MODULE 1

Course Objectives

1. To impart basic knowledge on Computer Aided Design methods and procedures
2. To introduce the fundamentals of solid modelling
3. To introduce the concepts of finite element analysis procedures.

Expected outcome

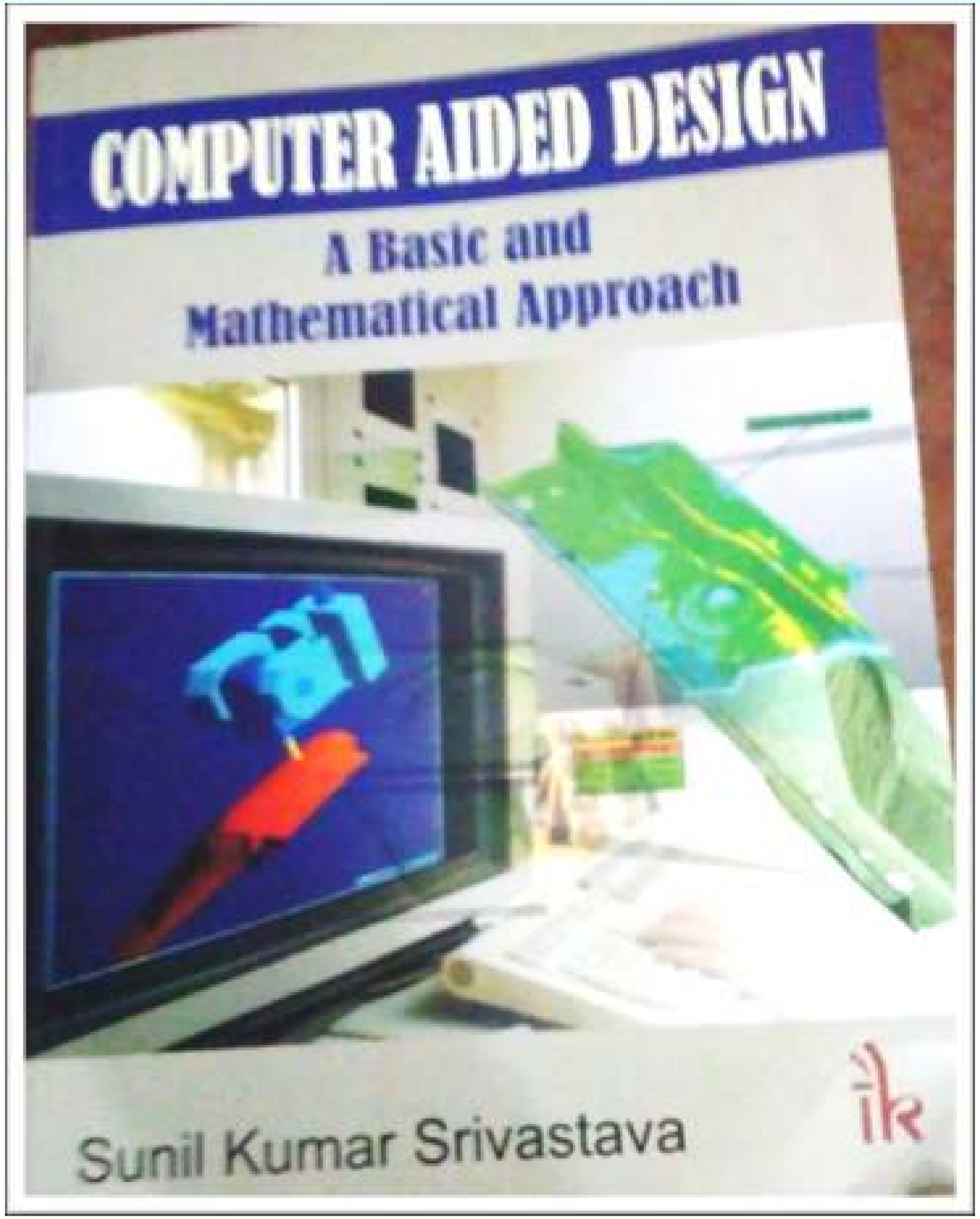
The students will be able to,
1. Gain a basic knowledge on Computer Aided Design methods and procedures
2. Understand the fundamentals of solid modelling
3. Have a basic knowledge in finite element analysis procedures.

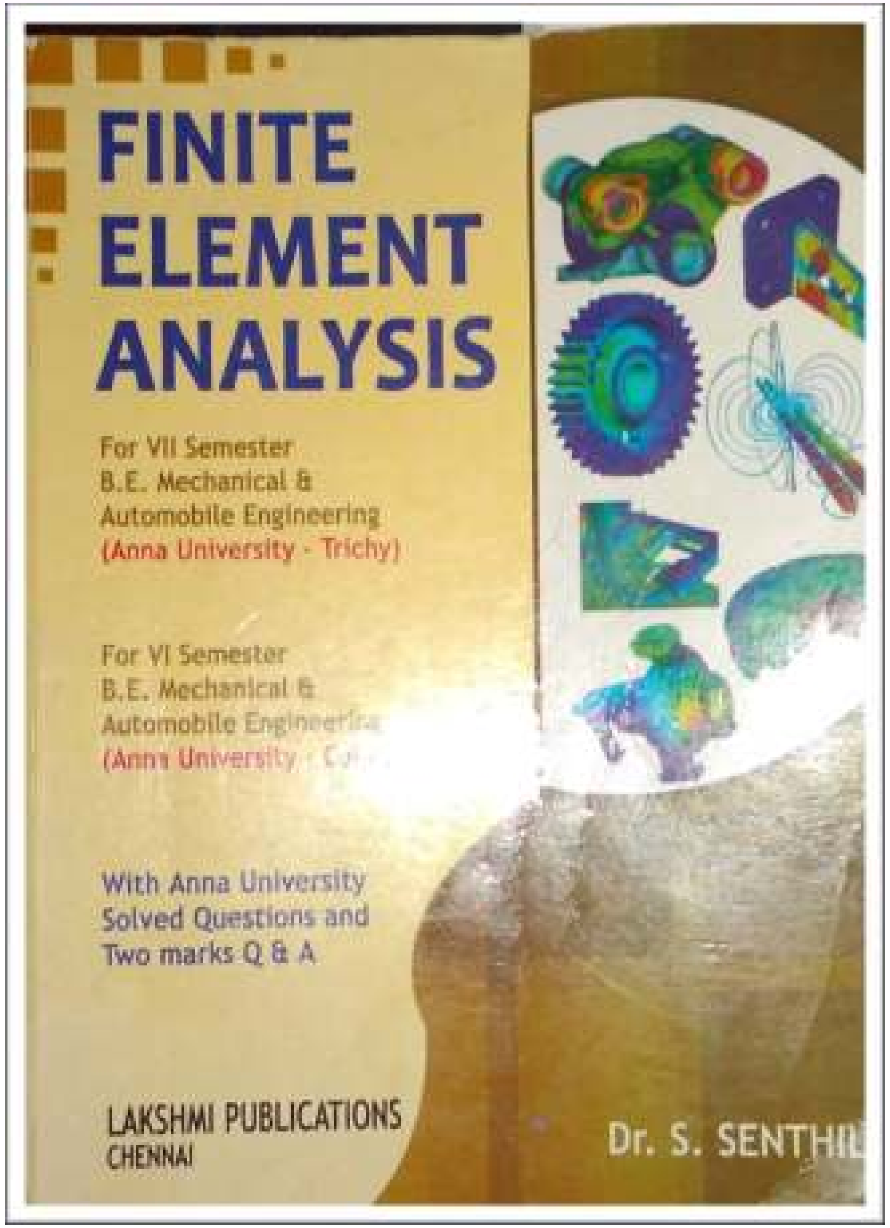
SYLLABUS

Module	Contents	Hours	End Sem. Exam Marks
I	Introduction to CAD , Historical developments, Industrial look at CAD, Comparison of CAD with traditional designing, Application of computers in Design	2	15%
	Basics of geometric and solid modeling, Packages for CAD/CAM/CAE/CAPP	1	
	Hardware in CAD components, user interaction devices, design database, graphic Standards, data Exchange Formats, virtual Reality.	4	
II	Transformation of points and line, 2-D rotation, reflection, scaling and combined transformation, homogeneous coordinates, 3-D scaling.	4	15%
	Shearing,rotation, reflection and translation, combined transformations, orthographic and perspective projections, reconstruction of 3-D objects.	3	
FIRST INTERNAL EXAM			
III	Algebraic and geometric forms, tangents and normal, blending functions, reparametrization, straight lines, conics, cubic splines, Bezier curves and B-spline curves.	4	15%
	Plane surface, ruled surface, surface of revolution, tabulated cylinder, bi-cubic surface, bezier surface, B-spline surfaces and their modeling techniques.	3	

IV	Solid models and representation scheme, boundary representation, constructive solid geometry.	3
	Sweep representation, cell decomposition, spatial occupancy enumeration, coordinate systems for solid modeling.	4
SECOND INTERNAL EXAM		
V	Introduction to finite element analysis - steps involved in FEM- Preprocessing phase – discretisation - types of elements	2
	Formulation of stiffness matrix (direct method, 1-D element) - formulation of load vector - assembly of global equations - implementation of boundary conditions - solution procedure - post processing phase	3
	Simple problems with axial bar element (structural problems only)	2
VI	Interpolation – selection of interpolation functions - CST element - isoparametric formulation (using minimum PE theorem) – Gauss-quadrature	4
	Solution of 2D plane stress solid mechanics problems (linear static analysis)	3
END SEMESTER EXAM		

TEXTBOOKS


COMPUTER AIDED DESIGN
A Basic and
Mathematical Approach
Sunil Kumar Srivastava


FINITE
ELEMENT
ANALYSIS
For VII Semester
B.E. Mechanical &
Automobile Engineering
(Anna University - Trichy)
For VI Semester
B.E. Mechanical &
With Anna University
Solved Questions and
Two marks Q & A
LAKSHMI PUBLICATIONS
CHENNAI

Product cycle

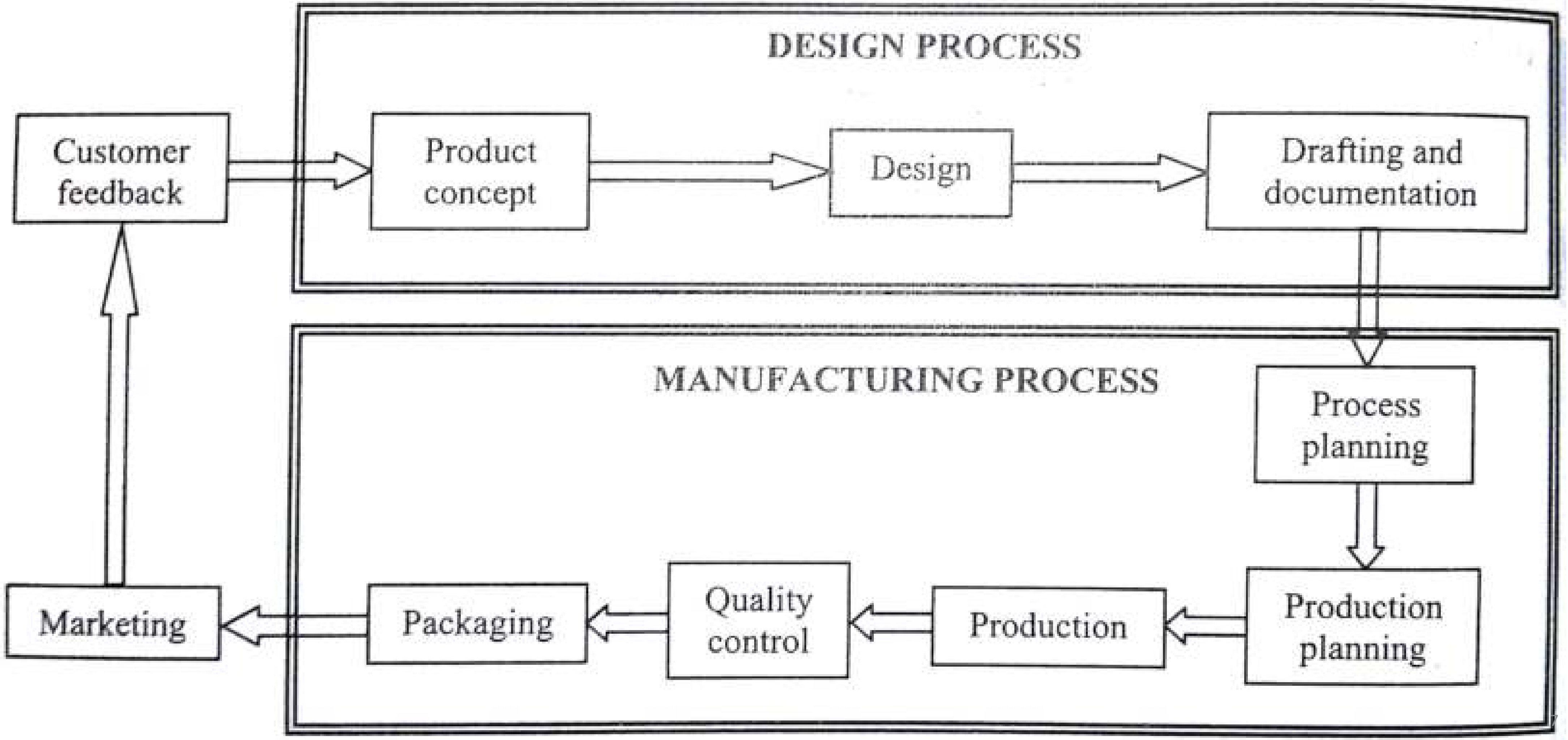
- The product begins with a need which is identified based on **customers' and markets' demands.**
- The product goes through two main processes from the idea conceptualization to the finished product:

1. **The design process.**
2. **The manufacturing process.**

The main sub-processes that constitute the design process are:

1. Synthesis.
2. Analysis and Optimization.

Various stages in conventional Product cycle



Various stages in the conventional product cycle.

- The steps in a production cycle start from the **customer feedback** to develop the product concept,
- The **design** steps (**synthesis, analysis, optimization, etc.)** to draft and document the design of product including components drawing, assembly drawing, material specifications, etc,
- Next, the design is handed over to the **manufacturing department,** wherein, the **process planning** (sequence of the manufacturing operations) is formulated,
- Followed by the **production planning** and actual manufacture of the product.
- Thereafter, the **inspection and testing** of products is carried out to ensure certain quality standards.
- Finally, the products are packed and **shipped for marketing**.
- The **customer feedback** is continuously accepted and used to improve the design process.


PHILIPS

Capacitive vs Resistive touch screen

What is an engineering design?

- **Design is an activity that felicitate the realization of new products or processes through which technology satisfy the human needs and aspirations.**
- An act of working out the form of something by making sketch or outline or plan.
- An act converting functioning requirements into products.

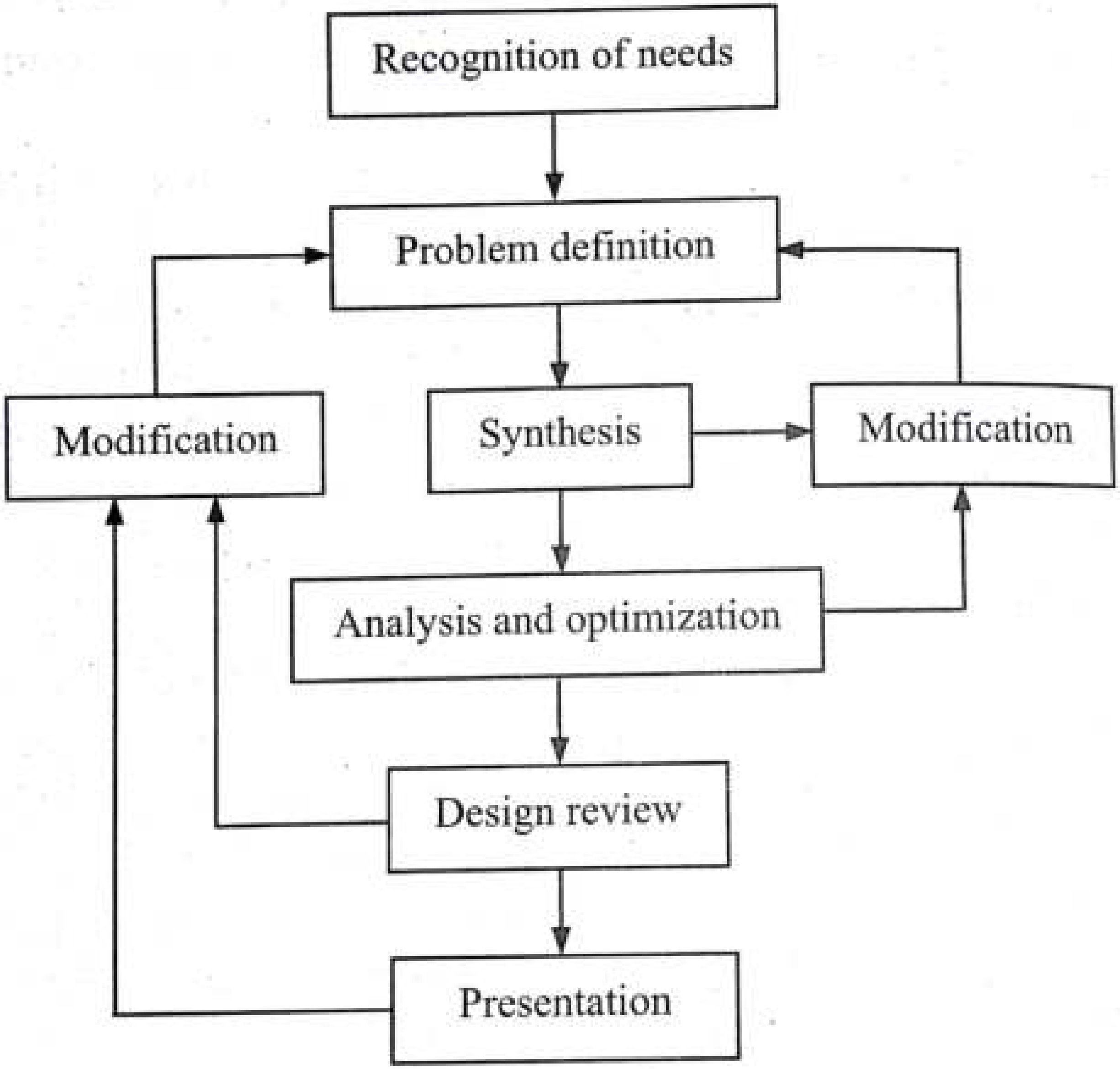
Examples of designed products

- Design of aircraft career, Web-page, Highway and Gear Box

What is Computer Aided Design?

- Use of computer systems to assist in creation, modification, analysis and optimization of the designs.

Conventional design process



Recognition of need

- Adoption of existing design
- Modification of existing design
- Completely new design

Problem definition (i.e., specification)

- The designer collects different information, about the existing products of similar type, about the market potential, about the manufacturing constraints, about the legal requirements and standards, and so on.

Synthesis (i.e., conceptualization)

- Synthesis forms a design solution to satisfy the need. The end goal of synthesis is a **conceptual design** of the product. Synthesis sub process generates the information regarding design of the product. In this phase, **sketches of different components and assembly are drawn**.

Analysis and Optimization

- Every synthesis must follow the analysis**. Analysis means critically examining an already existing or proposed design to judge the suitability for the task that is to be performed by the designer**.
- Analysis determines whether the performance complies with the requirements or not. The analysis sub process selects suitable **material** and its associative mechanical properties. Calculations are performed to **determine the size or parameters** using the physical laws (i.e., laws of momentum, motion, energy conservation, etc.).
- The different types of engineering ' analyses are stress-strain analysis, kinematic analysis, dynamic analysis, vibration analysis, thermal analysis, fluid-flow analysis, etc.
- Optimization means the best possible solution for the given objectives. All possible solutions are analyzed and optimum is selected. After every phase of design process, the designer may go to the previous steps and modify them.

Design review (i.e., evaluation):

- Evaluation means measuring the design against the specifications set in the problem definition. It usually involves **prototype building and testing** of the ‘product to ascertain operating performance or factors such as reliability. The result of the evaluation phase may yield a satisfactory design or it may lead to the further modifications in the design parameters.

Presentation (i.e., drafting)

- The final stage in the design process is the presentation and documentation of the design on the paper. This forms an interface between design and the manufacturing.

What is Computer Aided Design?

- Use of computer systems to assist in creation, modification, analysis and optimization of the designs.
- Computer Aided Design uses computer as a tool/medium for the design.
- CAD is the use of computer to aid in the design process of an individual part, a subsystem or a total system.
- It's an automation of design process.

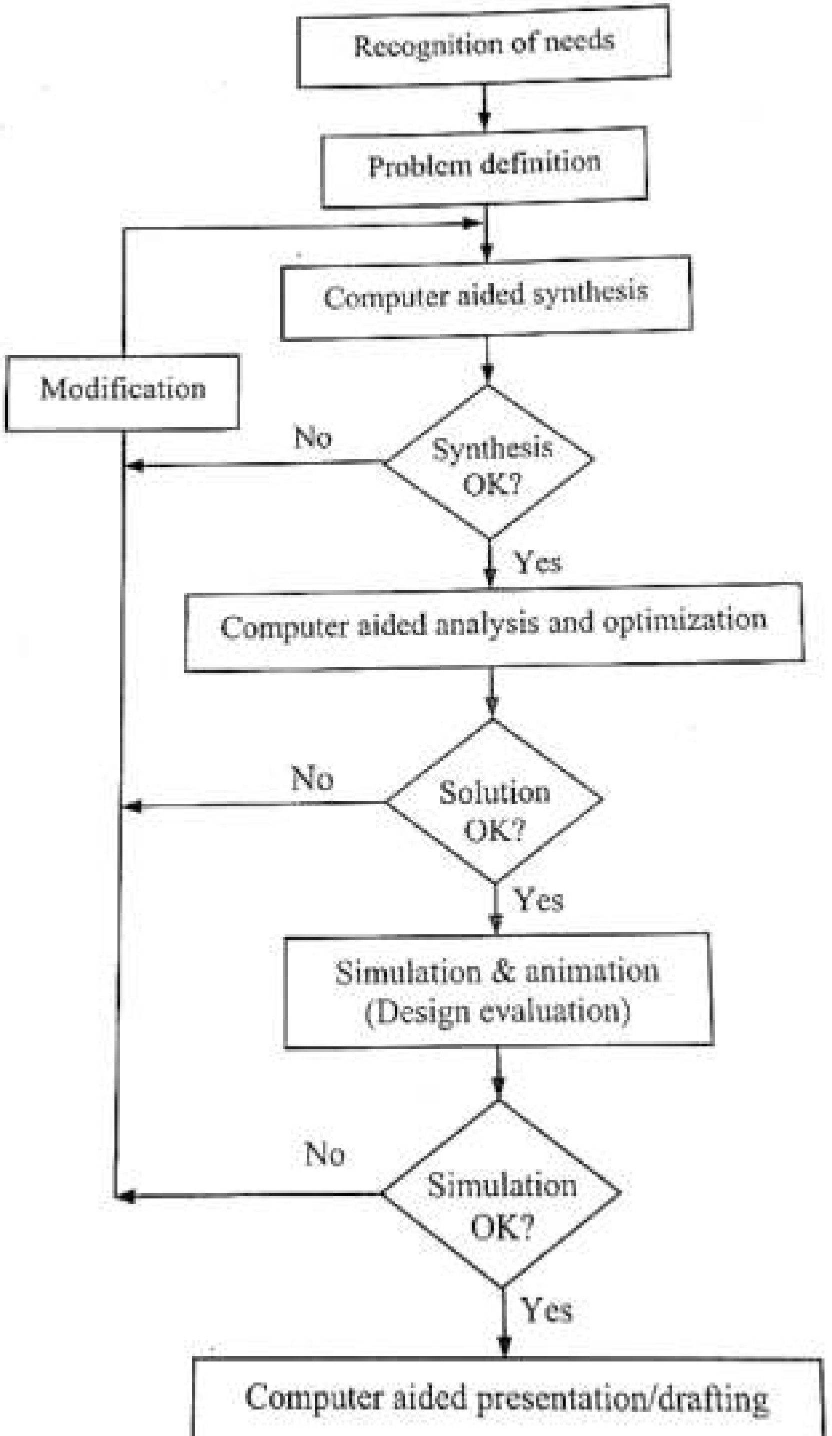
INTRODUCTION TO CAD

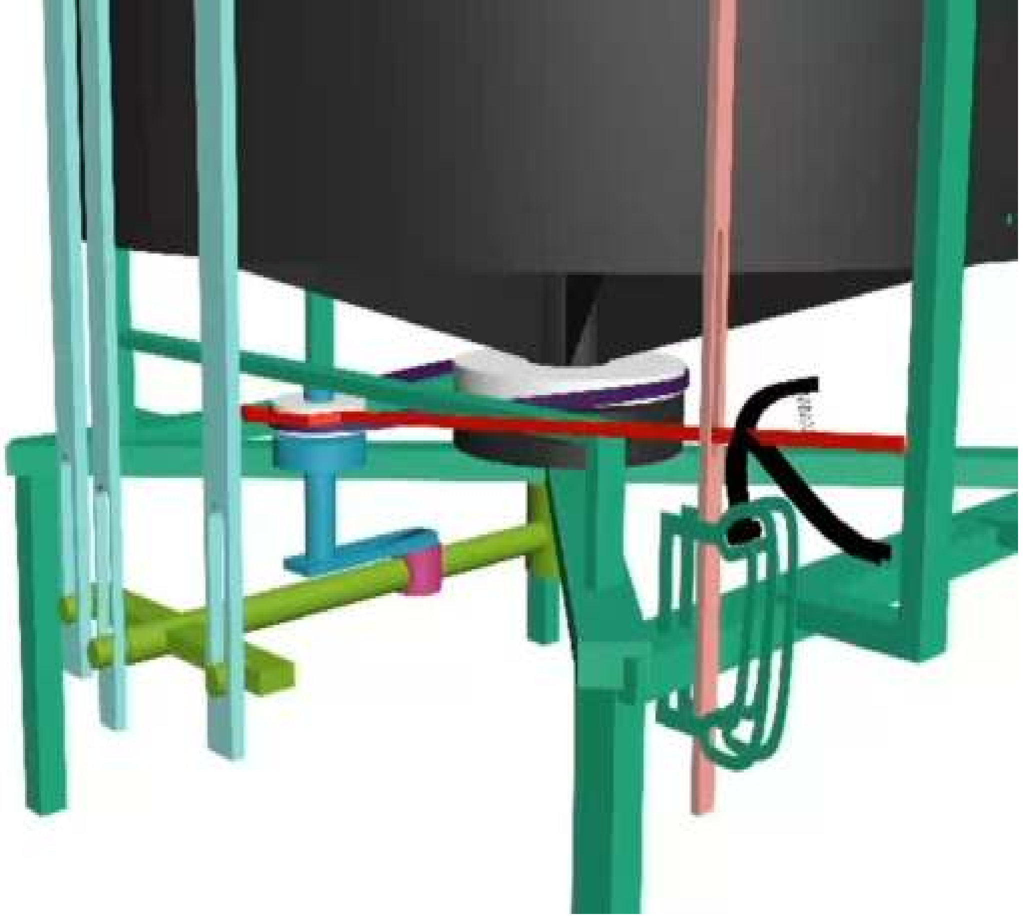
- A computer, as a piece **of hardware, consists of input and output devices, arithmetic and control units, and a memory**. Equally essential **the software**, the program of instructions, tells the computer how to process data, i.e., it includes all types of programming instructions that facilitate the utilization of computer hardware.

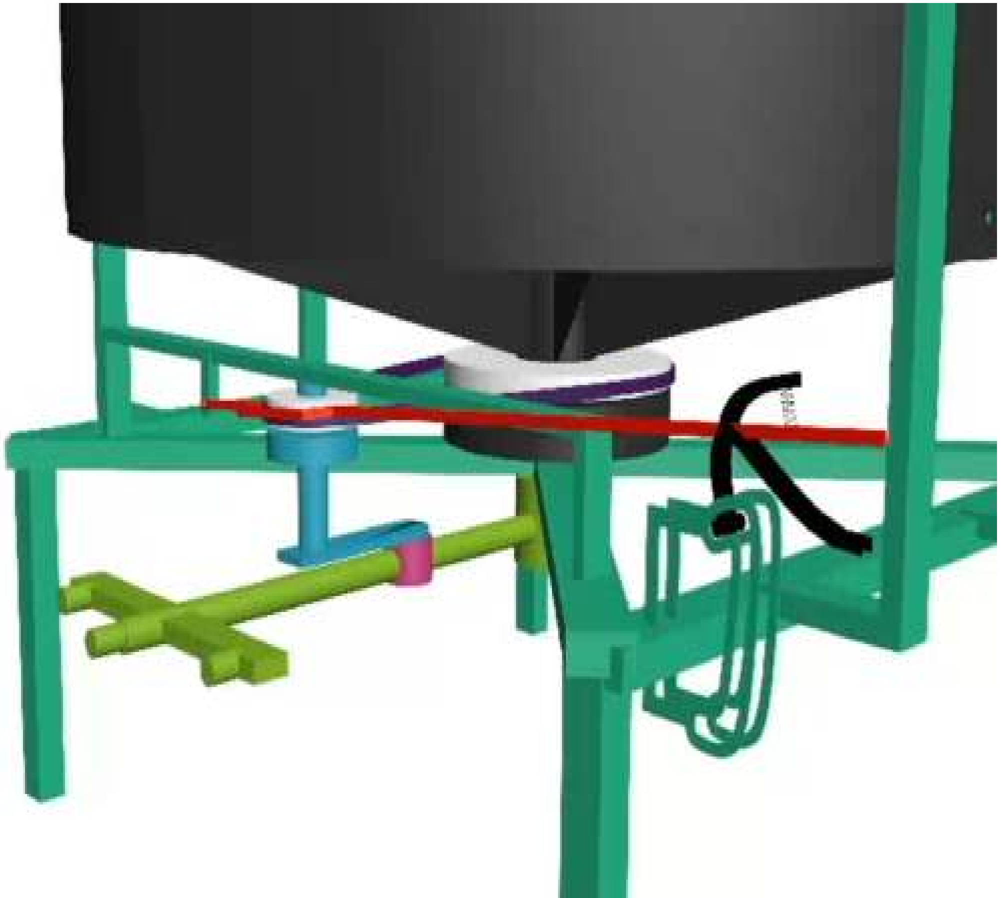
- The development of hardware includes the peripherals associated with inp**ut, storage and output units**. Research and development in software is focused upon improving.

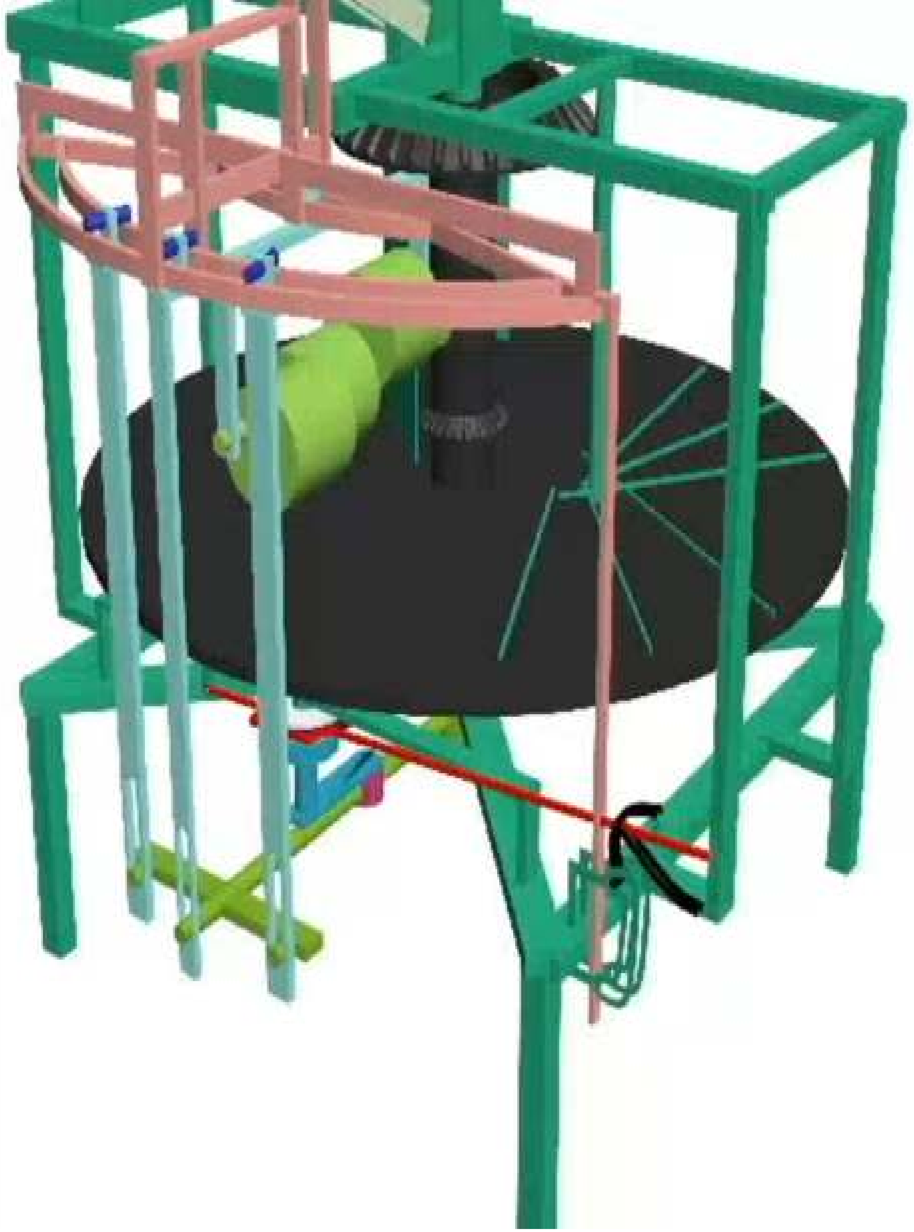
- A designer should have good amount of software and hardware knowledge to carry out the design process effectively and efficiently. The computer can respond to the designer whatever the designer has put, leading to an active medium (monitor screen) for the design process.

Computer Aided Design

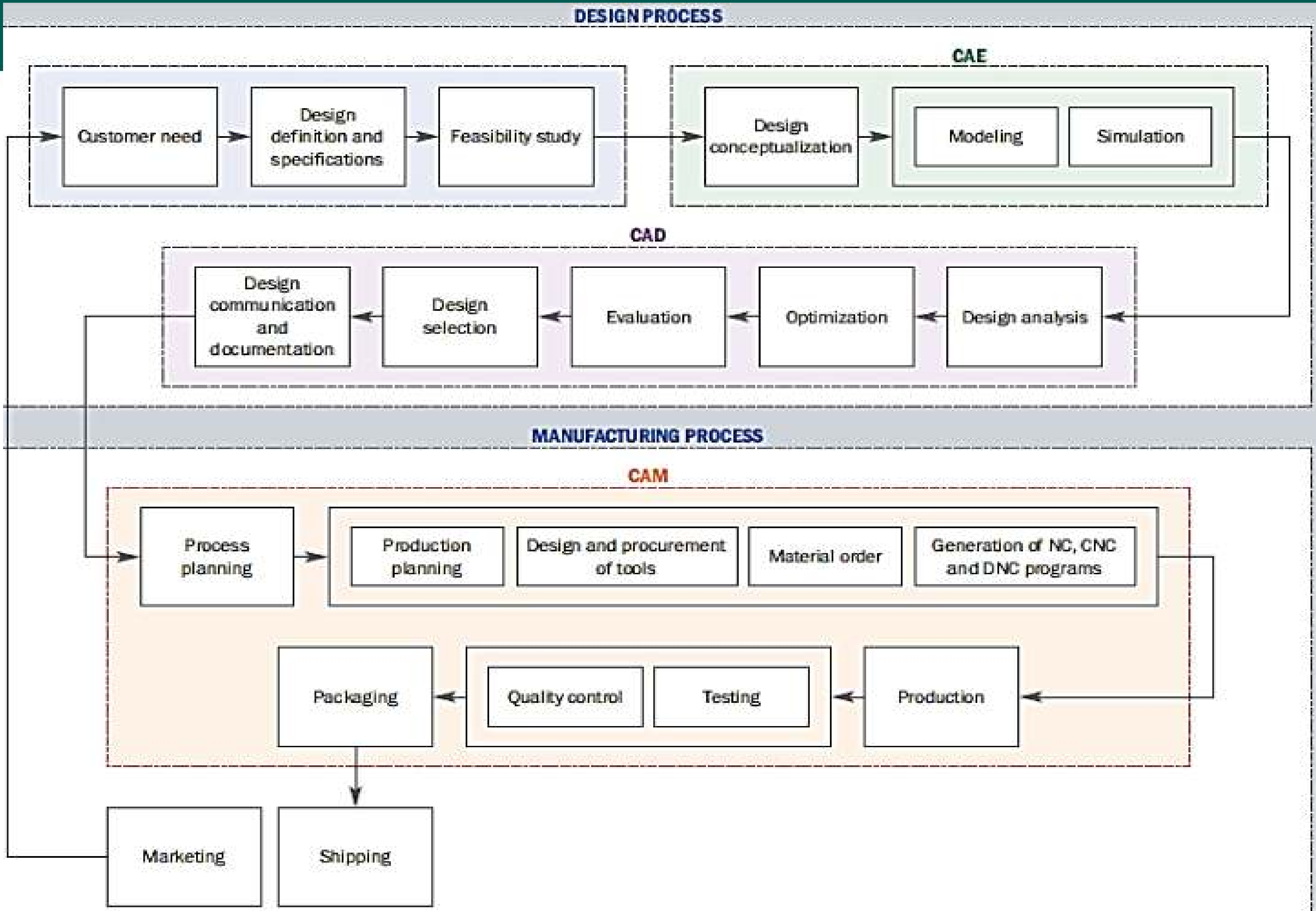








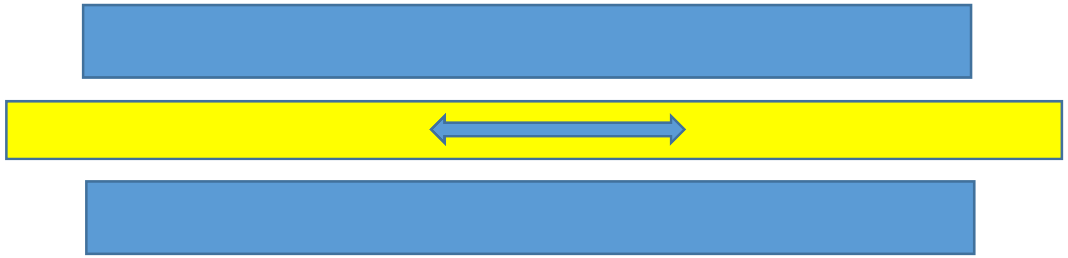
PRODUCT LIFE CYCLE



Different Stages of Design

- Conceptual Design Or Preliminary Design
- Configuration Design
- Detail Design Or Routine Design

Example : design a subsystem for linear motion



Computers have found to assist in all stages of design process

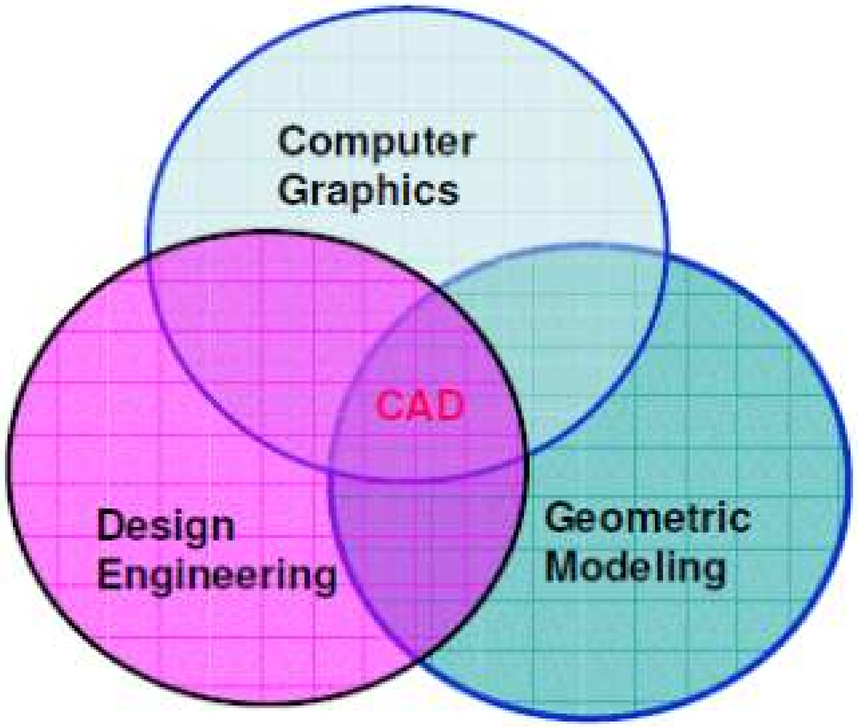
CAD/CAM?

CAD/CAM = Computer Aided Design and Computer Aided Manufacturing.

It is the technology concerned with the use of computers to perform design and manufacturing functions.

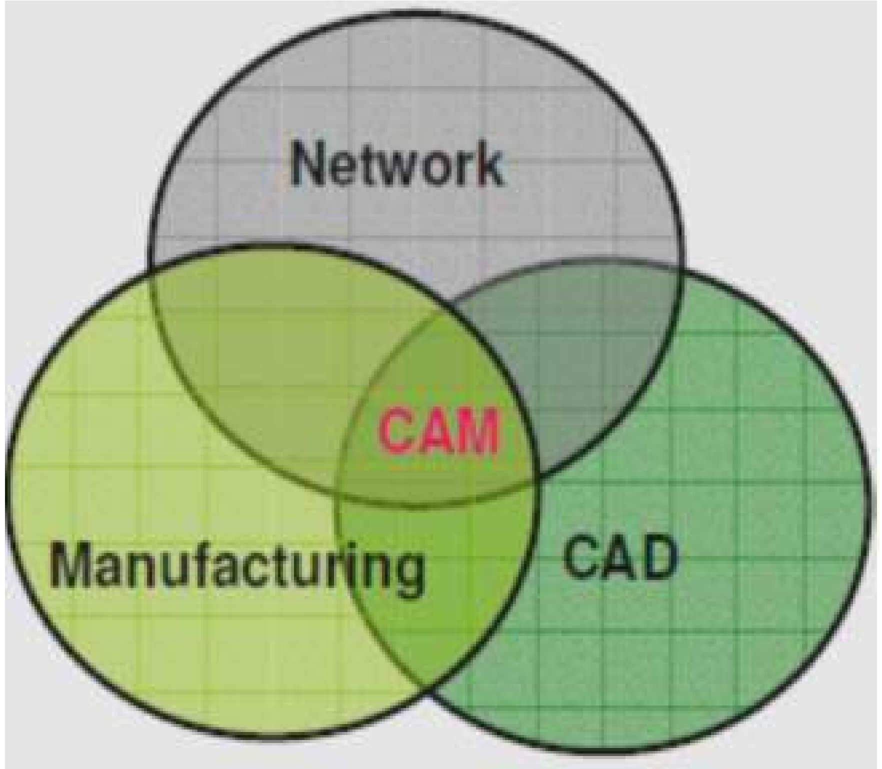
Computer-Aided design (CAD)

Computer systems is used to assist in the creation, modification, analysis, or optimization of a design.



Computer-Aided Manufacturing(CAM)

Computer systems are used to plan, manage, and control the operations of a manufacturing plant through direct or indirect computer interface with plant's resources.



Need for CAD/CAM

- To increase productivity of the designer
- To improve quality of the design
- To improve communications
- To create a manufacturing database
- To create and test toolpaths and optimize them
- To help in production scheduling and MRP models
- To have effective shop floor control

The Product Cycle & CAD/CAM

Various activities + functions = Product Cycle

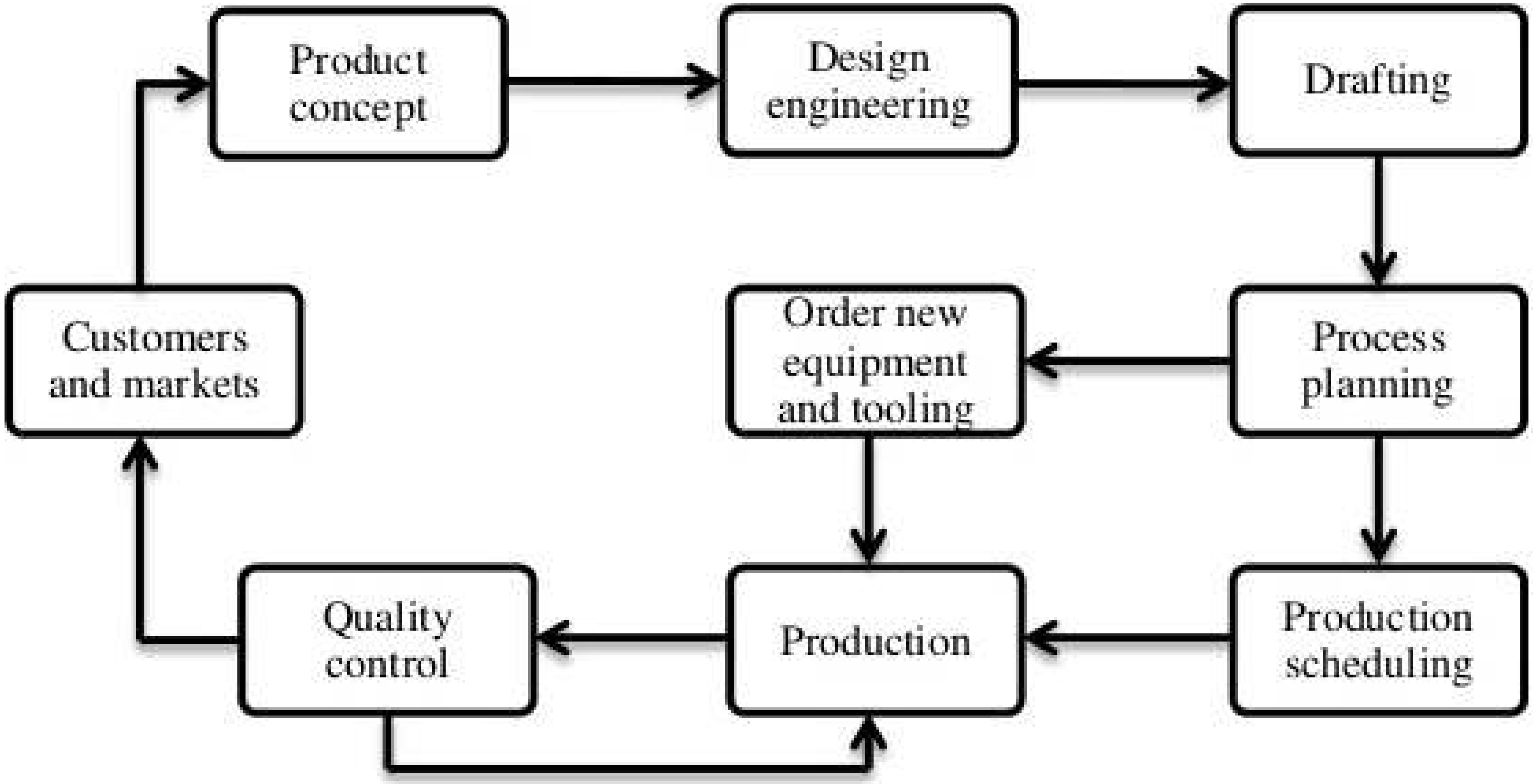
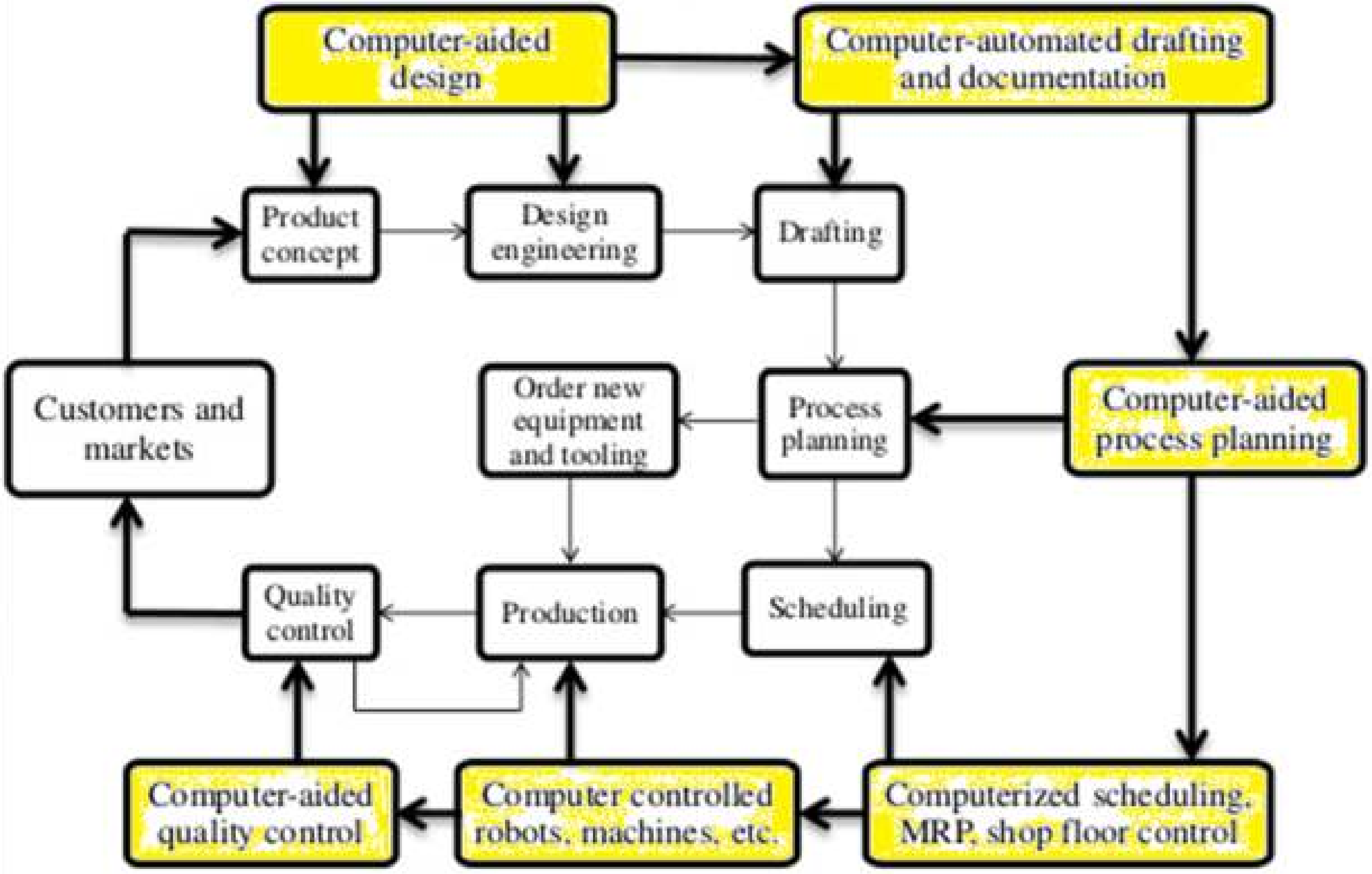


Fig. 5: Product Cycle

The Product Cycle & CAD/CAM



Product Cycle revised with CAD/CAM overlaid

CAD Tools Required to Support the Design Process

Design phase	Required CAD tools
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; BOM; NC.
Design communication and documentation	Drafting and detailing…

CAM Tools Required to Support the Design Process

Manufacturing phase	Required CAM tools
Process planning	CAPP techniques; cost analysis; material and tooling specification.
Part programming	NC programming
Inspection	CAQ; and Inspection software
Assembly	Robotics simulation and programming

Computer-Aided Engineering (CAE)

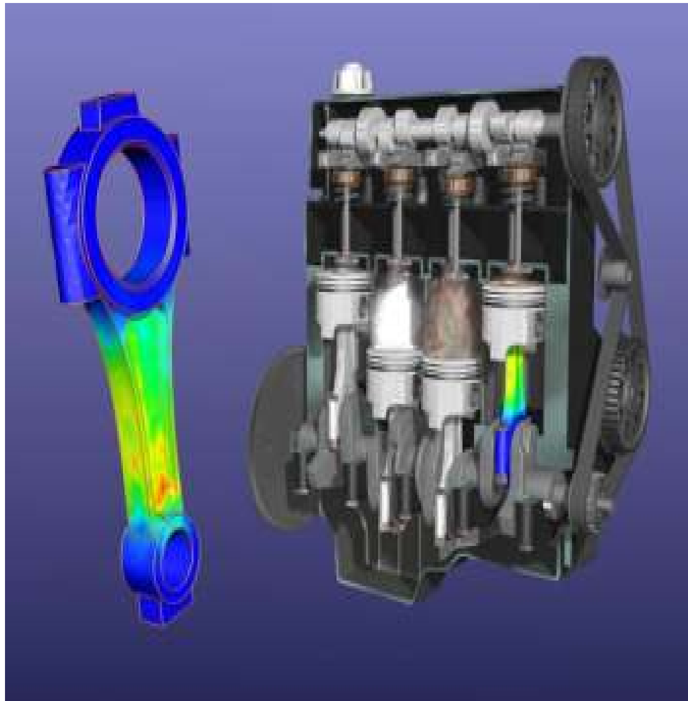
- Computer aided engineering (CAE) is a philosophy of the product design and development that brings tougher entire engineering activities related to the design and production of a product in industries.

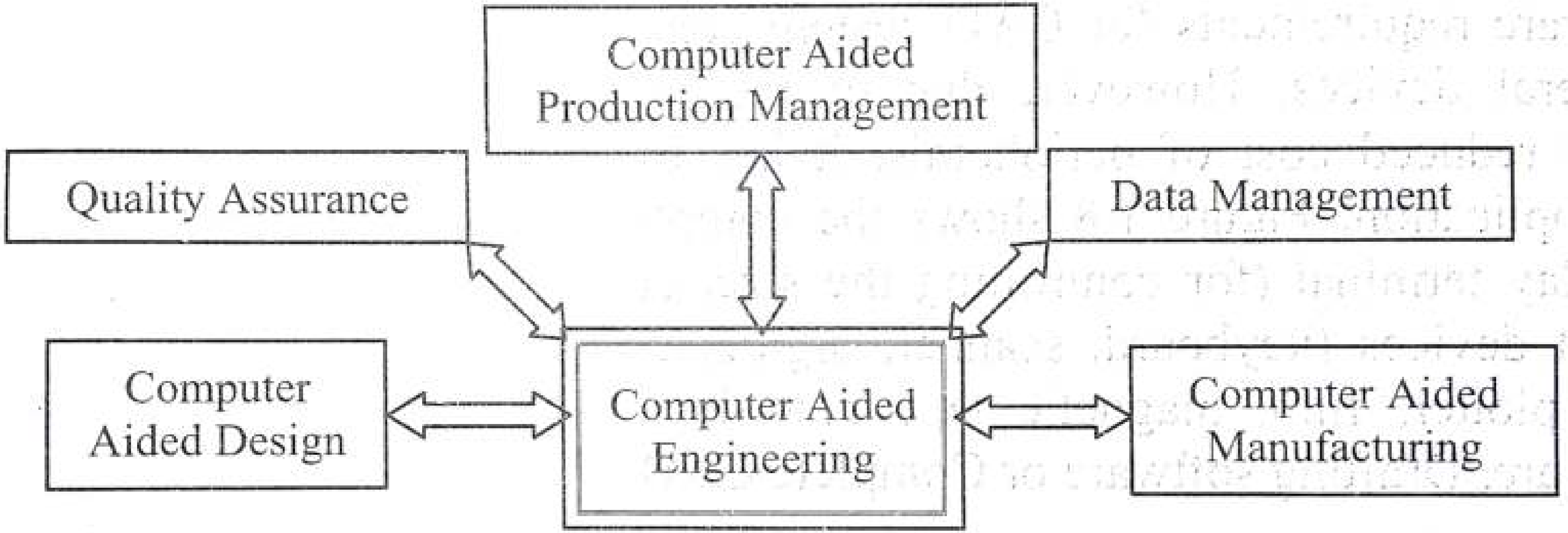
- Use of computer systems to analyze CAD geometry

- Allows designer to simulate and study how the product will behave, allowing for optimization

- Finite-element method (FEM)
  - Divides model into interconnected elements
  - Solves continuous field problems

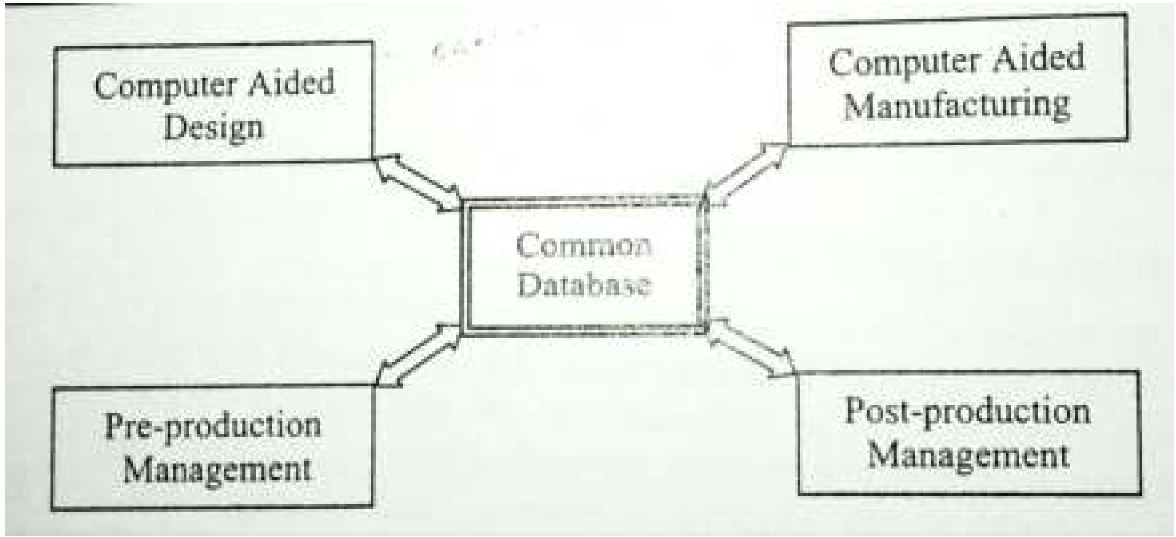
- Other Capabilities
  - ***tolerance analysis,***
  - ***design optimisation,***
  - ***mechanism analysis, and***
  - ***massproperty analysis***




Computer Aided
Production Management
Quality Assurance
Data Management
Computer
Aided Design
Computer Aided
Engineering
Computer Aided
Manufacturing

Computer Integrated Manufacturing- CIM

- CAD presents the concept of physical description of a product on a common data base and CAM translates this definition into a tangible hardware on that database.
- CIM is an integration of CAD/CAM system that controls all the activities from the design to manufacturing to shipping of a product.
- **The entire business of a product including sales and management control is referred to as CIM.**



HISTORICAL DEVELOPMENT

- **4 major phases of development**
- **1950s: Start of interactive computer graphics,**
  - CRT (Cathode Ray Tube), NC (Numerical Control), APT (Automatically Programmed Tools)
- **1960s: Critical research period for interactive computer graphics**
  - Sketchpad by Ivan Sutherland
  - Lockheed initiated CADAM,
  - Storage tube-based turnkey system
- **1970s: Potential of interactive computer graphics was realized by industry,**
  - SIGGRAPH, NCGA, IGES,
  - Golden era for computer drafting,
  - Wireframe modeling

HISTORICAL DEVELOPMENT

- First mechanical computer by Charles babbage in 1822
- First programmable computer by Konrad in 1936
- First electronic programmable computer by Tommy Flowers in1943
- First digital computer by John Vincent in 1942

- First stored program computer in 1949
- First IBM computer 1953
- First with RAM 1955

- First Mini computer 1960
- First Desktop 1964
- First Personal computer 1975
- First Portable/laptop computer 1975

HISTORICAL DEVELOPMENT

- **1980s: CAD/CAM heady years of research,**
  - Integration, Solid modeling, synthetic curves and surfaces
- **1990s: Management of CAD/CAM capabilities**
  - CIM, EDB, PDM, CALS, VR
  - Improvement in communication medium and networking
  - Reduced cost of hardware and software
- **2000s:** Wireless transmission, Reduced cost of high performance computing, Reverse engineering - Rapid prototyping

EDB: Exchange database
VR: Virtual Reality
PDM: PRODUCT DATA MANAGAEMENT
CALS:Computer aided logistics support

Source : Ibrahim zeid

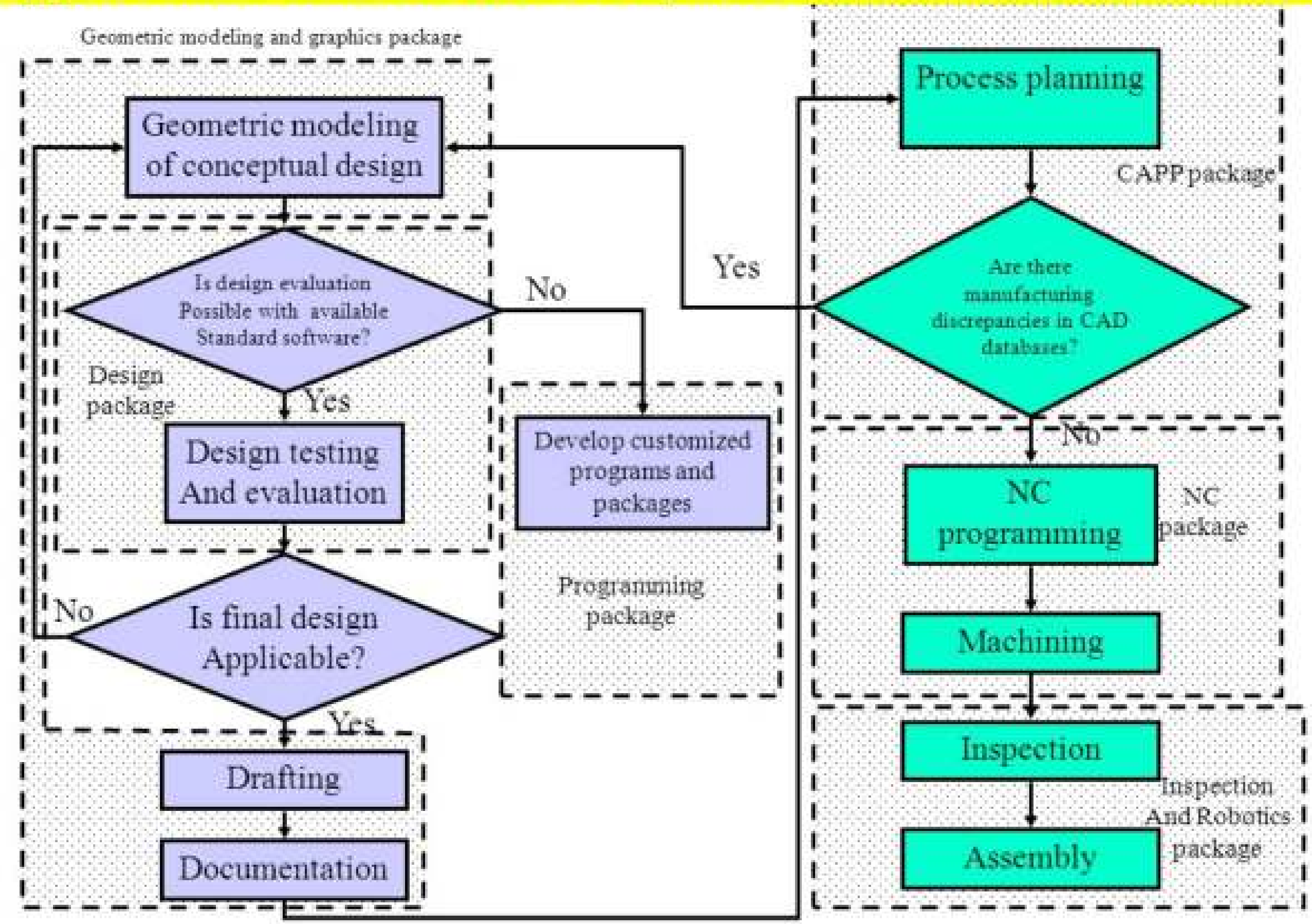
INDUSTRIAL LOOK AT CAD

- The major components or packages of CAD are,
  - Programming packages
  - Design packages
  - Geometric modeling and graphics packages
- The three available type of modeling are wireframes, surface and solid modeling
- Graphics encompass such functions as geometric transformation, drafting and documentation, shading coloring and layering.
- The design application includes mass property calculations, finite element modeling and analysis, tolerance stack analysis, mechanism modeling and interference checking.
- If a design or manufacturing application encountered where the systems standard software cannot be utilized, a customized software may be developed using programming language provided.
- Ones the design is completed , drafting and documentation are performed on the model database.

Source : Ibrahim zeid

INDUSTRIAL LOOK AT CAD

Typical Utilization of CAD/CAM Systems in an Industrial Environmen



Source : Ibrahim zeid

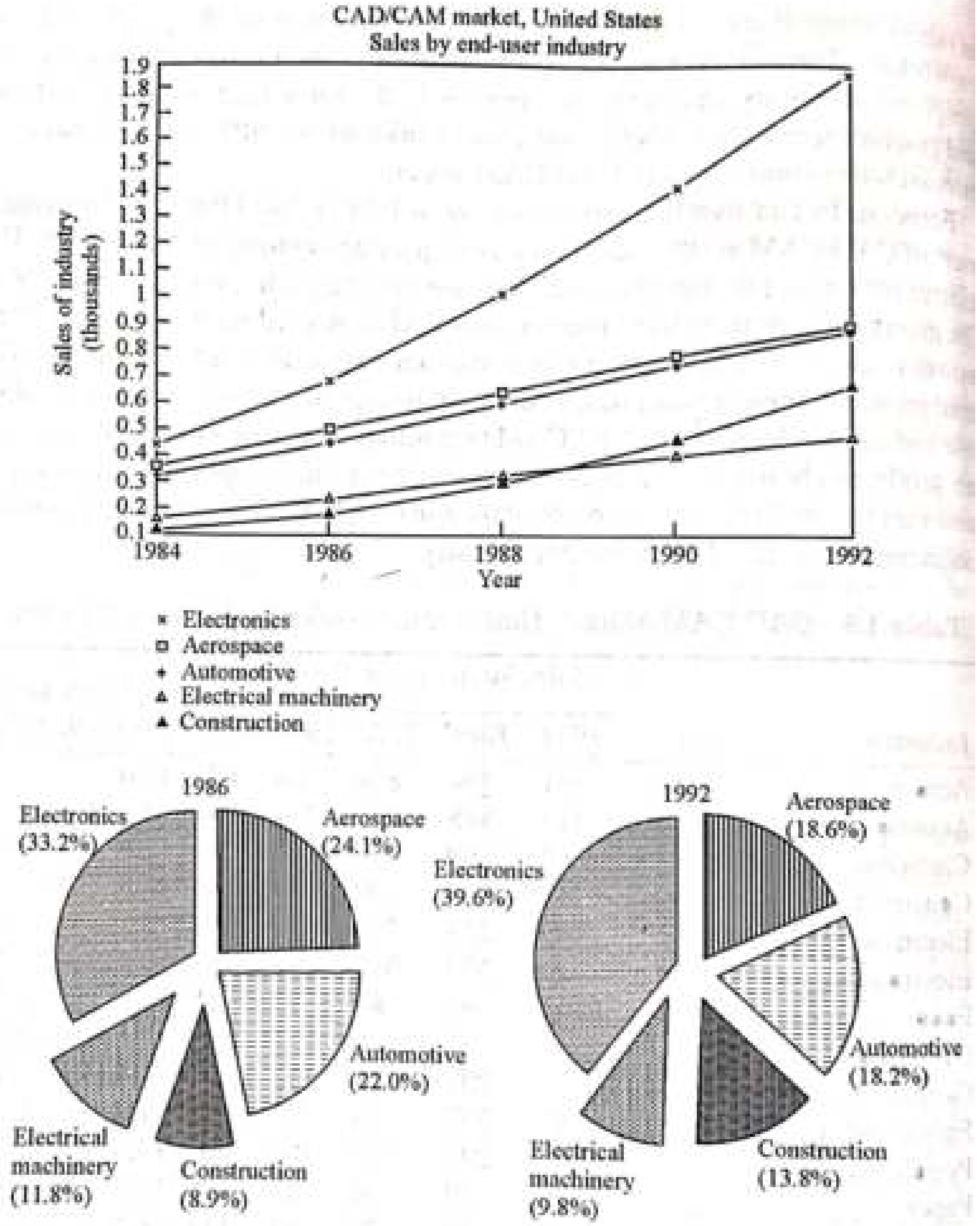


Fig. 1.4 *CAD/CAM Market by End-user Industry*

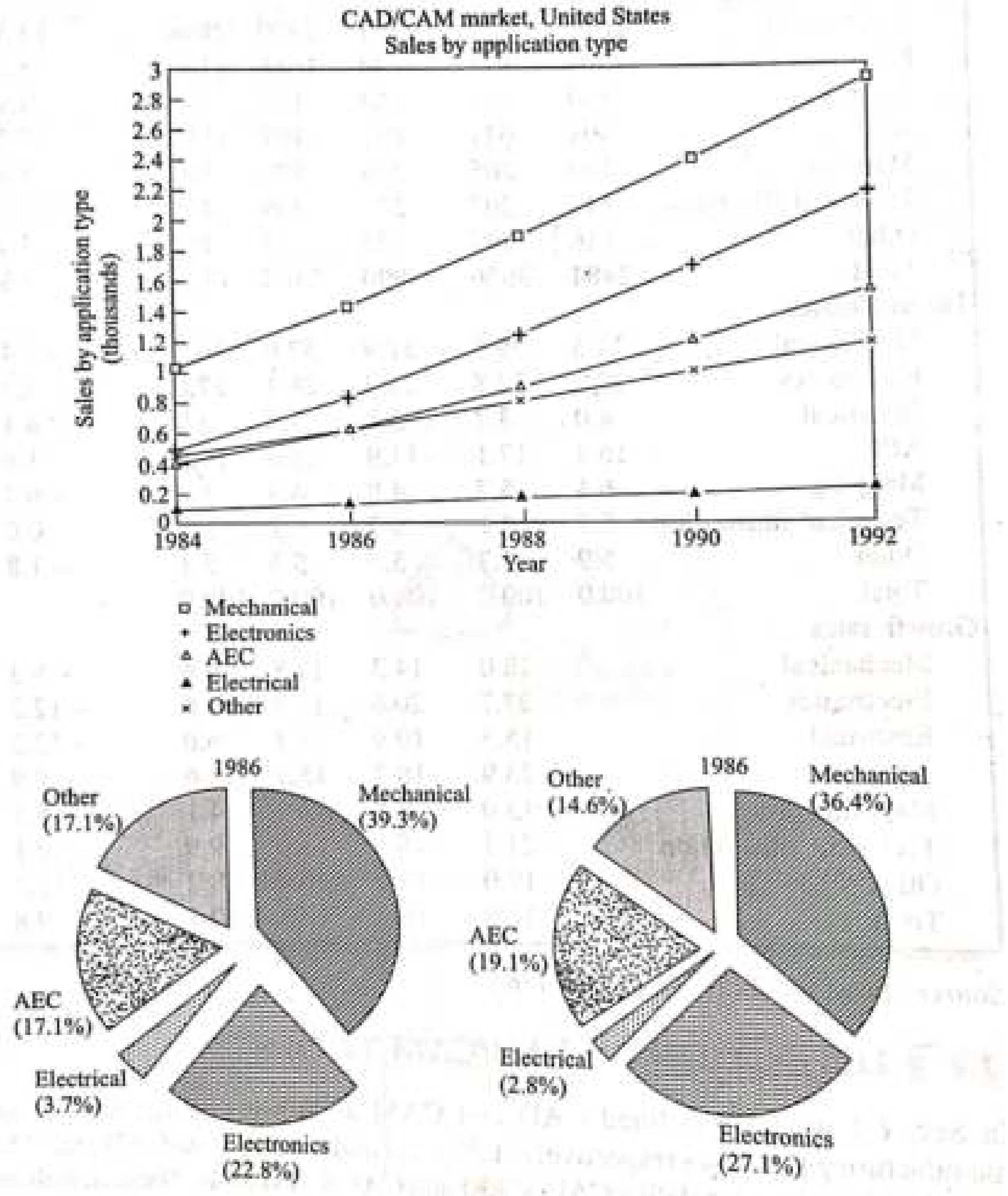
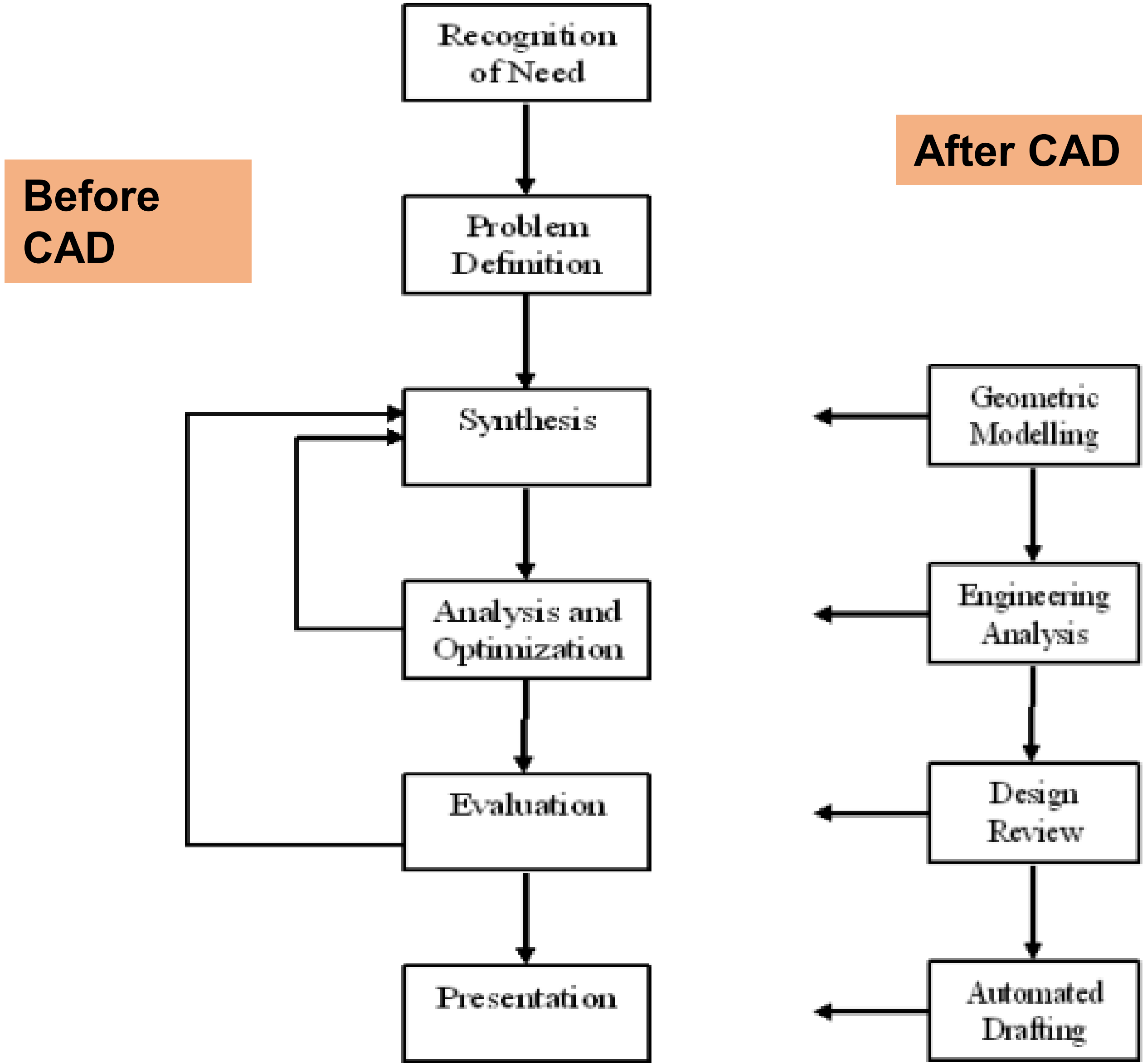


Fig. 1.5 *CAD/CAM Market by Application Type*

Source : Ibrahim zeid

COMPARISON OF CAD WITH TRADITIONAL DESIGNING



Source : Sunil kumar

COMPARISON OF CAD WITH TRADITIONAL DESIGNING

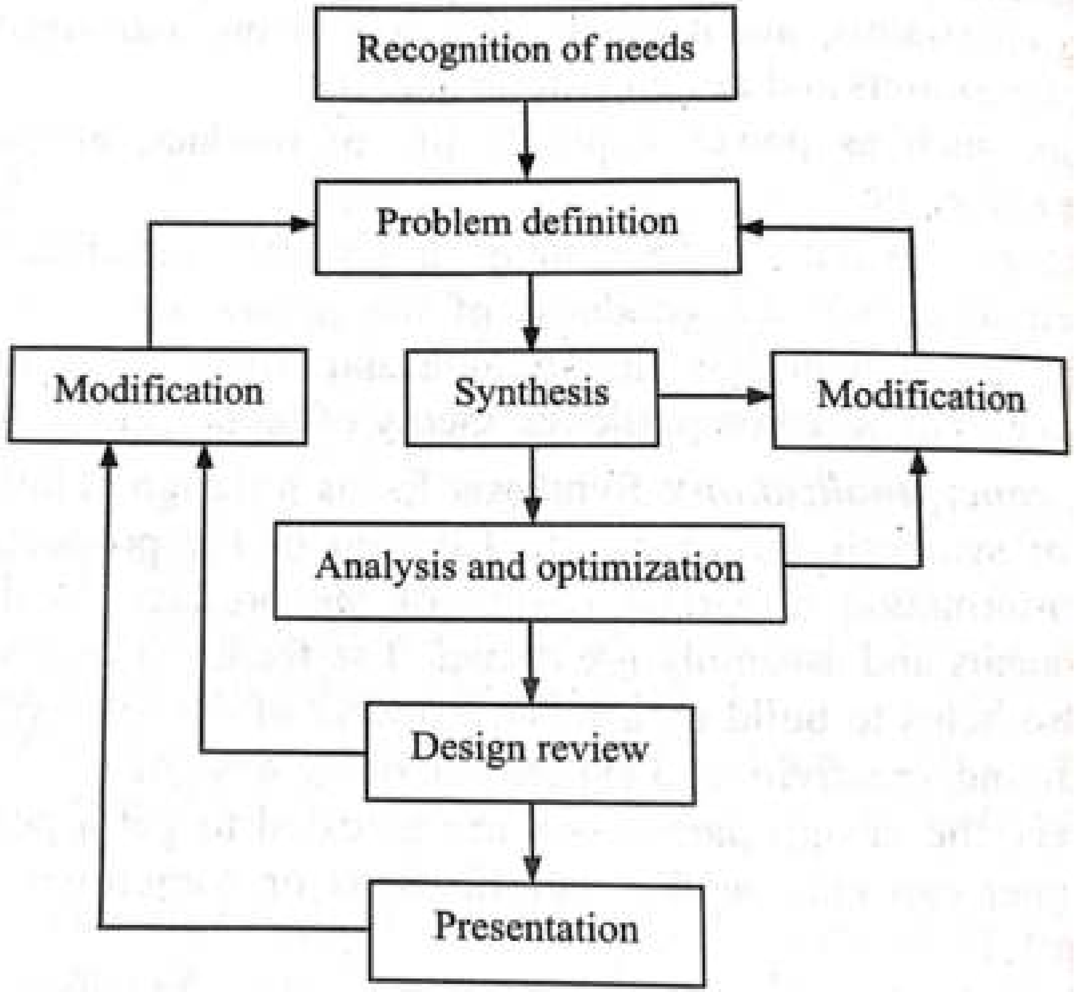


Fig. 1.2. Conventional design process

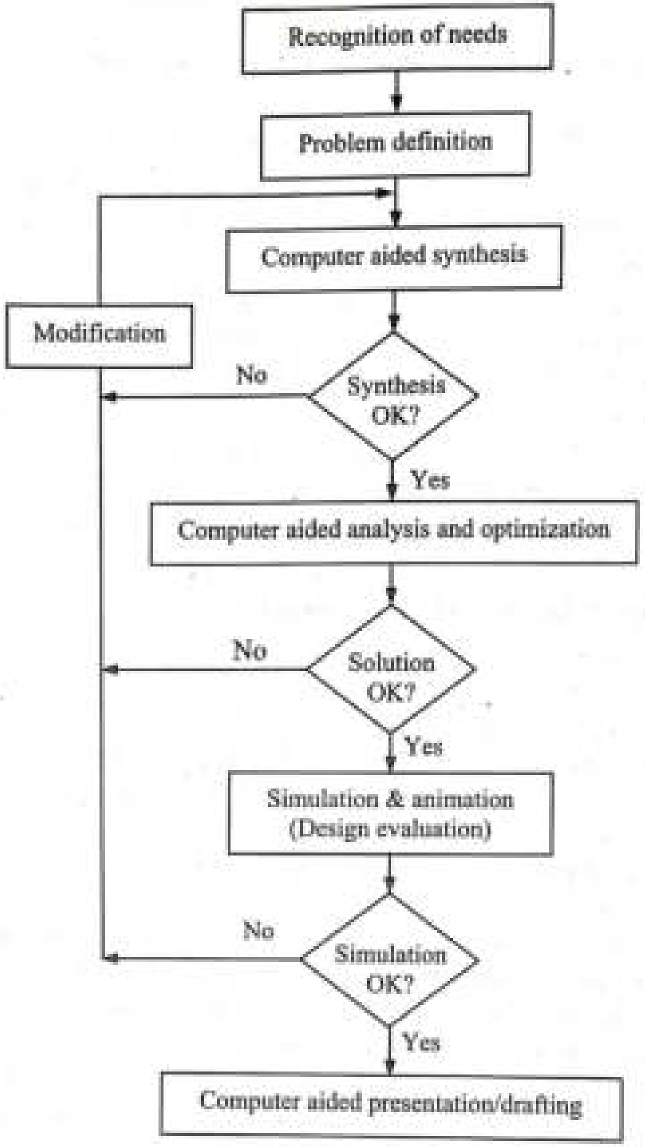


Fig. 1.4. Computer aided design process

Source : Sunil kumar

APPLICATION OF COMPUTERS IN DESIGN

1. Computer Aided Drafting
2. Geometric Modeling
3. Computer Aided assembly
4. Computer aided analysis
5. Computer aided optimization
6. Virtual Prototyping
7. Collaborative Design

Source : NPTEL By Dr. Madhusudhan Rao

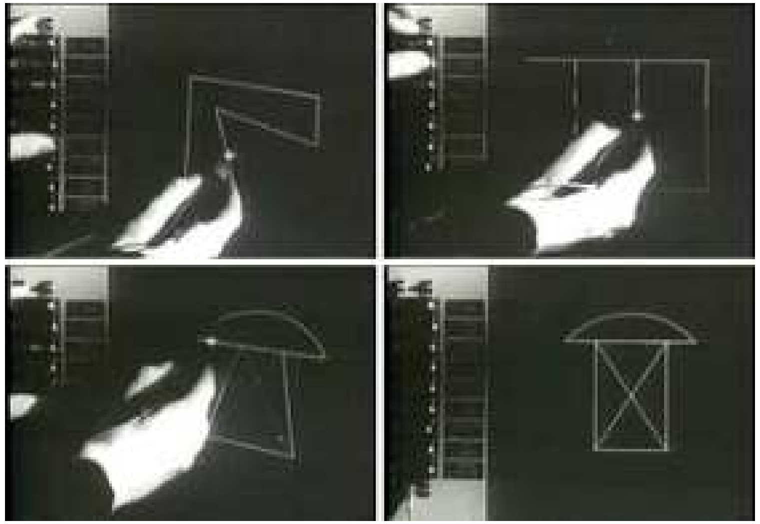
5.1 Computer Aided Drafting

- Computer aided drafting is one of the earliest applications of computer in the design process.

