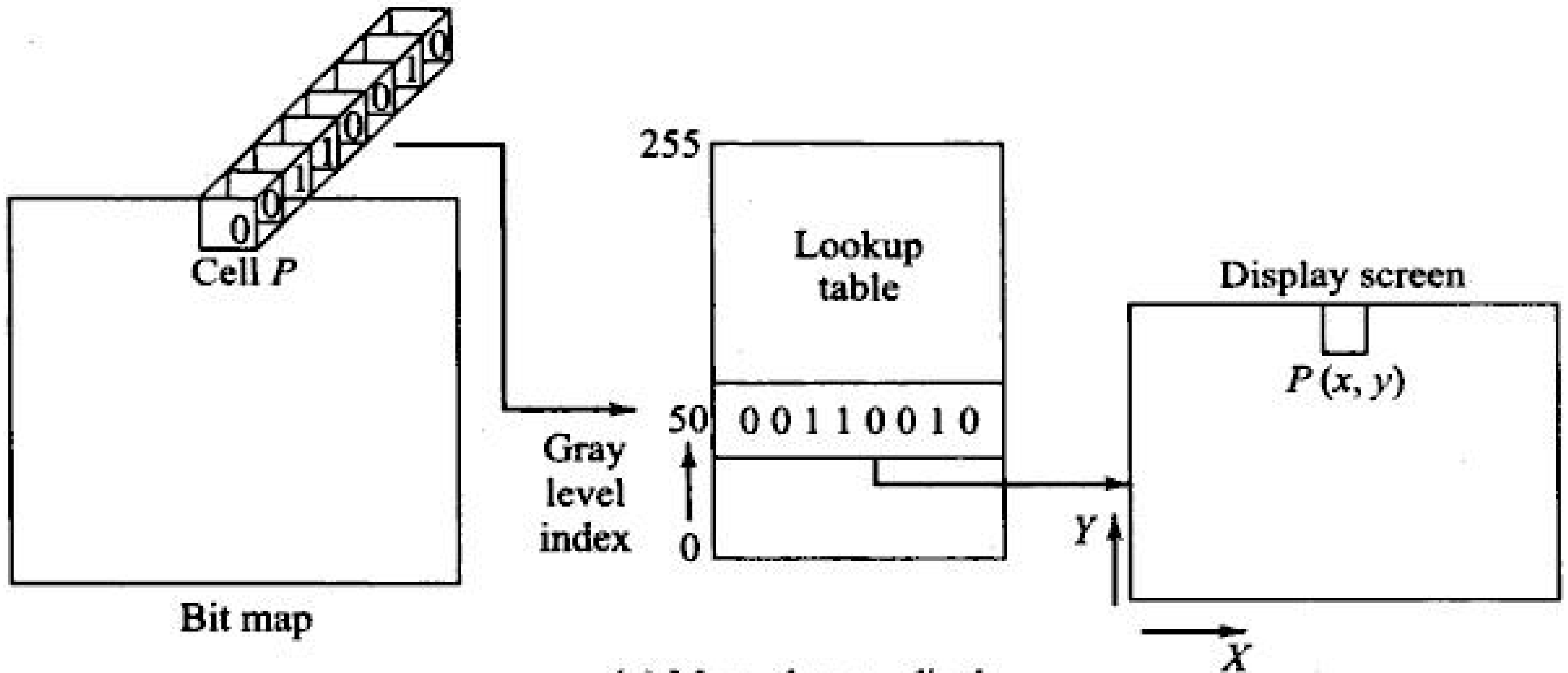


Fig. 2.26 *Color Raster Display with Eight Planes*

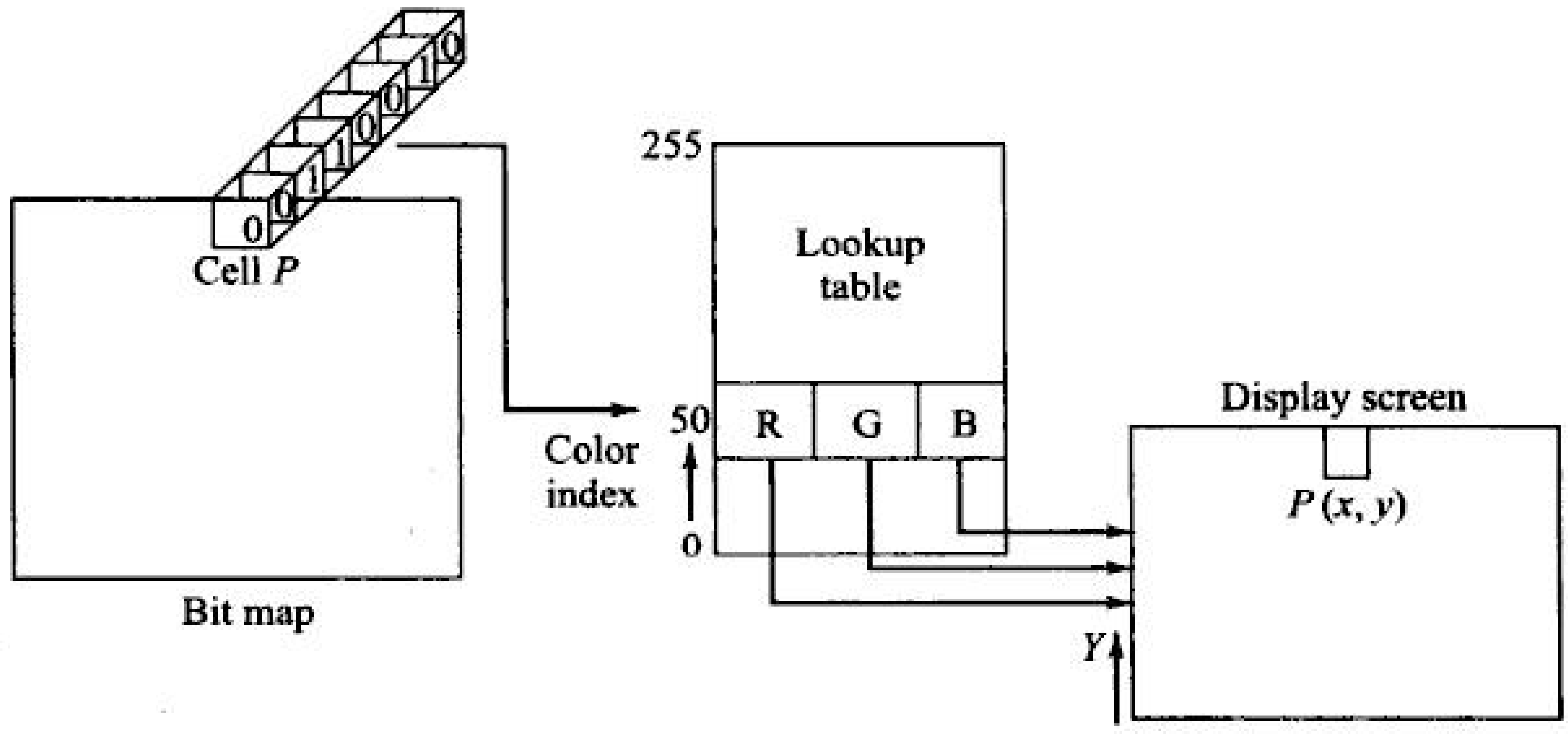
The values of the pixels of a display screen that result from the scan-conversion process are stored in an area or memory called frame buffer or bit map refresh buffer (bit map, for short), as shown if Fig. 2.26. Each pixel value determines its brightness (gray level) or most often its color on the screen. There is a one-to-one correspondence between every cell in the bit map memory and every pixel on the screen. The display processor maps every cell into its corresponding screen pixel brightness or color. In order to maintain a flicker-free image on the screen, the screen must be refreshed at the rate of 30 or 60 Hz, as in the case of refresh displays. The refresh process is performed by passing the pixel values in the bit map to the display processor every refresh cycle regardless of whether these values represent the image or the background. Therefore the refresh process is independent of the complexity of the image and the number of its graphics items. Thus there is no chance of a flicker problem with the increased complexity of the image as in the case of refresh displays.

To understand the performance of raster displays and to evaluate them, one must ask the following question: how many bits are required in the bit map to adequately represent the intensity of any one pixel on the display screen? The trivial answer of one bit/pixel produces only a two-level image (bright or dark) which is very unsatisfactory to basic applications. The practice suggests that 8 bits/pixel are needed to produce satisfactory continuous shades of gray for monochrome displays. For color displays, 24 bits/pixel would be needed: 8 bits for each primary color red, blue and green. This would provide 2^{24} different colors, which are far more than needed in real applications. Typically, 4 to 8 bits/pixel are adequate for both monochrome and color displays utilized in most engineering applications. Specialized image processing applications may require more than that. The bit map memory is arranged conceptually as a series of planes, one for each bit in the pixel value. Thus an eight-plane memory provides 8 bits/pixel, as shown in Fig. 2.26. This provides 2^8 different gray levels or different colors that can be displayed simultaneously in one image. The number of bits per pixel directly affects the quality of its display and consequently its price.

The value of a pixel in the bit map memory is translated to a gray level or a color through a lookup table (also called a color table or color map for a color display). The pixel value is used as an index for this lookup table to find the corresponding table entry value which is then used by the display system (display processor and beam deflection system) to control the gray level or color. Figure 2.27 shows how the pixel value is related to the lookup table in an eight-plane display. If cell P in the bit map corresponds to pixel P at the location $P(x, y)$ on the screen, then the gray level of this pixel is 50 (00110010) or its corresponding color is 50.



(*a*) Monochrome display



(*b*) Color display

Fig. 2.27 *Relationship between pixel value and a lookup table*

Figure 2.27 shows raster displays with what is called direct-definition systems in which the lookup table always has as many entries as there are pixel values in the bit map. For the eight-plane display shown in the figure, the lookup table has 256 (2^8) entries, which correspond to all possible values a pixel may have. For color displays this may imply that the number of bits per pixel must be increased to increase the number of entries in the color map and therefore increase the available number of colors to the user. This, however, is not true and leads to increasing the size of the bit map memory and the cost of the display. Thus, how can the number of color indexes in the color map increase while keeping the pixel definition (number of bits per pixel) in the bit map to a minimum? For example, how can a display have 4 bits/pixel with 24 bits of color output (2^{24} different colors)? This is achieved by designing a color map with 2^{24} (16.7 million) available color indexes. The 4 bits/pixel provides 16 (2^4) simultaneous colors, in an image, which can be chosen from the color map. A pixel value (0 to 15) can be used to set the value of the color index which corresponds to the proper color to be displayed. This scheme, in this example, provides 16 simultaneous colors from a palette of 16.7 million. To the user, the color map is made available where colors are chosen and the application program relates the chosen color to the proper pixel value. For example, if the user chooses the color purple for an image element, the corresponding program sets the corresponding pixels to reflect the color purple.

While raster displays are now a standard offering from nearly all CAD/CAM vendors, the quality of the displayed images is affected by flicker and aliasing problems. The flicker of an image is reduced by simply reducing the time of the refresh cycle. The image refresh system (Fig. 2.26) may use an interlaced scan of two fields. In the interlaced scan (as in the home television), the refresh cycle of $\frac{1}{30}$ second is divided into two subcycles each lasting $\frac{1}{60}$ second. The first subcycle displays the odd-numbered scan lines and the second displays the even-numbered scan lines. This technique produces an image with almost a refresh rate of 60 Hz instead of 30 Hz. The interlaced scan scheme does not work very well if the adjacent scan lines do not display similar information. Another scheme is to use a noninterlaced scan of one field by operating at a higher refresh rate such as 60 Hz. In this scheme, the entire scan lines are refreshed once every $\frac{1}{60}$ second. This high rate means more and faster accesses to the bit map (refresh buffer) per second and higher bandwidth deflection amplifiers used in the deflection system shown in Fig. 2.26.

The aliasing problem is directly related to the resolution of the display which determines how good or bad is the raster approximation of geometric information. The jaggedness of lines at angles other than multiples of 45 degrees, assuming square pixels, is the feature of a raster display known as aliasing. Various methods of antialiasing exist which use various intensity levels to soften the edges of the lines or shades. Of course, the aliasing problem diminishes as the resolution increases and is only related to the screen image and not to the geometric representation in the computer or to drawings plotted on paper.

Example 2.1 An eight-plane raster display has a resolution of 1280 horizontal × 1024 vertical and a refresh rate of 60 Hz noninterlaced. Find:

(*a*) The RAM size of the bit map (refresh buffer).
(*b*) The time required to display a scan line and a pixel.
(*c*) The active display area of the screen if the resolution is 78 pixels (dots) per inch.
(*d*) The optimal design if the bit map size is to be reduced by half.

Solution

(*a*) The RAM size of the bit map $= 8 \times 1280 \times 1024 = 1.3$ Mbytes.
(*b*) The times to display a scan line t_s and a pixel t_p are given by

2.5.2.2 Electrostatic Plotters

They are considered dot matrix or raster plotters. The image in vector form, as lines, arcs, characters and symbols, has to be converted into raster form and sorted. Then these rows of dots can be printed across the width of the paper or plastic film as it slowly moves through the plotter. Typically the plotter resolution is 200 dots/inch or more and each dot is arranged to overlap adjacent ones. This provides a relatively high quality image. Electrostatic plotters have the virtue of being quiet, usually trouble free, undemanding of the operator's time and about an order of magnitude faster than pen plotters. However, they are also an order of magnitude more expensive than pen plotters. It is important to know where the vector-to-raster conversion and sorting is done due to the time they take. If they are performed at the central or the host computer, then the response time at workstations is expected to degrade. Separate processors can be provided specifically for this conversion task. Electrostatic plotters are normally monochrome, that is, black images on a white, translucent, or transparent medium. Color plotters are available but are quite expensive.

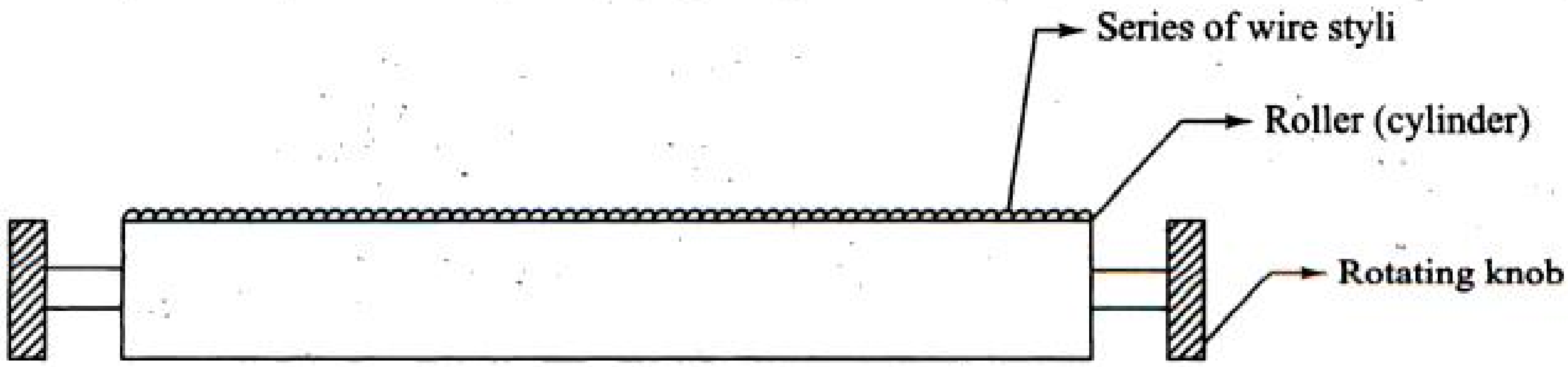


Fig. 2.29 *Electrostatic Plotter*

2.5.2.3 Hardcopy Units

Each workstation in a CAD/CAM system should have its own hardcopy unit to produce quick low-quality copies of screen images (screen dump) and to provide a copy of user input and system output information which is always useful in tracking errors and mistakes. These units are particularly useful when a central plotter is located some distance from workstations. Hardcopy units (Fig. 2.30) can take the form of small electrostatic plotters with a relatively coarse raster grid, impact dot matrix printers using normal paper, or other devices that use light-sensitive paper. A drawback with such paper is that it darkens with time.

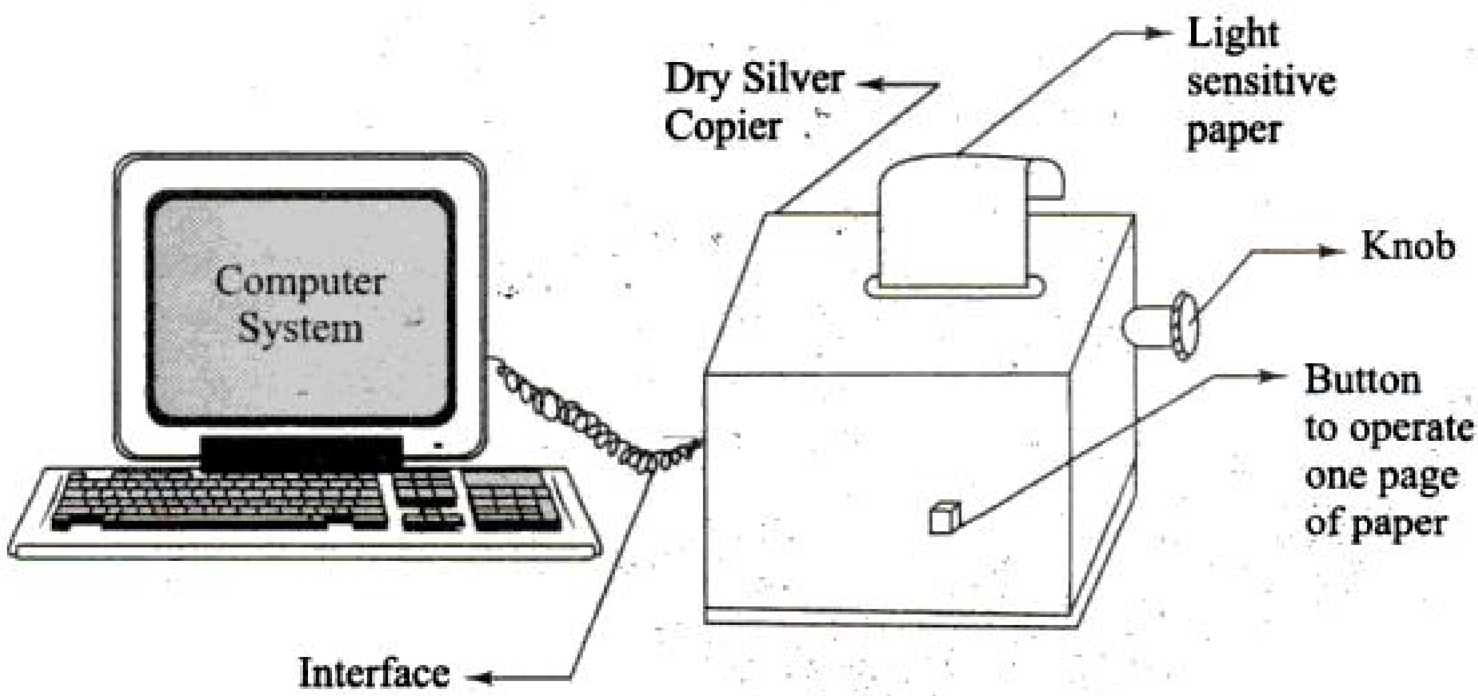


Fig. 2.30 *Hard Copy Unit*

2.5.2.7 Computer Output-to-Microfilm (COM)Unit

These electronic devices do not produce drawings on large drawing sheets. Instead, these units reproduce them on a thin microfilm. They are very expensive and hence not widely used. There are several advantages of these types of devices.

- They have got high storage capacity. A firm might deal with many complicated assembly drawings and hence storage of all these drawings would be very difficult. Reducing the size of each drawing into a small microfilm will indeed lead to significant storage benefit. The data stored in the film can be easily retrieved whenever necessary. If a full size drawing is required, the data can be photographically enlarged.
- Faster retrieval of data can be achieved. These devices produce a microfilm copy much faster than drum type plotters, pedestal plotters, electrostatic plotters and line printers.

In spite of several advantages, there are some limitations too.

- They are very expensive
- It makes use of highly sophisticated technology
- Writing notes on the film during the design and drafting process is not possible as is done in the case of hardcopy unit.
- If the data are phtographically enlarged, the output drawing will lose its clarity and accuracy at high magnification. Hence the drawings produced out of these devices are much inferior to drawings produced out of pen plotters.

2.6 HARDWARE INTEGRATION AND NETWORKING

As discussed in the previous section, there exists various CAD/CAM systems and input and output devices. The integration and networking between the various components and peripherals of a system ensures the success of CAD/CAM installations. The need for hardware integration and networking in CAD/CAM are manifold. CAD/CAM is interdisciplinary by nature and therefore its functions are distributed among various departments, such as design and manufacturing, in many organizations. The hardware components in these departments must communicate together and have access to common databases. Another need for networking is to share common resources and peripherals such as plotters and printers. Stand-alone workstations are most often networked together and to central computing facilities. The need to expand a CAD/CAM system by adding new workstations in an incremental fashion necessitates networking. It is also common to have a need to network devices that complement each other. An example is connecting a high-end CAD/CAM system to a low-end system to allow database transfer.

Local area networks (LANs) are the main communication technology available. A LAN is a data communication system that allows various types of digital devices to talk to each other over a common transmission medium. Low-cost and low-

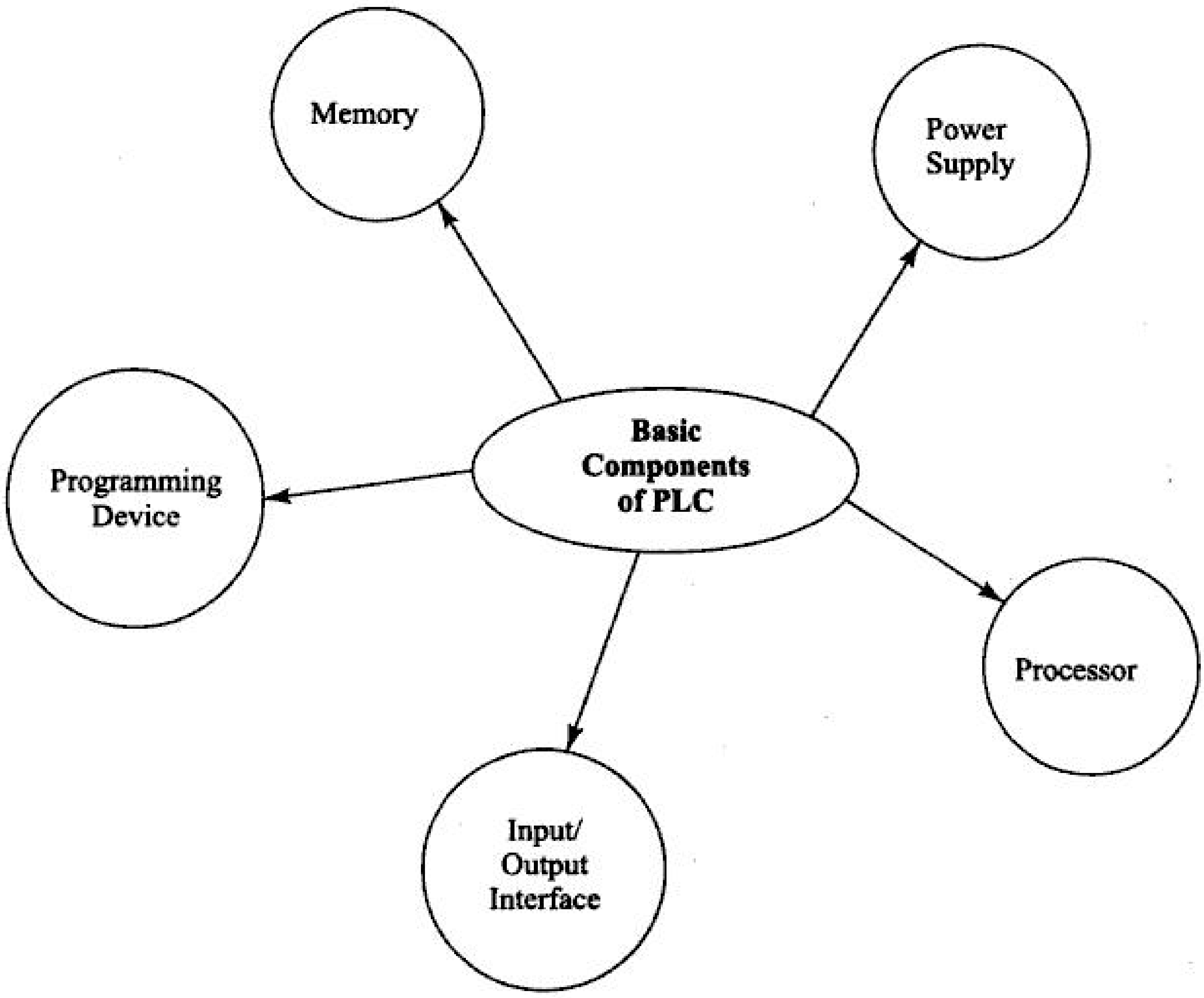


Fig. 2.34

PLCs are designed to be connected to industrial equipment. the input interface will receive process and machine signals and convert them into a form acceptable and understandable by PLC. The output interface converts PLC control signals into a form that is acceptable and understandable by processing equipment.

2.7.2 Hardware Configuration of PLC

The five elements of PLC are connected as indicated in Fig. 2.35 (a).

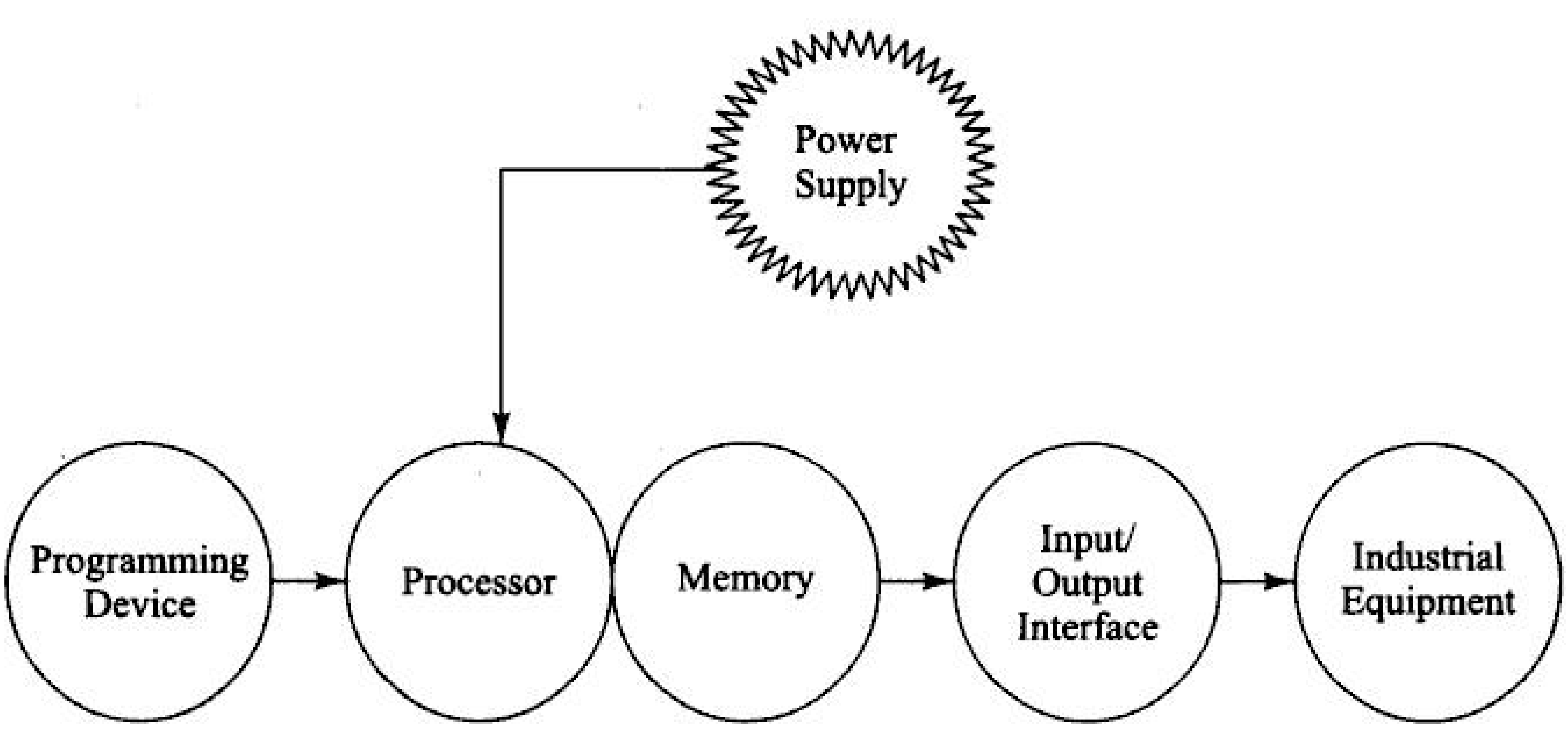


Fig. 2.35 (a)

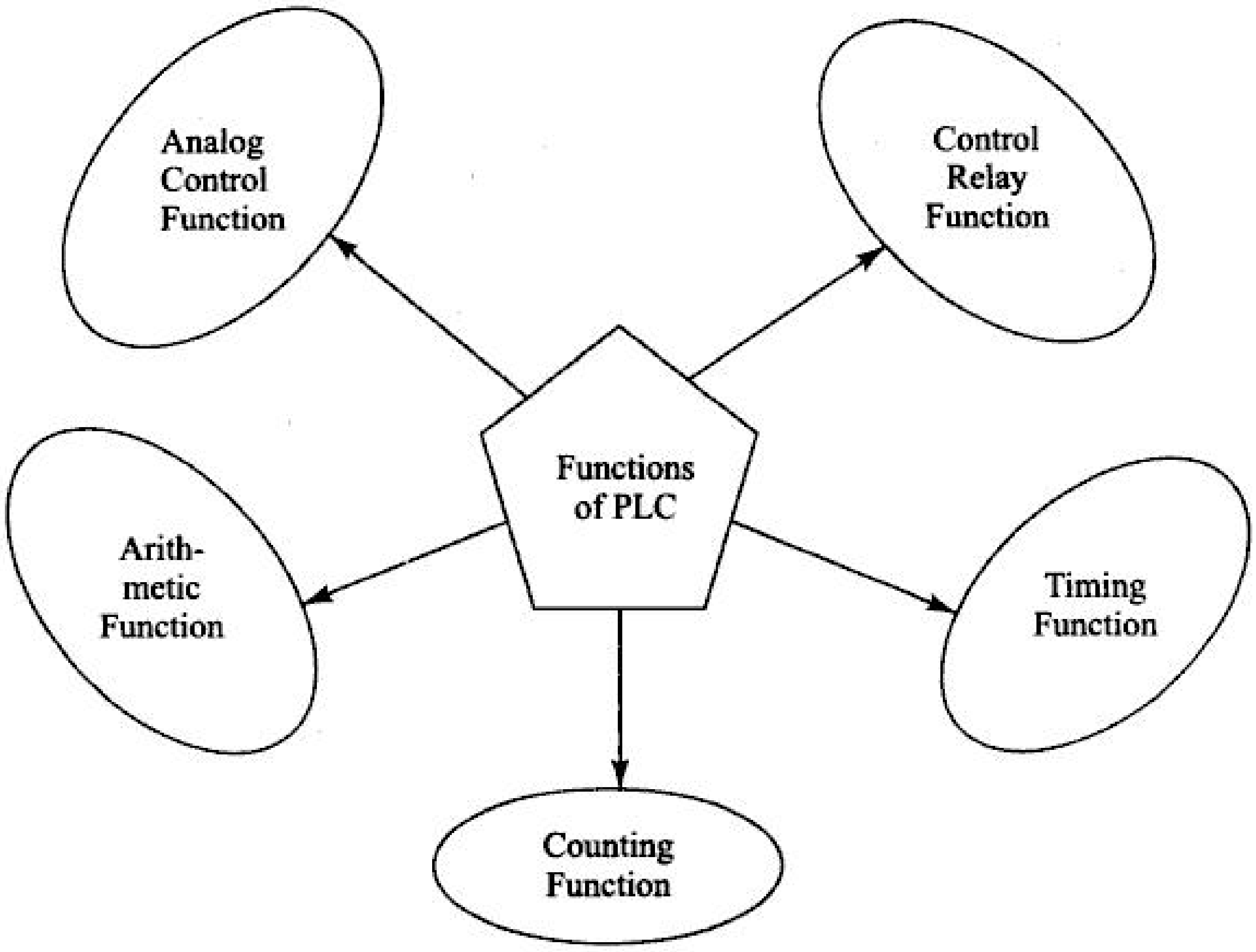


Fig. 2.36

2.7.4 Advantages of PLC

- Programming is easier
- Reprogramming is possible
- Requires less floor space
- Maintenance is easy
- Better reliability
- Can be interfaced with plant computer system

2.7.5 Comparison between PLC and Computers

S.No.	*PLC*	*Computer*
1.	Can be interfaced with industrial processes	Requires special arrangements for connection with processing equipments
2.	Can withstand noise, vibration, humidity, electrical disturbances and temperatures	Difficult for computers to work in such an environment
3.	PLCs use relay ladder diagrams and other as programming languages	Computer languages are totally different. They use high level languages.