- It is used to store/documents/communicate all types of design such as mechanical design, architectural design, electric/electronic circuit etc.

- Computer aided drafting has many advantages over manual drafting.
- it can be used conveniently to manipulate the designs and also to store and make copies very easily compared to many of the conventional drafting which is usually done on using a drafter and drawing board.

- One of the first computer aided drafting system which was designed in 1963.
- This is called as a sketch pad which is basically, the system
- Designed by Sutherland's at MIT and this is considered to be a beginning of a computer aided design.



5.2 Geometric Modeling

- Geometric Modeling considered to be a core of CAD System.
- Geometric modeling refers to computer compactible mathematical representation of geometry. It deals with representation of curves, surface and solids.
- Geometric modeling is the basis for creating, representing, manipulating and storage of design in todays CAD systems.
- Geometric model also forms the basis for integrating design with other life-cycle activities such as manufacturing and inspection.

Geometric modeling

(computer compactible mathematical representation of geometry)

Surface modeling

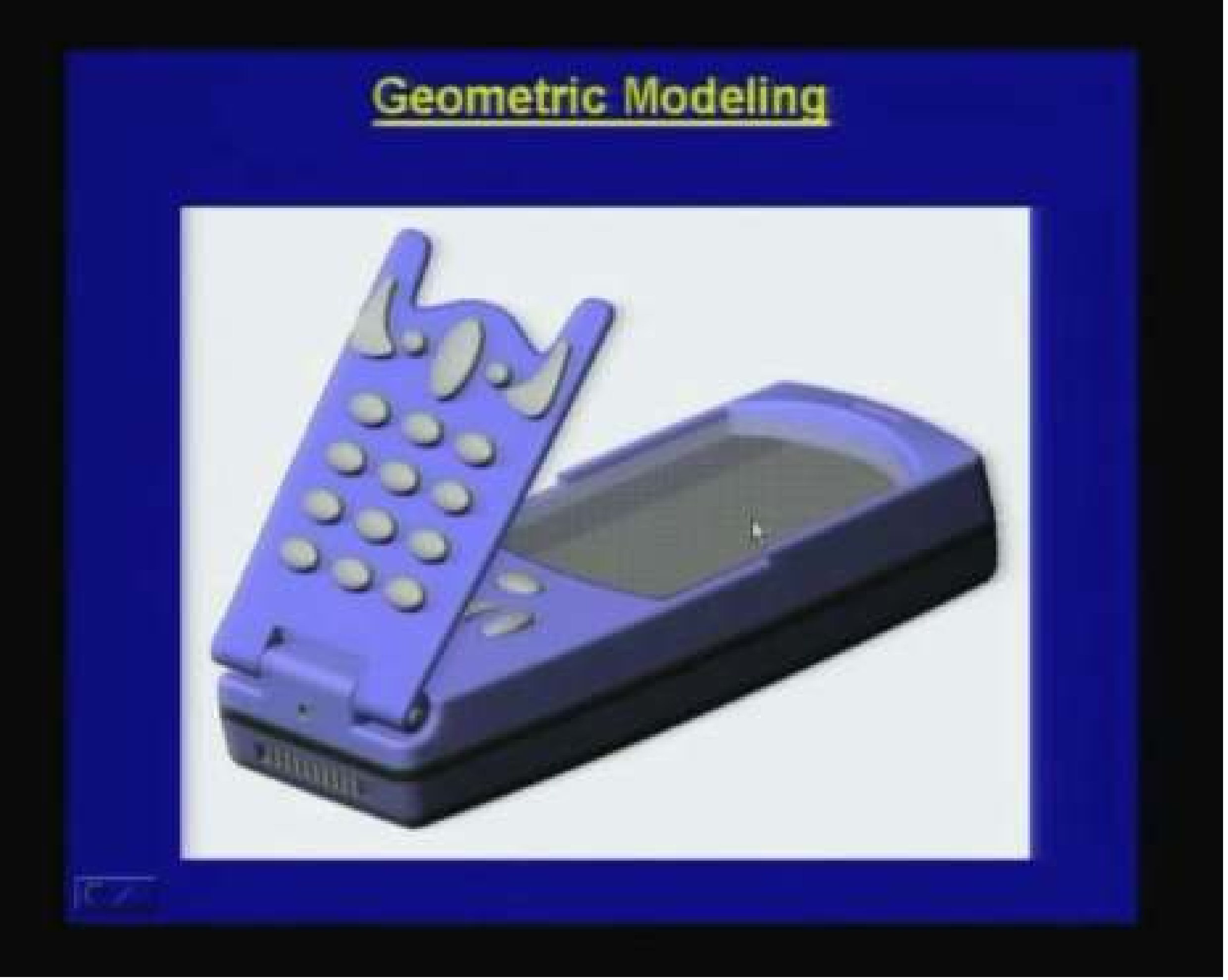
Line/Plane

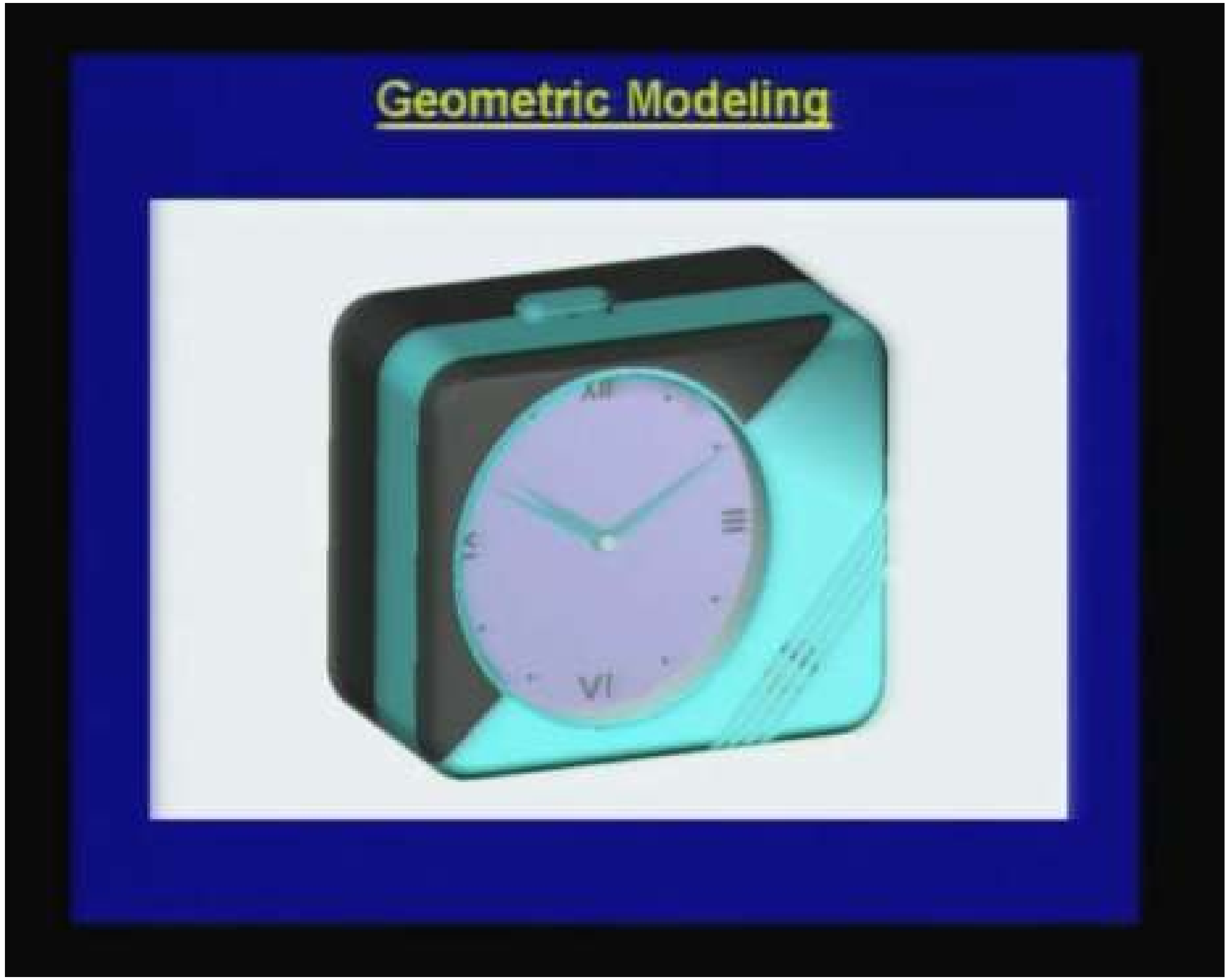
NURBS

Solid modeling

Parametric/Variational
Polyhedral
Spatial Enumeration
CSG
B-Rep

Surface and solid modelling concepts is core of any CAD system in most of the today's high end and low end CAD systems.





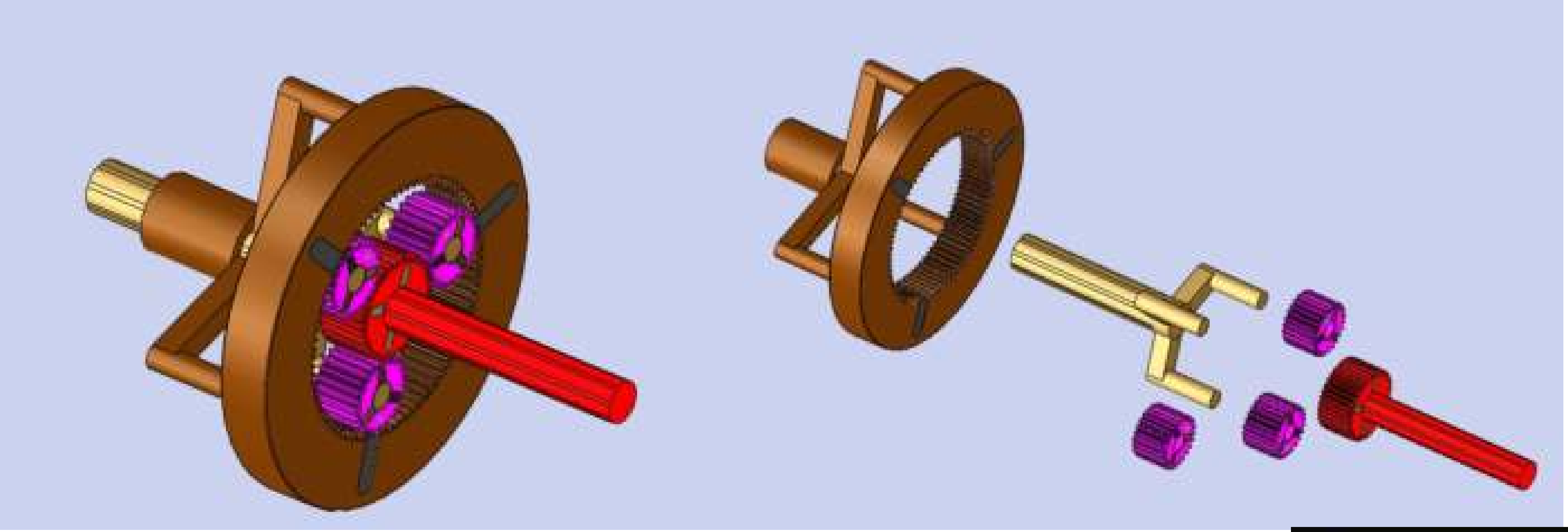
- Creo\Part1 (2).SLDPRT
- Creo\propeller.SLDPRT

5.3 Computer Aided assembly

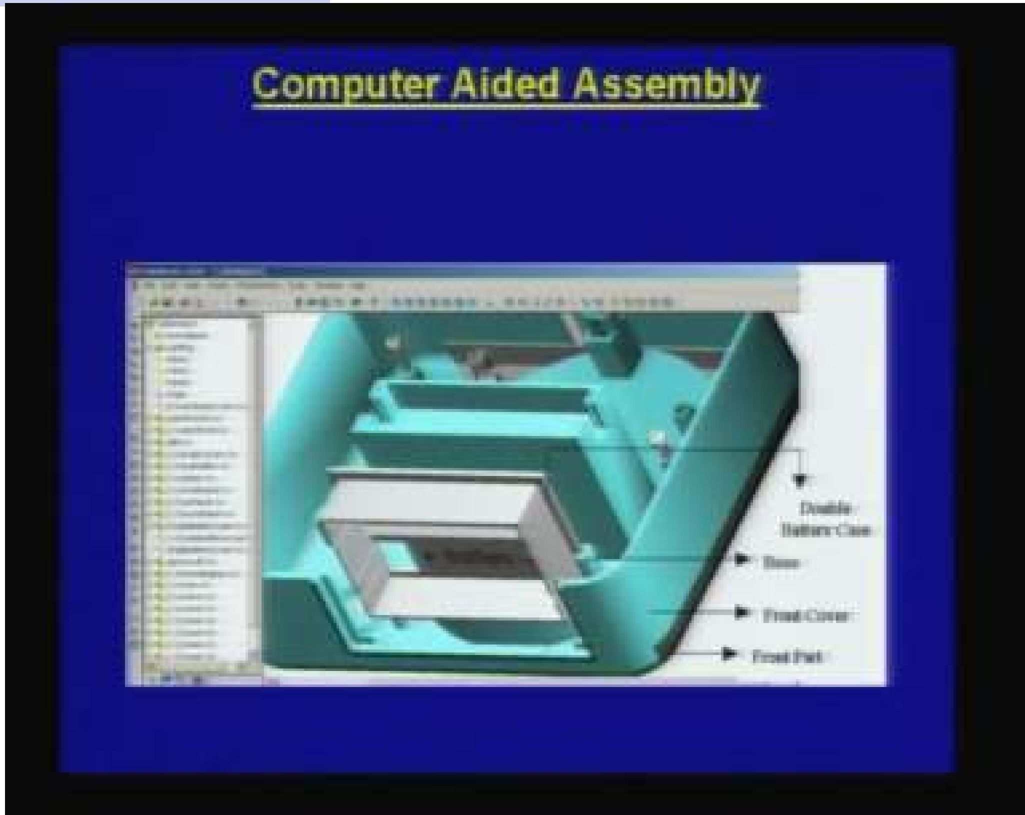
- Computer can be used to build assembly models of products by defining mating relationships between its components.
- Computer can be used to evaluate designs and redesign products for ease of assembly (DFA).
- A product configuration can be designed in number of way, then one can look or basically select or optimize from the ease of assembly point of view.
- Computer can be used to evaluate designs and redesign products for ease of dis-assembly.

- Dismantling a product into various components is really necessary for,
  - Replace a part of particular product for maintenance
  - Finally dismantle the product in terms of components for recycling once its life is over

- Qatar's shipping-container football stadium can be taken apart and reassembled after the match
- The Spanish architecture practice's modular design means the stadium can be dismantled and moved to a new location after the football tournament.

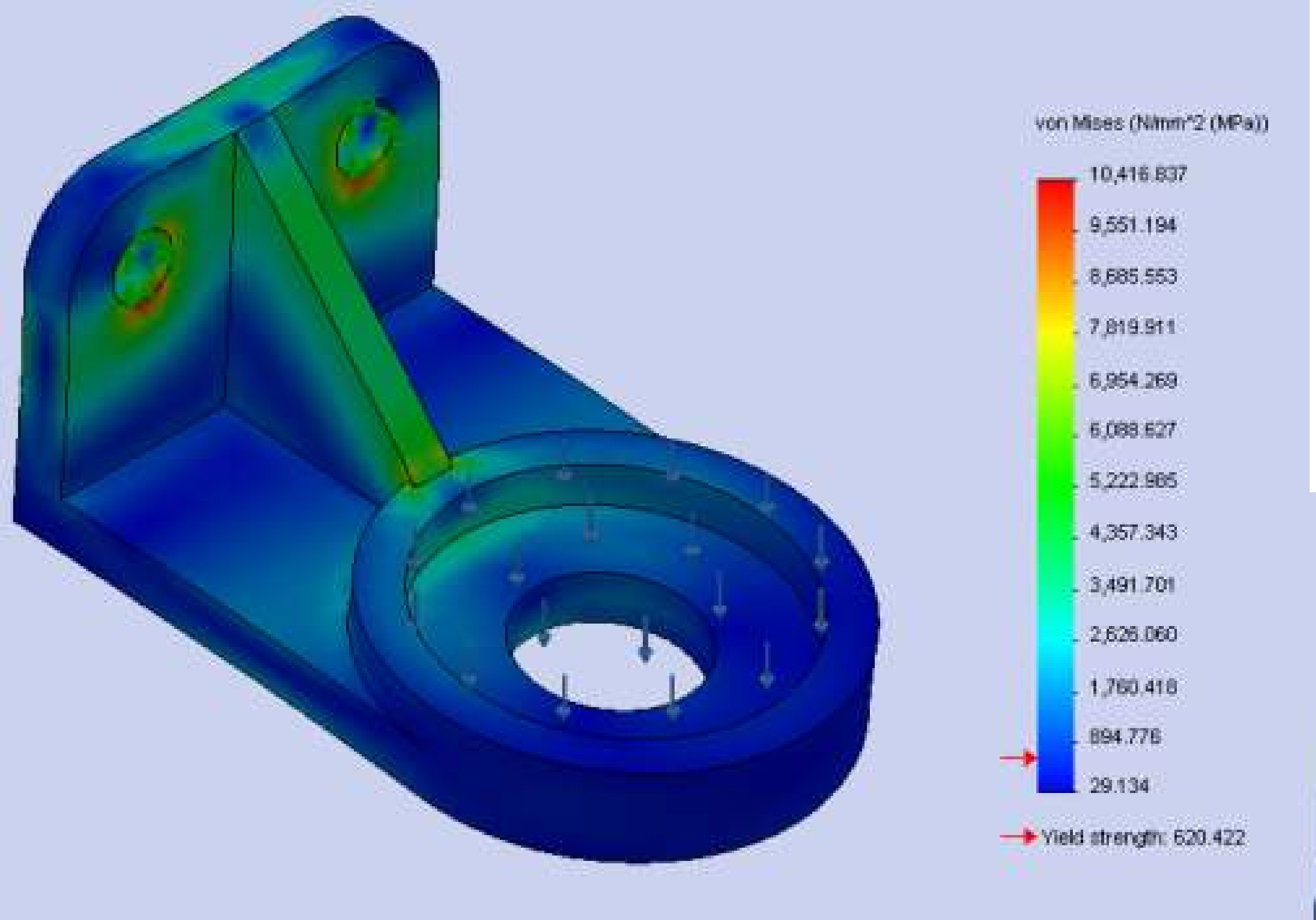


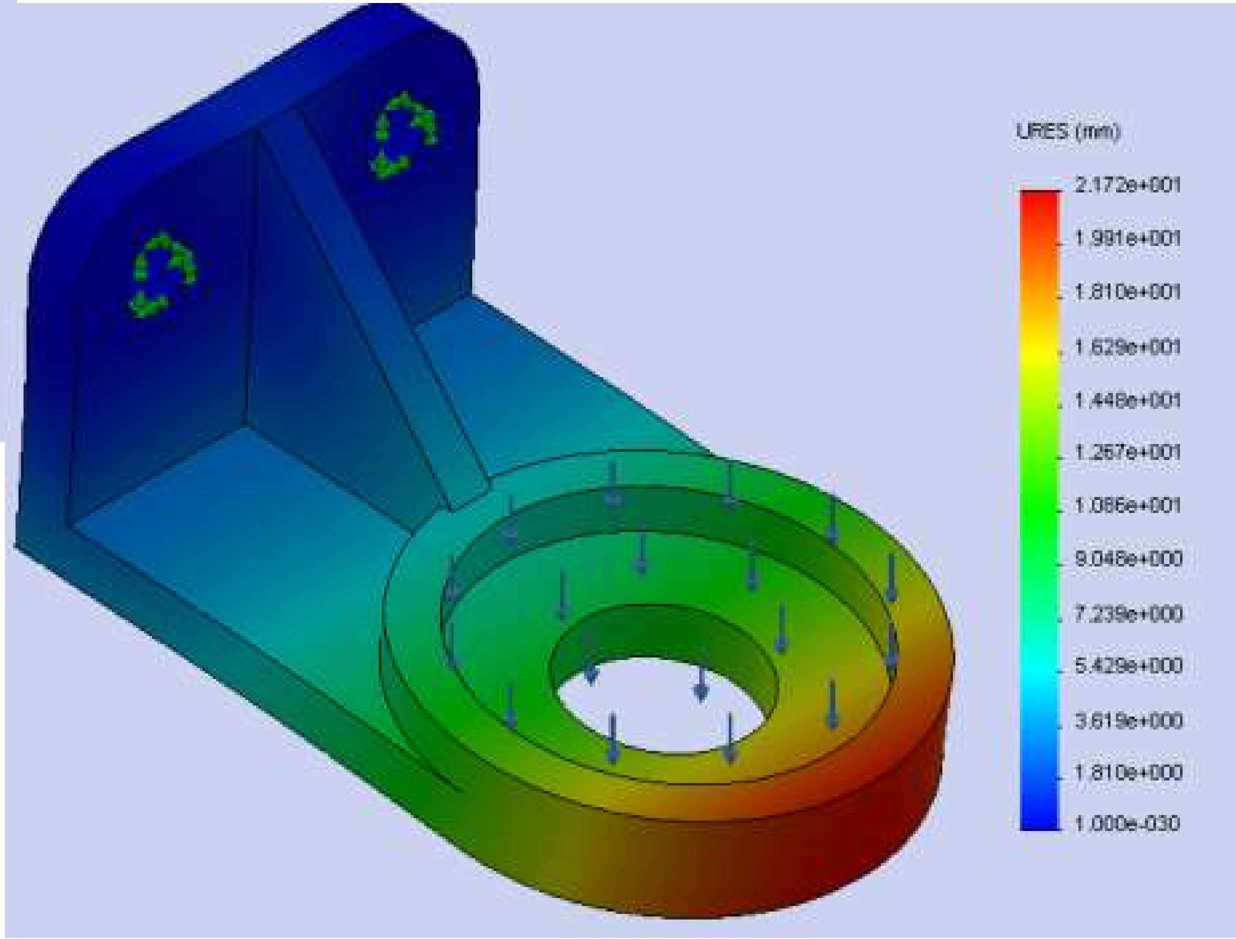
- [Assem1.SLDASM]()



5.4 Computer aided analysis

- Computer aided analysis tools are used for routine and final design checks.
- Computers are used extensively for analysis such as stress analysis, Heat transfer analysis, Fluid flow analysis, Electromagnetic analysis etc.
- The three stages of computer aided analysis consist of pre-processing, analysis and post-processing.
  - Pre-processing prepares a model or let say a finite element model for the analysis .
  - Post-processing is basically used as an interactive tool to visualize the results.





Part1 (2).SLDPRT

[redacted]

[redacted]

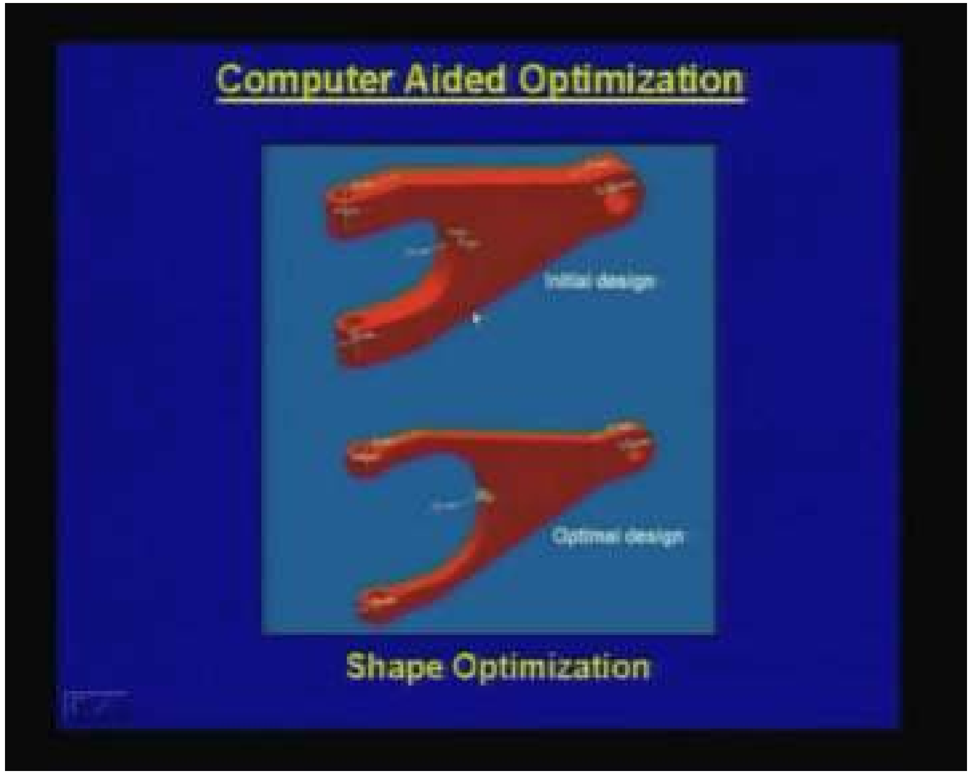
- Result of a finite element analysis which is carried out on a simple mechanical elements.
- Different colors which indicate various stress levels and deflection level .
- This gives enough feedback to designers as to what are the critical portions where the strengthening is required or what are the portions where the stresses are not very high or the deflections are not very high where one can go back and reduce the sections or may be remove material in some form in order to optimize the design.
- Analysis tools are used extensively and particularly computer aided analysis tools are commonly used in many design processes.

5.5 Computer aided optimization

- Computer are used for arriving at optimum design whenever there are many alternatives for feasible designs.

- Various types of optimizations which can be carried out include

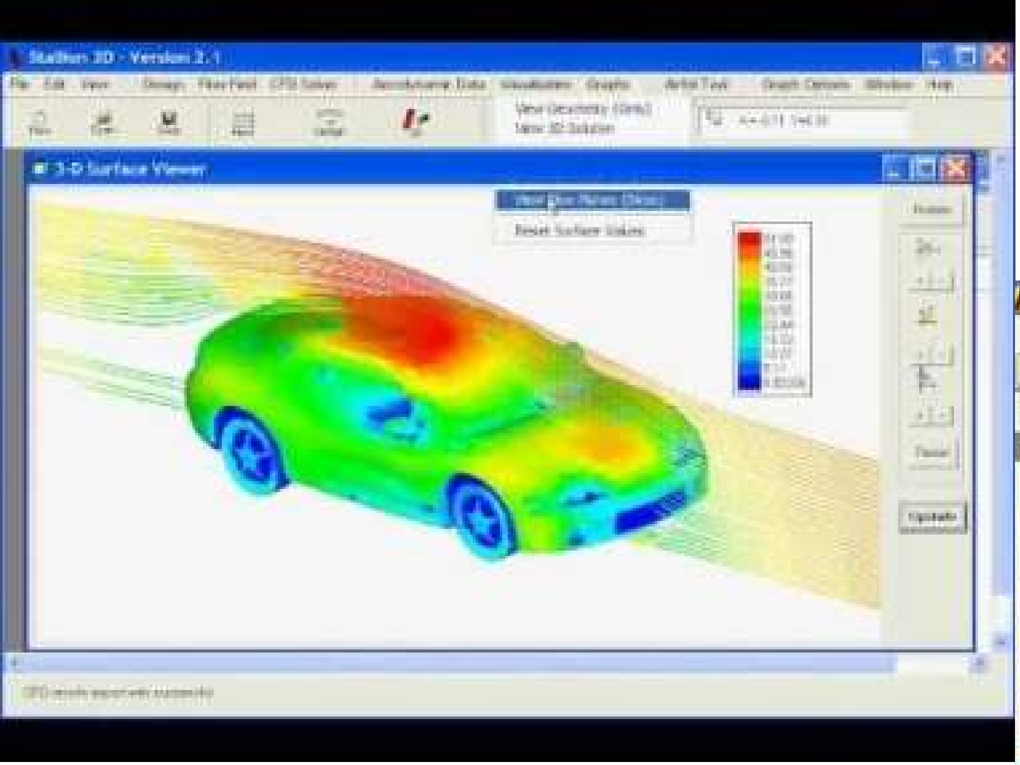
  - Parameter optimization
  - Shape and size optimization
  - Topology optimization
  - Combinatorial optimization

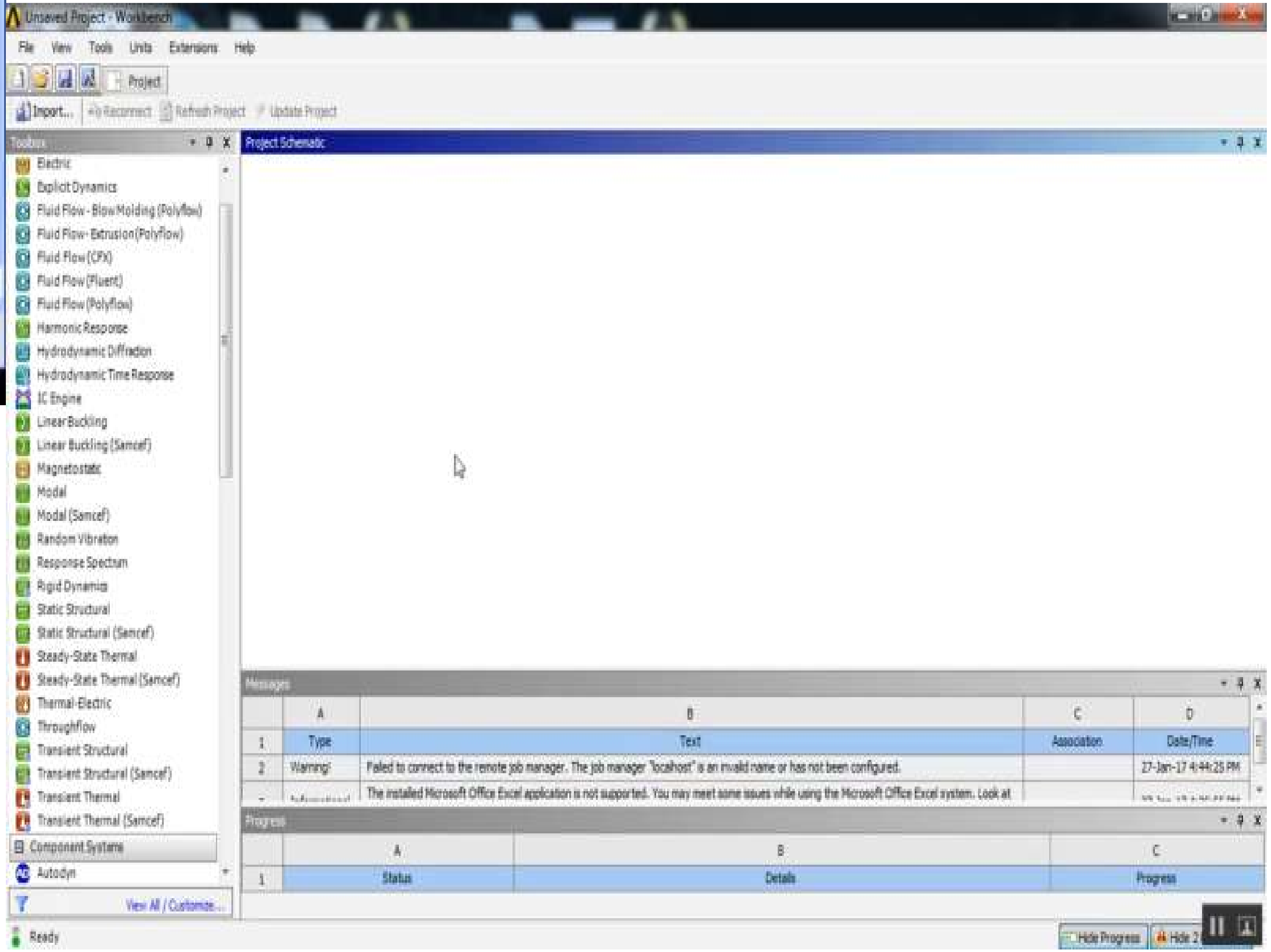


- Here in this example where there are two designs which are depicted. One is an initial design, as a part of a design process designer has arrived at a design which is a feasible design but this may not be optimum.
- What is shown here is an optimal design for the same component. Now the difference is that both of them they do the intended functions satisfactorily but the optimal design weighs much less than the original design or let's say an initial design or let's say the cost of this particular manufacturing, cost of this particular component would be much less than the other one because this uses less of raw material.

5.6 Virtual Prototyping

- Functional testing of component/products is often carried out as part of design.
- Virtual prototyping offers a quick and economical alternative to physical prototyping
- In a Virtual prototyping computers is used to model the environment in which component/product is to be used and simulate the behavior of a component/product under these environment uses laws of nature.
- In a virtual prototyping , actually not going for a physical fabrication of a product but the functionality can be tested by simulating the environment under which the component or product is supposed to work.
- The advantage of using virtual prototyping is that it saves a lot of time and cost because a physical prototyping involves investment in terms of manufacturing and then getting product, getting the components then assembling them and then carrying out the physical test.





Testing of automobile aerodynamic efficiency

5.7 Collaborative Design

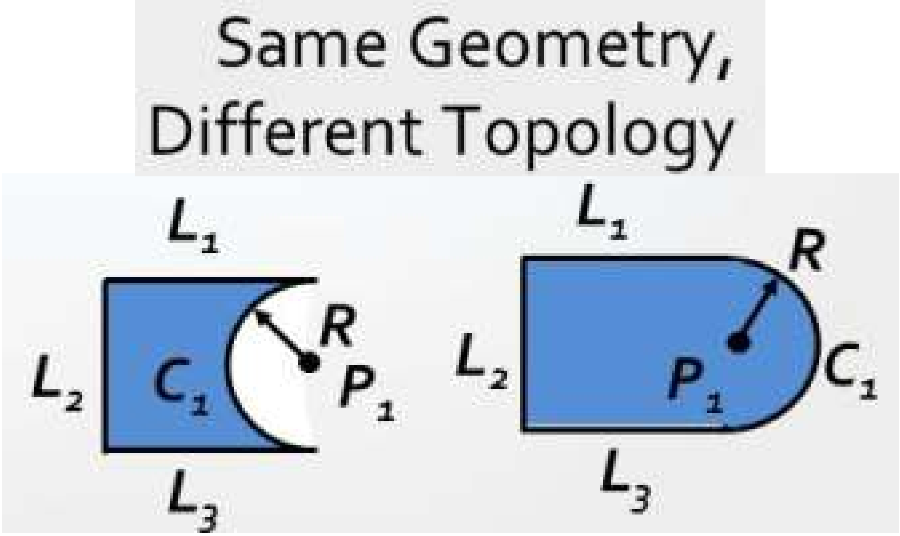
- In many cases design is collaborative involving many people who are geographically distributed participating in design process.
- Computers capability of communicating is used to carry out collaboration process.
- Collaborative design involve exchange of voice, video, and data among designers through a computer network.

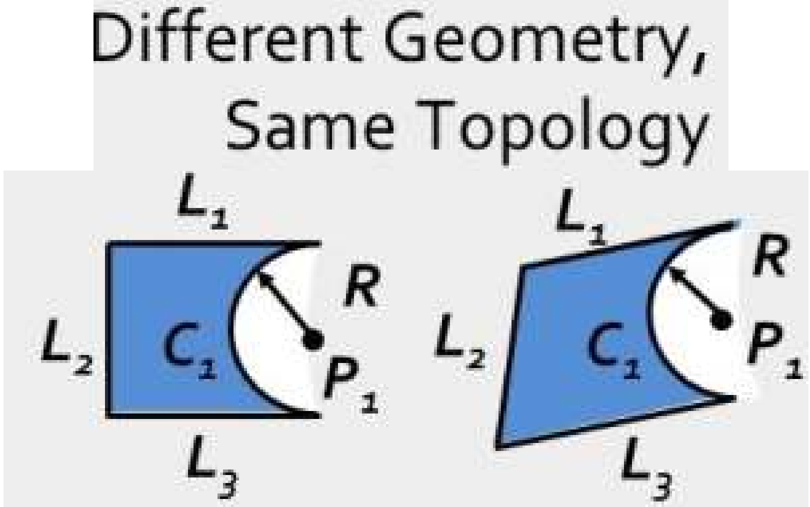
- Geometric modeling refers to computer compactible mathematical representation of geometry.
- **Computer representation of the geometry of a component using a software** ➡ Image can be displayed and manipulated through graphics terminal
- 3 types of commands to construct graphical image on CRT:
  - First type: generates basic geometry elements: point, line, circle etc..
  - Second: to accomplish scaling, rotation or other transformations on basic elements.
  - Third: causes elements to join to take desired shape of the object created on ICG
- During GM, computer converts commands into a mathematical model, stores in the data file and displays it as an image on CRT screen.
- **Types of Geometric Modelling**:
  - Wireframe Model
  - Surface Model
  - Solid Model

Source : Sunil kumar

Requirements of Geometric Modelling

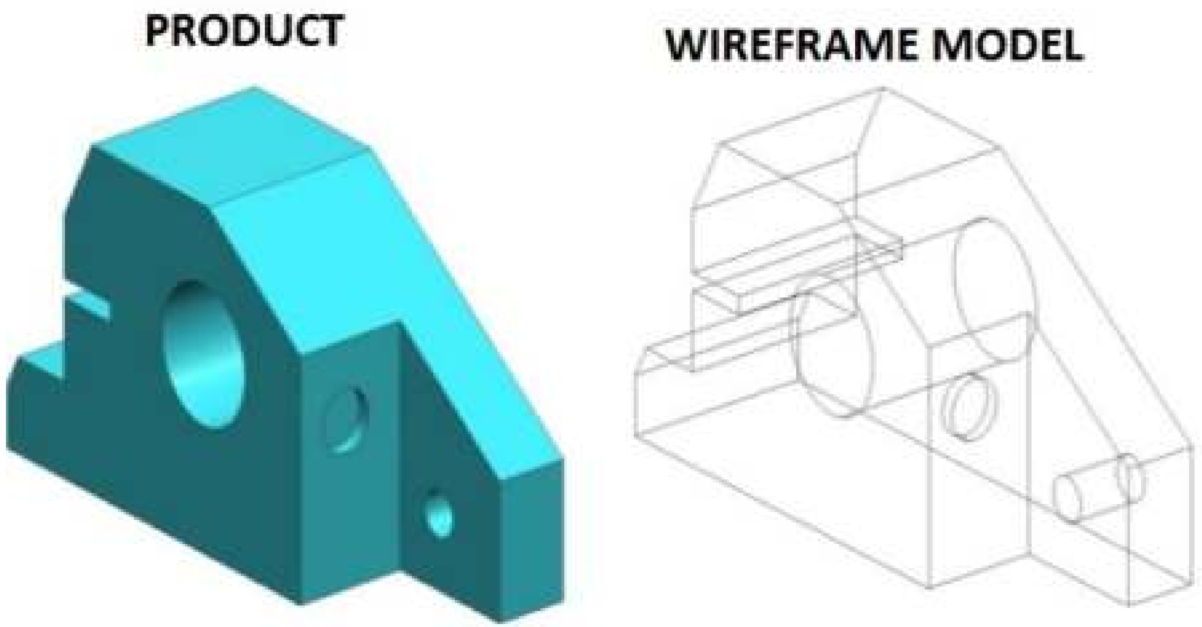
- Complete part representation including topological and geometrical data.
  - Geometry: shape and dimensions
  - Topology: the connectivity and associativity of the object entities; it determines the relational information between object entities
- Able to transfer data directly from CAD to CAE and CAM.
- Support various engineering applications, including Mass property analysis, FEA etc.



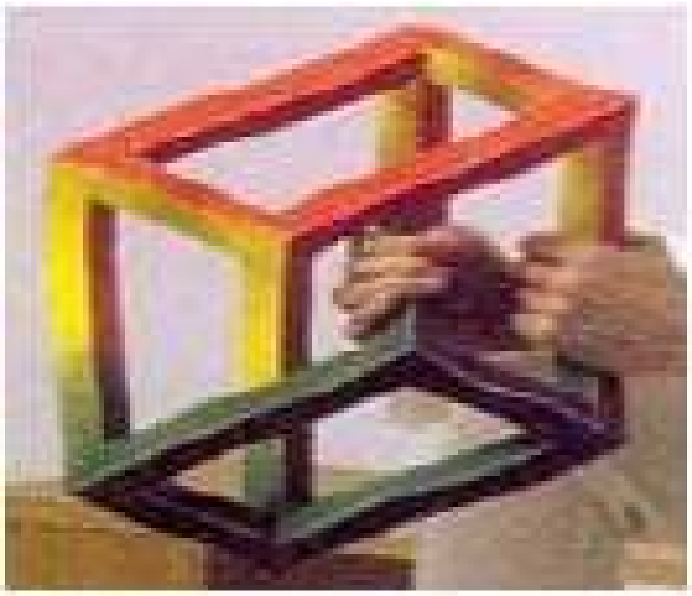


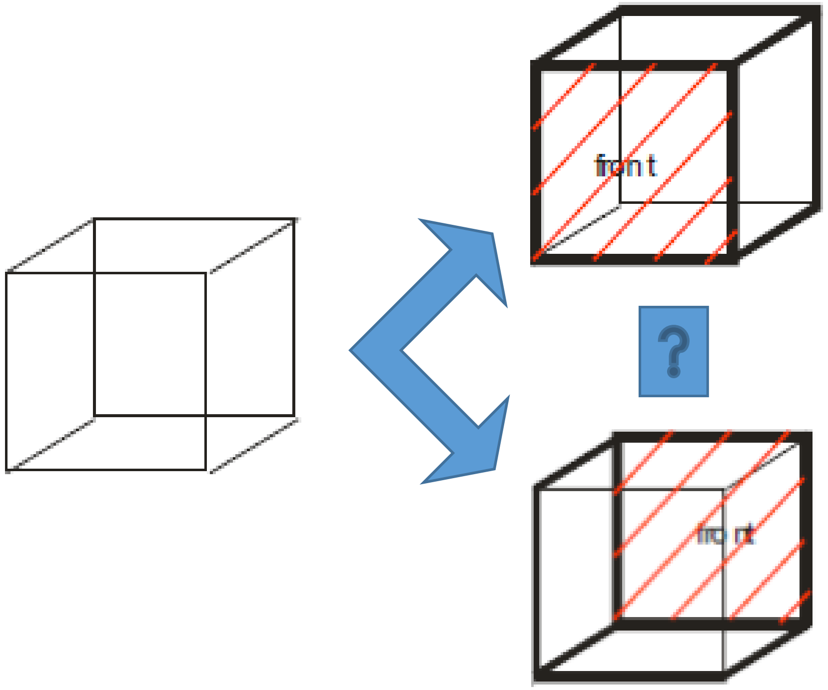
Wireframe Modelling

- Object is represented by its edges. The object appears as if it is made out of thin wires.
- In initial stages, wire frame models were in 2-D and 2 ½ D. Subsequently 3-D wire frame modeling software was introduced.

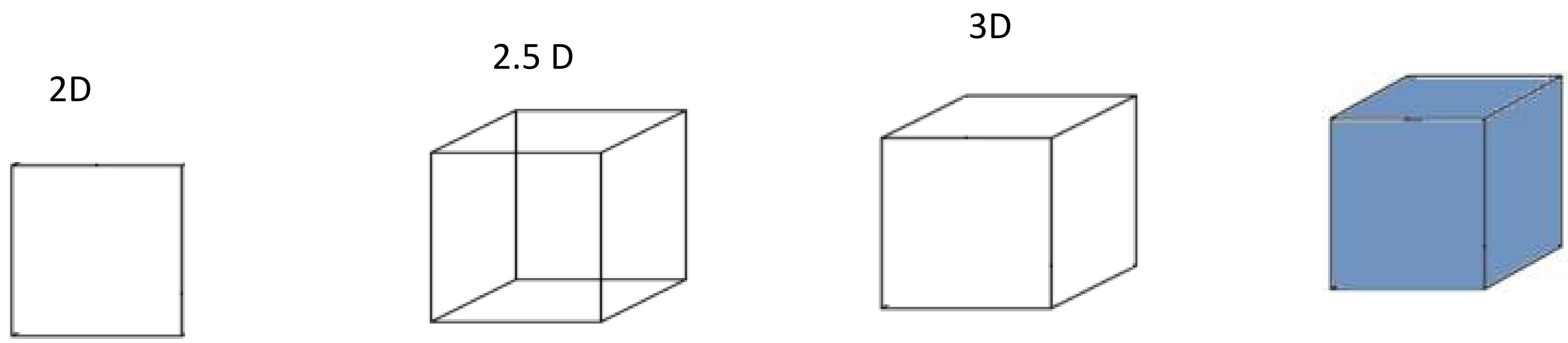


- Developed in 1960s and referred as “a stick figure” or “an edge representation”
- The word “wireframe” is related to the fact that *one may imagine a wire that is bent to follow the object edges to generate a model*.
- Model consists entirely of points, lines, arcs and circles, conics, and curves.

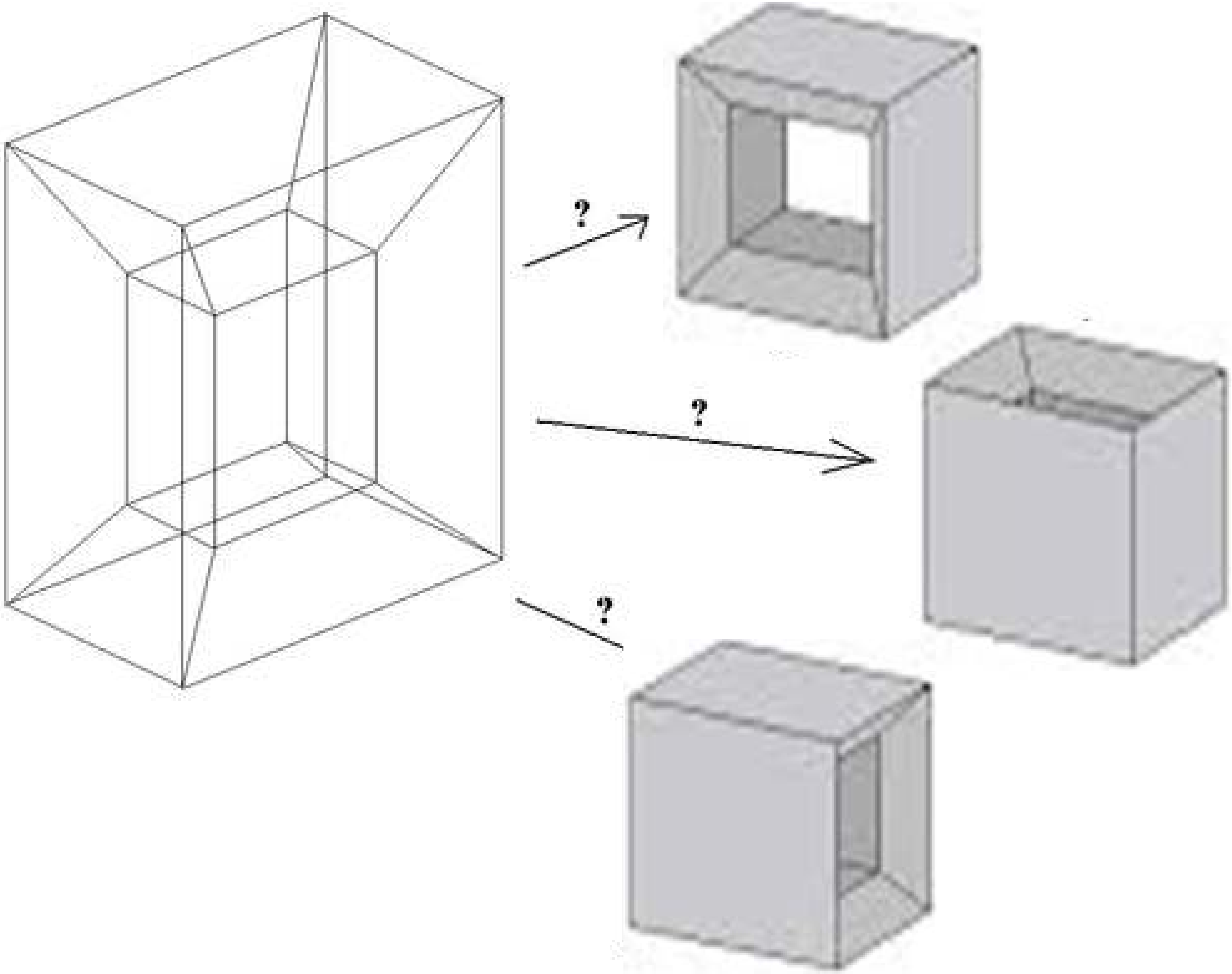




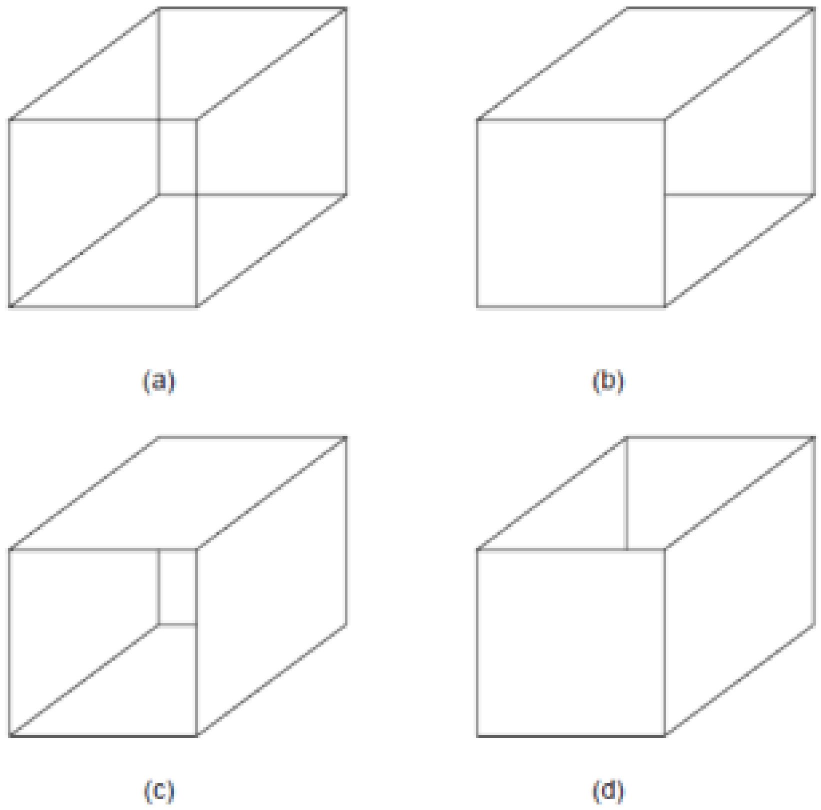
Wireframe Modelling



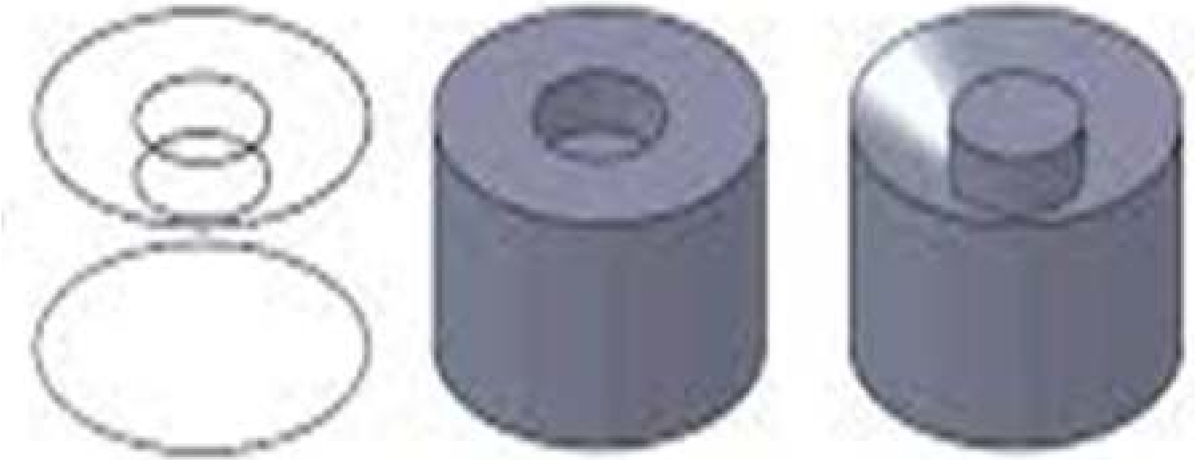
- In 3D wireframe model, an object is not recorded as a solid.
- Instead the vertices that define the boundary of the object, or the intersections of the edges of the object boundary are recorded as a collection of points and their connectivity.



Wireframe Modelling



Ambiguity in Wire Frame Modeling



Advantages

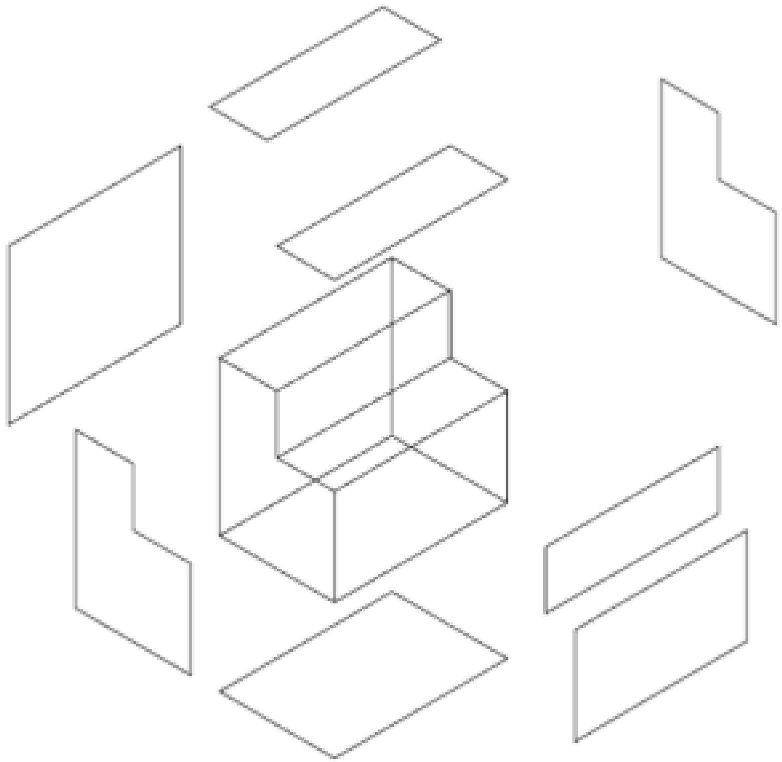
- Simple to construct
- Does not require as much as computer time and memory as does surface or solid modeling (manufacturing display)
- As a natural extension of drafting, it does not require extensive training of users.
- Form the basis for surface modeling as most surface algorithms require wireframe entities (such as points, lines and curves)

Disadvantages

- Ambiguous
- The input time is substantial and increases rapidly with the complexity of the object
- Both topological and geometric data need to be user-input; while solid modeling requires only the input of geometric data.
- Unless the object is two-and-a-half dimensional, volume and mass properties, NC tool path generation, cross-sectioning and interference cannot be calculated.

Surface Modelling

- A component is represented by its surfaces which in turn are represented by their vertices and edges.
- Surface models take the modeling of an object one step beyond wireframe models by providing information on the surface connecting the object edges.
- A surface model consists of wireframe entities that form the basis to create surface entities.
- Surface modeling :useful in the development of manufacturing codes for automobile panels and the complex doubly curved shapes of aerospace structures and dies and moulds.



Surface Representation

Surface entities

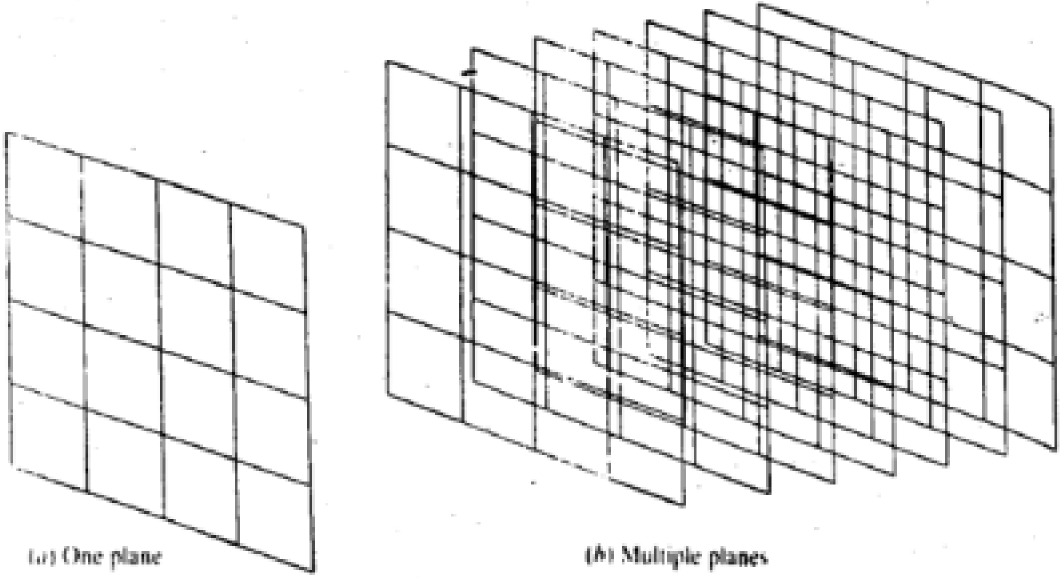
1. Analytic entities

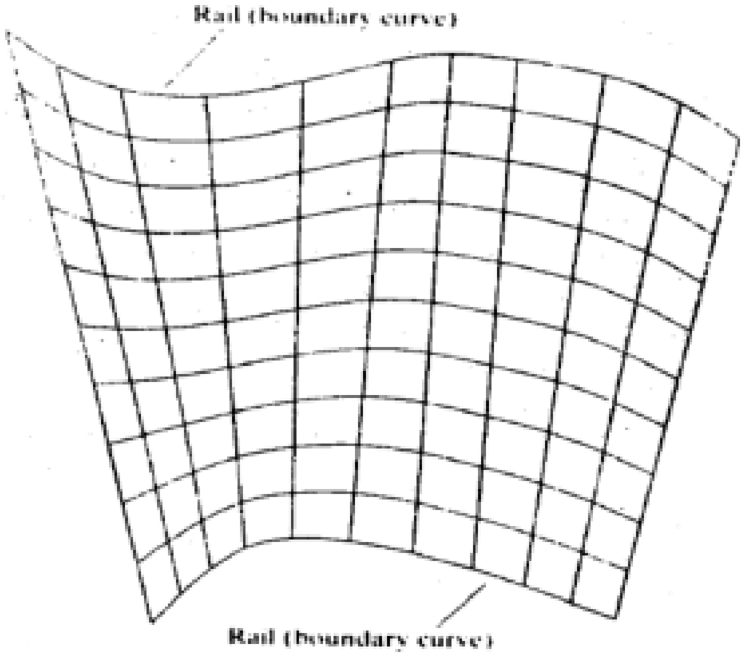
- Includes – Plane surface, ruled surface, surface of revolution and tabulated cylinder.

2. Synthetic

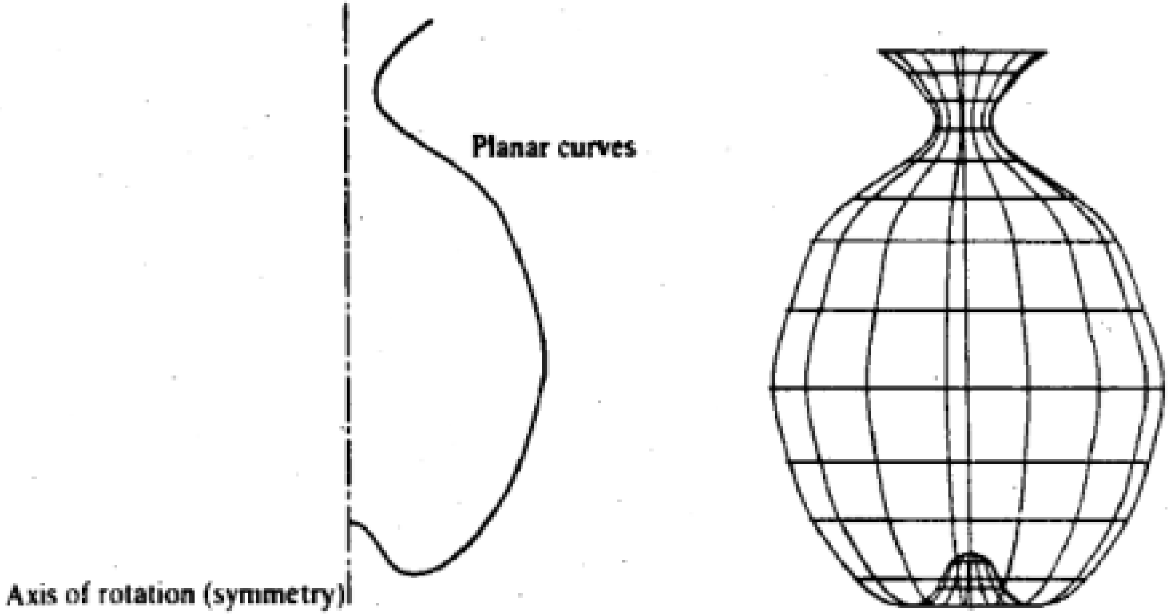
- Includes – Bicubic, Hermite spline surface, B – Spline surface, rectangular and triangular Bezier patches, rectangular and triangular Coons patches and Gordon surface

Source : ZEID

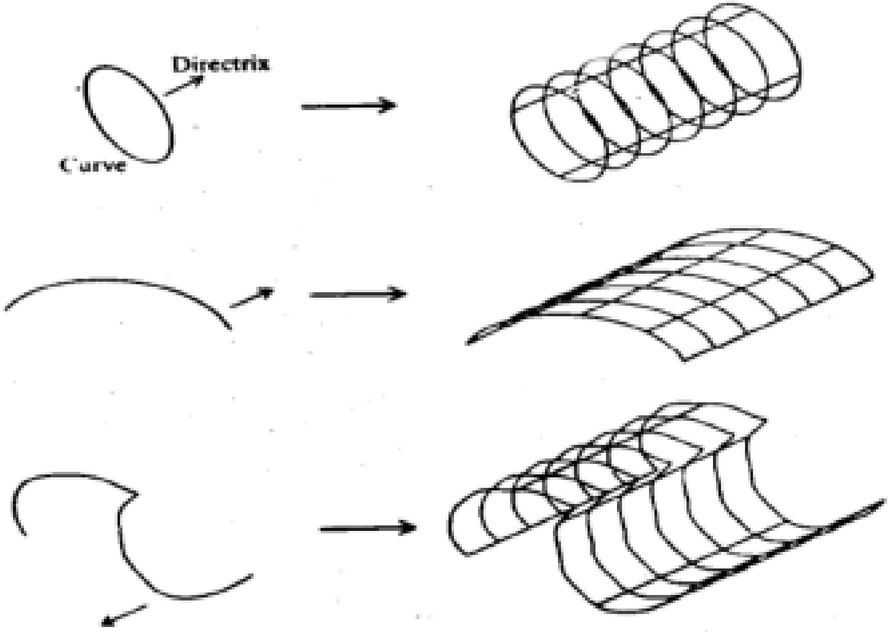




Ruled surface

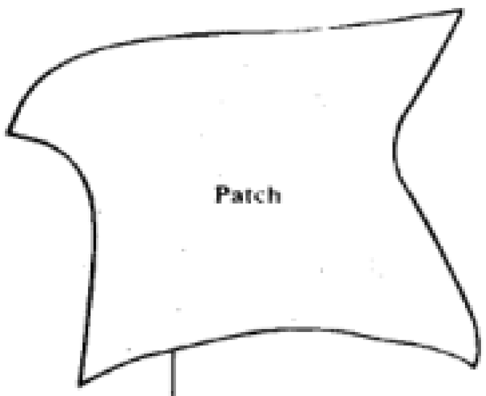


Surface of revolution

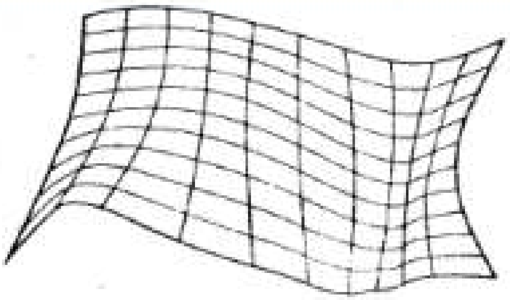


Tabulated cylinder

Coons patch

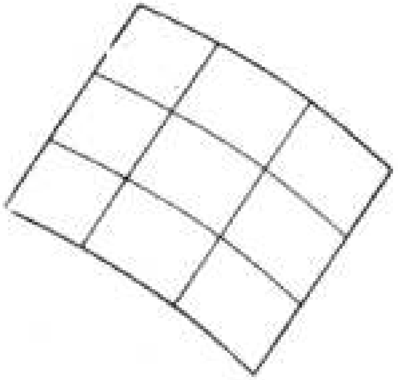


B-spline surface



B-Spline surface

Bezier surface



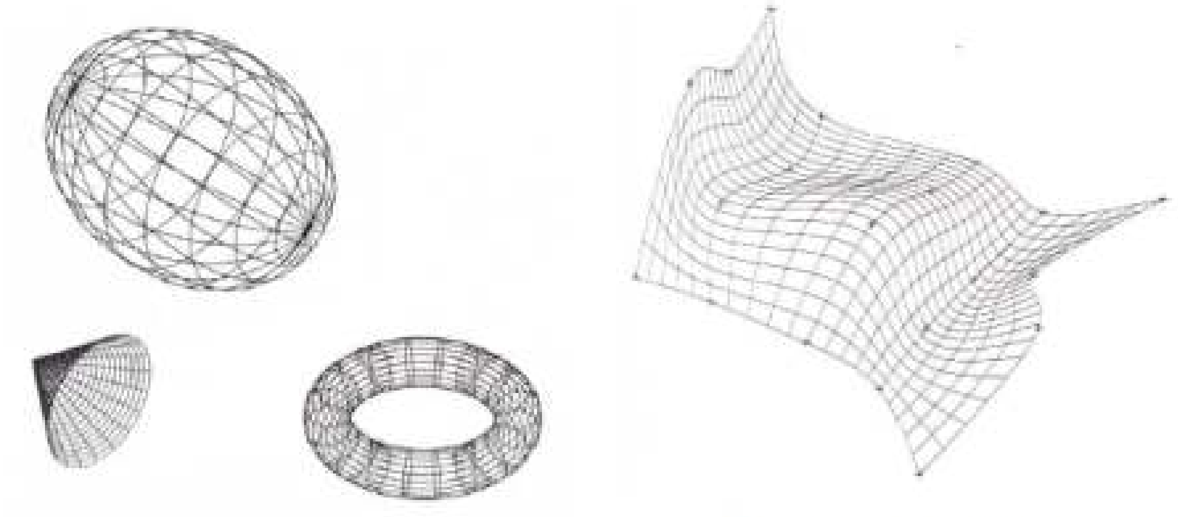
Bezier surface.

Surface Modelling

- In general, a wireframe model can be extracted from a surface model by deleting or blanking all surface entities
- Shape design and representation of complex objects such as car, ship, and airplane bodies as well as castings

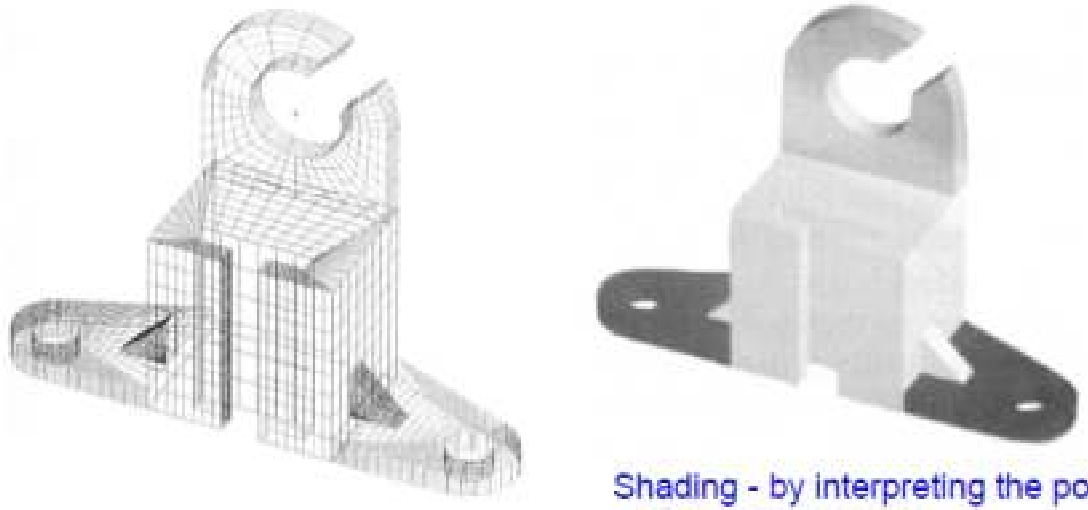
Examples of Surface Models

- Surface models define only the geometry, no topology.
- Shading is possible



Analytical Surfaces

Free-form, Curved, or Sculptured Surface



Shading - by interpreting the polygons'
- Direction (normal)
- Spatial order

Surface Modelling

Advantages:

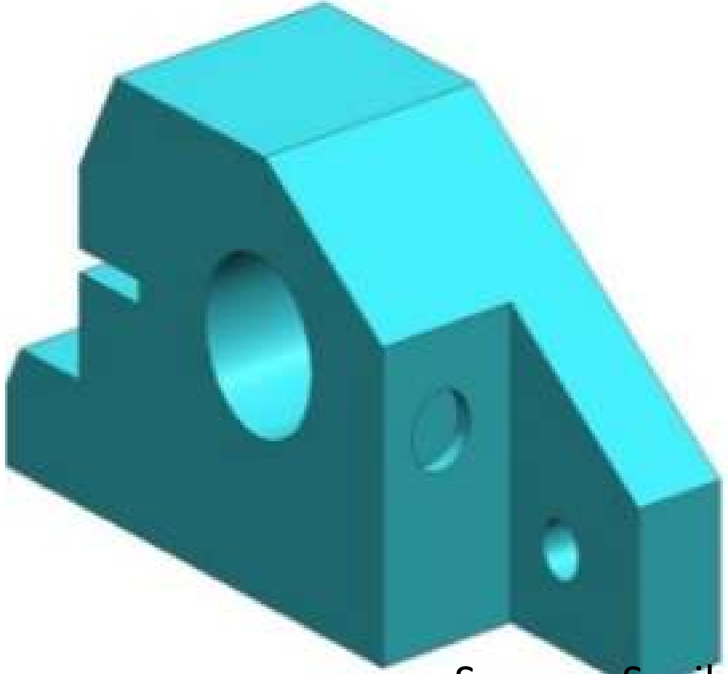
- Less ambiguous than wire frame
- Provide hidden line and surface algorithms to add realism to the displayed geometry
- Support shading
- Support volume and mass calculation, finite element modeling, NC path generation, cross sectioning, and interference detection. (when complete)

Disadvantages

- Require more training and mathematical background of the users
- Require more CPU time and memory
- Still ambiguous; no topological information
- Awkward to construct

Solid Modelling

- Models are displayed as solid objects to the viewer in 3D, with very little risk of misinterpretation.
- When color is added to the image, resulting image will be more realistic.
- Store both geometric and topological information; can verify whether two objects occupy the same space
- Solid models are,
  - Bounded
  - Homogeneous and finite



Source : Sunil kumar

Why Solid Modelling

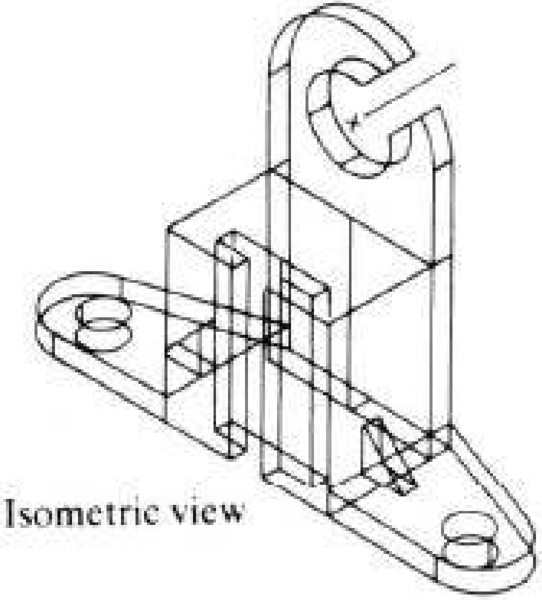
Solid Modeling Supports,

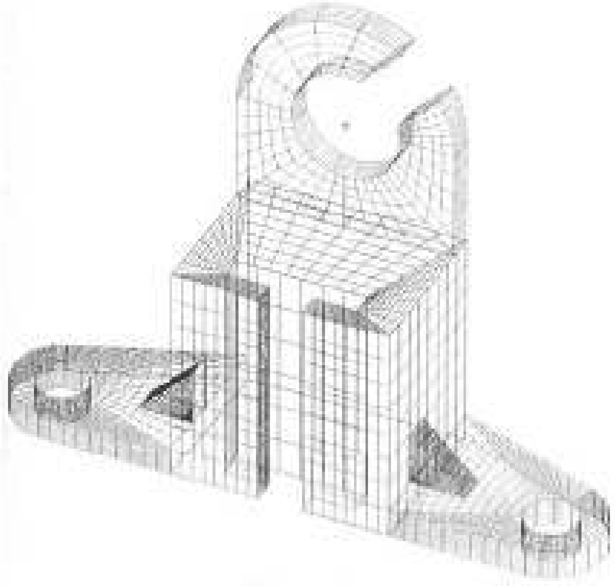
Use of volume information

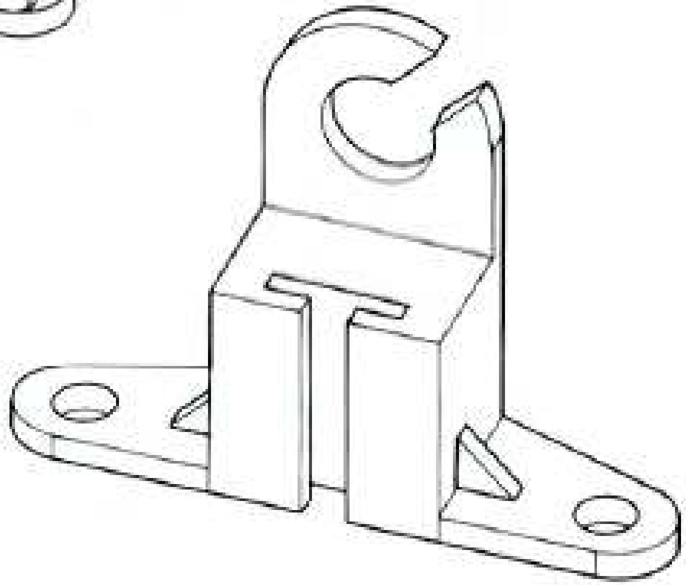
- Weight or volume calculation, centroids, moments of inertia calculation,
- Stress analysis (finite elements analysis), heat conduction calculations, dynamic analysis,
- System dynamics analysis

Use of volume and boundary information

- Generation of CNC codes, and robotic and assembly simulation







Information complete, unambiguous, accurate **solid model**

Modelling packages includes three packages,

- **Constructive solid geometry (CSG or C-Rep):**

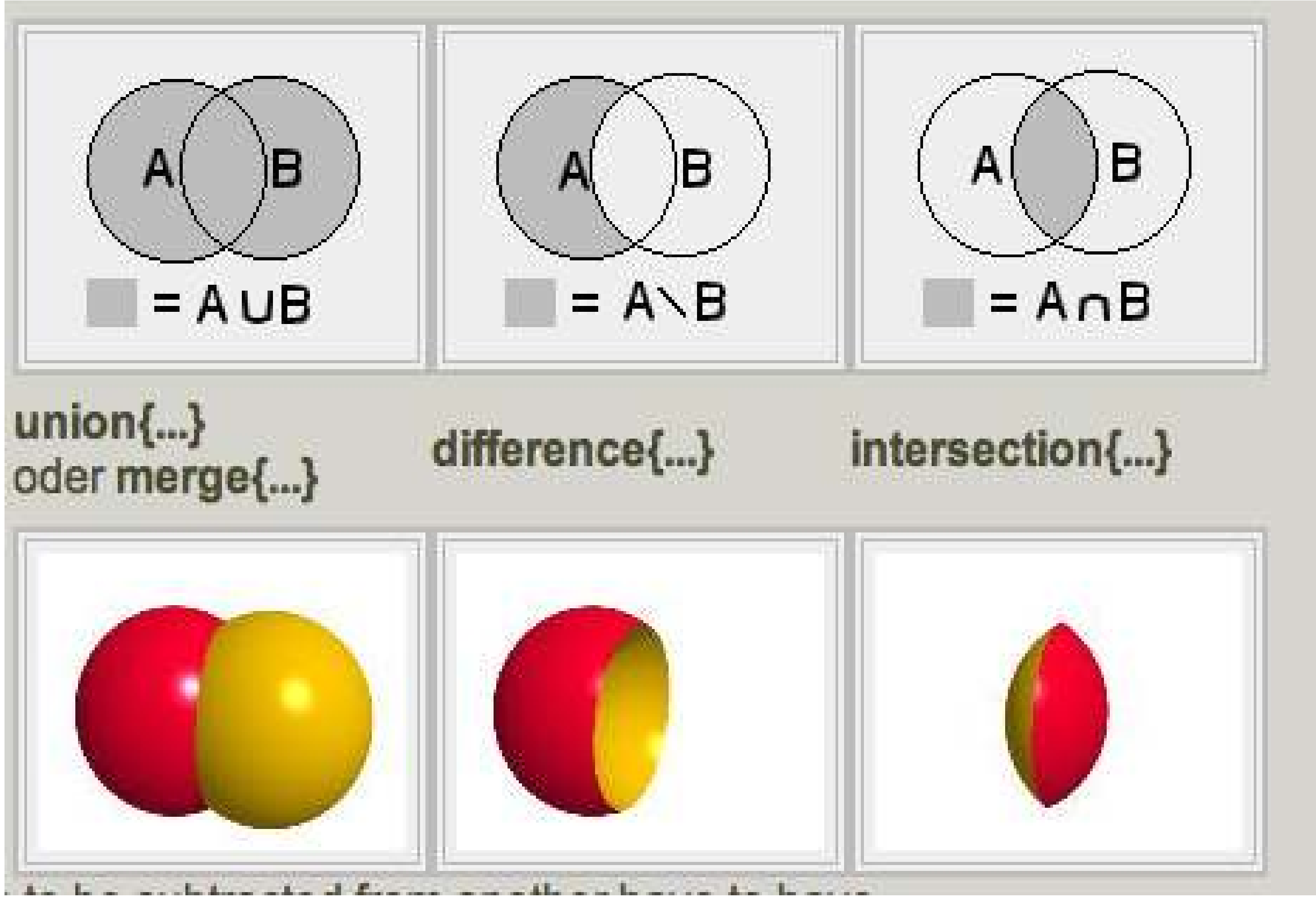
  In a CSG, physical objects are created by combining elementary shapes known as primitives like blocks, cylinders, cones, pyramids and spheres. The Boolean operations like union (∪), difference (–) and intersection (∩) are used to carry out this task.

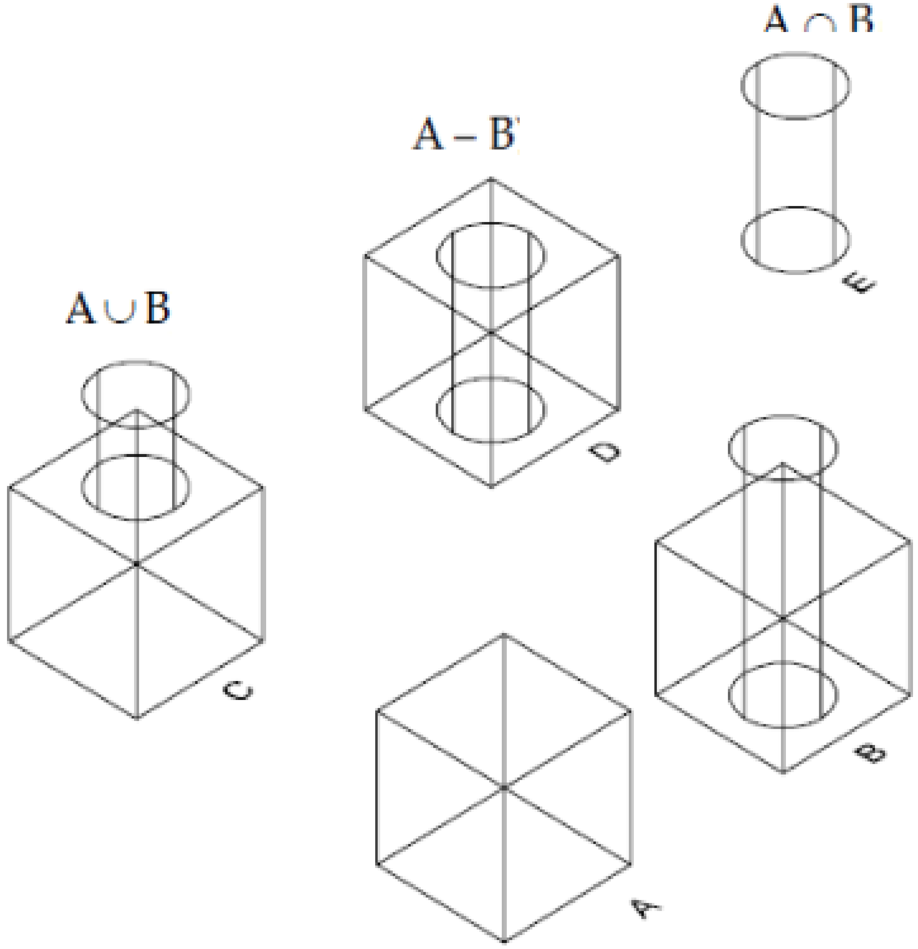
- **Boundary representation (B-Rep):**

  The solid is represented by its boundary which consists of a set of faces, a set of edges and a set of vertices as well as their topological relations.

- **Sweep Representation**

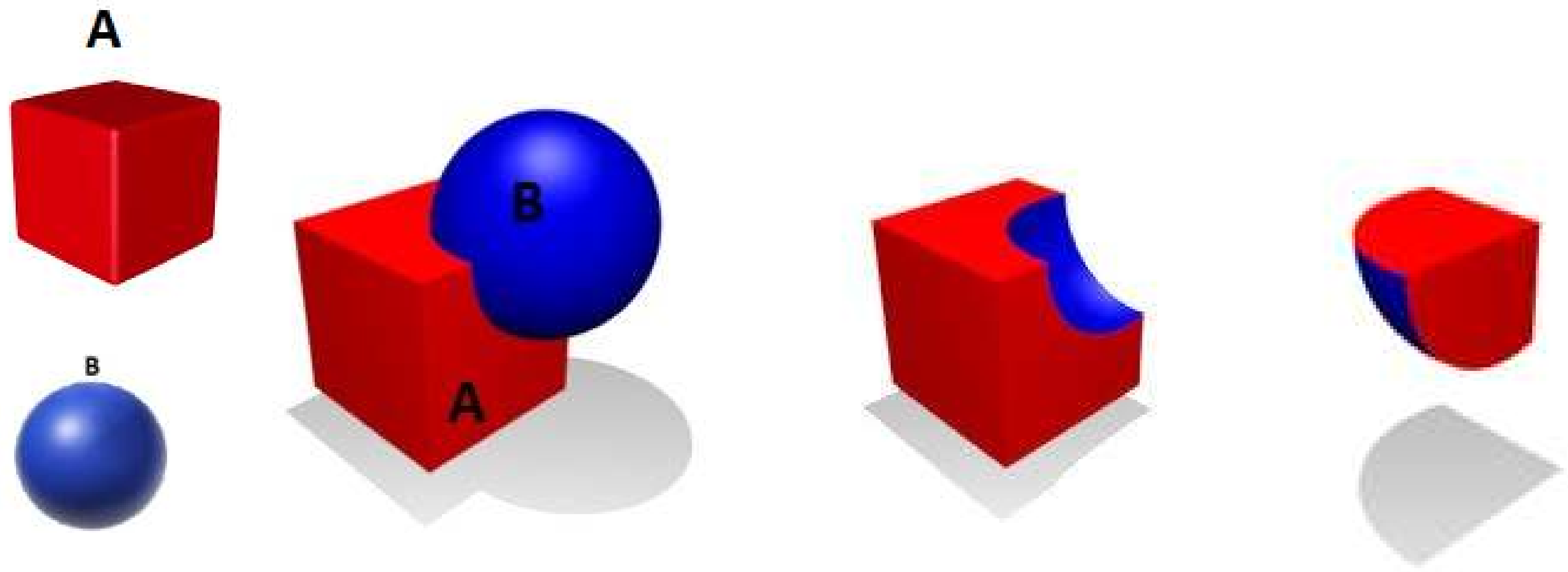
Constructive Solid Geometry (CSG)

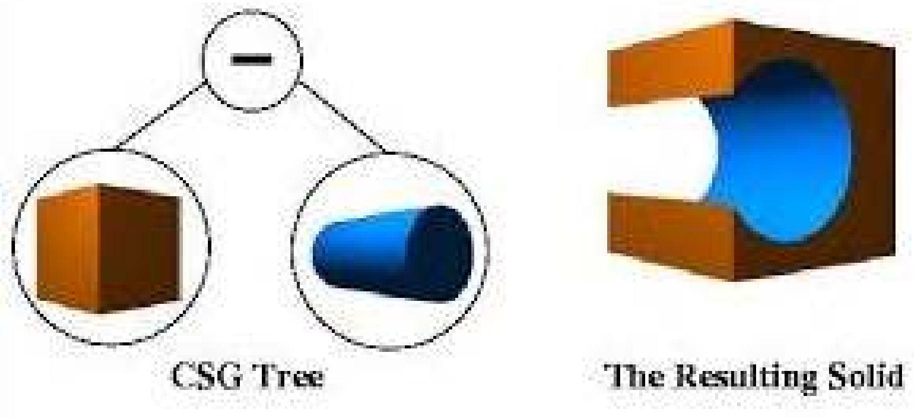


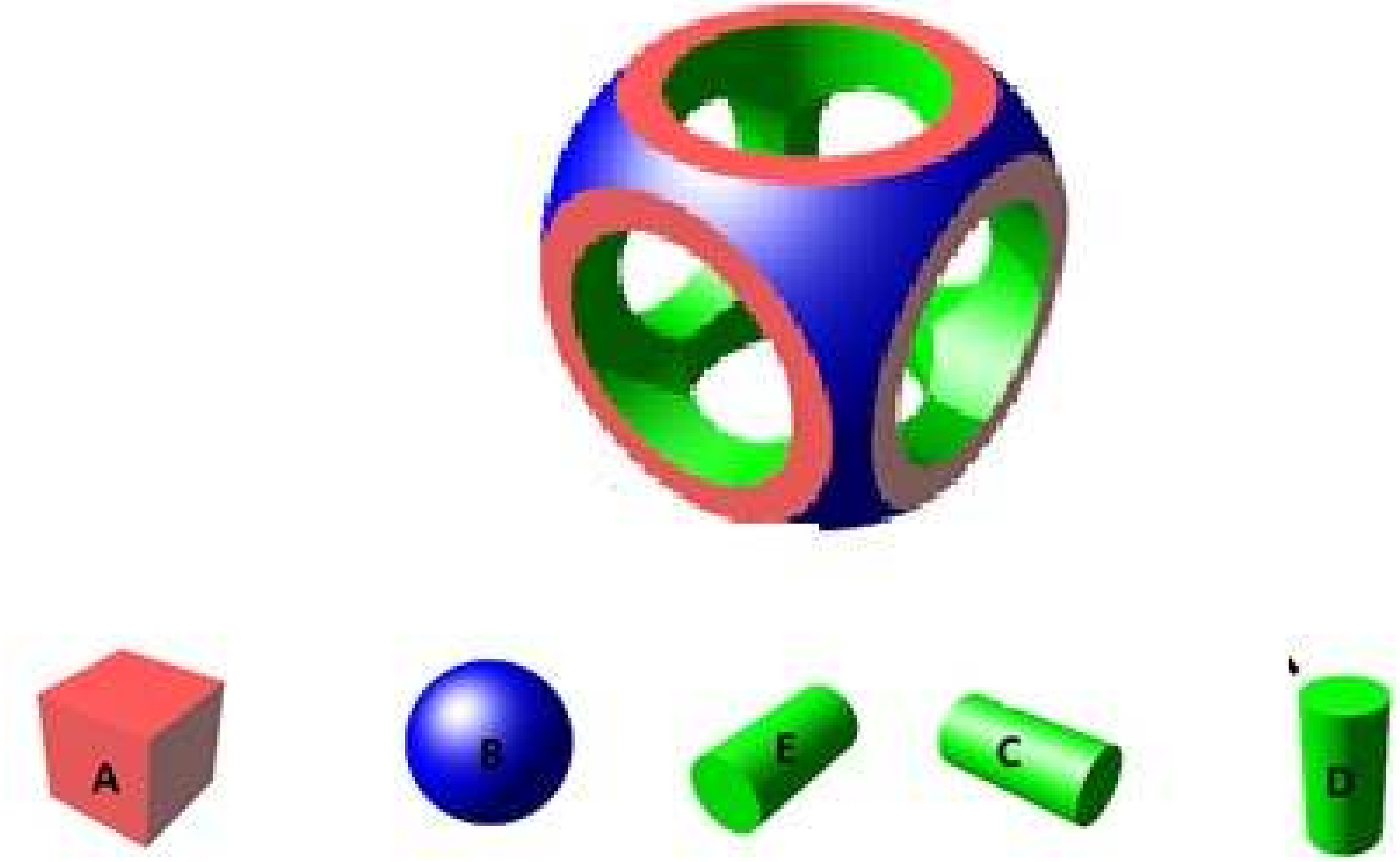


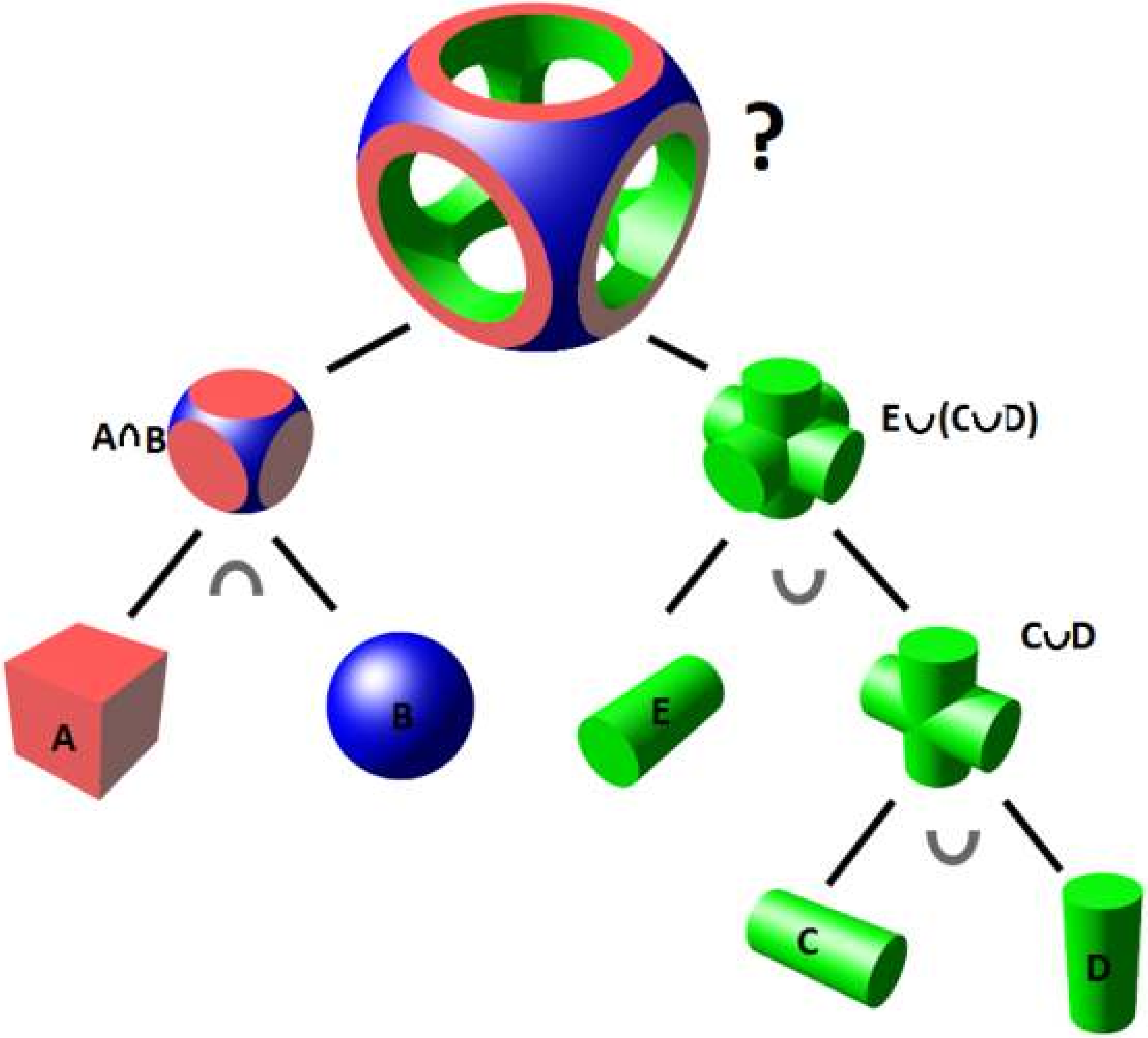
CSG Operation

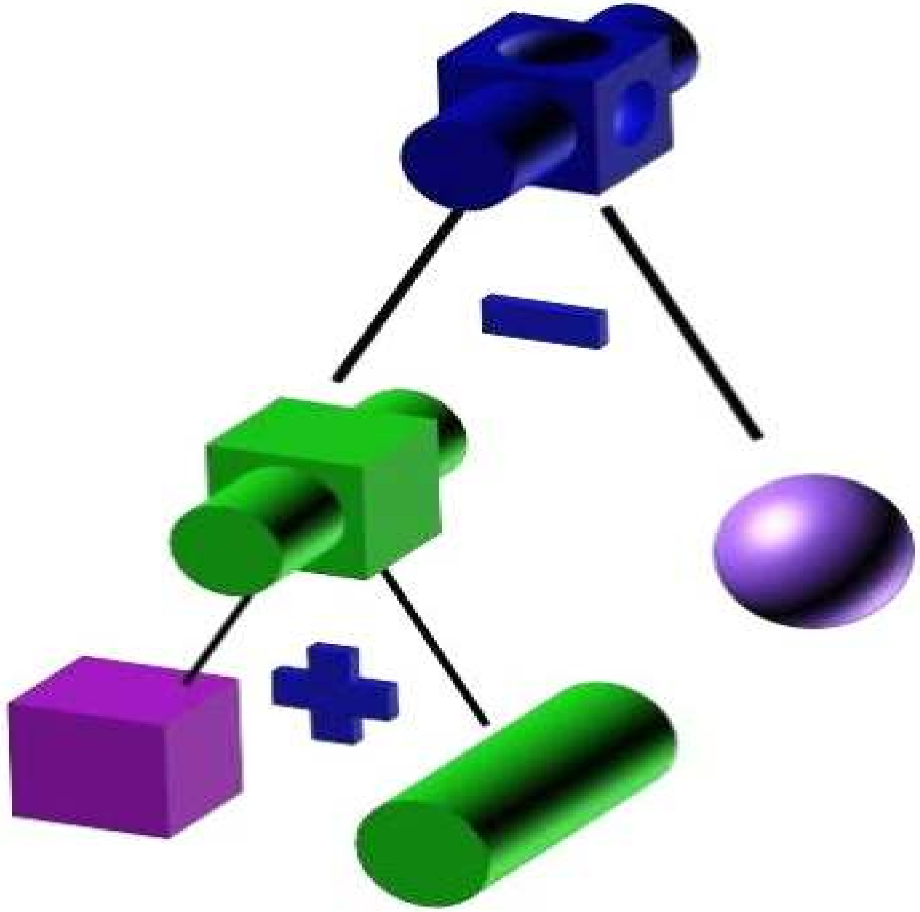
Boolean Operations in CSG





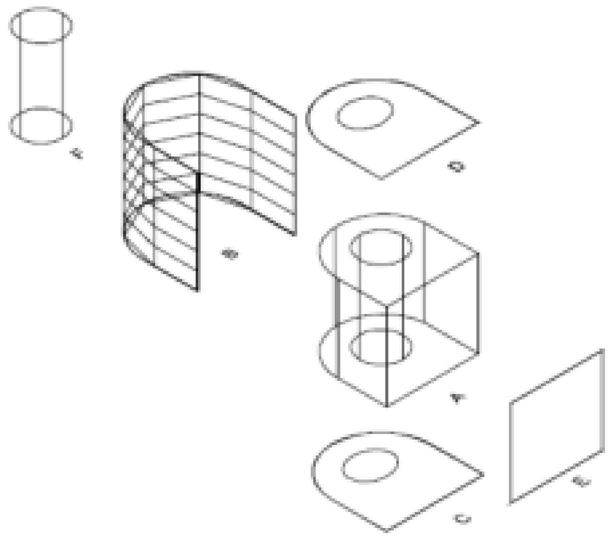

A
B
E
C
D


?
A∩B
E∪(C∪D)
∩
∪
C∪D
A
B
E
∪
C
D



B-Rep Model

- The boundary representation method represents a solid as a collection of boundary surfaces.
- The database records both of the surface geometry and the topological relations among these surfaces.
- Boundary representation does not guarantee that a group of boundary surfaces (often polygons) form a closed solid. The data are also not in the ideal form for model calculations.
- This representation is used mainly for graphical displays.
- Many CAD systems have a hybrid data structure, using both
- CSG and B-rep at the same time (i.e. Pro/E).



B-Rep Model

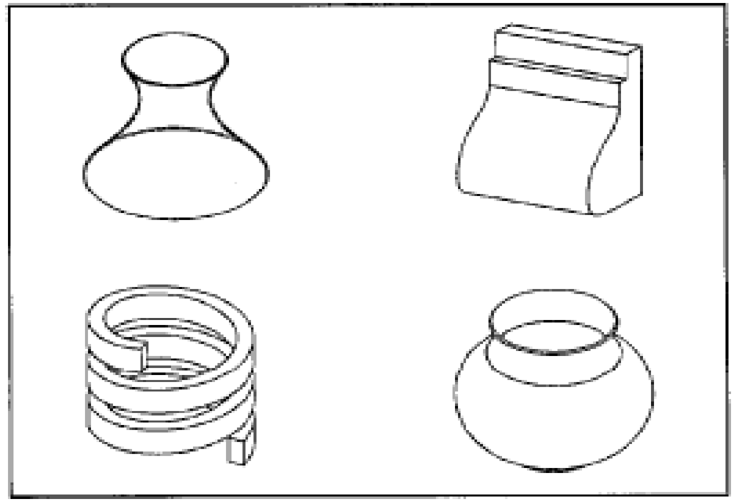
Sweeping

Sweeping can be carried out in two different forms:

- *Extrusion* - to produce an object model from a 2D cross-section shape, the direction of extrusion, and a given depth. Advanced applications include curved extrusion guideline and varying cross-sections.
- *Revolving* - to produce a rotation part, either in solid or in shell shape. Revolving a 2D cross-section that is specified by a closed curve around the axis of symmetry forms the model of an axially symmetric object.

Sweeping is most convenient for solids with translational or rotational symmetry. Sweeping also has the capability to guarantee a closed object.

Advanced: spatial sweeping; & varying cross-section



C:\Users\user\Documents\CREO\eldho\prt0001.prt.1

C:\Users\user\Documents\CREO\prt0001.prt.1

Some Solid Modelers in Practice

Modeler	Developer	Primary Scheme	User Input
CATIA	**IBM**	CSG	BREP+CSG
GEOMOD / I-DEAS	**SDRC/EDS**	BREP	BREP+CSG
PATRAN-G	**PDA ENGG.**	ASM	**HYPERPATCHES+CSG**
PADL-2	**CORNELL UNI.**	CSG	CSG
SOLIDESIGN	**COMPUTER VISION**	BREP	BREP+CSG
UNISOLIDS / UNIGRAPHICS	**McDONELL DOUGLAS**	CSG	BREP+CSG
PRO-E	**PARAMETRIC**	BREP	BREP+CSG
SOL. MOD. SYS	**INTERGRAPH**	BREP	BREP+CSG

Computer Technology

- The central and essential ingredient of CAD/CAM is the digital computer.
- The modern digital computer is an electronic machine that can perform mathematical and logical calculations and data processing functions in accordance with a predetermined program of instructions.
- There are three basic hardware components of a general-purpose digital computer:-
  a. Central processing unit (CPU)
  b. Memory
  c. Input/output (I/O) Section

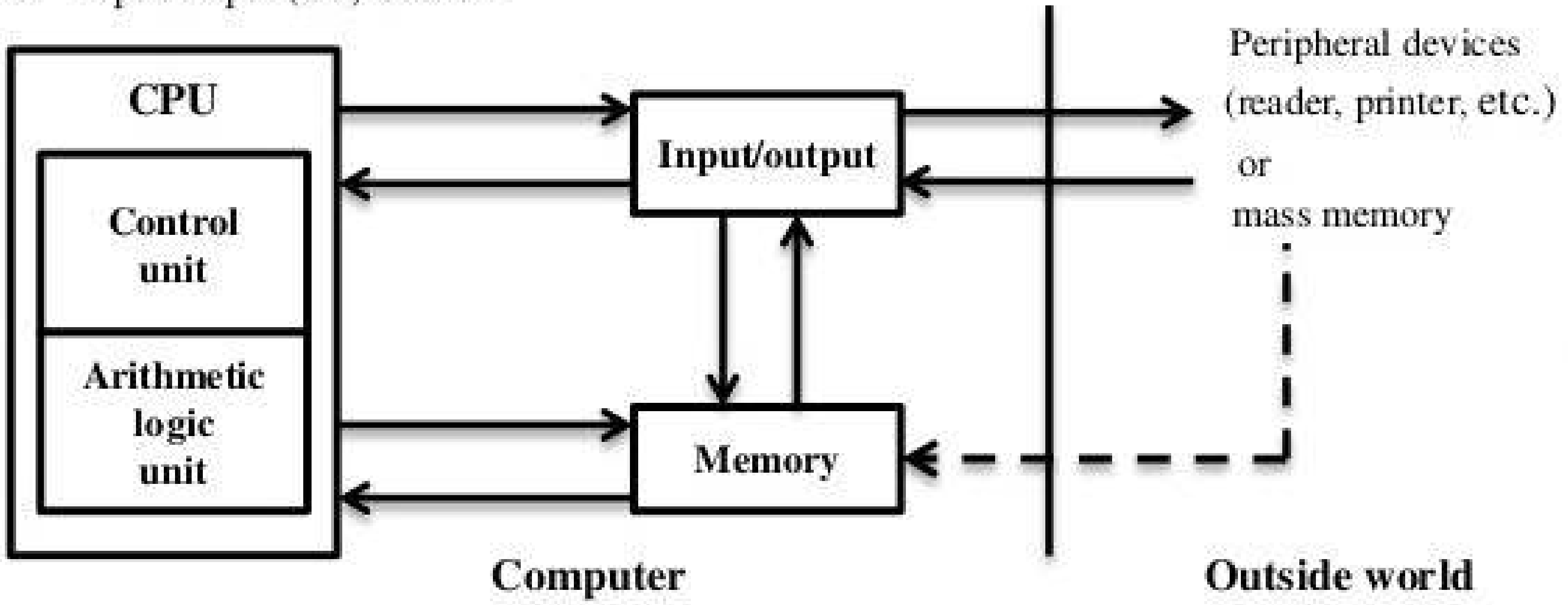


Fig. 7: Basic hardware structure of a digital computer

COMPONENTS OF CAD/CAM/CAE SYSTEMS

- **Hardware and Software**
- Hardware

  Graphic devices and their peripherals for input and output operations
- Software

  Packages that manipulate or analyze shapes according to user interaction

Hardware Components

Graphic device is composed of a display processing unit, a display device, and one or more input devices

Input devices:
- Mouse
- Space ball
- Data tablet with a puck or stylus
- Keyboard

Output Devices:
- Plotters
- Color laser printers

Packages for CAD/CAM/CAE/CAPP

ASSIGNMENT

Hardware in CAD components and user interaction devices

ASSIGNMENT

CAD Software

- A CAD software is an interactive program typically written in standard programming language like FORTRAN, Pascal, C…
- The most important characteristic of CAD/CAM software is its fully three-dimensional, associative, centralized and integrated database.
- Such a database is always rich in information needed for both the design and manufacturing processes.
- The centralized concept implies that any change in or addition to a geometric model in one of its views is automatically reflected in the existing views or any views that may be defined later.
- Integrated concept implies that a geometric model of an object can be utilized in all various phases of a product cycles
- Associativity concept implies that input information can be retrieved in various forms. For example, if the two endpoints of a line are input, the line length and its dimension can be output.

- Graphics Software Standards Graphics software performs various activities to display the graphics images for the following CAD applications:
  - Display of drawings
  - Solid model and its components
  - Wireframe geometry of the model
  - Animation of assembly
  - Art and paint applications
  - Generation of documents, reports, etc.
- The computer programs for CAD applications are either developed in any one of the high-level programming languages, or based on the application software.

- There are, in general, two types o graphics software:

1. <u>General programming software</u>

   A high-level programming language such as C and C++ is used for the development of generate programming software. The set of graphics functions, available in the Graphics Library (GL are used in software for generating the output primitives such as line, ellipse, spline, polygo etc.

2. Application based software
   - The application based software are used by designers and engineers.
   - Eg. AutoCad,SolidEdge,ANSYS,CREO,CATIA…

DESIGN DATABASE

- A database is defined as an organized collection of graphics and non-graphics data stored on secondary storage in the computer.
- Its an art of storing or the implementation of data structure into the computer.
- 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.

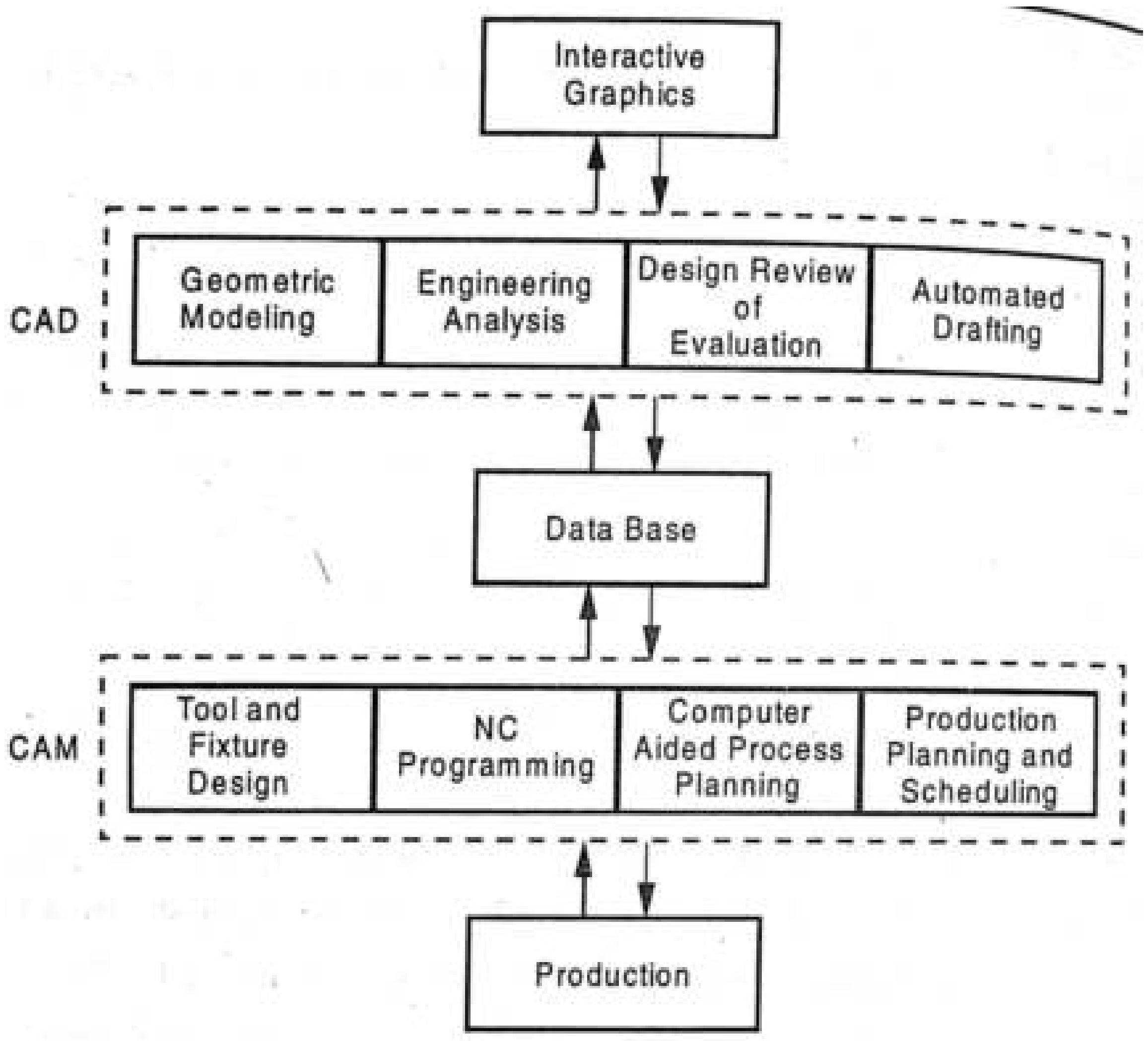


Fig. 1.8 Relationship of CAD/CAM Database to Production

The advantages that accrue from having centralized control of the data, or a centralized database, is manifold.

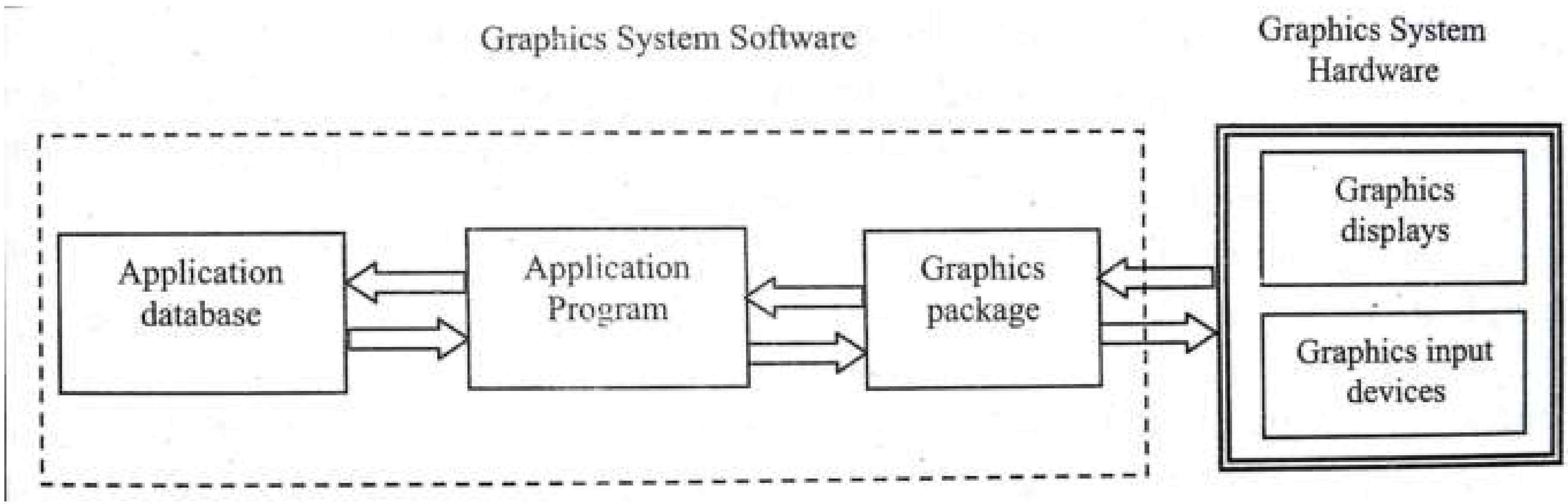
- Eliminate Redundancy
  - The database should be rich enough to support all various phases of product design and manufacturing. If both design and manufacturing departments, for example, have access to the same database, inconsistent and conflicting decisions are inherently eliminated and data is shared by all applications.
- Enforce Standards
  - With central control of the database, both national and international standards are followed. Dimensioning and tolerancing are examples. In addition, a company can develop its own internal standards required by various departments. Standards are desirable for data interchange or migration between systems.
- Apply Security Restrictions
  - Access to sensitive data and projects can be checked and controlled by assigning each user the proper access code (read, write, delete, copy and or none) to various parts of the database.

- Maintain Integrity
  - The integrity of the database ensures its accuracy. Integrity precedes consistency. Lack of database integrity can result in inputting inconsistent data.
  - Requirements Compromises can easily be made when designing a model of the centralized database to provide its overall best performance. If, for example, a software is designed solely for design and modeling, one would expect inadequate performance in manufacturing functions.
- Balance Conflicting Requirements
  - Compromises can easily be made when designing a model of the centralized database to provide its overall best performance. If, for example, a software is designed solely for design and modeling, one would expect inadequate performance in manufacturing functions.

GRAPHIC STANDARDS

- CAD/CAM software may be perceived as an application program supported by a graphics system.
- The graphics system performs all related graphics techniques.
- The graphics software is the collection of commands or programs written to make it convenient to the user to operate the graphics system.
- It perform all the graphics related activities when it receives the command from the source code of the application program.
- In the actual source code of the application program the graphics system is embedded in the form of subroutine calls.
- Therefore, software becomes inevitably device-dependent. If input output devices change or become obsolete, its related software becomes obsolete as well unless significant resources are dedicated to modify such software.
- Graphic software needs standards since the software **becomes inevitably device independent;**

Source : Sunil kumar



- Graphics package: the interaction between the user and the graphics system is achieved through the graphics package. It provides an interface between the user and the application programe.
- Application programe: Construct the design model of a particular design problem on the display device.
- Application database: the content stored here, can be displayed either on the screen or obtained in hard copy from different graphics o/p device

Standards for computer graphics

- Graphic software needs standards since the software becomes inevitably device dependent;
- Major, benefits of introducing standards for basic computer graphics.
- **Application program portability**.
  - This avoids hardware dependence of the program. A program can use in both raster display and DVST display
- **Picture data portability**.
  - Description and storage of pictures should be independent of different graphics devices.
- **Text portability.**
- **Object data base portability.**
  - Can transfer design and manufacturing database from one system to another.

GRAPHIC STANDARDS

- The focus of standards is that the application program should be device-independent and should interface to any input device through a device handler and to any graphics display through a device driver.
- By using standards CAD system is portable from one graphics system to another.
- **If a device becomes obsolete or a new one is to be supported, only the device handler driver is to be written or modified.**

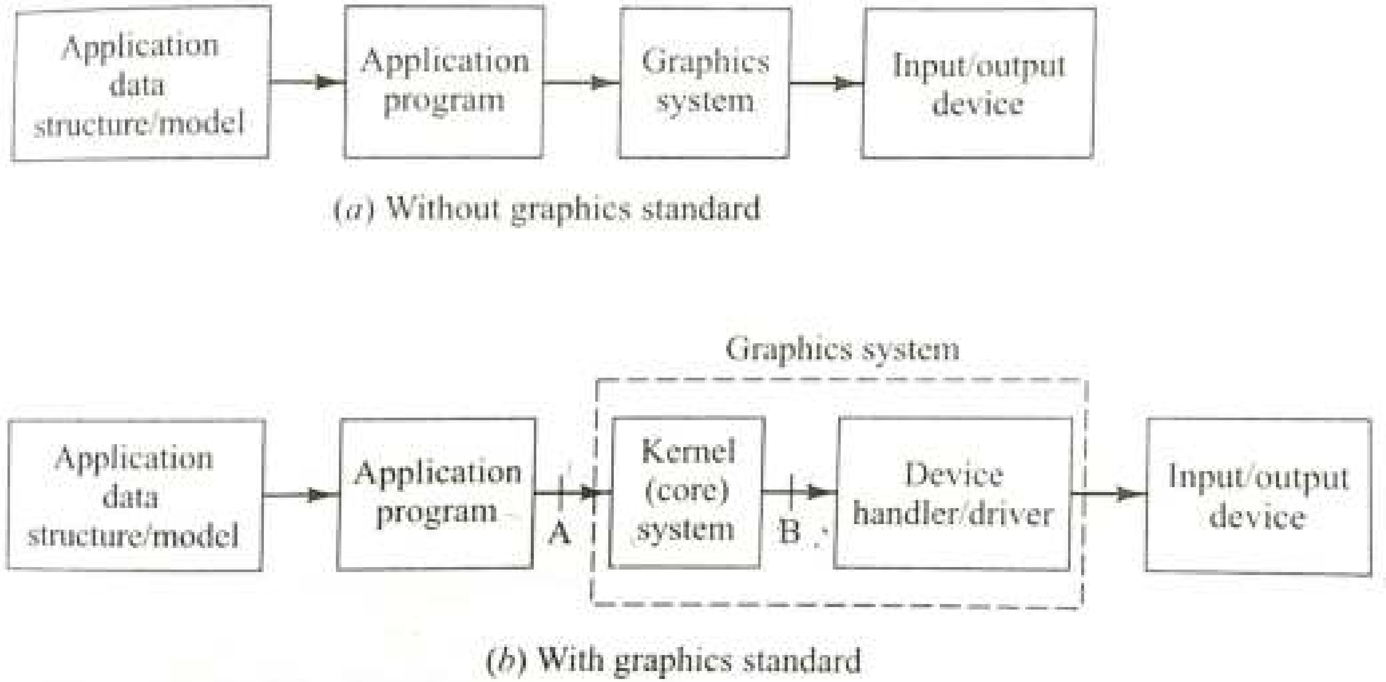


Fig. 3.2 *Organization of a typical CAD/CAM Software*

- In a CAD software with graphics standard
- The graphics system is divided into two parts. **The kernel (core) system(GKS)(or graphics standard**), which is hardware independent and the device handler/driver, which is naturally hardware-dependent.
- The kernel system, therefore acts as a buffer between application program and the specific hardware to ensure the independence and portability of the program.

Source : Ibrahim zaid

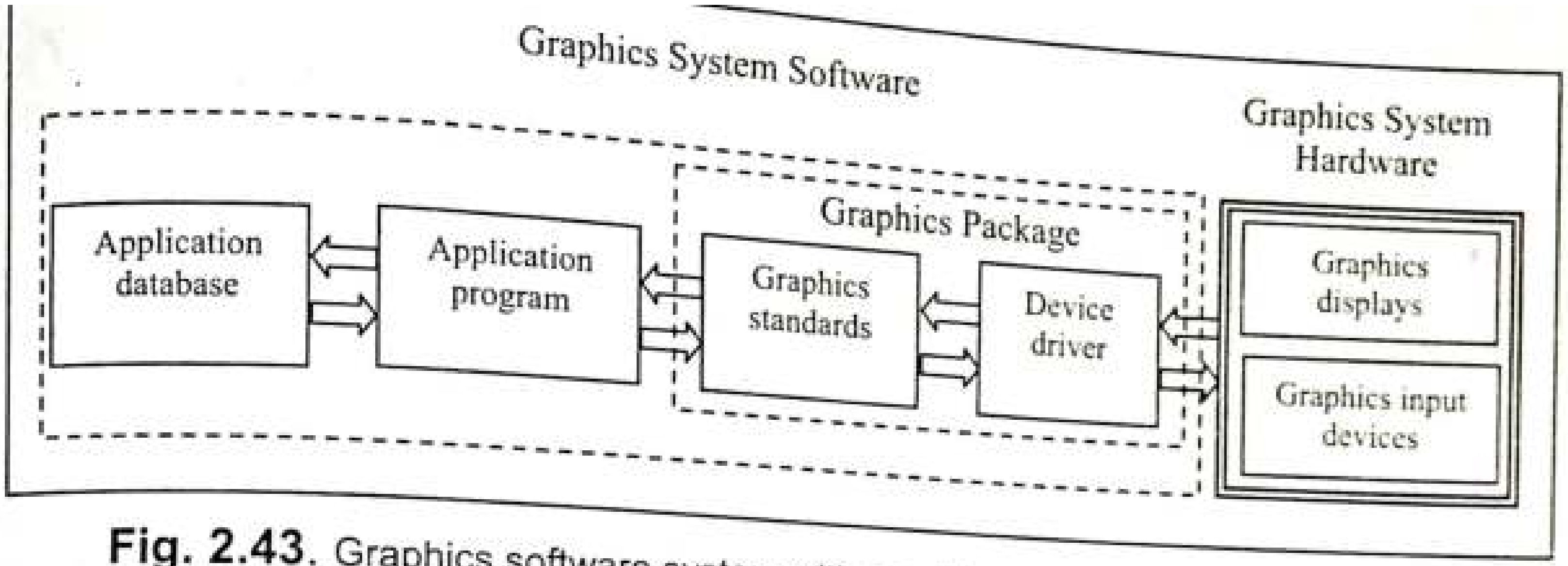


Fig. 2.43. Graphics software system with graphics standards and device driver

- In this software system application software interacts with graphics software, which in turn interacts with device driver, and the interaction end with input/output devices.
- If a device becomes obsolete or a new one is to be supported, only the device handler driver is to be written or modified.

GRAPHIC STANDARDS

The following graphics standards are used in various levels of CAD System.

1. GKS (Graphics kernel system) is an ANSI and ISO standard. It is device-independent, host-system independent and application-independent. It supports both two dimensional and three-dimensional data and viewing. It interfaces the application program with the graphics support package at interface A
2. PHIGS (Programmer's Hierarchical Interactive Graphics System) is intended to support high function workstations and their related CAD/CAM applications. The significant extensions it offers beyond GKS-3D are in supporting segmentation used to display graphics and the dynamic ability to modify segment contents and relationships. PHIGS operates at the same level as GKS (interface A).
3. VDM (Virtual Device Metafile) defines the functions needed to describe a picture. Such description can be stored or transmitted from one graphics device to another. It functions at the level just above device drivers. VDM is now called CGM (Computer Graphics Metafile).

Source : Ibrahim zaid

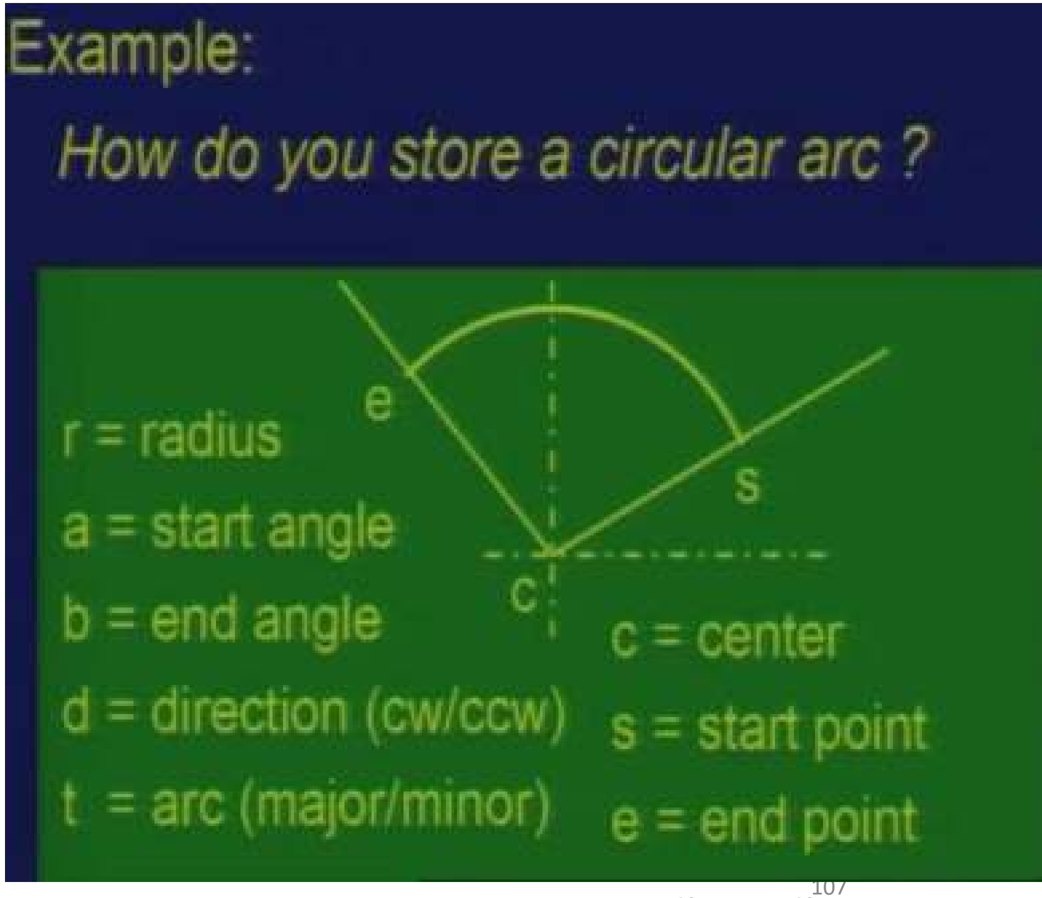
4. VDI (Virtual Device Interface) lies between GKS or PHIGS and the device handler driver code (interface B). Thus VDI is the lowest device- independent interface in a graphics system. It shares many characteristics with CGM. VDI is designed to interface plotters to GKS or PHIGS. It is not suitable to interface intelligent workstations. It is also not well matched to a distributed or network environment. VDI is now called CGI (Computer Graphics Interface).
5. IGES (Initial Graphics Exchange-Specification) It enables an exchange of model data bases among CAD/CAM systems. IGES functions at the level of the object database or application data structure. This standard contain geometric entities such as curves, surfaces, solid primitives, Boolean operations, wireframe, surface and solid modeling softwares can be developed by IGES
6. NAPLPS (North. American Presentation-Level Protocol Syntax) It describes text and graphics in the form of sequences of bytes in ASCII code.
7. DMIS (Dimensional measurement interface specifications)
8. GKSM (GKS Metafile)

Various CADCAM users and application or system programmers may be interested in one or more of the above standards. Awareness of these standards can be used as a guideline in evaluating various CAD/CAM systems. For example, mechanical design requires three-dimensional modeling. Therefore a system that supports GKS-3D or PHIGS is required. However, for two-dimensional applications such as VLSI design, GKS-2D is adequate.

DATA EXCHANGE FORMATS

- Geometric data exchange refers to how the geometry is basically exchanged between two CAD systems or how geometry has to be transferred from one system to another system.

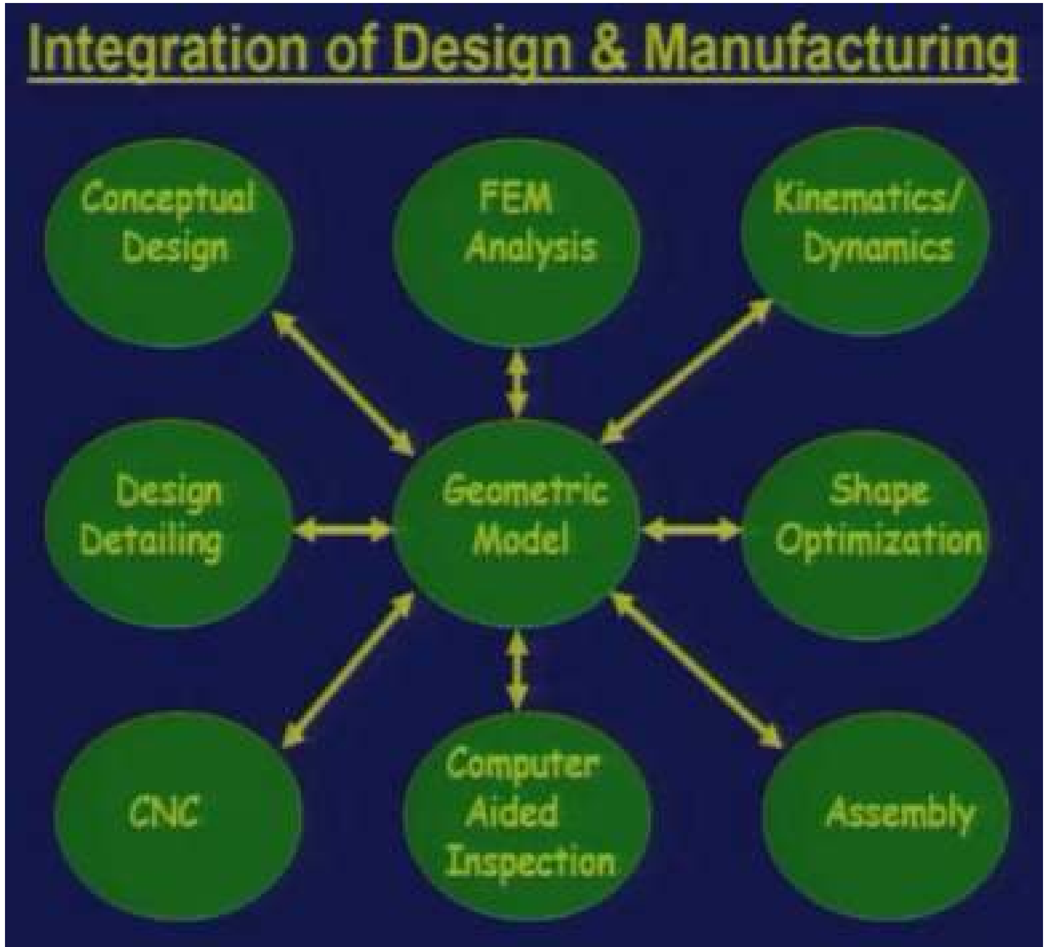
- Everybody will not selects the same set of parameters to represent an arc.
- Somebody may be more comfortable with selecting start angle, end angle. Others may prefer to choose let's say start point or end point. Somebody may select center, others may select radius.
- Different packages are storing data in different formats. Also it store data in a specific order.



Source : NPTEL By Dr. Madhusudhan Rao

Need of geometric data exchange

- Heterogeneous expertise
- Use of application specific packages
- Migration from one system to another
- Data exchange with collaborators/customers and suppliers



Different types of formats

- Native formats
  - Not a standard format, purely native and this file can be read only by that package, it cannot be read in other software except when we have a translation facility exchange.
- Neutral formats
  - Neutral format is the packages give you an option like to put the geometric information in the form of a file and which can be very easily interpreted by outside world. They also give like the complete format details and how the information is stored, what is the order in this information is stored.
- Standard format
  - Any software can always has an interpreter to read the standard format and it can interrupt what the geometry is.
- Binary format
  - Its like native format, input output option becomes much convenient, it becomes much faster.
- ASCII format
  - ASCII formats can read a file, interpret what are the characters and the phrases and others in order to know what the geometry is. And often the neutral and standard formats are ASCII formats.

Contents of geometric data

- Geometry
- Topology
- Attributes (Color, material)
- Design information (tolerance information)
- Manufacturing information (surface roughness)

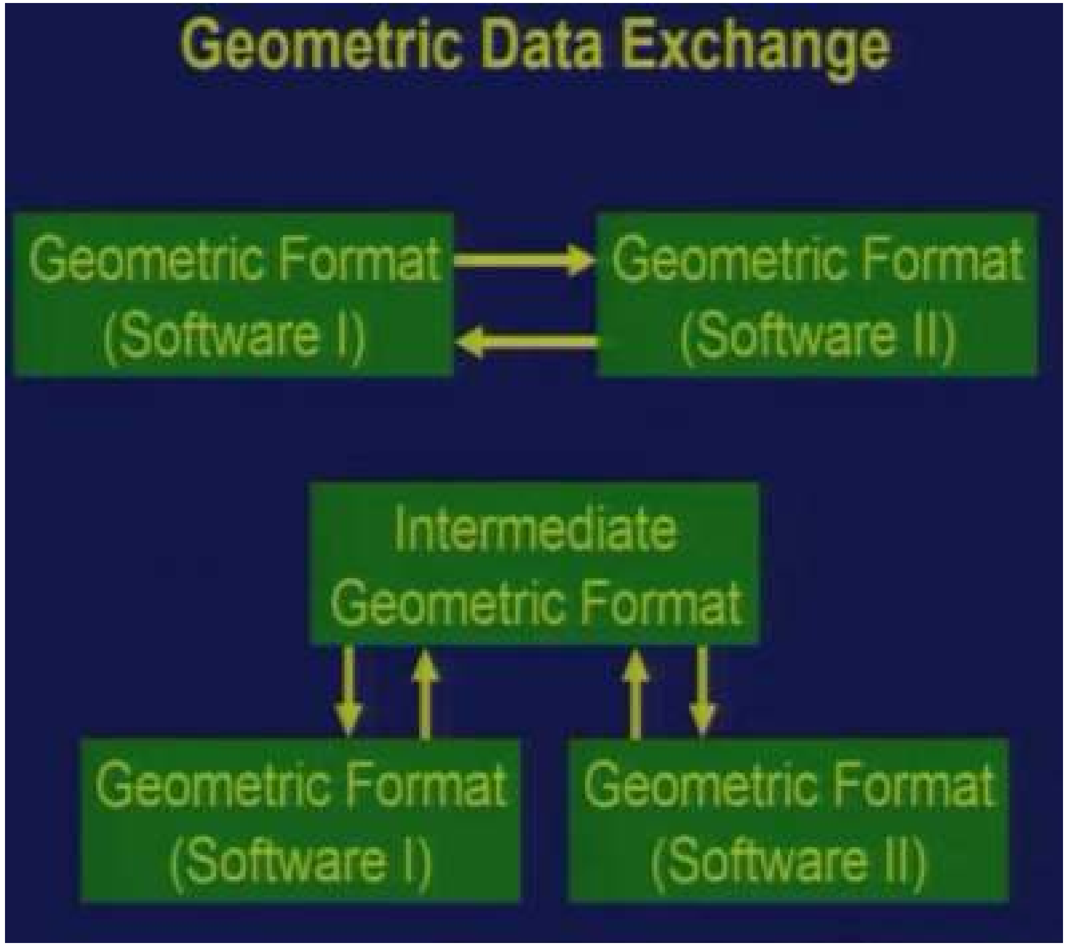
Three options for data exchange

- One native format to another (using a translator)
- One native format to another via a neutral format
  - An interpreter which reads this neutral format and writes an output as a native or another neutral format.
- One native format to another via a standard format
  - Intermediate format which is used is an international standard or a national standard which is used to do a data exchange

Source : NPTEL By Dr. Madhusudhan Rao

Requirements for the Exchange

- **Shape data:** both geometric and topological information, part or form features. Fonts, color, annotation are considered part of the geometric information.
- **Non-shape data:** graphics data such as shaded images, and model global data as measuring units of the database and the resolution of storing the database numerical values.
- **Design data**: information that designers generate from geometric models for analysis purposes. Mass property and finite element mesh data belong to this type of data.
- **Manufacturing data:** information as tooling, NC tool paths, tolerancing, process planning, tool design, and bill of materials (BOM).

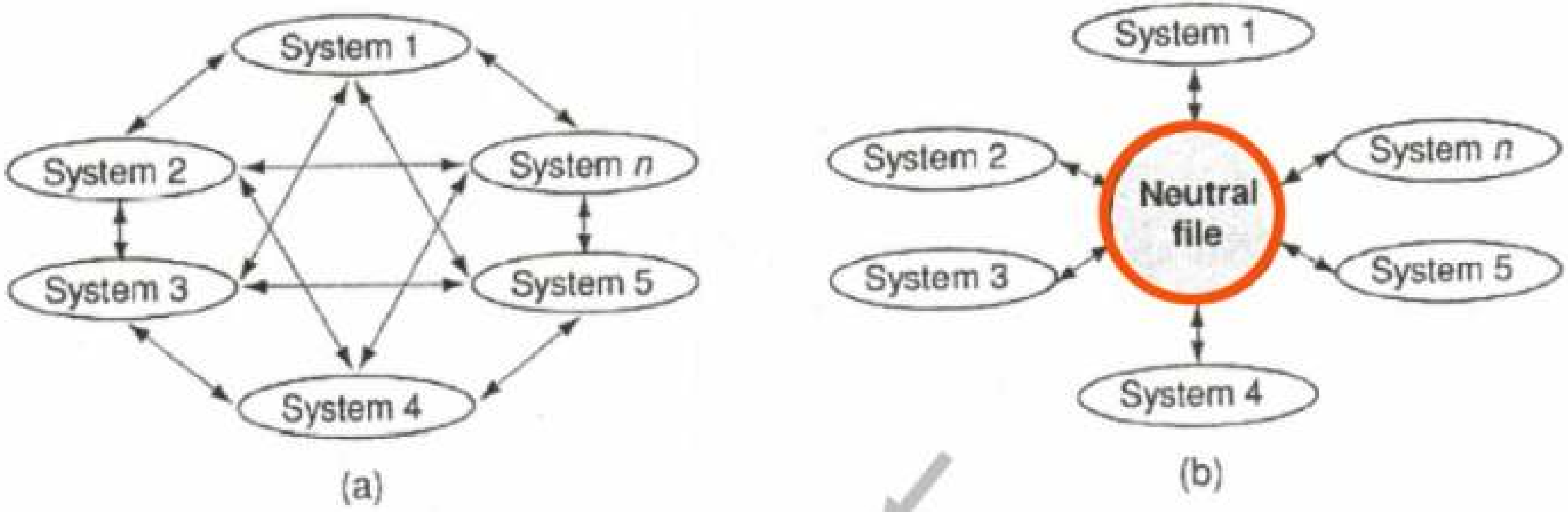


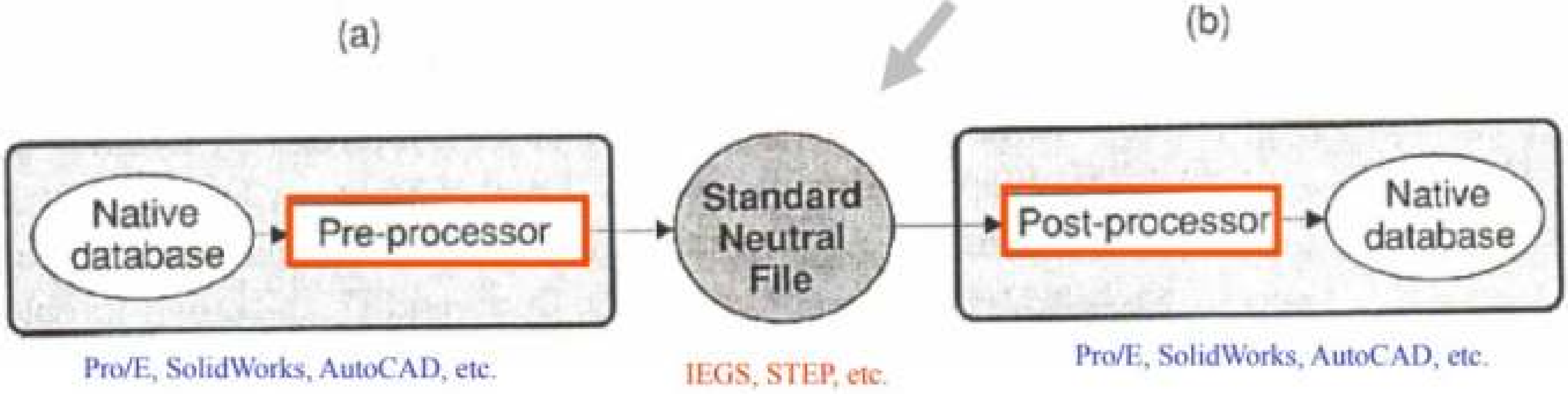
Intermediate format can be an international standard or national standard or it can be a neutral format which is given by the package.