2.8 HARDWARE TRENDS[2]

Although there seems to be new hardware systems and components available every day from various vendors, the underlying concept of any one system falls under one of the four types discussed in Sec. 2.2. The reader must distinguish between hardware technology and hardware concepts. Technology is defined as the implementation of a concept. Technology usually changes rapidly. Therefore, it becomes difficult to predict over a long period of time and also results in confusing end users when making hardware-related decisions. On the other hand, hardware concepts seem to take a longer time to develop and, once a concept evolves, it lasts for quite some time. Considering PCs, the underlying concept is the single-user or personal computing environment. However, the existing PC technology includes infinite models and types and keeps changing every day. It should also be realized that concepts and technology influence each other and both originate from the end user and criticism of existing technology. A methodology for technology prediction based on the available hardware concepts is presented at the end of this section.

It is expected that engineering design and manufacturing usages will increase dramatically in the future due to the steady decrease in computer response time and steady decrease in hardware and software costs. For example, MCAE (Mechanical Computer Aided Engineering) systems (see Figs 2.37) are dedicated to design functions. Hardware networking and true integration between CAD, CAM and CAD and CAM software will play a major role in increasing these usages.

Another future trend is the increasing modularity of CAD/CAM and increasing standardization of many CAD/CAM components. These standards will provide users with more options and the ability to configure their systems to their needs. As technology advances and user needs change, new standards will replace the old. It is therefore not necessary to have permanent hardware standards. What is happening and will continue to evolve is that every component of CAD/CAM systems, from operating systems to LANs, will be standardized among some key manufacturers.

Fig. 2.37 Desktop MCAE System Based on an Enhanced IBM PC/AT. *(Courtesy of Aries Technology, Inc.)*

Upgradability of existing CAD/CAM systems seems and will continue to be difficult. Technology progresses too fast to allow major vendors to invest into providing fully upgradable systems. They generally provide only an ability to convert

[2] This section includes excerpts from the Frost and Sullivan report 1564.

data from one upgrade to the other. Other than data, hardware, software and training investment become obsolete.

Other trends can be summarized as follows. There has been a shift toward more open architecture. Prices will continue to decline. There will be a continuation of the trend toward more desktop systems. Mainframes will be used to support centralized management of software and data, rapid computations, archival storage and network management.

Tables 2.4 and 2.5 summarize forecasts of CAD/CAM systems by configuration type in dollars and units terms respectively for the period 1984-1992. Measured in 1985 U.S. dollar values, as shown in Table 2.4, CAD/CAM systems sales are expected to increase from $2484 million in 1984 to $7951 million in 1992 at a growth rate of about 15 percent per year. The table also shows the clear trend that workstation-based and microcomputer-based CAD/CAM systems will become prevailing concepts in the years to come. Figures 2.38 to 2.40 present Table 2.4 in terms of line graphs and pie charts. As seen from Table 2.5, the growth rate of workstations is expected to be the largest, at an average yearly rate of 27.4 percent in the period 1985-1992. Personal computer-based, mainframe-based and turnkey systems sales are to grow by 20.1, 16.5 and 15.1 percent respectively. When CAD/CAM sales are measured in the number of graphics seats (the maximum number of simultaneous interactive graphics users), as shown in Table 2.5, workstations are followed in growth by turnkey CAD/CAM system seats, personal computer-based units and mainframe-based seats respectively. Figure 2.41 to 2.44 present Table 2.5 as line graphs and pie charts.

In order to predict long-term changes in computer hardware a methodology for technology prediction is described here. The method is based on the observation that parameters governing computer advances in technology from year to year are relatively constant. These parameters, for example, include the cost per chip and cost per unit weight. The method presents a computer tier model based on the following axioms:

1. Regardless of their size, the cost of computer hardware is about $200/lb when adjusted for inflation.
2. Weight is the dominant factor in deciding computer use independently of time. Other factors such as memory, disk space and speed of execution seem to be less dominant. For example, a person is expected to use 50 lb of computer in the office at all times.
3. Based on axiom 2, there exists seven tiers, as shown in Table 2.6. There is a factor of ten in weight separating the tiers. The same factor exists in price based on axiom 1.
4. A factor-of-ten improvement is achieved every seven years. This is suggested by the fact that there is about 35 percent per year aggregate technology improvement across the tiers. This implies that technology migrates from one tier to the next less-capable tier in seven years.
5. The transition from one tier to another introduces qualitative changes in computing usage which are usually the most difficult to predict. For example, while the tier model could have predicted the emergence of

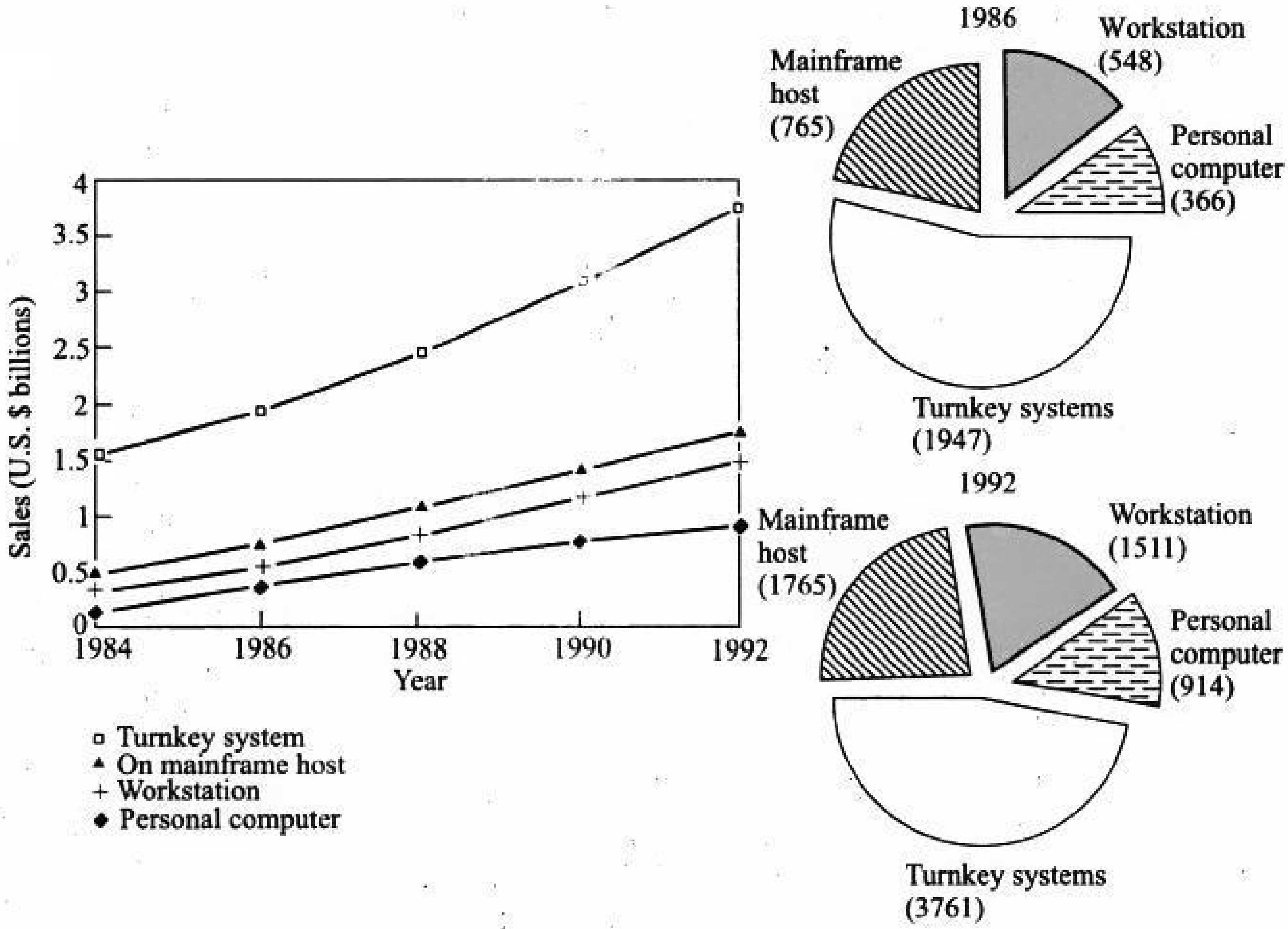


Fig. 2.38 *CAD/CAM Market Sales.*

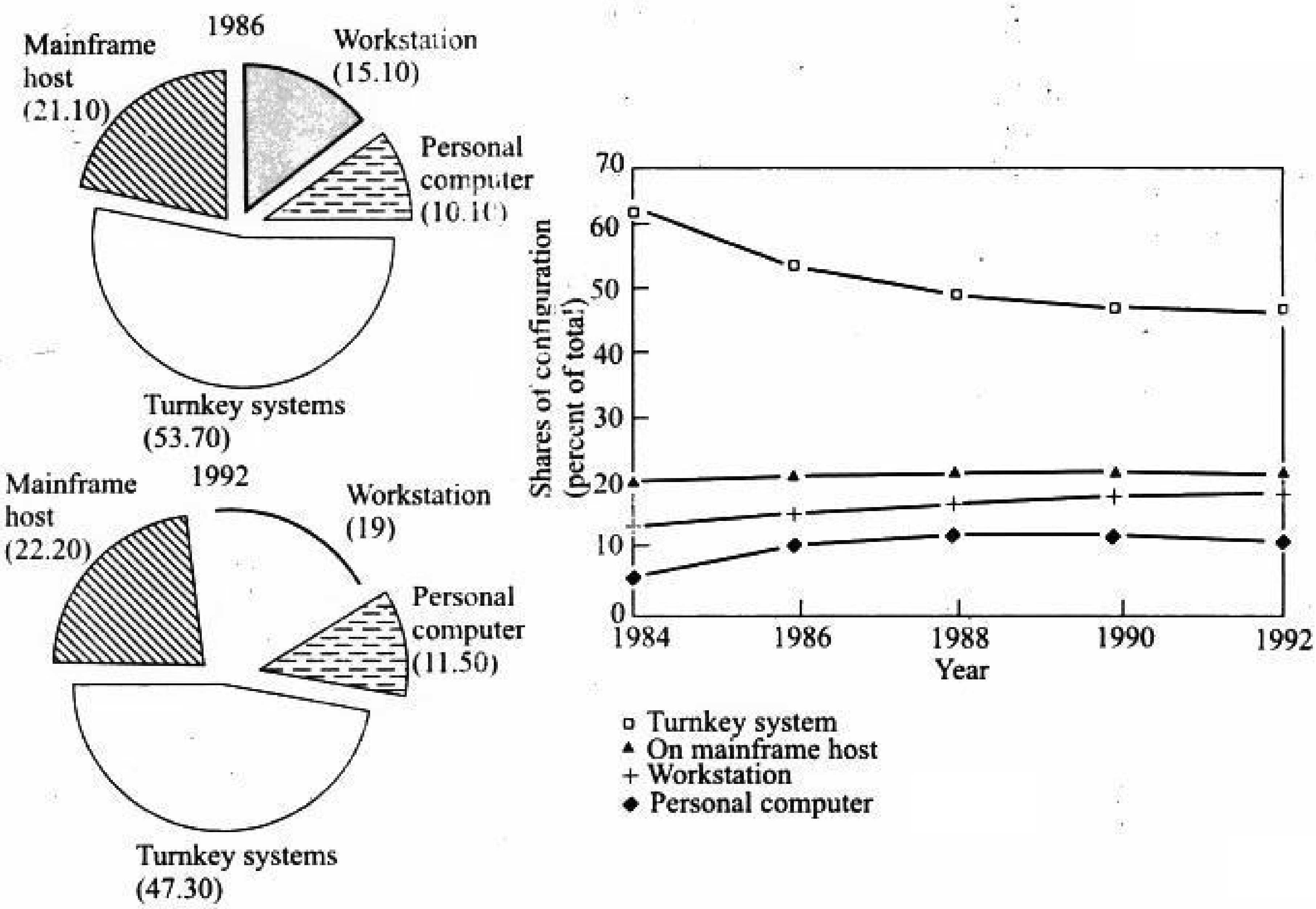


Fig. 2.39 *Shares of CAD/CAM Systems.*

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Vaughan, J.: "Solids Modeling: Hardware Grabs the Spotlight," *Digital Des.*, pp. 50–53, August 1986.

Wang, P. C. C. (Ed.): Advances in CAD/CAM *Workstations; Case Studies*, Kulwer Academic, Hingham, Mass., 1986.

Williams, T.: "Graphics Processing Migrates from Host to Workstation," *Computer Des.*, pp. 49–57, July 1985.

Yoshikawa, K., H. Yamaguchi and T. Asano: "Captive Touch Panel Using Uniform Resistive Film," *Fujitsu Sci. Tech. J.*, vol. 22, no. 2, pp. 124–131, 1986.

Zyda, M. J.: "Workstation Graphics Capabilities for the 1990s and Beyond," *Computer Graphics,* 86, *NCGA*, vol. III, pp. 442–453, 1986.

installation in an industrial environment is the support of the company management during and after the training period. The post-training support is very crucial because the trainee's early productivity on the CAD/CAM system to complete a design task may be less than that needed to complete the same task utilizing conventional ways.

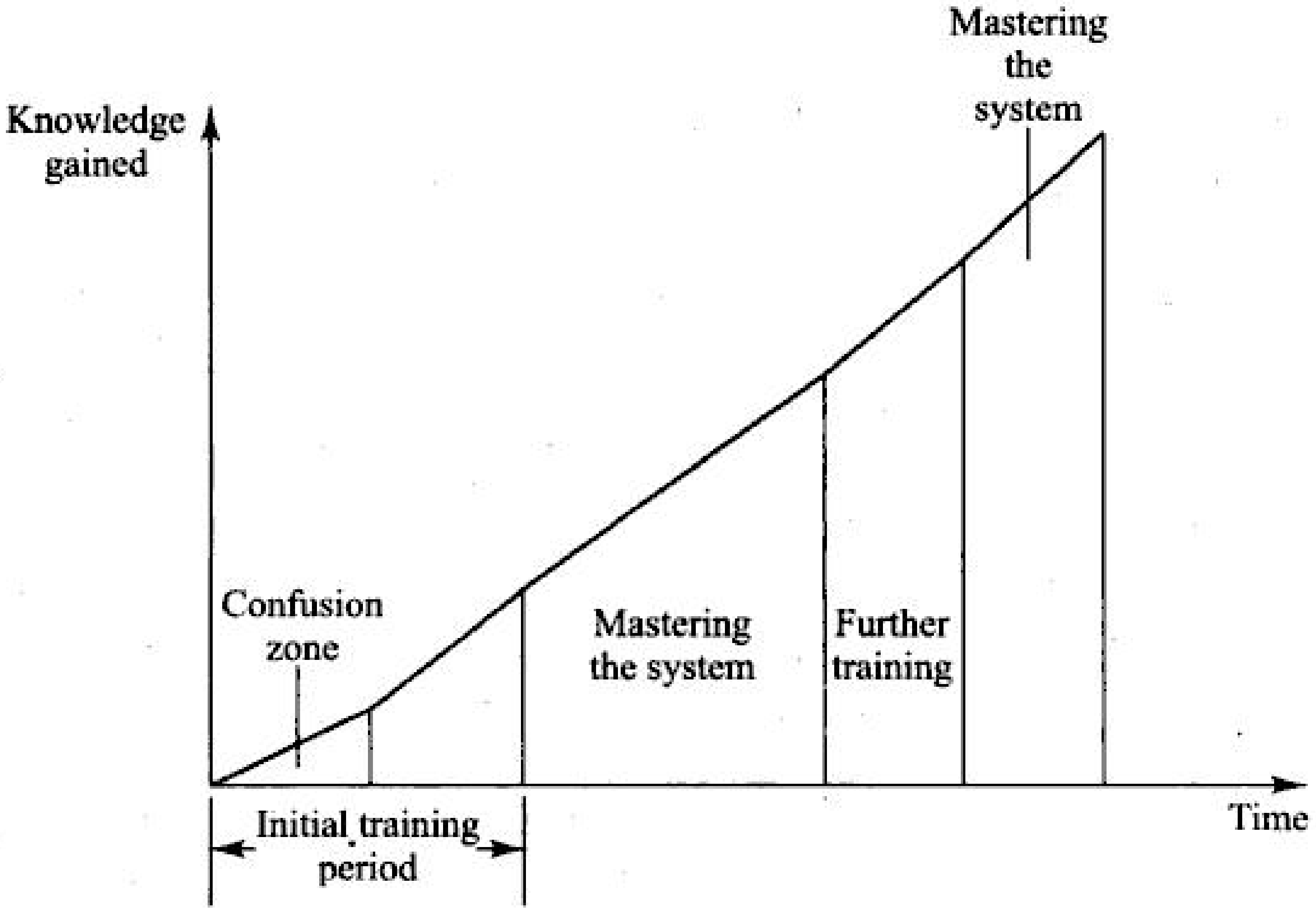


Fig. 3.1 *Typical Learning Curve During CAD/CAM Training*

Users of CAD/CAM software can be classified into three groups: software operators, applications programmers and system programmers. The majority of users including engineers and designers fall into the operators category. The main concern of this group is to master using the software so that the anticipated productivity increases are achieved. A typical operator tends to specialize in one or two modules of the software. For example, a designer may master geometric modeling and finite element modeling and analysis modules while a draftsman concentrates primarily on the drafting and detailing module. Operators are usually assisted by their system manager and vendors' customer response centers and hot lines to resolve any problems and answer any questions they may encounter in their day-to-day use of the software. Applications programmers can develop new programs and link them with the software, but they are not allowed to modify the existing source code. Such a need arises when a CAD/CAM system is to be customized by adding special applications and means to its software. These programmers are also experienced operators of the software with extensive programming background. They are usually familiar with the programming module of the software. In contrast to applications programmers, system programmers have the privilege to change the source code. In essence, they are the developers of the software itself. They are usually knowledgeable of the internal organization of the software, its database structure and its database management system. They also know how to modify the user interface and usually possess backgrounds in computer graphics, engineering analysis and computer science. They usually work for either turnkey vendors or R&D groups and centers.

system must also be considered to avoid locking the system into software that will be unnecessarily difficult to upgrade over the coming years. Finally, knowledge of these standards and their functions might stimulate engineers to think of developing design and manufacturing standards, through engineering organizations and enforce them on CAD/CAM vendors.

3.3 BASIC DEFINITIONS

This section introduces some of the basic terms and their definitions. The knowledge gained in this section and the remainder of the chapter should enable the readers to understand how their in-house CAD/CAM systems are structured, how they work and how to go about using them effectively in an engineering environment.

3.3.1 Data Structure

Formally a data structure is defined as a set of data items or elements that are related to each other by a set of relations. Applying these relations to the elements of the set results in a meaningful object. From a CAD/CAM point of view, a data structure is a scheme, logic, or a sequence of steps developed to achieve a certain graphics, non-graphics and/or a programming goal.

As an example consider the object shown in Fig. 3.3. Three different types of data structures have been identified to construct the object. They are based on edges, vertices, or blocks. Within the context of the above formal definition of a data structure, the set of edges, vertices, or blocks is the set of data items for each type and edges, vertices, or blocks are the data items themselves. Furthermore, the connectivity vertices for the first type, the edge information for the second and the set operators for the third form the set of relations required by each type. As an example, [1, A & B] in Fig. 3.3*b* indicates that vertex 1 is shared by edges A and B while in Fig. 3.3*c*, [A, 1 & 4] indicates that edge A has the two vertices 1 and 4.

3.3.2 Database

The term "database" is commonly used and may mean different things to different users. Casually, it is synonymous with the terms "files" and "collection of files." Formally, a database is defined as an organized collection of graphics and nongraphics data stored on secondary storage in the computer. It could, therefore, be viewed as the art of storing or the implementation of data structure into the computer. Hence, it is a repository for stored data. From a software development point of view, a decision on the data structure has to be made first, followed by a choice of a database to implement such a structure. There may exist more than one alternative of database to implement a given data structure.

The objective of a database is to collect and maintain data in a central storage so that it will be available for operations and decision-making. The advantages that accrue from having centralized control of the data, or a centralized database, is manifold:

database readily accessible for applications. Object-oriented database models include the entity relationship model, complex object representation, molecular object representation and abstract data model. The abstract data model is close to solid modeling databases. It employs abstract objects as primitives in the design of the database. Figure 3.7 shows an example of this database. Primitives are constructed from input data and form the lowest field or record of storage in the database.

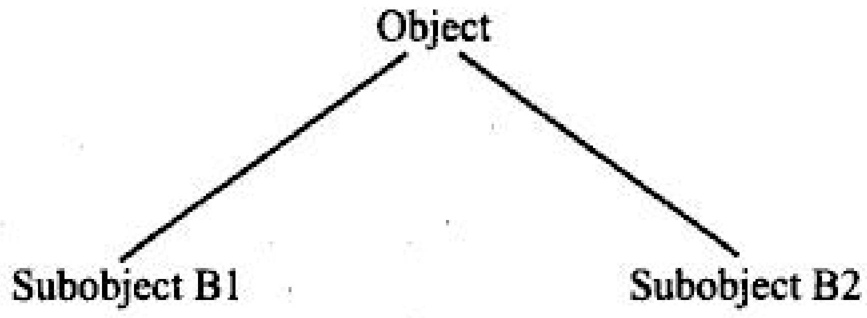


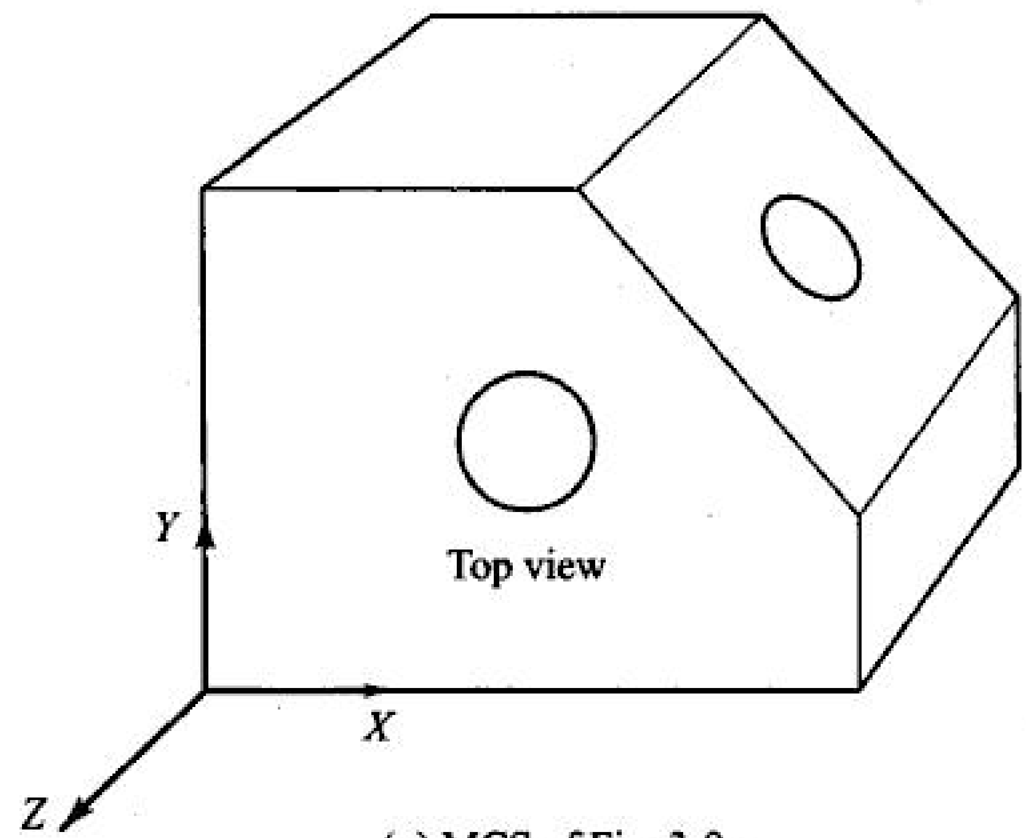
Fig. 3.7 *Sample Object-oriented Database of Object Shown in Fig. 3.3*

Object-oriented databases seem to be ideal for CAD/CAM applications. Hybrid database models may also be useful. The following are some of the functional requirements and specifications that CAD/CAM databases must support:

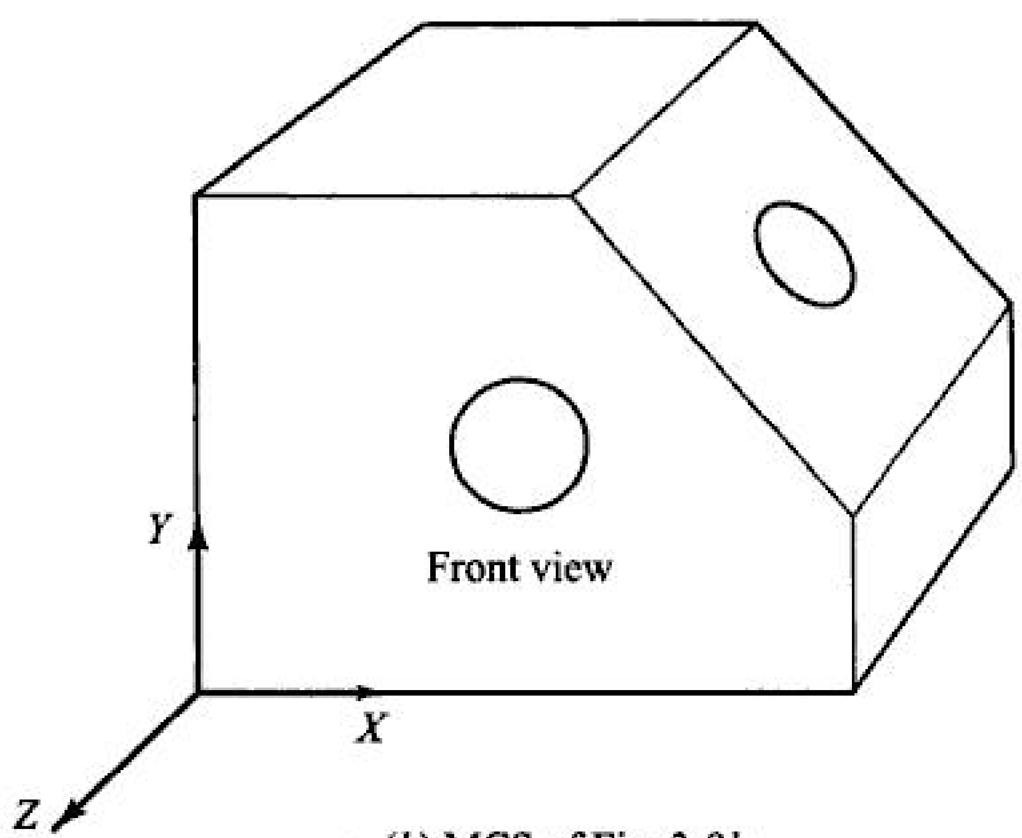
1. Multiple engineering applications from conceptual design to manufacturing operations.
2. Dynamic modification and extension of the database and its associativity.
3. The iterative nature of design. This nature is not common in business data processing. CAD/CAM database management systems must support the ten tative, iterative and evolutionary nature of the design process.
4. Design versions and levels of detail. CAD databases must provide a capability for storage and management of multiple design solutions that may exist for a particular design. There is seldom a unique solution to a design problem and there may exist several optimal solutions.
5. Concurrent and multiple users must be supported from the database. Large design projects usually involve multiple designers working simultaneously on multiple aspects of a project.
6. Temporary database support. Due to the iterative nature of design, earlier generated data may not be committed to the database until the design process is completed.
7. Free design sequence. The database system should not impose constraints on the designer to follow because different designs require different sequences.
8. Easy access. Application programs requiring data from a CAD/CAM database should not require extensive knowledge of the database structure to extract the data needed. This is important in customizing CAD/CAM systems for specific design and manufacturing procedures.