Source : NPTEL By Dr. Madhusudhan Rao

DATA EXCHANGE FORMATS

Exchange Methods





Source : NPTEL By Dr. Madhusudhan Rao

Standard data exchange formats

Standard neutral data formats:

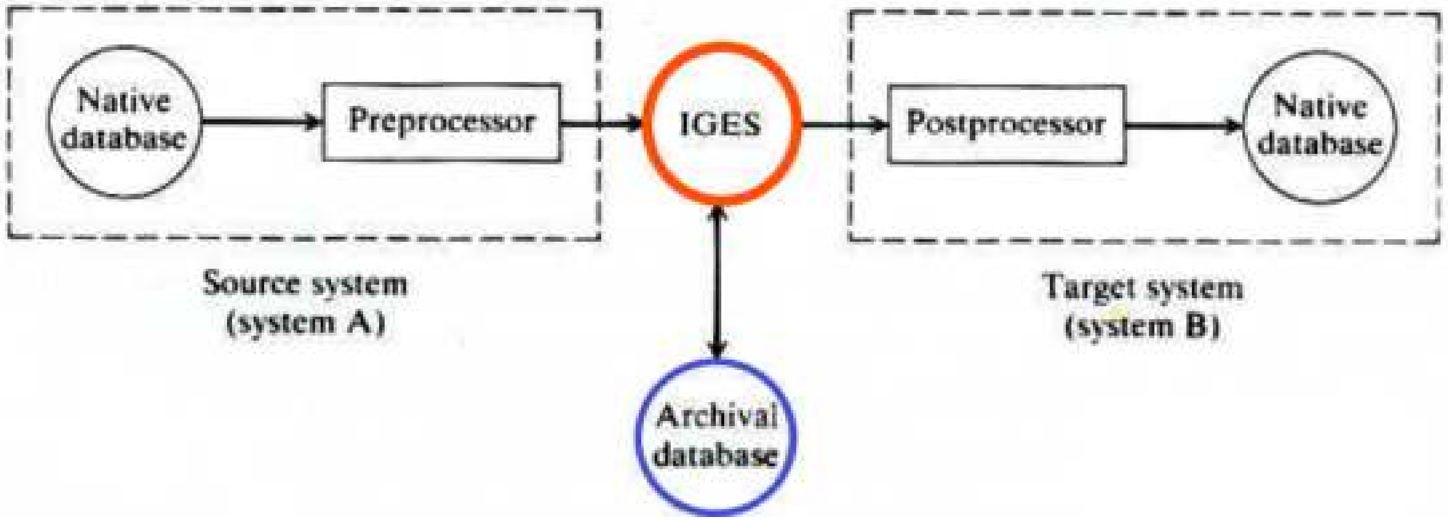
- Initial Graphics Exchange Specification (**IGES**) - the most popular format of the neutral file, supported by all CAD/CAE/CAM systems and defined by the international standard organization (ISO).
- Drawing Interchange Format (**DXF**) - a format originated by AutoDesk and used mainly for the exchange of drawing data.
- Standard for The Exchange of Product Model Data (**STEP**) - the standard data format used to store all the data relevant to the entire life cycle of a product, including design, analysis, manufacturing, quality assurance, testing, and maintenance, in addition to the simple product definition data. The data format was also called **PDES** (Product Design Exchange Specification)

Source : NPTEL By Dr. Madhusudhan Rao

DATA EXCHANGE FORMATS

IGES (Initial Graphics Exchange Specification)

- First developed by National Institute of Standards and Technology (NIST) in 1980.
- Then adopted by the American National Standards Institute (ANSI) in the same year.
- Exchanges primarily shape (both geometric and topological) and non-shape data, which is referred as CAD-to-CAD exchange
- It codes a superset of common entities of all CAD/CAM systems to facilitate the translation between various systems



IGES Format

An IGES file is composed of six sections in the following order. A record is a line comprising 80 characters.

1. Flag (optional),
2. Start,
3. Global, — background
4. Directory Entry (DE), — content
5. Parameter Data (PD), and — data
6. Terminate

Originally based on FORTRAN Format
- ASCII and Binary
- 80 Characters per Line
- Data by Field

- Start section which tells who has generated this IGES file or which software IGES file can be created in a binary as well as ASCII option

Source : Ibrahim zaid

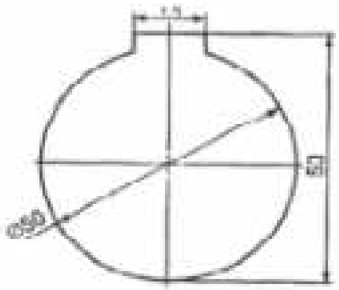
Example IGES File

Global Section

```
IGES file generated from an AutoCAD drawing by the IGES                S0000001
translator from Autodesk, Inc., translator version IGESOUT 3.04.       S0000002
```

Directory Entry

An Example IGES File for the 2D Keyed Hole.



Parameter Data

DATA EXCHANGE FORMATS

DRAWING INTERCHANGE FORMAT (DXF)

Drawing interchange format (DXF) files were originally developed to give users flexibility in managing data and translating AutoCAD' drawings into file formats that could be read and used by other CAD/CAM/CAE systems.

Because of the popularity of AutoCAD, DXF became the de facto standard of interchanging CAD drawing files for almost all CAD/CAM/CAE systems. In fact, almost every newly introduced CAD/CAM/CAE system tends to provide translators to and from the DXF file.

A DXF file is an ASCII text file and consists of five sections:

- **Header** - describes the AutoCAD drawing environment that existed when the DXF file was created.
- **Table** - contains information about line types, layers, text styles, and views that may have been defined in the drawing.
- **Block** - contains a list of graphic entities that are defined as a group.
- **Entity** - immediately follows the Block section, and serves as the main part of the DXF file, with all entities of the drawing described in it.
- **Terminate** - indicates the end of the file.

background

content

data

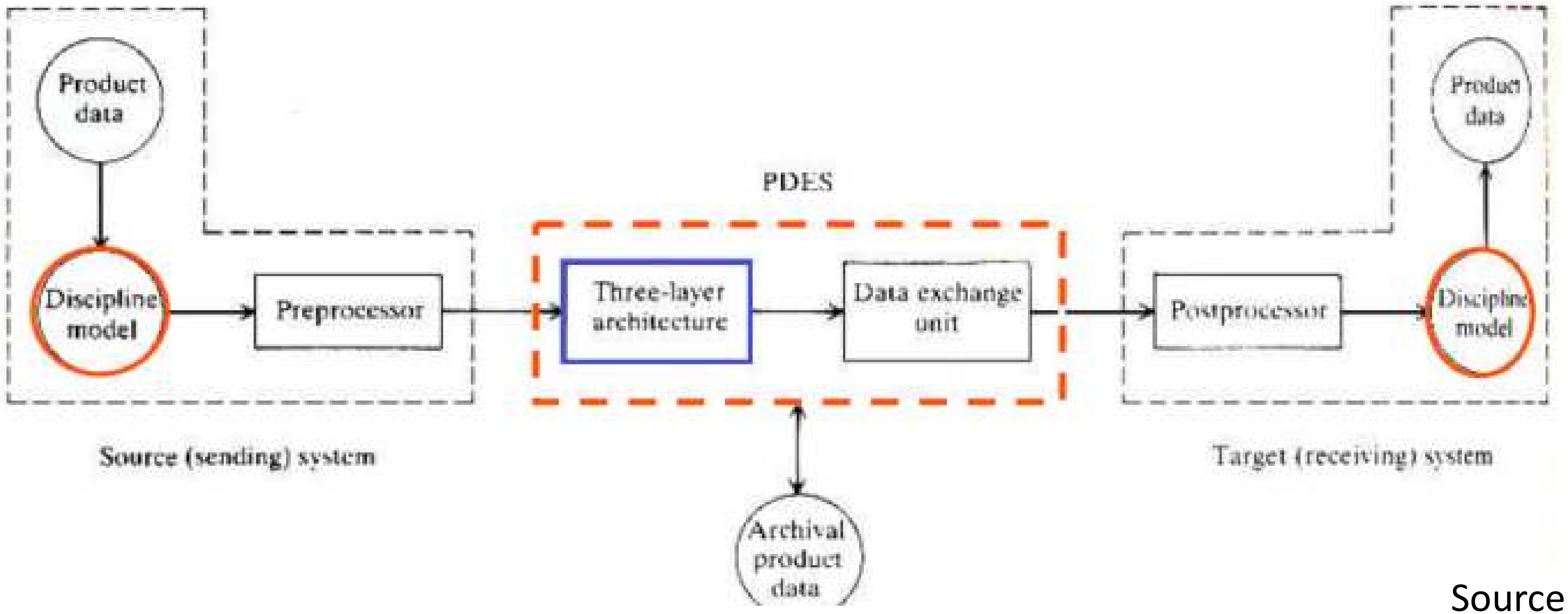
Source : Ibrahim zaid

DATA EXCHANGE FORMATS

- A simple and popular neutral format for data exchange
- AutoCad has own native format which have an extension like DWG or can save as DXF file.
- The advantage of DXF file is, while writing an application, programmer can read the DXF file and interpret and use that geometry for his applications. Though it was developed by one specific company but it's quite popular for variety of applications where it has almost become a standard in many senses because many packages give an option of DXF as a data exchange option.

PDES (Product Data Exchange Standard) or STEP (Standard for Exchange of Product Data)

- To support any industrial application such as mechanical, electric, plant design, and architecture and engineering construction
- To include all four types of data which is relevant to the entire life-cycle of a product: design, analysis, manufacturing, quality assurance, testing, support, etc.



Source : Ibrahim zaid

DATA EXCHANGE FORMATS

- STEP is again an initiative of many countries European, US, Japanese and many others and primarily again initiative of you can say US Bureau, US National Bureau of Standards and it has a ISO number 10303.
- Option to exchange assembly or tolerance data using a STEP
- STEP is not a single thing. It has many application protocols which are called as Ap's.
- In step AP 201 It can exchange drawings

  two dimensional drawings.

AP 201 - Draughting
AP 203 - Configuration-controlled design
AP 207 - Sheet metal die planning and design
AP 209 - Structural Analysis
AP 213 - NC process plans for machined parts
AP 219 - Dimension inspection
AP 224 - Mech parts def for process planning
AP 229 - Design/manu product info for forged parts.

SET (standard d'echange et de transfert)

- SET standard by French government efforts is to come up with a format which is more compact than IGES.
- IGES is a very verbose format
- A single line or a triangle and try to generate a file, IGES contain hundred lines of code which is basically because you have to represent the complete information which is necessary including all the attributes, delimiters etc.
- SET standards is more compact for representing
- For a work with French industry and if they have an option of SET then this may be an advantage.

Source : Ibrahim zaid

VMRL (Virtual reality modeling language)

- VRML is another standard. This is again more of an international effort where idea was to come up with exchange of geometry over a World Wide Web application.
- CAD model of the object and convert into VRML format and post it on the web, so anybody who has a VRML browser can open and see it.
- The advantage of this is it also gives you, the user can interact with the object.
- VRML is used by certain people is for assembly animation like I buy a product and this product which I buy consists of many components.

VDA

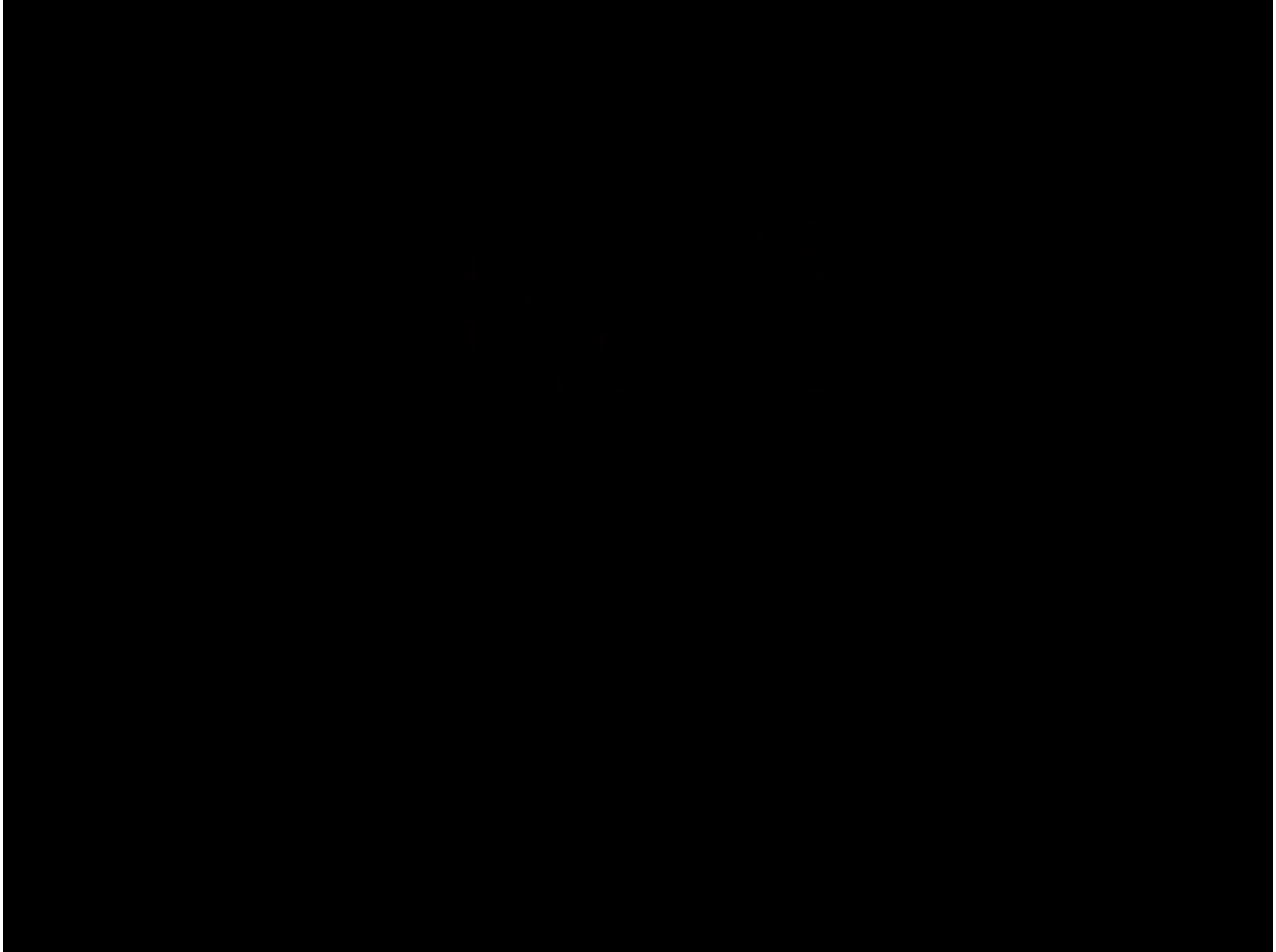
- VDA is another geometric data exchange standard.
- This is actually a national standard; it's a German standard for CAD data exchange.

VIRTUAL REALITY

- Virtual Reality refers to a high-end user interface that involves real-time simulation and interactions through multiple sensorial channels.
- Virtual reality (VR) is a computer technology that uses virtual reality headsets or multi-projected environments, sometimes in combination with physical environments, to generate realistic images, sounds and other sensations that simulate a user's physical presence in a virtual or imaginary environment.
- Why VR?

  VR is able to immerse you in a computer-generated world of your own making: a room, a city, the interior of human body. With VR, you can explore any uncharted territory of the human imagination

EMULATE3D

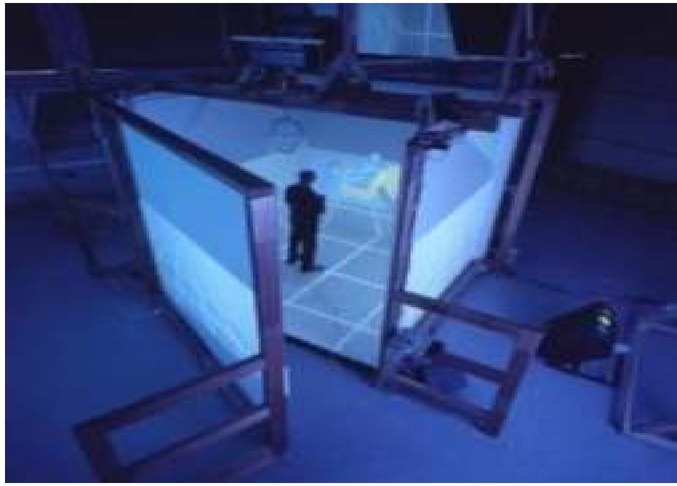


Technologies of VR--Hardware

- Head-Mounted Display (HMD)
  - A Helmet or a face mask providing the visual and auditory displays.
  - Use LCD or CRT to display stereo images.
  - May include built-in head-tracker and stereo headphones

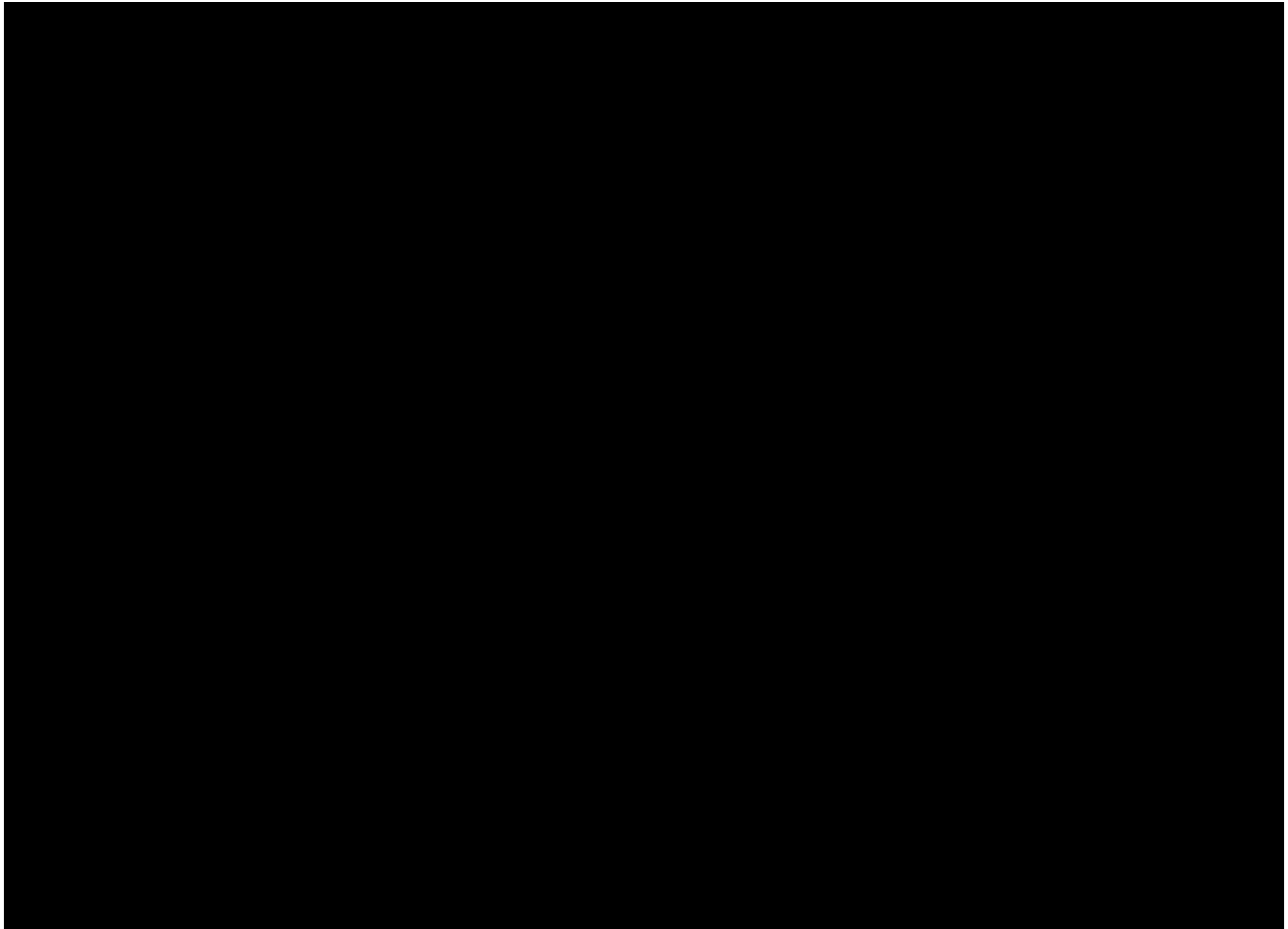
- Binocular Omni-Orientation Monitor (BOOM)
  - Head-coupled stereoscopic display device.
  - Uses CRT to provide high-resolution display.
  - Convenient to use.
  - Fast and accurate built-in tracking.

- **Cave Automatic Virtual Environment (CAVE)**
  - Provides the illusion of immersion by projecting stereo images on the walls and floor of a room-sized cube.
  - A head tracking system continuously adjust the stereo projection to the current position of the leading viewer.



- **Data Glove**
  - Outfitted with sensors on the fingers as well as an overall position/orientation tracking equipment.
  - Enables natural interaction with virtual objects by hand gesture recognition.
- **Toolkits**
  - Programming libraries.
  - Provide function libraries (C & C++).
- **Authoring systems**
  - Complete programs with graphical interfaces for creating worlds without resorting to detailed programming.

CAVE

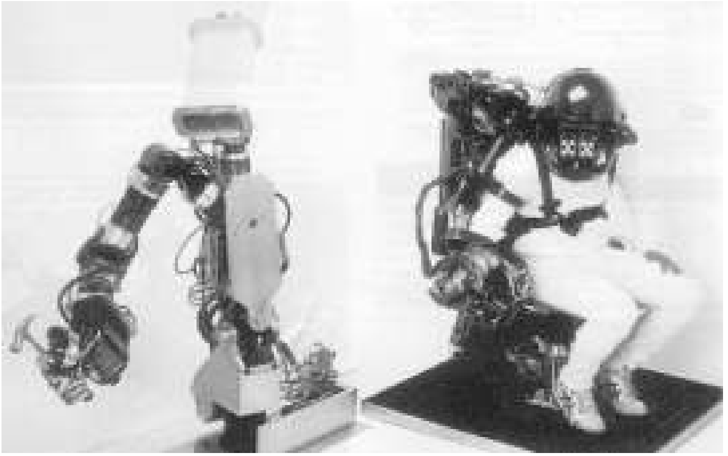


Types of VR System

- **Windows on World (WoW)-** Desktop VR.
  - Using a conventional computer monitor to display the 3D virtual world.
  - Perfect for the field of medicine. Typically using a desktop monitor , it allows its ***user to visualize complex medical procedures such a surgeries or colonoscopies.***

- **Immersive VR**
  - Completely immerse the user's personal viewpoint inside the virtual 3D world.
  - The user has no visual contact with the physical word.
  - Treadmill interface to simulate the experience of walking through virtual space.
  - Often equipped with a BOOM or Head Mounted Display (HMD).

- **Telepresence**
  - feeling of being in a location other than where you actually are.
  - A variation of visualizing complete computer generated worlds.
  - Links remote sensors in the real world with the senses of a human operator. The remote sensors might be located on a robot. Useful for performing operations in dangerous environments.



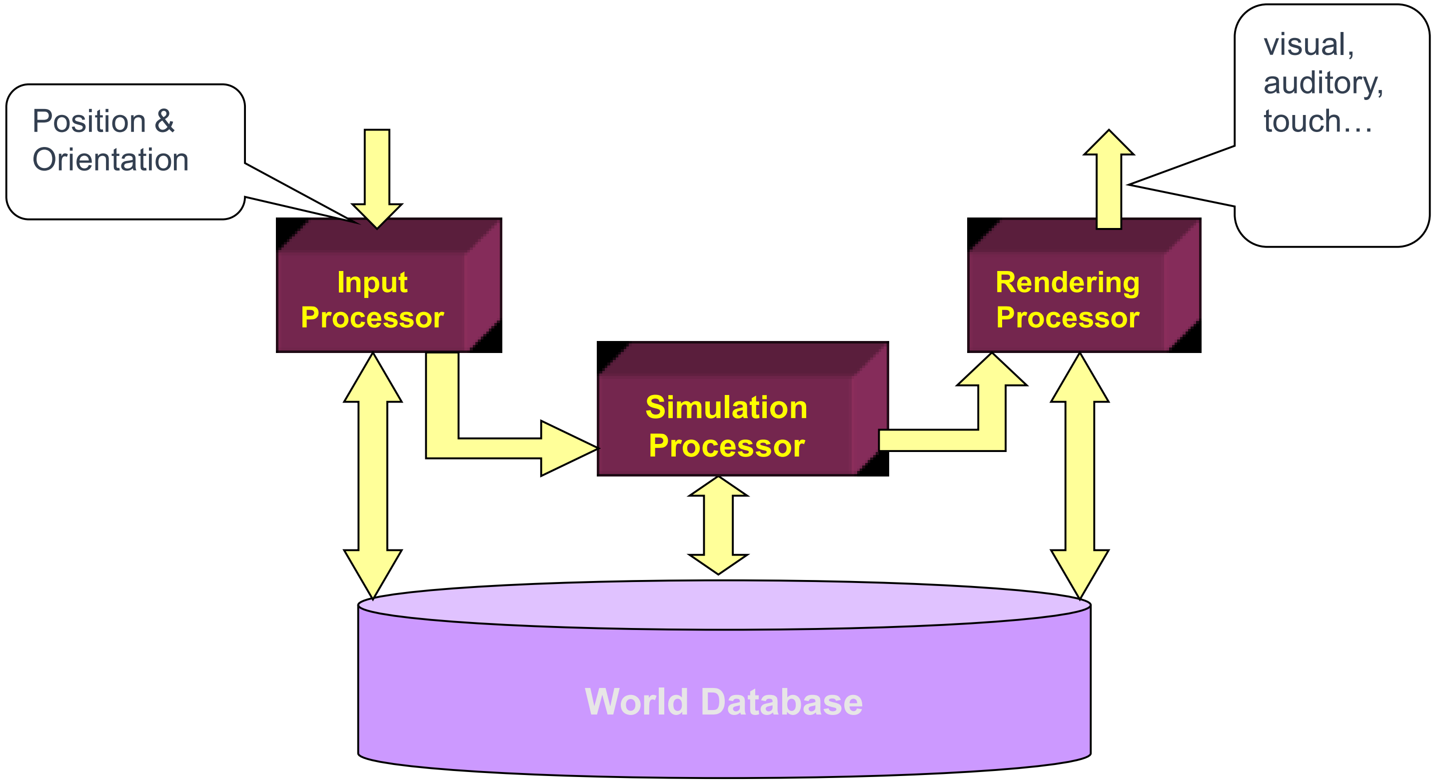
- **Distributed VR**
  - A simulated world runs on several computers which are connected over network and the people are able to interact in real time, sharing the same virtual world.

- **Mixed Reality(Augmented Reality)**
  - A variation of immersive virtual reality is Augmented Reality where a see-through layer of computer graphics is superimposed over the real world to highlight certain features and enhance understanding.

Augmented Reality

Architecture of VR System

Input Processor, Simulation Processor, Rendering Processor and World Database.



Components of VR System

- Input Processor
  - The Input Processes of a VR program control the devices used to input information to the computer.
  - input devices: keyboard, mouse, trackball, joystick, 3D & 6D position trackers (glove, wand, head tracker, body suit, etc.).
  - A networked VR system would add inputs received from net. A voice recognition system is also a good augmentation for VR.
  - The object is to get the coordinate data to the rest of the system with minimal lag time.
  - Some position sensor systems add some filtering and data smoothing processing.
  - Some glove systems add gesture recognition. This processing step examines the glove inputs and determines when a specific gesture has been made. Thus it can provide a higher level of input to the simulation.

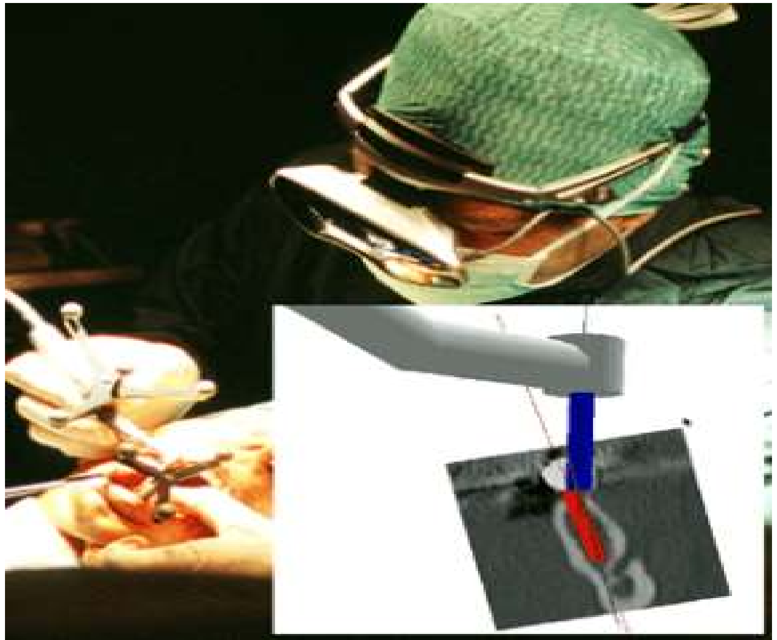
VIRTUAL REALITY

- Simulation Processor
  - Core of a VR system.
  - It handles the interactions, the scripted object actions, simulations of physical laws (real or imaginary) and determines the world status.
  - Takes the user inputs along with any tasks programmed into the world and determine the actions that will take place in the virtual world.
- Rendering Processor
  - Create the sensations that are output to the user.
  - Separate rendering processes are used for visual, auditory, and haptic other sensory systems.
- World Database (World Description Files)
  - Store the objects that inhabit the world, scripts that describe actions of those objects.

VIRTUAL REALITY

Applications of VR

- Entertainment
  - More vivid
  - Move exciting
  - More attractive



- Medicine
  - Practice performing surgery.
  - Perform surgery on a remote patient.
  - Teach new skills in a safe, controlled environment.

VIRTUAL REALITY

- Manufacturing
  - Easy to modify
  - Low cost
  - High efficient

- Education & Training
  - Driving simulators.
  - Flight simulators.
  - Ship simulators.
  - Tank simulators.

Thank you…

MODULE 2

Geometric Transformations and Projections

Syllabus

Transformation of points and line, 2-D rotation, reflection, scaling and combined transformation, homogeneous coordinates, 3-D scaling.

Shearing, rotation, reflection and translation, combined transformations, orthographic and perspective projections, reconstruction of 3-D objects.

Objective

To introduce the fundamentals of geometric transformation

Outcome

Students will understand the basic mathematical fundamentals of CAD geometric transformation

1. What is Geometric Transformations?

- **Thus, the geometric transformation is defined as a set of operations on coordinate system (alternatively. Object) that results in a change in the position (or location) of the coordinate system**

- Geometric transformations are used to modify the images on the display devices and to reposition the objects in the database by altering its coordinate descriptions.

- Typical CAD geometric transformations like **translation, rotate, reflection, scaling** are used a lot during geometric modeling. Also it is used in SET operations.

Why transformations?

- In many CAD applications, the **designs and layouts of picture scene are created** by rearranging the components of varying sizes, shapes and at different orientations.
- The **animations** are performed by moving the objects along the animation path.
- There is a need of manipulating the displayed objects during **feature design**.
- Such changes in size, shapes and orientations are accomplished with the help of Geometric transformation.
- Object transformations are also used in applications like **robotics and virtual reality.**

Transformation of geometric models

- During geometric transformation model perform relative motion with respect to the reference point in the device coordinate system.

- Rigid body motion
  - In which the relative distances between the object particles (or coordinates) remain constant.
  - The object does not deform during the motion.
  - Geometric transformations that describe the rigid body transformations include translation, rotation, scaling and reflection
- Shear body motion
  - The distortion causes sliding of the internal layers over the other.

- A point is a **basic entity** for the object representation.
- A line is represented by its endpoints. Similarly, curves, surfaces and solids are the collection of several points.
- The geometric transformation of a given **point P(x, y, z)** of a geometric model to the corresponding new point **$P_T(x_T, y_T, z_T)$** is given as

$$P_T = f(\,P,\ \text{transformation parameter})$$

- Transformed coordinates **P_T** is a function of original coordinates P and the motion parameters corresponding to the given geometric transformation.
- However. if an object consists of several points, all points of the object must transform corresponding to the given geometric transformation.

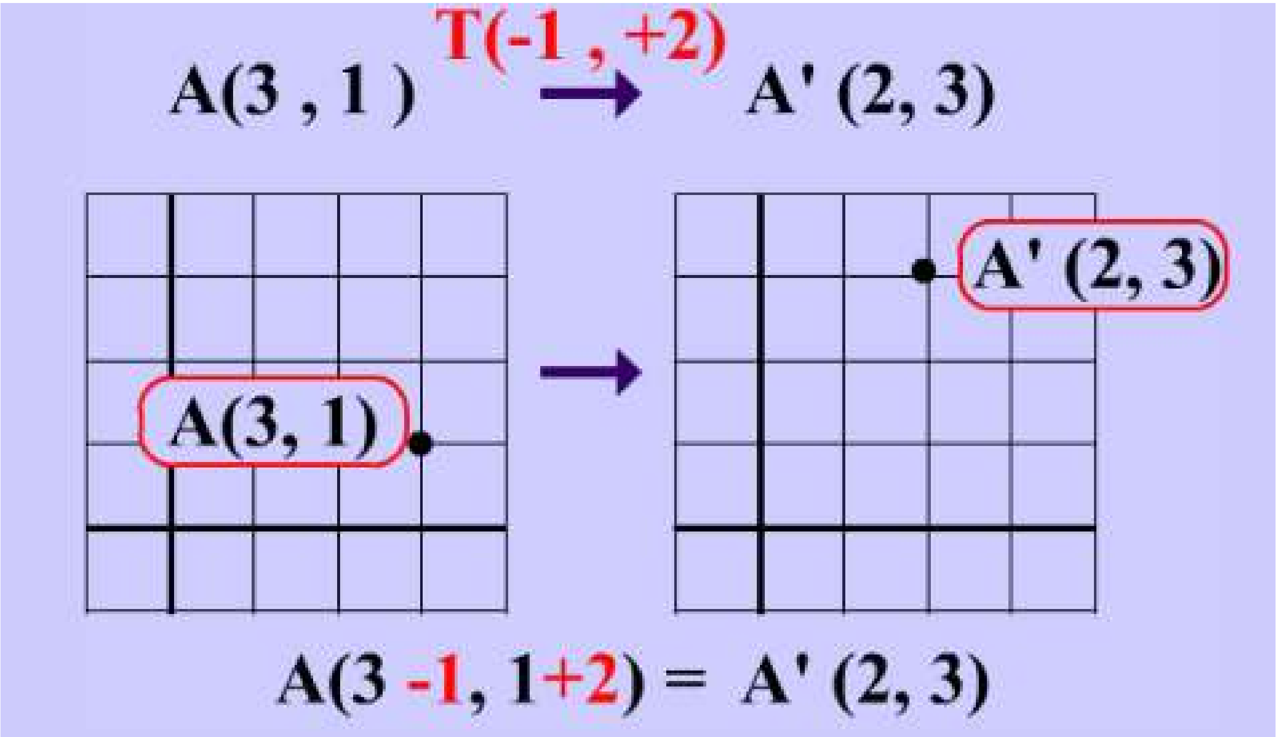
2-D TRANSLATION

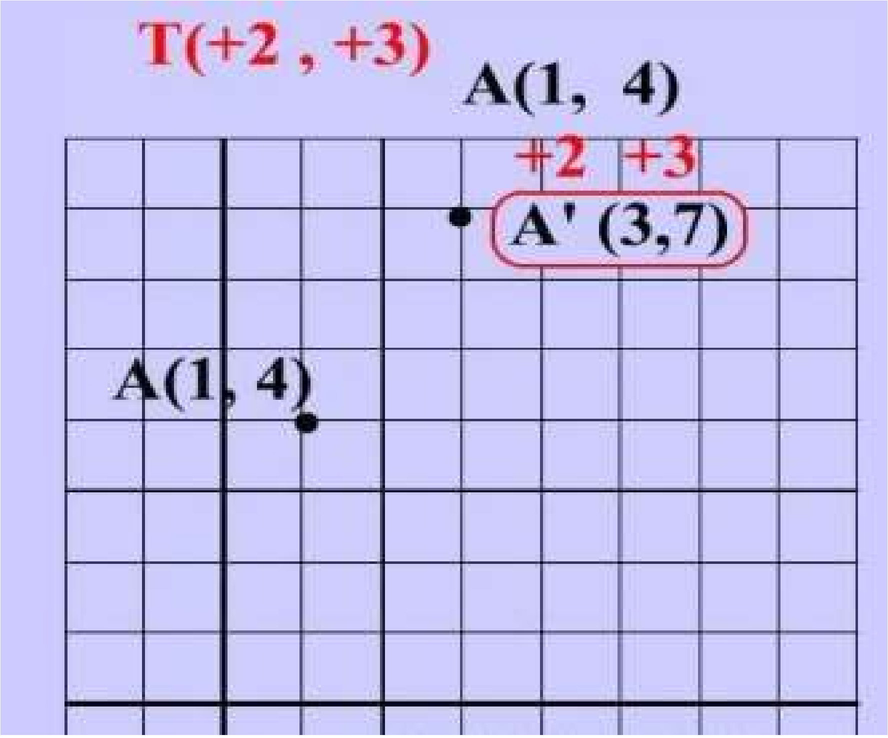
- Whenever an entity of geometric model remains parallel to its initial positions, the rigid body transformation of the model is termed as translation.
- It is rigid body transformation
- Translation allows the **repositioning of an object** from one place to another along a straight line; therefore, every point in the model moves an equal distance in a given direction.
- Translation of an object is obtained by adding the translational distances t_x and t_y to the original coordinate positions (x, y) of the model.
- Thus, a point **P(x, y)** is transformed to **$P_T(x_T, y_T)$** by adding translational distances t_x and t_y as

$$x_T = x + t_x$$

$$y_T = y + t_y$$

2-D TRANSLATION





2-D TRANSLATION

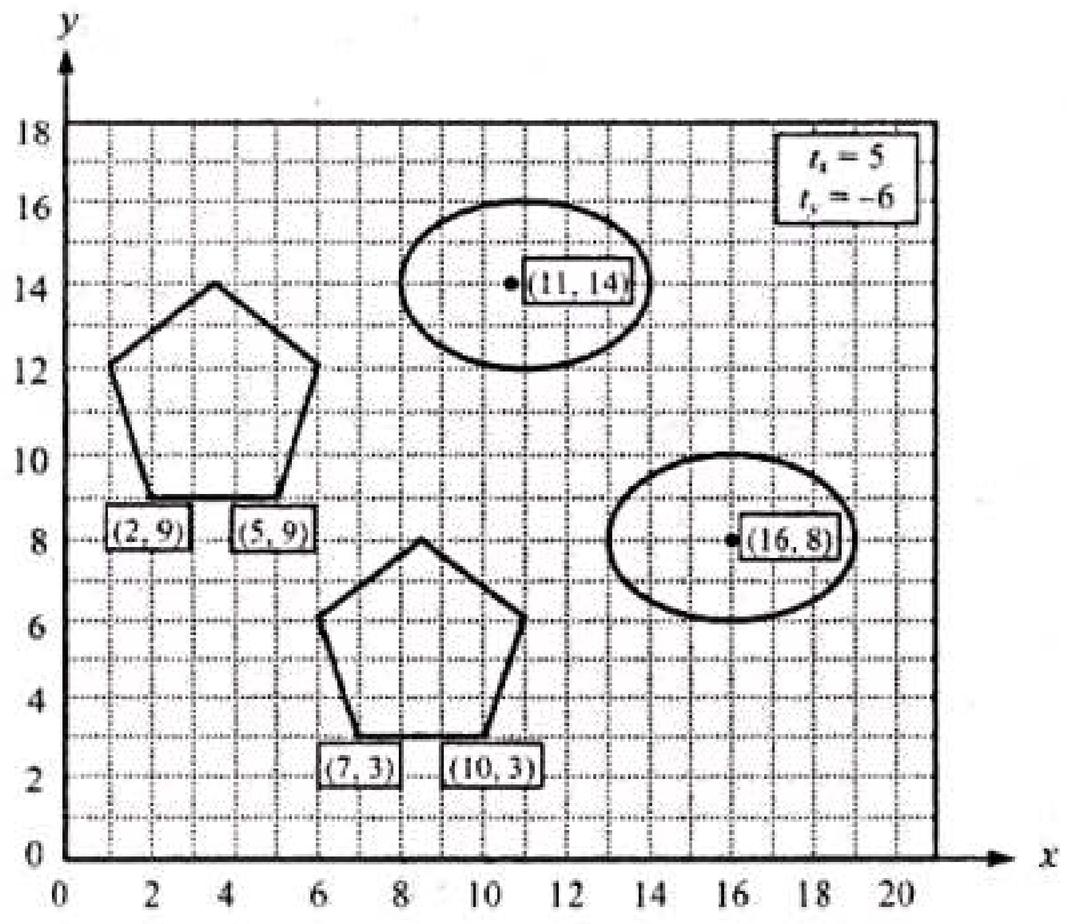
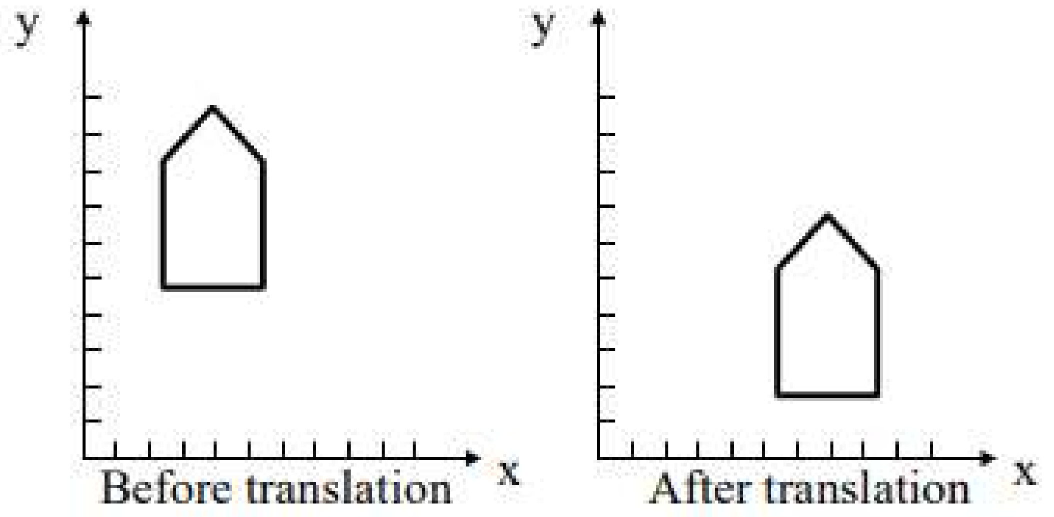
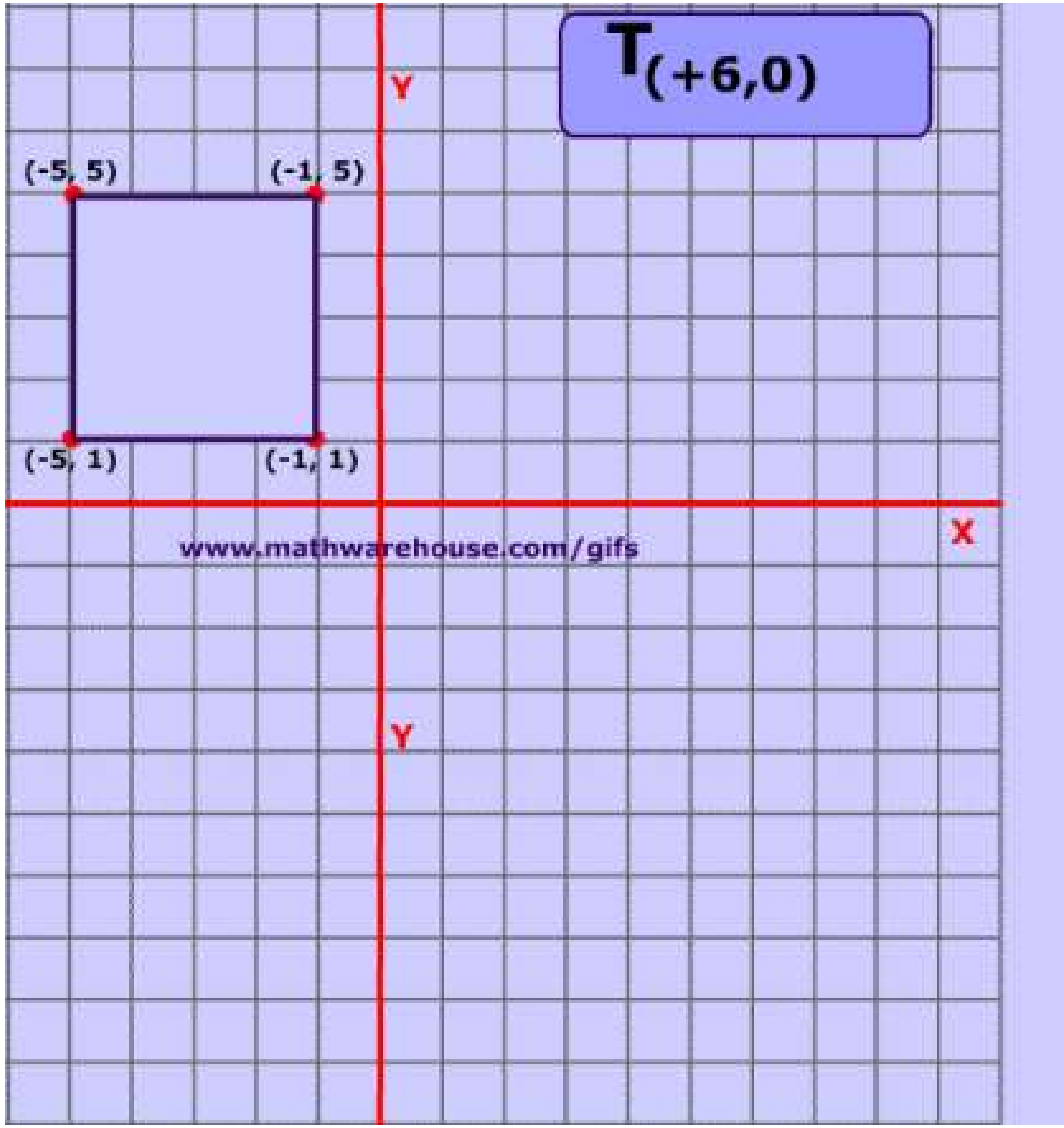


Fig. 4.1. Translation of a lamina and an ellipse



2-D TRANSLATION



2-D TRANSLATION

Matrix representation

for a point, **(x_T , y_T) = (x, y)+ T**

Where T is the translation matrix **= (t_x , t_y)**

(x+ y) initial coordinates

(x_T , y_T) are Final coordinates

for a line,

$$\begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix} + \begin{bmatrix} dx_1 & dx_2 \\ dy_1 & dy_2 \end{bmatrix} = \begin{bmatrix} x_{T1} & x_{T2} \\ y_{T1} & y_{T2} \end{bmatrix}$$

$\begin{matrix} x_1 & x_2 \\ y_1 & y_2 \end{matrix}$ Coordinates of line PQ

Point P displaced by dx_1 in X-direction and dy_1 in y-direction

$$x' = x + t_x \qquad y' = y + t_y$$

Matrix form:

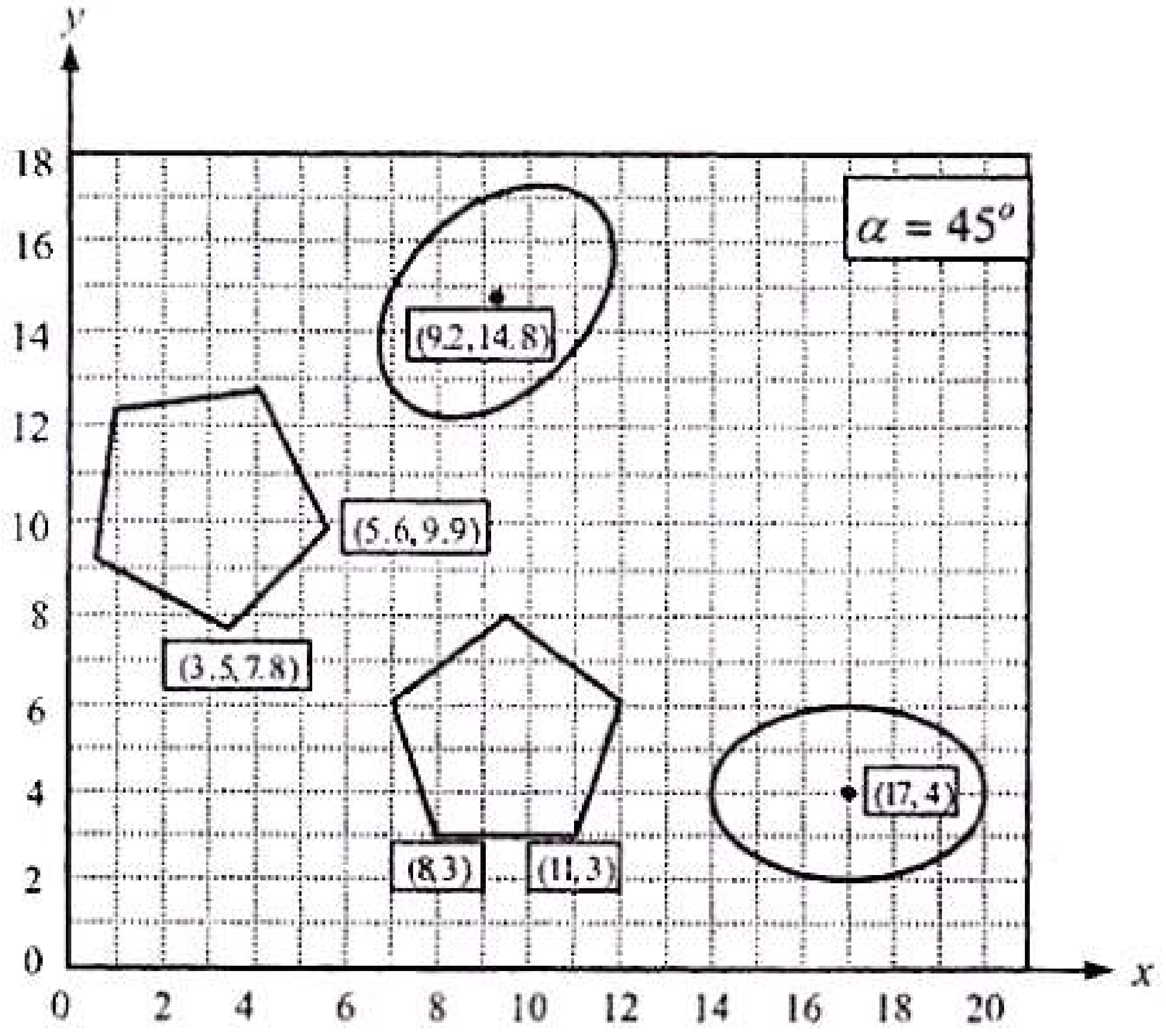
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix},$$

$$\mathbf{P' = P + T}$$

2-D ROTATION

- Used in creation of entities arranged in **a circular pattern** by creating the basic entity once and then copying/rotating the same on the circumference of a circle.
- **Axisymmetric models** can be generated using the rotation geometric transformation.
- Rotations are **rigid body** transformations that move objects without deformation
- A two-dimensional rotation can be defined by repositioning the object on a circular path in the x-y plane.
- To achieve the rotation, the following information is required:
  - Reference point about which the object is to be rotated
  - Rotation angle and direction of rotation (clockwise (+ve) or counterclockwise (-ve))

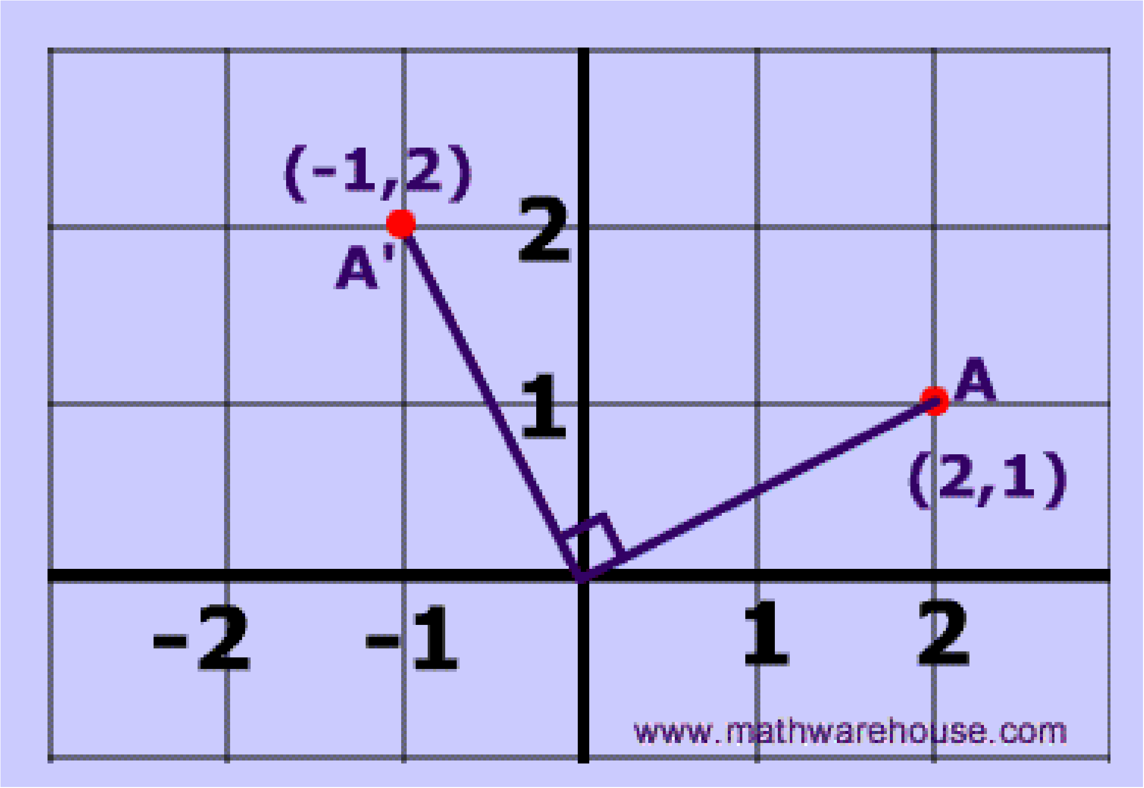
2-D ROTATION



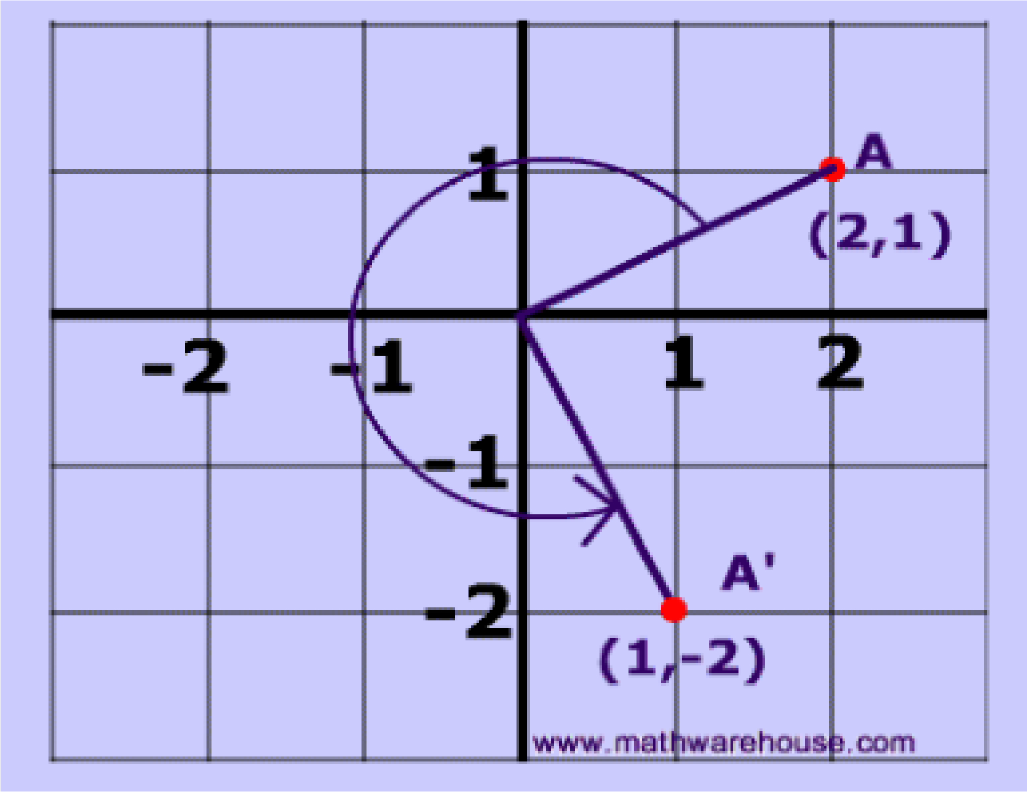
ig. 4.3. Rotation of a lamina and an ellipse about the origin

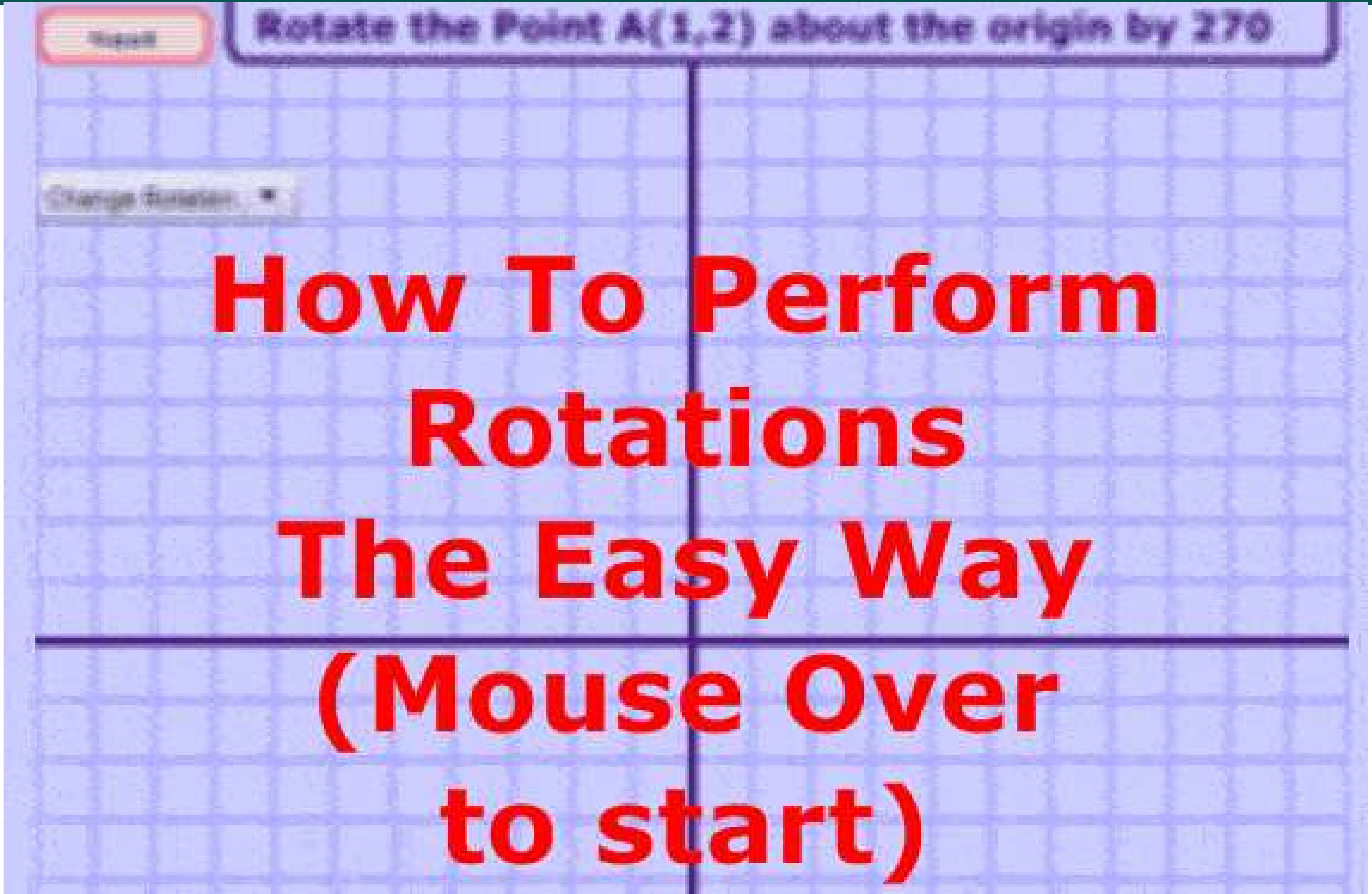
2-D ROTATION

Rotation by 90° about the origin: $R_{(origin,\ 90°)}$



Rotation by 270° about the origin: R(origin, 270°)

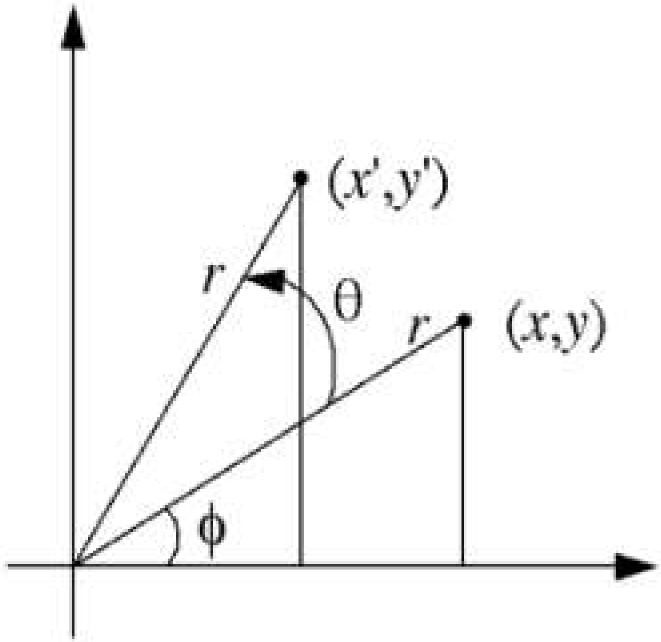



Rotate the Point A(1,2) about the origin by 270
How To Perform Rotations The Easy Way (Mouse Over to start)

2-D ROTATION

- Matrix representation

Rotation about origin:



Original polar coordinates:

$$x = r\cos\phi \qquad y = r\sin\theta$$

$$x' = r\cos(\phi+\theta) = r\cos\phi\cos\theta - r\sin\phi\sin\theta$$

$$y' = r\sin(\phi+\theta) = r\cos\phi\sin\theta + r\sin\phi\cos\theta$$

After substitution:

$$x' = x\cos\theta - y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$

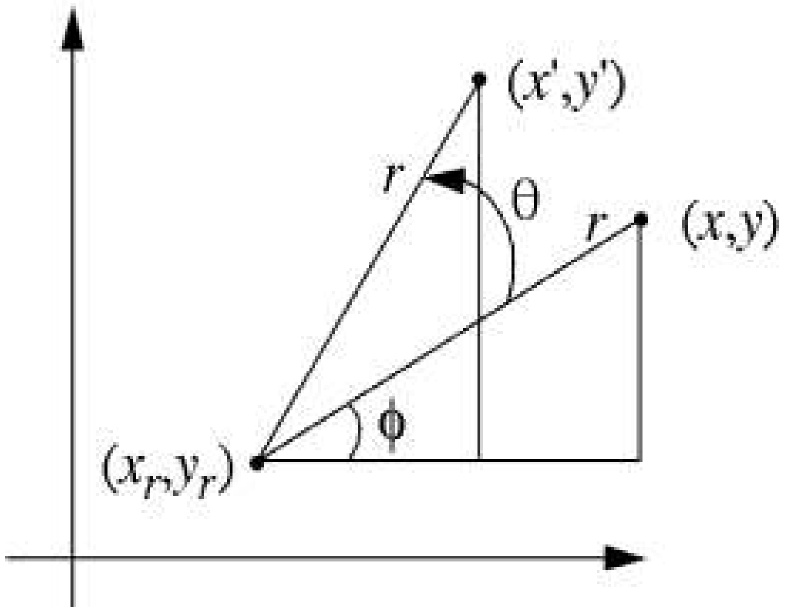
Matrix form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix},$$

$$\mathbf{P'} = \mathbf{R} \cdot \mathbf{P}$$

2-D ROTATION

Rotation about an arbitrary pivot point:



After substitution:

$$x' = x_r + (x - x_r)\cos\theta - (y - y_r)\sin\theta$$
$$y' = y_r + (x - x_r)\sin\theta + (y - y_r)\cos\theta$$

2-D REFLECTION

- A reflection (or mirroring) is a geometric transformation that produces mirror image of an object
- Reflection transformation is used for constructing the symmetric models
- If a model symmetric with respect to a plane, then one-half of the geometry is created at first, followed by the reflection transformation to generate the full model.
- A geometric model may be reflected through a plane, a line, or a point in space. The mirror image of 2D reflection is generated with respect to an axis of reflection.
- Reflection of a geometric model is achieved by rotating the object by 180° about the reflection axis.

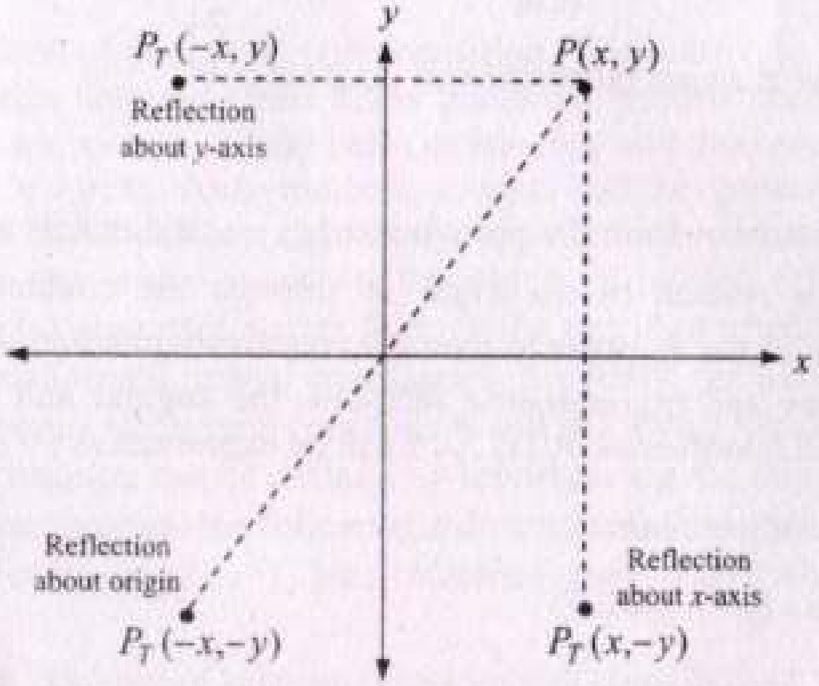


Fig. 4.5. Reflection of a point about the coordinate axes and the origin

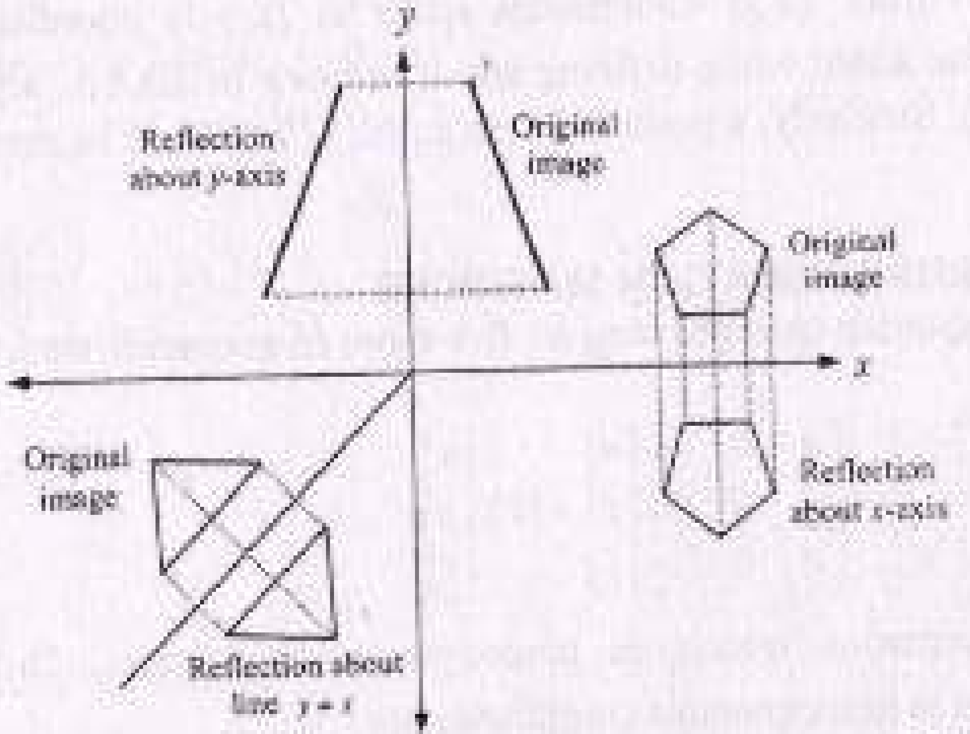
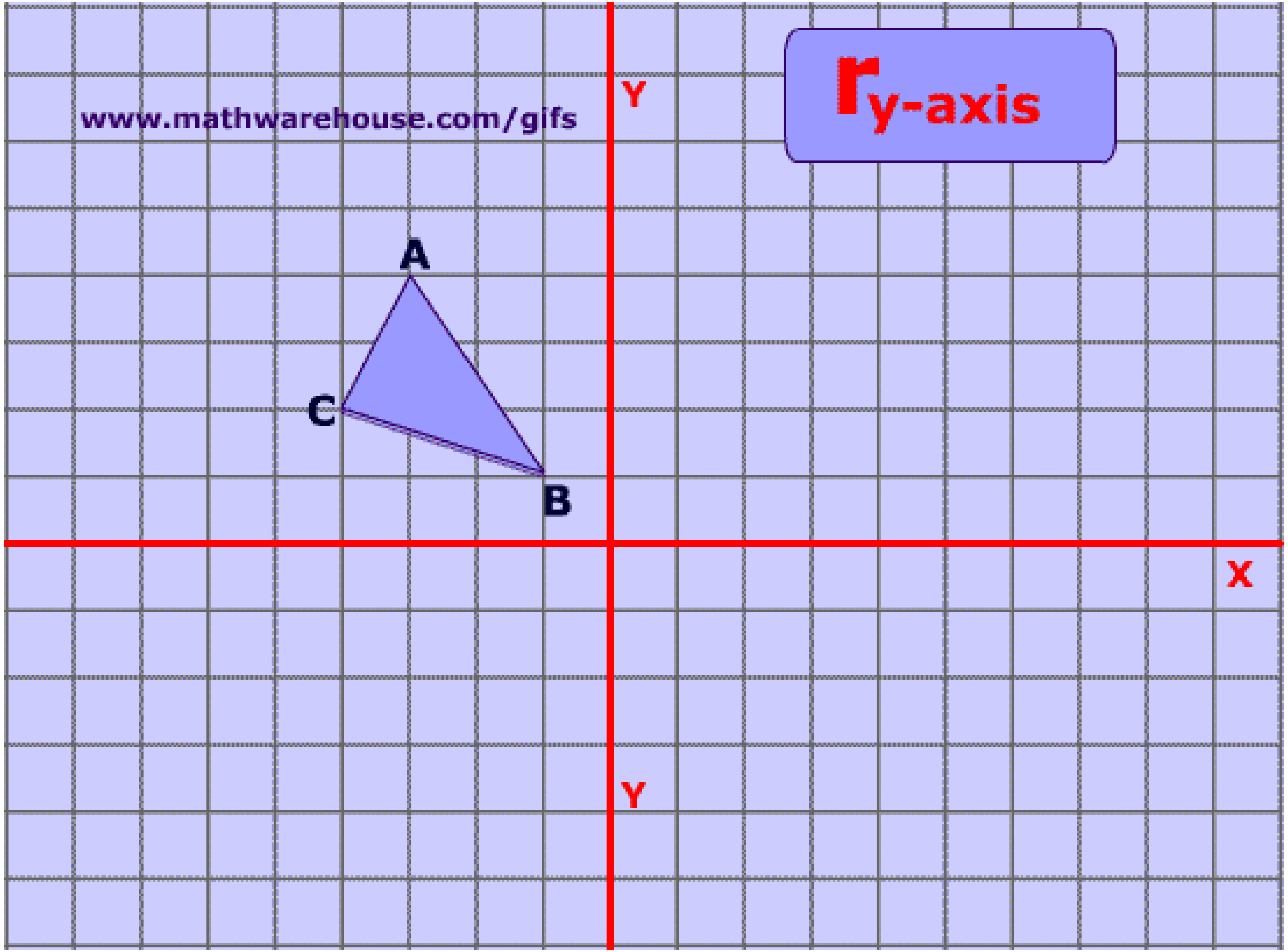


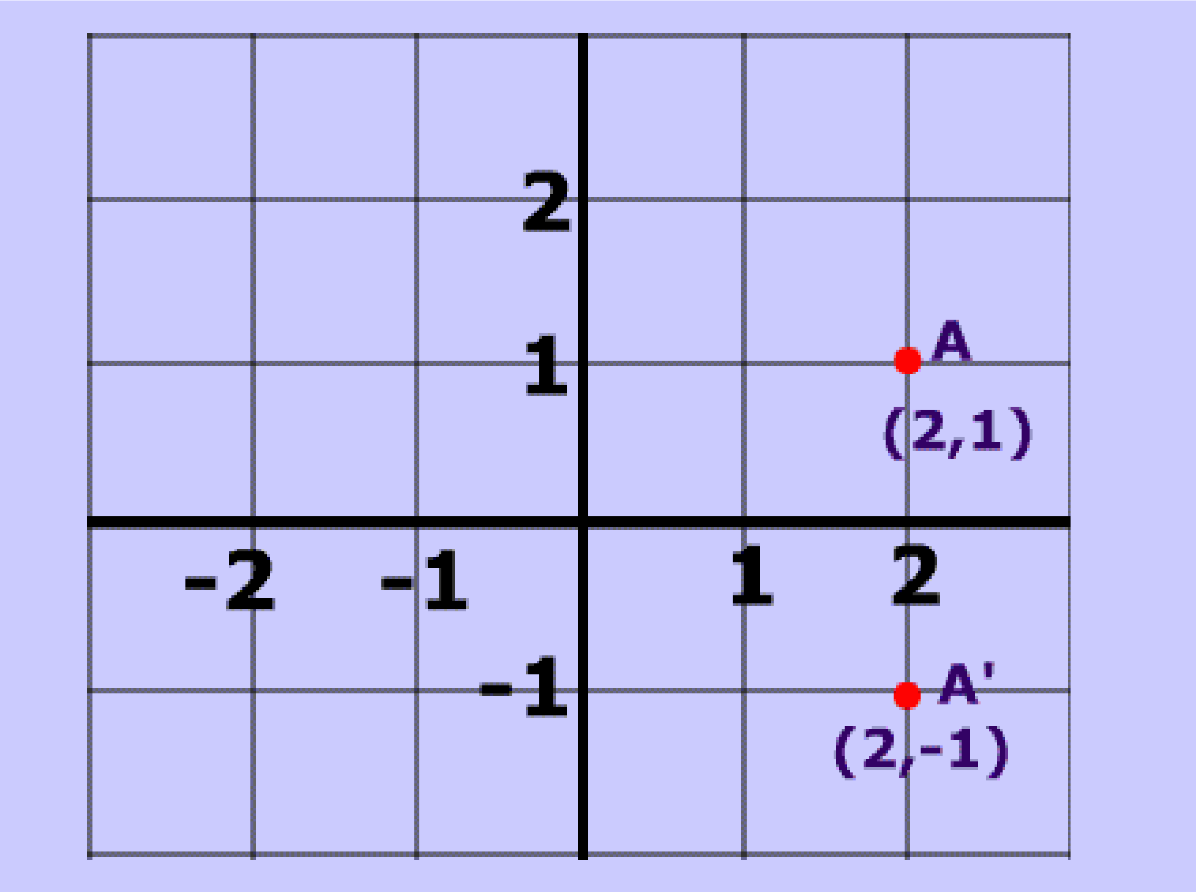
Fig. 4.6. Reflections of 2D objects about the coordinate axes and the line $y = x$

2-D REFLECTION

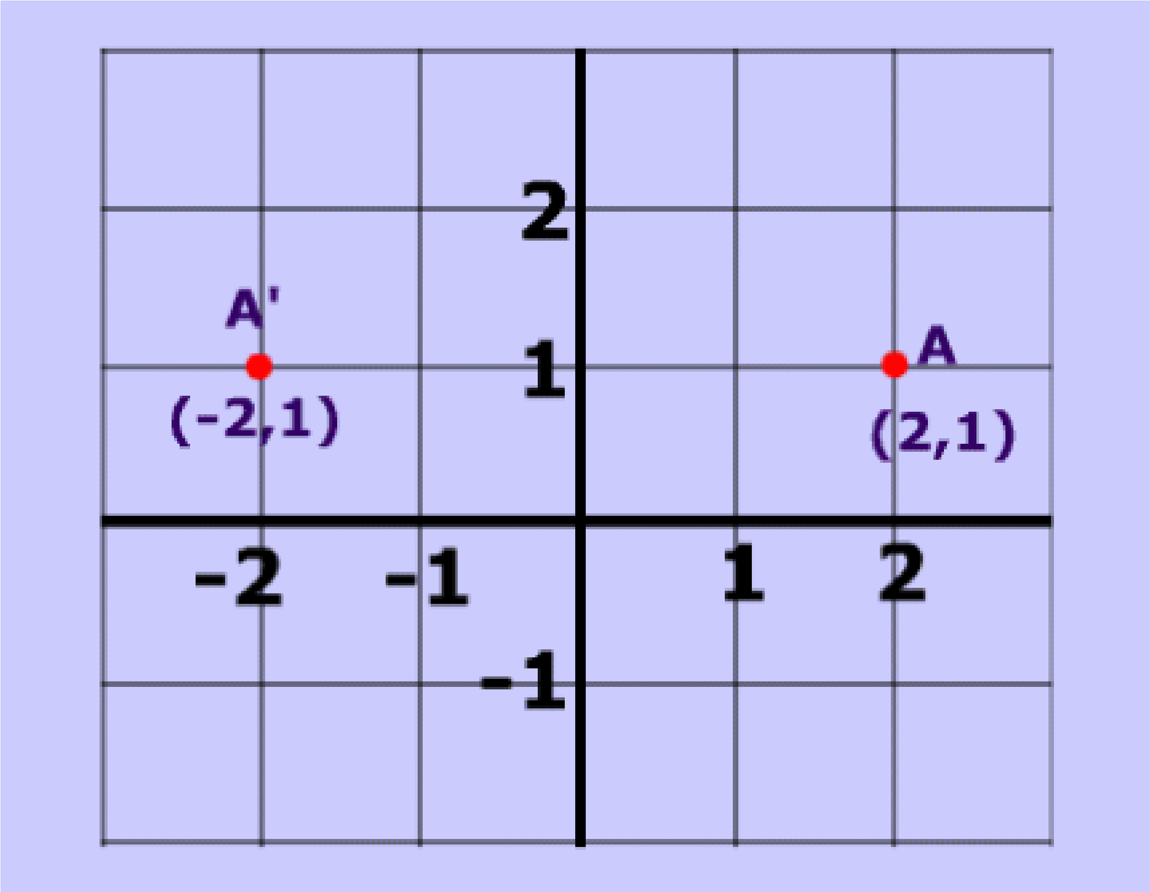


2-D REFLECTION

Reflection about X-Axis

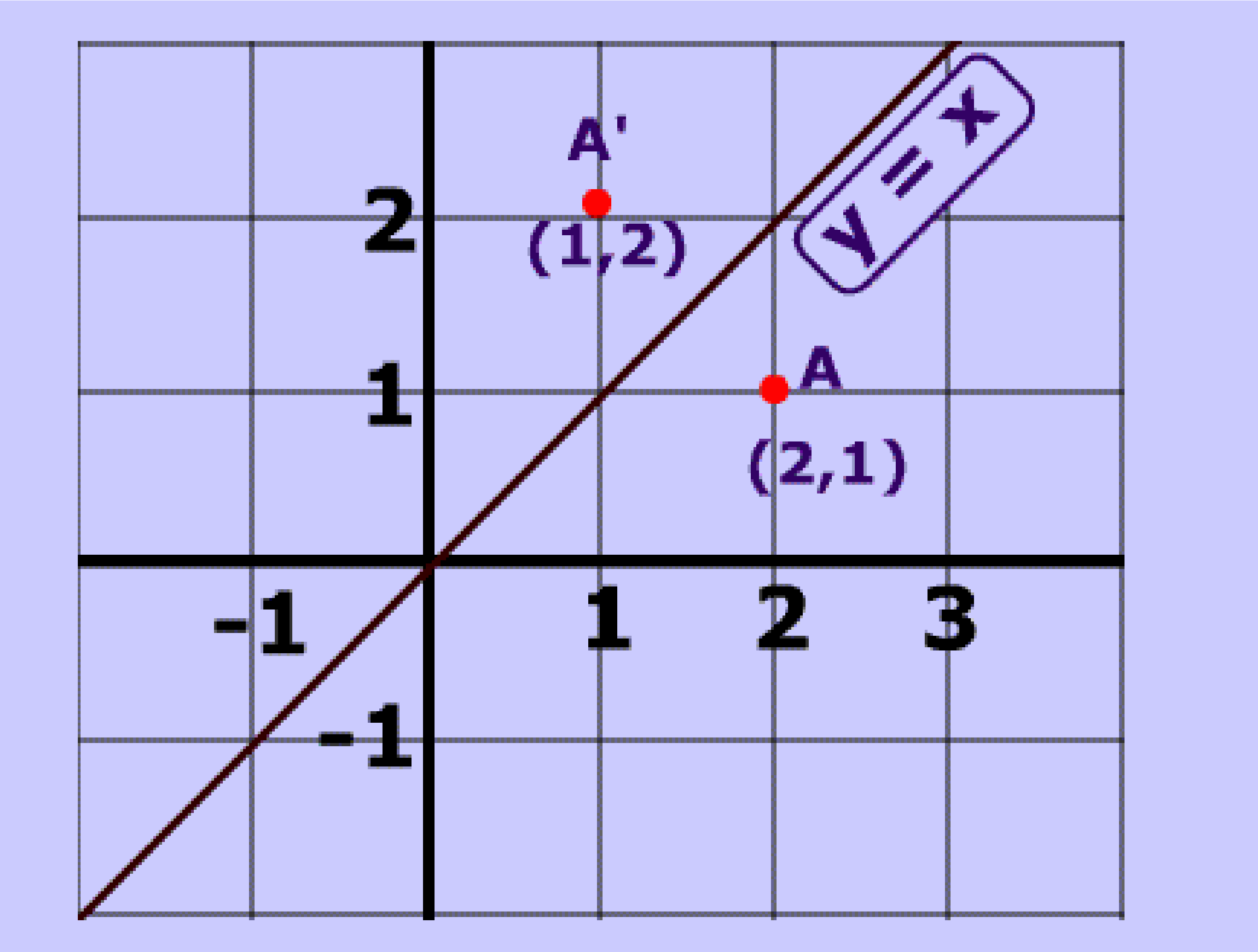


Reflection about Y-Axis



2-D REFLECTION

Reflection about Y=X Line



REFLECTION OF A POINTS

Reflection of a point about x-axis

$$\begin{Bmatrix} x_T \\ y_T \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix}$$

$$\begin{Bmatrix} x_T \\ y_T \end{Bmatrix} = \begin{Bmatrix} x \\ -y \end{Bmatrix}$$

2-D SCALING

- Scaling is used to change, increase or decrease, the size of an object.
- The scaling transformation can be applied whenever enlargement/magnification or reduction in size of an object within the image is required.
- Scaling can be done either in x or y direction or in both the directions simultaneously. Scaling factors can be specified as S_x and S_y along x and y coordinate axes, respectively.
- For polygons, scaling is performed by multiplying the coordinate values (x, y) of each vertex by the scaling factors S_x and S_y , respectively.
- Thus, transformed coordinates will be

$$x_T = S_x * x$$

$$y_T = S_y * y$$

2-D SCALING

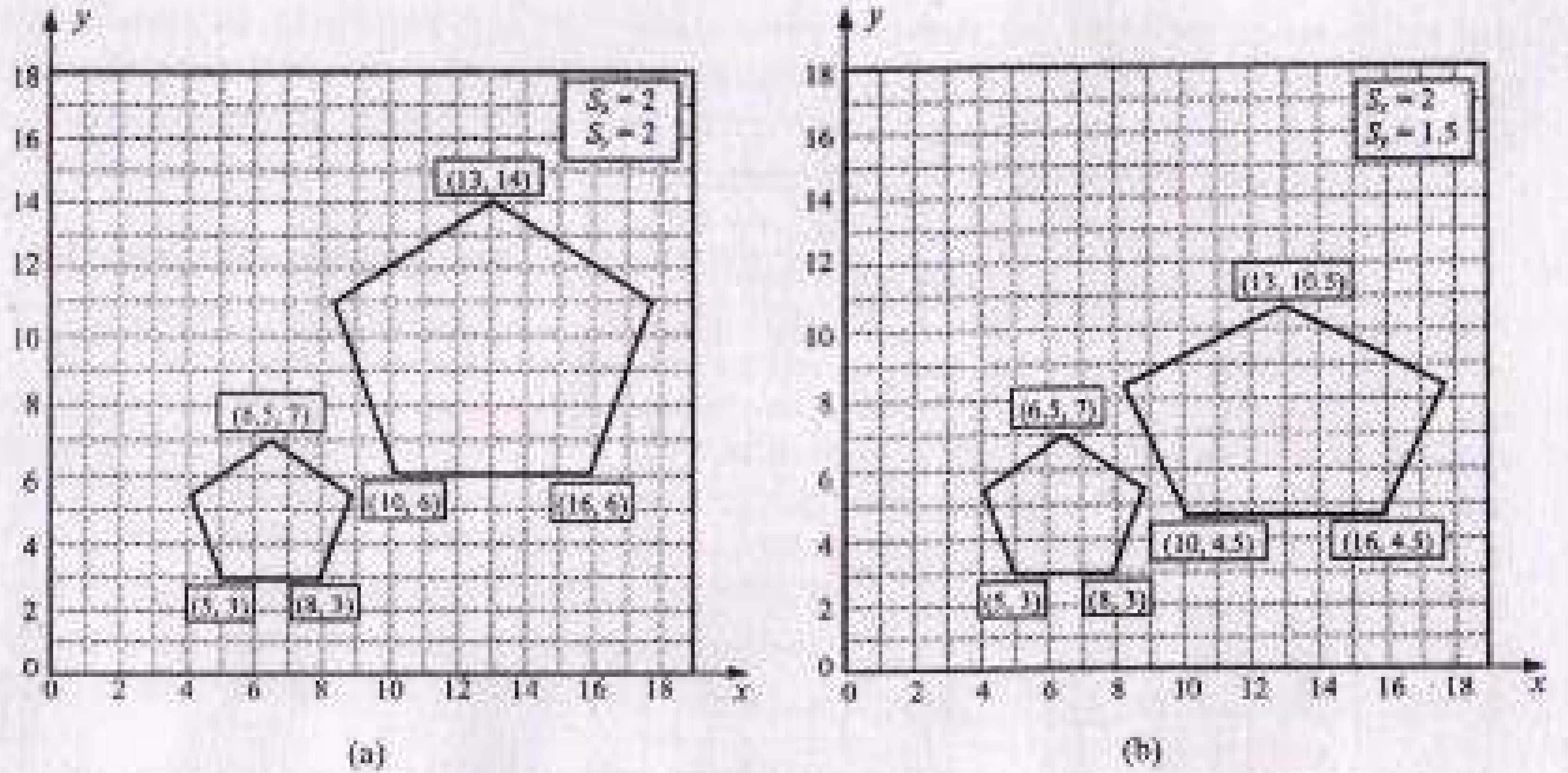
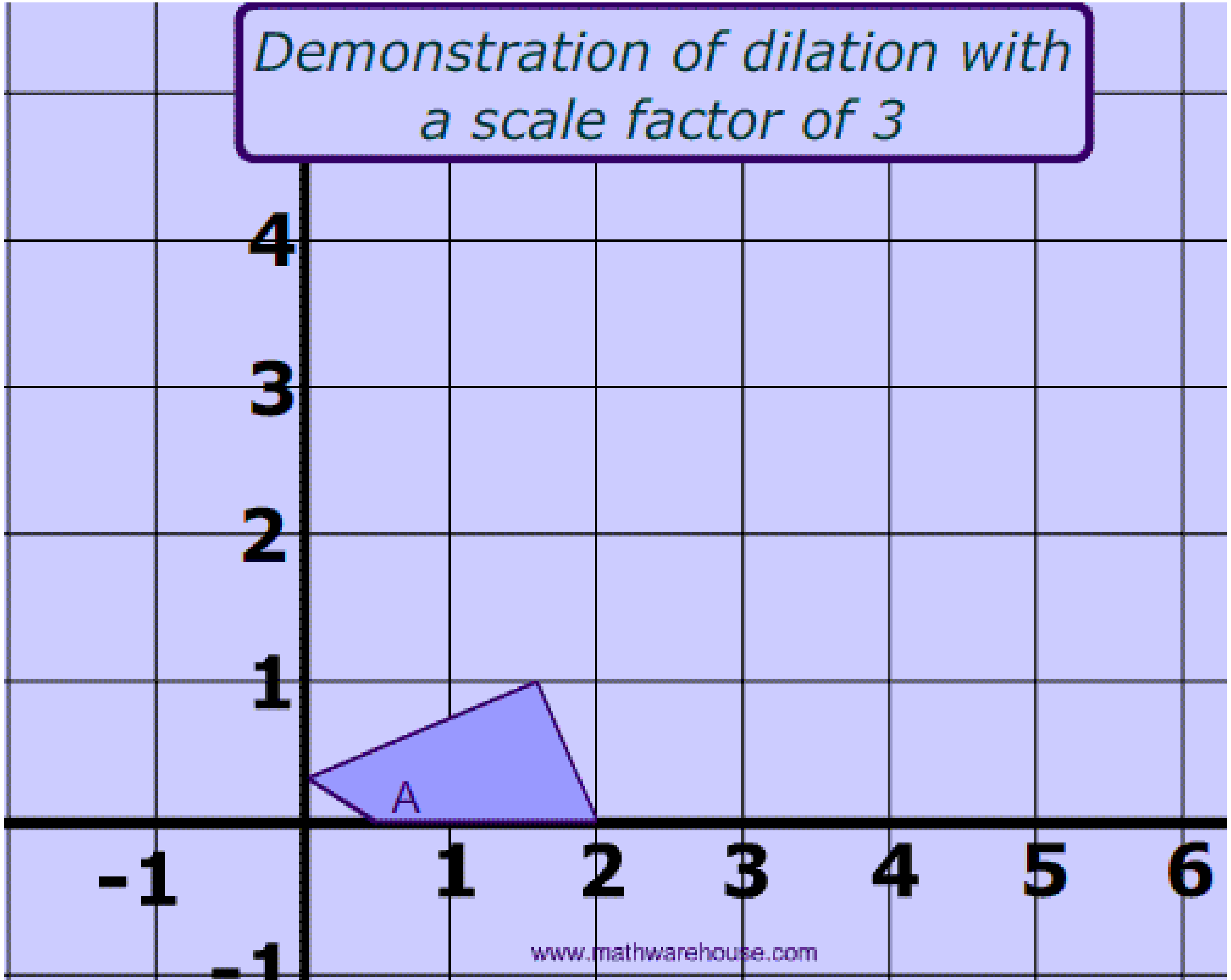


Fig. 4.4. Scaling of a lamina about the origin (a) uniform scaling (b) differential scaling

2-D SCALING


Demonstration of dilation with a scale factor of 3
4
3
2
1
A
-1
1
2
3
4
5
6
www.mathwarehouse.com

2-D SCALING

Scaling of a line

$$\begin{Bmatrix} x_{1T} & x_{2T} \\ y_{1T} & y_{2T} \end{Bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & Sy \end{bmatrix} \begin{Bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{Bmatrix}$$

S_x and S_y are the scaling factor in x and y direction

2-D SHEAR

- The geometric transformation that distorts shape of an object such that the transformed shape appears as if the object were composed of internal layers that slide over each other, is termed shear
- In other words, the sliding of internal layers of an object over the other layers to distort its shape is termed shear transformation
- Shear is the controlled distortion of an object model in x and y coordinates.
- Two common shear transformations are those in which x coordinate depends on y coordinate and vice versa.
- The transformed coordinates of a point P(x,y) , when shear along the x-direction takes place, is given as

$$x_T = x+S_{hx}*y \qquad \text{and} \qquad y_T = y$$

- Moreover. shear along the y-direction will be

$$y_T = y+S_{hy}*x \qquad \text{and} \qquad x_T = x$$

2-D SHEAR

- takes place, is given as

$$x_T = x + S_{hx} * y \qquad \text{and} \qquad y_T = y$$

- Moreover. shear along the y-direction will be

$$y_T = y + S_{hy} * x \qquad \text{and} \qquad x_T = x$$

$$\begin{Bmatrix} x_{1T} & x_{2T} \\ y_{1T} & y_{2T} \end{Bmatrix} = \begin{bmatrix} 1 & S_{hx} \\ S_{hy} & 1 \end{bmatrix} \begin{Bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{Bmatrix}$$

$$\begin{Bmatrix} x_T \\ y_T \end{Bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} = [T] \begin{Bmatrix} x \\ y \end{Bmatrix} \tag{4.14}$$

where $[T]$ is the transformation matrix. Depending upon the values and sign of different elements (A, B, C and D) of transformation matrix $[T]$, different types of geometric transformations can be obtained. The transformation matrices $[T]$ are different to achieve the rotation, reflection, scaling and shear transformations of an object model. Thus, the matrix equations for these transformations may be expressed as

$$\{P_T\} = [T_r].\{P\} \quad \text{for rotation} \tag{4.15a}$$

$$\{P_T\} = [R].\{P\} \quad \text{for reflection} \tag{4.15b}$$

$$\{P_T\} = [S].\{P\} \quad \text{for scaling} \tag{4.15c}$$

$$\{P_T\} = [Sh].\{P\} \quad \text{for shearing} \tag{4.15d}$$

The right-hand side terms of these matrix equations are the product of two matrices. Unfortunately, translation is not possible with this type of matrix representation because of addition of constant terms t_x and t_y associated with x_T and y_T coordinates (eqn. 4.1), i.e.

$$x_T = x + t_x \tag{4.16a}$$

$$y_T = y + t_y \tag{4.16b}$$

Alternatively, $\{P_T\} = [T_t] + \{P\}$ (4.16c)

The difficulty in image manipulation, incorporating all the five types of geometric transformations, can be removed if represented by a single matrix equation. This is possible if points are represented in homogeneous coordinates. The homogeneous coordinates are obtained by adding the third coordinate to a point. This facilitates the image manipulation with a single transformation matrix for all types of geometric transformations. Thus, there are mainly two advantages of using homogeneous coordinates representation:

I. It is possible to calculate overall transformation matrix through the matrix multiplications corresponding to each geometric transformation.

II. It also helps to achieve advanced type of transformation such as projection.

III. It removes many anomalies encountered in Cartesian geometry such as represe points at infinity and non-intersection of parallel lines.

In homogeneous coordinates system, mapping between the n-dimensional spaces with $(n+1)$-dimensional spaces occurs, if points in n-dimensional coordinates are represented by the corresponding $(n+1)$-dimensional coordinates. This is obtained by introducing a scale factor along the Cartesian coordinates. In 2D coordinates, instead of being represented by a pair (x, y), each point is represented by triple coordinates (x', y', h), where $h \neq 0$ is the scale factor. The relationship between the Cartesian coordinates and homogeneous coordinates of a point is given by

$$x = x'/h, \quad y = y'/h$$

or

$$x' = x.h, \quad y' = y.h \tag{4.17}$$

Generally, $h = 1$ represents a homogeneous coordinate $(x, y, 1)$ for a point (x, y) in computer graphics.

4.7 HOMOGENEOUS TRANSFORMATION MATRICES

In generalized form, the matrix equation incorporating all five types of geometric transformations may be expressed as

$$\begin{Bmatrix} x_T \\ y_T \\ 1 \end{Bmatrix} = \begin{bmatrix} A & B & 0 \\ C & D & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix} = [T] . \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix} \tag{4.18}$$

where $[T]$ represents a transformation matrix in homogeneous coordinates. Different 2D geometric transformation matrices in homogeneous coordinates are

Translation:

$$[T_t] = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.19}$$

Rotation: Counterclockwise rotation (ccw) in the xy plane, $[T_r] = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$ (4.20a)

Clockwise rotation (cw) in the xy plane, $[T_r] = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$ (4.20b)

Scaling:
$$[S] = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.21}$$

Reflection: About the x-axis,
$$[R_x] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.22a}$$

About the y-axis,
$$[R_y] = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.22b}$$

About the origin,
$$[R] = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.22c}$$

2D TRANSLATION OF POINTS

Translation matrix (T) in 3X3 matrix format

$$\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix}$$

For *translation* operation, $\begin{Bmatrix} x_T \\ y_T \\ 1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix}$ or $\begin{Bmatrix} x_T \\ y_T \\ 1 \end{Bmatrix} = \begin{Bmatrix} x + t_x \\ y + t_y \\ 1 \end{Bmatrix}$

This mathematical notation means that original coordinates x and y are transformed as

$$x_T = x + t_x, \quad y_T = y + t_y$$

2D TRANSLATION OF LINE

$$\begin{Bmatrix} x_{1T} & x_{2T} \\ y_{1T} & y_{2T} \\ 1 & 1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ 1 & 1 \end{Bmatrix}$$

$$x_{1T} = x_1 + t_X \qquad x_{2T} = x_2 + t_X$$

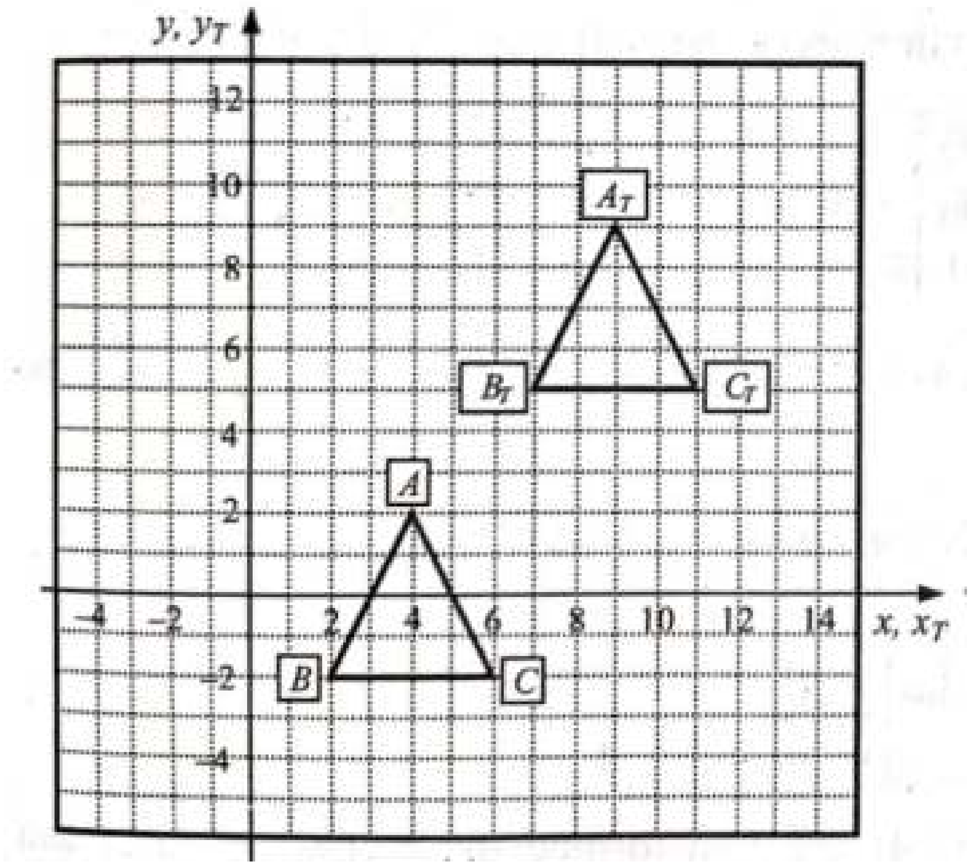
$$y_{1T} = y_1 + t_y \qquad y_{2T} = y_2 + t_y$$

- Triangle

$$\begin{Bmatrix} x_{1T} & x_{2T} & x_{3T} \\ y_{1T} & y_{2T} & y_{3T} \\ 1 & 1 & 1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{Bmatrix}$$

Question ????

- Triangular lamina having vertices A(4,2) , B(2,-2), AND C(6,-2) is subjected to the translation with translational distance t_x = 5 and t_y = 7 along the coordinate axes.



2D ROTATION OF POINTS

For *rotation* operation,

For Counter clockwise rotation

$$\begin{Bmatrix} x_T \\ y_T \\ 1 \end{Bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix} \quad \text{or} \quad \begin{Bmatrix} x_T \\ y_T \\ 1 \end{Bmatrix} = \begin{Bmatrix} x.\cos\alpha - y.\sin\alpha \\ x.\sin\alpha + y.\cos\alpha \\ 1 \end{Bmatrix}$$

For *rotation* operation,

For Clockwise rotation

$$\begin{Bmatrix} x_T \\ y_T \\ 1 \end{Bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix} \quad \text{or} \quad \begin{Bmatrix} x_T \\ y_T \\ 1 \end{Bmatrix} = \begin{Bmatrix} x.\cos\alpha - y.\sin\alpha \\ x.\sin\alpha + y.\cos\alpha \\ 1 \end{Bmatrix}$$

2D ROTATION OF LINE

$$\begin{Bmatrix} x_{1T} & x_{2T} \\ y_{1T} & y_{2T} \\ 1 & 1 \end{Bmatrix} = \begin{bmatrix} \cos\propto & -sin\propto & 0 \\ \sin\propto & cos\propto & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ 1 & 1 \end{Bmatrix}$$

For Counter clockwise rotation

$$\begin{Bmatrix} x_{1T} & x_{2T} \\ y_{1T} & y_{2T} \\ 1 & 1 \end{Bmatrix} = \begin{bmatrix} \cos\propto & sin\propto & 0 \\ -\sin\propto & cos\propto & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ 1 & 1 \end{Bmatrix}$$

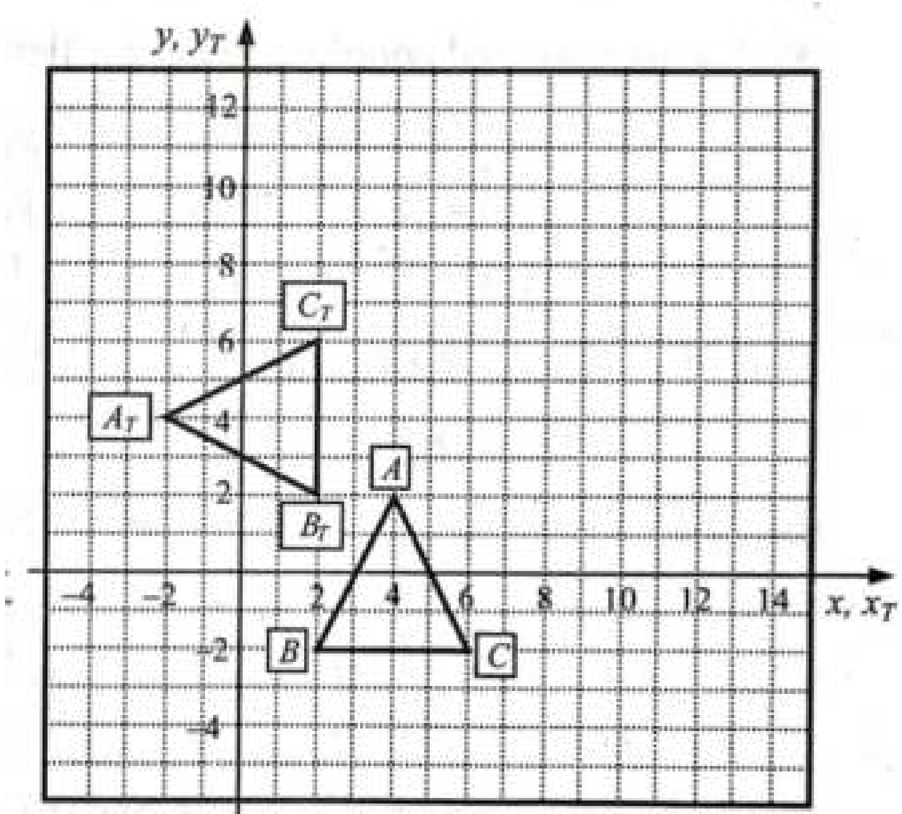
For Clockwise rotation

- Triangle

$$\begin{Bmatrix} x_{1T} & x_{2T} & x_{3T} \\ y_{1T} & y_{2T} & y_{3T} \\ 1 & 1 & 1 \end{Bmatrix} = \begin{bmatrix} \cos\propto & -sin\propto & 0 \\ \sin\propto & cos\propto & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{Bmatrix}$$

Question ????

- Triangular lamina having vertices A(4,2) , B(2,-2), and C(6,-2) is rotated 90^0 in counterclockwise direction about the origin. Calculate the transformed coordinates.



2-D ROTATION

Transformation Matrix for rotation

***Rotation*:** Counterclockwise rotation (ccw) in the xy plane, $[T_r]=\begin{bmatrix}\cos\alpha & -\sin\alpha & 0\\ \sin\alpha & \cos\alpha & 0\\ 0 & 0 & 1\end{bmatrix}$

Clockwise rotation (cw) in the xy plane, $[T_r]=\begin{bmatrix}\cos\alpha & \sin\alpha & 0\\ -\sin\alpha & \cos\alpha & 0\\ 0 & 0 & 1\end{bmatrix}$

2-D REFLECTION

- Matrix representation
  - **Transformation Matrix for reflection**

Reflection: About the x-axis, $[R_x]=\begin{bmatrix}1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{bmatrix}$

About the y-axis, $[R_y]=\begin{bmatrix}-1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$

About the origin, $[R]=\begin{bmatrix}-1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{bmatrix}$

About the line $y=x$, $[R]=\begin{bmatrix}0 & 1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 1\end{bmatrix}$

REFLECTION OF A POINTS

Reflection about x-axis

$$\begin{Bmatrix} x_T \\ y_T \\ 1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix}$$

$$\begin{Bmatrix} x_T \\ y_T \\ 1 \end{Bmatrix} = \begin{Bmatrix} x \\ -y \\ 1 \end{Bmatrix}$$

REFLECTION OF LINE

$$\begin{Bmatrix} x_{1T} & x_{2T} \\ y_{1T} & y_{2T} \\ 1 & 1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ 1 & 1 \end{Bmatrix}$$

Reflection about x axis

$$\begin{Bmatrix} x_{1T} & x_{2T} \\ y_{1T} & y_{2T} \\ 1 & 1 \end{Bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ 1 & 1 \end{Bmatrix}$$

Reflection about y axis

$$\begin{Bmatrix} x_{1T} & x_{2T} \\ y_{1T} & y_{2T} \\ 1 & 1 \end{Bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ 1 & 1 \end{Bmatrix}$$

Reflection about origin

$$\begin{Bmatrix} x_{1T} & x_{2T} \\ y_{1T} & y_{2T} \\ 1 & 1 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ 1 & 1 \end{Bmatrix}$$

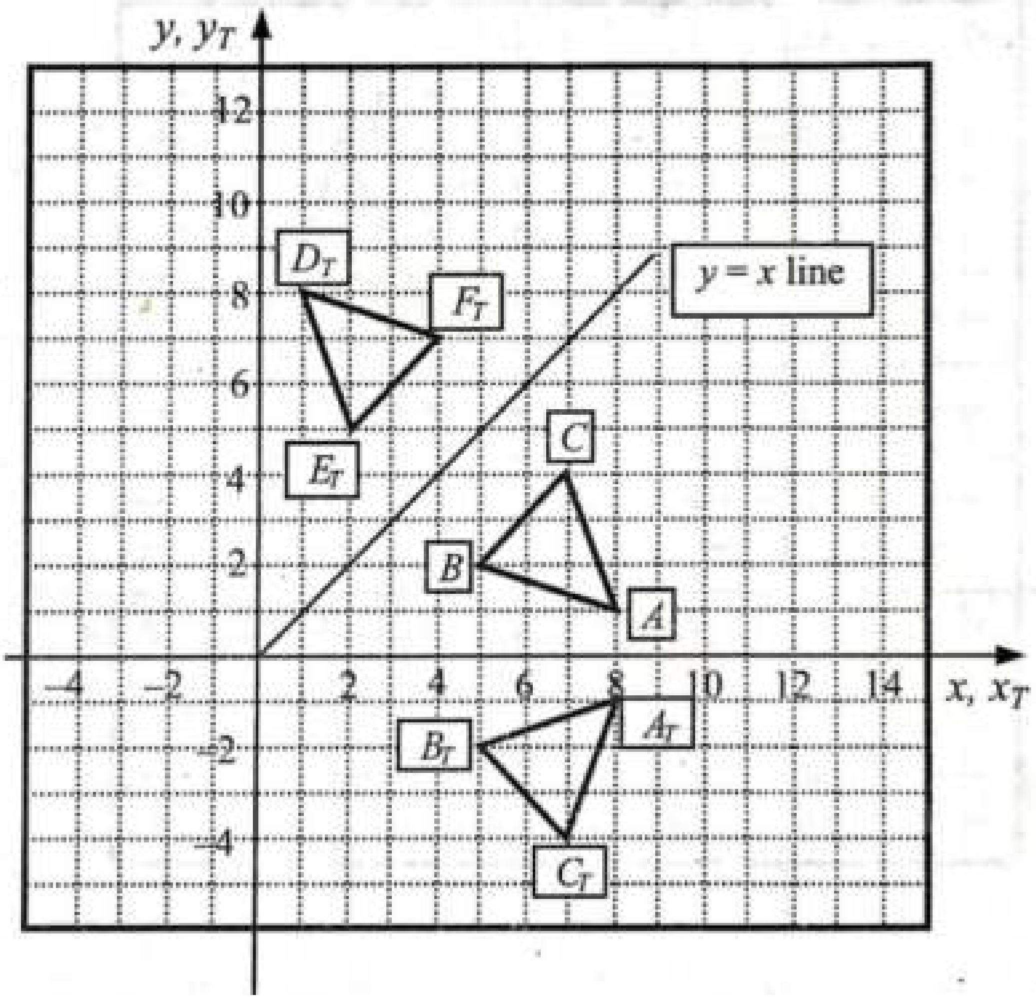
Reflection about x=y axis

- Triangle about x axis

$$\begin{Bmatrix} x_{1T} & x_{2T} & x_{3T} \\ y_{1T} & y_{2T} & y_{3T} \\ 1 & 1 & 1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{Bmatrix}$$

Question ????

- Triangular lamina having vertices A(8,1) , B(5,2), and C(7,4) is subjected to reflection a) about x-axis   b) about line x=y. Calculate the transformed coordinates.



Matrix representation

Transformation Matrix for Scaling

Scaling:

$$[S] = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Scaling of a line

$$\begin{Bmatrix} x_{1T} & x_{2T} \\ y_{1T} & y_{2T} \\ 1 & 1 \end{Bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ 1 & 1 \end{Bmatrix}$$

S_x and S_y are the scaling factor in x and y direction

Question ????

- Triangular lamina having vertices A(4,2) , B(4,4), and C(2,4) is sujectedd to the uniform scaling with scaling factor $S_X = S_y = 2$ and non uniform scaling with scaling factor $S_X = 3$ and $S_y = 0.5$, both about the origin. Calculate the transformed coordinates.

2-D SHEAR

Shear : Along the x-axis, $[Sh_x] = \begin{bmatrix} 1 & Sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Along the y-axis, $[Sh_y] = \begin{bmatrix} 1 & 0 & 0 \\ Sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Along the x and y-axes, $[Sh] = \begin{bmatrix} 1 & Sh_x & 0 \\ Sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Scaling and Shear: The effect of elements A, B, C and D in 3×3 transform (eqn. 4.18) may be separately identified. The terms B and C cause shear along x and (eqn. 4.23c), respectively. The terms A and D act as scale factors (eqn. 4.21). Thus 3×3 transformation matrix (eqn. 4.18) producing a combination of shear and scaling, is e

$$[T] = \begin{bmatrix} S_x & Sh_x & 0 \\ Sh_y & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Question ????

- Triangular lamina having vertices A(9,4) , B(6,4), and C(6,8) is subjected to shear along the coordinate axes with shear factors S_{hX} =1/2 and S_{hy} = 1/3 about the origin. Calculate the transformed coordinates.

- In CAD desired orientation may require more than one geometric transformation.
- The process of applying several transformation in succession to form one overall transformation.
- Any sequence of transformation termed as composite or combined transformation.
- This can be achieved by multiplying all the transformation matrices in sequence and overall transformation is represented by a single matrix. This is also termed as **concatenation of transformation**.

2D COMBINED TRANSFORMATION

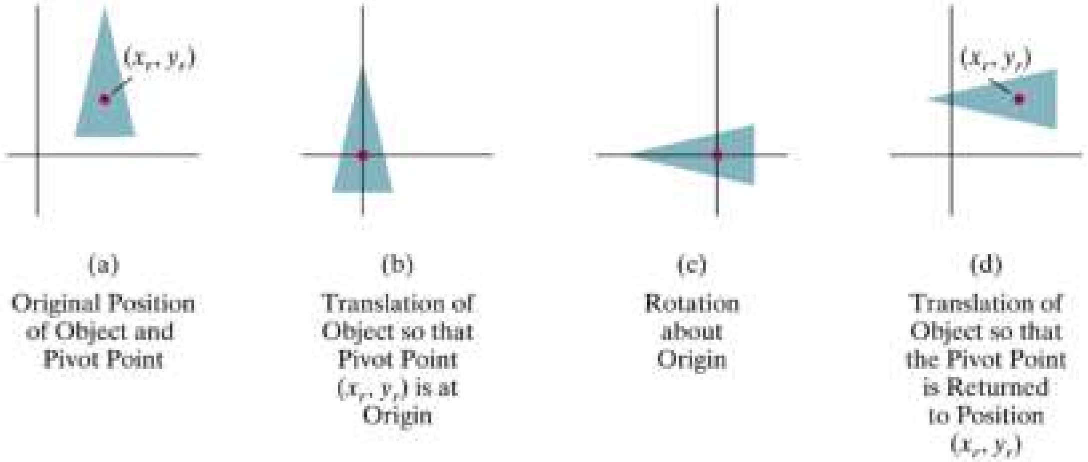


Figure 5-9

A trans-formation sequence for rotating an object about a specified pivot point using the rotation matrix $\mathbf{R}(\theta)$ of transformation 5-19.

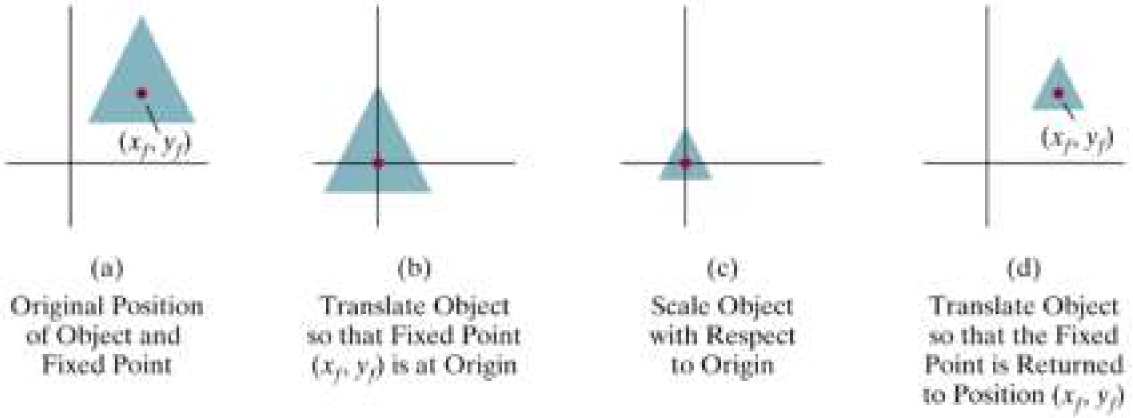


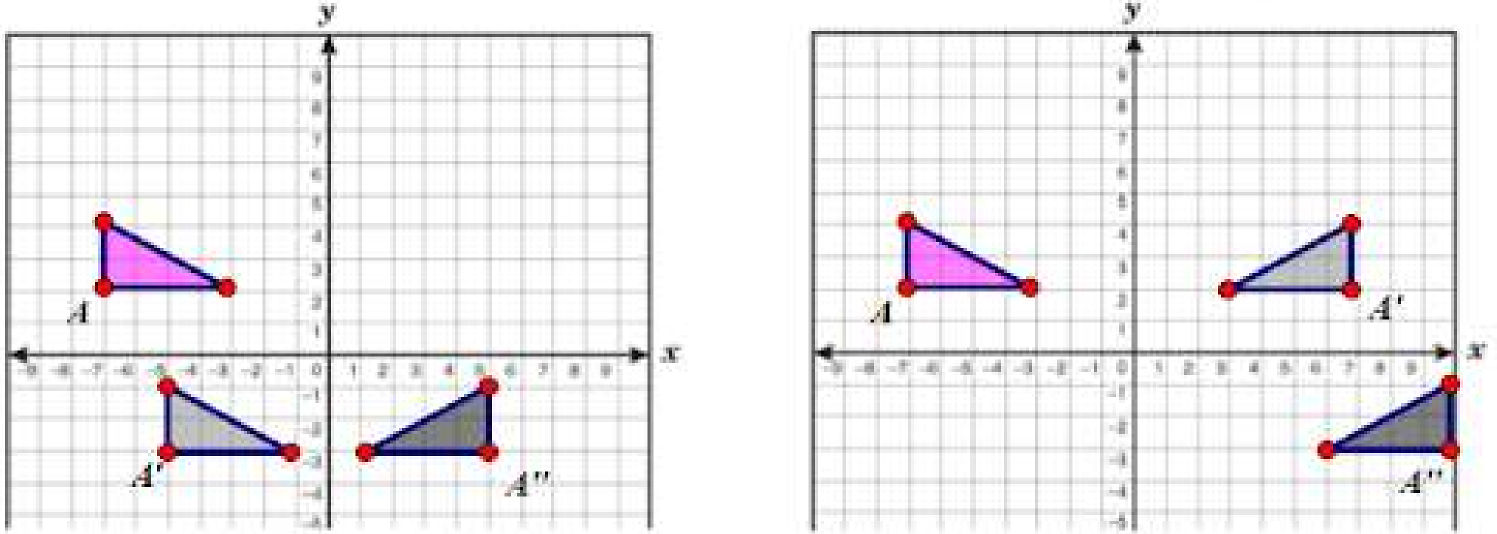
Figure 5-10

A trans-formation sequence for scaling an object with respect to a specified fixed position using the scaling matrix $\mathbf{S}(s_x, s_y)$ of transformation 5-21.

- Matrix multiplication is non-commutative (A.B ≠ B.A)

A translation of <3,-5> followed by a reflection over the y axis.

A reflection over the y axis followed by a translation of <3,-5>.



Notice that the two composite transformations result in different locations,
thus the order that transformations are done is very important to the result.

- Matrix associative (A.B.C = (A.B).C = (A. (B.C))

2D COMBINED TRANSFORMATION

If numbers of geometric transformations are taking place successively, e.g., operation sequence is

$$\text{Rotation } [T_r] \Rightarrow \text{Scaling } [S] \Rightarrow \text{Reflection } [R] \Rightarrow \text{Translation } [T_t]$$

overall transformation matrix is expressed as

$$[T] = [T_t].[R].[S].[T_r]$$

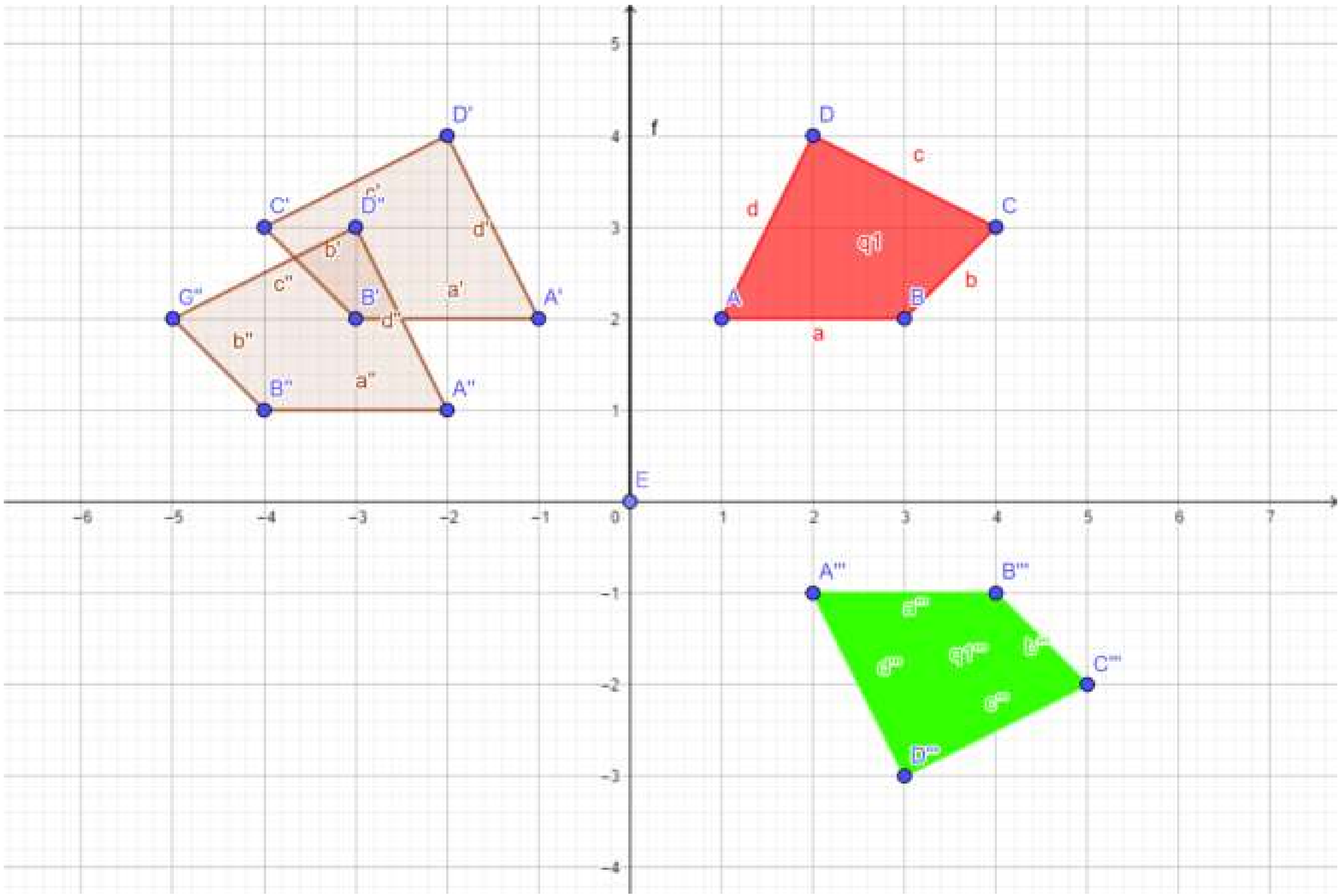
Moreover, matrix equation for the transformation of a point with position vector $\{X\}$ is expressed as

$$\{X_T\} = [T].\{X\}$$

If numbers of vertices are connected to form a 2D object, then matrix equation for the composite (or combined) geometric transformation (eqn. 4.39) is expressed as

$$\begin{Bmatrix} x_{T1} & x_{T2} & x_{T3} & - & - & x_{Tn} \\ y_{T1} & y_{T2} & y_{T3} & - & - & y_{Tn} \\ 1 & 1 & 1 & 1 & 1 & 1 \end{Bmatrix} = [T].\begin{Bmatrix} x_1 & x_2 & x_3 & - & - & x_n \\ y_1 & y_2 & y_3 & - & - & y_n \\ 1 & 1 & 1 & 1 & 1 & 1 \end{Bmatrix}$$

2D COMBINED TRANSFORMATION



ROTATION ABOUT AN ARBITRARY POINT

In subsection 4.3.2.1, the rotations about the origin are considered. In general, rotations about an arbitrary point (x_r, y_r), other than the origin, can be accomplished by performing three steps taken in sequence:

I. Translate point (x_r, y_r) to the origin.

II. Rotate the object by the desired angle direction about the origin (alternatively, about an axis perpendicular to the plane of rotation).

III. Translate point (x_r, y_r) back to the original centre of rotation.

Thus, matrix equation for the rotation of position vector $\{X\}$ about the point (x_r, y_r), by an angle α in ccw direction in the xy plane is expressed as

$$\{X_T\} = [T_t]^{-1}.[T_r].[T_t].\{X\}$$

where $[T_t]$ = translation matrix obtained by adding translational distances $-x_r$ and $-y_r$ to the position vector $\{X\}$ so that it coincides with the origin

$[T_r]$ = rotation matrix for rotation of position vector $\{X\}$ by an angle α in ccw direction so that it rotates about the origin in xy plane

$[T_t]^{-1}$ = Inverse translation matrix obtained by adding translational distances x_r and y_r to the position vector $\{X\}$ so that it translates back to the original position

2D COMBINED TRANSFORMATION

$$\begin{Bmatrix} x_T \\ y_T \\ 1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix} . \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix}$$

By carrying out the products of interior matrices, we have

$$\begin{Bmatrix} x_T \\ y_T \\ 1 \end{Bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & x_r.(1-\cos\alpha)+y_r.\sin\alpha \\ \sin\alpha & \cos\alpha & y_r.(1-\cos\alpha)-y_r.\sin\alpha \\ 0 & 0 & 1 \end{bmatrix} . \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix}$$

SCALING ABOUT AN ARBITRARY POINT

In subsection 4.3.3, the scaling transformation about the origin is considered. In general, scaling about an arbitrary point (x_s, y_s), other than the origin, can be accomplished by performing the three steps in sequence:

I. Translate point (x_s, y_s) to the origin.

II. Scale the object along x and y directions about the origin with scaling factors S_x and S_y.

III. Translate point (x_s, y_s) back to the original position.

Thus, matrix equation for scaling of position vector $\{X\}$ about the point having coordinates (x_s, y_s) is given as

$$\{X_T\} = [T_s]^{-1}.[S].[T_s].\{X\}$$

where $[T_s]$ = translation matrix obtained by adding translational distances $-x_s$ and $-y_s$ to the position vector $\{X\}$ so that it coincides with the origin.

$[S]$ = scaling matrix with scaling factors S_x and S_y about the origin.

$[T_s]^{-1}$ = inverse translation matrix obtained by adding translational distances x_s and y_s to the position vector $\{X\}$ so that it translates back to the original position.

$$\begin{Bmatrix} x_T \\ y_T \\ 1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & x_s \\ 0 & 1 & y_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_s \\ 0 & 1 & -y_s \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix}$$

By carrying out the products of interior matrices, we have

$$\begin{Bmatrix} x_T \\ y_T \\ 1 \end{Bmatrix} = \begin{bmatrix} S_x & 0 & x_s.(1-S_x) \\ 0 & S_y & y_s.(1-S_y) \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix}$$

- **Reflection through an arbitrary line**

Axis of Reflection Passes through Origin

The axis of reflection may be a line $(y = mx)$ passing through the origin having a slope $m = \tan\phi$ where ϕ is the angle made by the line with the x-axis. The overall transformation matrix for the reflection about this line can be achieved by performing the following transformations steps in sequence:

I. Rotate the line and the object about the origin until line is coincident with one of the coordinate axes (say, the line is rotated by an angle ϕ in clockwise direction about the z-axis so that it becomes coincident with the x-axis).

II. Reflect the object through the coordinate axis (say, x-axis).

III. Inverse rotate the object about the coordinate axis (say, z-axis).

Thus, the resulting concatenated matrix is

$$[T] = [T_r]^{-1}.[R].[T_r]$$

The rotations and reflections are also applied to the object to be transformed. The matrix equation for the reflection of position vector $\{X\}$ about the line $(y = mx)$ is given by

$$\{X_T\} = [T].\{X\}$$

Axis of Reflection is Arbitrary Line

When the axis of reflection is a line $(y = mx + c)$, not passes through the origin, having a slope $m = \tan\phi$ where ϕ is the angle, which line makes with the x-axis. The overall transformation matrix can be obtained by performing the following steps in sequence:

I. Translate the line and the object so that the line passes through the origin.

II. Rotate the line and object about the z-axis until the line becomes coincident with one of the coordinate axes (say, line is rotated by an angle ϕ in clockwise direction so that it becomes coincident with the x-axis).

III. Reflect the object through the coordinate axis (say, x-axis).

IV. Inverse rotate the object about the coordinate axis (say, z-axis).

V. Translate back the line/object to the original position

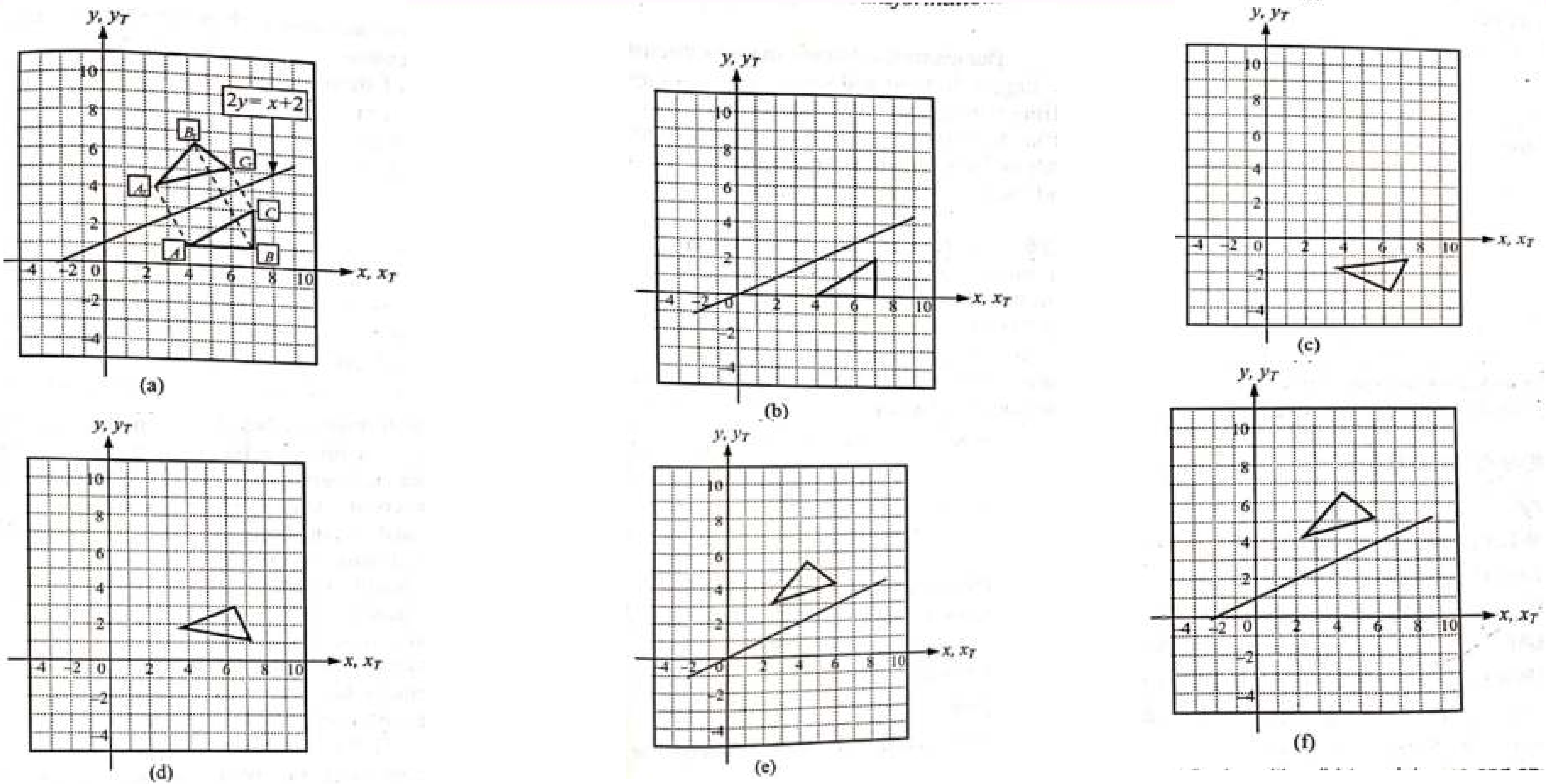
Thus, the resulting concatenated matrix is given by

$$[T] = [T_t]^{-1}.[T_r]^{-1}.[R].[T_r].[T_t]$$

The translations, rotations and reflection are also applied to the object. The matrix equation for the reflection of position vector $\{X\}$ about the line $(y = mx + c)$ is given as

$$\{X_T\} = [T].\{X\}$$

2D COMBINED TRANSFORMATION



(a) (b) (c) (d) (e) (f)

- **AFFINE TRANSFORMATION**

  - The geometric transformations such as translation, rotation, scaling, reflections and shear are the examples of two dimensional affine transformations.
  - Any 2D affine transformation can be expressed as a combination of these five basic transformations.
  - An affine transformation such as translation, -rotation and reflection preserves the angle and lengths, as well as lines remain parallel after the transformation. For these three transformations, the lengths and angles between the two lines remain same after the transformation.

- 2D analysis can be extended to 3D analysis.
- The matrices developed for two-dimensions can be extended to three dimensions.
- 3D geometric transformations can be obtained by including z-axis in the modeling.
- The matrices developed for two-dimensional transformations can be extended to three dimensions. Similar to two-dimensional cases, any sequence of transformations is represented as a single transformation matrix, formed by concatenating the matrices corresponding to the individual geometric transformations in sequence.

$$\begin{Bmatrix} x' \\ y' \\ z' \\ h \end{Bmatrix} = [T].\begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix}$$

where $[T]$ is the transformation matrix corresponding to a particular geometric transformation such as translation, rotation, scaling, reflection and shear.

$$\begin{Bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{Bmatrix} = [T].\begin{Bmatrix} x'/h \\ y'/h \\ z'/h \\ 1 \end{Bmatrix} = [T].\begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} \tag{5.2}$$

The general 4×4 matrix for three-dimensional homogeneous coordinates is expressed as

$$[T] = \begin{bmatrix} A & B & C & P \\ D & E & F & Q \\ G & I & J & R \\ L & M & N & S \end{bmatrix} \tag{5.3}$$

The above 4×4 transformation matrix can be partitioned into four separate sections as

$$[T] = \left[\begin{array}{ccc:c} A & B & C & P \\ D & E & F & Q \\ G & I & J & R \\ \hdashline L & M & N & S \end{array}\right] \tag{5.4}$$

Linear Transformation

- A linear transformation transforms an initial linear combination of vectors into the same line combination of transformed vectors. The upper-left 3 X 3 submatrix produces a linear transformation corresponding to the scaling, rotation, shear and reflection.

Translation

- The upper right 3 X 1 submatrix produces translation transformation.

Perspective Transformation

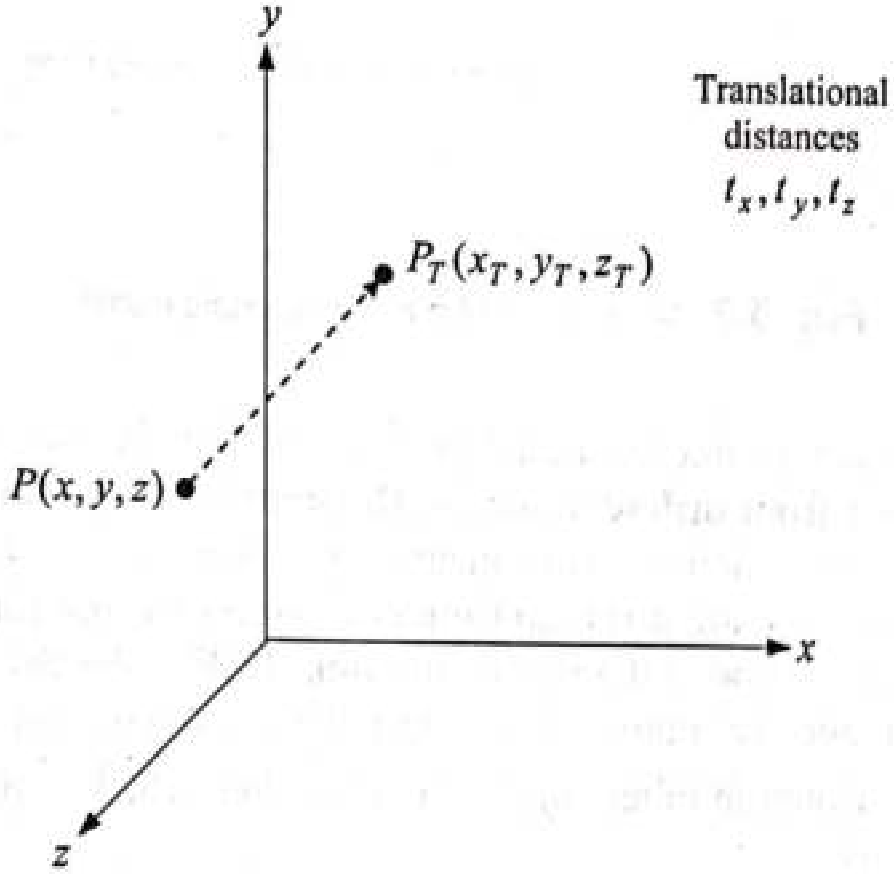
- The lower left 1 X 3 submatrix produces a perspective transformation.

Overall Scaling

- The lower right hand 1 X 1 submatrix produces overall scaling, i.e., all components of position vector are equally scaled.

In general, 4 >< 4 matrix yields a combination of shear, local scaling, rotation, reflection, translation, perspective projection and overall scaling of the object.

TRANSLATION



- In three-dimensional homogeneous coordinates, a point translate from P (x,y,z,1) to $P_T(x_T,y_T,z_T,1)$ using the matrix equation

$$\begin{Bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{Bmatrix} = [T_t].\begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.\begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix}$$

where $[T_t]$ is the translation matrix and t_x, t_y and t_z are the translational distances (any real values) along the coordinate axes, respectively. The matrix eqn. is equivalent to three equations

$$x_T = x + t_x,\; y_T = y + t_y,\; z_T = z + t_z$$

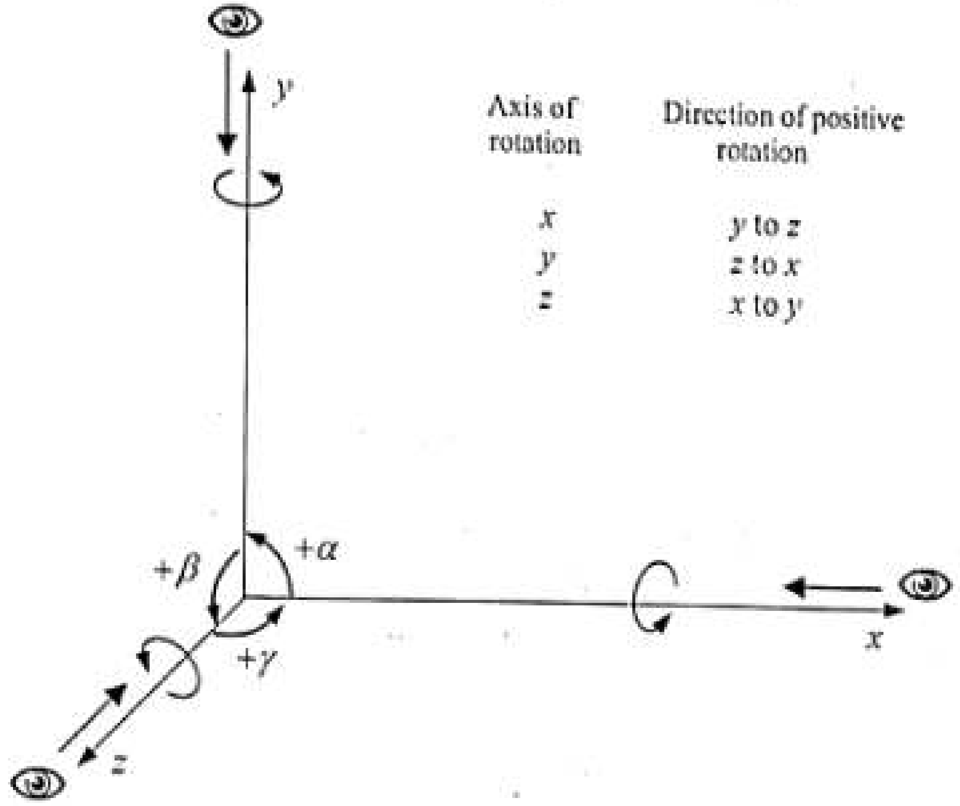
ROTATION

- Rotation is a **rigid body transformation** that moves the object without deformation. This means that every point on the object is rotated through the same angle.
- To generate a rotation transformation for the three-dimensional object, we must specify **the axis of rotation and angle of rotation**. Unlike the two-dimensional rotations, wherein rotation occurs in the xy plane, the easiest three-dimensional rotation may be taken about the coordinates axes.
- Rotation of an object about the line, which is parallel to the coordinate axes, is obtained in two steps:

I. Translate line so that it becomes coincident with any one of the coordinate axes.

II. Rotate object about the coordinate axes by the desired angle and direction.

- For the rotation about the x axis the x coordinate of position vector remains same because the rotation occurs in a plane perpendicular to the x axis.



g. 5.3. Positive rotations about the coordinate axes using right-hand coordinate system

The two-dimensional *z*-axis rotation can be extended to three dimensions using the equations

$$x_T = x.\cos\alpha - y.\sin\alpha$$
$$y_T = x.\sin\alpha + y.\cos\alpha$$
$$z_T = z$$

Parameter α is the positive rotation angle (ccw) about the *z*-axis in the *xy* plane. In homogeneous coordinates, three-dimensional *z*-axis rotation equation in matrix form is expressed as

$$[T_{rz}] = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

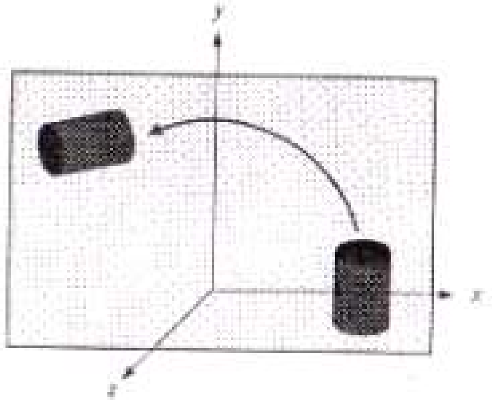


Figure shows the rotation of an object about the z-axis in a plane parallel to the xy plane. Since the object rotates about the z-axis; therefore, z coordinates remain unaltered after the rotation. This can be observed in the transformation matnx in which third row, third column element is equal to unity.

Transformation equations for the rotations about the other two axes can be obtained with a cyclic permutation of the coordinate parameters x, y and z in eqn. (5.8). Thus, we can replace $x \to y \to z \to x$. For x-axis rotation by an angle β in counterclockwise direction, the eqn. (5.8) is modified as

$$y_T = y.\cos\beta - z.\sin\beta$$

$$z_T = y.\sin\beta + z.\cos\beta$$

$$x_T = x$$

In homogeneous coordinates, the matrix equation is expressed as

$$[T_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta & 0 \\ 0 & \sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

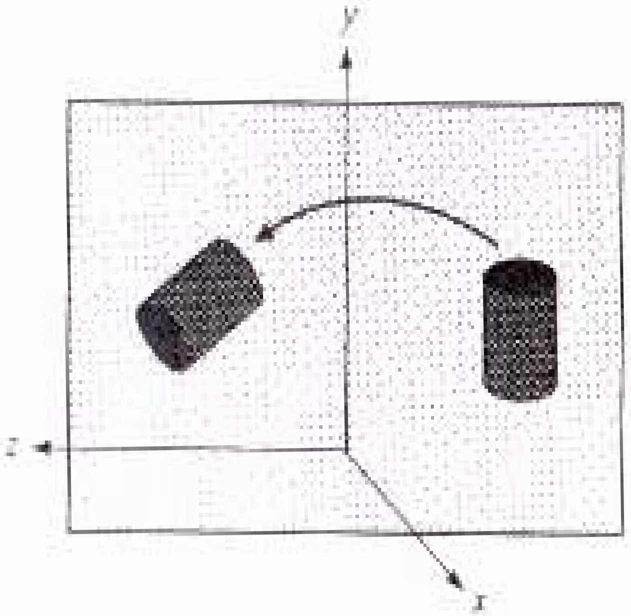


Fig. 5.5. Rotation of an object about the x-axis

Figure shows the rotation of an object about the x-axis in a plane parallel to the yz plane. Since the object rotates about the x-axis; therefore, x coordinates remain unaltered after the rotation. This can be observed in the transformation matnx in which first row, first column element is equal to unity.

For y-axis rotation by an angle θ in counterclockwise direction, the eqns. (5.10) are modified as

$$z_T = z.\cos\theta - x.\sin\theta$$

$$x_T = z.\sin\theta + x.\cos\theta$$

$$y_T = y$$

In homogeneous coordinates, the matrix equation for y-axis rotation is expressed as

$$[T_{ry}] = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

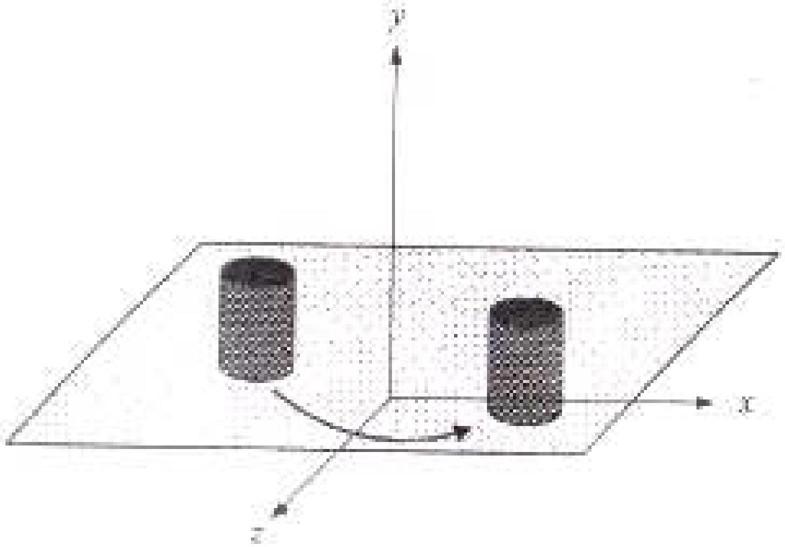


Fig. 5.6. Rotation of an object about the y-axis

SCALING

Scaling changes the size and position of object relative to origin in the database. In homogeneous coordinates, the matrix representation for scaling transformation of a point $P(x,y,z)$, relative to the coordinate origin, may be expressed as

$$\begin{Bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{Bmatrix} = [S].\begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.\begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} \qquad (5.14)$$

where S_x, S_y and S_z are the scaling factors along the coordinate axes, respectively, which can be assigned any positive value. The transformed coordinates for the scaling relative to the coordinate origin may be expressed as

$$x_T = S_x.x\,,\; y_T = S_y.y\,,\; z_T = S_z.z \qquad (5.15)$$

If scaling factors are not equal, the relative dimensions of the object are modified. However, the original shape of objects maintain if equal values of scaling factors, i.e., $S_x = S_y = S_z$ are considered. Figure 5.7 shows the uniform scaling of object with scaling factors set at 3.

Similar to the two dimensions, the overall scaling for three-dimensional objects can be obtained by assigning the fourth diagonal element of eqn. (5.14), the value other than 1. Hence, matrix equation for the overall scaling may be expressed as

$$\begin{Bmatrix} x' \\ y' \\ z' \\ h \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & S \end{bmatrix} \cdot \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} = \begin{Bmatrix} x \\ y \\ z \\ S \end{Bmatrix}$$

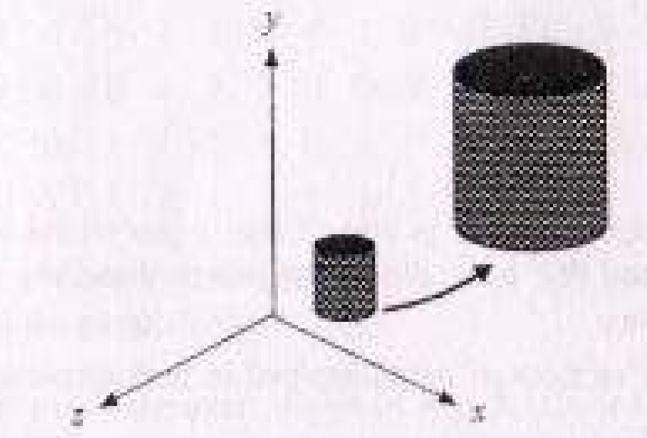


Fig. 5.7. Scaling of an object about the origin

The physical coordinates on $h = 1$ plane, by normalizing the coordinates, is given

$$\begin{Bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{Bmatrix} = \begin{Bmatrix} x/S \\ y/S \\ z/S \\ 1 \end{Bmatrix} \qquad (5.17)$$

If overall scaling factor $S < 1$, expansion occurs whereas compression occurs if $S > 1$. For example, if size of a unit cube is to be doubled, then $S = \frac{1}{2} = 0.5$, which is less than 1; hence, expansion occurs and vice versa.