3.3.3 Database Management System (DBMS)

A DBMS is defined as the software that allows access to use and/or modify data stored in a database. The DBMS forms a layer of software between the physical



(*a*) MCS of Fig. 3-9*a*

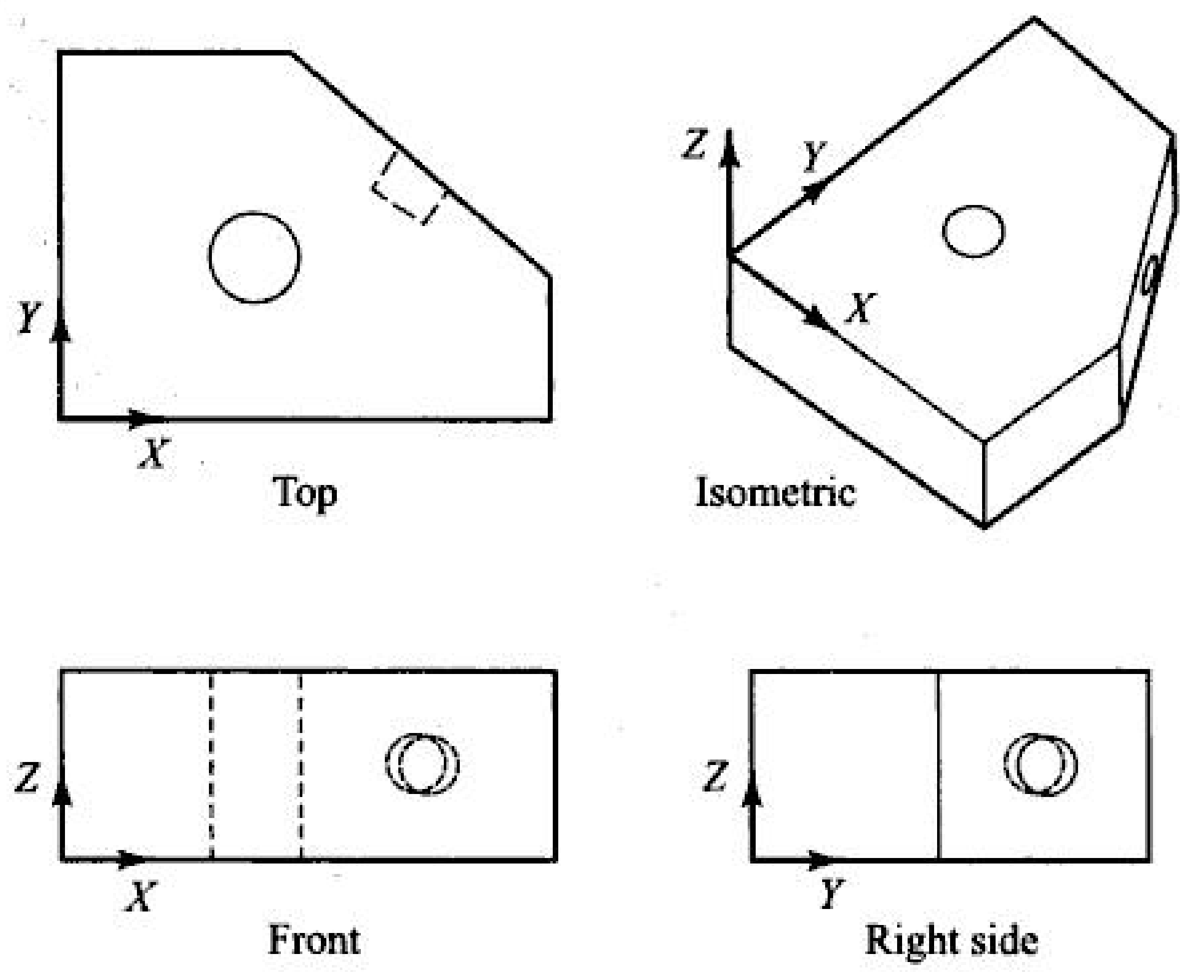


(*b*) MCS of Fig. 3-9*b*

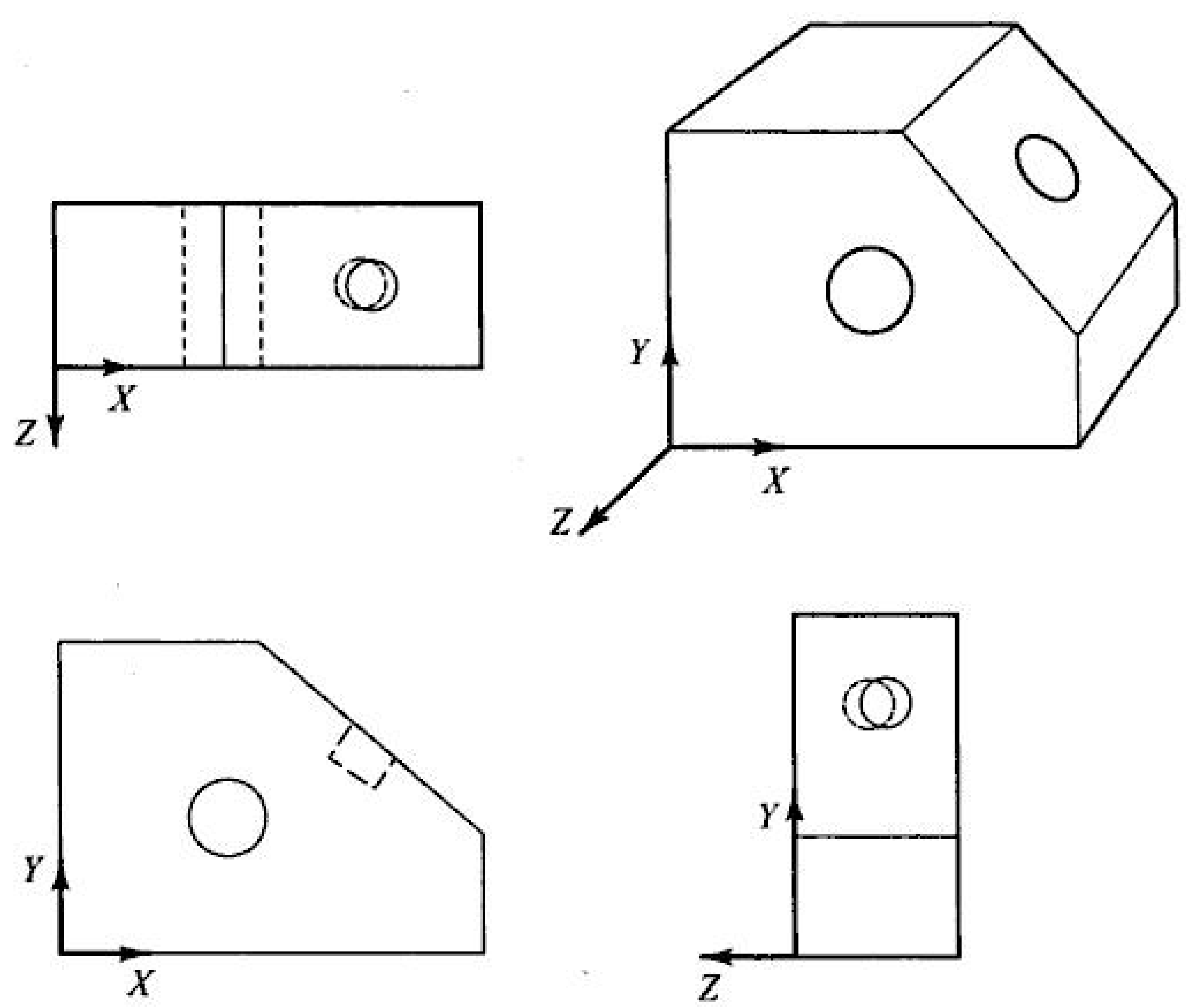
Fig. 3.11 *Orientation of MCS Relative to the Model*

3.3.5 Working Coordinate System

It is often convenient in the development of geometric models and the input of geometrical data to refer to an auxiliary coordinate system instead of the MCS. This is usually useful when a desired plane (face) of construction is not easily defined as one of the MCS orthogonal planes, as in the case of inclined faces of a model. The user can define a cartesian coordinate system whose X Y plane is coincident with the desired plane of construction. That system is the working coordinate system (WCS). It is a convenient user-defined system that facilitates geometric construction. It can be established at any position and orientation in space that the user desires. While the user can input data in reference to the WCS, the software performs the necessary transformations to the MCS before storing the data. The ability to use two separate coordinate systems within the same model database in relation to one another gives the user great flexibility. Some software such as those of Computervision refer to the WCS as a construction plane.



(*a*) Utilizing MCS of Fig. 3-9*a*



(*b*) Utilizing MCS of Fig. 3-9*b*

Fig. 3.12 *Views of Object Shown in Fig. 3.10*

The definition of a WCS requires three noncollinear points. The first defines the origin and the first with the second define the X axis. The third point is used to define the $X\,Y$ plane of the WCS. The Z axis is determined as the cross-product of the two unit vectors in the directions defined by the lines connecting the first and the second (the X axis) and the first and the third points. The Y axis is determined as the cross product of the Z and X unit vectors (see Prob. 3.3). We will use the subscript W to distinguish the WCS axes from those of the MCS. The $X_W Y_W$ plane becomes the active construction (working) plane if the user defines a WCS. In this case, the WCS and its corresponding $X_w Y_w$ plane override the MCS and the default construction plane respectively. As a matter of fact, the MCS with its default construction plane can be viewed by the user as the default WCS with its $X_w Y_w$ plane. All CAD/CAM software packages provide users with three standard WCSs that correspond to the three standard views of front, top and right sides. Other WCSs can be defined by the users.

There is only one active WCS at any one time. If the user defines few WCSs in one session during a model construction, the software recognizes only the last one and stores it with the model database if the user stores the model by filing the session work. When retrieved later, the user is advised to display the axes of the current WCS before beginning construction to check its origin and orientation. If confused, the user can simply set the WCS back to the MCS by using the same command that defines a WCS but with the proper modifiers.

Once a WCS is defined, user coordinate inputs are interpreted by the software in reference to this system. At the mean time, the software calculates the corresponding homogeneous transformation matrix between the WCS and the MCS to convert these inputs into coordinates relative to the MCS before storing them in the database. The transformation equation can be written as

$$P = [T]P_W \tag{3.1}$$

where P is the position vector of a point relative to the MCS and P_w is the vector of a point relative to the active WCS. Each vector is given by

$$P = [x \quad y \quad z \quad 1]^T \tag{3.2}$$

The matrix $[T]$ is the homogeneous transformation matrix. It is a 4 x 4 matrix and is given by

$$[T] = \left[\begin{array}{ccc|c} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{c|c} {}^{M}_{W}[R] & {}^{M}P_{W,org} \\ \hline 0\ 0\ 0 & 1 \end{array}\right] \tag{3.3}$$

where ${}^{M}_{W}[R]$ is the rotation matrix that defines the orientation of the WCS relative to the MCS and ${}^{M}P_{W,\text{org}}$ is the position vector that describes the origin of the WCS relative to the MCS. The columns of ${}^{M}_{W}[R]$ give the direction cosines of the unit vectors in the X_w, Y_w and Z_w directions relative to the MCS, as shown in Fig. 3.13.

The WCS serves another function during geometric construction. Its $X_w Y_w$ plane is used by the software as the default plane of circles. A circle plane is usually not defined using its center and radius. In addition, the Z_w axis of a WCS can be useful in defining a projection direction which may be helpful in geometric construction.

Example 3.2 Write a procedure to construct the holes shown in the model used in Example 3.1. Use the MCS shown in Fig. 3.9*a*.

Solution Let us assume that the user has defined the $(\text{WCS})_1$ as shown in Fig. 3.14 to construct the model without the holes. The procedure to construct the holes becomes:

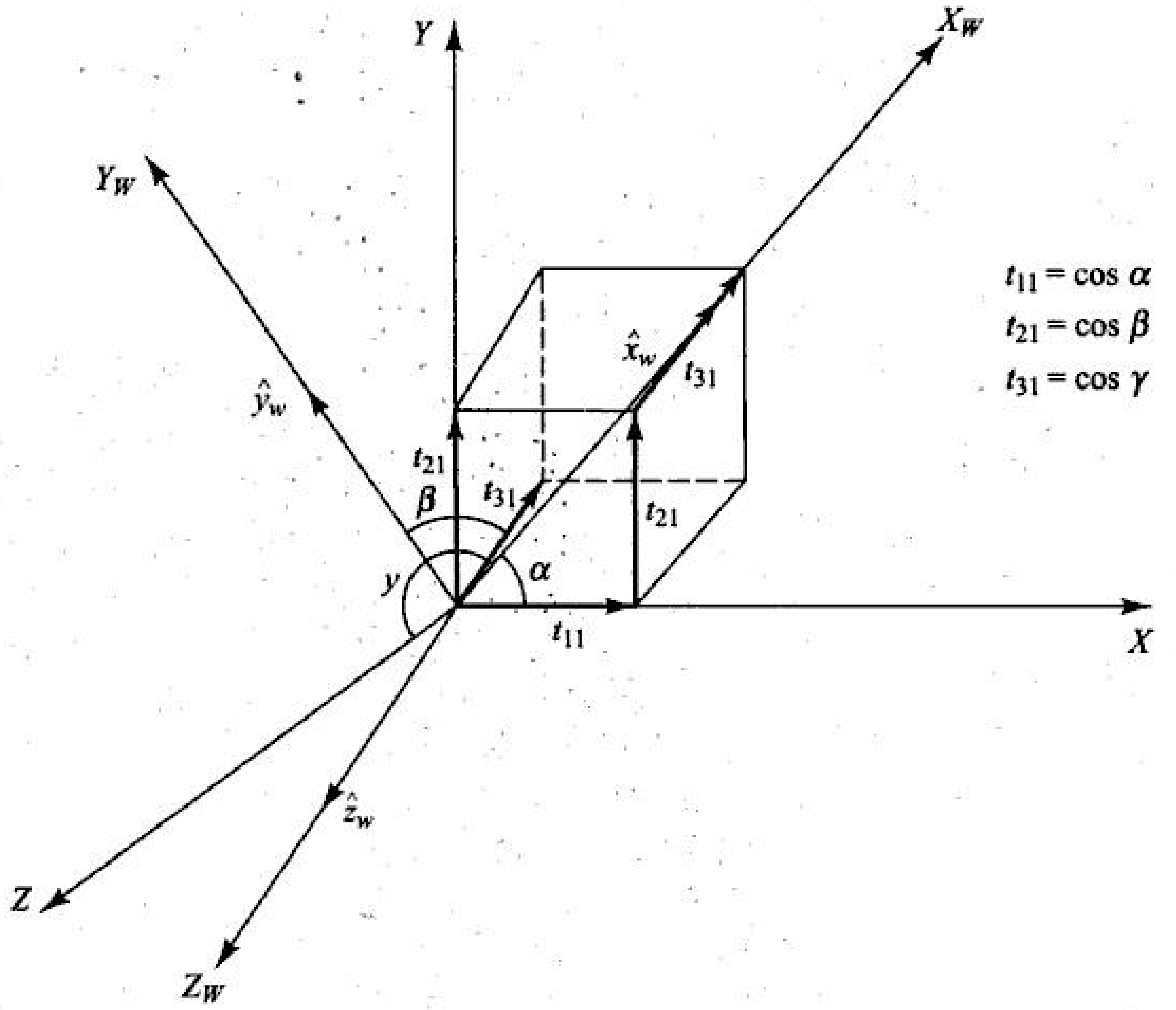


Fig. 3.13 *Direction Cosines of WCS Relative to MCS*

1. With the $(WCS)_1$ active, construct circle A with center (1,1,0) and radius 0.25.
2. Construct hole A by projecting circle A at a distance of – 1.0 (in the opposite direction to Z_{W1}).
3. Define $(WCS)_2$ as shown by using points E_1, E_2 and E_3.
4. Construct circle B with center (x_c, y_c) and radius 0.25. The center can easily be found implicitly as the midpoint of line $E_2 E_3$.
5. Repeat step 2 but with a distance —0.5 (in the opposite direction to Z_{W2}).

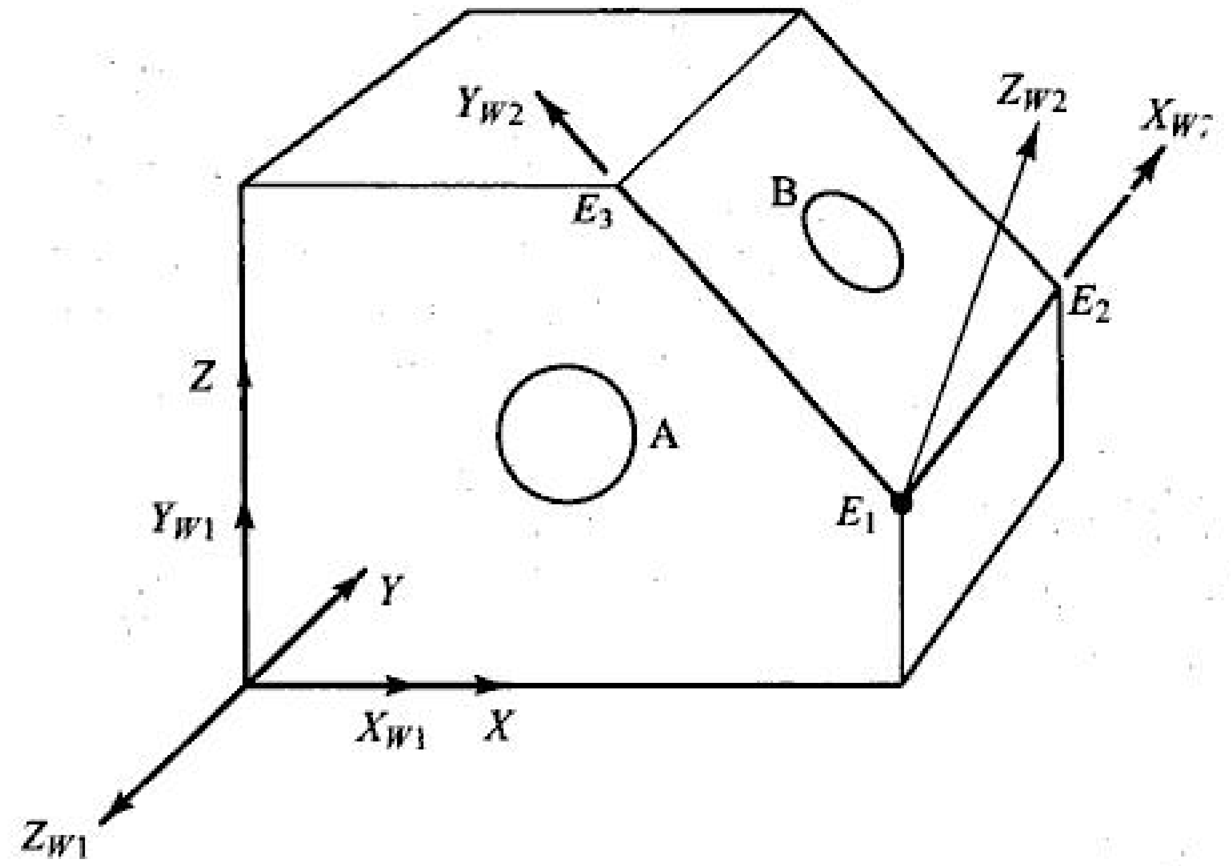


Fig. 3.14 *WCSs Required to Construct Holes A and B*

The coordinates in step 1 are given relative to $(WCS)_1$ which is active at the time of construction. With reference to Fig. 3.14, these coordinates are (1, 0, 1) relative to the MCS and these are the values that are stored in the model database. To verify this, using Eq. (3.3) we can write:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \tag{3.4}$$

A similar approach can be followed to find the center of the hole B and is left as an exercise for the reader at the end of the chapter (see Prob. 3.4).

3.3.6 Screen Coordinate System

In contrast to the MCS and WCS, the SCS is defined as a two-dimensional device-dependent coordinate system whose origin is usually located at the lower left corner of the graphics display, as shown in Fig. 3.15. The physical dimensions of a device screen (aspect ratio) and the type of device (vector or raster) determine the range and the measurement unit of the SCS. The SCS is mostly used in view-related digitizes such as definitions of view origin and window or digitizing a view to select it for graphics operations.

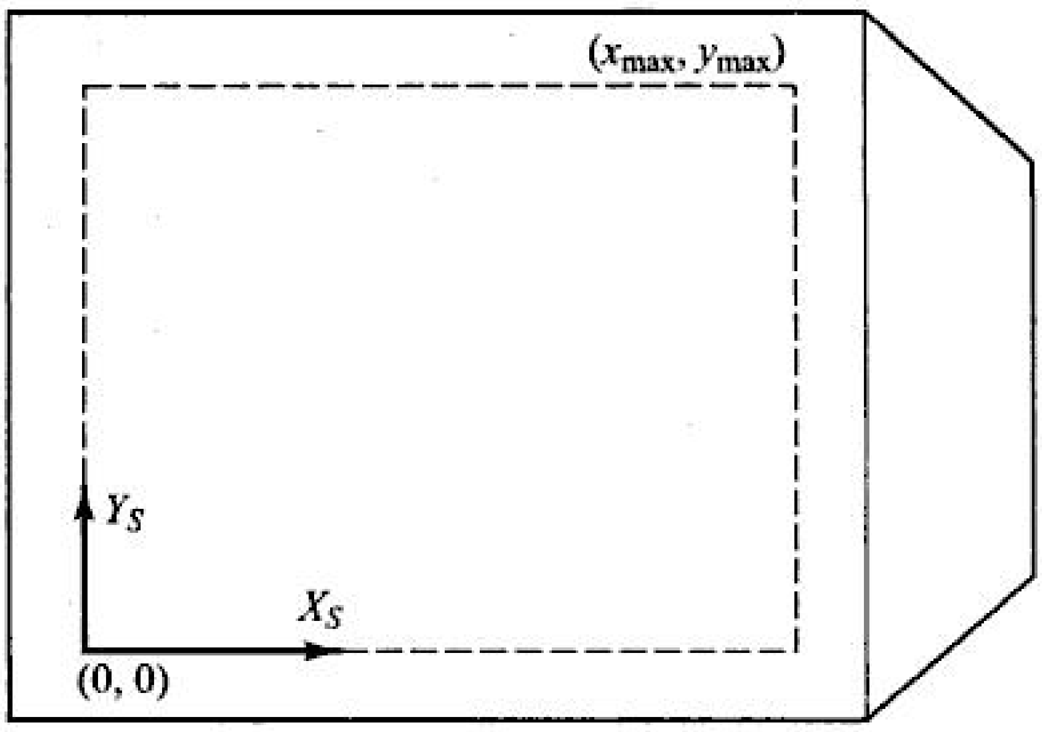


Fig. 3.15 *Typical SCS*

The range and the measurement unit of an SCS can be determined in three different methods. For raster graphics displays, the pixel grid serves as the SCS. A 1024 × 1024 display has an SCS with a range of (0, 0) to (1024, 1024). The center of the screen has coordinates of (512, 512). This SCS is used by the CAD/CAM software to display relevant graphics by converting directly from MCS coordinates to SCS (physical device) coordinates. This approach of defining SCSs is appropriate if the software supports or drives only one type of graphics display. For software packages that must drive multiple display units, it is convenient to define a normalized coordinate system that cap be utilized to represent an image. Such representation can be translated by device-dependent codes to the appropriate physical device coordinates. In such a case, the range of the SCS can be chosen

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

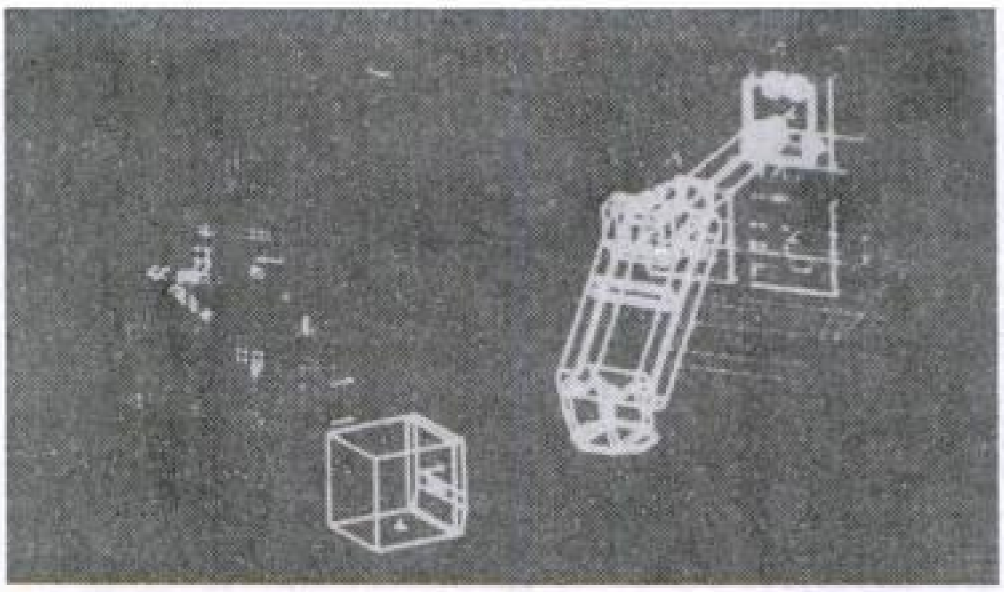


Fig. 3.38 *Robot Simulation Via Offline Programming. (Courtesy of GE (U.S.A.) Calma.)*

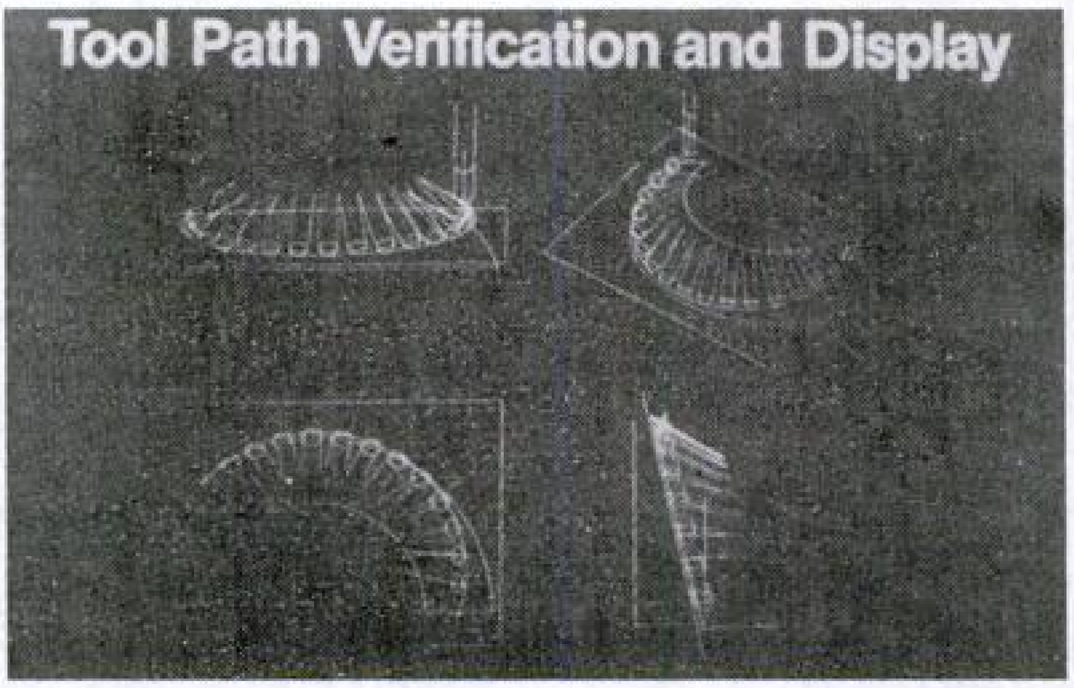


Fig. 3.39 *NC Tool Path Verification. (Courtesy of Computervision Corporation.)*

3.7 MODELING AND VIEWING

Modeling is the art of abstracting or representing a phenomenon and geometric modeling is no exception. Geometric modeling and simulation via computers have reached a level to replace the real-life prototypes or tests. A geometric model is defined as the complete representation of an object that includes both its graphical and nongraphical information. Objects can be classified into three types from a geometric construction point of view. These are two-and-a-half dimensional, three dimensional, or a combination of both. Figures 3.40 and 3.41 show some examples. As Fig. 3.40 shows, two-and-a-half-dimensional objects are classified to have uniform cross sections and thicknesses in directions perpendicular to the planes of the cross sections. Constructing such an object via wireframe modeling requires only constructing the proper entities (faces), projecting them along the proper directions by the thickness value and then creating the proper edges along these directions. This is a much more efficient way of construction than calculating and inputing the coordinates of all the corner points of the model. The construction of a true three-dimensional object requires the coordinate input of key points and then connecting them with the proper types of entities. Figure 3.41 shows a phone model. The model is two-and-a-half dimensional although it may seem to be three dimensional at first glance.

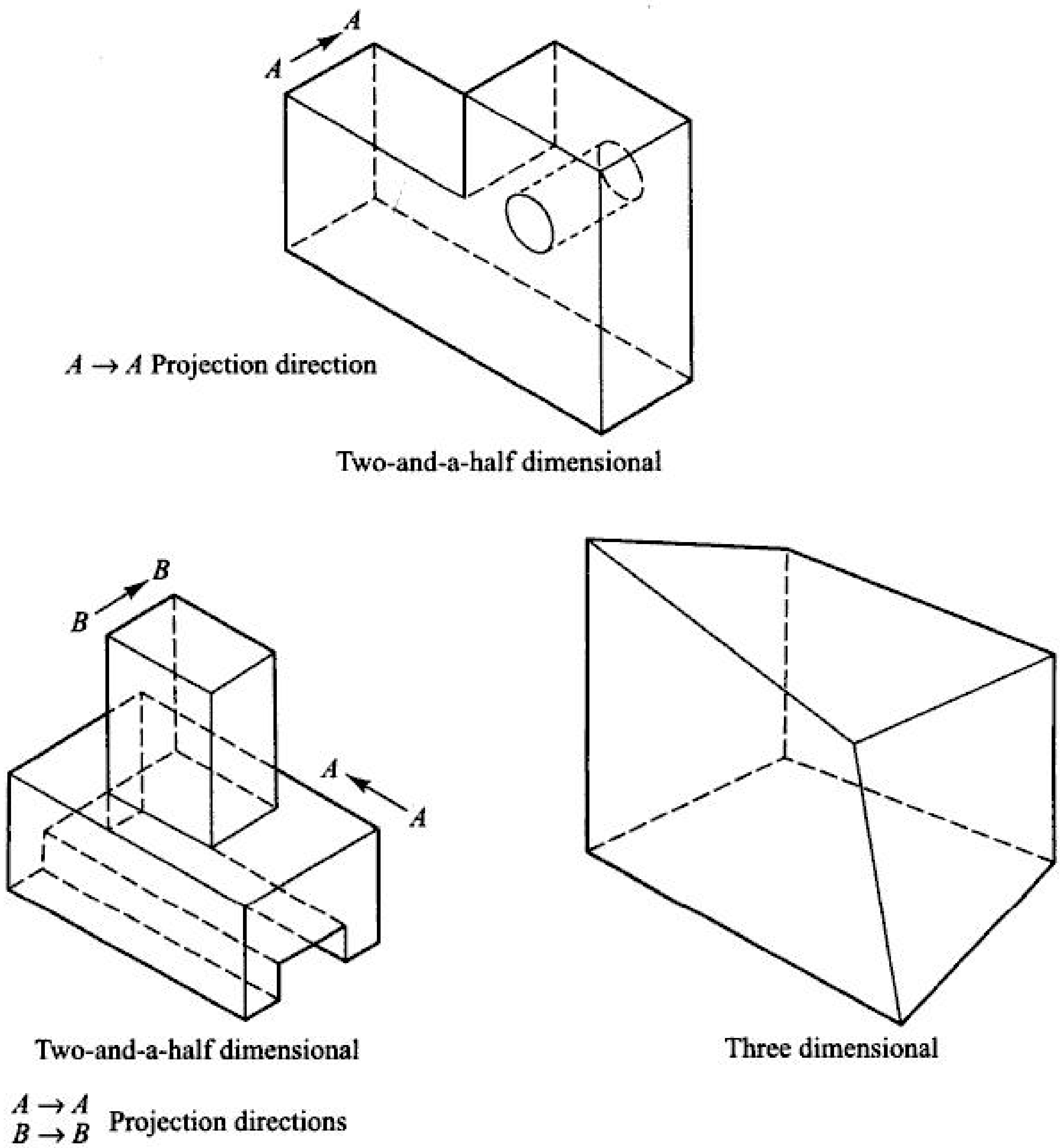


Fig. 3.40 *Two-and-a-half-dimensional and Three-dimensional Geometric Models*

Irrespective of the specific syntax of any software package, a particular model setup procedure is usually needed by the package to organize the model database before the user is allowed to construct geometry. The procedure can be listed in a generic form in the following order:

1. INITIATE NEW MODEL.
2. CHOOSE A SCREEN LAYOUT.
3. DEFINE THE WINDOWS OF THE LAYOUT AS THE MODEL DESIRED VIEWS.
4. SELECT THE PROPER CONSTRUCTION PLANE OR WCS.

In the first step, the model name becomes the file name that stores the geometric model information. The setup procedure results in a model database with the hierarchy shown in Fig. 3.42. Thus, the user cannot define views before choosing a screen layout. The user can choose the layouts and define their views at any time during construction and not only during the model setup. Dealing with a centralized database has two consequences on software response to user's commands regarding screen layouts and views. First, whenever the user defines a new view in a layout, existing model geometry in the model database is automatically transformed and displayed in the view window (viewpart). Second, if a view is deleted from a layout or an entire layout is deleted from the database, graphics entities making the view or the layout are not deleted. Only the view or the layout display disappear from the screen. As a matter of fact, the user can delete all views and all screen layouts from the database and still have the model entities invisible to the user. Entities are deleted only if the user does so explicitly.

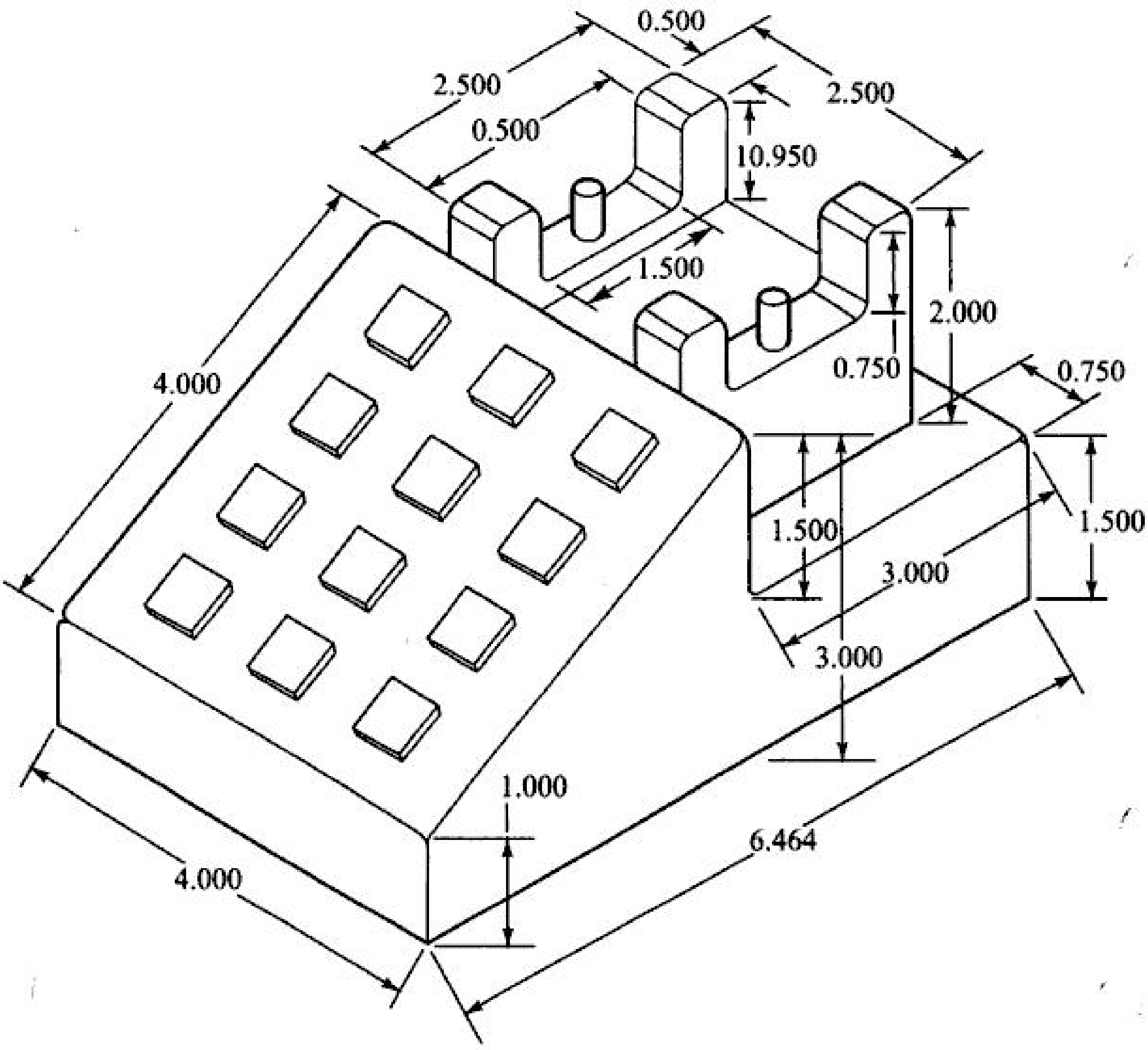


Fig. 3.41 *A Telephone Geometric Model*

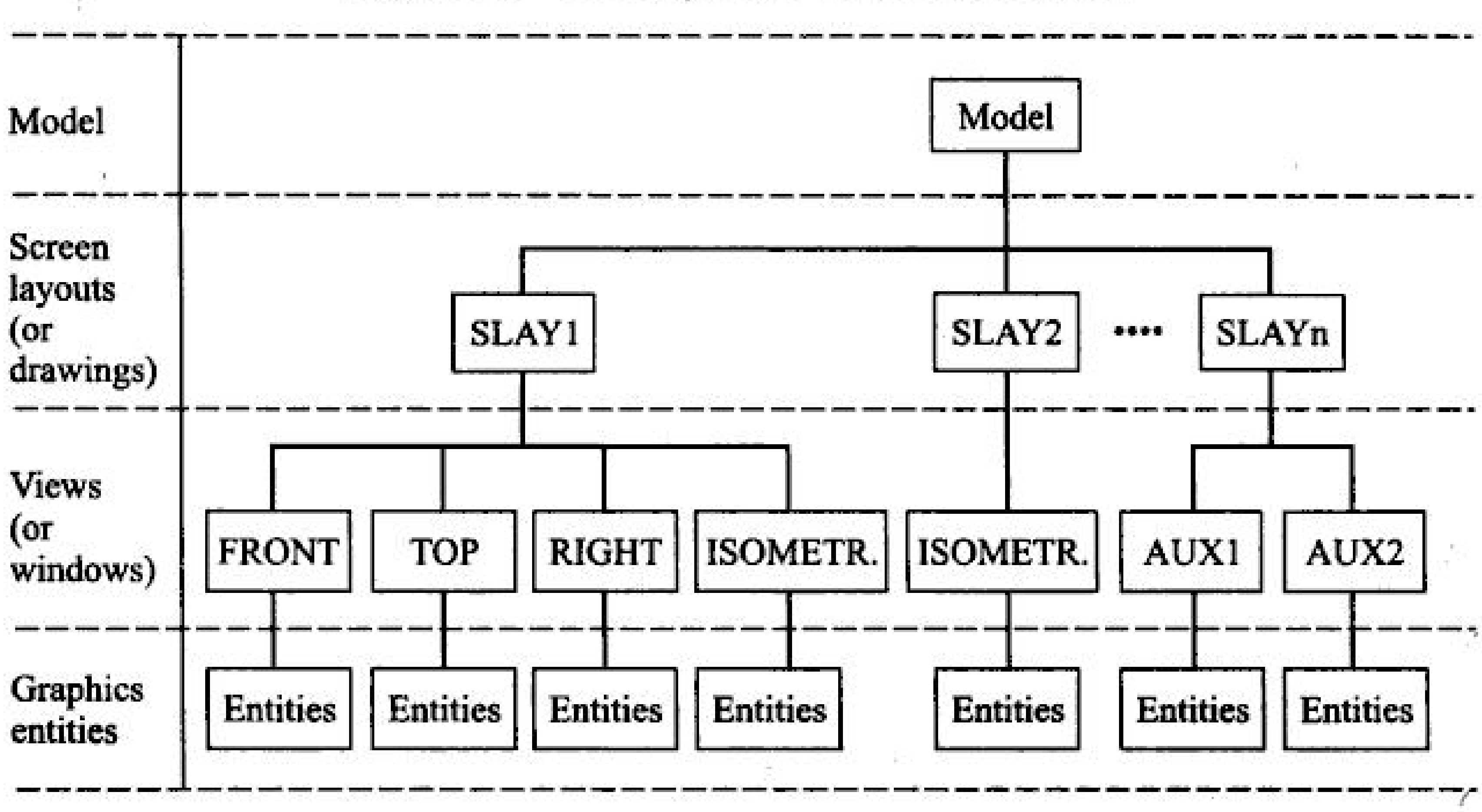


Fig. 3.42 *Typical Hierarchy of a Geometric Model Database*

level requires modifying and/or accessing the database. This level always requires extensive knowledge of the database structure of the software. While programs are more difficult to develop utilizing the second level of programming, they are usually more efficient to run.

3.10 EFFICIENT USE OF CAD/CAM SOFTWARE

CAD/CAM software is usually a complex software that requires training and understanding of its philosophy and underlying principles. Once this is achieved the user should develop the habit of devising a strategy to construct geometric models or achieve other goals before logging into the system. The following recommendations may be helpful to users:

1. Develop an efficient planning strategy. A good strategy for complex models can result in great time savings to create the model database. The user must decide first on the type of object at hand, if it is two-and-a-half-dimensional or three-dimensional and the type of geometric modeling desired, if it is wireframes, surfaces, or solids. The key faces should be identified for two-and-a-half-dimensional objects. If more than one alternative exists, choose the one that makes it easier to construct the model.

   Consider, as an example, the model shown in Fig. 3.46. The easiest way to construct this object is by constructing the entities of face (A) and projecting them back along the direction shown and then creating the holes. Other alternatives exist but they are not optimal. For both two-and-a-half-dimensional

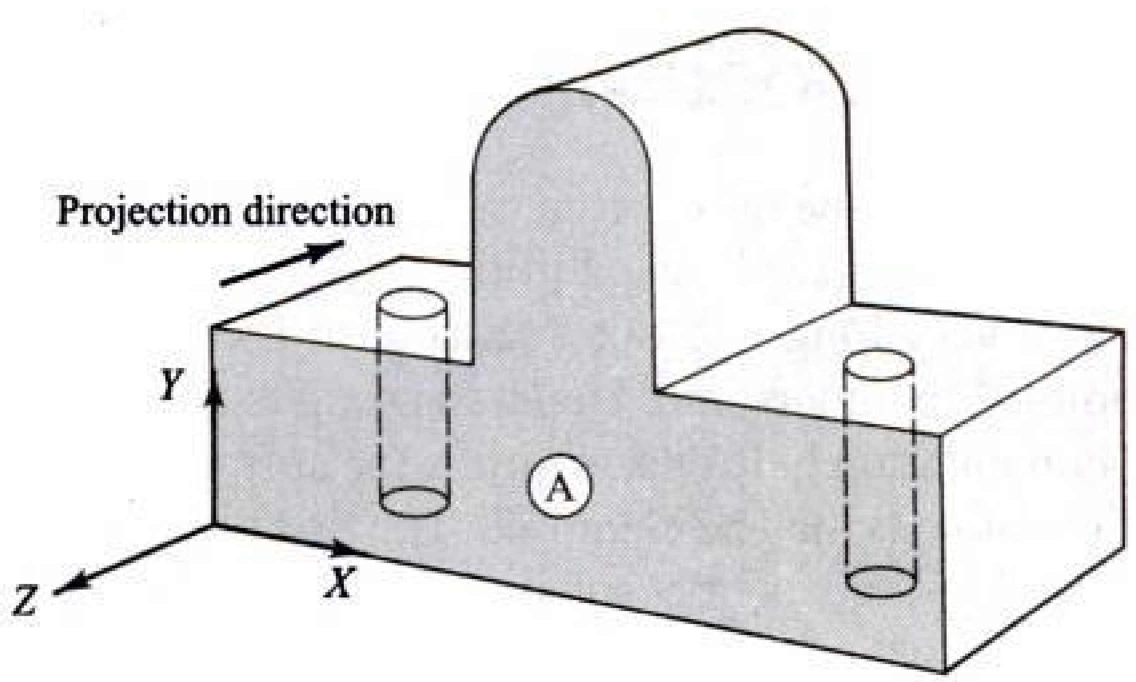


Fig. 3.46 *Plan to Construct a Two-and-a-half-dimensional Object*

   and three-dimensional objects, the user must develop the key coordinates required to construct the model relative to the chosen MCS and possible WCSs. The general rule here is that the user should avoid excessive coordinate calculations. The software usually provides users with many tools in this regard, such as WSCs, geometric modifiers and graphic manipulations. Chapters 10 and 11 cover these topics in detail.

   Planning strategy should also include the choice of the MSC origin and orientation, the screen layout, views and colors. Typically, a screen layout with either an isometric view or four views (front, top, right and isometric)

structure and management as well as on the completeness and uniqueness of the geometric modeling techniques utilized to develop these databases. Among the various available techniques, solid modeling seems to be the key technique to automate and integrate the CAD and CAM functions.

Adaptive analysis and optimization are other characteristics that CAD/CAM software should possess to better serve design and engineering applications. These characteristics reflect the iterative nature of design. Instead of leaving the burden of adaptation and optimization to the user, it is very beneficial if software can provide it. Expert and learning systems as well as AI (artificial intelligence) techniques are useful in this regard.

PROBLEMS

Part 1: Theory

3.1 What is meant by segmentation?

3.2 Define the following graphic elements in interactive computer graphics.

(a) Points (b) Surfaces (c) Curves

Part 2: Theory

3.1 Give an example of how the centralized integrated database concept can help the "what if" situations that arise during the design process.

3.2 Discuss the contents of a database for a line, a circle and an arc.

3.3 Can you define a nonorthogonal WCS? How is the three-point definition interpreted by software?

Part 3: Laboratory

Use your in-house CAD/CAM system to answer the questions in this part.

3.4 What is "graphic software"? What are the functions of it? Explain briefly.

3.5 Draw the graphic software modules configuration. Write the functions of each.

3.6 Explain the following transformation in 2D & 3D concept of computer graphics with individual examples.

(a) Translation (b) Scaling (c) Rotation

3.7 What are the needs for graphic standards? Explain in detail.

3.8 Discuss various standards functioning at different levels of graphic system.

3.9 Nearly all functions of a CAD system depend on its database. Where does it reside and what are its contents? Write model application patterns based on their suggested database organization.

3.10 The model database can be organized in various ways. Briefly explain the following possible data structures.

(*a*) Storing the coordinates of geometry
(*b*) Including the graph based model
(*c*) Using Boolean operations

3.11 With a neat diagram, explain briefly the organization of a typical CAD/CAM software.

3.12 What are the advantages of having a centralized database?

during the construction phase and on the expected utilization of the resulting database later in the design and manufacturing processes. Regardless of the chosen technique, the user constructs a geometric model of an object on a CAD/CAM system by inputting the object data as required by the modeling technique via the user interface provided by the software. The software then converts such data into a mathematical representation which it stores in the model database for later use. The user may retrieve and/or modify the model during the design and/or manufacturing processes.

To convey the importance of geometric modeling to the CAD/CAM process, one may refer to other engineering disciplines and make the following analogy. Geometric modeling to CAD/CAM is as important as governing equilibrium equations to classical engineering fields as mechanics and thermal fluids. From an engineering point of view, modeling of objects is by itself unimportant. Rather, it is a means (tool) to enable useful engineering analysis and judgment. As a matter of fact, the amount of time and effort a designer spends in creating a geometric model cannot be justified unless the resulting database is utilized by the applications module discussed in Chapter. 3.

The need to study the mathematical basis of geometric modeling is manifold. From a strictly modeling point of view, it provides a good understanding of terminology encountered in the CAD/CAM field as well as CAD/CAM system documentation. It also enables users to decide intelligently on the types of entities necessary to use in a particular model to meet certain geometric requirements such as slopes and/or curvatures. In addition, users become able to interpret any unexpected results they may encounter from using a particular CAD/CAM system. Moreover, those who are involved in the decision-making process and evaluations of CAD/CAM systems become equipped with better evaluation criteria.

From an engineering and design point of view, studying geometric modeling provides engineers and designers with new sets of tools and capabilities that they can use in their daily engineering assignments. This is an important issue because, historically, engineers cannot think in terms of tools they have not learned to use or been exposed to. The tools are powerful if utilized innovatively in engineering applications. It is usually left to the individual imagination to apply these tools usefully to applications in a new context. For example, the mere fact that CAD/CAM databases are centralized and associative provides great capabilities that are utilized in Sec. 4.8 of this chapter. These capabilities are usually more efficient than writing analyses and plotting programs on conventional computers.

Having established the need for geometric modeling, what is the most useful geometric model to engineering applications? Unfortunately, there is no direct answer to this question. Nevertheless, the following answer may be offered. In this book, the answer has two levels. At one level, engineers may agree that some sort of geometry is required to carry engineering analysis. The degree of geometric detail depends on the analysis procedure that utilizes the geometry. Engineers may also agree that there is no model that is sufficient to study all behavioral aspects of an engineering component or a system. A machine part, for example, can be modeled as a lumped mass rigid body on one occasion or as a distributed mass continuum on another occasion.

model construction. Part IV of the book covers these tools in detail. These tools, in general, help users to manage geometry as well as to avoid unnecessary calculations. For example, typical CAD/CAM systems provide users with possibly three modes to input coordinates: cartesian, cylindrical, or spherical. Each mode has explicit or implicit inputs. Explicit input could be absolute or incremental coordinates. Implicit input involves user digitizes. Another example is the geometric modifiers which automatically identify specific locations such as end- or midpoints of entities that are convenient to access once these entities are created by the system.

Despite its many disadvantages, the major advantage of wireframe modeling is its simplicity to construct. Therefore, it does not require as much computer time and memory as does surface or solid modeling. However, the user or terminal time needed to prepare and/or input data is substantial and increases rapidly with the complexity of the object being modeled. 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 (refer to Chap. 5). Lastly, the CPU time required to retrieve, edit, or update a wireframe model is usually small compared to surface or solid models.

The disadvantages of wireframe models are manifold. Primarily, these 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. To overcome this confusion, lines can be hidden, dashed, or blanked. If done manually, these operations are very tedious, error-prone and can result in "nonsense" objects. Automatic hidden line removal algorithms based on wireframe modeling are usually helpful. Another disadvantage is the lack of visual coherence and information to determine the object profile. The obvious example is the representation of a hole or a curved portion of the object. In most systems, the hole is displayed as two parallel circles separated by the hole length. Some systems may connect a line between the two circles on one side of the hole. In many cases, users add edges of the hole for appearance purposes at the drafting mode or may use a cylindrical surface to represent the hole which introduces problems later on during the model clean-up phase. In adding the edges, inexperienced users tend to attempt to create tangent lines between the hole circles which obviously does not work. Figure 4.1 shows possible cases to display holes and/or curved ends of objects. Representing the intersection of plane faces with cylinders, cylinders with cylinders, or tangent surfaces in general is usually a problem in wireframe modeling and requires user manipulations.

Wireframe models are also considered lengthy or verbose when it comes to the amount of defining data and command sequence required to construct them. For example, compare the creation of a simple box as a wireframe and as a solid. In the latter, the location of one corner, the length, width and height are the required input while in the former the coordinates of at least four corners of one

The tangent vector of the line is given by

$$\mathbf{P}' = \mathbf{P}_2 - \mathbf{P}_1 \tag{4.13}$$

or, in scalar form,

$$\begin{aligned} x' &= x_2 - x_1 \\ y' &= y_2 - y_1 \\ z' &= z_2 - z_1 \end{aligned} \tag{4.14}$$

The independence of the tangent vector from u reflects the constant slope of the straight line. For a two-dimensional line, the known infinite (vertical line) and zero (horizontal line) slope conditions can be generated from Eq. (4.14).

The unit vector $\hat{\mathbf{n}}$ in the direction of the line (Fig. 4.14) is given by

$$\hat{\mathbf{n}} = \frac{\mathbf{P}_2 - \mathbf{P}_1}{L} \tag{4.15}$$

where L is the length of the line:

$$L = |\mathbf{P}_2 - \mathbf{P}_1| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{5.16}$$

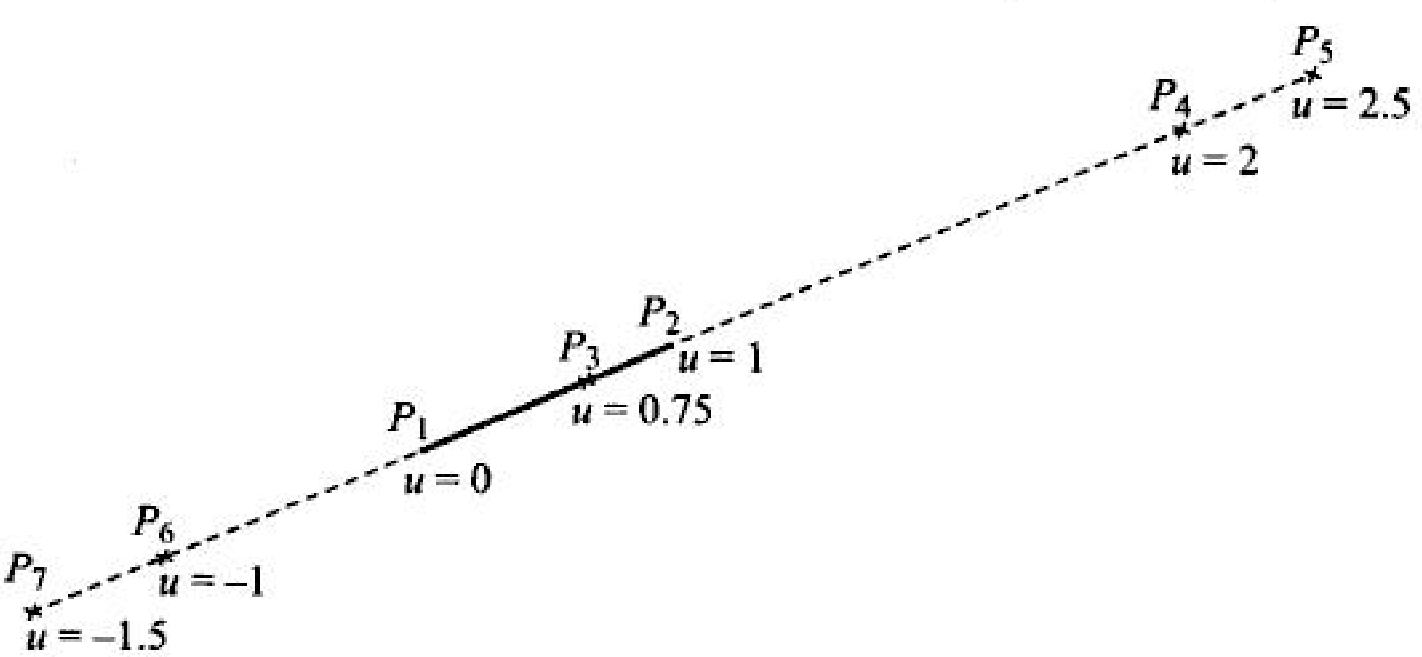


Fig. 4.15 *Locating Points on an Existing Line*

Regardless of the user input to create a line, a line database stores its two endpoints and additional information such as its font, width, color and layer. Equations (4.11) and (4.13) show that the endpoints are enough to provide all geometric properties and characteristics of the line. They are also sufficient to construct and display the line. For reference purposes, CAD/CAM software usually identifies the first point input by the user during line construction as P_1, where $u = 0$. These two equations can be programmed into a subroutine that can reside in a graphics library of the software and which can be invoked, via the user interface, to construct lines. Point commands (or definitions) on most systems provide users with a modifier to specify a u value relative to an entity to generate points on it. In the case of a line, the value is substituted into Eq. (4.11) to find the point coordinates.

2. A line passing through a point P_1 in a direction defined by the unit vector $\hat{\mathbf{n}}$ (Fig. 4.16). Case 1 is considered the basic method to create a line because

where SQRT is the square root. For a two-dimensional case, this equation reduces to

$$D = \frac{1}{L}\,[(x_3 - x_1)(y_2 - y_1) - (y_3 - y_1)(x_2 - x_1)]$$

Example 4.9 Find the unit tangent vector in the direction of a line:

(*a*) Parallel to an existing line.
(*b*) Perpendicular to an existing line.

Solution The conditions of parallelism or perpendicularity of two lines given in the vector algebra reviewed in Sec. 4.5.1 are useful if the vector equations of the two lines exist. If one equation is not available, which is mostly the case in practical problems, the conditions should be reduced to find the unit tangent vector of the missing line in terms of the existing one. Figure 4.20 shows the existing line as L_1 with a known unit tangent vector $\hat{\mathbf{n}}_1$. The unit tangent vector $\hat{\mathbf{n}}_2$ is to be found in terms of $\hat{\mathbf{n}}_1$.

(*a*) For L_1 and L_2 to be parallel (Fig. 4.20*a*),

$$\hat{\mathbf{n}}_2 = \hat{\mathbf{n}}_1$$

or

$$[n_{2x} \quad n_{2y} \quad n_{2z}]^T = [n_{1x} \quad n_{1y} \quad n_{1z}]^T$$

This equation defines an infinite number of lines in an infinite number of planes in space. Additional geometric conditions are required to define a specific line. This equation is equivalent to the condition of the equality of the two slopes of the two lines in the two-dimensional case. Neglecting the Z component, the above equation gives the condition for this case as

$$[n_{2x} \quad n_{2y}]^T = [n_{1x} \quad n_{1y}]T$$

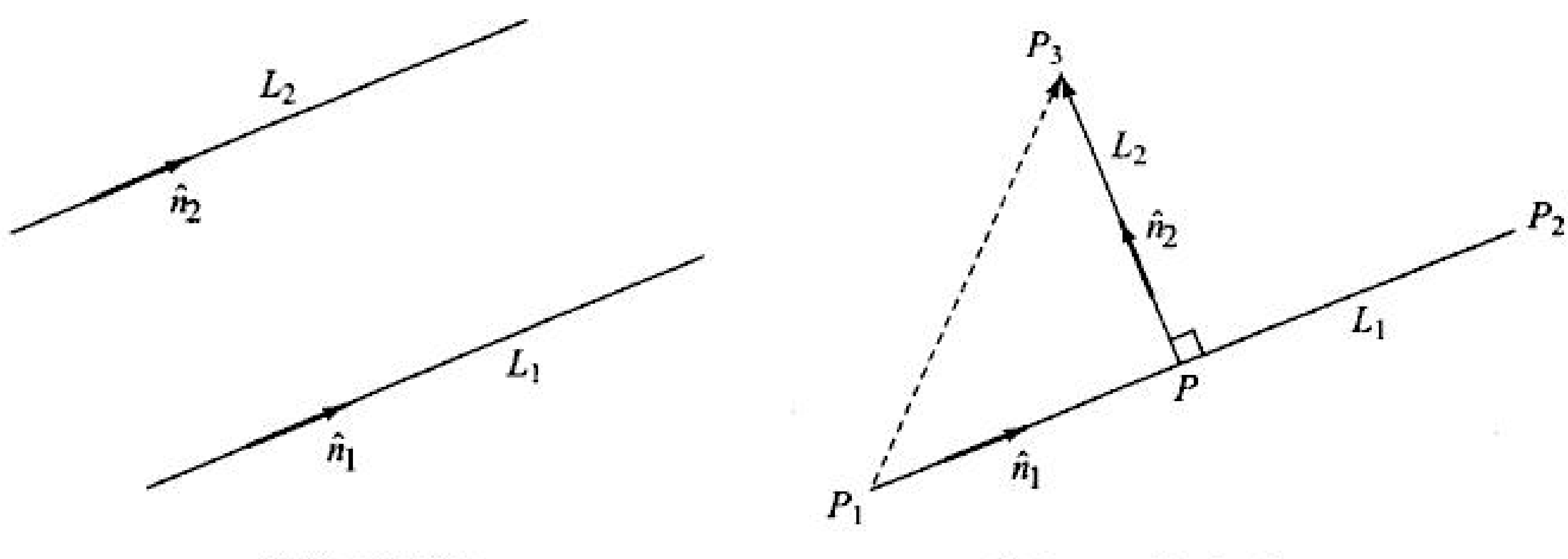
or

$$\frac{n_{2y}}{n_{2x}} = \frac{n_{1y}}{n_{1x}}$$

or

$$m_2 = m_1$$

where m_1 and m_2 are the slopes of the lines.