REFLECTION

- Reflection is used for generating the symmetric models. The mirror image of 2D reflection is obtained by rotating the object 180 about the reflection axis.
- A three-dimensional reflection can be obtained either relative to a reflection axis or through a reflection plane.
- Reflection through a given axis is equivalent to 180° rotation about that axis.

The transformed coordinates for the reflection through the *xy* coordinate plane changes the sign of *z* coordinates of position vectors only, keeping *x* and *y* values unchanged. The reflection matrix through the *xy* plane is given as

$$[R_{xy}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.18)$$

Similar matrices for the reflections through the *xz* plane and *yz* plane, respectively, are

$$[R_{xz}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.19)$$

REFLECTION

The transformed coordinates for the reflection through the *xy* coordinate plane changes the sign of *z* coordinates of position vectors only, keeping *x* and *y* values unchanged. The reflection matrix through the *xy* plane is given as

$$[R_{xy}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.18)$$

Similar matrices for the reflections through the *xz* plane and *yz* plane, respectively, are

$$[R_{xz}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.19)$$

REFLECTION

$$[R_{yz}] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5.20)$$

In general, reflections about the other planes are obtained by the composite transformations such as rotations, followed by the reflections through the coordinate planes.

REFLECTION

$$[R_{yz}] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.20}$$

In general, reflections about the other planes are obtained by the composite transformations such as rotations, followed by the reflections through the coordinate planes.

SHEAR

Shear is the controlled distortion of an object model. Shear transformations can be used to distort the shape of an object model by sliding of internal layers over the other layers. In two dimensions, individual and simultaneous shear transformations of x and y coordinates are considered. In three dimensions, shear is extended relative to the z-axis. If we ignore the shear along the x and y axes, the following transformation produces z-axis shear

$$[Sh_z] = \begin{bmatrix} 1 & 0 & Sh_{13} & 0 \\ 0 & 1 & Sh_{23} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.21}$$

The transformation matrix (eqn. 5.21) is equivalent to the following shear equations:

$$\begin{aligned} x_T &= x + Sh_{13}.z \\ y_T &= y + Sh_{23}.z \\ z_T &= z \end{aligned} \tag{5.22}$$

Shear parameters Sh_{13} and Sh_{23} may be assigned any real values. The effect of z-axis shear is to change x and y coordinates (keeping z coordinate unchanged) of an object by an amount that is proportional to the z coordinate value. Thus, boundaries of planes that are perpendicular to the z-axis are shifted by an amount proportional to z. Figure 5.9 shows a unit cube sheared by the z-axis shear (eqn. 5.21) with shear parameters $Sh_{13} = Sh_{23} = 1$.

Similar matrices for the *x*-axis shear and *y*-axis shear, respectively, may be written as

$$[Sh_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ Sh_{21} & 1 & 0 & 0 \\ Sh_{31} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.23}$$

$$[Sh_y] = \begin{bmatrix} 1 & Sh_{12} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & Sh_{32} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.24}$$

The general form of shear transformation matrix is given as

$$[Sh] = \begin{bmatrix} 1 & Sh_{12} & Sh_{13} & 0 \\ Sh_{21} & 1 & Sh_{23} & 0 \\ Sh_{31} & Sh_{32} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.25}$$

The shear transformation matrix (eqn. 5.25) is equivalent to the following shear equations:

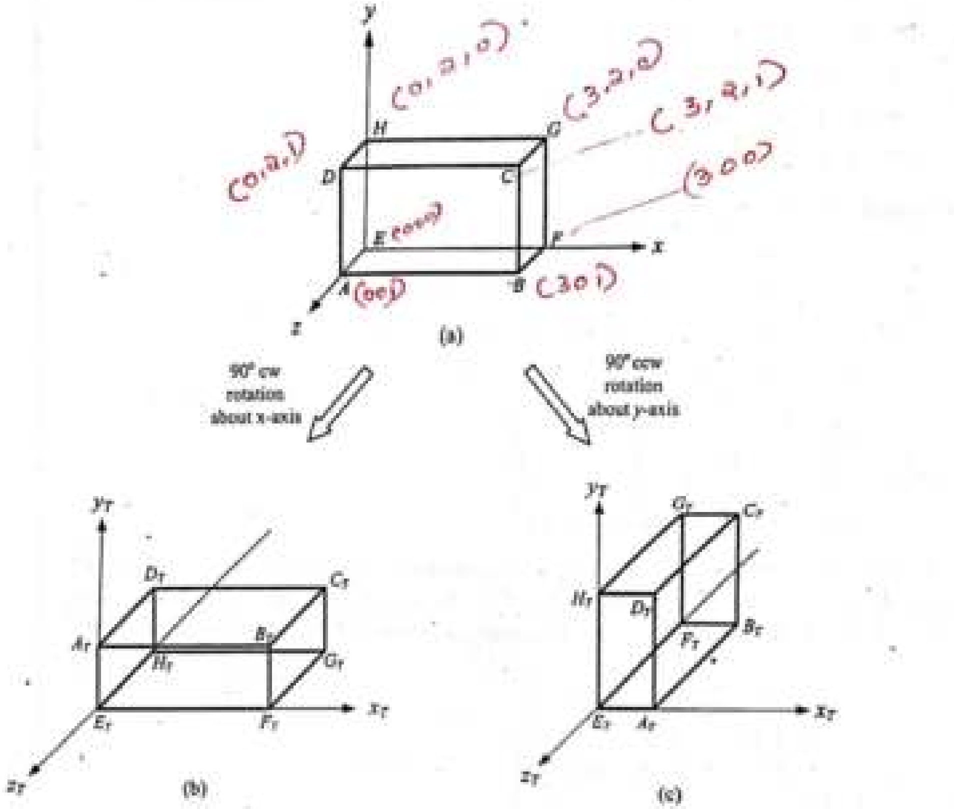
$$\begin{aligned} x_T &= x + Sh_{12}.y + Sh_{13}.z \\ y_T &= Sh_{21}.x + y + Sh_{23}.z \\ z_T &= Sh_{31}.x + Sh_{32}.y + z \end{aligned} \tag{5.26}$$

QUESTION ???

Problem 5.1: *A rectangular parallelepiped, shown in Fig. 5.10a, is subjected to individual rotations of 90° cw about the x-axis and 90° ccw about the y-axis. Find the transformed coordinates of its vertices for these rotations.*

Solution: The position vector matrix of vertices of parallelepiped in homogeneous coordinate system can be expressed as

$$\begin{array}{c} \begin{array}{cccccccc} A & B & C & D & E & F & G & H \end{array} \\ \{X\} = \begin{bmatrix} 0 & 3 & 3 & 0 & 0 & 3 & 3 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 & 2 & 2 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$



ORTHOGRAPHIC PROJECTION

- Parallel/Orthographic Projection
- Perspective Projection

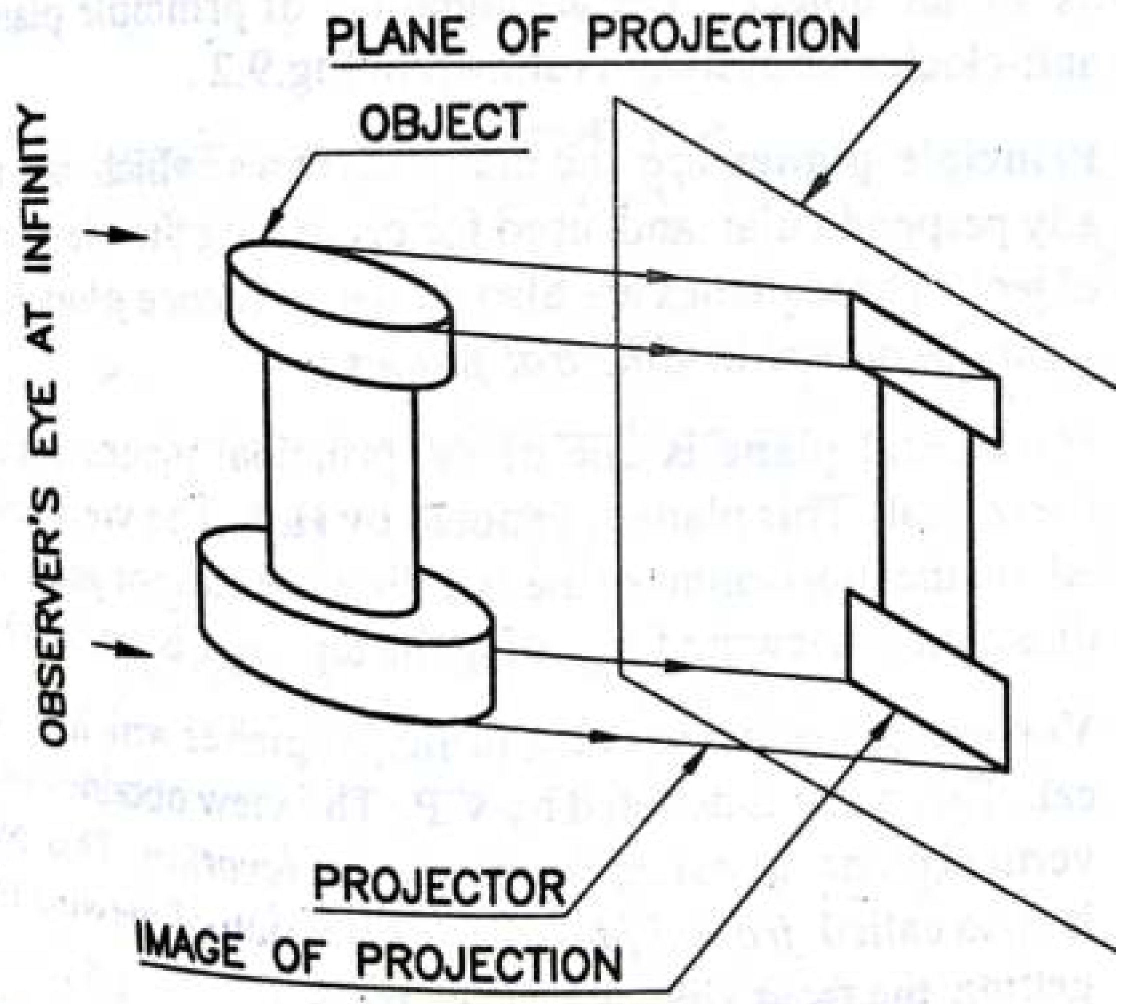
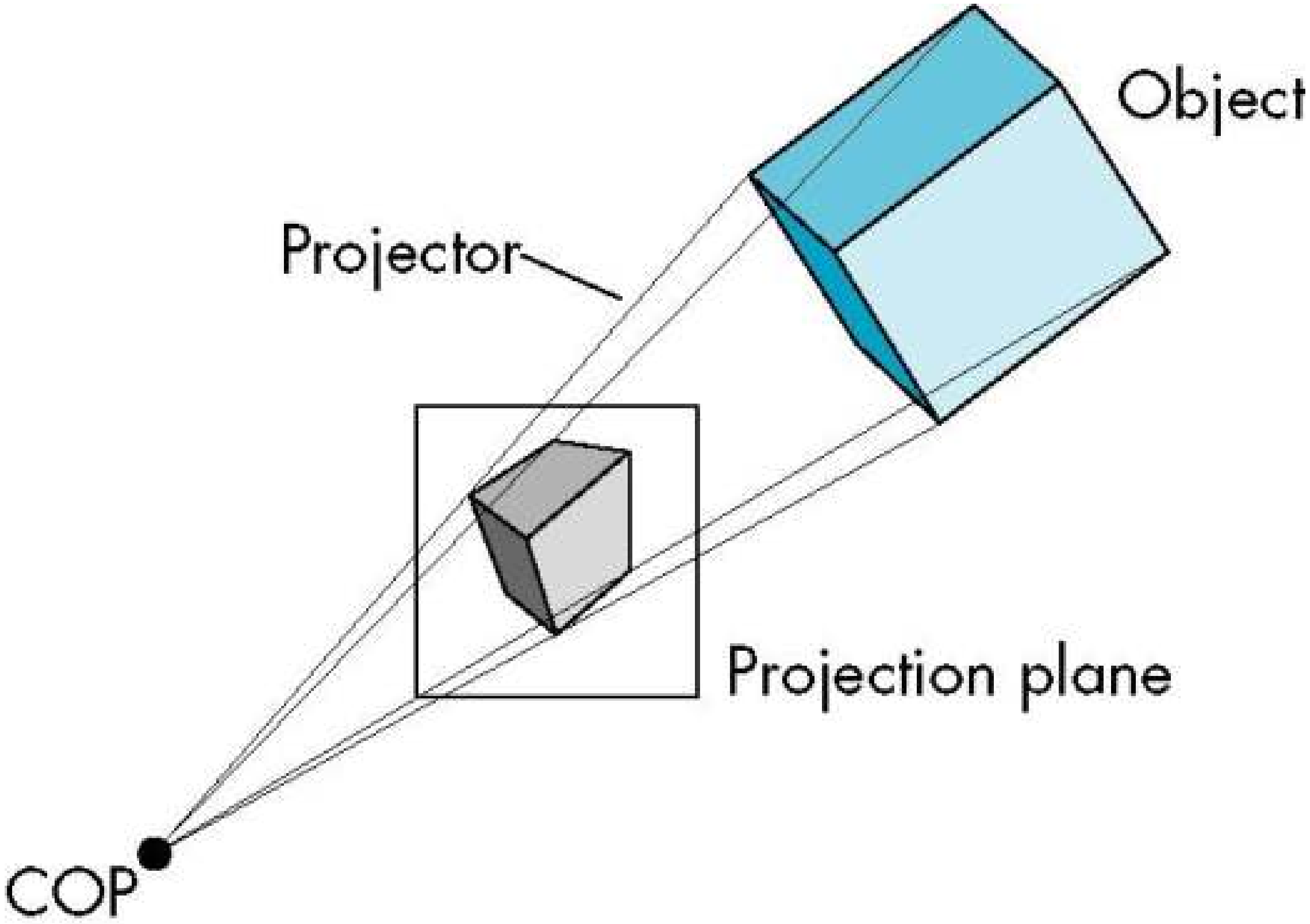


Fig. 9.1 Elements of projection

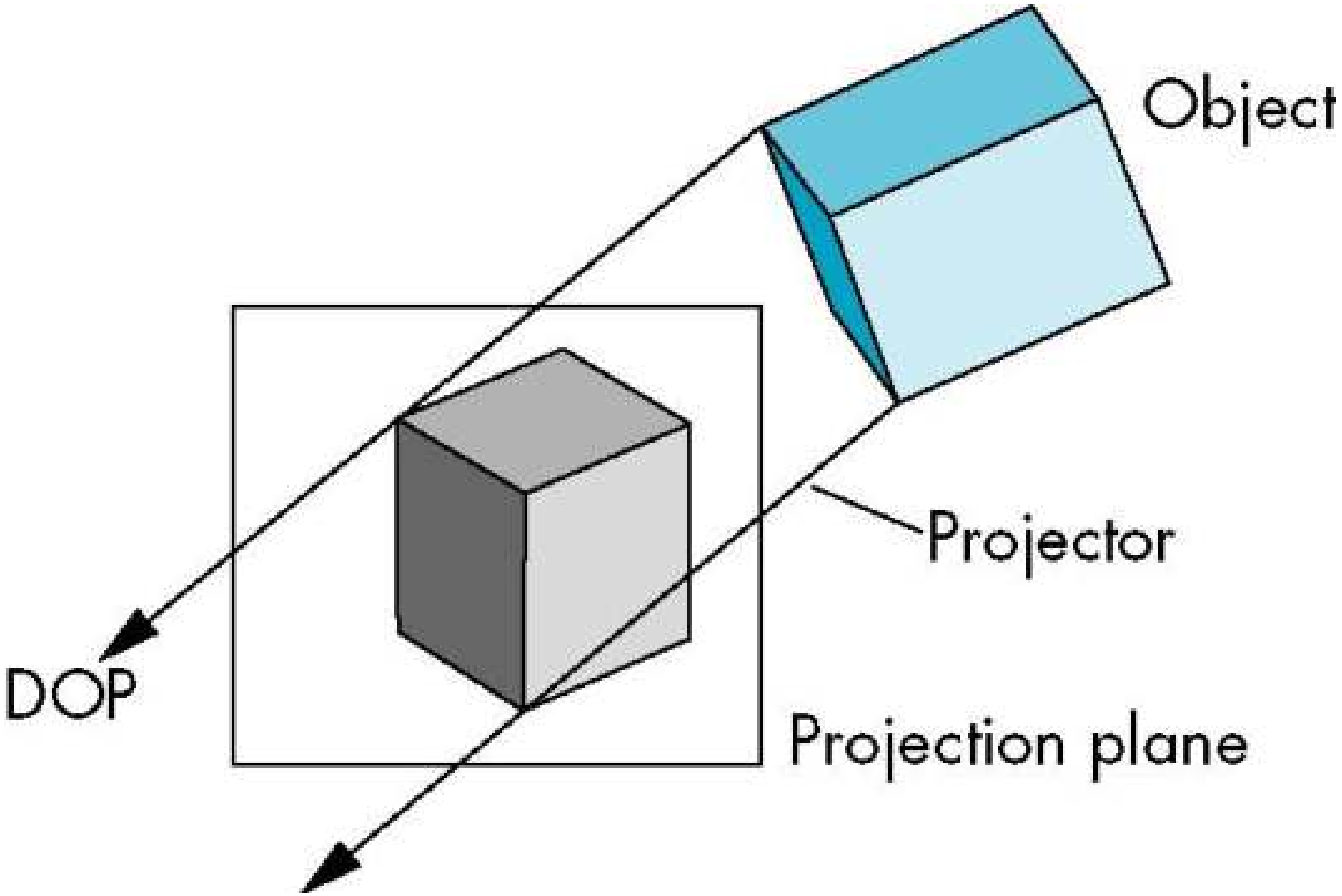
Planar Geometric Projections

- Standard projections project onto a plane
- Projectors are lines that either
  - converge at a center of projection
  - are parallel
- Such projections preserve lines
  - but not necessarily angles
- Nonplanar projections are needed for applications such as map construction

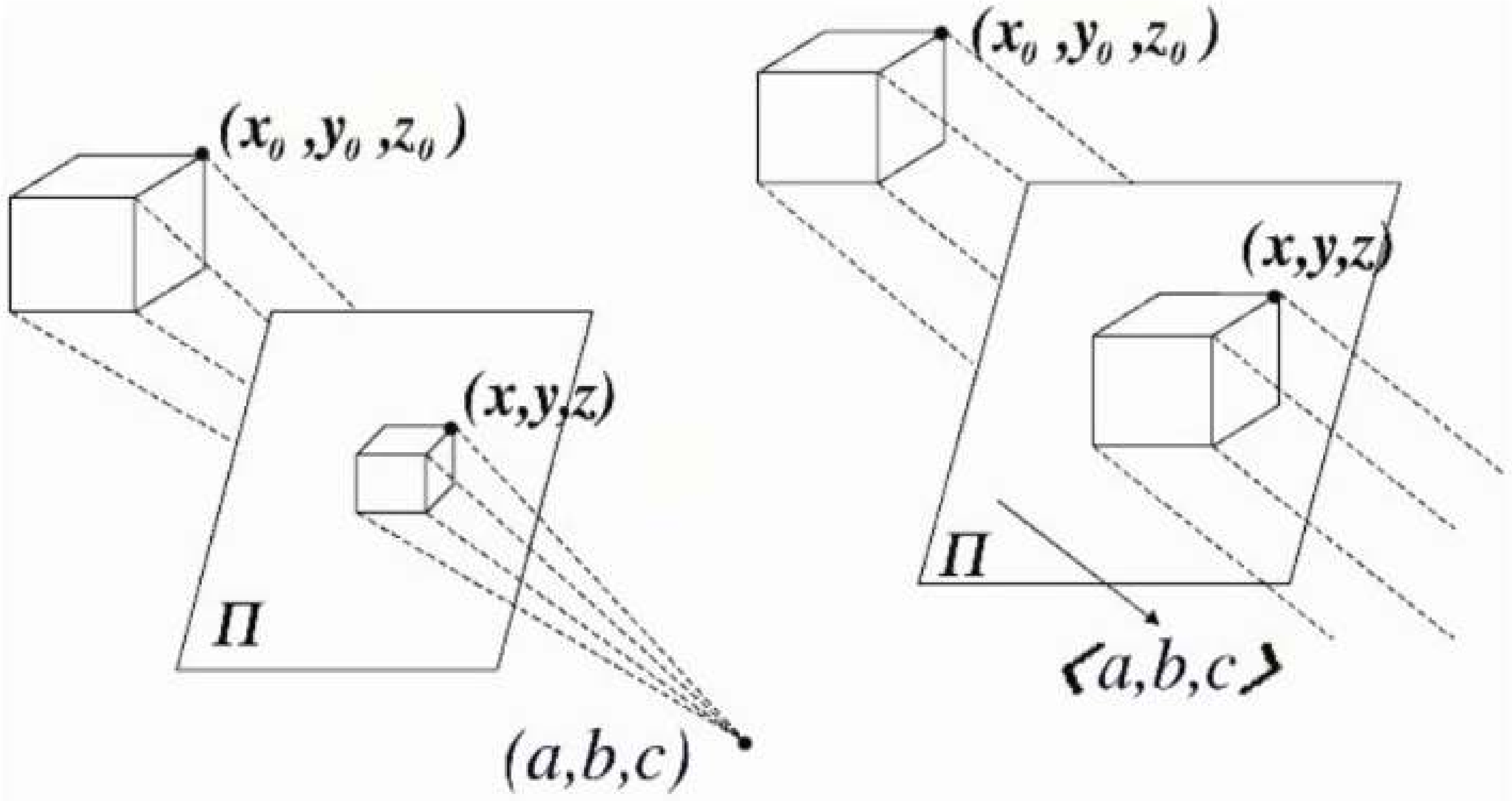
Perspective Projection



Parallel Projection

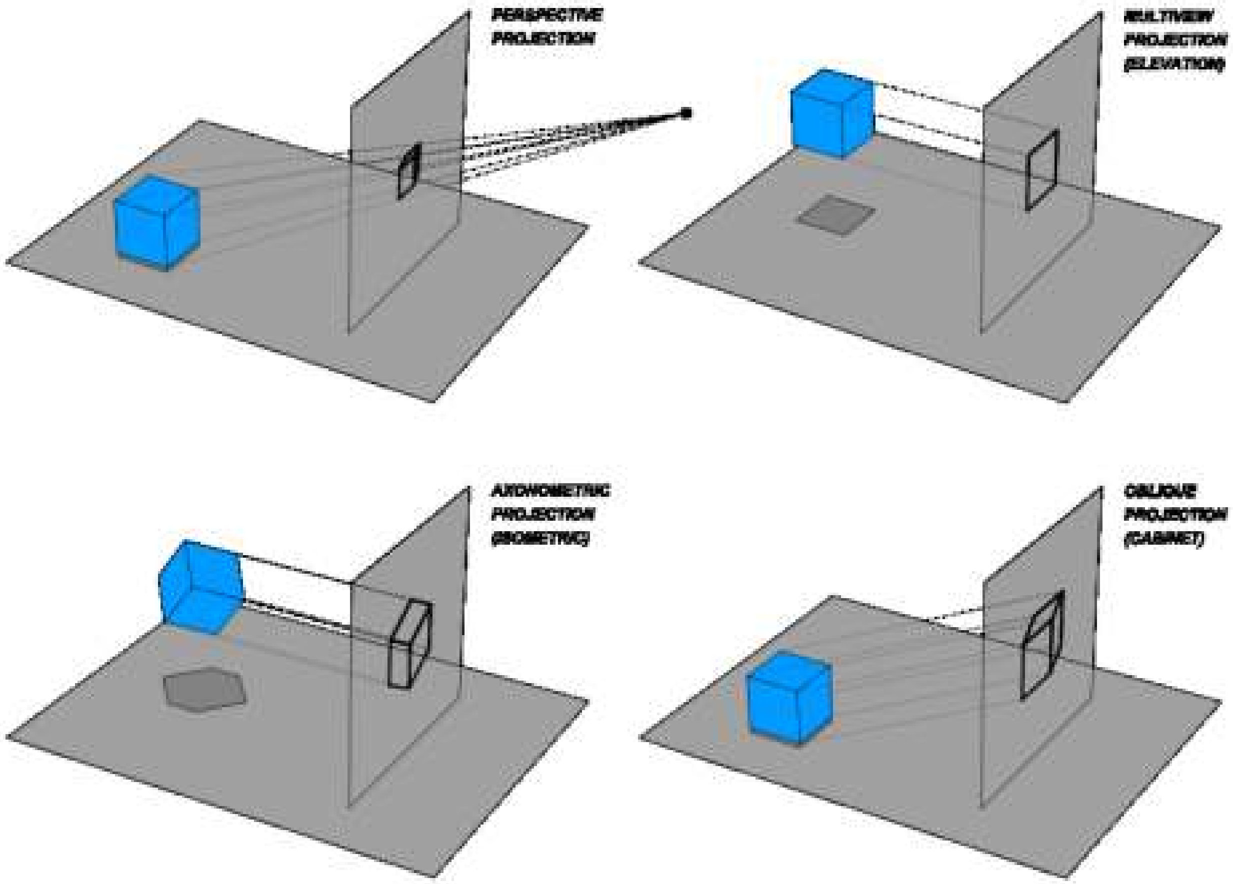


ORTHOGRAPHIC PROJECTION

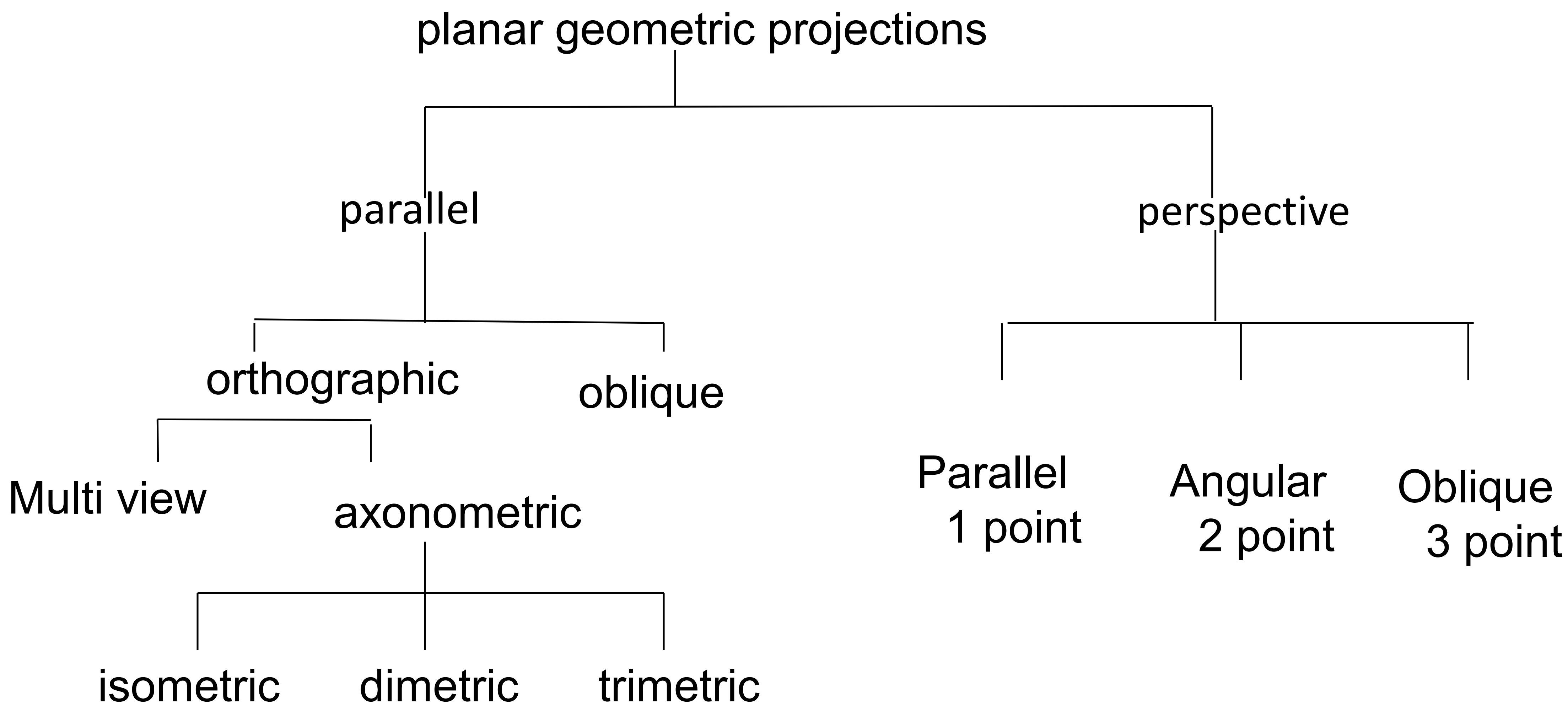


Perspective v's Parallel Projection

ORTHOGRAPHIC PROJECTION

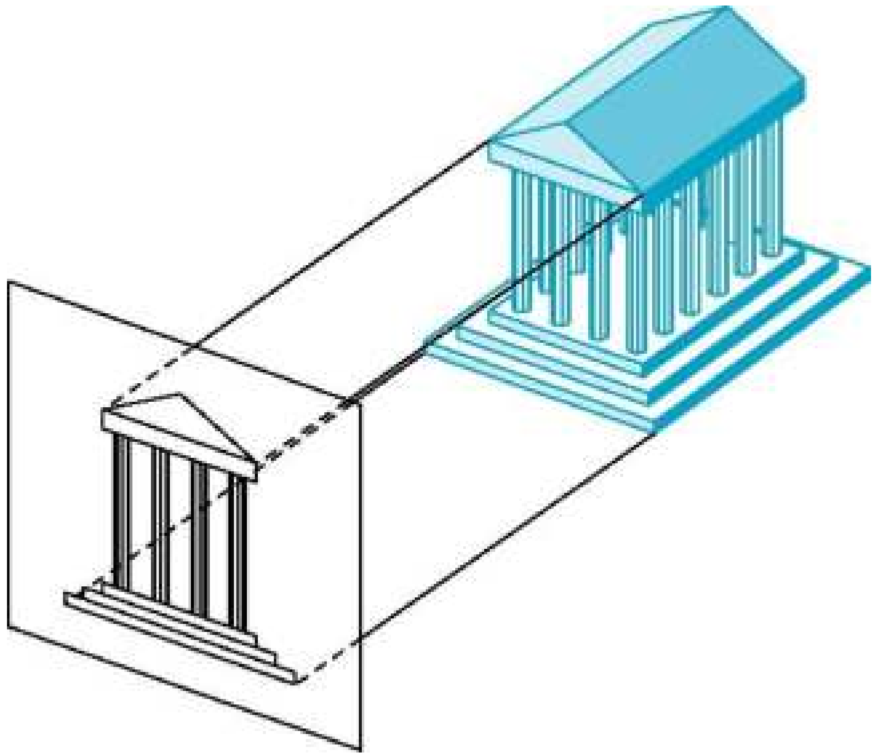


Taxonomy of Planar Geometric Projections



ORTHOGRAPHIC PROJECTION

Projectors are orthogonal to projection plane



Multiview Orthographic Projection

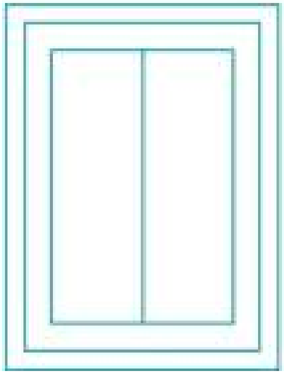
- Projection plane parallel to principal face
- Usually form front, top, side views

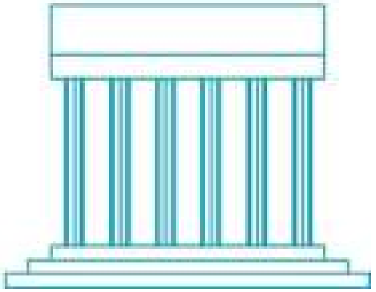
Isometric (not multiview orthographic view)

front

In CAD and architecture, we often display three multiviews plus isometric

top





side

ORTHOGRAPHIC PROJECTION

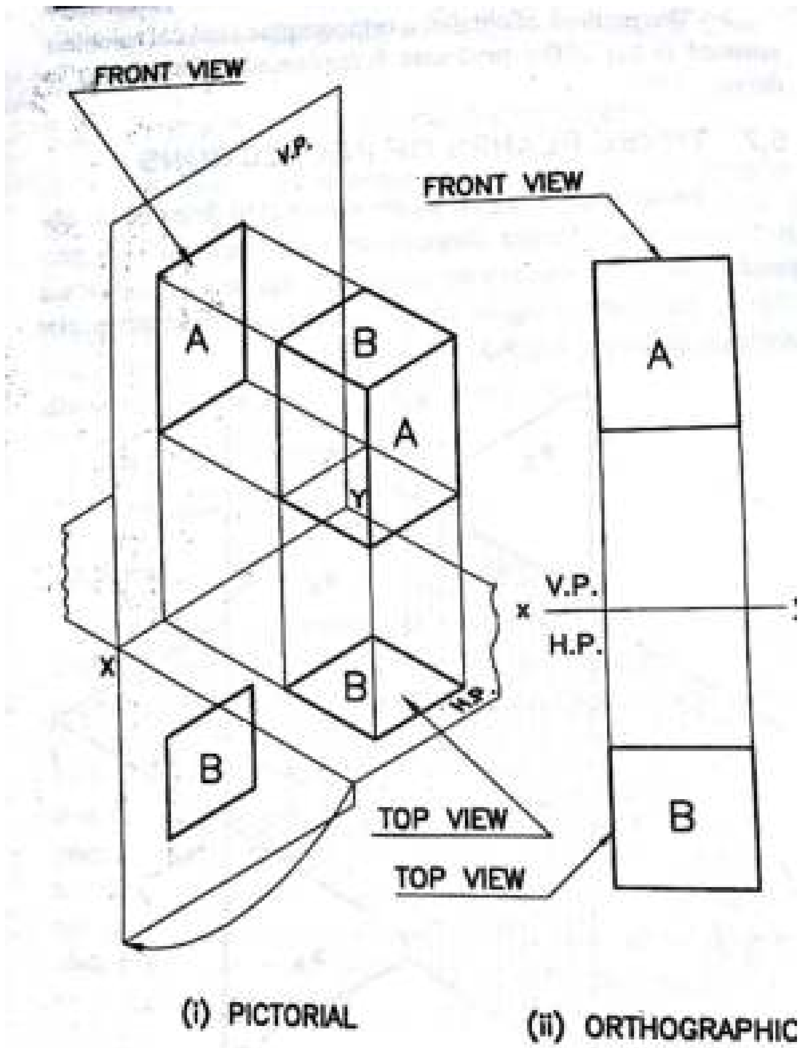
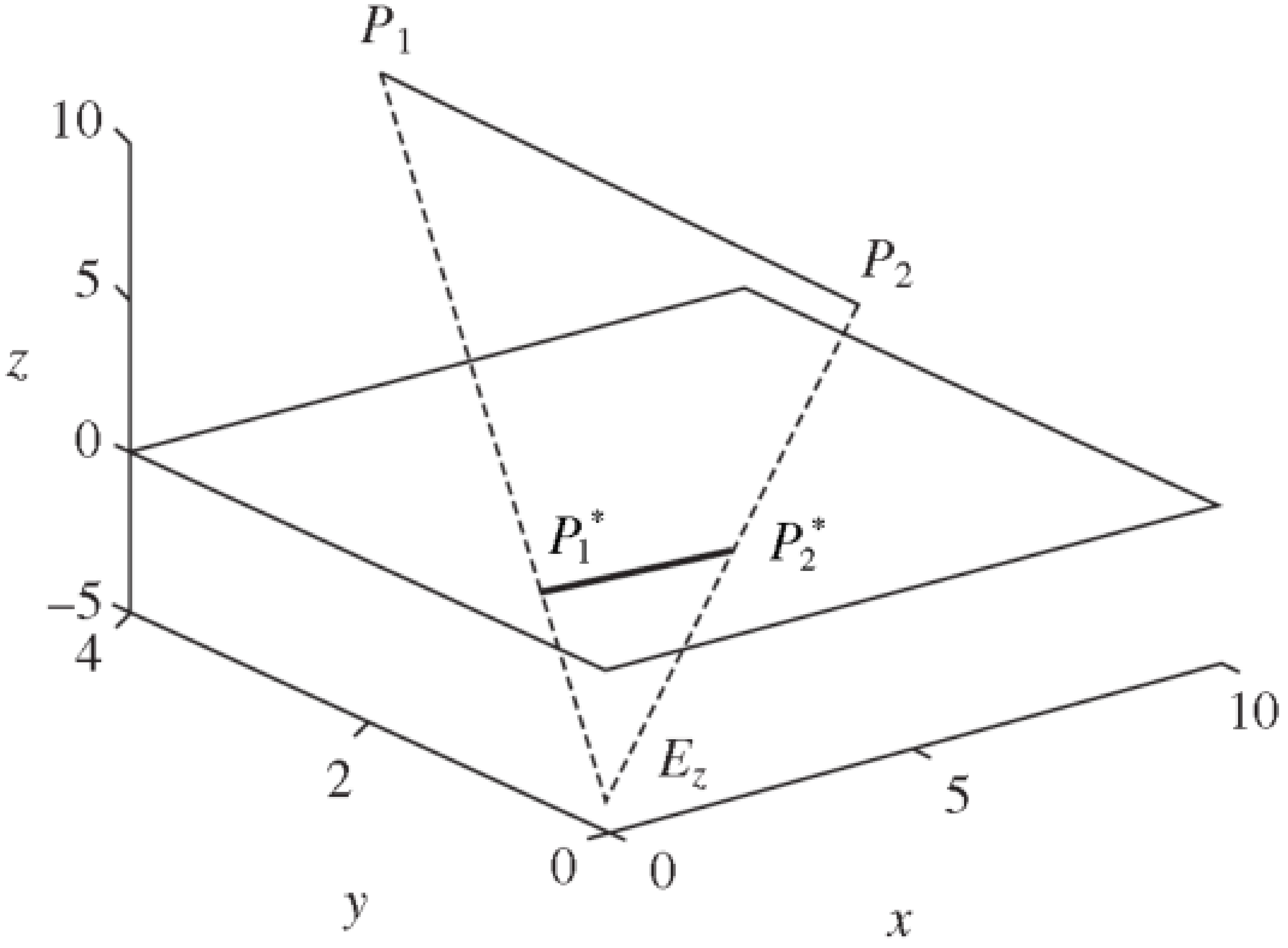


Fig. 9.4 First angle projection of a solid

Advantages and Disadvantages

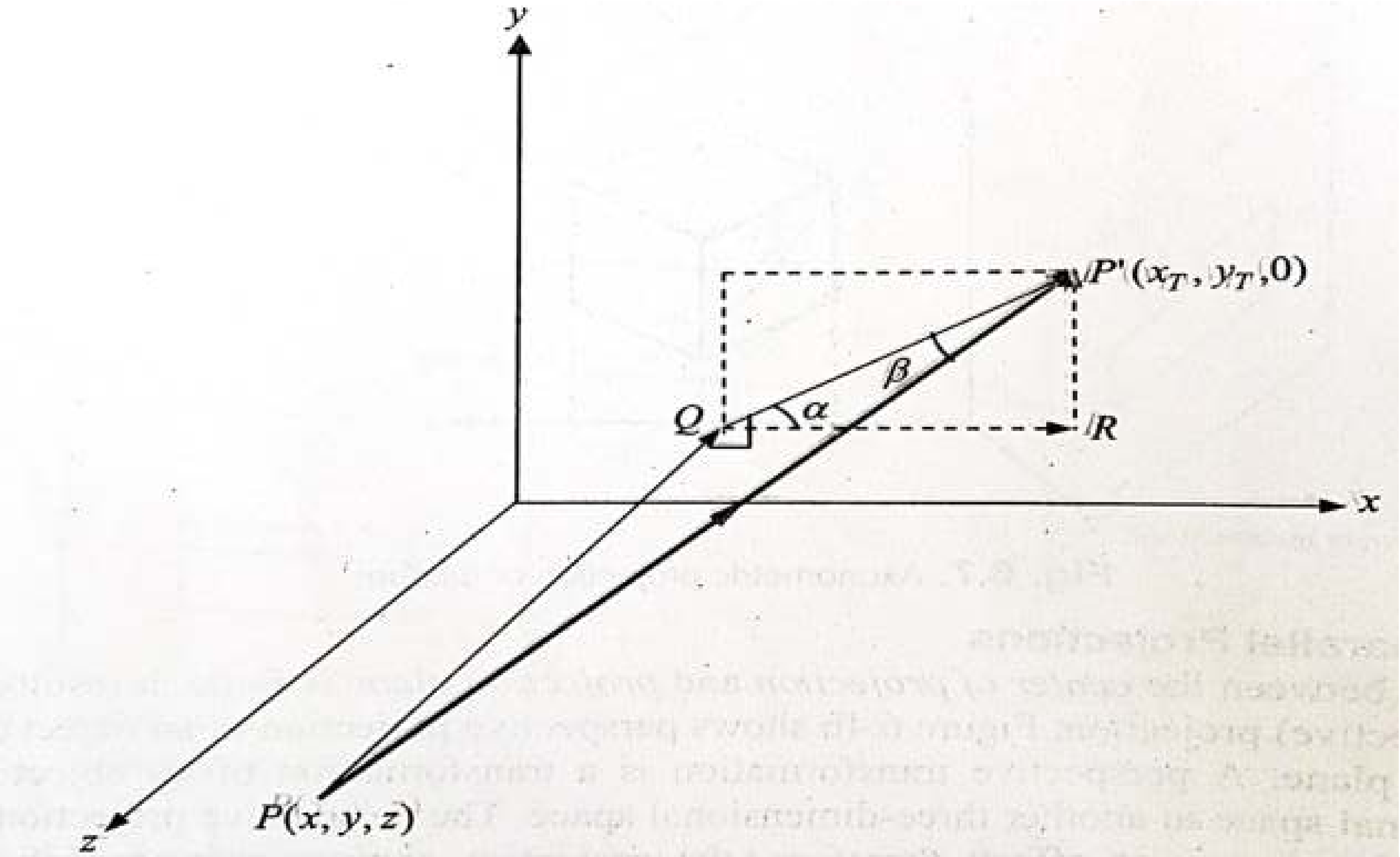
- Preserves both distances and angles
  - Shapes preserved
  - Can be used for measurements
    - Building plans
    - Manuals
- Cannot see what object really looks like because many surfaces hidden from view
  - Often we add the isometric

ORTHOGRAPHIC PROJECTION



ORTHOGRAPHIC PROJECTIONS

P(x,y,z)- point in 3D space P' (x_T,y_T,0) is the projection on x-y plane.



- Parametric equation of a line segment P1 (x1,y1,z1) and P2 (x2,y2,z2) is

  P(t) = P1 + (P2 - P1) t ; t [0,1]

Thus scalar forms:

x = x1+(x2-x1)t = x1+u.t

y = y1+(y2-y1)t = y1+v.t

z = z1+(z2-z1)t = z1+w.t

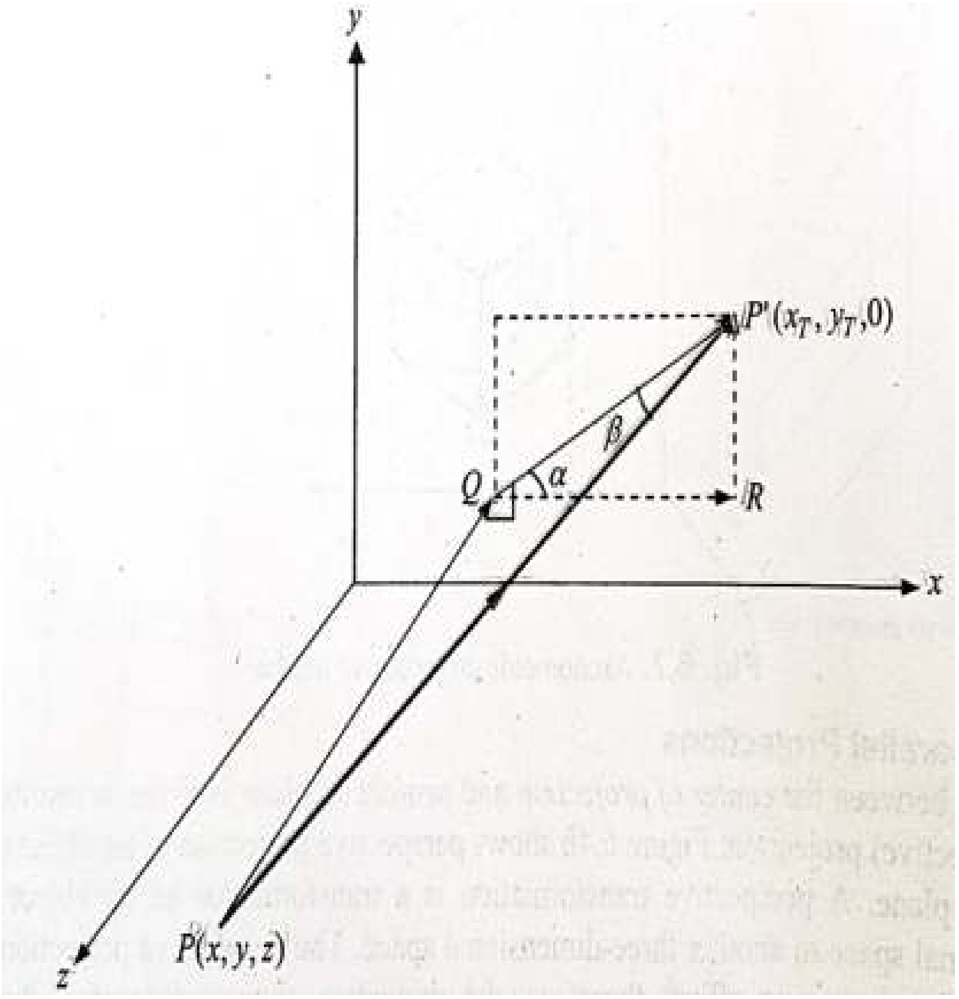
From Fig,

QP’ = PP’ cos β = cos β

u = QP’ cos α = cos β cos α

v = QP’ sin α = cos β sin α

w = QP = -PQ = -PP’ sin β = -sin β



For Orthographic projection on z=0 ie, X-Y plane,

$z=0$ >>>> $t = -z_1/w$

>> $x_T = x_1 + (\cot\beta\cos\alpha).z_1$

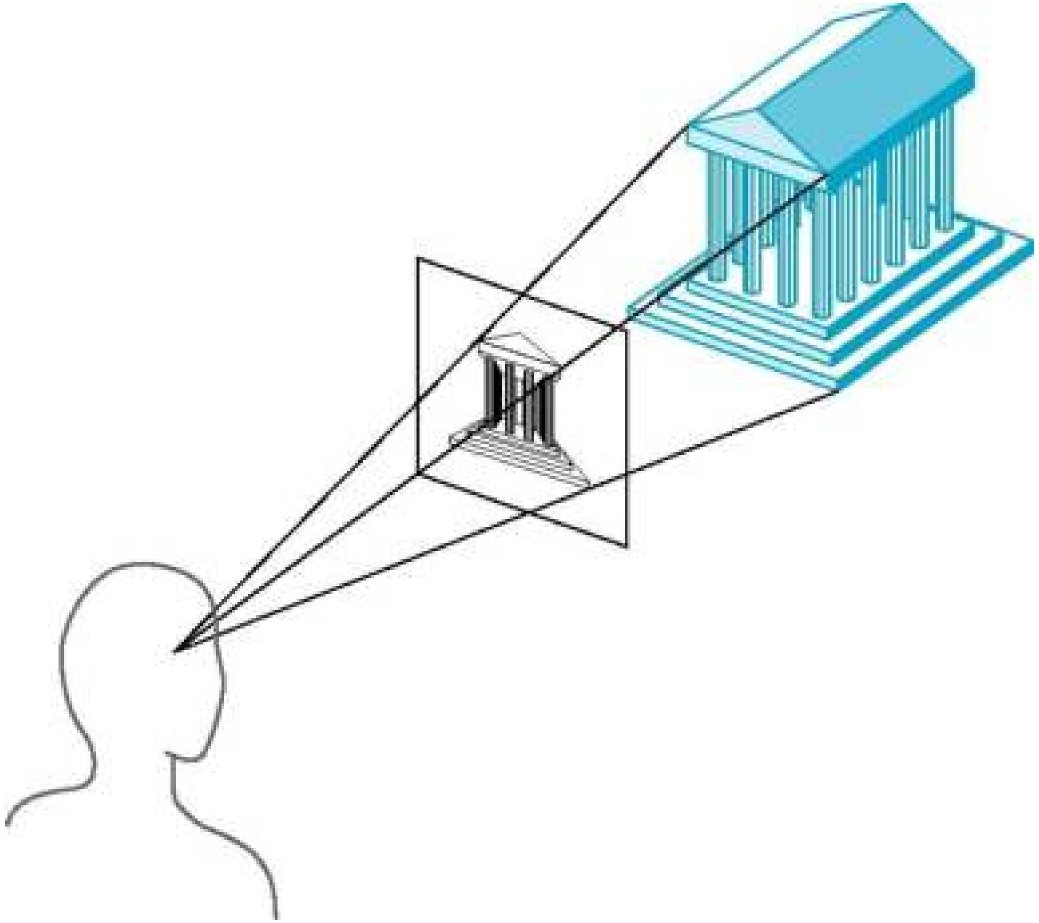
& $y_T = y_1 + (\cot\beta\sin\alpha).z_1$ and $z_T = 0$

- Matrix form: $\begin{pmatrix} x_T \\ y_T \\ z_T \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & \cot\beta\cos\alpha & 0 \\ 0 & 1 & \cot\beta\sin\alpha & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$

- For FRONT VIEW $\beta = 90$; $\alpha = 0$
- For TOP VIEW $\beta = 0$; $\alpha = 0$
- For SIDE VIEW $\beta = 90$; $\alpha = 90$

PERSPECTIVE PROJECTION

Projectors converge at center of projection



PERSPECTIVE PROJECTION

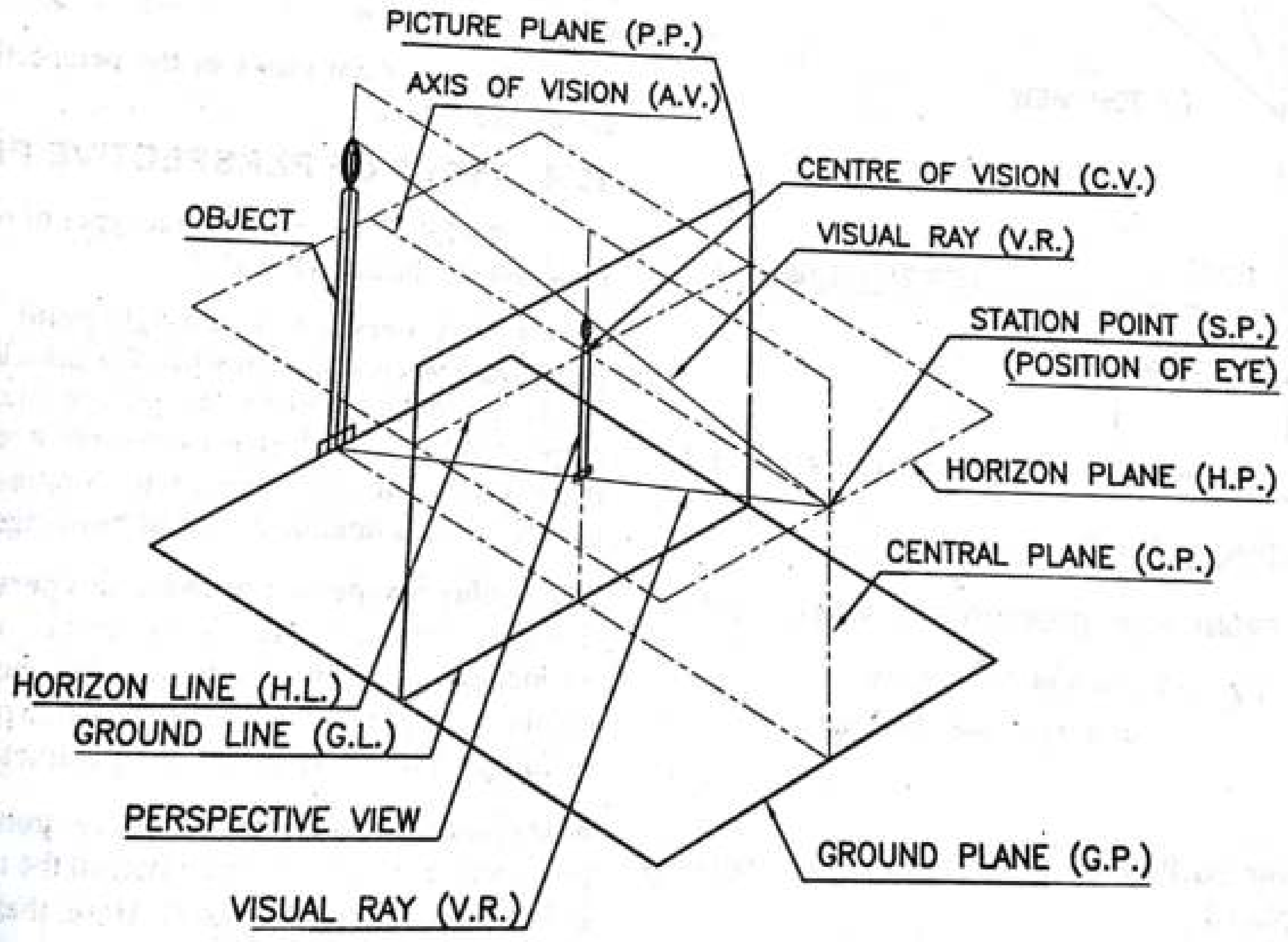
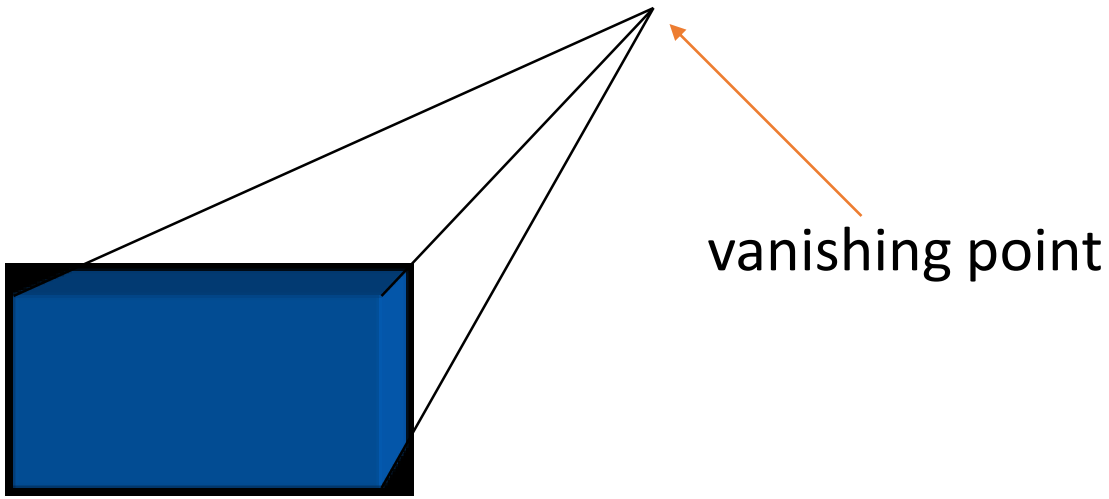


Fig. 19.1 Nomenclature of perspective projection

Vanishing Points

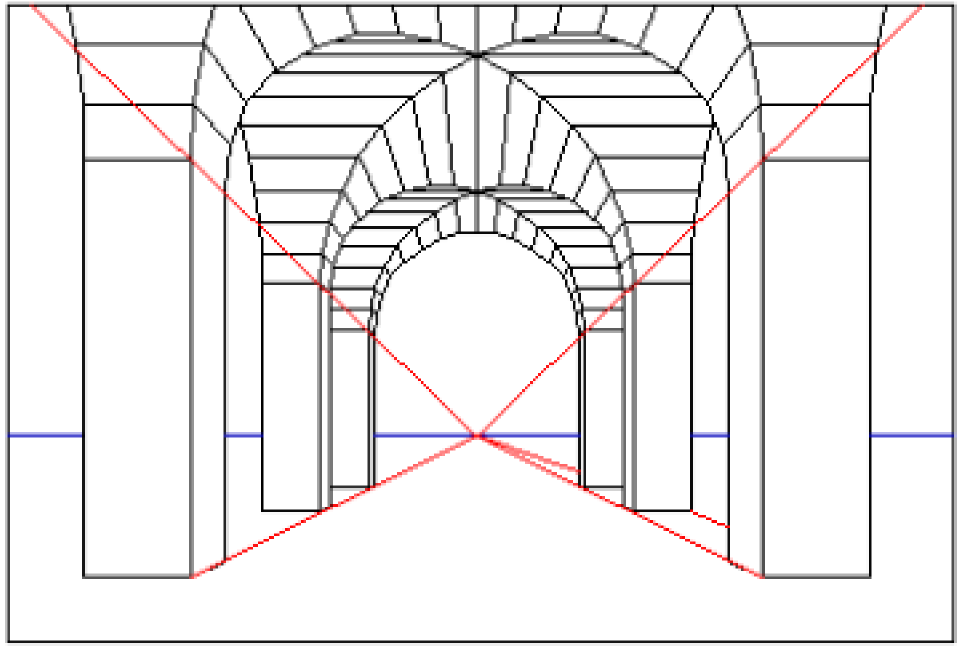
- Parallel lines (not parallel to the projection plan) on the object converge at a single point in the projection (the *vanishing point*)
- Drawing simple perspectives by hand uses these vanishing point(s)



PERSPECTIVE PROJECTION

One-Point Perspective

- One principal face parallel to projection plane
- One vanishing point for cube



PERSPECTIVE PROJECTION

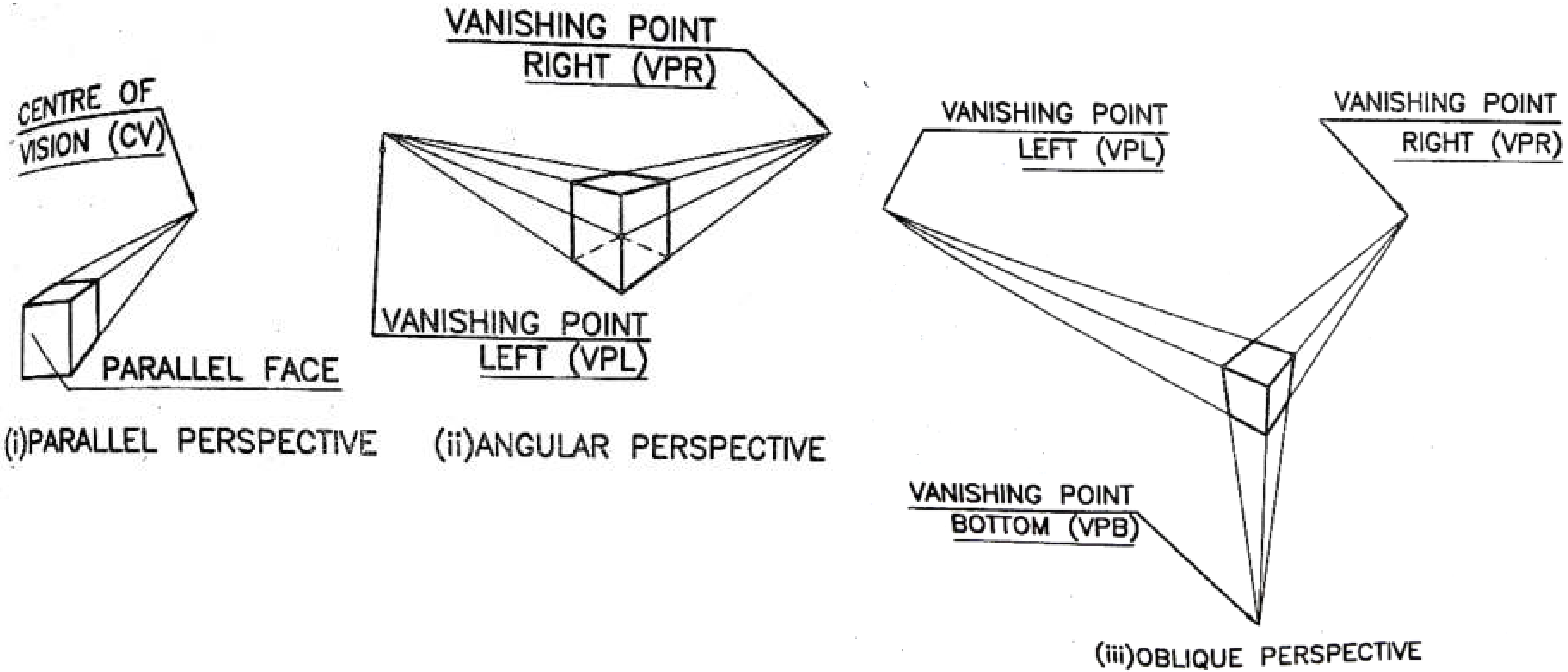
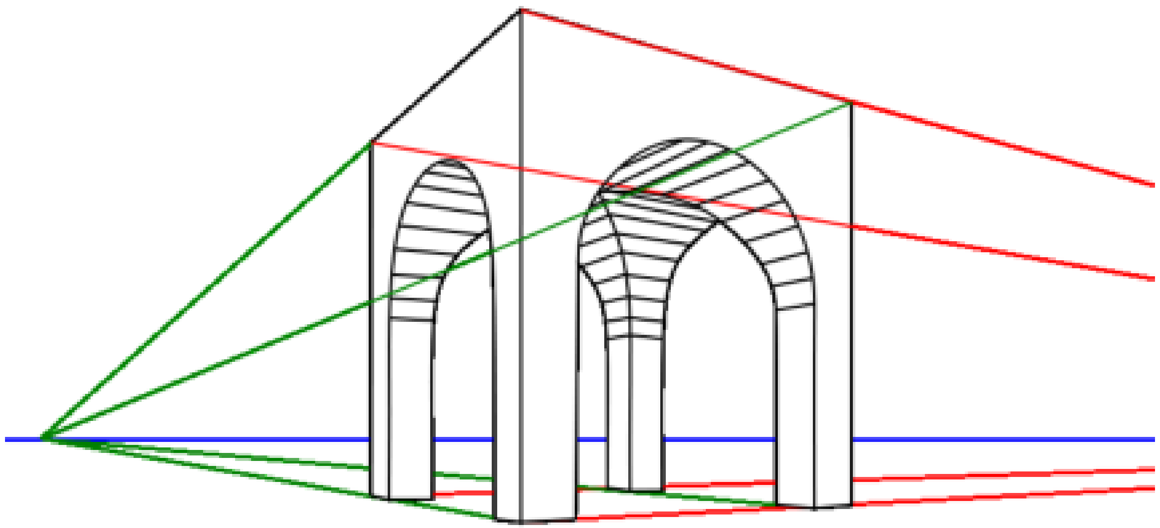


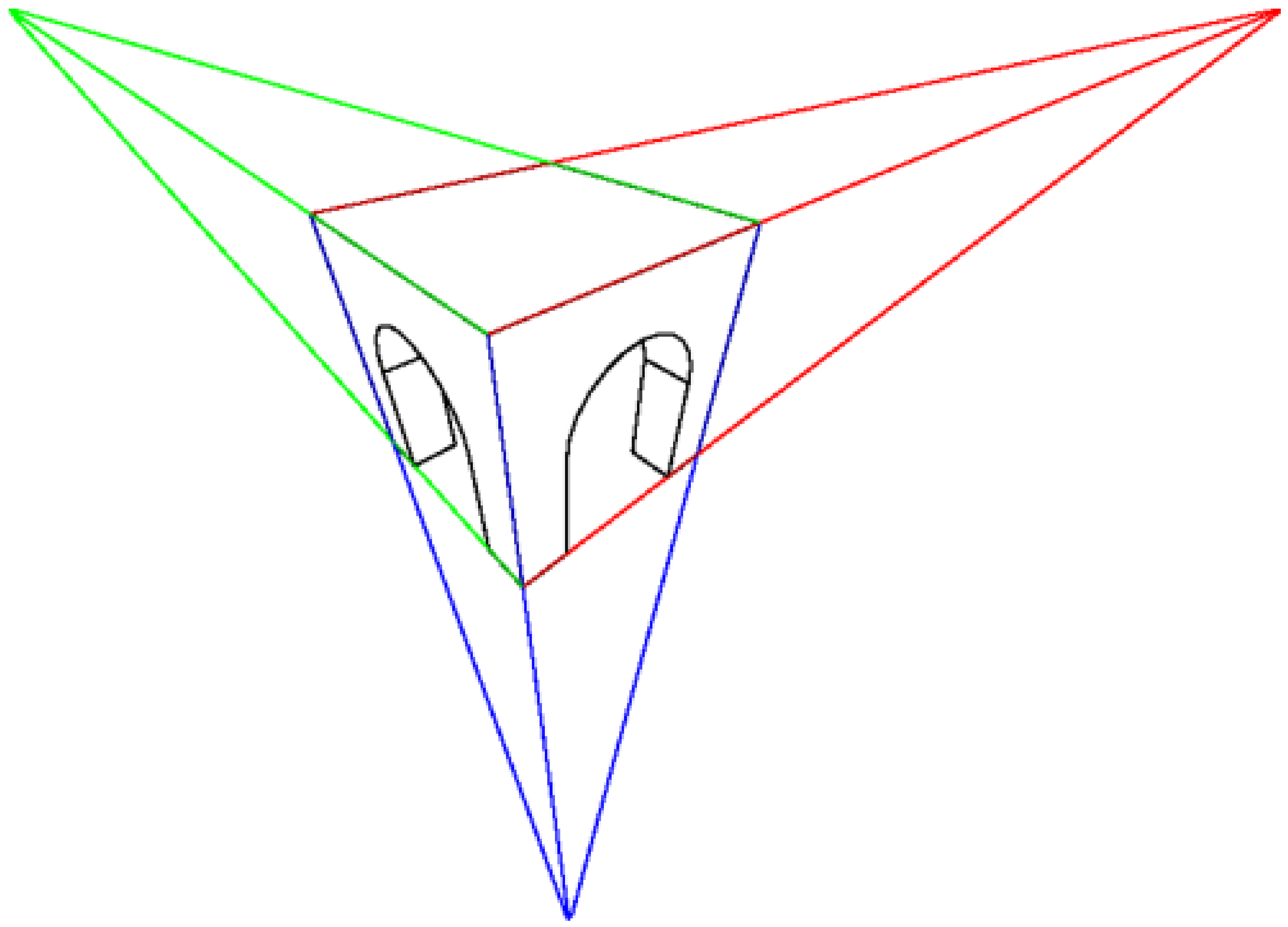
Fig. 19.3 Types of perspective projections

Two-Point Perspective

- On principal direction parallel to projection plane
- Two vanishing points for cube



Three point Perspective



Advantages and Disadvantages

- Objects further from viewer are projected smaller than the same sized objects closer to the viewer (*diminution*)
  - Looks realistic
- Equal distances along a line are not projected into equal distances (*nonuniform foreshortening*)
- Angles preserved only in planes parallel to the projection plane
- More difficult to construct by hand than parallel projections (but not more difficult by computer)

- Consider a point P (x,y,z) -E as the observer's eye.
- Image plane: *x-y* plane
- line *EP* intersect image plane at ***P* (x*, y*, 0).***

P*B = x* and P*D = y*

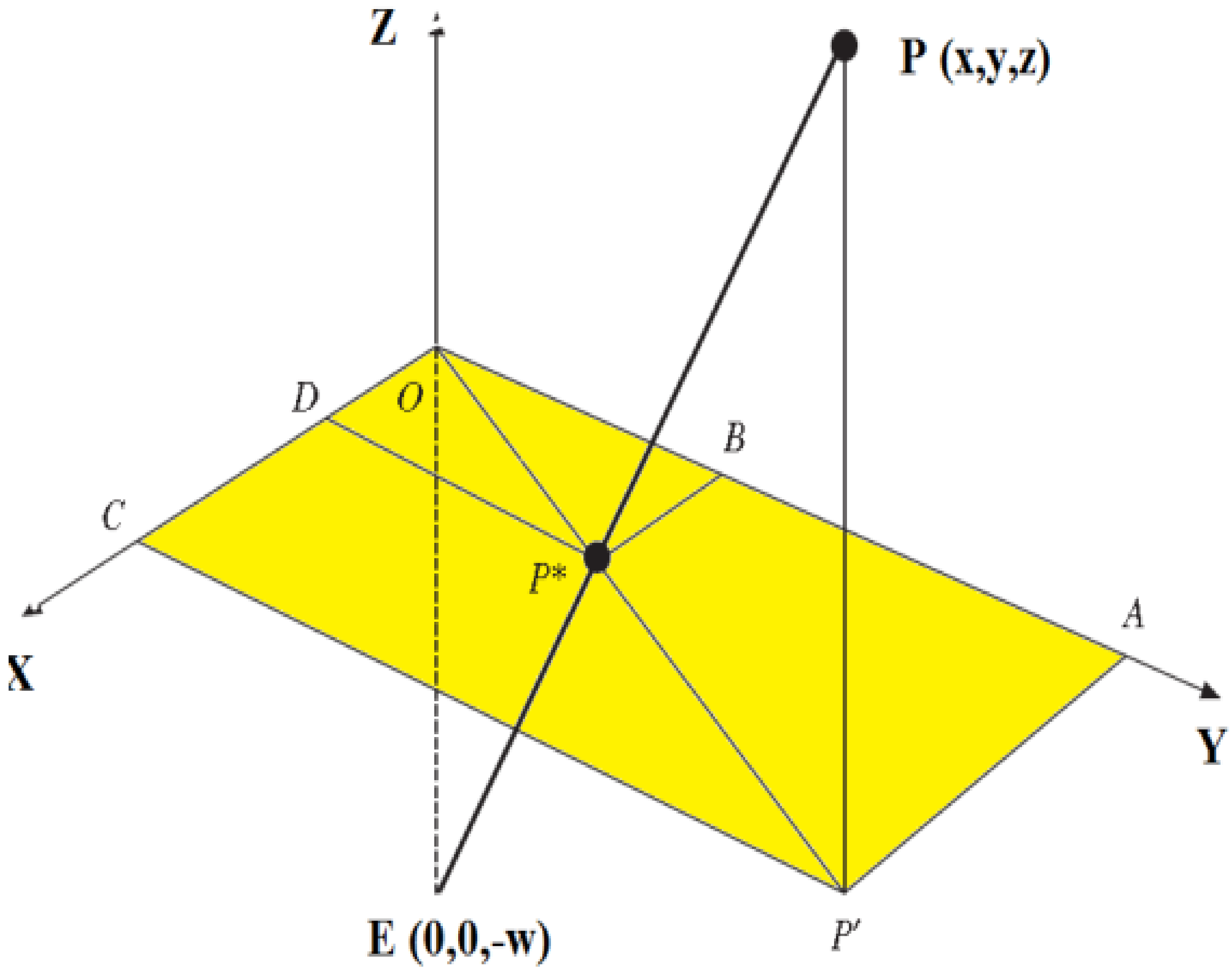
P'C = y and P'A = x

similar $\Delta P^*OE = \Delta P^*P'P$

$$\frac{|OE|}{|PP'|} = \frac{|OP^*|}{|P^*P'|} = \frac{|EP^*|}{|P^*P|}$$

$$|EP^*| = \left(\frac{w}{z}\right)|P^*P|$$

$$EP^* = \left(\frac{w}{z}\right)P^*P$$



Perspective projection of *P* on the *x-y* plane

ONE POINT PERSPECTIVE PROJECTION

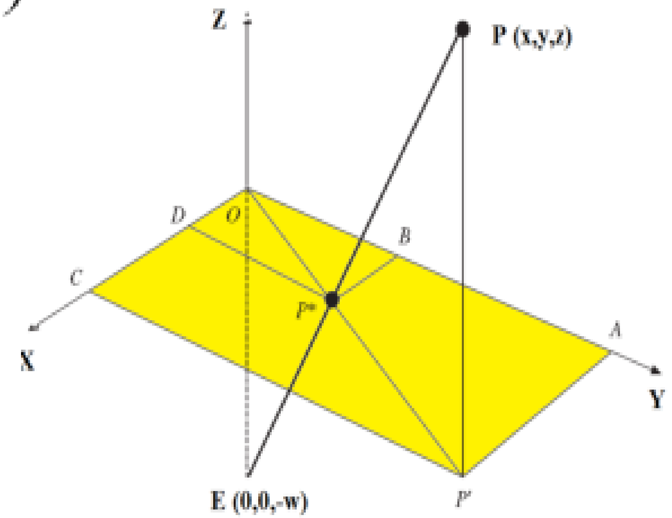
$$\mathbf{OP}^* = x^*\mathbf{i} + y^*\mathbf{j} + z^*\mathbf{k} = \mathbf{OE} + \mathbf{EP}^*$$

$$= -w\mathbf{k} + \left(\frac{w}{z}\right)\mathbf{P}^*\mathbf{P} = -w\mathbf{k} + \left(\frac{w}{z}\right)(x - x^*)\mathbf{i} + \left(\frac{w}{z}\right)(y - y^*)\mathbf{j} + \left(\frac{w}{z}\right)(z - z^*)\mathbf{k}$$

$$\text{Thus, } x^* = \frac{w}{z}(x - x^*),$$

$$\boxed{x^* = \frac{wx}{z+w}, \quad y^* = \frac{wy}{z+w}, \quad \text{and } z^* = 0}$$

$$P^* = \left[\frac{wx}{z+w}, \frac{wy}{z+w}, 0, 1\right]$$



Perspective projection of P on the x-y plane

$$\mathbf{P}^* = \begin{bmatrix} x^* \\ y^* \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{wx}{z+w} \\ \frac{wy}{z+w} \\ 0 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} x \\ y \\ 0 \\ \frac{z}{w} + 1 \end{bmatrix} = \mathbf{P}_{ers}\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{w} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- We can develop similar perspective projection matrices for the human eye to be on the *x* and *y* axis, respectively, using cyclic symmetry.