(a) Parallel lines (*b*) Perpendicular lines

Fig. 4.20 *Unit Tangent Vectors of Various Lines*

Circular arcs are considered a special case of circles. Therefore, all discussions covered here regarding circles can easily be extended to arcs. A circular arc equation can be written as

$$\left.\begin{aligned} x &= x_c + R\cos u \\ y &= y_c + R\sin u \\ z &= z_c \end{aligned}\right\} \quad u_s \le u \le u_e \tag{4.26}$$

where u_s and u_e are the starting and ending angles of the arc respectively. An arc database includes its center and radius, as a circle, as well as its starting and ending angles. Most user inputs offered by software to create arcs are similar to these offered to create circles. In fact, some software packages do not even offer arcs, in which case users have to create circles and then trim them using the proper trimming boundaries. As indicated from Eq. (4.26) and Fig. 4.21, the arc always connects its beginning and ending points in a counterclockwise direction. This rule is usually the default of most CAD/CAM packages when the user input is not sufficient to determine the arc position in space. For example, Fig. 4.22 shows the two possibilities to create an arc given two input points P_1 and P_2 to define its diameter. In this case, the arc is obviously half a circle and P_1 is always its starting point as it is input first by the user.

Following are some examples that show how various geometric data and constraints, which can be thought of as user inputs required by software packages, can be converted to a radius and center before its storage by the software in the corresponding circle database. They also show that three constraints (points and/or tangent vectors) are required to create a circle except for the obvious case of a center and radius. In all these examples, the reader can verify the resulting equations by comparing their results with those of an accessible CAD/CAM software package.

Example 4.10 Find the radius and the center of a circle whose diameter is given by two points.

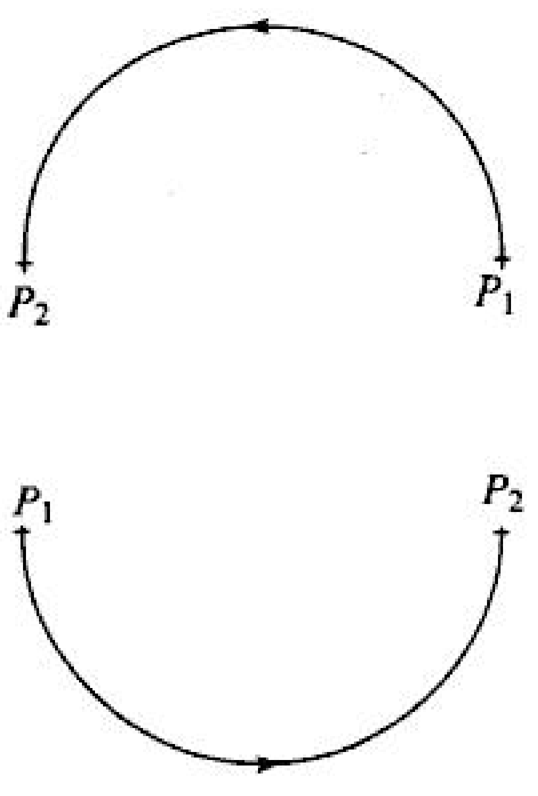


Fig. 4.22 *Arc Creation Following Counterclockwise Direction*

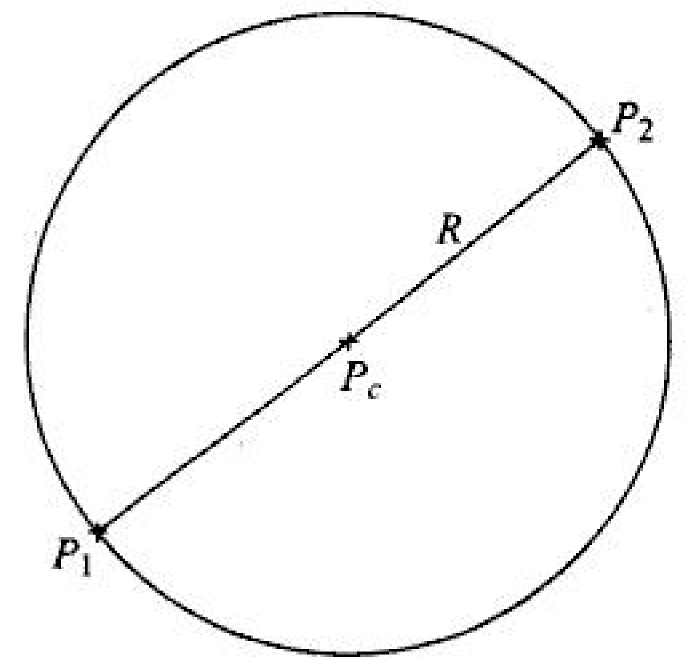


Fig. 4.23 *Circle Defined by Diameter P_1P_2*

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

These equations can be solved for the components n_{hx}, n_{hy} and n_{hz} using the matrix approach utilized in Example 4.11. The unit vector $\hat{\mathbf{n}}_v$ can be found as

$$\hat{\mathbf{n}}_v = \hat{\mathbf{n}}_3 \cdot \hat{\mathbf{n}}_h$$

With $\hat{\mathbf{n}}_h$, $\hat{\mathbf{n}}_v$ *and* $\hat{\mathbf{n}}_3$ known, the orientation of the ellipse is completely defined in space and transformation of points on the ellipse from the WCS system to the MCS can be performed for display and plotting purposes.

Note: case (*b*) of Example 4.13 is a special case of this example.

Example 4.15 Find the tangent to an ellipse from a given point P_1 outside the ellipse.

Solution Two tangents can be drawn to the ellipse from P_1 as shown in Fig. 4.31. Assume the tangency point is P_T. First, transform P_1 from the MCS to the ellipse local WCS system using the equation

$$\mathbf{P}_1 = [T]\mathbf{P}_{1w} \tag{4.49}$$

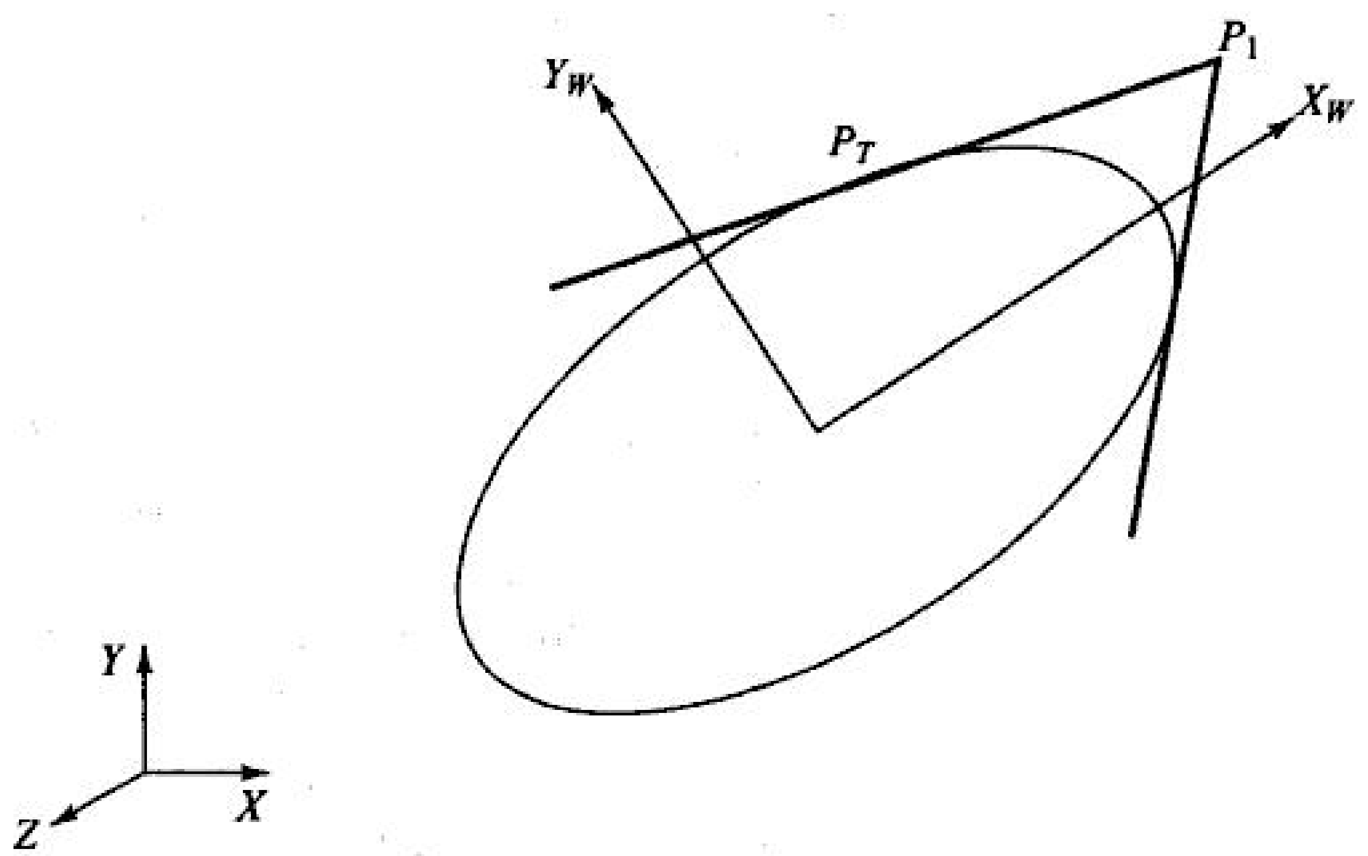


Fig. 4.31 *Tangent to an Ellipse from an Outside Point*

The transformation matrix $[T]$ is known because the orientation of the ellipse is known. P_{1W} holds the local coordinates of P_1 which should be $[x_{1W} \quad y_{1W} \quad 0]$. Therefore:

$$\mathbf{P}_{1W} = [T]^{-1}\mathbf{P}_1$$

The tangent vector to the ellipse is given by

$$\mathbf{P}' = [-A \sin u \quad B \cos u \quad 0]^T$$

At point P_T, this vector becomes

$$\mathbf{P}' = [-A \sin u_T \quad B \cos u_T \quad 0]^T$$

and the slope of the tangent is given by

$$S = -\frac{B \cos u_T}{A \sin u_T}$$

In order to find the coefficients $\mathbf{C}_i$, consider the cubic spline curve with the two endpoints P_0 and P_1 shown in Fig. 4.40. Applying the boundary conditions ($\mathbf{P}_0$, $\mathbf{P}'_0$ at $u = 0$ and $\mathbf{P}_1$, $\mathbf{P}'_1$ at $u = 1$), Eqs. (4.74) and (4.78) give

$$\begin{aligned}\mathbf{P}_0 &= \mathbf{C}_0\\ \mathbf{P}'_0 &= \mathbf{C}_1\\ \mathbf{P}_1 &= \mathbf{C}_3 + \mathbf{C}_2 + \mathbf{C}_1 + \mathbf{C}_0\\ \mathbf{P}'_1 &= 3\mathbf{C}_3 + 2\mathbf{C}_2 + \mathbf{C}_1\end{aligned} \tag{4.79}$$

Solving these four equations simultaneously for the coefficients gives

$$\begin{aligned}\mathbf{C}_0 &= \mathbf{P}_0\\ \mathbf{C}_1 &= \mathbf{P}'_0\\ \mathbf{C}_2 &= 3(\mathbf{P}_1 - \mathbf{P}_0) - 2(\mathbf{P}'_0 - \mathbf{P}'_1)\\ \mathbf{C}_3 &= 2(\mathbf{P}_0 - \mathbf{P}_1) + \mathbf{P}'_0 + \mathbf{P}'_1\end{aligned} \tag{4.80}$$

Substituting Eqs. (4.80) into Eq. (4.76) and rearranging gives

$$\begin{aligned}\mathbf{P}(u) = {} & (2u^3 - 3u^2 + 1)\mathbf{P}_0 + (-2u^3 + 3u^2)\mathbf{P}_1\\ & + (u^3 - 2u^2 + u)\mathbf{P}'_0 + (u^3 - u^2)\mathbf{P}'_1, \quad 0 \le u \le 1\end{aligned} \tag{4.81}$$

$\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}'_0$ and $\mathbf{P}'_1$ are called geometric coefficients. The tangent vector becomes

$$\begin{aligned}\mathbf{P}'(u) = {} & (6u^2 - 6u)\mathbf{P}_0 + (-6u^2 + 6u)\mathbf{P}_1\\ & + (3u^2 - 4u + 1)\mathbf{P}'_0 + (3u^2 - 2u)\mathbf{P}'_1, \quad 0 \le u \le 1\end{aligned} \tag{4.82}$$

The functions of u in Eqs. (4.81) and (4.82) are called blending functions. The first two functions blend $\mathbf{P}_0$ and $\mathbf{P}_1$ and the second two blend $\mathbf{P}'_0$ and $\mathbf{P}'_1$ to produce the left-hand side in each equation.

Equation (4.81) can be written in a matrix form as

$$\mathbf{P}(u) = \mathbf{U}^T[M_H]\mathbf{V}, \quad 0 \le u \le 1 \tag{4.83}$$

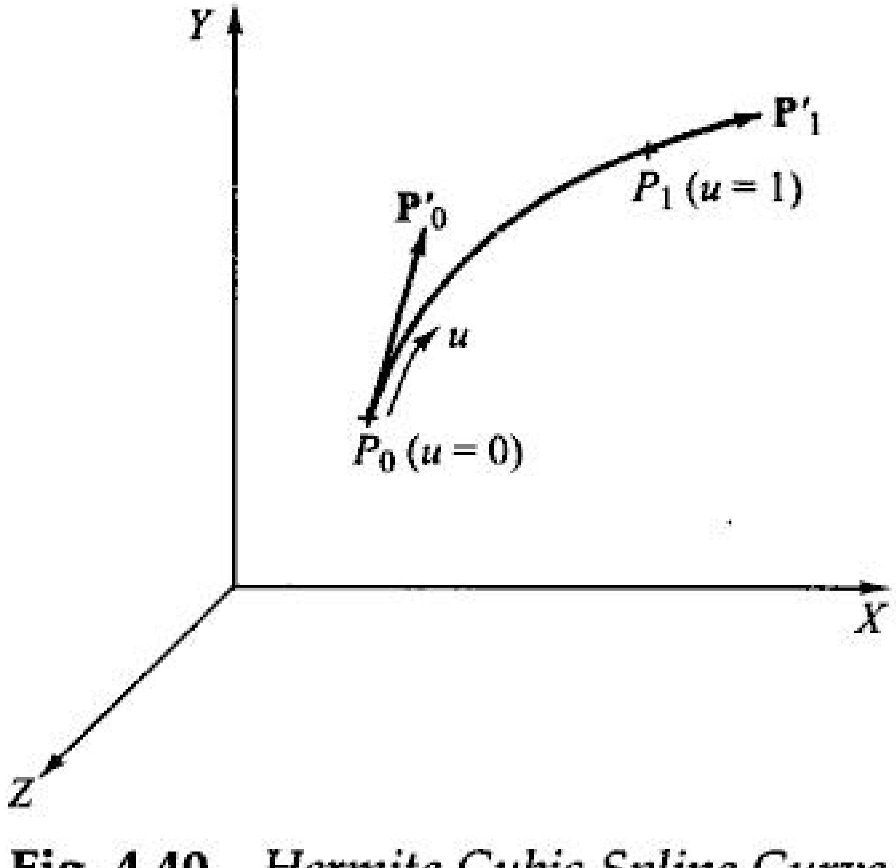


Fig. 4.40 *Hermite Cubic Spline Curve*

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

as it was so chosen arbitrarily for Bezier curves. The B-spline functions have the following properties:

Partition of unity: $\sum_{i=0}^{n} N_{i,k}(u) = 1$

Positivity: $N_{i,k}(u) \geq 0$

Local support: $N_{i,k}(u) = 0 \quad \text{if } u \notin [u_i, u_{i+k+1}]$

Continuity: $N_{i,k}(u)$ is $(k-2)$ times continuously differentiable

The first property ensures that the relationship between the curve and its defining control points is invariant under affine transformations. The second property guarantees that the curve segment lies completely within the convex hull of P_i. The third property indicates that each segment of a B-spline curve is influenced by only k control points or each control point affects only k curve segments. It is useful to notice that the Bernstein polynomial, $B_{i,n}(u)$, has the same first two properties mentioned above.

The B-spline function also has the property of recursion which is defined as

$$N_{i,k}(u) = (u - u_i)\frac{N_{i,k-1}(u)}{u_{i+k-1} - u_i} + (u_{i+k} - u)\frac{N_{i+1,k-1}(u)}{u_{i+k} - u_{i+1}} \tag{4.104}$$

where

$$N_{i,1} = \begin{cases} 1, & u_i \leq u \leq u_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{4.105}$$

Choose 0/0 = 0 if the denominators in Eq. (4.104) become zero. Equation (4.105) shows that $N_{i,1}$ is a unit step function.

Because $N_{i,1}$ is constant for $k = 1$, a general value of k produces a polynomial in u of degree $(k - 1)$ [see Eq. (4.104)] and therefore a curve of order k and degree $(k - 1)$. The u_i are called parametric knots or knot values. These values form a sequence of nondecreasing integers called the knot vector. The values of the u_i depend on whether the B-spline curve is an open (nonperiodic) or closed (periodic) curve. For an open curve, they are given by

$$u_j = \begin{cases} 0, & j < k \\ j - k + 1, & k \leq j \leq n \\ n - k + 2, & j > n \end{cases} \tag{4.106}$$

where

$$0 \leq j \leq n + k \tag{4.107}$$

and the range of u is

$$0 \leq u \leq n - k + 2 \tag{4.108}$$

Relation (4.107) shows that $(n + k + 1)$ knots are needed to create a $(k - 1)$ degree curve defined by $(n + 1)$ control points. These knots are evenly spaced over the range of u with unit separation ($\Delta u = 1$) between noncoincident knots. Multiple (coincident) knots for certain values of u may exist.

While the degree of the resulting B-spline curve is controlled by k, the range of the parameter u as given by Eq. (4.108) implies that there is a limit on k that is determined by the number of the given control points. This limit is found by requiring the upper bound in Eq. (4.108) to be greater than the lower bound for the u range to be valid, that is,

$$n - k + 2 > 0 \tag{4.109}$$

This relation shows that a minimum of two, three and four control points are required to define a linear, quadratic and cubic B-spline curve respectively.

The characteristics of B-spline curves that are useful in design can be summarized as follows:

1. The local control of the curve can be achieved by changing the position of a control point(s), using multiple control points by placing several points at the same location, or by choosing a different degree $(k - 1)$. As mentioned earlier, changing one control point affects only k segments. Figure 4.50 shows the local control for a cubic B-spline curve by moving P_3 to P_3^* and P_3^{**}. The four curve segments surrounding P_3 change only.
2. A nonperiodic B-spline curve passes through the first and last control points P_0 and P_{n+1} and is tangent to the first $(P_1 - P_0)$ and last $(P_{n+1} - P_n)$ segments of the control polygon, similar to the Bezier curve, as shown in Fig. 4.50.

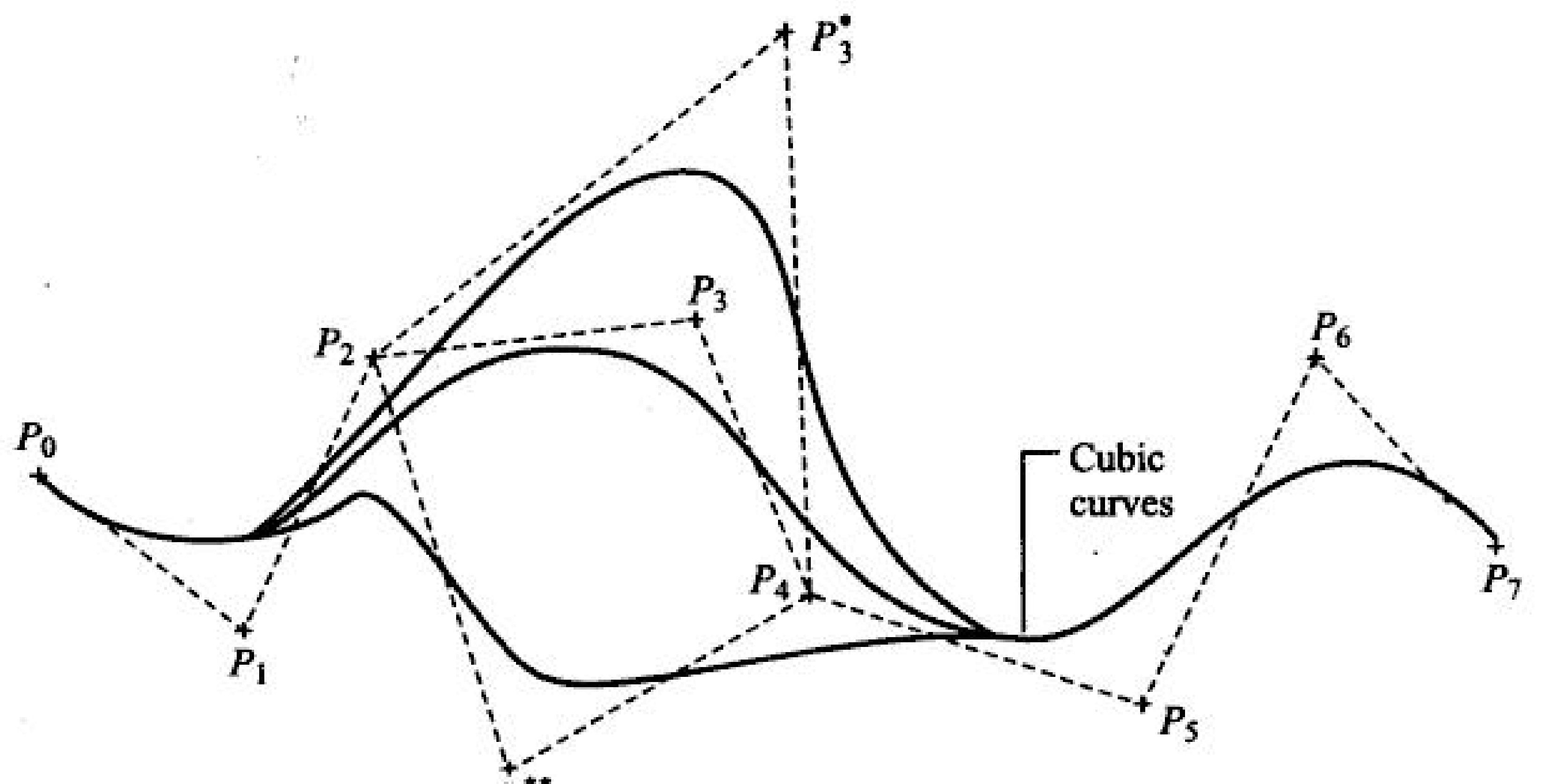


Fig. 4.50 *Local Control of B-spline Curves*

3. Increasing the degree of the curve tightens it. In general, the less the degree, the closer the curve gets to the control points, as shown in Fig. 4.51. When $k = 1$, a zero-degree curve results. The curve then becomes the control points themselves. When $k = 2$, the curve becomes the polygon segments themselves.
4. A second-degree curve is always tangent to the midpoints of all the internal polygon segments (see Fig. 4.51). This is not the case for other degrees.

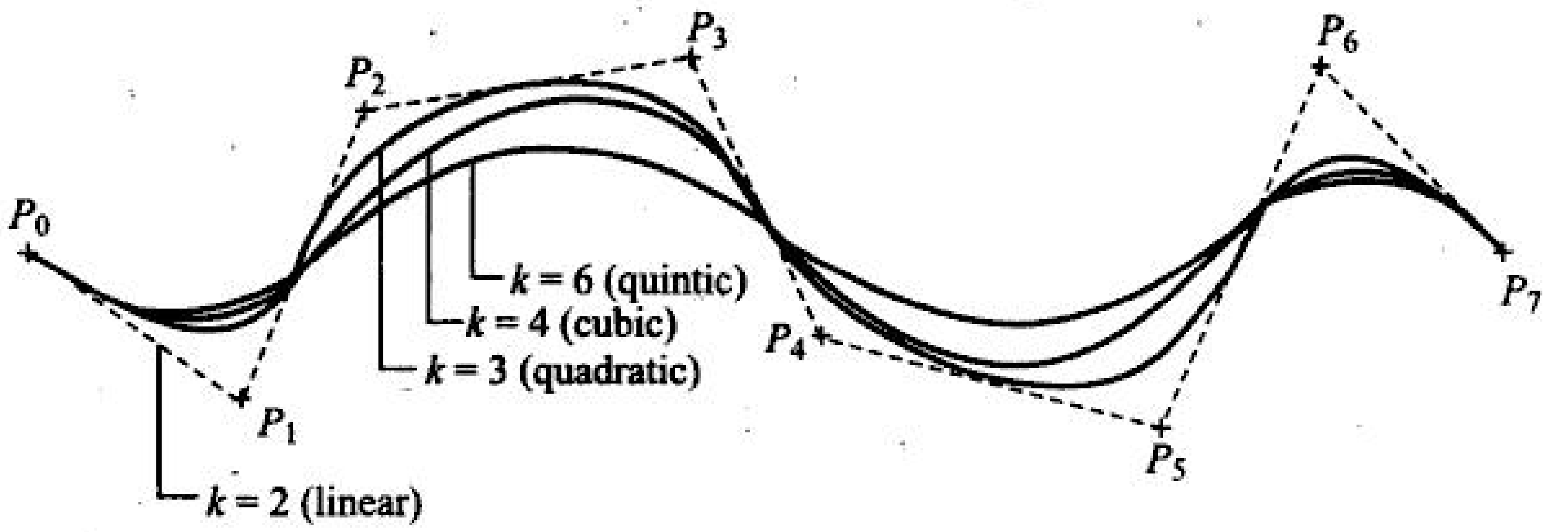
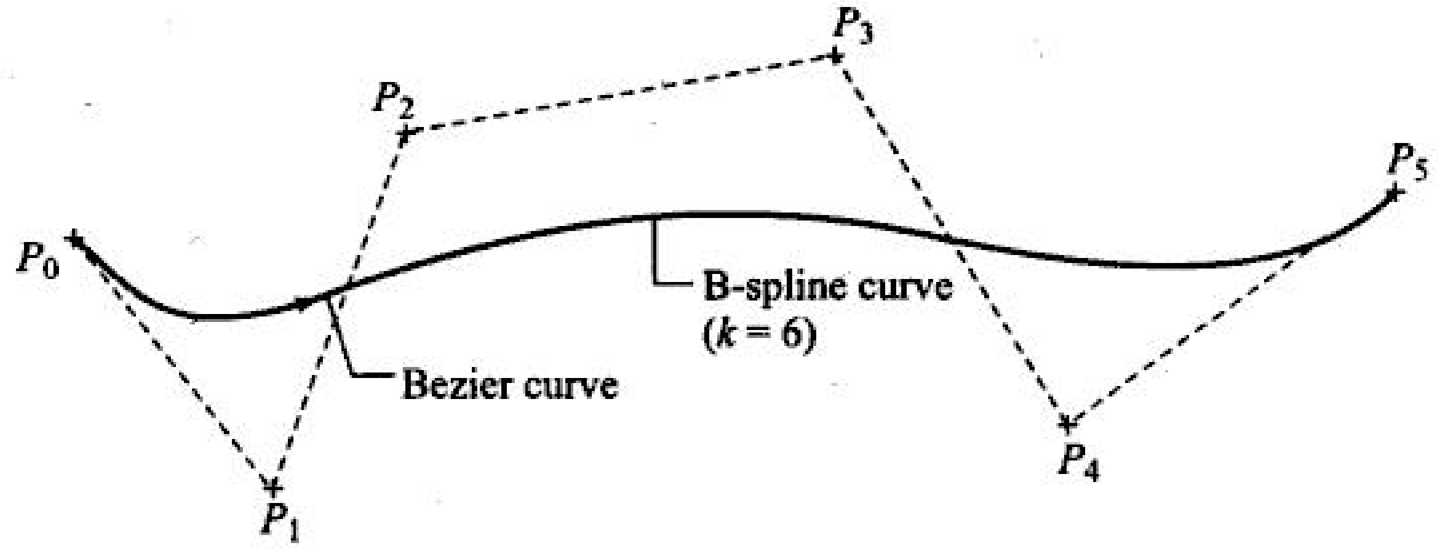
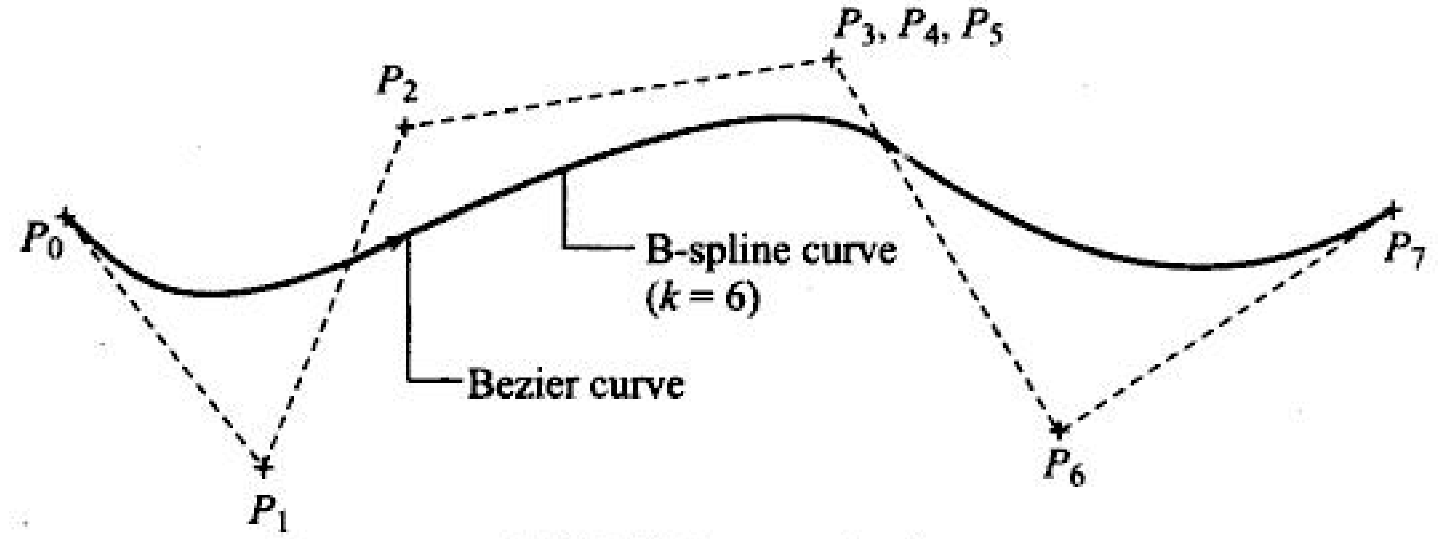


Fig. 4.51 *Effect of the Degree of B-spline Curve on its Shape*

5. If k equals the number of control points $(n + 1)$, then the resulting B-spline curve becomes a Bezier curve (see Fig. 4.52). In this case the range of u becomes zero to one [see Eq. (4.108)] as expected.
6. Multiple control points induce regions of high curvature of a B-spline curve. This is useful when creating sharp corners in the curve (see Fig. 4.53). This effect is equivalent to saying that the curve is pulled more towards a control point by increasing its multiplicity.
7. Increasing the degree of the curve makes it more difficult to control and to calculate accurately. Therefore, a cubic B-spline is sufficient for a large number of applications.



(*a*) No multiple control points



(*b*) Multiple control points

Fig. 4.52 *Identical B-spline and Bezier Curves*

where α_1 and α_2 are constants and **T** is the common unit tangent vector at the joint. It has already been shown how to use curve characteristics for blending purposes as in the case with Bezier curves. Another example is the blending of a Bezier curve and an open B-spline curve. For slope continuity at the joint, the last segment of the control polygon of the former and the first segment of the control polygon of the latter must be colinear. .

The third useful class of continuity is if the curvature (C^2 continuity) is to be continuous at the joint in addition to position and slope. To achieve curvature continuity is less straightforward and requires the binormal vector to a curve at a point. Figure 4.56 shows the tangent unit vector **T**, the normal unit vector **N**, the center of curvature O and the radius of curvature ρ at point P on a curve segment. The curvature at P is defined as $1/\rho$. The binormal vector **B** is defined as

$$\mathbf{B} = \mathbf{T} \times \mathbf{N} \tag{4.126}$$

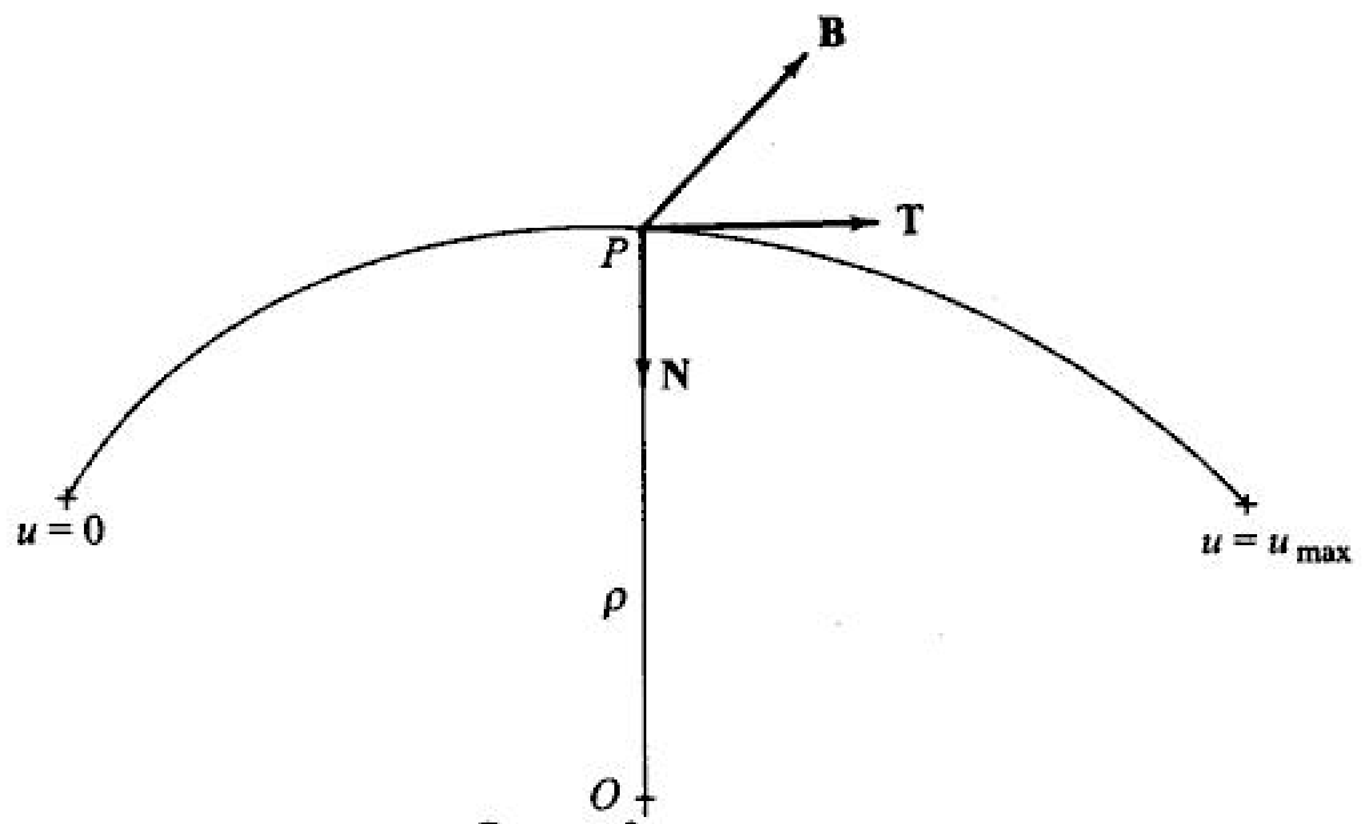


Fig. 4.56 *Binormal Vector to a Curve*

The curvature is related to the curve derivatives through the vector **B** by the following equation:

$$\frac{1}{\rho}\mathbf{B} = \frac{\mathbf{P}' \times \mathbf{P}''}{|\mathbf{P}'|^3} \tag{4.127}$$

where $\mathbf{P}''$ is the second derivative with respect to the parameter u. The following condition can then be written for curvature continuity at the joint:

$$\frac{\mathbf{P}_1'(a) \times \mathbf{P}_1''(a)}{|\mathbf{P}_1'(a)|^3} = \frac{\mathbf{P}_2'(0) \times \mathbf{P}_2''(0)}{|\mathbf{P}_2'(0)|^3} \tag{4.128}$$

Substituting Eqs. (4.125) into the above equation gives

$$\mathbf{T} \times \mathbf{P}_1''(a) = \left(\frac{\alpha_1}{\alpha_2}\right)^2 \mathbf{T} \times \mathbf{P}_2''(0) \tag{4.129}$$

4.7.7 Transformation

Manipulation or transformation of geometric entities during model construction or creating the model database offers a distinct advantage of CAD/CAM technology over traditional drafting methods. With transformation techniques, the designer can project, translate, rotate, mirror and scale various entities. Section 4.3 shows how two-and-a-half-dimensional objects can be constructed easily using a "project" command. It also shows how the "mirror" command helps construct symmetric objects. Transformation is also useful in studying the motion of mechanisms, robots and other objects in space. Animation techniques (Chap. 12) may be based on transforming an object into various positions and then replaying these positions continuously.

Simple transformations such as those mentioned above are usually referred to as rigid-body transformations. They can be directly applied to parametric representations of geometric models. To translate a model of a car, all points, lines, curves and surfaces forming the model must be translated.

Homogeneous transformation, as discussed in Chap. 8, offers a concise matrix form to perform all rigid-body transformations as matrix multiplications, which is a desired feature from the software development point of view. Equation (3.3) gives the general homogeneous transformation matrix $[T]$. The proper choice of the elements of this matrix produces the various rigid-body transformations.

One of the main characteristics of rigid-body transformations is that geometric properties of curves, surfaces and solids are invariant under these transformations. For example, originally parallel or perpendicular straight lines remain so after transformations. Intersection points of curves are transformed into the new intersection points, that is, one-to-one transformation. Chapter 9 discusses in more detail the subject of geometric transformation.

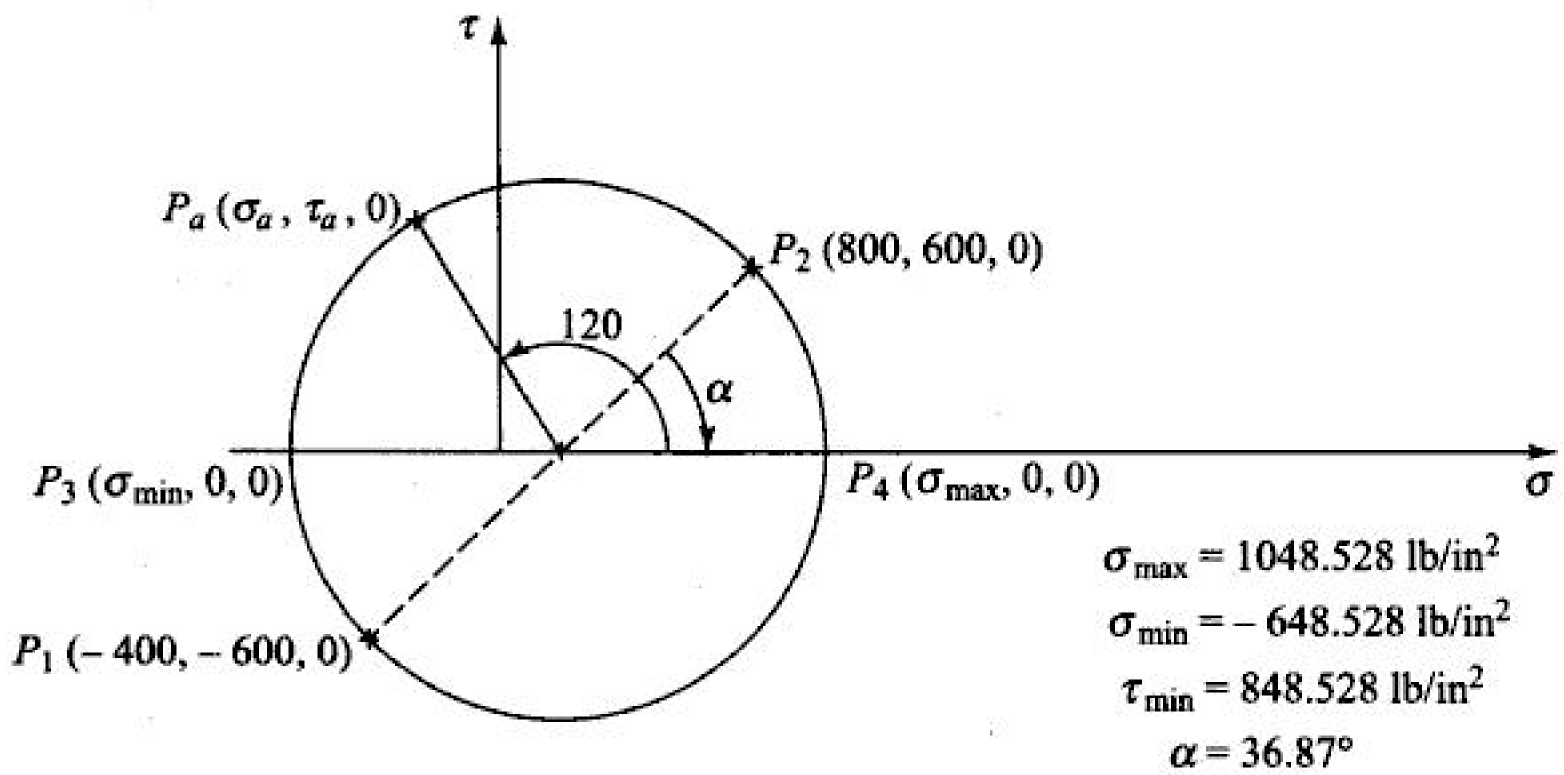
4.8 DESIGN AND ENGINEERING APPLICATIONS

This section presents some examples that show how theories covered in this chapter are applied to design and engineering problems. Consequently, it shows how existing CAD/CAM systems can be stretched beyond just using them for drafting, geometric modeling and beyond what application modules of these systems offer. The reader is advised to work these examples on an actual CAD/CAM system.

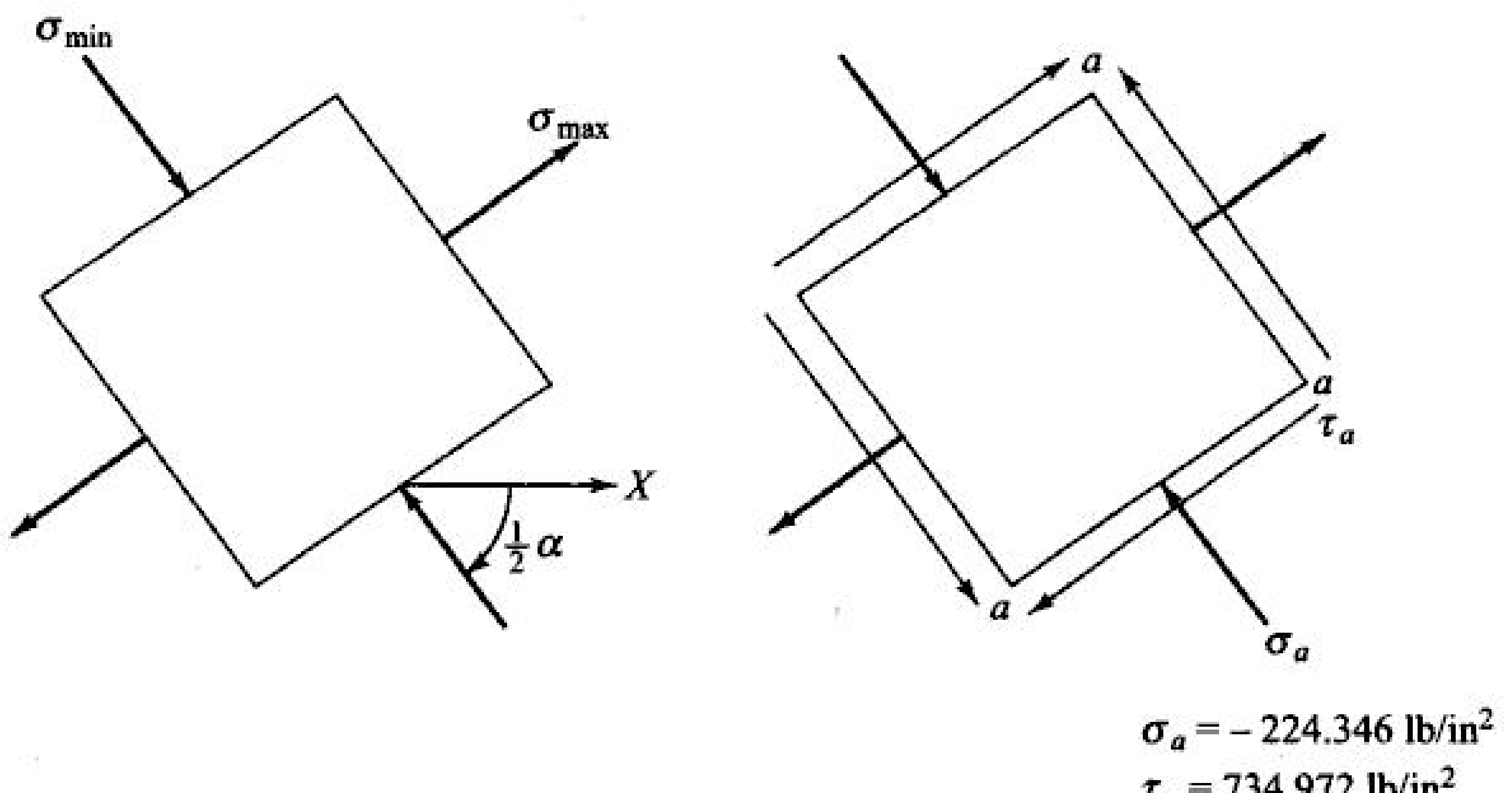
Example 4.25 For the state of plane stress shown in Fig. 4.61, determine:

(*a*) The principal stresses.
(*b*) The state of stress exerted on plane *a-a*.

Solution The solution of this typical problem is to use Mohr's circle which is a graphical method. The centralization of CAD/CAM databases can be used to simplify the use of the method. The designer follows the part setup procedure covered in Chapter 3 and utilizes the default view and construction plane. Then two lines are created, one horizontal and one vertical, passing through the origin of the MCS. These form the σ and τ axes (see Fig. 4.62). The stresses at the X and Y planes can be



(*a*) Mohr's circle



(*b*) Stress values

Fig. 4.62 *Stress Calculations via Mohr's Circle*

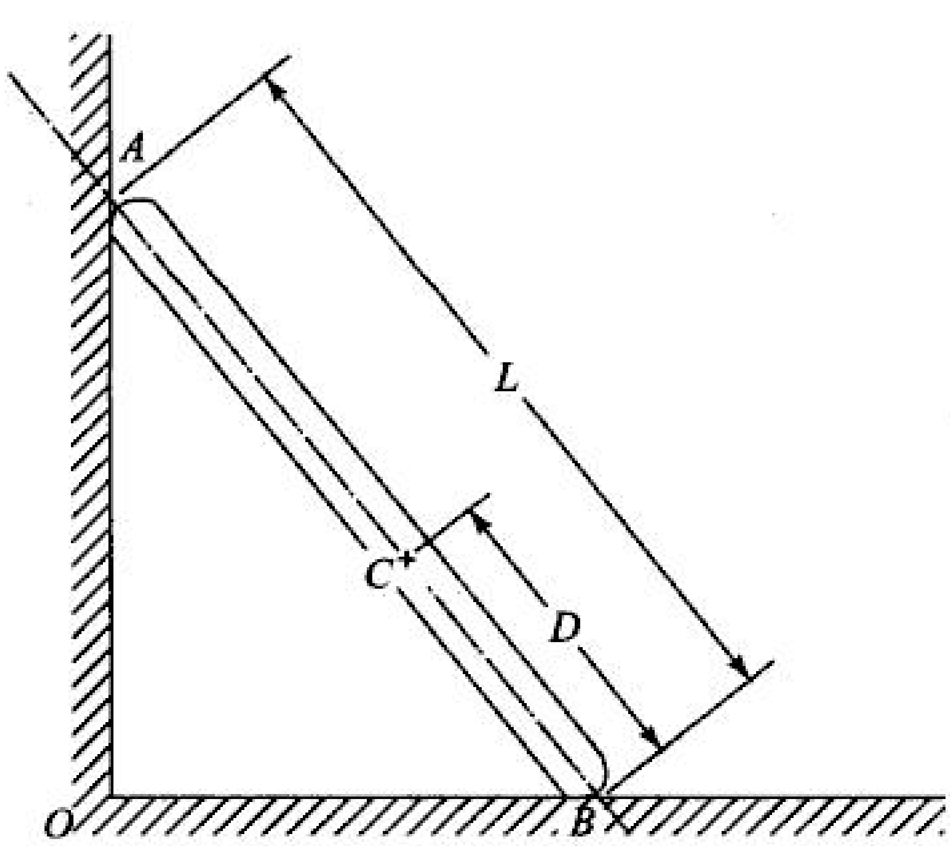
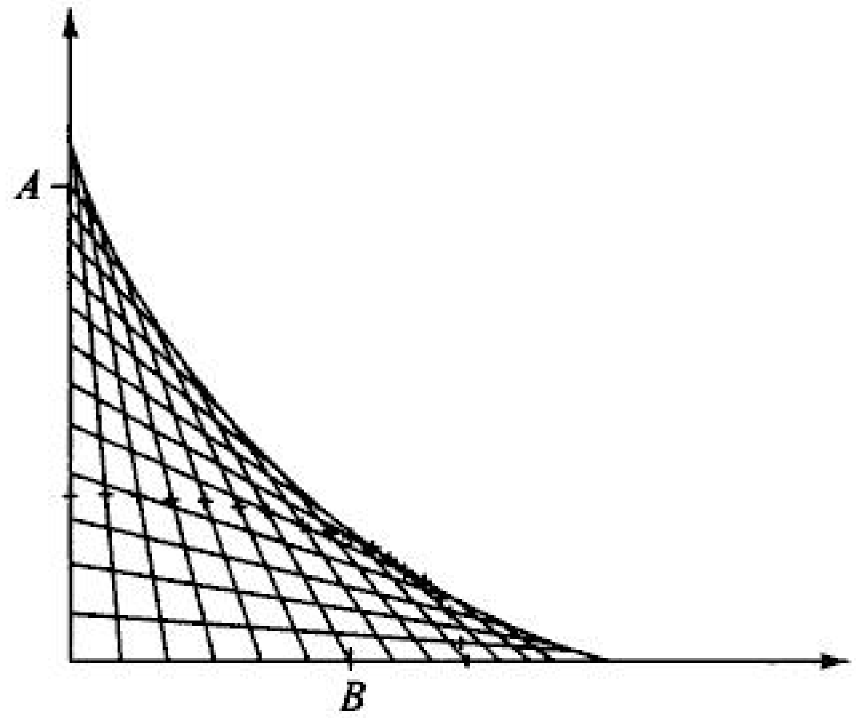


Fig. 4.63 *Bar AB in Plane Motion*



(a) Generating points on the locus

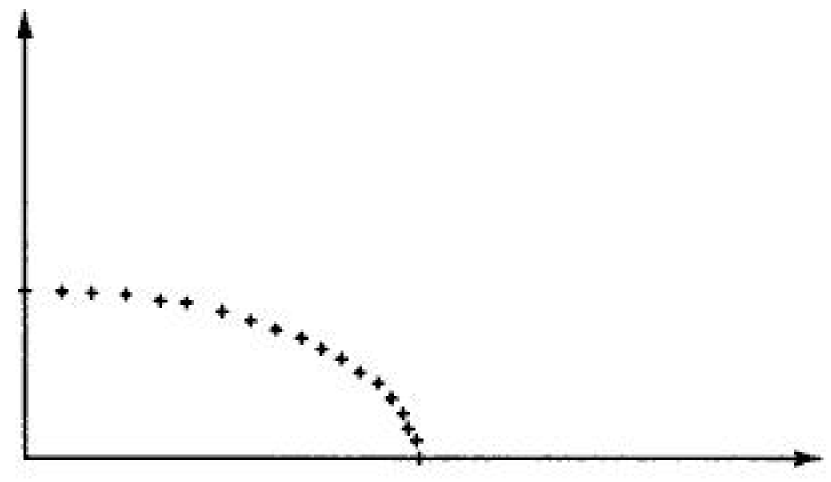

(b) Locus of point C

Fig. 4.64 *Locus of Point C on Bar AB*

The extreme positions of point *C* are at distances *D* and *L* — *D* from point *O* when the bar *AB* coincides with the vertical and horizontal walls respectively. To find any other point on the locus, let us start from the vertical position of *AB* and increment its motion until it becomes horizontal. Thus, we start when point *A* is at the location (0, *L*, 0). Assuming that ten points are enough to generate the locus, point *A* has the coordinates $(0, L - m\,\Delta L, 0)$ where $\Delta L = L/10$ and $0 < m < 10$. To find point *C* for any position of the bar, create a circle with center at *A* and radius equal to *L*. The intersection point between the circle and the horizontal line gives point *B* and, therefore, the orientation of the bar. Point *C* can be located on *AB* using a point command with a parameter *u* value equal to *D/L* or (1 – *D/L*) depending on whether point *B* or *A* is input first in the line command used to create the line. The algorithm to generate points on the locus can therefore be written as

LOOP C1 = circle with P_c = A, R = L
L1 = Line connecting P_0 = A and P_1 = Intersection of C1 and horizontal line
C = point at u = 1 – D/L
A = point at (0, L – m ΔL, 0)
when m is equal to 10 exit
Go to LOOP

Once all the points are generated, they are connected with a B-spline curve via a B-spline command. The locus is shown in Fig. 4.64. To facilitate the management of all graphics entities during construction, it is recommended that all possible aids such as layers, colors and fonts be used (see Chap. 10). This method lends itself to macro programming which is useful in parametric design studies.

If point *C* is the center of *AB*, the locus becomes a circle with center at *O* and radius equal to the distance *OC*. Indeed, the locus of *C* is an ellipse in general with center at *O* and major and minor axes of lengths equal to (*L* – *D*) and *D* if $D < L/2$ and vice versa if $D > L/2$. To prove this, consider Fig. 4.65, which gives the following coordinates of point *C*:

$$x = (L - D)\cos\theta \quad y = D\sin\theta \quad z = 0$$

which is an equation of an ellipse. If $D = L/2$, an equation of a circle results.

Example 4.28 A four-bar mechanism is shown Fig. 4.67. The input angular velocity of the link AB is ω_1 = 200 r/min clockwise. Point E, the center of the link CB, is connected to a valve that is not shown in the figure. The mechanism is to be redesigned such that:

Design criterion. The maximum linear velocity of point E, $v_{E,\max}$, must be greater than its current value by at least 5 inches/s.

Design constraints. (1) Only L and LL lengths can change and (2) AB must rotate the full 360°.

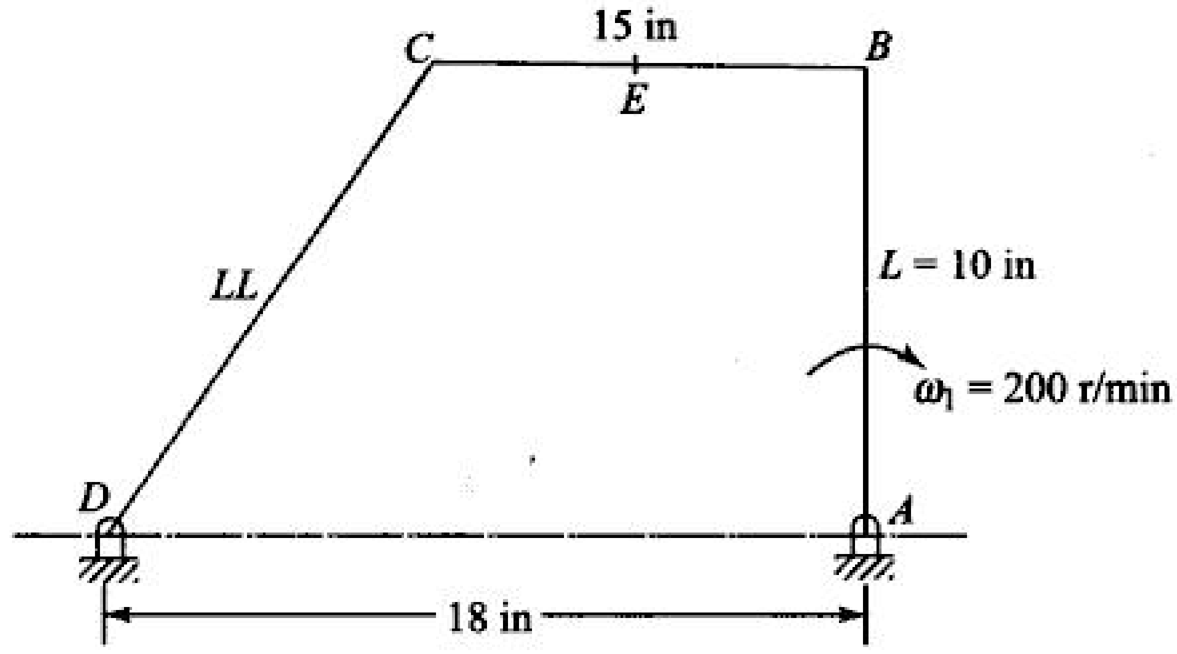


Fig. 4.67 *Four-bar Mechanism*

Solution The solution to this design problem requires the velocity analysis of mechanisms. Either the relative velocity or instantaneous center concept can be utilized. The latter is used here because it lends itself to CAD/CAM techniques. The velocity analysis is shown in Fig. 4.68. The instantaneous center of the mechanism is the intersection point T of links AB and CD for any configuration of the mechanism. The velocity analysis gives

$$\begin{aligned} v_B &= \omega_1 L = \omega_2 L_1 \\ \omega_2 &= \omega_1 L/L_1 \\ v_E &= \omega_2 L_2 = \omega_1 LL_2/L_1 \end{aligned} \tag{4.140}$$

The last equation gives the velocity of point E in terms of the input velocity and the lengths L, L_1 and L_2. To obtain $v_{E,\max}$, rotate the link AB around point A an angle θ, construct the mechanism in the new position, find L_1 and L_2 and substitute in Eqs. (4.140) to find v_E. Repeat for the admissible range of the angle θ. Connect the resulting points [each point has coordinates (θ, v_E, 0)] with a B-spline curve. Find $v_{E,\max}$ from the curve. To construct the mechanism at any angle θ, rotate AB about A, the required angle. This defines point B. Point C is the intersection of two circles. The first has a center at B and a radius of 15 in and the second has a center at D and a radius equal to LL. Thus, the admissible range of angle of rotation (θ) of link AB for a certain set of dimensions is found when the above two circles do not intersect.

With the above strategy in mind, the designer can choose various values for L and LL in an attempt to achieve the design goal and the given design constraints. The major commands needed are "rotate, measure distance, B-spline" commands in addition to layer and utility (delete. . .) commands.

4.8 Find the normal vector to a cubic spline curve at any of its points.

4.9 For a cubic Bezier curve, carry a similar matrix formulation to a cubic spline. Compare $[M_B]$ and V for the two curves.

4.10 Find the condition that a cubic Bezier curve degenerates to a straight line connecting $\mathbf{P}_0$ and $\mathbf{P}_3$.

4.11 Derive a method by which you can force a Bezier curve to pass through a given point in addition to the starting and ending points of its polygon. Achieve that by changing the position of only one control point, say $\mathbf{P}_2$.

4.12 Investigate the statement "each segment of a B-spline curve is influenced by only k control points or each control point affects only k curve segments." Use $n = 3$, $k = 2, 3, 4$.

4.13 Find the equation of an open quadratic B-spline curve defined by five control points.

4.14 For an open cubic B-spline curve defined by n control points, carry a similar matrix formulation to a cubic spline. Compare $[M_S]$ and V with those of the cubic spline and Bezier curves.

Hint: Start with $n = 5$, that is, six control points and then generalize the resulting $[M_S]$.

4.15 Given a point Q and a parametric curve in the cartesian space, find the closest point P on the curve to Q.

Hint: Find P such that $(Q - P)$ is perpendicular to the tangent vector.

4.16 A cubic Bezier curve is to be divided by a designer into two segments. Find the modified polygon points for each segment.

4.17 Explain why parametric representations have proved popular in computational geometry.

4.18 Briefly explain the advantages and disadvantages of wire frame modeling.

4.19 Why is 3D-modeling important? Explain the limitations of wire-frame modeling over solid modeling.

4.20 Write down the formulae for the Bezier-Bernstein blending functions for a five point curve.

4.21 A cubic Bezier curve is defined by the points (1, 1), (2, 3), (4, 4) and (6, 1). Calculate the coordinates of the parametric midpoint of this curve and verify that its gradient (dy/dx) is 1/7 at this point. Use this information to sketch the curve.