- **For the view point *Ex* at *x* = – *w* on the *x*-axis**, a line joining *Ex* and *P* will intersect the *y*-*z* image plane at

$$P^*_{yz}\begin{bmatrix} 0 & \dfrac{wy}{x+w} & \dfrac{wz}{x+w} & 1 \end{bmatrix}$$

- **For view point is shifted to *Ey* at *y* = –*w* on *y*-axis,** the line joining *Ey* and *P* will intersect *z*-*x* image plane at

$$P^*_{zx}\begin{bmatrix} \dfrac{wx}{y+w} & 0 & \dfrac{wz}{y+w} & 1 \end{bmatrix}$$

Question ????

A line P_1 P_2 has coordinates $P_1(4, 4, 10)$ and $P_2(8, 2, 4)$ and the observer's eye E_z is located at $(0, 0, -4)$. Find the perspective projection of the line on the x-y plane.

ANSWER: (8/7,8/7,0) and (4,1,0)

Module 3

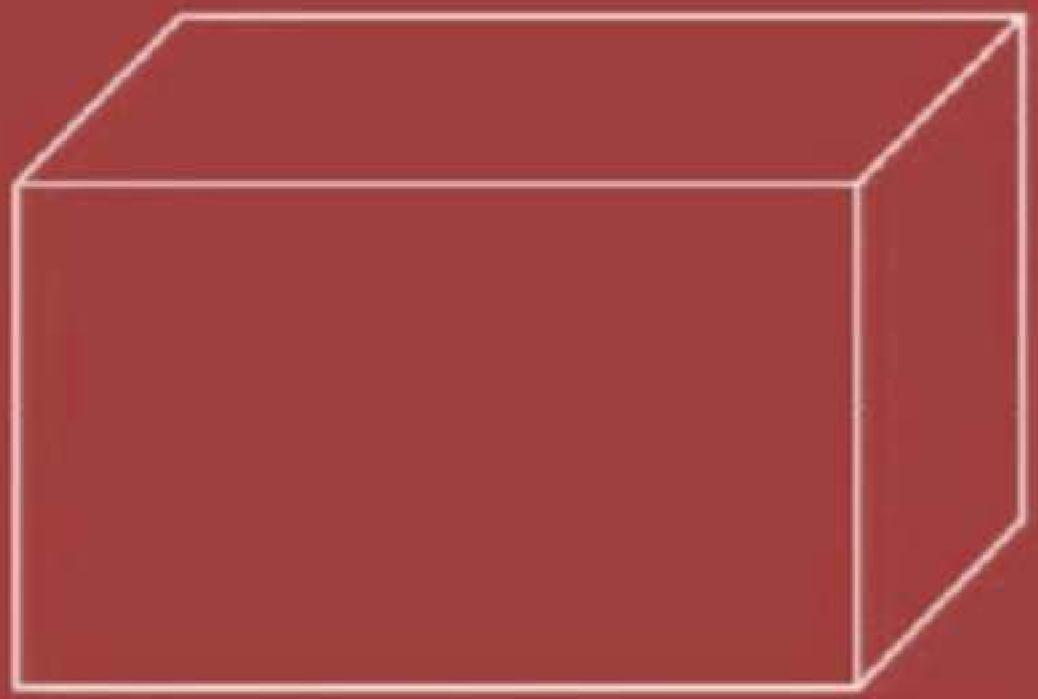
PARAMETRIC REPRESENTATION OF CURVES AND SURFACES

GEOMTRIC MODELING

Geometric modeling refers to computer compatible and mathematical representation of geometry.

- Only mathematical representation of geometry is not enough.
- Only visual representation of geometry is not enough.

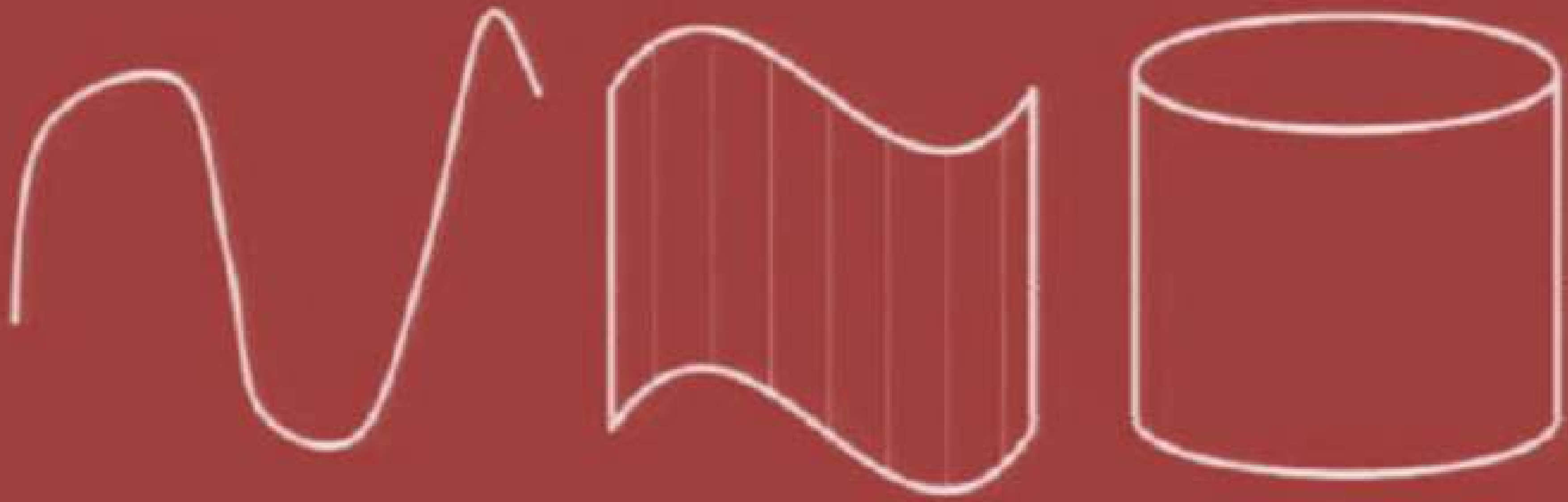
GEOMTRIC MODELING



The above picture is 2D or 3D ?

GEOMTRIC MODELING

Geometric modeling is concerned with representation of:



GEOMETRIC MODELING OF CURVES & SURFACES

Implicit Representation	Explicit Representation	Parametric Representation
$x^2 + y^2 - r^2 = 0$	$y = (r^2 - x^2)^{1/2}$	$x = r\cos(t)$ $y = r\sin(t)$
$ax + by + cz + d = 0$	$z = px + qy + r$	$x = a + bu + cw$ $y = d + eu + fw$ $z = g + hu + iw$

PARAMETRIC REPRESENTATION

- Parametric equations completely separate the roles dependent and independent variables.

What do these equations represent ?

$$x = r \cos(t)$$
$$y = r \sin(t)$$
$$z = h$$

$$r = 5;\ t = \pi\ ;\ z = 20;$$

PARAMETRIC REPRESENTATION

- Parametric equations completely separate the roles dependent and independent variables.

What do these equations represent ?

$$x = r\cos(t)$$
$$y = r\sin(t)$$
$$z = h$$

$$r = 5;\ -\pi \le t \le \pi;\ 0 \le z \le 20;$$

PARAMETRIC REPRESENTATION

- Parametric equations completely separate the roles dependent and independent variables.

What do these equations represent ?

$$x = r\cos(t)$$
$$y = r\sin(t)$$
$$z = h$$

$0 \leq r \leq 5$; $-\pi \leq t \leq \pi$; $0 \leq z \leq 20$;

PARAMETRIC REPRESENTATION

- Offers more degree of freedom for controlling the shape of curves & surfaces

Explicit form

$$y = p\,x^3 + q\,x^2 + r\,x + s$$

Implicit form

$$x = a\,u^3 + b\,u^2 + c\,u + d$$

$$y = e\,u^3 + f\,u^2 + g\,u + h$$

PARAMETRIC REPRESENTATION

- Transformations are easier to apply

Circle with center (0, 0) and radius 7 units	Circle with center (4, 3) and radius 7 units
$x = 7\cos(t)$ $y = 7\sin(t)$	$x = 4 + 7\cos(t)$ $y = 3 + 7\sin(t)$
$x^2 + y^2 - 49 = 0$	$x^2 + y^2 - 8x - 6y - 24 = 0$

PARAMETRIC REPRESENTATION

- Has an advantage in representation of curve and surface segments.

CIRCLE

$x = r\cos(t)$

$y = r\sin(t)$

$r = 8;\ -\pi \le t \le \pi;$

CIRCULAR ARC

$x = r\cos(t)$

$y = r\sin(t)$

$r = 8;\ -\pi \le t \le 0;$

PARAMETRIC REPRESENTATION

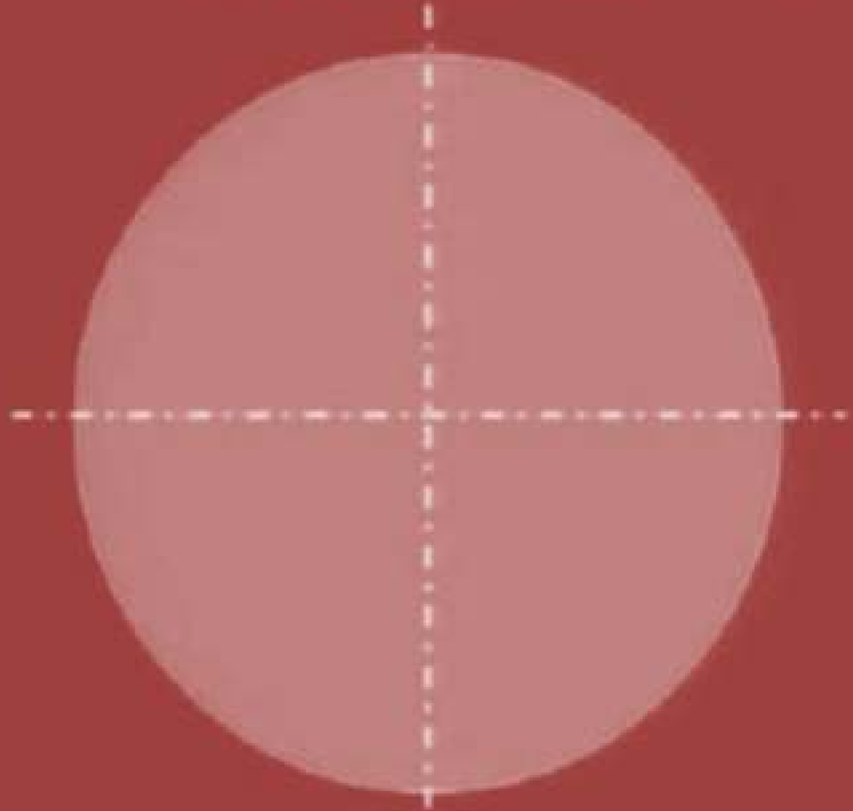
- Has an advantage in handling infinite slopes

Implicit/Explicit $dy / dx = \infty$

$$\text{Parametric} = \frac{dy / du}{dx / du} = \infty \quad \text{implies } dx / du = 0$$

PARAMETRIC REPRESENTATION

- Has an advantage in calculation of points for display and tool path.



$x^2 + y^2 - 64 = 0$

(Implicit)

$x = 8 \cos (t)$

$y = 8 \sin (t)$

(Parametric)

How to approximate circle with a line

PARAMETRIC CURVES

Curves can be classified in number of ways:

- Plane curves & Space curves

  Example: Circle & Helix

- Curves of known forms & Free-form curves

  Example: Circle vs. Bezier Curve

- Interpolation curves and Approximation curves

  Example: Hermite Curve vs. Bezier Curve

PARAMETRIC REPRESENTATION OF LINE

Line in xy-plane

$x = x1 + (x2 - x1)\,u$

$y = y1 + (y2 - y1)\,u$

$0 \leq u \leq 1.0$

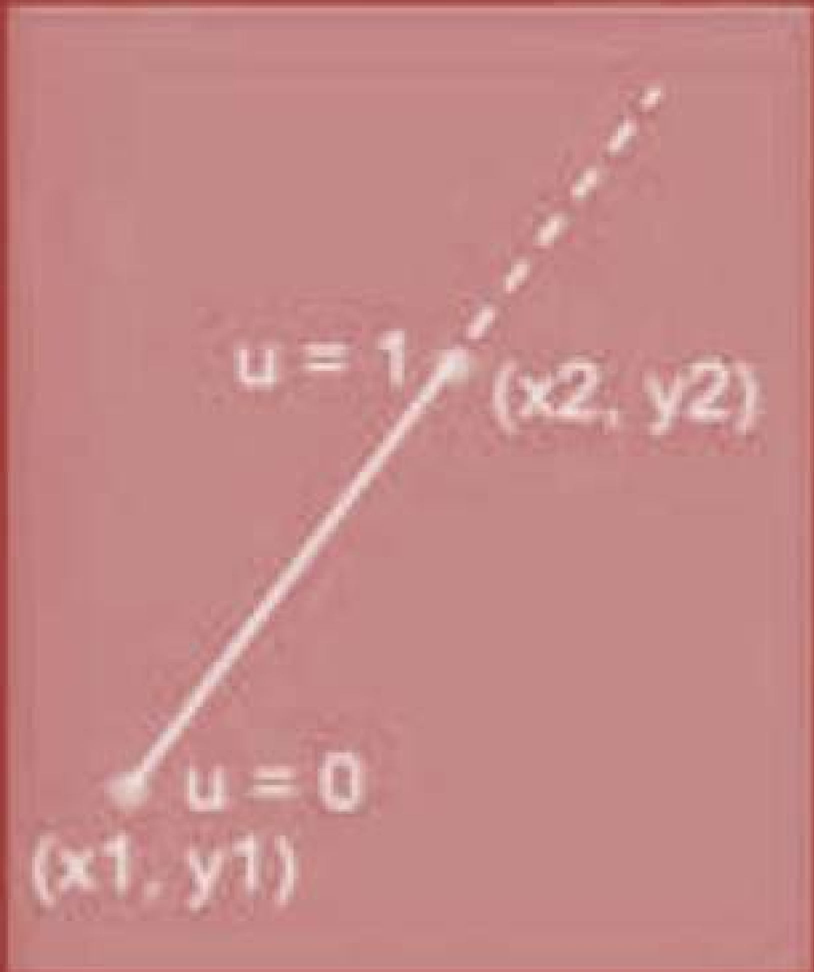
Line in space

$x = x1 + (x2 - x1)\,u$

$y = y1 + (y2 - y1)\,u$

$z = z1 + (z2 - z1)\,u$

$0 \leq u \leq 1.0$



PARAMETRIC REPRESENTATION OF CIRCLE

Circle in xy-plane

$x = x1 + r\cos(\theta)$

$y = y1 + r\sin(\theta)$

$0 \leq \theta \leq 2\pi$

Circle in xy-plane

$x = x1 + r\cos(2\pi u)$

$y = y1 + r\sin(2\pi u)$

$0 \leq u \leq 1$

PARAMETRIC REPRESENTATION OF CIRCLE

Circular Arc

$x = x1 + r\cos(\theta)$

$y = y1 + r\sin(\theta)$

$-\pi/2 \leq \theta \leq 0$

Circular Helix

$x = x1 + r\cos(2\pi u)$

$y = y1 + r\sin(2\pi u)$

$z = h u$

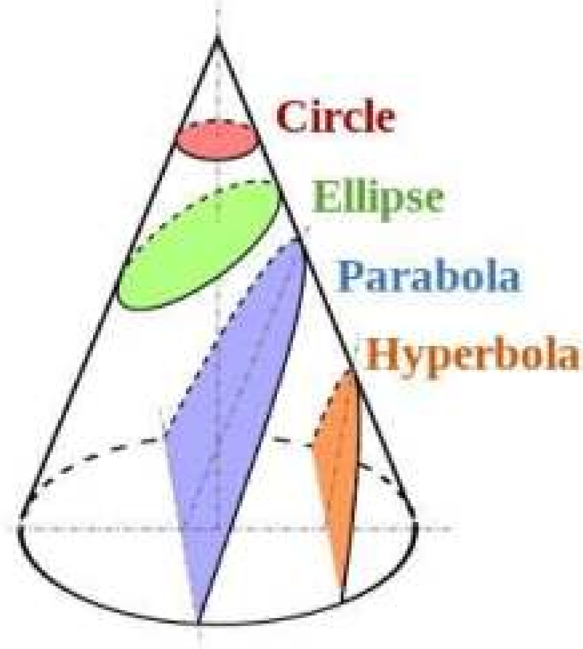
$0 \leq u \leq 1$

Conics in polar coordinates

$$r(\theta) = \frac{\ell}{1 + e\cos\theta}$$

e is called the eccentricity

ℓ is the semi-latus rectum



Relation with implicit form

conic section	equation	eccentricity (e)	linear eccentricity (c)	semi-latus rectum (ℓ)	focal parameter (p)
circle	$x^2 + y^2 = a^2$	0	0	a	∞
ellipse	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$	$\sqrt{1 - \frac{b^2}{a^2}}$	$\sqrt{a^2 - b^2}$	$\frac{b^2}{a}$	$\frac{b^2}{\sqrt{a^2 - b^2}}$
parabola	$y^2 = 4ax$	1	a	$2a$	$2a$
hyperbola	$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$	$\sqrt{1 + \frac{b^2}{a^2}}$	$\sqrt{a^2 + b^2}$	$\frac{b^2}{a}$	$\frac{b^2}{\sqrt{a^2 + b^2}}$

PARAMETRIC REPRESENTATION OF CONICS

Ellipse

$x = a\cos(\theta);\ y = b\sin(\theta)$ $\qquad 0 \le \theta \le 2\pi$

Parabola

$x = a u^2;\ y = 2 a u$ $\qquad 0 \le u \le 1$

Hyperbola

$x = a\sec(\theta);\ y = b\tan(\theta)$ $\qquad 0 \le \theta \le 2\pi$

$x = a\cosh(\theta);\ y = b\sinh(\theta)$

PARAMETRIC REPRESENTATION OF FREE-FORM CURVES

Interpolation Curves

- Parametric Cubic Curve

Approximation Curves

- Bezier Curve
- B-Spline Curve
- NURBS Curve

PARAMETRIC CUBIC CURVE

- It is also known as Hermite Curve
- It is an Interpolation Curve
- It has three different forms

Algebraic Form (12 algebraic coefficients)
Geometric Form (End points & tangent vectors)
Four - Point Form (Four points)

PARAMETRIC CUBIC CURVE

Algebraic Form

$$\left.\begin{aligned} x &= a_{3x}\, u^3 + a_{2x}\, u^2 + a_{1x}\, u + a_{0x} \\ y &= a_{3y}\, u^3 + a_{2y}\, u^2 + a_{1y}\, u + a_{0y} \\ z &= a_{3z}\, u^3 + a_{2z}\, u^2 + a_{1z}\, u + a_{0z} \end{aligned}\right\} \text{12 unknowns}$$

$$0 \le u \le 1$$

$$\mathbf{p}(u) = \mathbf{a}_3\, u^3 + \mathbf{a}_2\, u^2 + \mathbf{a}_1\, u + \mathbf{a}_0$$

(Vector Form)

PARAMETRIC CUBIC CURVE

Algebraic to Geometric Form

$$x = a_{3x}\,u^3 + a_{2x}\,u^2 + a_{1x}\,u + a_{0x}$$

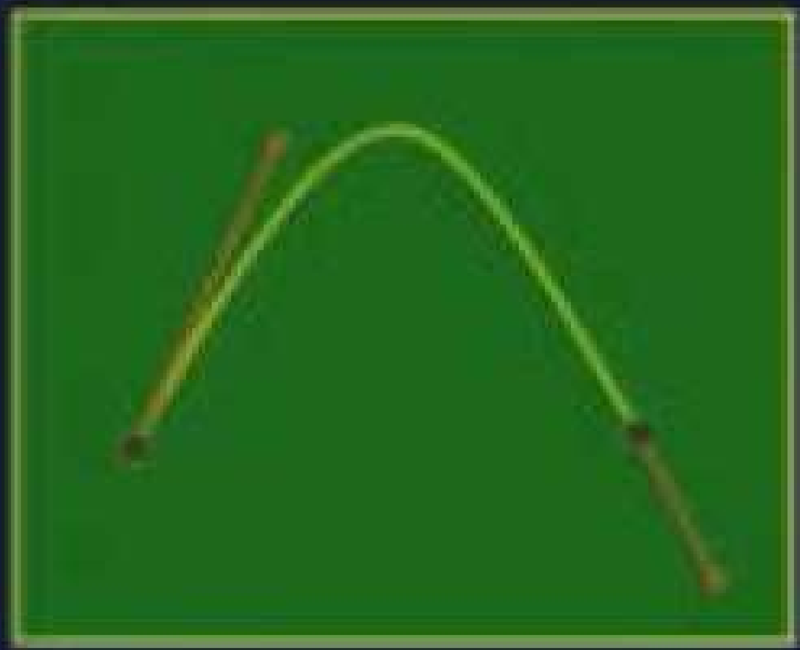
$$y = a_{3y}\,u^3 + a_{2y}\,u^2 + a_{1y}\,u + a_{0y} \quad 0 \le u \le 1$$

$$z = a_{3z}\,u^3 + a_{2z}\,u^2 + a_{1z}\,u + a_{0z}$$

$$\left.\begin{aligned} x' &= 3\,a_{3x}\,u^2 + 2\,a_{2x}\,u + a_{1x} \\ y' &= 3\,a_{3y}\,u^2 + 2\,a_{2y}\,u + a_{1y} \\ z' &= 3\,a_{3z}\,u^2 + 2\,a_{2z}\,u + a_{1z} \end{aligned}\right\} \begin{array}{c}\text{Tangent} \\ \text{Vector}\end{array}$$

PARAMETRIC CUBIC CURVE

Geometric Form



starting point

(x_0, y_0, z_0)

end point

(x_1, y_1, z_1)

starting tangent vector

(x'_0, y'_0, z'_0)

end tangent vector

(x'_1, y'_1, z'_1)

$$x'_0 = \left.\frac{dx}{du}\right|_{u=0}$$

PARAMETRIC CUBIC CURVE

Algebraic to Geometric Form

$$x = a_{3x} u^3 + a_{2x} u^2 + a_{1x} u + a_{0x}$$
$$y = a_{3y} u^3 + a_{2y} u^2 + a_{1y} u + a_{0y}$$
$$z = a_{3z} u^3 + a_{2z} u^2 + a_{1z} u + a_{0z}$$

$$x' = 3 a_{3x} u^2 + 2 a_{2x} u + a_{1x}$$
$$y' = 3 a_{3y} u^2 + 2 a_{2y} u + a_{1y}$$
$$z' = 3 a_{3z} u^2 + 2 a_{2z} u + a_{1z}$$

Substitute

$u = 0$

&

$u = 1$

PARAMETRIC CUBIC CURVE

Algebraic to Geometric Form

$x_0 = a_{0x}$ $\quad$ $x_1 = a_{3x} + a_{2x} + a_{1x} + a_{0x}$

$y_0 = a_{0y}$ $\quad$ $y_1 = a_{3y} + a_{2y} + a_{1y} + a_{0y}$

$z_0 = a_{0z}$ $\quad$ $z_1 = a_{3z} + a_{2z} + a_{1z} + a_{0z}$

$x'_0 = a_{1x}$ $\quad$ $x'_1 = 3\,a_{3x} + 2\,a_{2x} + a_{1x}$

$y'_0 = a_{1y}$ $\quad$ $y'_1 = 3\,a_{3y} + 2\,a_{2y} + a_{1y}$

$z'_0 = a_{1z}$ $\quad$ $z'_1 = 3\,a_{3z} + 2\,a_{2z} + a_{1z}$

PARAMETRIC CUBIC CURVE

Algebraic to Geometric Form

$$x(u) = (2u^3 - 3u^2 + 1)\, x_0 + (-2u^3 + 3u^2)\, x_1 + (u^3 - 2u^2 + u)\, x'_0 + (u^3 - u^2)\, x'_1$$

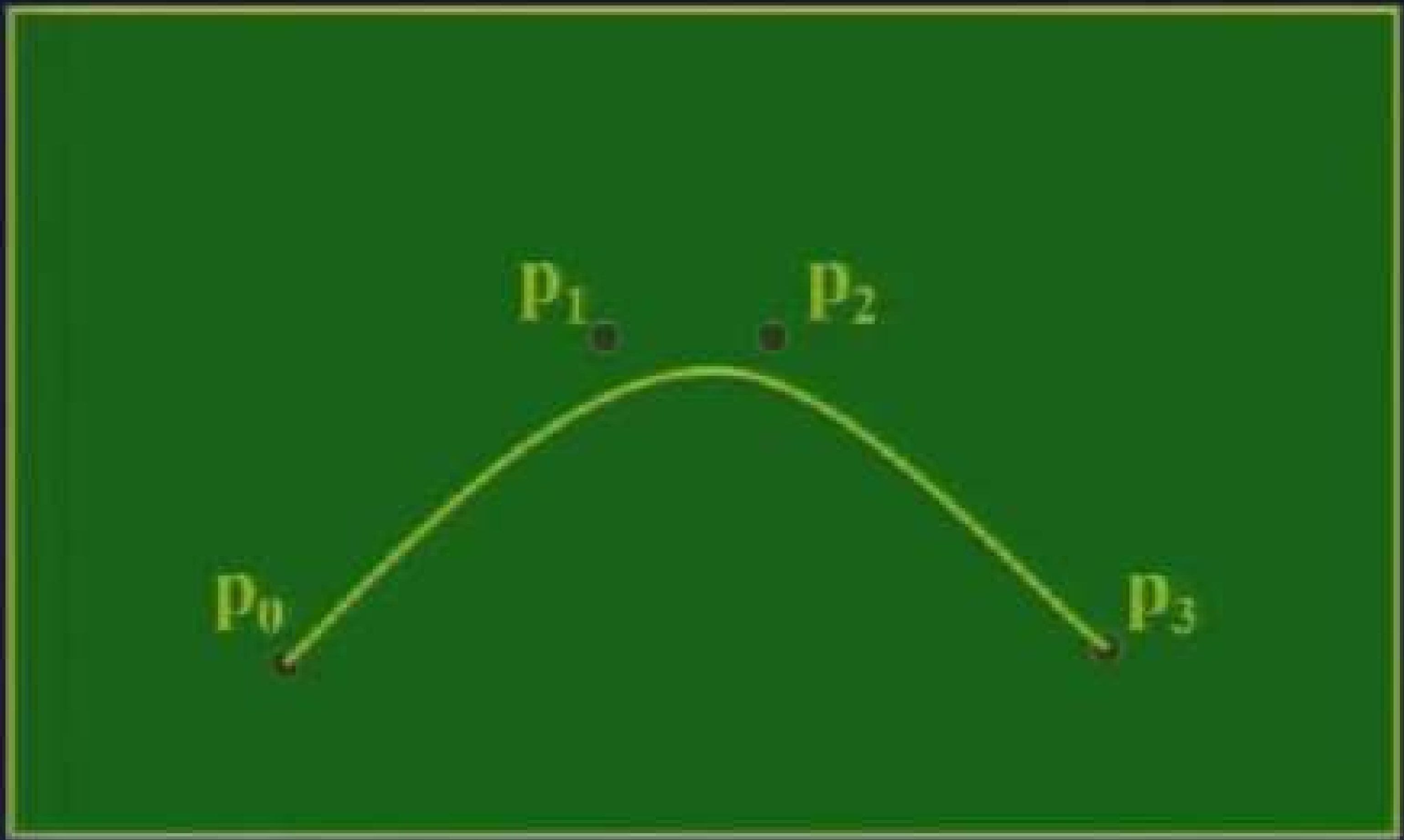
$$y(u) = (2u^3 - 3u^2 + 1)\, y_0 + (-2u^3 + 3u^2)\, y_1 + (u^3 - 2u^2 + u)\, y'_0 + (u^3 - u^2)\, y'_1$$

$$z(u) = (2u^3 - 3u^2 + 1)\, z_0 + (-2u^3 + 3u^2)\, z_1 + (u^3 - 2u^2 + u)\, z'_0 + (u^3 - u^2)\, z'_1$$

BEZIER CURVE

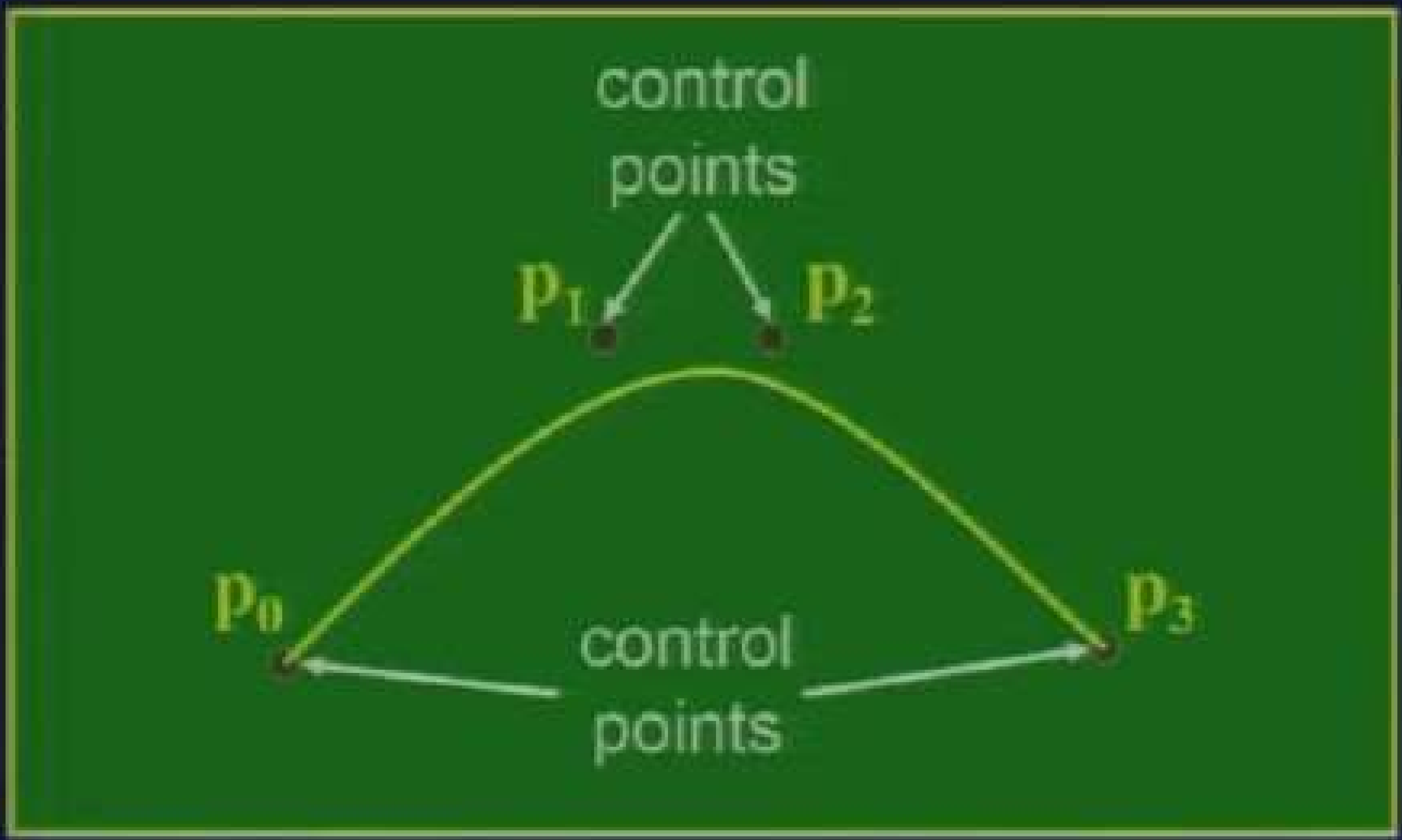
- Bezier curve is an approximation curve
- The curve was first proposed in 60's by P. Bezier
- The curve was first used to define sculptured surfaces of automobile bodies
- A cubic Bezier curve is defined by four control points

CUBIC BEZIER CURVE



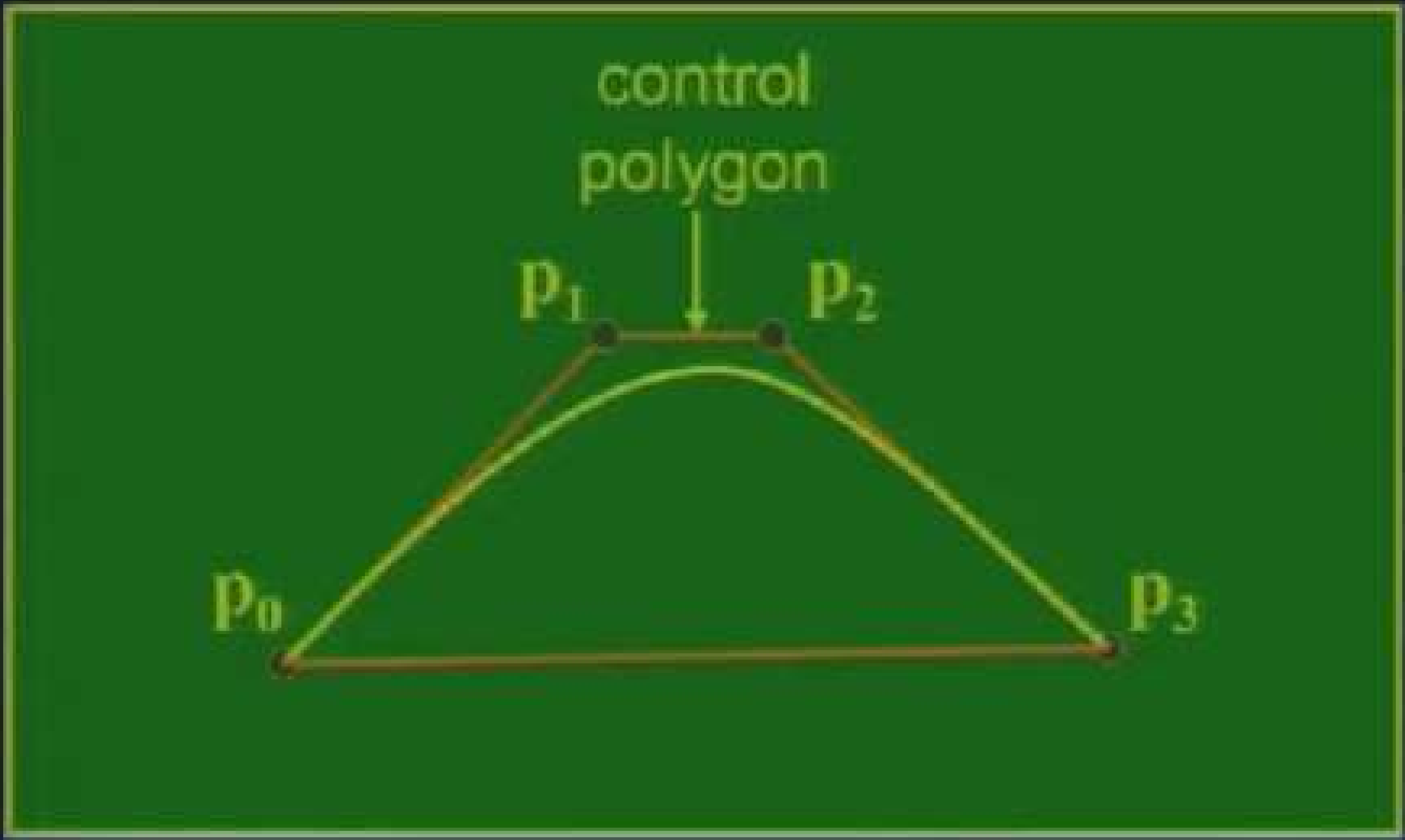
Cubic Bezier Curve

CUBIC BEZIER CURVE



Cubic Bezier Curve

CUBIC BEZIER CURVE



Cubic Bezier Curve

CUBIC BEZIER CURVE

Input to Cubic Bezier Curve

first point, $p_0 = (x_0, y_0, z_0)$

second point, $p_1 = (x_1, y_1, z_1)$

third point, $p_2 = (x_2, y_2, z_2)$

fourth point, $p_3 = (x_3, y_3, z_3)$

- *First point is starting point*
- *Fourth point is end point*
- *Order of points is important.*

CUBIC BEZIER CURVE

Definition

$$x(u) = (1-u)^3 x_0 + 3(1-u)^2 u\, x_1 + 3(1-u)u^2 x_2 + u^3 x_3$$

$$0 \le u \le 1$$

CUBIC BEZIER CURVE

Definition

$$x(u) = (1-u)^3 x_0 + 3(1-u)^2 u x_1 + 3(1-u) u^2 x_2 + u^3 x_3$$

$$y(u) = (1-u)^3 y_0 + 3(1-u)^2 u y_1 + 3(1-u) u^2 y_2 + u^3 y_3$$

$$z(u) = (1-u)^3 z_0 + 3(1-u)^2 u z_1 + 3(1-u) u^2 z_2 + u^3 z_3$$

$$0 \le u \le 1$$

CUBIC BEZIER CURVE

Definition

$$x(u) = (1-u)^3 x_0 + 3(1-u)^2 u\, x_1 + 3(1-u)\, u^2 x_2 + u^3 x_3$$

$$x(u) = {}^3c_0 (1-u)^3 x_0 + {}^3c_1 (1-u)^2 u\, x_1 + {}^3c_2 (1-u)\, u^2 x_2 + {}^3c_3\, u^3 x_3$$

$$x(u) = \sum_{i=0}^{i=3} {}^3c_i (1-u)^{3-i} u^i x_i$$

BEZIER CURVE

Cubic Bezier Curve

$$x(u) = \sum_{i=0}^{i=3} {}^3c_i (1-u)^{3-i} u^i x_i$$

Quartic Bezier Curve

$$x(u) = \sum_{i=0}^{i=4} {}^4c_i (1-u)^{4-i} u^i x_i$$

BEZIER CURVE

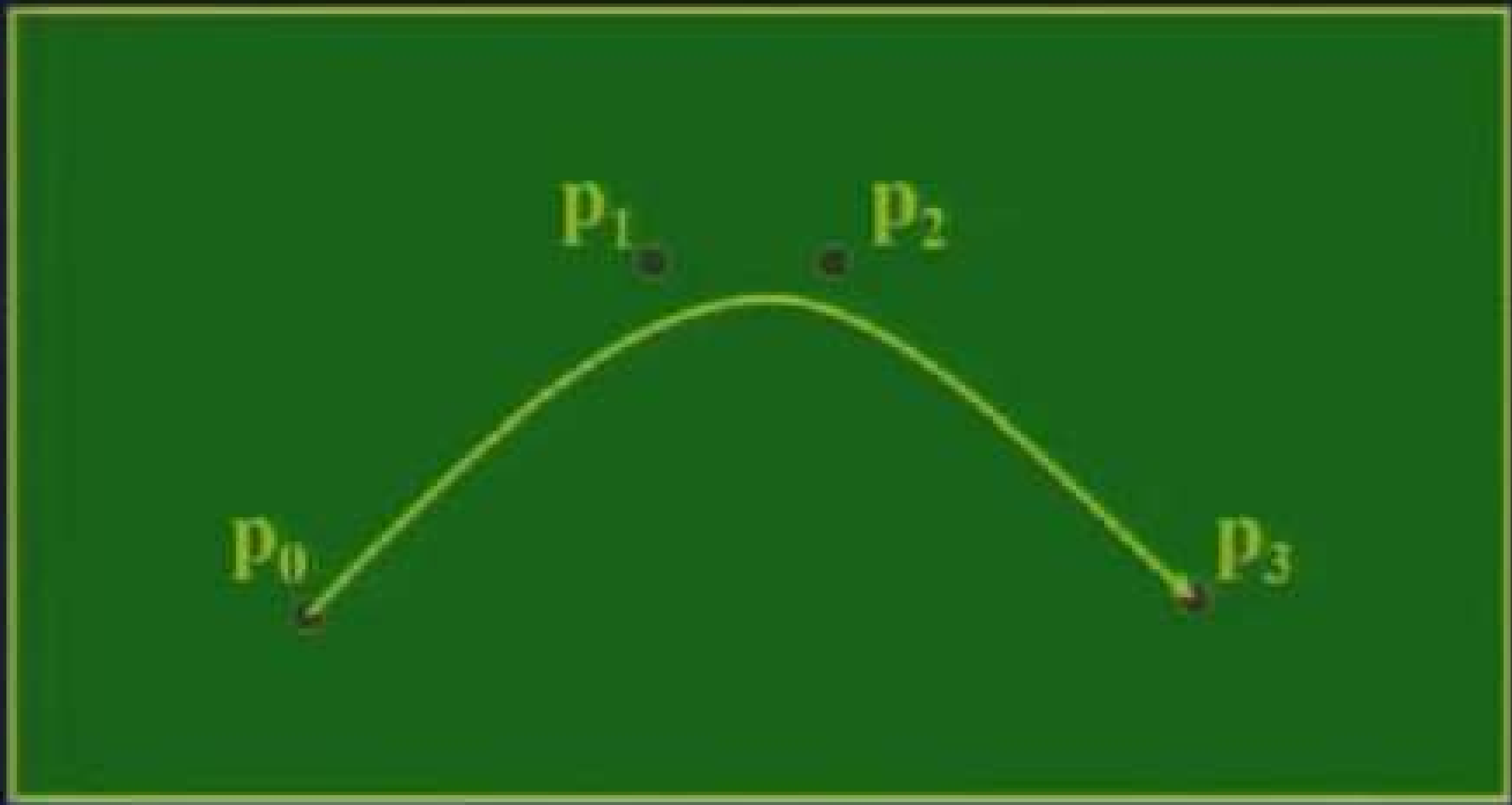
Quintic Bezier Curve

$$x(u) = \sum_{i=0}^{i=5} {}^{5}c_i \, (1-u)^{5-i} \, u^i \, x_i$$

Generic Bezier Curve

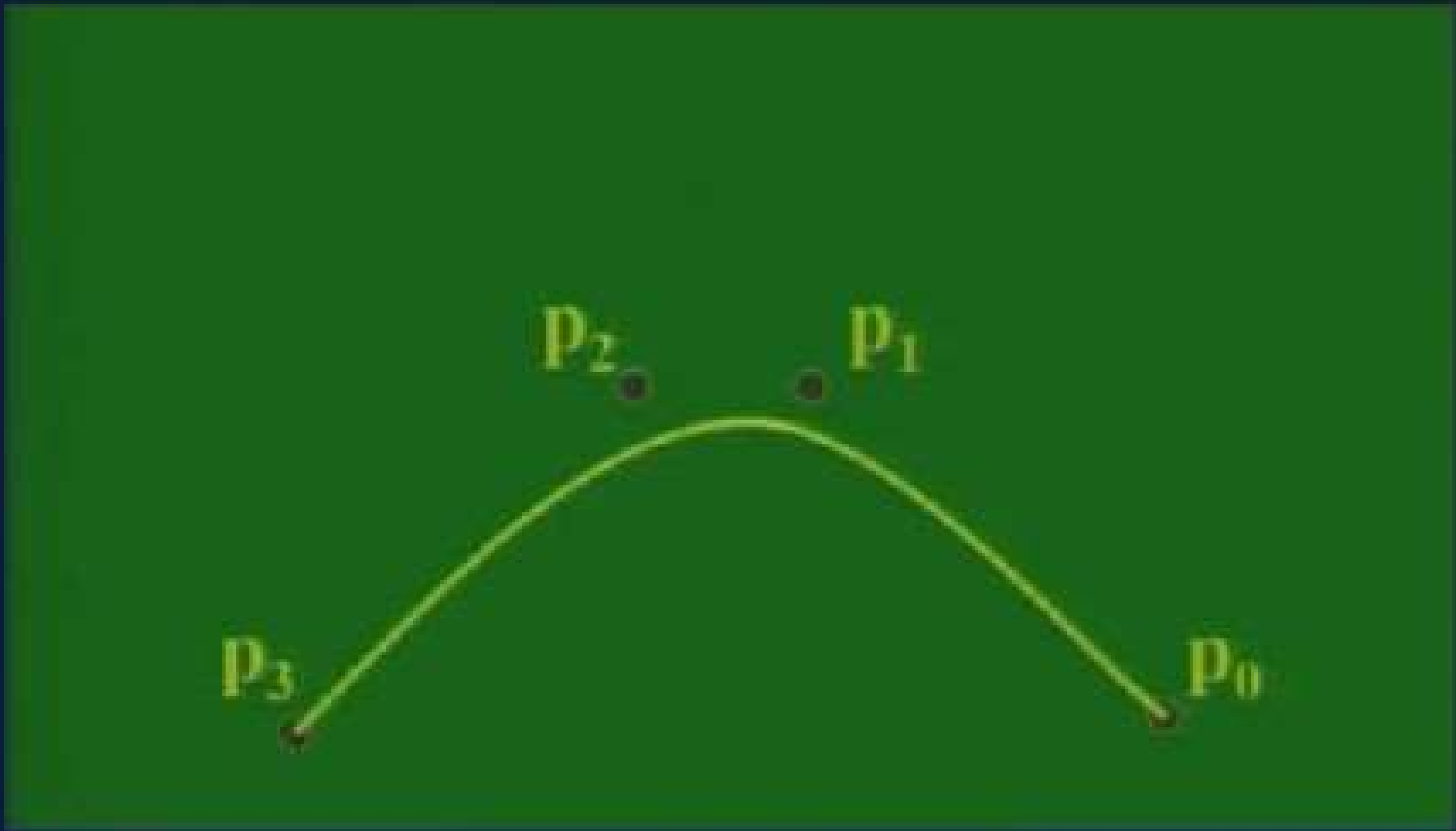
$$x(u) = \sum_{i=0}^{i=n} {}^{n}c_i \, (1-u)^{n-i} \, u^i \, x_i$$

PROPERTIES OF BEZIER CURVE



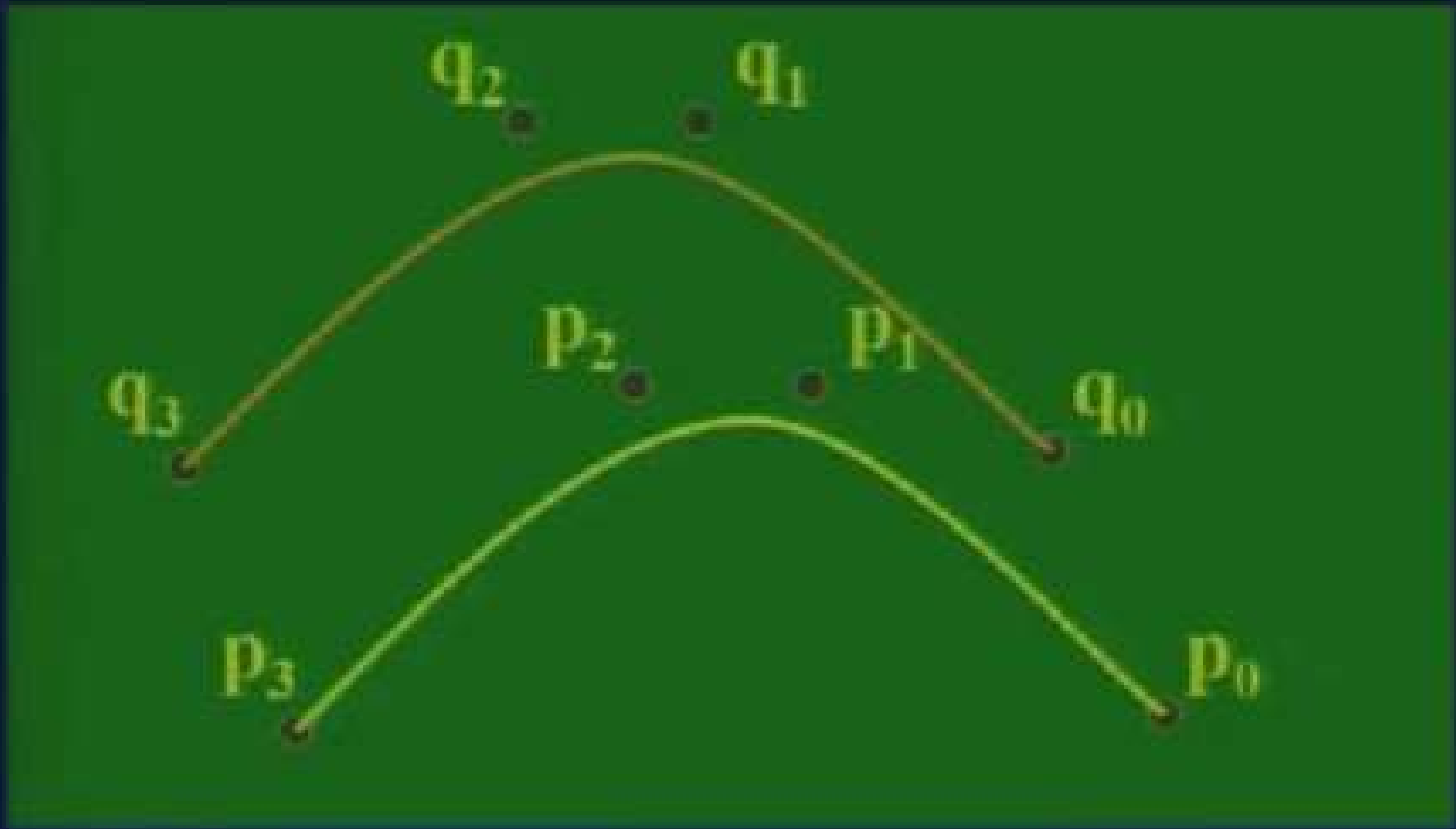
Reversing the sequence of control points does not change the shape of curve.

PROPERTIES OF BEZIER CURVE



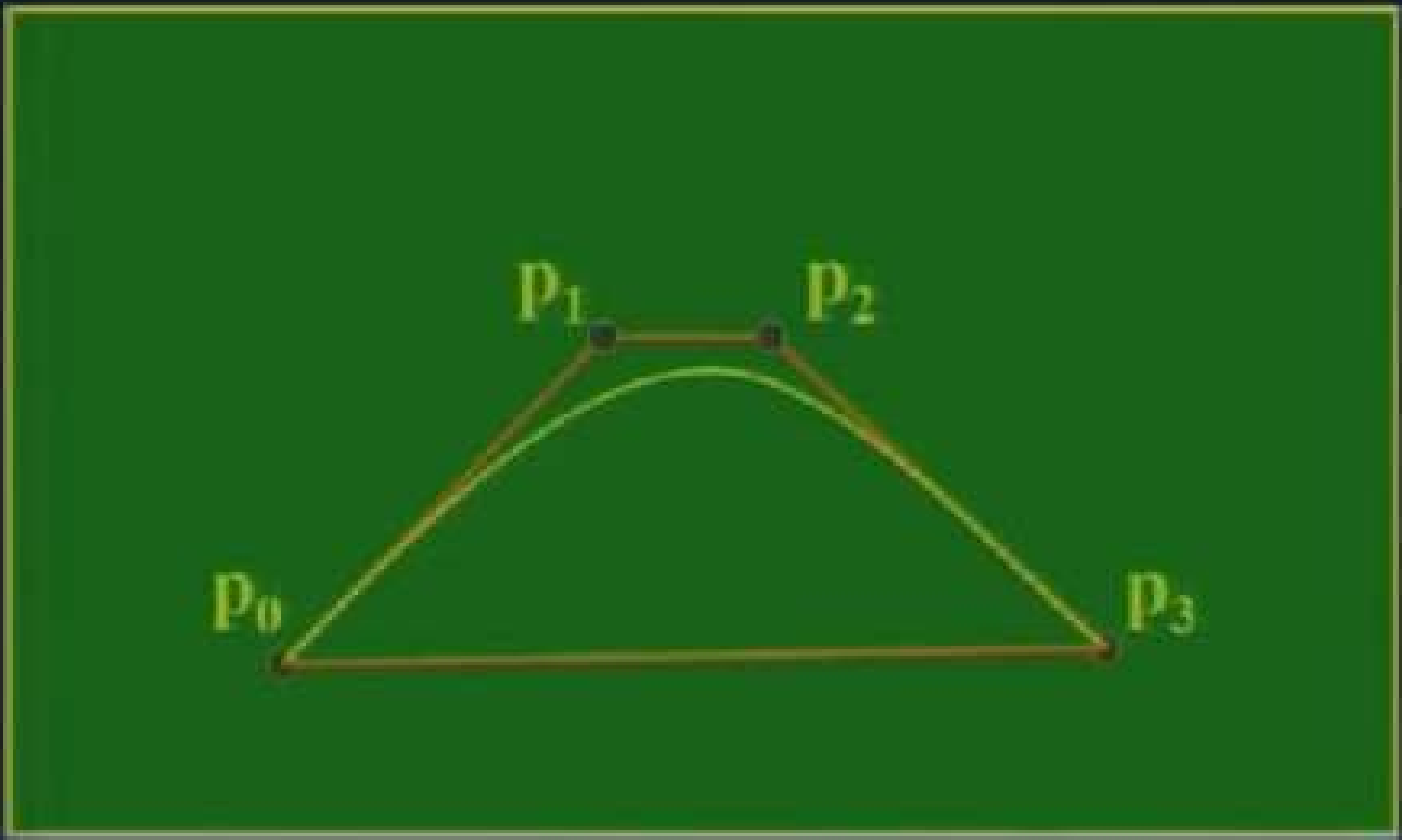
The curve is invariant under an affine transformation

PROPERTIES OF BEZIER CURVE



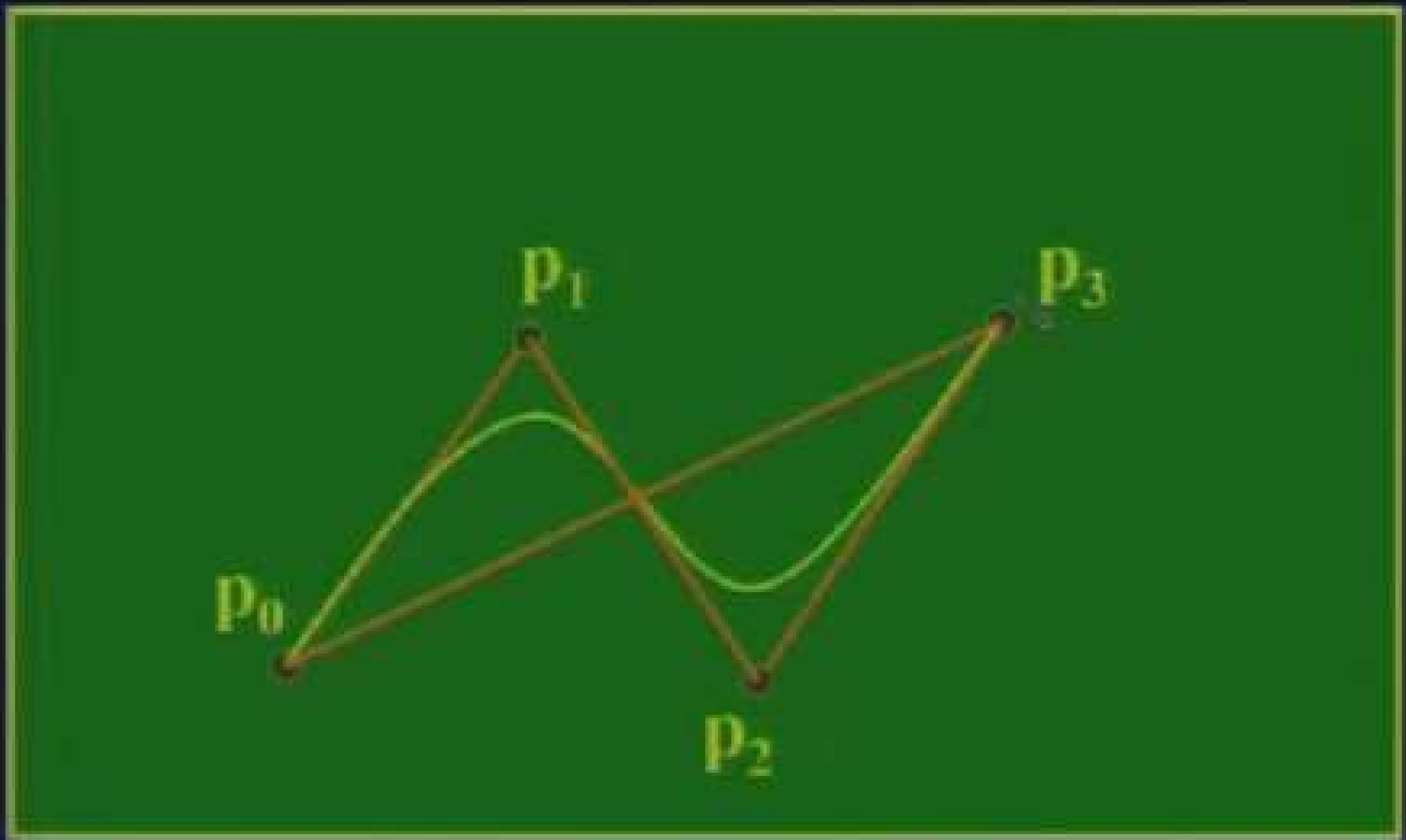
The curve is invariant under an affine transformation

PROPERTIES OF BEZIER CURVE



Convex Hull Property

PROPERTIES OF BEZIER CURVE



Convex Hull Property

PROPERTIES OF BEZIER CURVE

Partition of Unity Property

$$\sum_{i=0}^{i=n} {}^{n}C_i \,(1-u)^{n-i} u^i = 1$$

For i = 3,

$$(1-u)^3 + 3\,(1-u)^2 u + 3\,(1-u)\, u^2 + u^3 = 1$$

PROPERTIES OF BEZIER CURVE

Partition of Unity Property

For i = 3,

$(1-u)^3 + 3\,(1-u)^2\,u + 3\,(1-u)\,u^2 + u^3 = 1$

when u = 0.6

$0.064 + 3 * 0.16 * 0.6 + 3 * 0.4 * 0.36 + 0.216$

$0.064 + 0.288 + 0.432 + 0.216$

1.0

PROPERTIES OF BEZIER CURVE

Curve definition in matrix form

$$x(u) = (1-u)^3 x_0 + 3(1-u)^2 u\, x_1 + 3(1-u) u^2 x_2 + u^3 x_3$$

$$x(u) = (1 - 3u + 3u^2 - u^3) x_0 + (3u - 6u^2 + 3u^3) x_1 + (3u^2 - 3u^3) x_2 + u^3 x_3$$

PROPERTIES OF BEZIER CURVE

Curve definition in matrix form

$$x(u) = (1 - 3u + 3u^2 - u^3)\,x_0 + (3u - 6u^2 + 3u^3)\,x_1 + (3u^2 - 3u^3)\,x_2 + u^3\,x_3$$

$$x(u) = (-x_0 + 3x_1 - 3x_2 + x_3)\,u^3 + (3x_0 - 6x_1 + 3x_2)\,u^2 + (-3x_0 + 3x_1)\,u + x_0$$

PROPERTIES OF BEZIER CURVE

Curve definition in matrix form

$$x(u) = (1-u)^3 x_0 + 3(1-u)^2 u\, x_1 + 3(1-u) u^2 x_2 + u^3 x_3$$

$$x(u) = (1 - 3u + 3u^2 - u^3) x_0 + (3u - 6u^2 + 3u^3) x_1 + (3u^2 - 3u^3) x_2 + u^3 x_3$$

PROPERTIES OF BEZIER CURVE

Curve definition in matrix form

$$x(u) = [u^3\ u^2\ u\ 1] \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

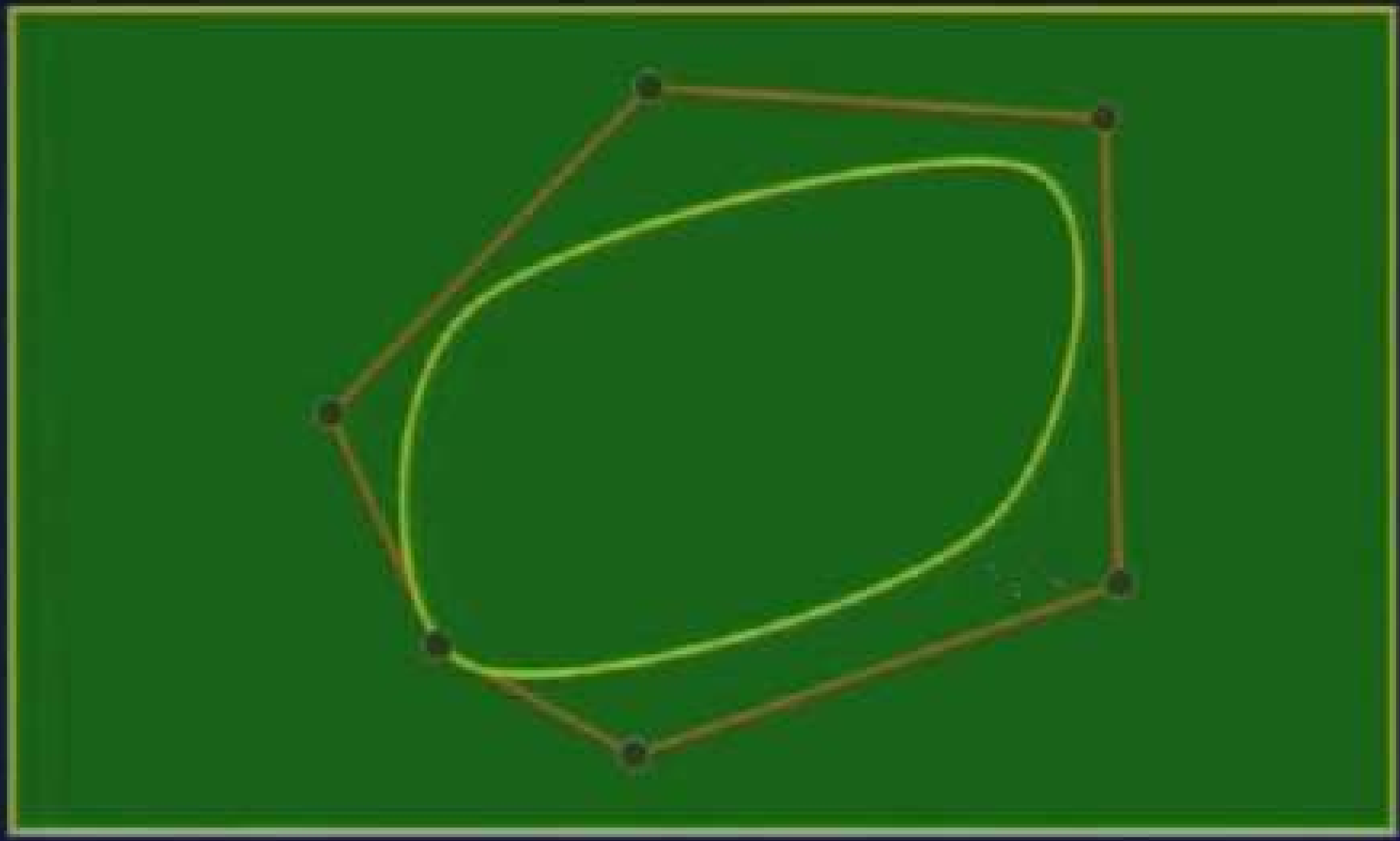
PROPERTIES OF BEZIER CURVE

Curve definition in matrix form

$$x(u) = (1 - 3u + 3u^2 - u^3)\,x_0 + (3u - 6u^2 + 3u^3)\,x_1 + (3u^2 - 3u^3)\,x_2 + u^3\,x_3$$

$$x(u) = (-x_0 + 3x_1 - 3x_2 + x_3)\,u^3 + (3x_0 - 6x_1 + 3x_2)\,u^2 + (-3x_0 + 3x_1)\,u + x_0$$

CLOSED BEZIER CURVE



Closed Bezier Curve

B-SPLINE CURVE

- Curve is defined by n+1 control points and the order (k) of the curve
- The curve has an advantage that it has local propagation unlike global propagation properties of Bezier curve.
- The curve can be used to define both open and closed curves.

BEZIER vs. B-SPLINE CURVE

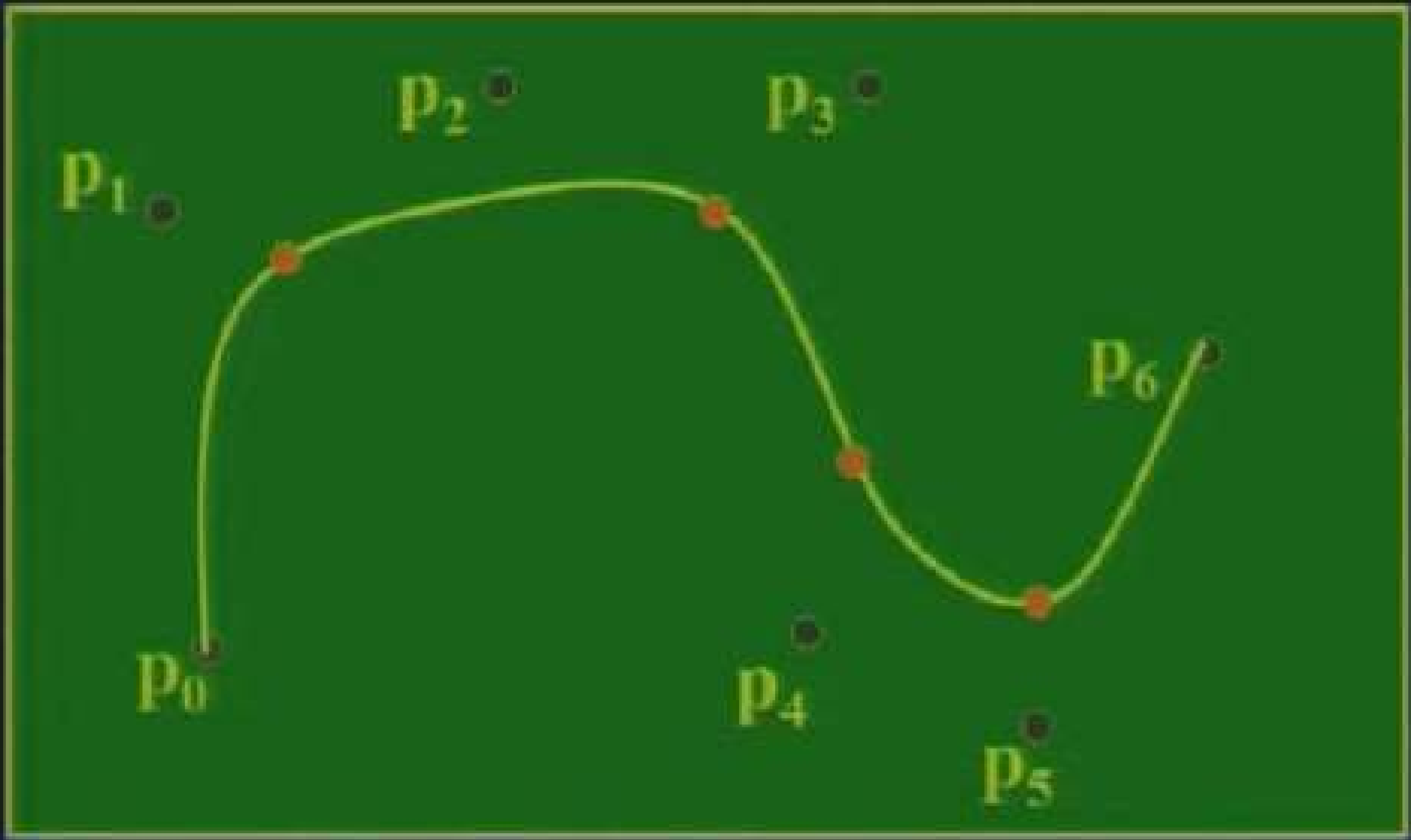
Bezier Curve

$$x(u) = \sum_{i=0}^{i=n} {}^{n}C_i \, (1-u)^{n-i} \, u^i \, x_i \qquad 0 \le u \le 1$$

B-Spline Curve

$$x(u) = \sum_{i=0}^{i=n} N_{i,k}(u) \, x_i \qquad 0 \le u \le n-k+2$$

NON-UNIFORM B-SPLINE CURVE



Non-Uniform B-Spline Curve
$(n = 6, k = 3)$

BEZIER vs. B-SPLINE CURVE

Bezier Curve (n = 5)

$$x = (1-u)^5 x_0 + 5(1-u)^4 u x_1 + 10(1-u)^3 u^2 x_2 + 10(1-u)^2 u^3 x_3 + 5(1-u) u^4 x_4 + u^5 x_5$$

$$0 \leq u \leq 1$$

Non-Uniform B-Spine Curve (n = 5, k = 3)

$$x(u) = N_{0,3}(u) x_0 + N_{1,3}(u) x_1 + N_{2,3}(u) x_2 + N_{3,3}(u) x_3 + N_{4,3}(u) x_4 + N_{5,3}(u) x_5$$

$$0 \leq u \leq 4$$

B-SPLINE CURVE

Non-Uniform B-Spline Curve (n = 5, k = 3)

$$x = (1-u)^2 x_0 + 0.5\,u\,(4-3u)\,x_1 + 0.5\,u^2 x_2$$

$$0 \leq u < 1$$

$$x = 0.5\,(2-u)^2 x_1 + 0.5\,u\,(-2u^2 + 6u - 3)\,x_2 + 0.5\,(u-1)^2 x_3$$

$$1 \leq u < 2$$

$$x = 0.5\,(3-u)^2 x_2 + 0.5\,u\,(-2u^2 + 10u - 11)\,x_3 + 0.5\,(u-2)^2 x_4$$

$$2 \leq u < 3$$

$$x = 0.5\,(4-u)^2 x_3 + 0.5\,u\,(-3u^2 + 20u - 32)\,x_4 + 0.5\,(u-3)^2 x_5$$

$$3 \leq u < 4$$

B-SPLINE CURVE

Non-Uniform B-Spline Curve Properties

- The curve is $C^{(k-2)}$ Continuous
- The curve is made up of (n-k+2) segments
- Only k control points affect any segment of the curve
- A given control point affects 1 or 2 or …k curve segments.

CUBIC BEZIER CURVE

Input to B-Spline Curve

1st point, $p_0 = (x_0, y_0, z_0)$

2nd point, $p_1 = (x_1, y_1, z_1)$

|

nth point, $p_n = (x_n, y_n, z_n)$

Order of curve = k

B-SPLINE BASIS FUNCTION

$$N_{i,k}(u) = \frac{(u-t_i)\,N_{i,k-1}(u)}{t_{i+k-1}-t_i} + \frac{(t_{i+k}-u)\,N_{i+1,k-1}(u)}{t_{i+k}-t_{i+1}}$$

t_i $(0 \le i \le n+k)$ are called knot values

$t_i = 0$ if $i < k$;

$t_i = i-k+1$ if $k \le i \le n$;

$t_i = n-k+2$ if $i > n$

$$N_{i,k}(u) = 1 \text{ if } t_i \le u \le t_{i+1}$$
$$= 0 \text{ otherwise}$$

B-SPLINE BASIS FUNCTION

n = 5 and k =3

t_i ($0 \leq i \leq 8$) are knot values

$t_i = \{0, 0, 0, 1, 2, 3, 4, 4, 4\}$

$$N_{0,3}(u) = (1-u)^2 N_{2,1}(u)$$

$$N_{1,3}(u) = 0.5\, u\,(4-3u)\, N_{2,1}(u) +$$

$$0.5\,(2-u)^2 N_{3,1}(u)$$

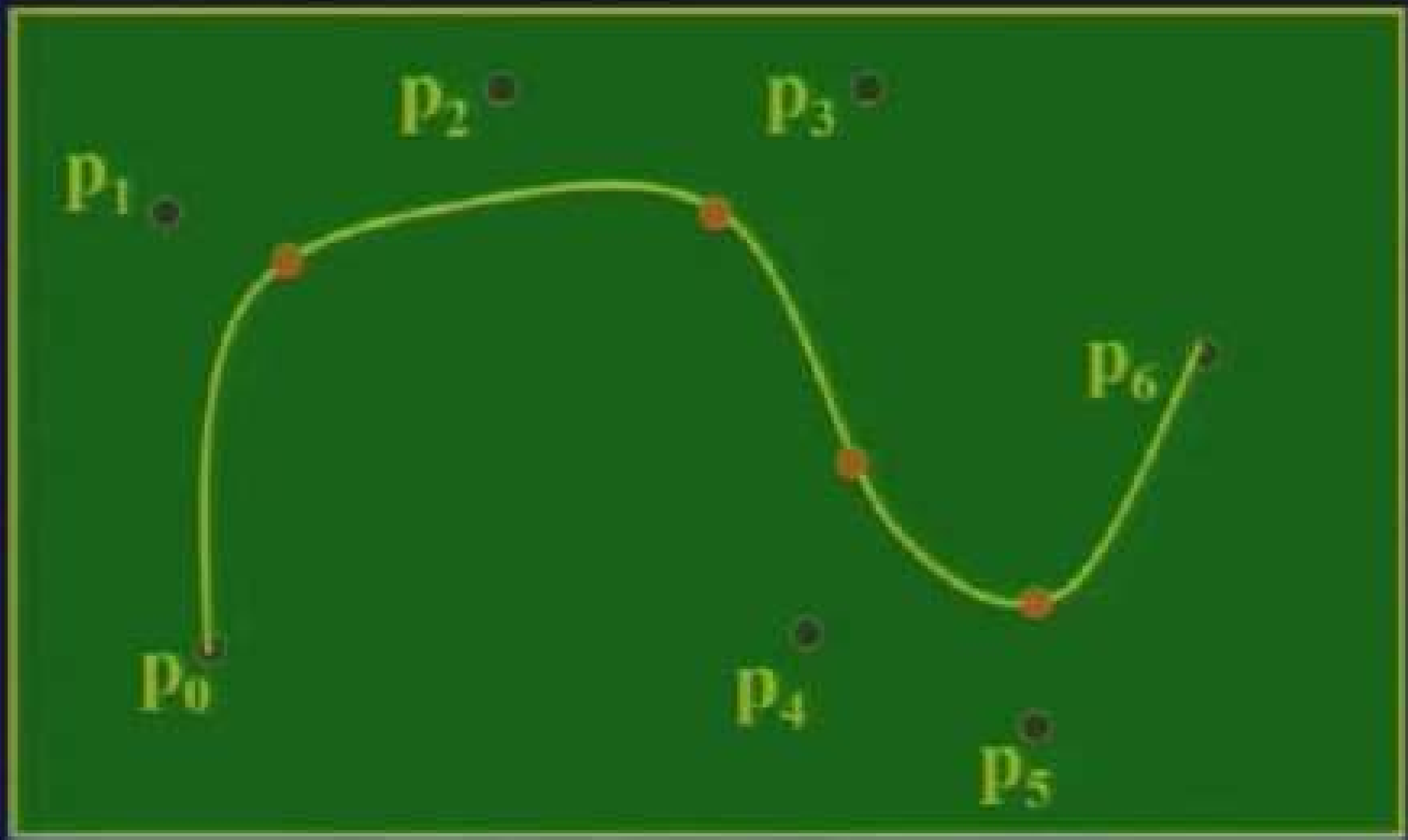
NURBS CURVE

Non-Uniform B-Spline Curve

$$x(u) = \sum_{i=0}^{i=n} N_{i,k}(u)\, x_i \qquad 0 \le u \le n-k+2$$

$$x(u) = \sum_{i=0}^{i=n} h_i N_{i,k}(u)\, x_i \Big/ \sum_{i=0}^{i=n} h_i N_{i,k}(u)$$

NURBS CURVE



NURBS Curve

(n = 6, k = 3)

PARAMETRIC SURFACES

- Implicit Surfaces

$$x^2 + y^2 + z^2 - r^2 = 0$$

- Explicit Surfaces

$$z = a\,x + b\,y + c$$

- Parametric Surfaces

$$x = r\cos(2\pi u)$$
$$y = r\sin(2\pi u)$$
$$z = h\,w$$

PARAMETRIC SURFACES

- Quadric Surface

$$A\,x^2 + B\,y^2 + C\,z^2 + 2\,D\,x\,y + 2\,E\,y\,z + 2\,F\,x\,z + 2\,G\,x + 2\,H\,y + 2\,J\,z + K = 0$$

Above equation can be used to represent:

Plane, Two parallel or intersecting planes, Line, Cylinder, Cone, Point, Ellipsoid, Paraboloid, Hyperboloid etc.