4.22 Use a CAD system to demonstrate the following curve features.
- Local modification of B-spline curve
- Global modification of cubic spline curve and Bezier curve

4.23 Briefly describe various curve manipulations.

Part 3: Laboratory

4.26 Obtain the three orthographic views (front, top and right side) and a perspective view looking along the Z axis at a distance 20 inches from the parts shown in Fig. P4.26 (all dimensions in inches). Obtain final drawings of these views. Follow the model clean-up and documentation procedure on your particular CAD/CAM system. Obtain the standard six isometric views of each model. Clean up each view.

4.27 Using your CAD/CAM system, investigate the line, circle, ellipse, parabola, conics, cubic spline, Bezier curve and B-spline curve commands and their related modifiers. Relate these modifiers to their theoretical background covered in this chapter.

4.28 Choose a mechanical element such as a gear and generate its geometric model.

Groover, M. P. and E. W. Zimmers: *CAD/CAM; Computer-Aided Design and Manufacturing*, Prentice-Hall, Englewood Cliffs, N.J., 1984.

Hagen, H.: "Geometric Spline Curves," *CAGD J.*, vol. 2, pp. 223–227, 1985.

Haug, E. J. and J. S. Arora: *Applied Optimum Design,* Jon Wiley, New York, 1979.

Lasser, D.: "Bernstein-Bezier Representation of Volumes," *CAGD J.*, vol. 2, pp. 145–149, 1985.

Lee, E. T. Y.: "Some Remarks Concerning B-Splines," *CAGD J.*, vol. 2, pp. 307–311, 1985.

Luzadder, W. J.: *Innovative Design with an Introduction to Design Graphics,* Prentice-Hall, Englewood Cliffs, N.Y., 1975.

Maccallum, K. J. and J. M. Zhang: "Curve Smoothing Techniques Using B-Splines," *The Computer J.*, vol. 29, no. 6, pp. 564–571, 1986.

Mortenson, M. E.: *Geometric Modeling,* John Wiley, New York, 1985.

Newman, W. M and R. F. Sproull: *Principles of Interactive Computer Graphics,* 2d ed., McGraw-Hill, New York, 1979.

Nielson, G. M.: "Rectangular Splines," *IEEE CG&A*, pp. 35–40, February 1986.

Patterson, R. R.: "Projective Transformation of the Parameter of a Bernstein-Bezier Curve," *ACM Trans, on Graphics,* vol. 4, no. 4, pp. 276–290, 1985.

Piegl, L.: "Recursive Algorithms for the Representation of Parametric Curves and Surfaces," *Computer Aided Des. J.,* vol. 17, no. 5, pp. 225–229, 1985.

Piegl, L.: "Infinite Control Points, A Method for Representing Surfaces of Revolution Using Boundary Data," *IEEE CG&A,* pp. 45–55, March 1987.

Piegl, L.: "Interactive Data Interpolation by Rational Bezier Curves," *IEEE CG&A,* pp. 45–58, April 1987.

Rogers, D. F. and J. A. Adams: *Mathematical Elements for Computer Graphics,* McGraw-Hill, New York, 1976.

Sablonniere, P.: "Bernstein-Bezier Methods for the Construction of Bivariate Spline Approximations," *CAGD J.*, vol. 2, pp. 29–36, 1985.

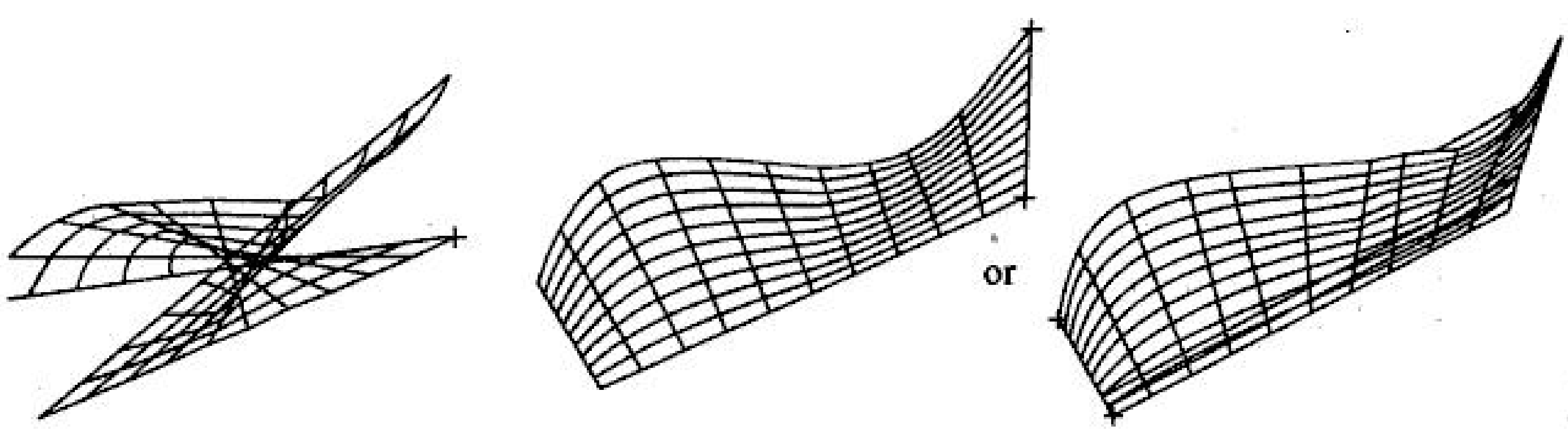
Sederberg, T. W.: "Degenerate Parametric Curves," *CAGD J.*, vol. 1, pp. 301–307, 1984.

Vandoni, C. E. (Ed.): *Eurographics '85*, Elsevier Science, New York, 1985.

Wilson, P. R.: "Conic Representations for Shape Description," *IEEE CG&A,* pp. 23–30, April 1987.

rails are digitized near the wrong ends. The +'s in the figure indicates the digitized locations. Another practical tip in constructing surface models is the change in mesh size of a surface entity. The most obvious way is to delete the surface entity and then reconstruct it with the desired mesh size. A better solution is to simply change the size of the mesh and then regenerate the surface display. Figure 5.2 shows surfaces of revolutions with a mesh size of 4 × 6 and Fig. 5.3 shows the regeneration of the surfaces with a 20 × 20 mesh size. It should be mentioned here that the finer the mesh size of surface entities in a model, the longer the CPU time to construct the entities and to update the graphics display and the longer it takes to plot the surface model. Finally, some CAD/CAM systems do not permit their users to delete wireframe entities used to create surface entities unless the latter are deleted first.

Surface models have considerable advantages over wireframe models. They are less ambiguous. They provide hidden line and surface algorithms to add realism to the displayed geometry. Shading algorithms are only available for surface and solid models. From an application point of view, surface models can be utilized in volume and mass property calculations, finite element modeling, NC path generation, cross sectioning and interference detections.



(*a*) Wrong digitized locations (*b*) Proper digitized locations

Fig. 5.1 *Construction of Improper and Proper Ruled Surfaces*

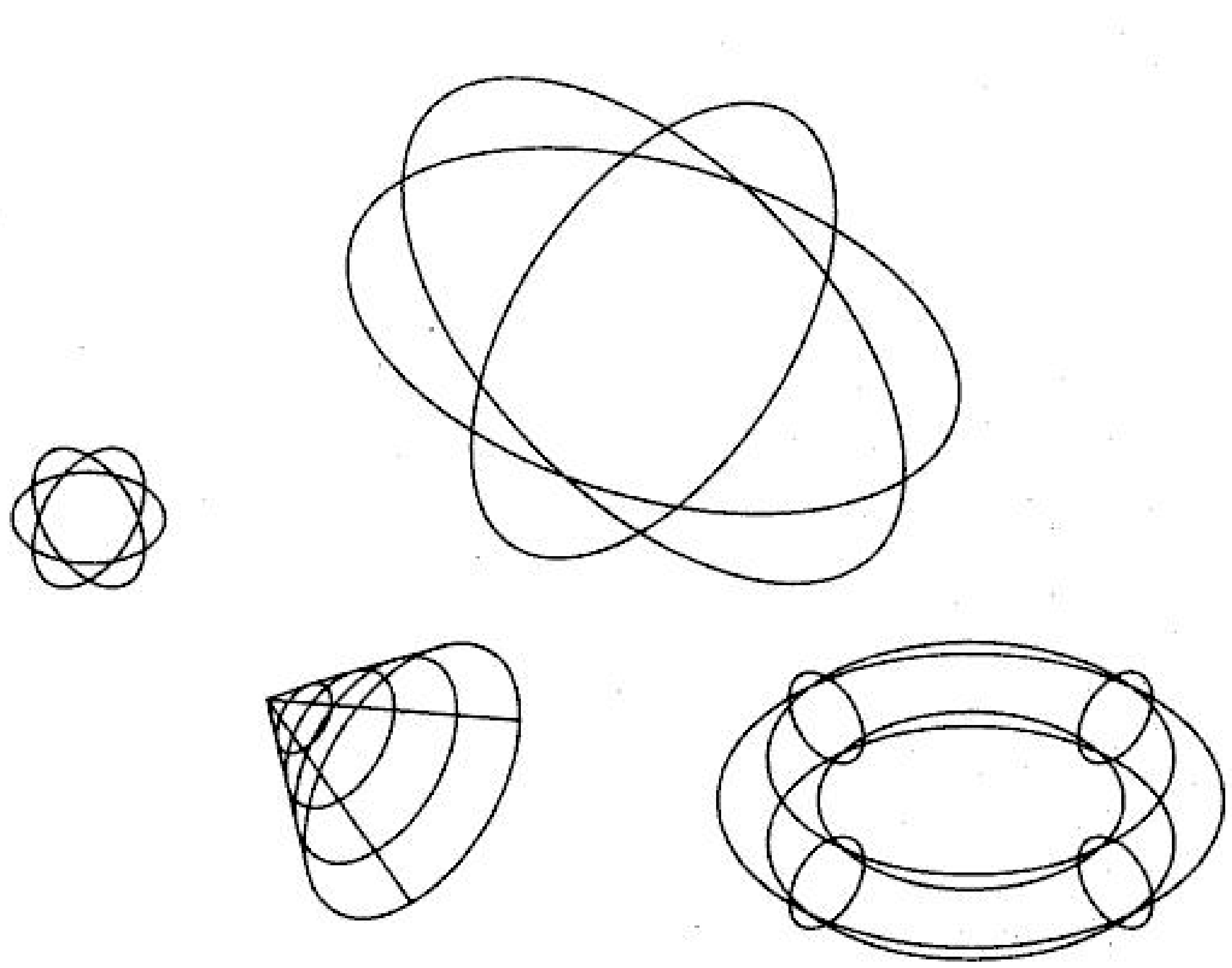


Fig. 5.2 *Surfaces of Revolution with a 4 × 4 Mesh Size*

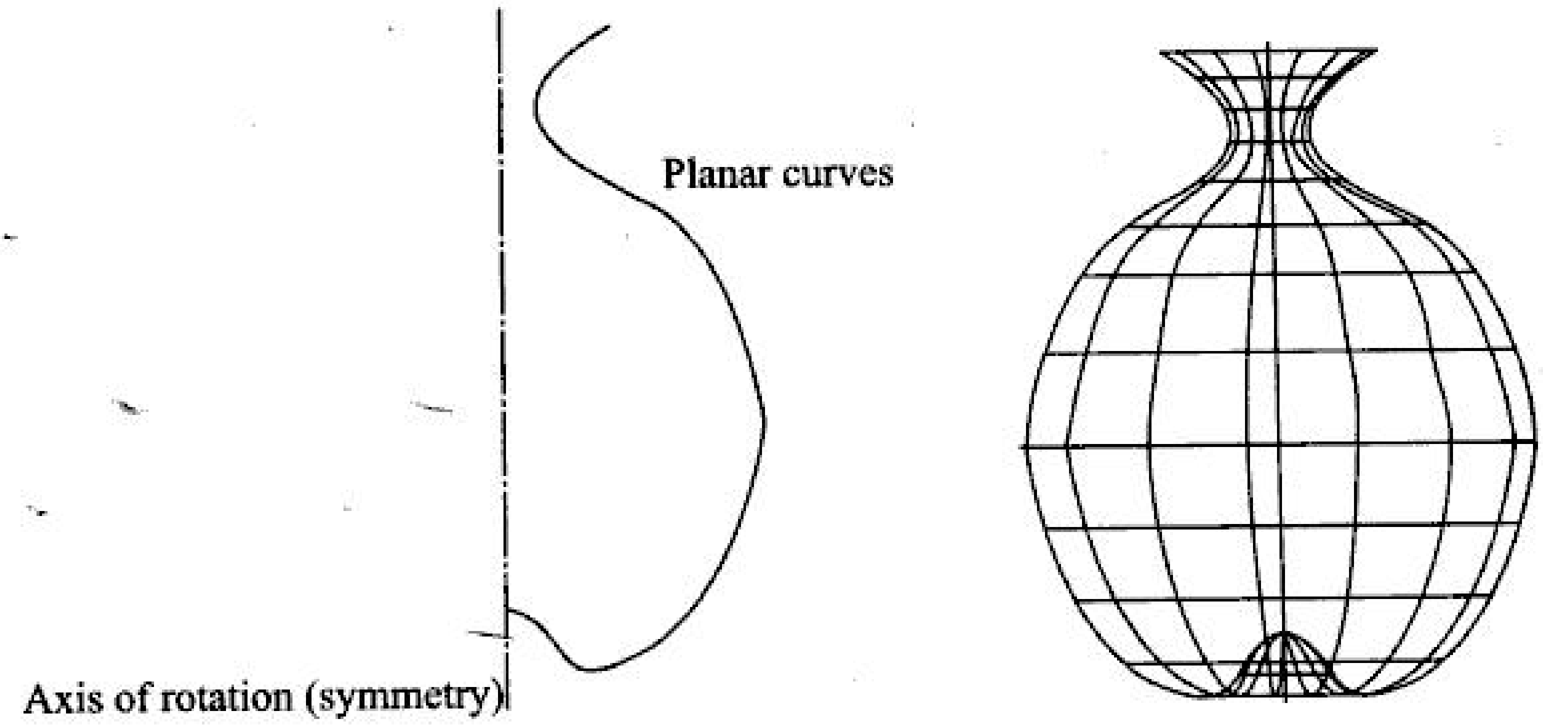


Fig. 5.6 *Surface of Revolution*

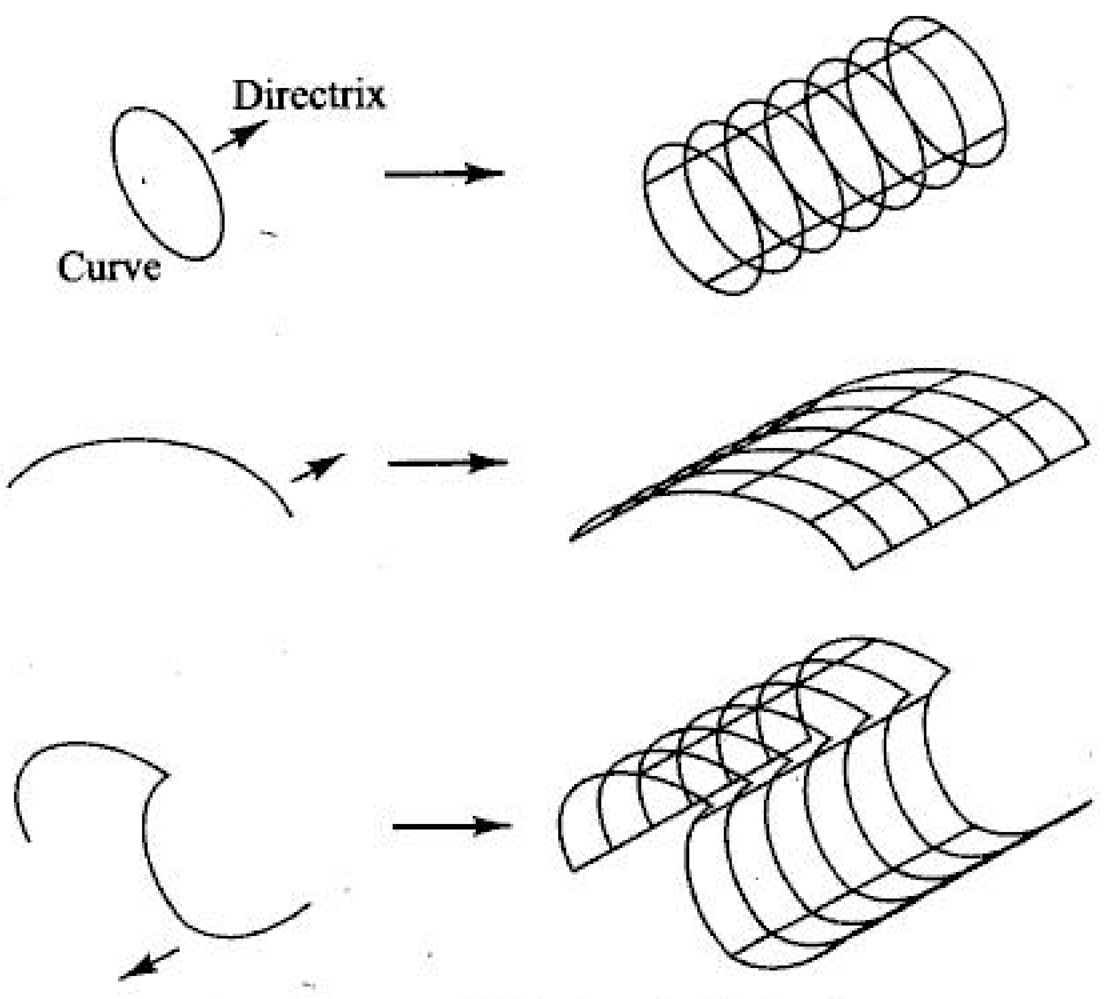


Fig. 5.7 *Tabulated Cylinder*

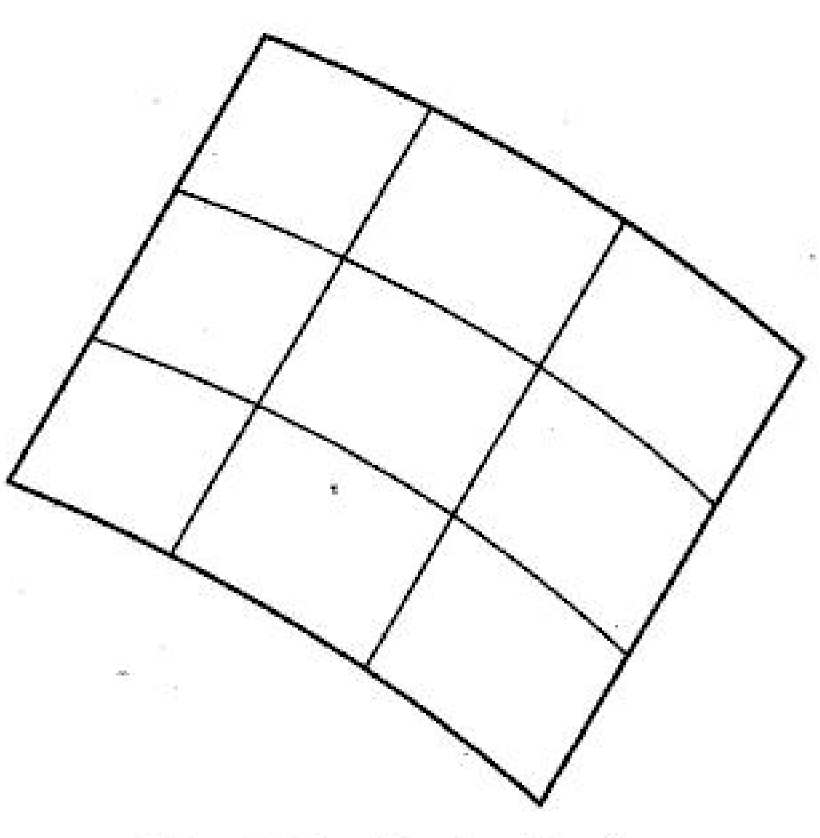
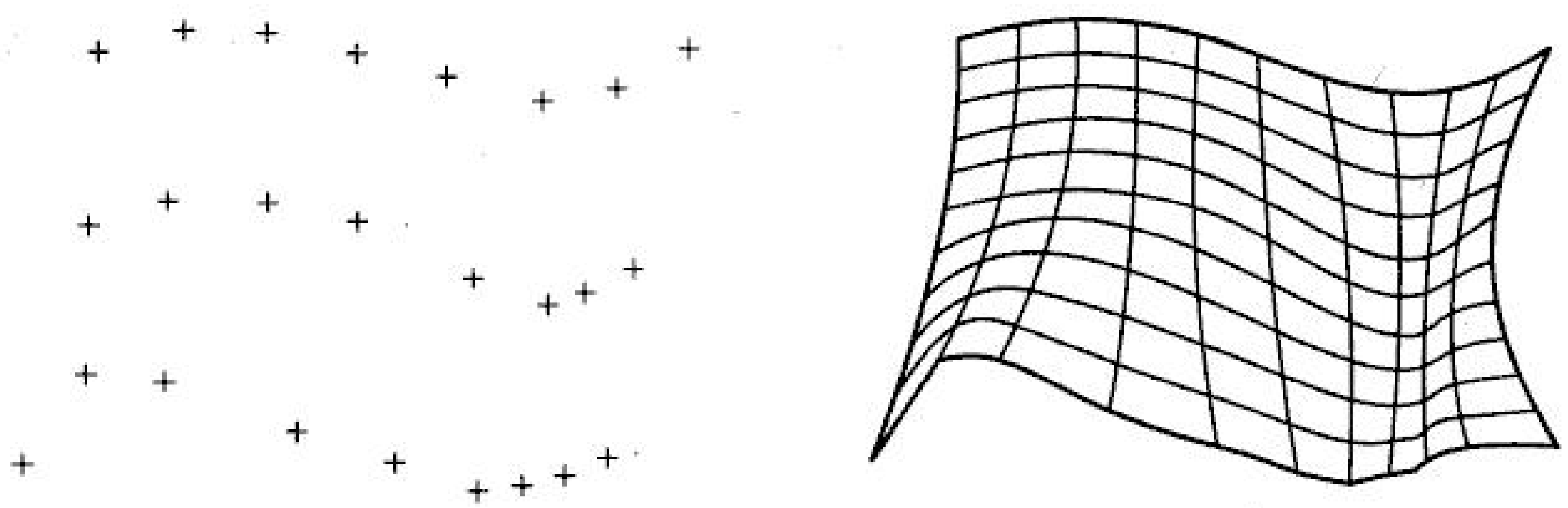


Fig. 5.8 *Bezier Surface*



(a) Data points (b) B-spline surface

Fig. 5.9 *B-spline Surface*

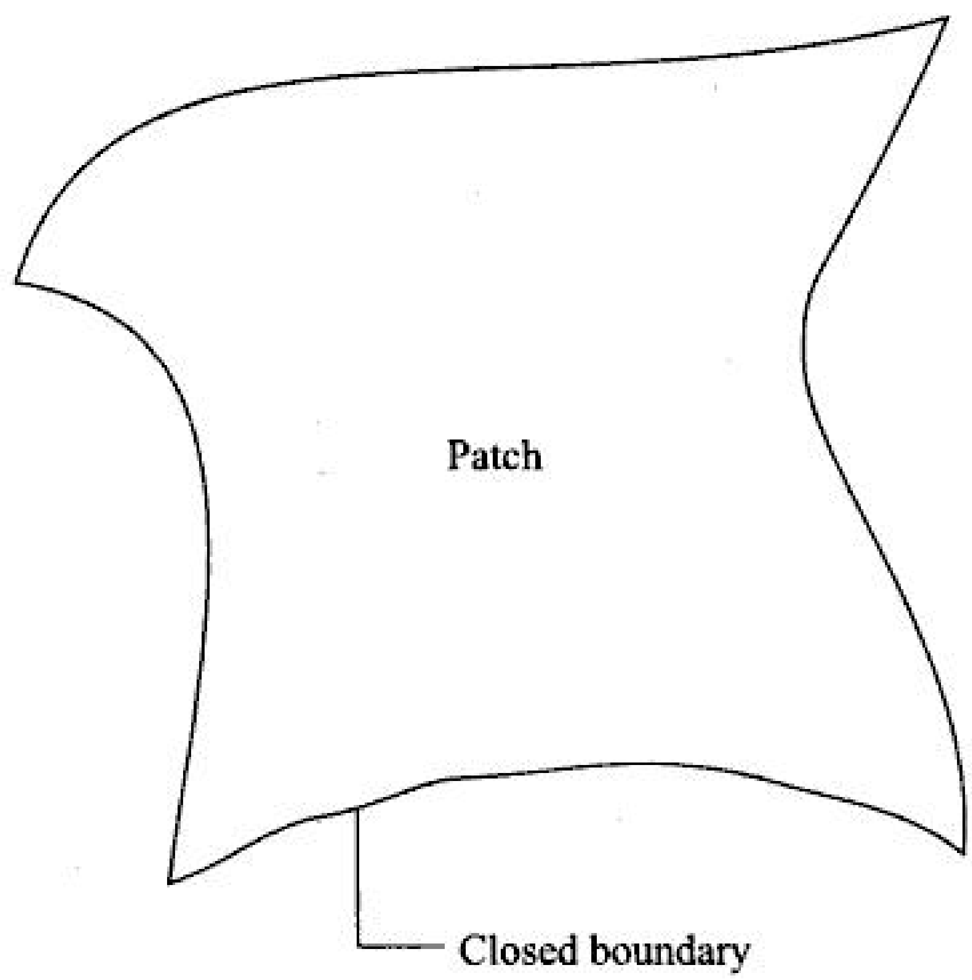


Fig. 5.10 *Coons Patch*

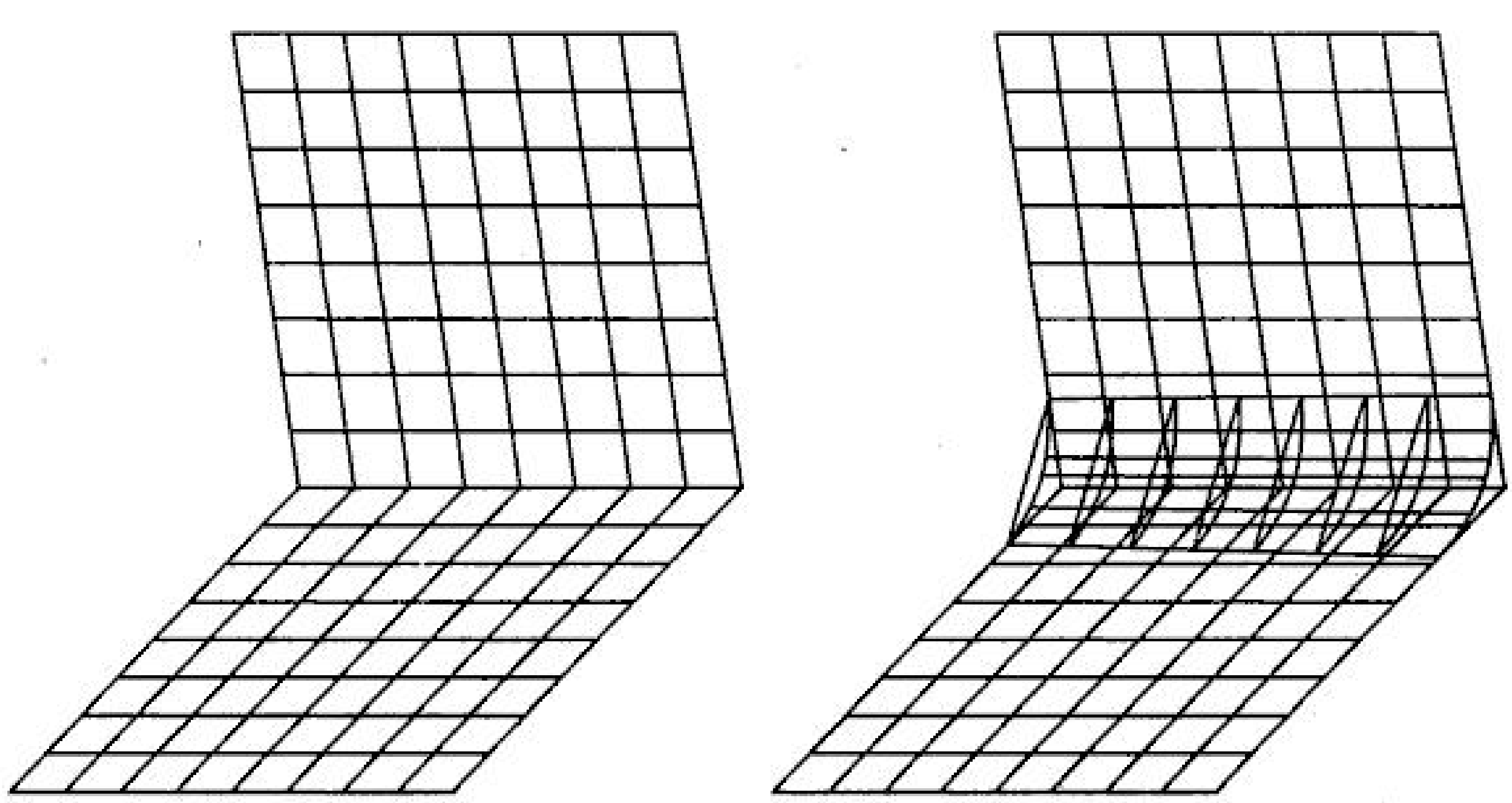


Fig. 5.11 *Fillet Surface*

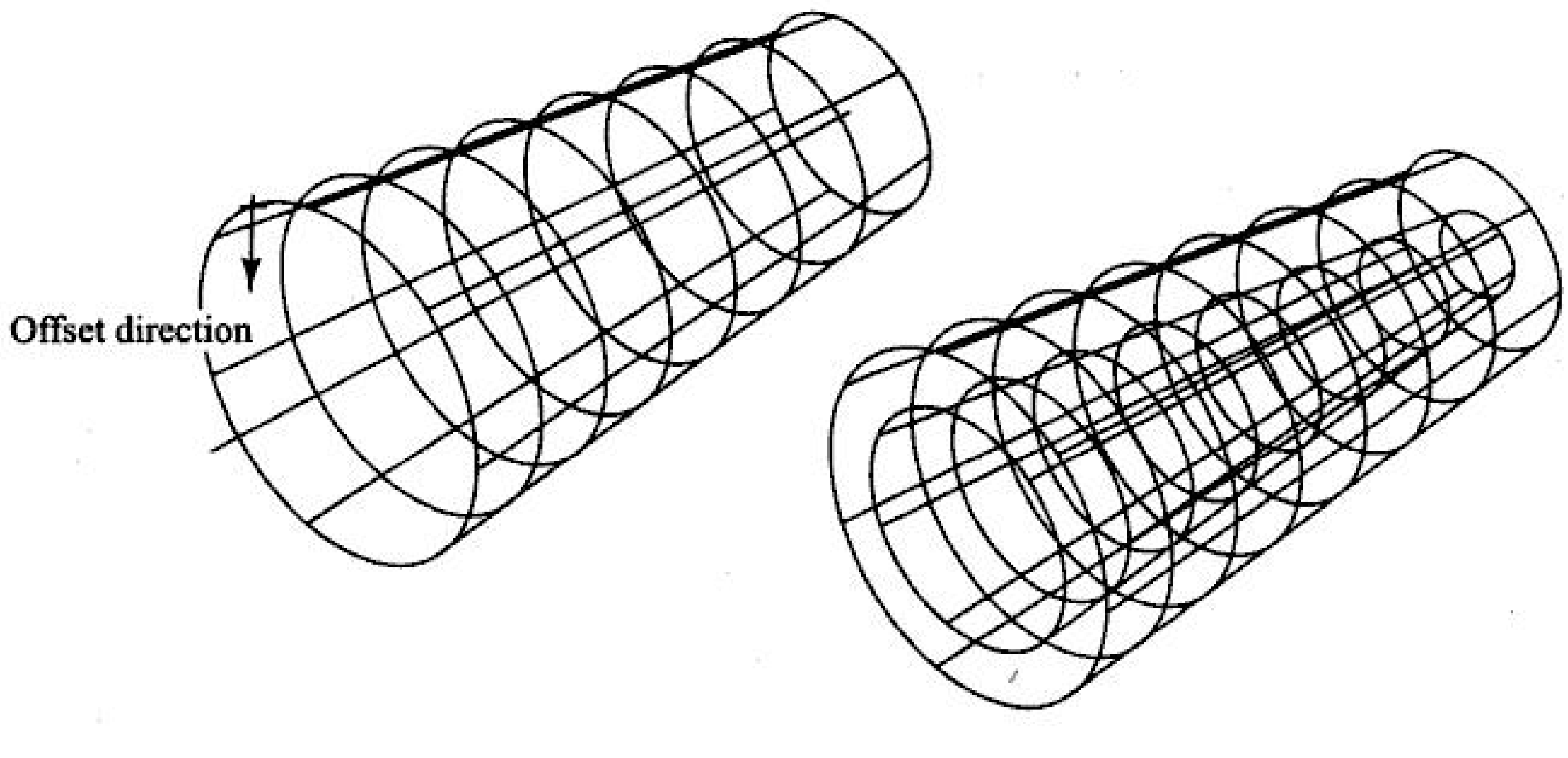


Fig. 5.12 *Offset Surface*

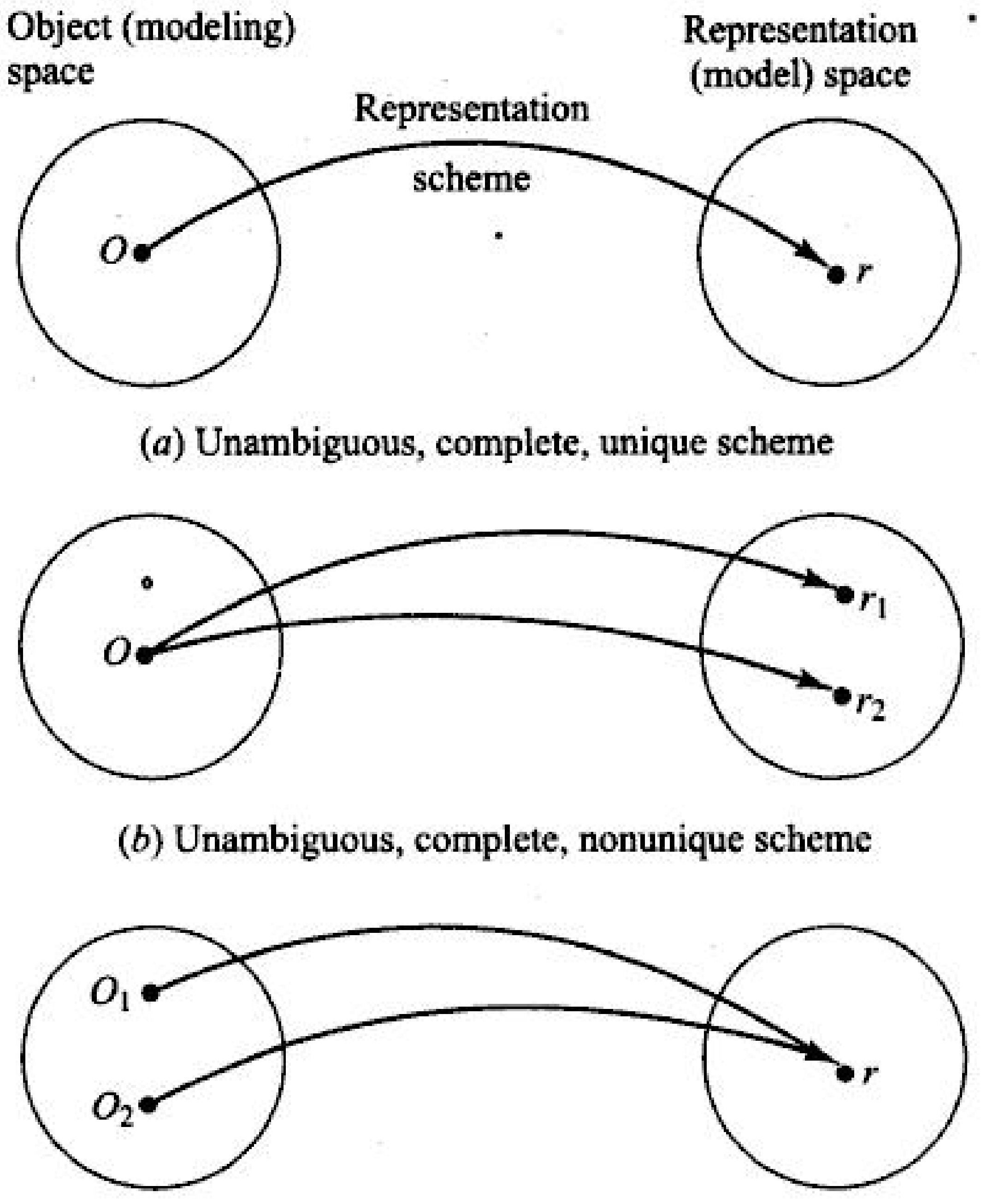
Example 5.1 Create the surface model of the guide bracket shown in Fig. 4.2.

Solution Before creating the surface model, the wireframe model of the bracket created in Example 4.1 is required and its database is assumed to be available for this example. A quick look at Fig. 4.2 reveals that ruled surfaces and tabulated cylinders are sufficient to construct the surface model. The following steps may be followed to construct the surface model:

1. Retrieve the wireframe model database of the bracket. It might be more practical to copy the wireframe model and use the copy to create the surfaces so that if the database is corrupted during surface creation, the original database can be copied again.
2. Select new layer(s) for surface entities to facilitate managing surface and wireframe entities.
3. Create surfaces on the right face of the model by using a ruled surface command. Referring to Fig. 5.13*a*, the line $P_5 P_6$ is divided first into two entities at the intersection point (P_7) of lines $P_2 P_3$ and P_5P_6. The ruled surface command is used twice to create two surfaces using lines $P_3 P_4$ and $P_5 P_7$ (identified by digitizes d_1 and d_2 in Fig. 5.13a) as rails for the first surface and lines P_1P_2 and $P_6 P_7$ (d_3 and d_4 in the figure) as rails for the other surfaces.
4. Use a mirror or duplicate command to copy these surfaces to create surfaces on the left face of the model.
5. Create ruled surfaces on the front face of the top part of the model as shown in Fig. 5.13*a*. Five surfaces are constructed. These surfaces use entities d_5 and d_6; d_7 and d_8; d_9 and d_{10}; d_{11} and d_{12}; and d_{13} and d_{14} as rails. The small circle has to be divided into five arc segments to create these surfaces.
6. Create the surface of the hole shown in Fig. 5.13*a*. Use a tabulated cylinder command with the circle as the cylinder generator.
7. Follow a similar approach to construct all the surfaces of the model. During construction, the user encounters either dividing existing entities or creating new ones to define the appropriate rails of the various ruled surfaces. Figure 5.13*b* shows the completed surface model while Fig. 5.13*c* shows a shaded image of the surface model for better visualization.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



(a) Unambiguous, complete, unique scheme

(b) Unambiguous, complete, nonunique scheme

(c) Ambiguous and incomplete scheme

Fig. 6.9 *Classification of Representation Schemes*

The formal properties of representation schemes which determine their usefulness in geometric modeling can be stated as follows:

1. ***Domain*** The domain of a representation scheme is the class of objects that the scheme can represent or it is the geometric coverage of the scheme.
2. ***Validity*** The validity of a representation scheme is determined by its range, that is, the set of valid representations or models it can produce. If a scheme produces an invalid model, the CAD/CAM system in use may crash or the model database may be lost or corrupted if an algorithm is invoked on the model database. Validity checks can be achieved in three ways: test the resulting databases via a given algorithm, build checks into the scheme generator itself, or design scheme elements (such as primitives) that can be manipulated via a given syntax.
3. ***Completeness or Unambiguousness*** This property determines the ability of the scheme to support analysis and other engineering applications. A complete scheme must provide models with sufficient data for any geometric calculation to be performed on them.
4. ***Uniqueness*** This property is useful to determine object equality. It is a custom in algebra to check for uniqueness but it is rare to do so in geometry. This is because it is difficult to develop algorithms to detect the equivalence of two objects and it is computationally expensive to implement these algorithms if they exist. Positional and permutational nonuniqueness are

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

is defined unambiguously by its boundary and that non-r-sets are not defined unambiguously by their boundaries. The validity of B-rep models is ensured via Euler operations which can be built into the syntax of a CAD/CAM system. However, these models are not unique because the boundary of any object can be divided into faces, edges and vertices in many ways. Verification of uniqueness of boundary models is computationally expensive and is not performed in practice.

6.7.1 Basic Elements

If a solid modeling system is to be designed, the domain of its representation scheme (objects that can be modeled) must be defined, the basic elements (primitives) needed to cover such modeling domain must be identified, the proper operators that enable the system users to build complex objects by combining the primitives must be developed and finally a suitable data structure must be designed to store all relevant data and information of the solid model. Other system and geometric utilities (such as intersection algorithms) may also need to be designed. Let us apply these ingredients to a B-rep system.

Objects that are often encountered in engineering applications can be classified as either polyhedral or curved objects. A polyhedral object (plane-faced polyhedron) consists of planar faces (or sides) connected at straight (linear) edges which, in turn, are connected at vertices. A cube or a tetrahedron is an obvious example. A curved object (curved polyhedron) is similar to a polyhedral object but with curved faces and edges instead. The identification of faces, edges and vertices for curved closed objects such as a sphere or a cylinder needs careful attention, as will be seen later in this section. Polyhedral objects are simpler to deal with and are covered first.

The reader might have jumped intuitively to the conclusion that the primitives of a B-rep scheme are faces, edges and vertices. This is true if we can answer the following two questions. First, what is a face, edge, or a vertex? Second, knowing the answer to the first question, how can we know that when we combine these primitives we would create valid objects? Answers to these questions can help users to create B-rep solid models of objects successfully. To show that these answers are not always simple, consider the polyhedral objects shown in Fig. 6.23. Polyhedral objects can be classified into four classes. The first class (Fig. 6.23*a*) is the simple polyhedra. These do not have holes (through or not through) and each face is bounded by a single set of connected edges, that is, bounded by one loop of edges. The second class (Fig. 6.23*b*) is similar to the first with the exception that a face may be bounded by more than one loop of edges (inner loops are sometimes called rings). The third class (Fig. 6.23*c*) includes objects with holes that do not go through the entire object. For this class, a hole may have a face coincident with the object boundary; in this case we call it a boundary hole. On the other hand, if it is an interior hole (as a void or crack inside the object), it has no faces on the boundary. The fourth and the last class (Fig. 6.23*d*) includes objects that have holes that go through the entire objects. Topologically, these through holes are called handles.

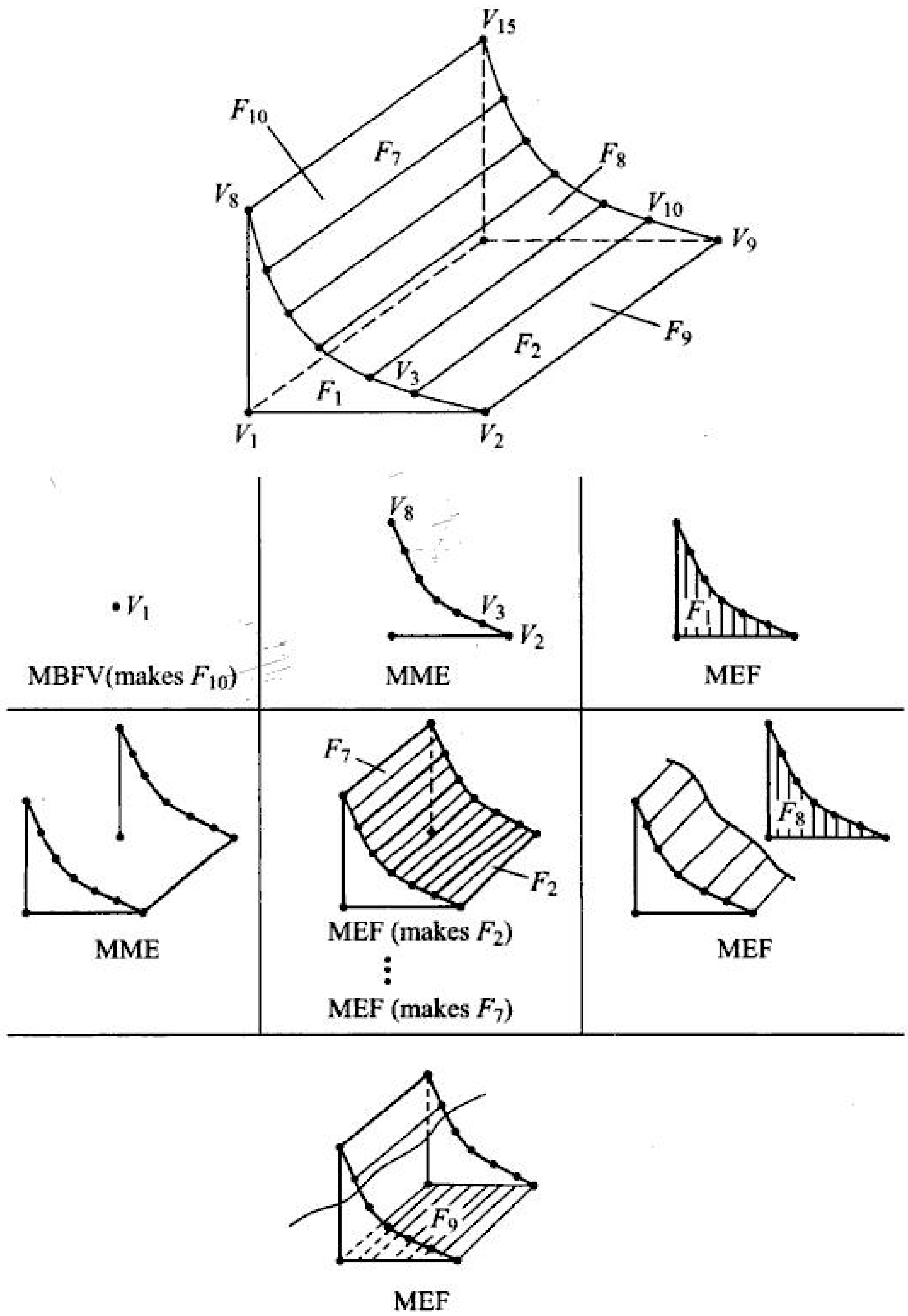


Fig. 6.35 *Creation of Boundary Model of Solid Fillet*

The reader can perhaps find a totally different set of steps than those shown in Fig. 6.34 to construct the boundary model, or these steps can change significantly depending on the available set of Euler operators. For example, if composite Euler operators for linear sweep and making cylinders are available, the model can easily be constructed in a smaller number of steps. The reader is encouraged to investigate this route.

Example 6.6 Create the boundary model of the solid fillet shown in Fig. 6.21.

Solution Fig. 6.35 shows the boundary model of the solid fillet and its creation. The curved face has been approximated by six facets. The construction steps follow

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

$$A - B = \text{Glue } (A \text{ out } B, B \text{ in } A)$$

or

$$B - A = \text{Glue } (B \text{ out } A, A \text{ in } B)$$

The gluing operator for $A \cup B$ and $A - B$ is simple because all the subsolids involved are closed objects. The KFEVB operator described in Tables 6.4 and 6.5 can be used to glue the subsolids to give the proper results, that is, closed solids. However, if the same gluing operator is used in $A \cap B$ and $B - A$, open (unregularized open sets) objects would result. This is because one of its operands (A in B) is an open object—in this example the two-dimensional intersection face. In such a case, the same previous gluing operator, KFEVB, can be used with the difference that it kills only one face instead of two. Therefore, the operator satisfies Eq. (6.58) and is used to kill the open object (intersection face).

The above described algorithm can be applied to any two boundary models whose classifications with each other do not yield A on B and/or B on A cases. This is why we mentioned in the beginning that A and B should not touch each other. The reader can extend solid B to pierce through A and apply the above steps.

The reader is also encouraged to apply this algorithm to the two solids shown in Fig. 6.36*b*. It can be assumed that an algorithm that sorts vertices by their planes is available. In this case, six null edges on each of A and B, four loops for A, eight faces for A, two loops for B and twelve faces for B are created as intermediate results. After killing the null edges and splitting A and B, four and eight faces are created to split A and B respectively to give the final result. The gluing process is exactly as above except that the number of faces the gluing operator has to kill is four instead of two.

6.7.3 Remarks

The B-rep scheme is very popular and has a strong history in computer graphics because it is closely related to traditional drafting. Its main advantage is that it is very appropriate to construct solid models of unusual shapes that are difficult to build using primitives. Examples are aircraft fuselage and automobile body styling. Another major advantage is that it is relatively simple to convert a B-rep model into a wireframe model because the model's boundary definition is similar to the wireframe definition. For engineering applications studied to date, algorithms based on B-rep are reliable and competitive with those based on CSG.

One of the major disadvantages of the boundary model is that it requires large amounts of storage because it stores the explicit definition of the model boundaries. It is also a verbose scheme—more verbose than CSG. The model is defined by its faces, edges and vertices which tend to grow fairly fast for complex models. If B-rep systems do not have a CSG-compatible user interface, then it becomes slow and inconvenient to use Euler operators in a design and production environment. In addition, faceted B-rep is not suitable for many applications such as tool path generations.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

Example 6.12 Figure 6.61 shows a $3 \times 3 \times 1$ inch block. A curved slot of 0.3 inch deep is to be milled in the block using a ball-end mill of 0.25 inch diameter. The equation of the centerline of the slot on the top face of the block is given by $z = -(1.5 - \sqrt[3]{x^2 / 3.5})$. Create the solid model of the block with the slot in it and show the swept volume of the tool if it moves perpendicular to the top plane of the block.

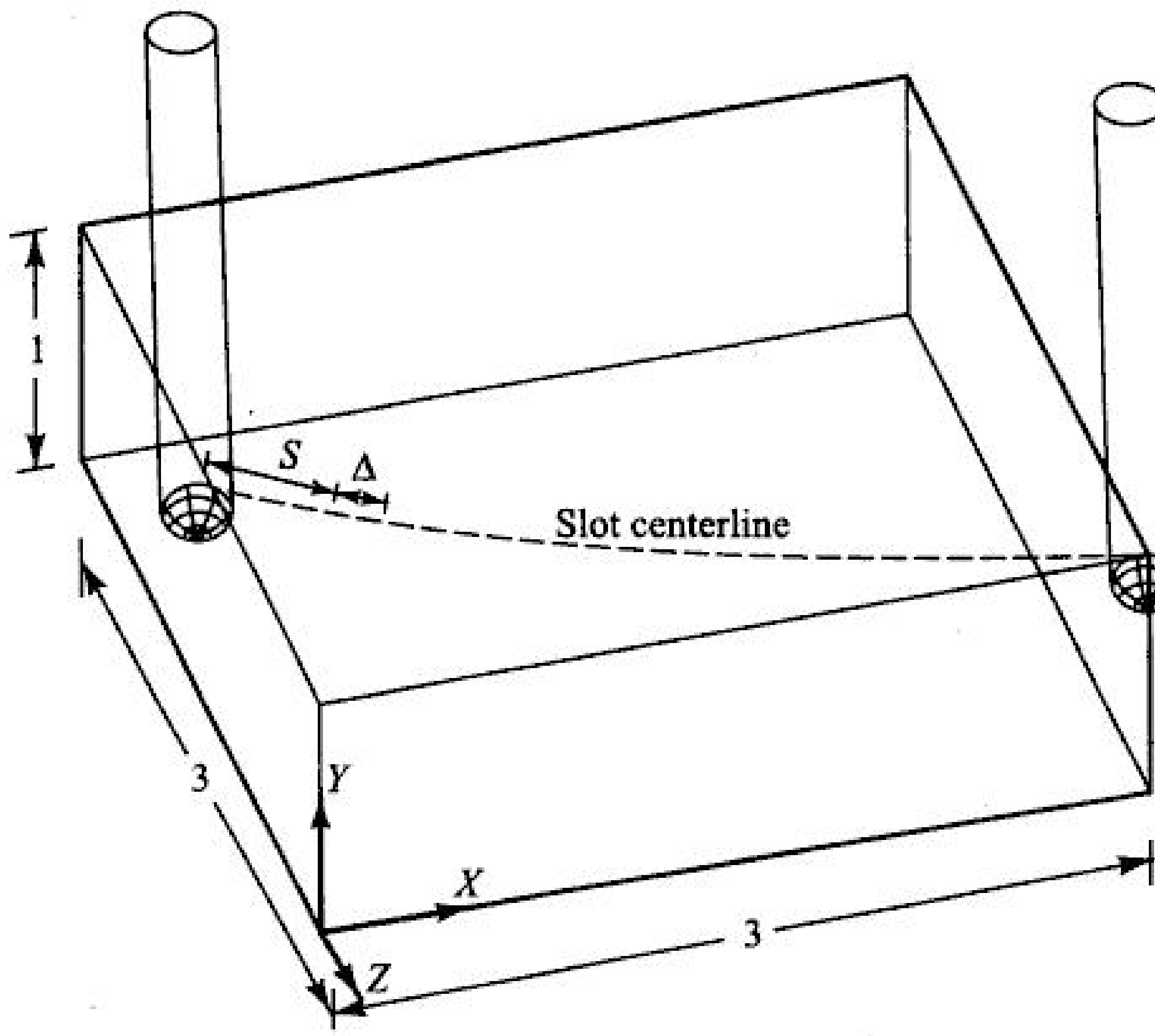


Fig. 6.61 *A Block with a Curved Slot*

Solution This example illustrates how complex shapes can be approximated to fit within the modeling domain of a given solid modeler. Here, we are assuming that the modeler supports boolean operations and has natural quadrics as its minimum set of primitives. If the block had a slot with a straight centerline in any orientation relative to it, its modeling would have been exact and trivial. In the case of a curved centerline, the tool motion is approximated by line segments along the centerline. The solid model of the block becomes a block primitive from which the tool, in its proper position and orientation, is subtracted. The swept volume of the tool is the union of the tool instances.

The tool is the union of a sphere and a cylinder both positioned at 0.175 inch below the top face. This position (0.175) assumes the slot is created in the block by removing all the material in one cut. The original position of the centerline of the tool is at the beginning of the slot centerline, as shown in Fig. 6.61. In this position, the tool is oriented vertically along the Y axis. In order to obtain a fairly smooth slot, the tool is positioned every $d/4$, where d is the tool diameter, that is, every 0.0625 inch. The top view of the profile of the tool swept volume is shown in Fig. 6.62. The curve length S of the slot must be calculated to determine the required number of tool positions, N, to sweep the slot. This length is given by

$$S = \int \sqrt{1 + \left(\frac{dz}{dx}\right)^2}\, dx \qquad (6.104)$$

BIBLIOGRAPHY

Baer, A., C. Eastman and M. Henrion: "Geometric Modeling: A Survey," *Computer Aided Des.* (*CAD*) *J.*, vol. 11, no. 5, pp. 253–272, 1979.

Baumgart, B. G.: "Winged Edge Polyhedron Representation," AIM-179, report STAN-CS-320, Stanford University, 1972.

Baumgart, B. G.: "Geometric Modeling for Computer Vision," AIM-249, report CS–463, Artificial Intelligence Laboratory, Stanford University, 1974.

Berlin, E.: "Solid Modeling on a Microcomputer," *Computer Graphics World*, pp. 39–42, November 1984.

Bin, H.: "Inputting Constructive Solid Geometry Representations Directly from 2D Orthographic Engineering Drawings," *CAD J.*, vol. 18, no. 3, pp. 147–155, 1986.

Bobrow, J. E.: "NC Machine Tool Path Generation from CSG Part Representation," *CAD J.*, vol. 17, no. 2, pp. 69–76, 1985.

Bowerman, R. G.: "Drafting Links Up with Solid Modeling," *Much. Des.*, pp. 46–49, September 12, 1985.

Boyse, J. W.: "Interference Detection among Solids and Surfaces," *Commun. ACM*, vol. 22, no. 1, pp. 3–9, 1979.

Braid, I. C: "The Synthesis of Solids Bounded by Many Faces," *Commun. ACM*, vol. 18, no. 4, pp. 209–218, 1975.

Bronsvoort, W. F., J. J. Wijk and F. W. Jansen: "Two Methods for Improving the Efficiency of Ray Casting in Solid Modeling," *CAD J.*, vol. 16, no. 1, pp. 51–55, 1984.

CAM-I, Inc.: "Design of an Experimental Boundary Representation and Management System for Solid Objects," report R-80-GM-02, CAM-I, Arlington, Tex., 1980.

CAM-I, Inc.: "Boundary File Design," report R-81-GM-02.1, CAM-I, Arlington, Tex., 1981.

CAM-I, Inc.: " Boundary Representation for Solid Objects," report R-82-GM-02.1, CAM-I, Arlington, Tex., 1982.

CAM-I, Inc.: "Extended Geometric Facilities for Surface and Solid Objects," report R-83-GM-02.1, CAM-I, Arlington, Tex., 1983.

CAM-I, Inc.: "Solid Modeling Applications; The Real Payback," *Proc. CAM-I's 3rd Geometric Modeling Seminar*, March 19–20, 1985, Nashville, Tenn.

Casale, M.: "Free-Form Solid Modeling with Trimmed Surface Patches," *IEEE CG&A*, pp. 33–43, January 1987.

Casale, M. S. and Stanton, E. L.: "An Overview of Analytic Solid Modeling," *IEEE CG&A*, pp 45–56, February 1985.

Childress, R. L.: *Sets, Matrices and Linear Programming*, Prentice-Hall, Englewood Cliffs, N.J., 1974.

Chiyakura, H. and F. Kimura: "A Method of Representing the Solid Design Process," *IEEE CA&A*, pp. 32–41, April 1985.

Choi, B. K., M. M. Barash and D. C. Anderson: "Automatic Recognition of Machined Surfaces from a 3D Solid Model," *CAD J.*, vol. 16, no. 2, pp. 81–86, 1984.

Clark, A. L.: "Roughing It: Realistic Surface Types and Textures in Solid Modeling," *Computers In Mechanical Engineering* (*CIME*) *Mag.*, pp. 12–16, March 1985.

Congdon, R. M. and D. C. Gossard: "Interactive Graphic Input of Plane-Faced Solid Models," *Proc. Conf. on CAD/CAM Technology in Mechanical Engineering*, March 24–26, 1982, pp. 350–360, MIT, Cambridge, Mass.

Crocker, G. A.: "Screen-Area Coherence for Interactive Scanline Display Algorithms," *IEEE CA&G*, pp. 10–17, September 1987.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

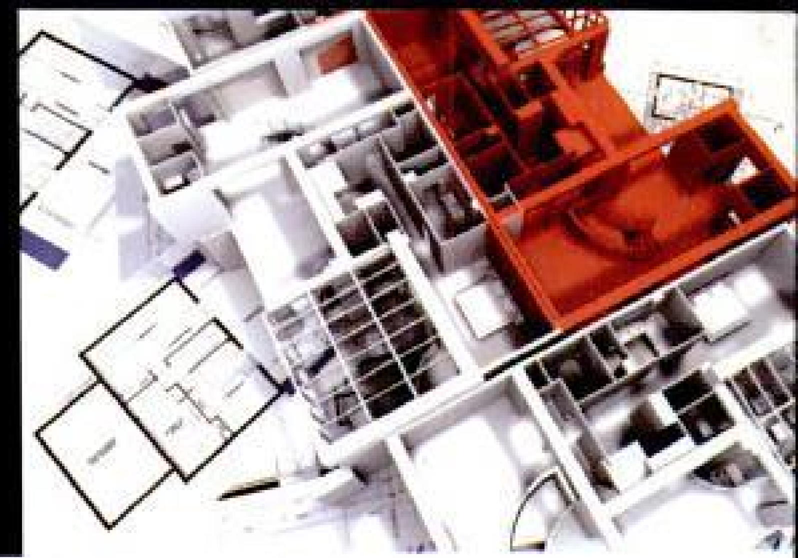
You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

CAD/CAM

Theory and Practice



Second Edition

The revised and updated edition of this classic text provides a comprehensive coverage of CAD/CAM with focus on the application aspects. Replete with numerous solved examples and practical problems, it provides an excellent learning experience to students and practicing engineers.

Salient features

* Exhaustive coverage of Geometric Modeling
* Separate chapters on 'Visual Realism' and 'Finite Element Modeling and Analysis'
* New sections on
    * Computer Aided Process Planning
    * Computer Assisted Production Management System
    * Shop Floor Control System
    * Flexible Manufacturing Systems
    * Computer Monitoring and Control System
    * Parts-Based Manufacturing Information System
    * Computer Integrated Manufacturing
* Pedagogy
    * 94 Solved examples
    * 448 Theoretical questions
    * 69 Programming exercises
    * 88 Laboratory exercises

URL: http://www.mhhe.com/zeid/cctp2e

Visit us at : www.tatamcgrawhill.com

The McGraw·Hill Companies

ISBN-13: 978-0-07-015134-5
ISBN-10: 0-07-015134-2

9 780070 151345

McGraw Hill Higher Education