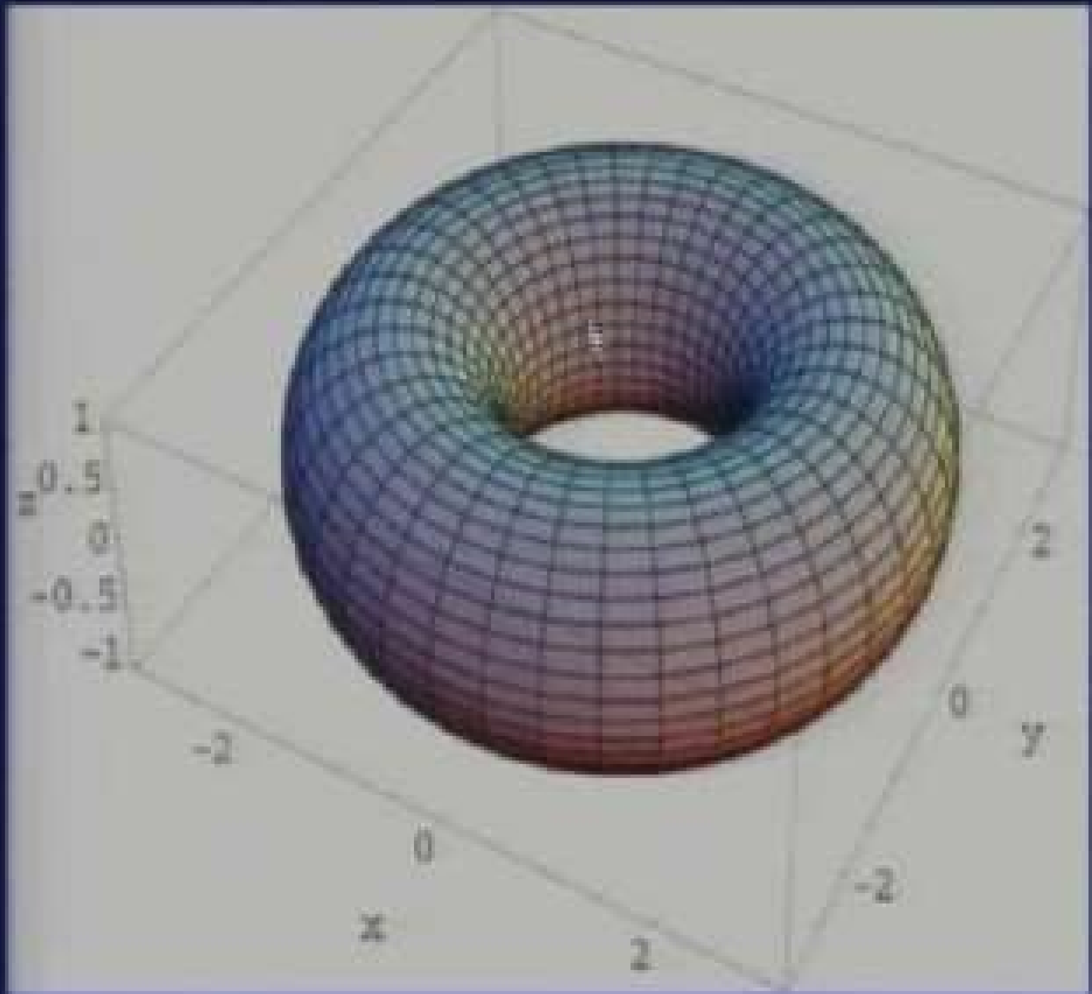
PARAMETRIC SURFACES

Surfaces of Known Form

- Plane surface
- Cylindrical surface
- Conical surface
- Spherical Surface
- Toroidal Surface

PARAMETRIC SURFACES

Toroidal Surface



PARAMETRIC SURFACES

Swept Surfaces

A curve (generatrix curve) is swept along a path defined by another curve (directrix curve). The area swept is a surface.

- Linearly Swept Surface
- Circularly Swept Surface
- Generic Swept Surface

PARAMETRIC SURFACES

Surfaces of Known Form can be modelled as swept surfaces

- Plane surface (line swept along another line)
- Cylindrical surface (line swept along circle or circle swept along line)
- Conical surface (line swept along circle)
- Spherical Surface (circle swept along circle)
- Toroidal Surface (circle swept along circle)

PARAMETRIC SURFACES

Extension of Free-form Parametric Curves to Surfaces

Hermite Surface Patch

- Algebraic form
- Geometric form
- n-point form

Bezier Surface Patch (m by n points)

B-spline Surface Patch (m by n points)

NURBS Surface Patch (m by n points & weights)

PARAMETRIC SURFACES

Parametric Surfaces Defined by Boundary Curves

- Coons Surface Patch (four boundary curves)
- Ferguson Patch (four boundary curves)
- Ruled Surface (two boundary curves)
- Bilinear Surface (four corner points)

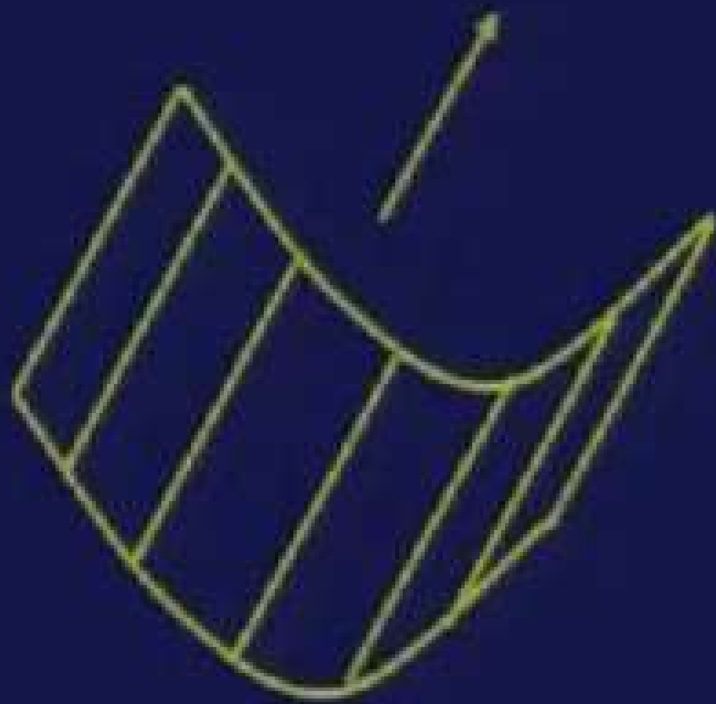
PARAMETRIC SURFACES

Linearly Swept Surface

Curve { $x(u)$, $y(u)$, $z(u)$ } is swept along unit vector { l, m, n } by distance d

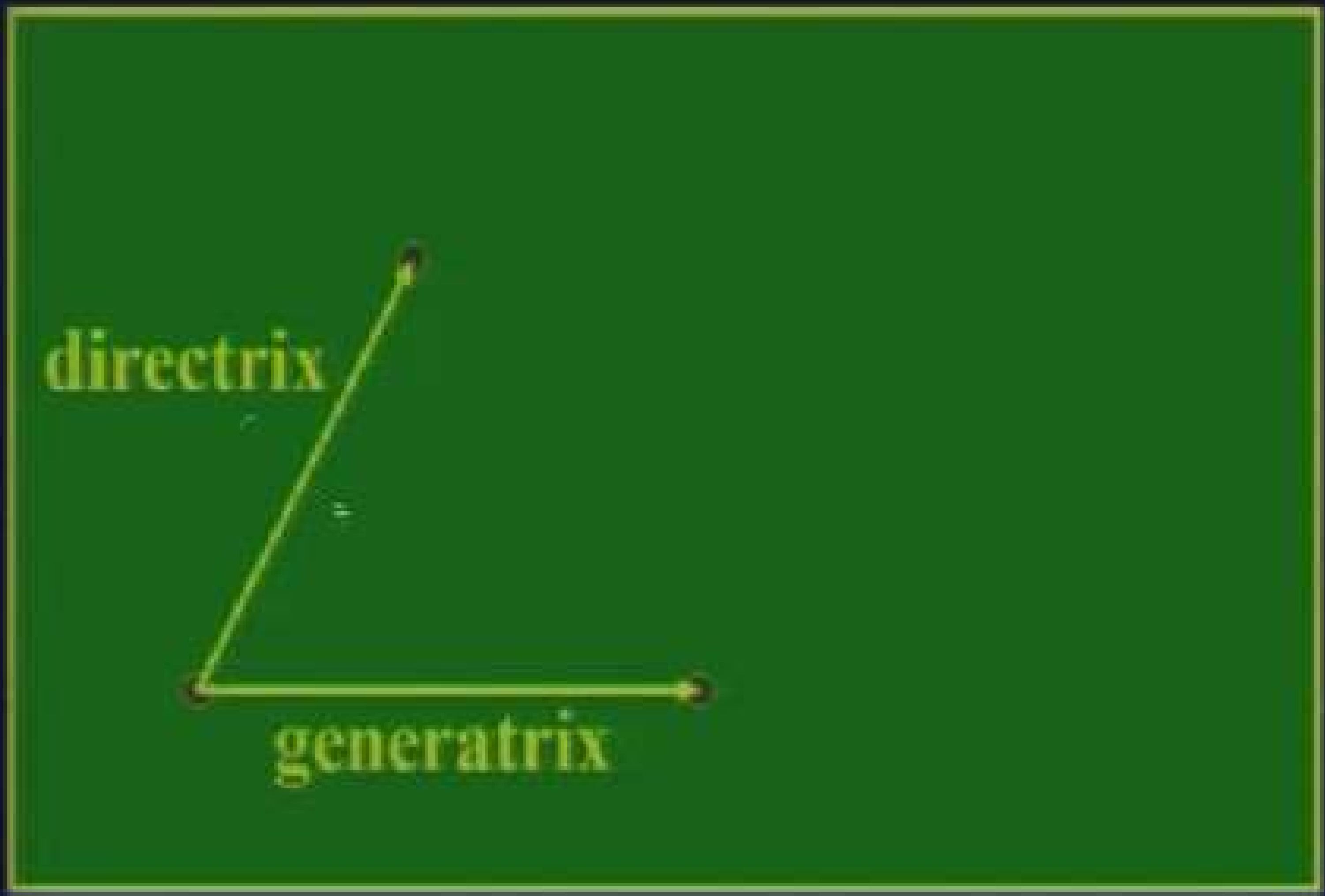
$$x(u, w) = x(u) + l\,d\,w$$
$$y(u, w) = y(u) + m\,d\,w$$
$$z(u, w) = z(u) + n\,d\,w$$

$0 \leq u \leq 1 \quad 0 \leq w \leq 1$



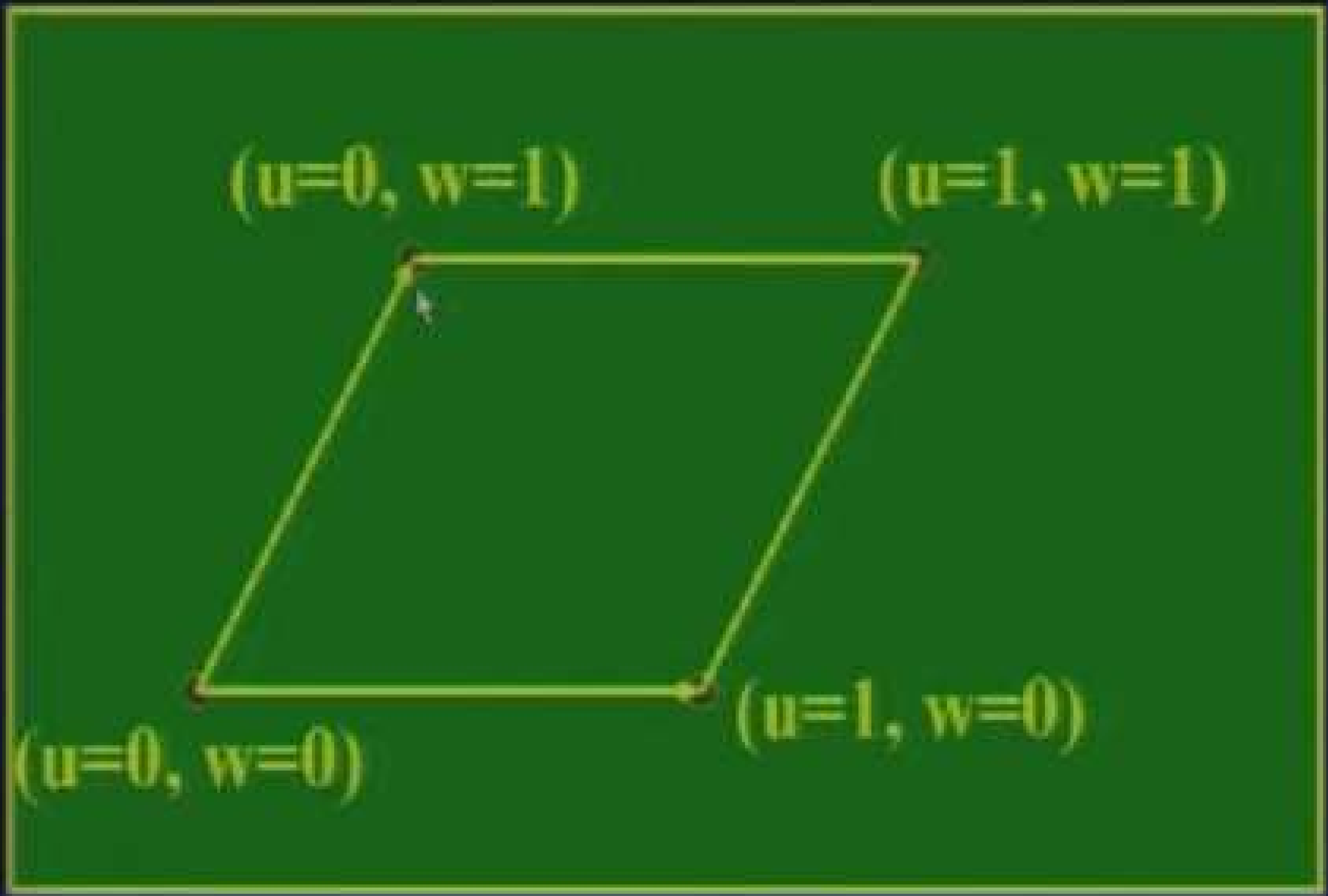
PARAMETRIC SURFACES

Plane Surface



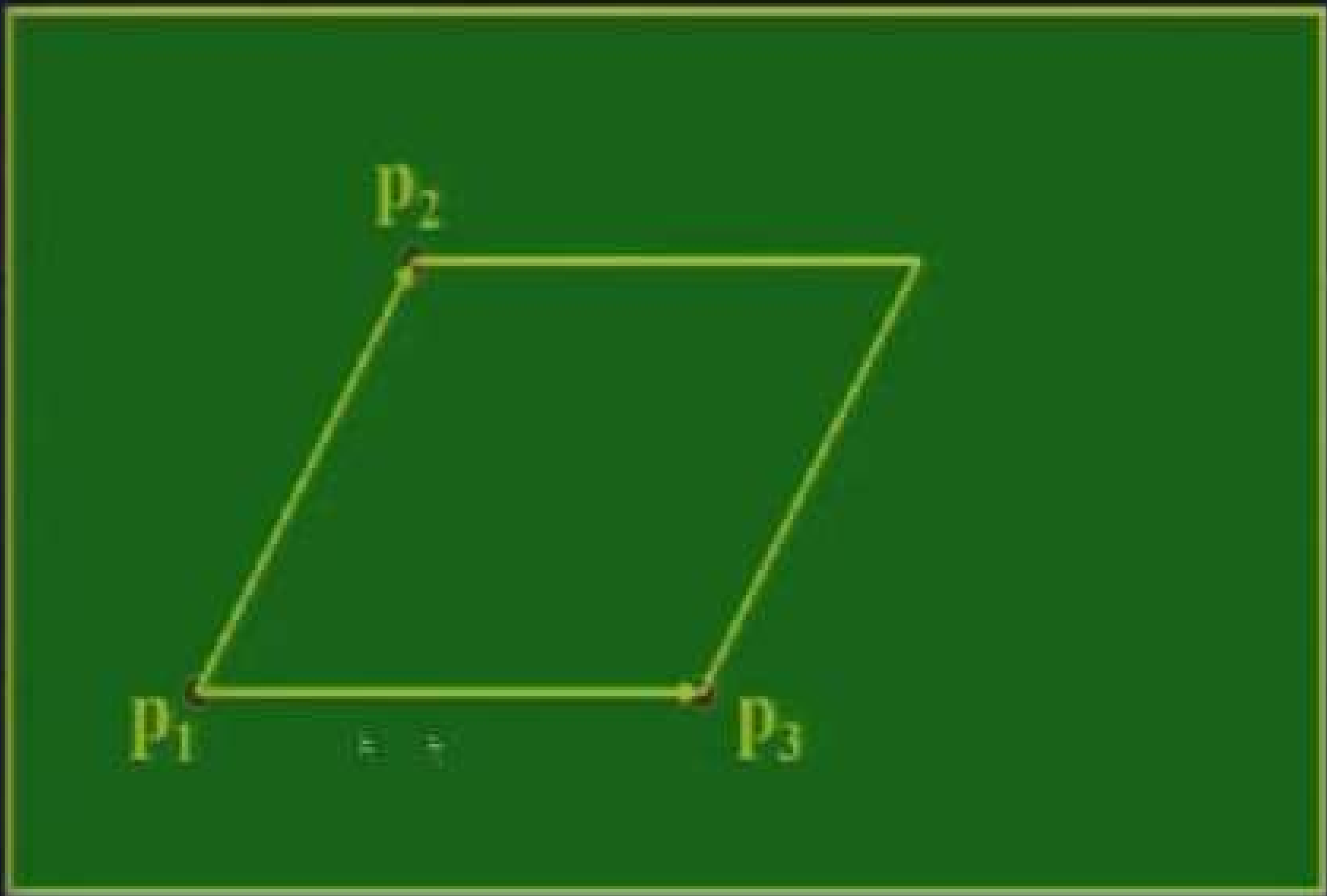
PARAMETRIC SURFACES

Plane Surface



PARAMETRIC SURFACES

Plane Surface



PARAMETRIC SURFACES

Plane Surface

Defined by three points in a space p_1, p_2, p_3

$(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)$

Parametric equation of line joining p_1 and p_2

$x(u) = x_1 + (x_2 - x_1)\, u$

$y(u) = y_1 + (y_2 - y_1)\, u$

$z(u) = z_1 + (z_2 - z_1)\, u$

$0 \le u \le 1$

PARAMETRIC SURFACES

Plane Surface

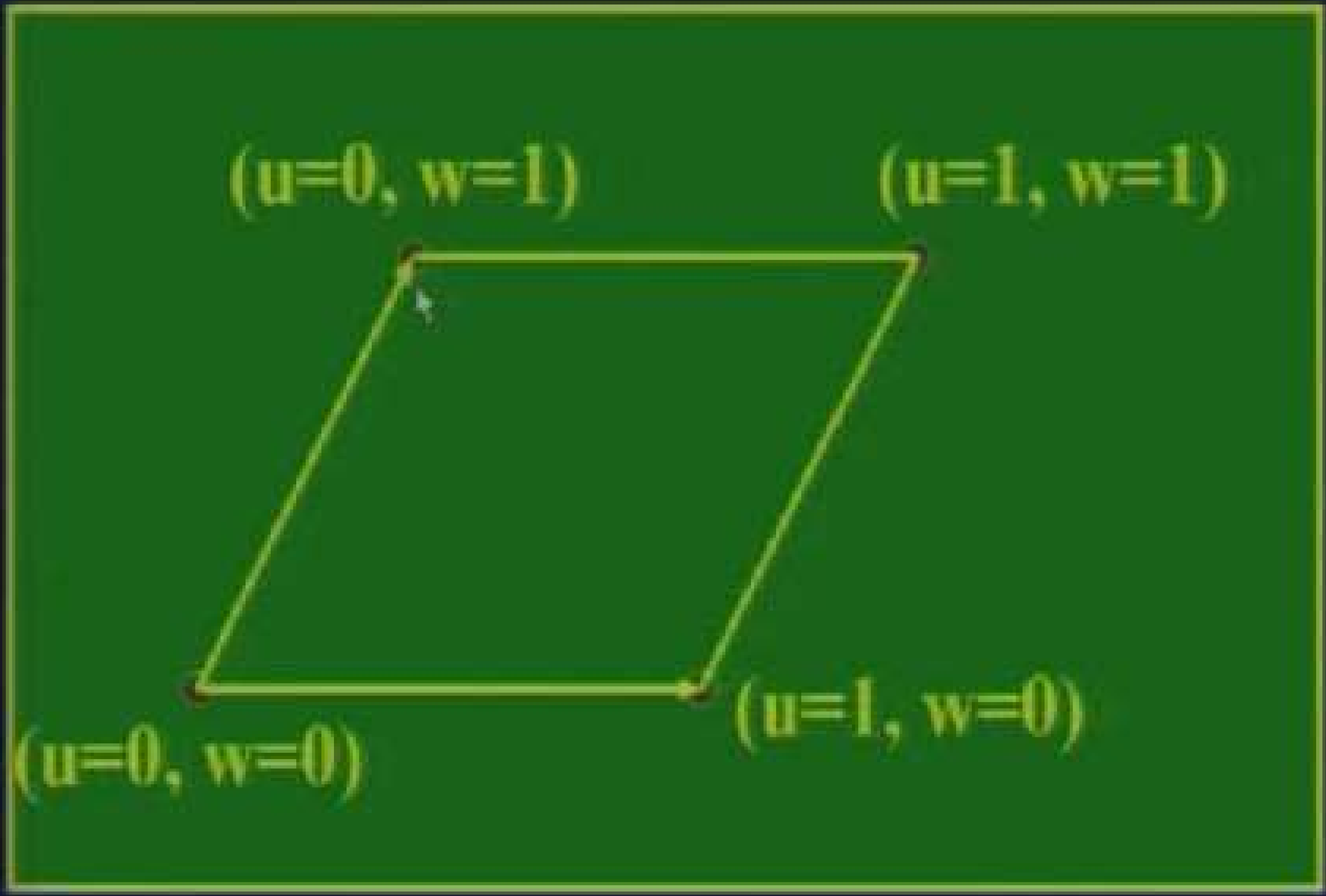
Line joining points $\mathbf{p}_1$ and $\mathbf{p}_2$, is swept along vector $\mathbf{p}_3 - \mathbf{p}_1$

$$x(u) = x_1 + (x_2 - x_1)\,u + (x_3 - x_1)\,w$$
$$y(u) = y_1 + (y_2 - y_1)\,u + (y_3 - y_1)\,w$$
$$z(u) = z_1 + (z_2 - z_1)\,u + (z_3 - z_1)\,w$$

$$0 \le u \le 1 \quad 0 \le w \le 1$$

PARAMETRIC SURFACES

Plane Surface



PARAMETRIC SURFACES

Circularly Swept Surface

x(u) and y (u) is swept along y-axis

$X(u, w) = x(u) \cos(2\pi w)$

$Y(u, w) = y(u)$

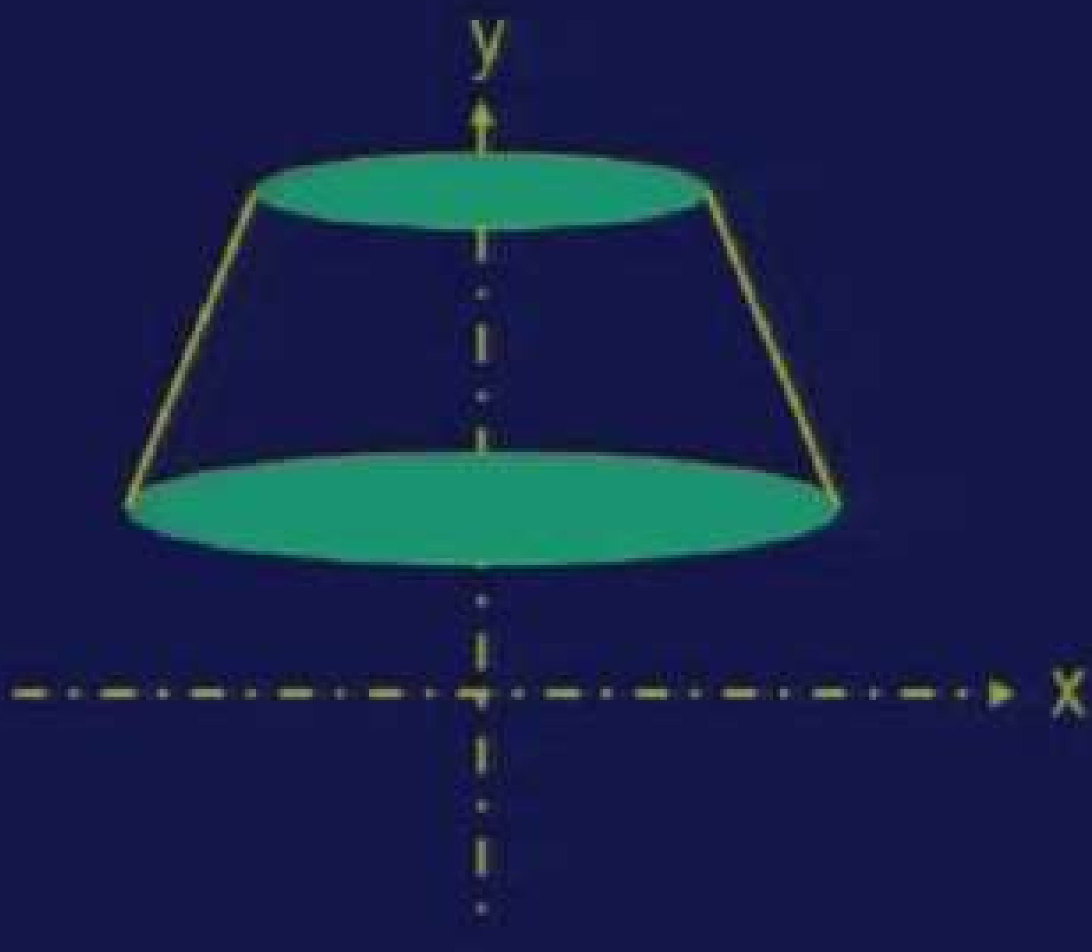
$Z(u, w) = x(u) \sin(2\pi w)$

PARAMETRIC SURFACES

Circularly Swept Surface

PARAMETRIC SURFACES

Circularly Swept Surface



PARAMETRIC SURFACES

Cylindrical Surface

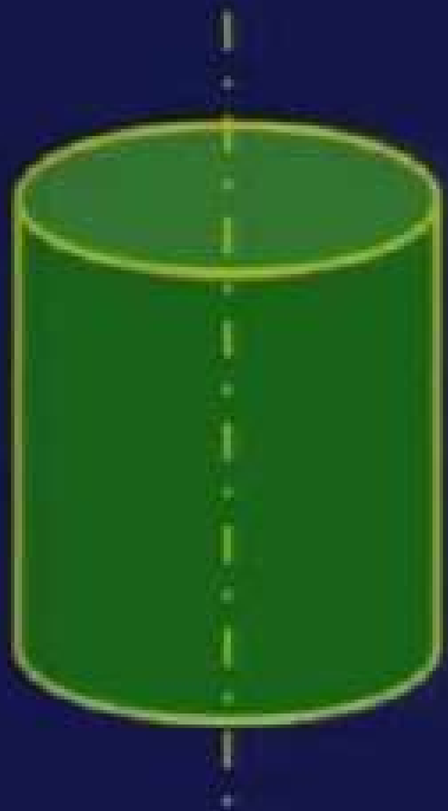
Axis of cylinder = z axis, radius = r, height = h

$x(u, w) = r\cos(2\pi u)$

$y(u, w) = r\sin(2\pi u)$

$x(u, w) = h\,w$

$0 \le u \le 1 \quad 0 \le w \le 1$



PARAMETRIC SURFACES

Conical Surface

Axis of cone = z axis, radius1 = r_1,

radius2 = r_2 height = h

$$x(u, w) = \{ r1 + (r2 - r1)\, w \} \cos(2\pi u)$$

$$y(u, w) = \{ r1 + (r2 - r1)\, w \} \sin(2\pi u)$$

$$x(u, w) = h\, w$$

$0 \leq u \leq 1 \quad 0 \leq w \leq 1$

PARAMETRIC SURFACES

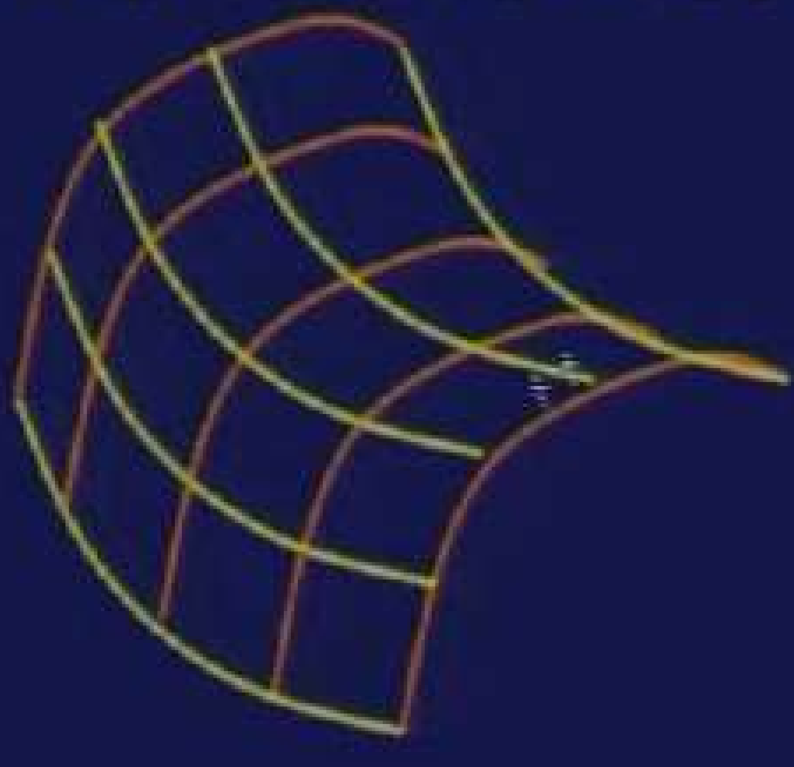
Generic Swept Surface

Curve { $x(u)$, $y(u)$, $z(u)$ } is swept along another curves { $x'(w)$, $y'(w)$, $z'(w)$ }

PARAMETRIC SURFACES

Generic Swept Surface

Curve { x(u), y(u), z(u) } is swept along another curves { x'(w), y'(w), z'(w) }



PARAMETRIC SURFACES

Extension of Free-form Parametric Curves to Surfaces

Hermite Surface Patch

- Algebraic form
- Geometric form
- n-point form

Bezier Surface Patch (m by n points)

B-spline Surface Patch (m by n points)

NURBS Surface Patch (m by n points & weights)

BEZIER PATCH

Cubic Bezier Curve

$$x(u) = \sum_{i=0}^{i=3} {}^3c_i \, (1-u)^{3-i} \, u^i \, x_i$$

Bicubic Bezier Patch

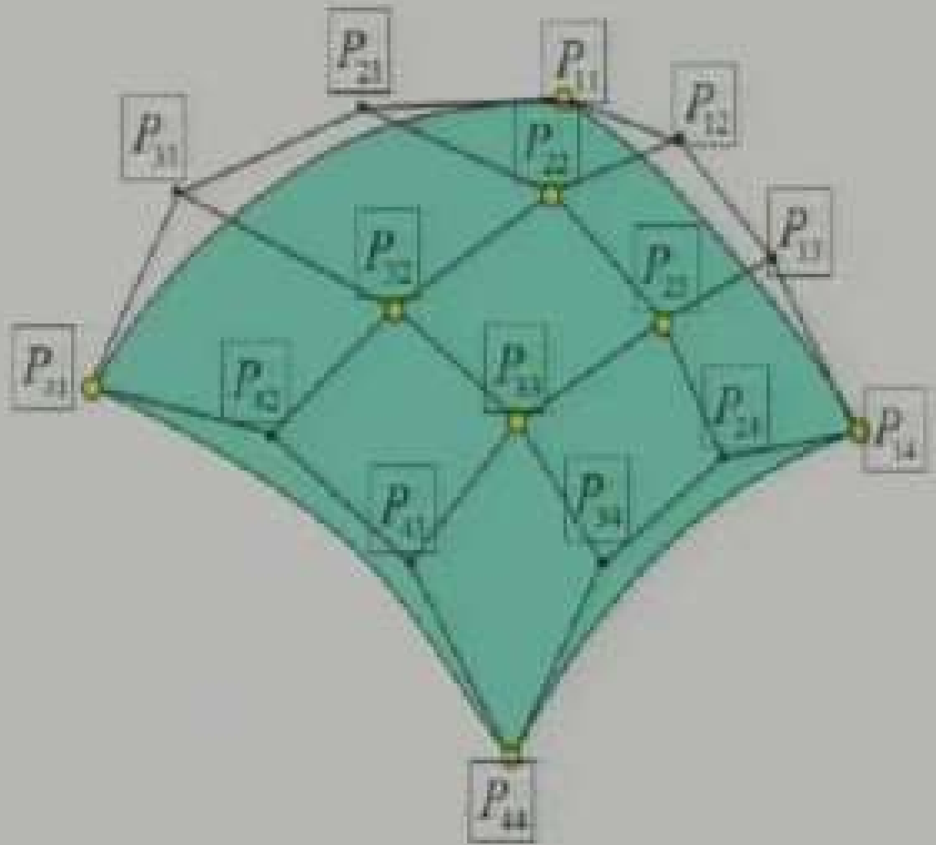
$$x(u, w) = \sum_{i=0}^{i=3} \sum_{j=0}^{j=3} {}^3c_i \, (1-u)^{3-i} \, u^i \, {}^3c_j \, (1-w)^{3-j} \, w^j \, x_{ij}$$

BICUBIC BEZIER PATCH

Sixteen Control Points

$$\begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{pmatrix}$$

BICUBIC BEZIER PATCH



B-SPLINE SURFACE PATCH

B-Spline Curve

$$x(u) = \sum_{i=0}^{i=n} N_{i,k}(u)\, x_i$$

B-Spline Patch

$$x(u,w) = \sum_{i=0}^{i=m} \sum_{j=0}^{j=n} N_{i,k}(u)\, N_{j,l}(w)\, x_{ij}$$

NURBS PATCH

$$x(u,w) = \frac{\sum_{i=0}^{i=m} \sum_{j=0}^{j=n} h_i h_j N_{i,k}(u) N_{j,l}(w) x_{ij}}{\sum_{i=0}^{i=m} \sum_{j=0}^{j=n} h_i h_j N_{i,k}(u) N_{j,l}(w)}$$

PARAMETRIC SURFACES

Parametric Surfaces Defined by Boundary Curves

- Coons Surface Patch (four boundary curves)
- Ferguson Patch (four boundary curves)
- Ruled Surface (two boundary curves)
- Bilinear Surface (four corner points)

RULED SURFACE

Input

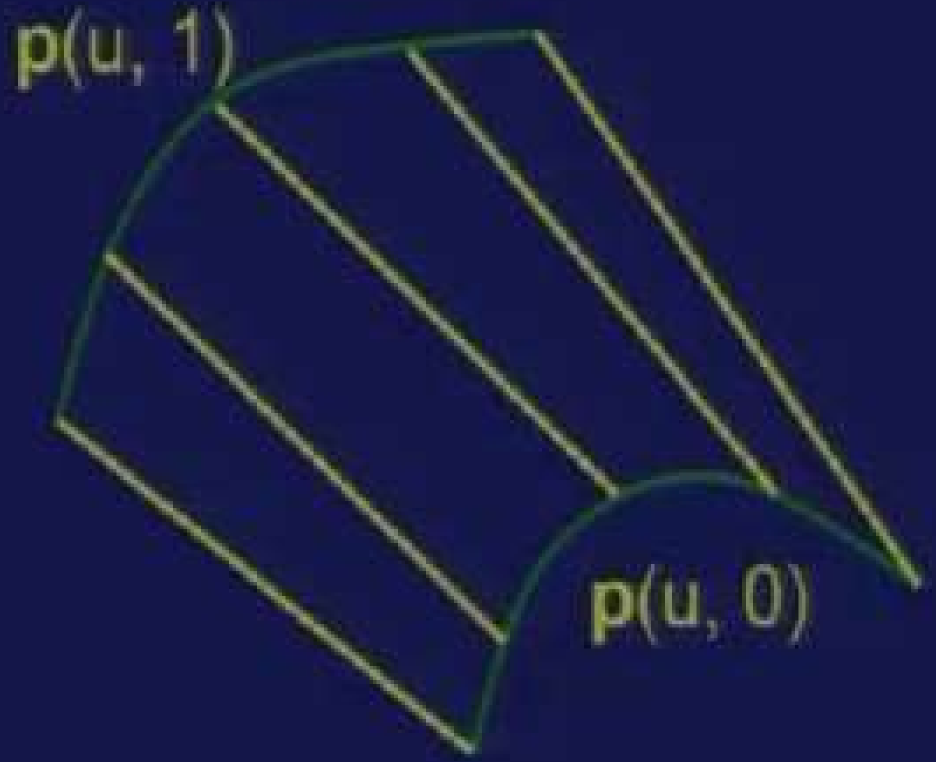
Two curves $\mathbf{p}(u, 0)$, $\mathbf{p}(u, 1)$

$$\mathbf{p}(u, w) = (1-w)\,\mathbf{p}(u, 0) + w\,\mathbf{p}(u, 1)$$

Alternate

Line and directrix curve

RULED SURFACE



Module 4

Solid Modeling

Basics of geometric and solid modeling

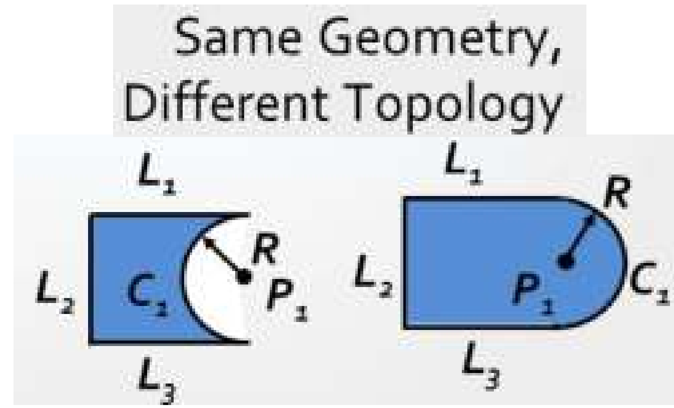
Geometric modeling refers to computer compactible mathematical representation of geometry.

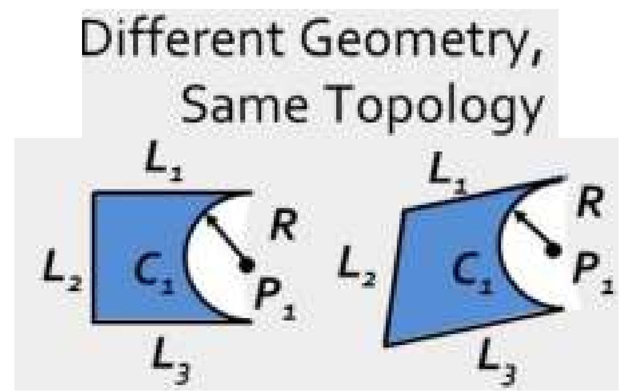
Computer representation of the geometry of a component using a software
Image can be displayed and manipulated through graphics terminal

- 3 types of commands to construct graphical image on CRT:
    I. First type: generates basic geometry elements: point, line, circle etc..
    II. Second: to accomplish scaling, rotation or other transformations on basic elements.
    III. Third: causes elements to join to take desired shape of the object created on ICG
- During GM, computer converts commands into a **mathematical model,** stores in the data file and displays it as an image on CRT screen.
- **Types of Geometric Modelling**:
    - Wireframe Model
    - Surface Model
    - Solid Model

Requirements of Geometric Modelling

- Complete part representation including topological and geometrical data.
  - Geometry: shape and dimensions
  - Topology: the connectivity and associativity of the object entities; it determines the relational information between object entities
- Able to transfer data directly from CAD to CAE and CAM.
- Support various engineering applications, including Mass property analysis, FEA etc.

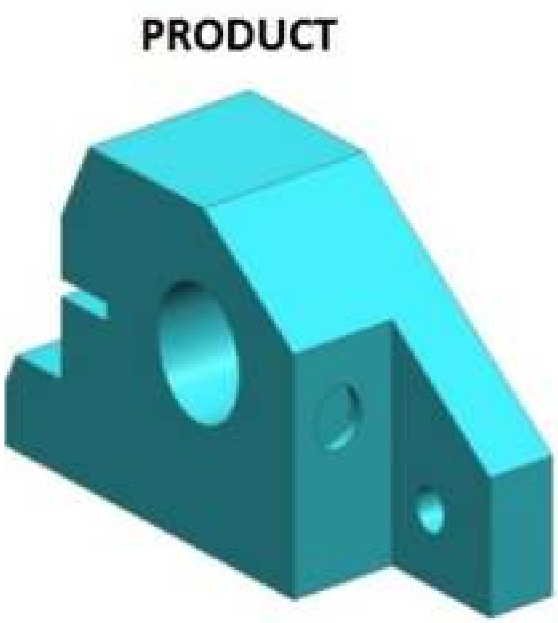


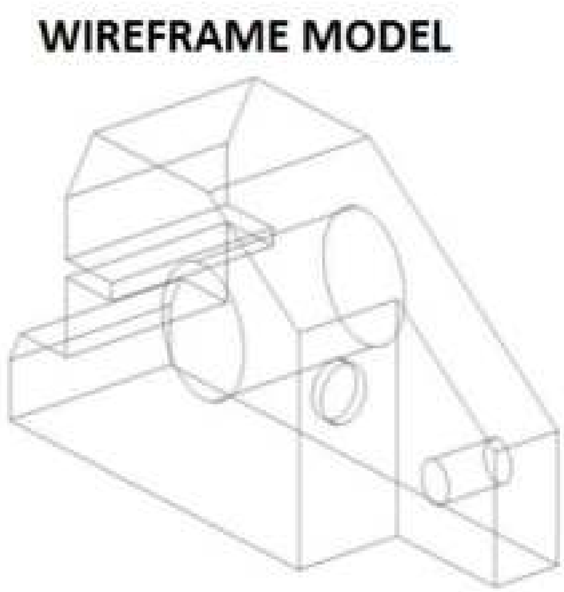


Wireframe Modelling

Object is represented by its edges. The object appears as if it is made out of thin wires.

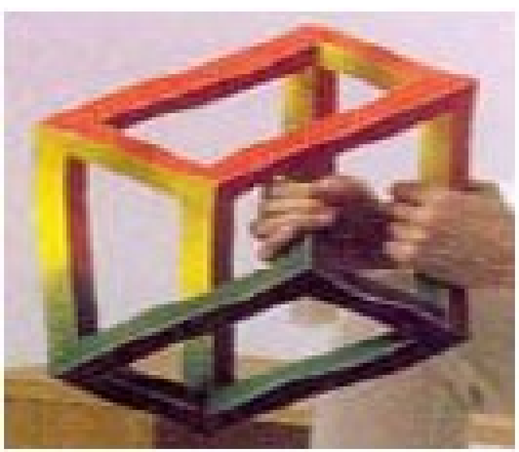
- In initial stages, wire frame models were in 2-D and 2 ½ D. Subsequently 3-D wire frame modeling software was introduced.

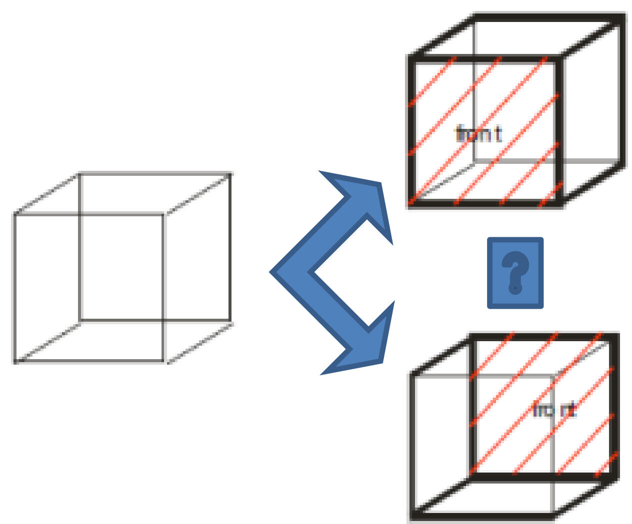




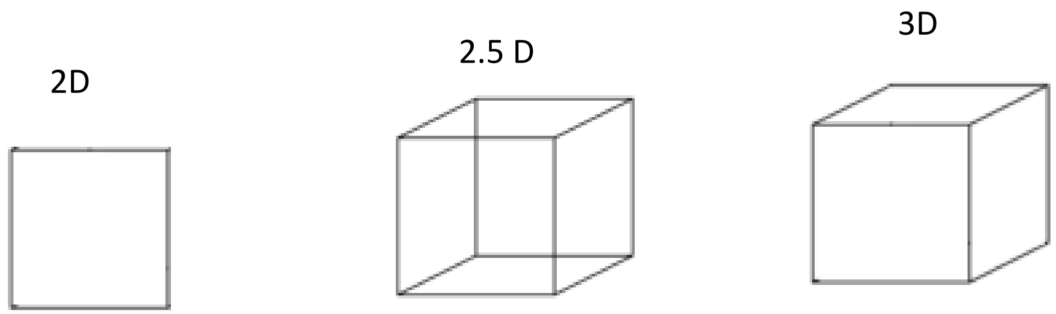
Wireframe Modelling

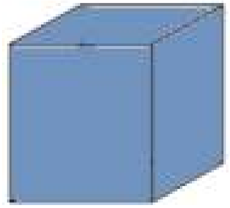
- Developed in 1960s and referred as "a stick figure" or "an edge representation"
- The word "wireframe" is related to the fact that *one may imagine a wire that is bent to follow the object edges to generate a model*.
- Model consists entirely of points, lines, arcs and circles, conics, and curves.



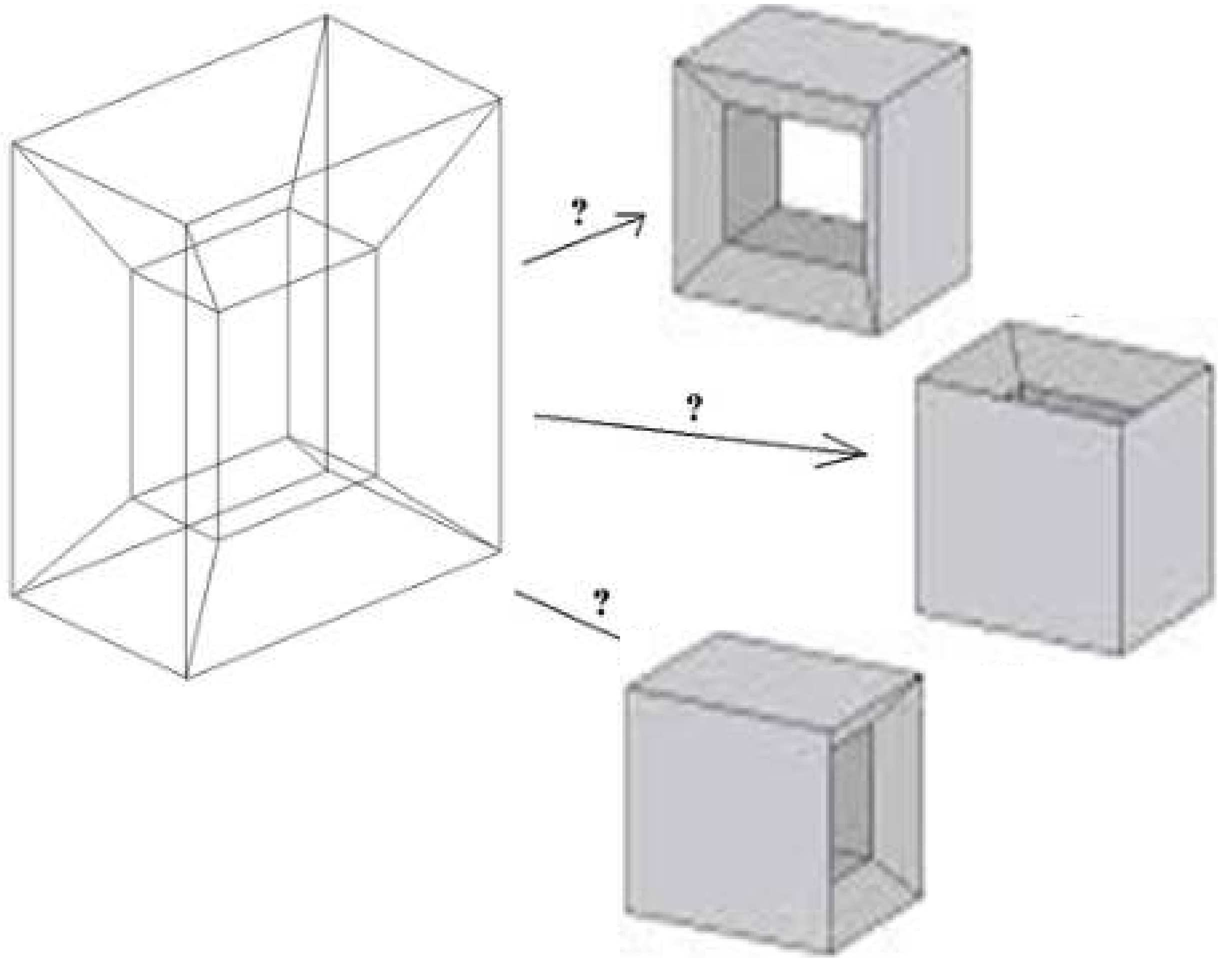


Wireframe Modelling

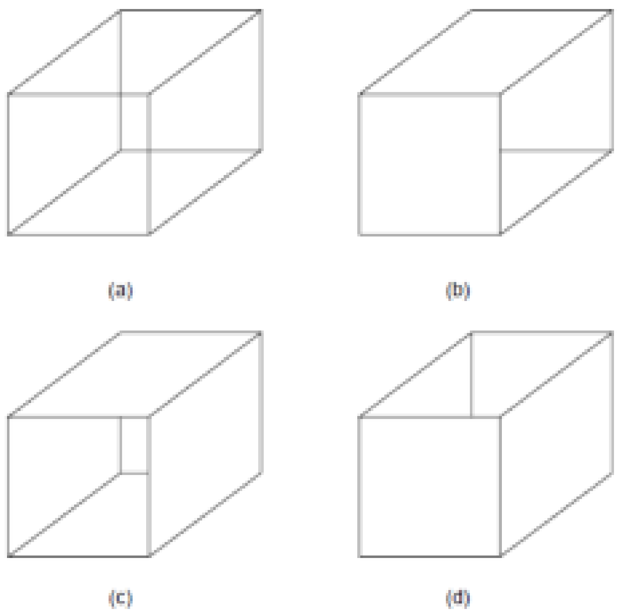




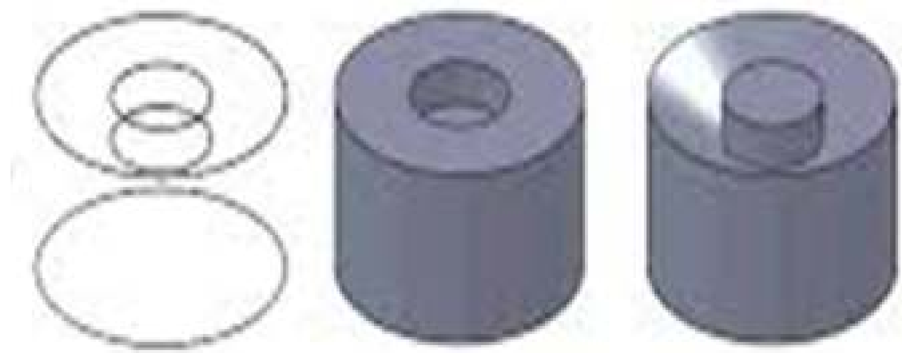
- In 3D wireframe model, an object is not recorded as a solid.
- Instead the vertices that define the boundary of the object, or the intersections of the edges of the object boundary are recorded as a collection of points and their connectivity.



Wireframe Modelling



Ambiguity in Wire Frame Modeling



Wireframe Modelling

Advantages

- Simple to construct
- Does not require as much as computer time and memory as does surface or solid modeling (manufacturing display)
- As a natural extension of drafting, it does not require extensive training of users.
- Form the basis for surface modeling as most surface algorithms require wireframe entities (such as points, lines and curves)

Disadvantages

- Ambiguous
- The input time is substantial and increases rapidly with the complexity of the object
- Both topological and geometric data need to be user-input; while solid modeling requires only the input of geometric data.
- Unless the object is two-and-a-half dimensional, volume and mass properties, NC tool path generation, cross-sectioning and interference cannot be calculated.

Surface Modelling

- A component is represented by its surfaces which in turn are represented by their vertices and edges.
- Surface models take the modeling of an object one step beyond wireframe models by providing information on the surface connecting the object edges.
- A surface model consists of wireframe entities that form the basis to create surface entities.
- Surface modeling :useful in the development of manufacturing codes for ***automobile panels*** and the complex doubly curved shapes of ***aerospace structures and dies and moulds***.

Surface Representation

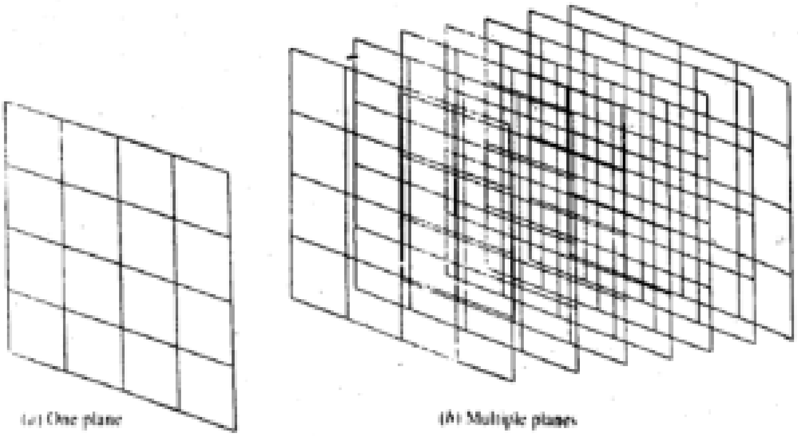
Surface entities

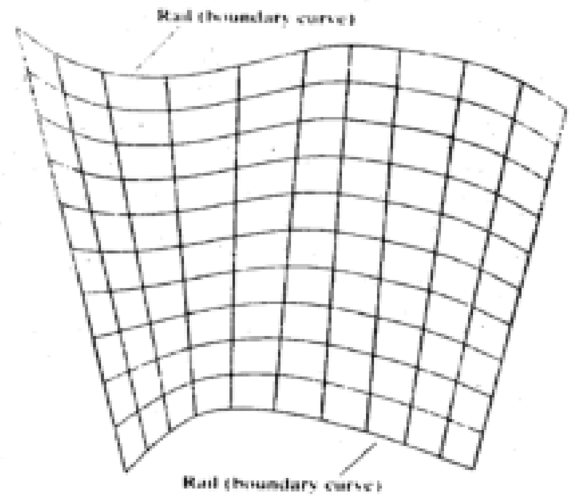
1. Analytic entities

– Includes – Plane surface, ruled surface, surface of revolution and tabulated cylinder.

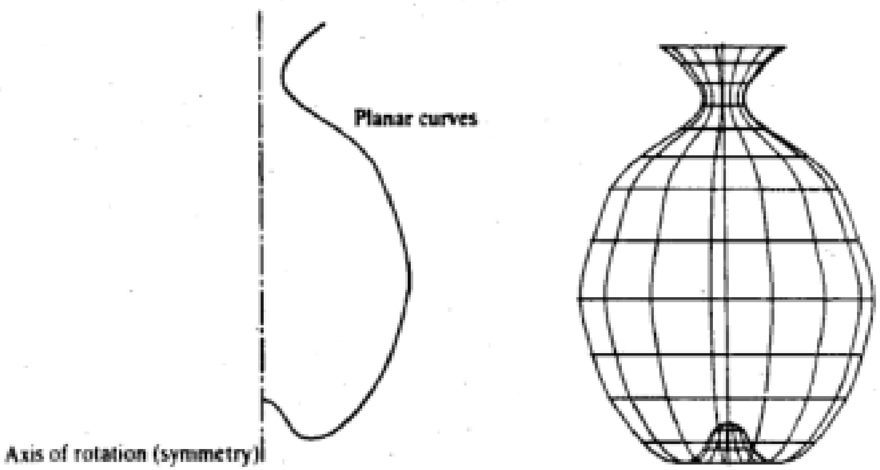
2. Synthetic

– Includes – Bicubic, Hermite spline surface, B – Spline surface, rectangular and triangular Bezier patches, rectangular and triangular Coons patches and Gordon surface

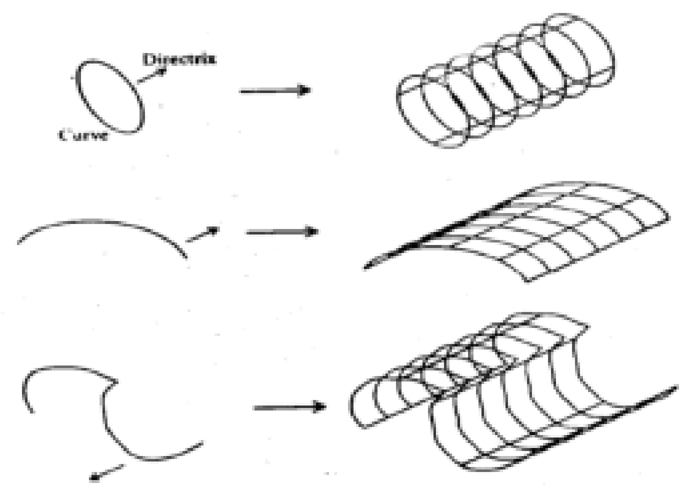




Ruled surface

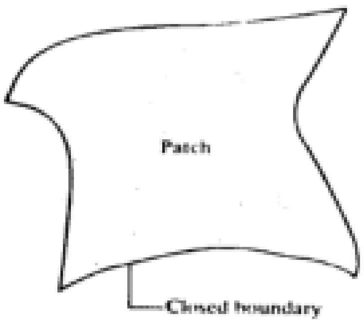


Surface of revolution

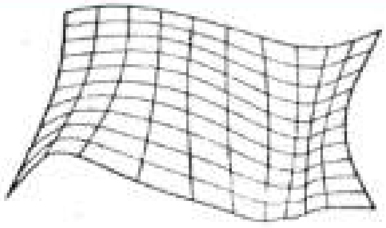


Tabulated cylinder

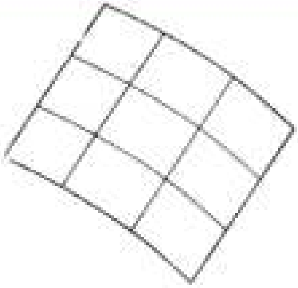
Coons patch



B-spline surface



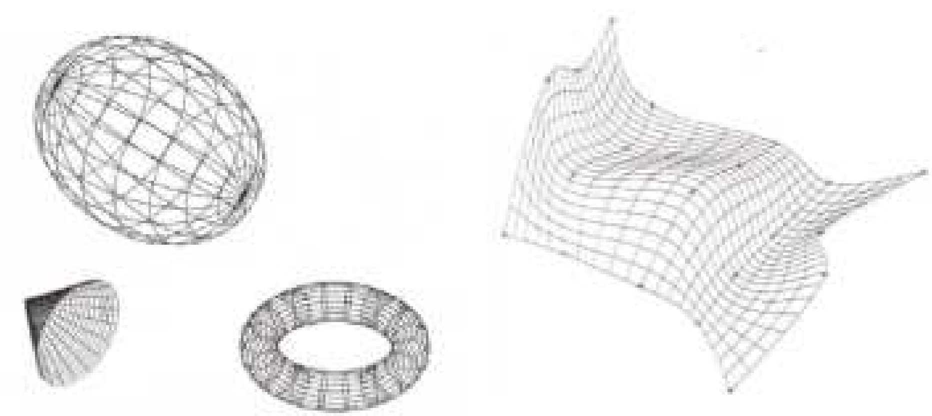
Bezier surface



Surface Modelling

- In general, a wireframe model can be extracted from a surface model by deleting or blanking all surface entities
- Shape design and representation of complex objects such as car, ship, and airplane bodies as well as castings

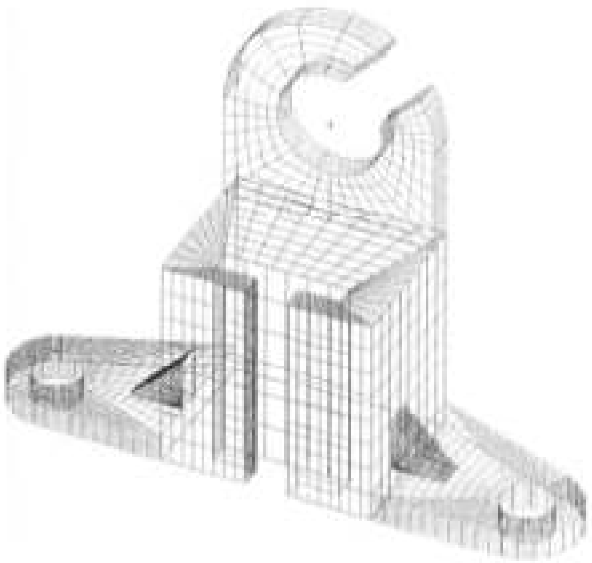
Examples of Surface Models

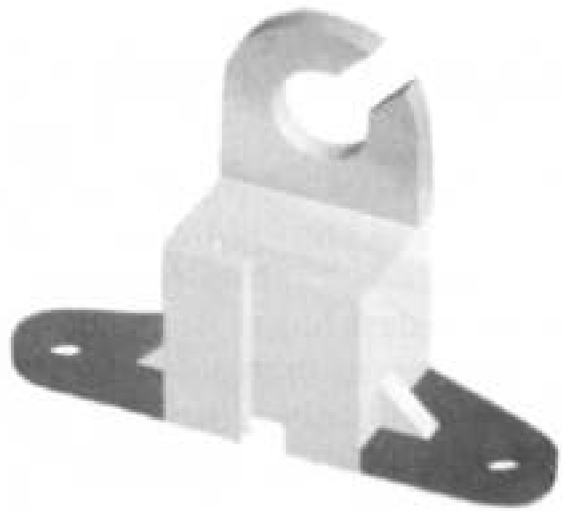


Analytical Surfaces

Free-form, Curved, or Sculptured Surface

- Surface models define only the geometry, no topology.
- Shading is possible





Shading - by interpreting the polygons'
- Direction (normal)
- Spatial order

Surface Modelling

Advantages:

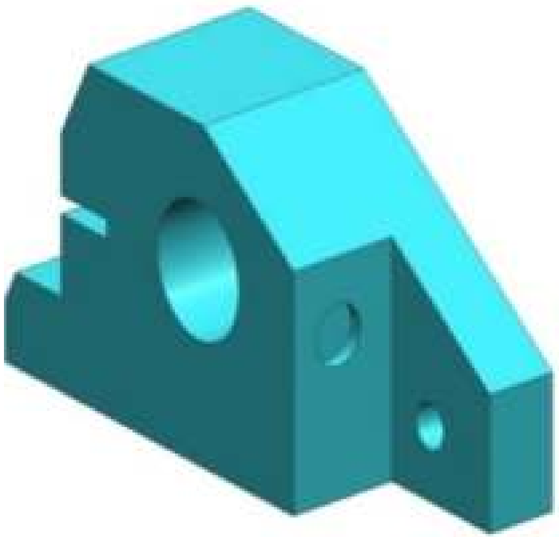
- Less ambiguous than wire frame
- Provide hidden line and surface algorithms to add realism to the displayed geometry
- Support shading
- Support volume and mass calculation, finite element modeling, NC path generation, cross sectioning, and interference detection. (when complete)

Disadvantages

- Require more training and mathematical background of the users
- Require more CPU time and memory
- Still ambiguous; no topological information
- Awkward to construct

Solid Modelling

- Models are displayed as solid objects to the viewer in 3D, with very little risk of misinterpretation.
- When color is added to the image, resulting image will be more realistic.
- Store both **geometric and topological information;** can verify whether two objects occupy the same space
- Solid models are,
  - Bounded
  - Homogeneous and finite



Modelling packages includes three packages,

- **Constructive solid geometry (CSG or C-Rep):**

  In a CSG, physical objects are created by combining elementary shapes known as primitives like blocks, cylinders, cones, pyramids and spheres. The Boolean operations like union (∪), difference (–) and intersection (∩) are used to carry out this task.

- **Boundary representation (B-Rep):**

  The solid is represented by its boundary which consists of a set of faces, a set of edges and a set of vertices as well as their topological relations.

- **Sweep Representation**

- **Spatial occupancy enumeration**

- **Cell decomposition**

Why Solid Modelling

Solid Modeling Supports,

Use of volume information

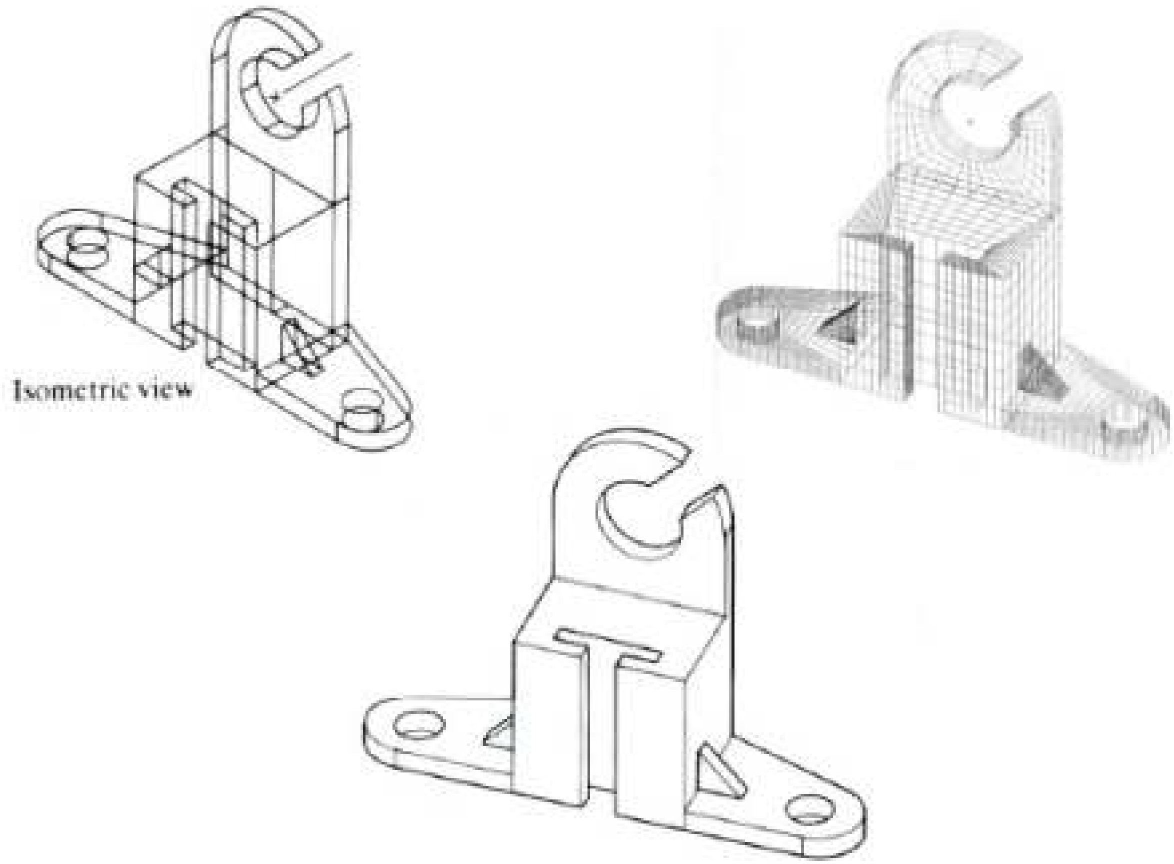
– Weight or volume calculation, centroids, moments of inertia calculation,

– Stress analysis (finite elements analysis), heat conduction calculations, dynamic analysis,

– System dynamics analysis

Use of volume and boundary information

– Generation of CNC codes, and robotic and assembly simulation



Information complete, unambiguous, accurate **solid model**

Some Solid Modelers in Practice

Modeler	Developer	Primary Scheme	User Input
CATIA	IBM	CSG	BREP+CSG
GEOMOD / I-DEAS	SDRC/EDS	BREP	BREP+CSG
PATRAN-G	PDA ENGG.	ASM	HYPERPATCHES+CSG
PADL-2	CORNELL UNI.	CSG	CSG
SOLIDESIGN	COMPUTER VISION	BREP	BREP+CSG
UNISOLIDS / UNIGRAPHICS	McDONELL DOUGLAS	CSG	BREP+CSG
PRO-E	PARAMETRIC	BREP	BREP+CSG
SOL. MOD. SYS	INTERGRAPH	BREP	BREP+CSG

Solid modeling

- **Weakness of wireframe and surface modeling**
  - Ambiguous geometric description
  - incomplete geometric description
  - lack topological information
  - Tedious modeling process
  - Awkward user interface

Solid model

- Solid modeling is based on *complete, valid and unambiguous* geometric representation of physical object.

  – Complete → points in space can be classified.(inside/ outside)
  – Valid →vertices, edges, faces are connected properly.
  – Unambiguous → there can only be one interpretation of object

Advantages of Solid Models

Unlike wireframes and surface representations which contain only geometrical data, the solid model uses topological information in addition to the geometrical information to represent the object unambiguously and completely. Solid model results in accurate design, helps to further the goal of CAD/ CAM like CIM, Flexible manufacturing leading to better automation of the manufacturing process.

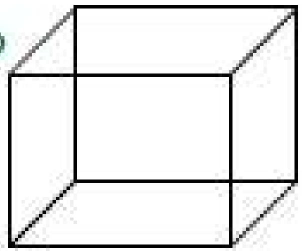
Geometry: The graphical information of dimension, length, angle, area and transformations

Topology: The invisible information about the connectivity, neighborhood, associatively etc

Is a solid model just a shaded image?

Three dimensional addressability?

Suitable for automation?



Wireframe Model

Solid Model

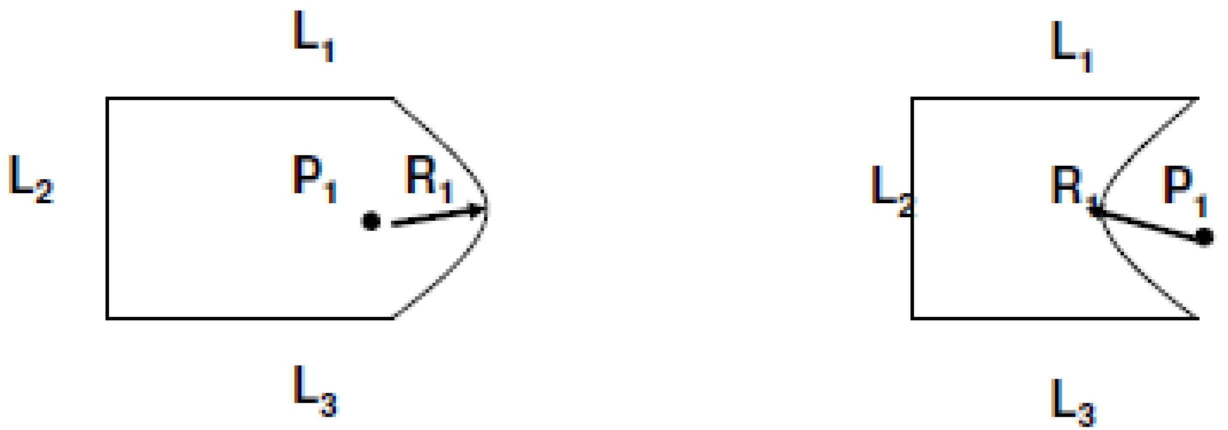
Geometry Vs Topology

Geometry:

Metrics and dimensions of the solid object. Location of the object in a chosen coordinate system

Topology:

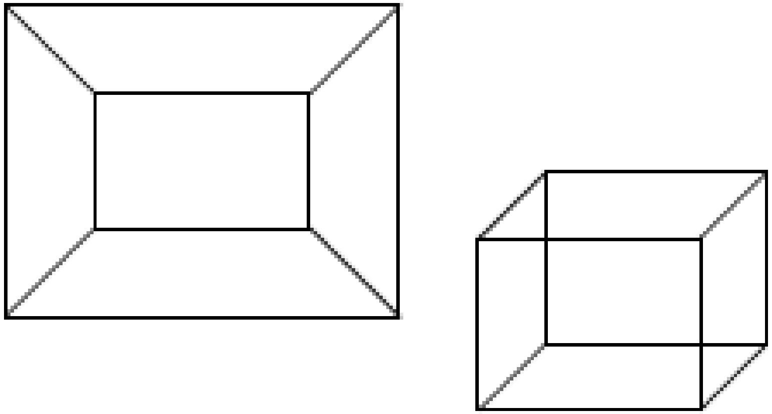
Combinatorial information like connectivity, associativity and neighborhood information. Invisible relationship information.



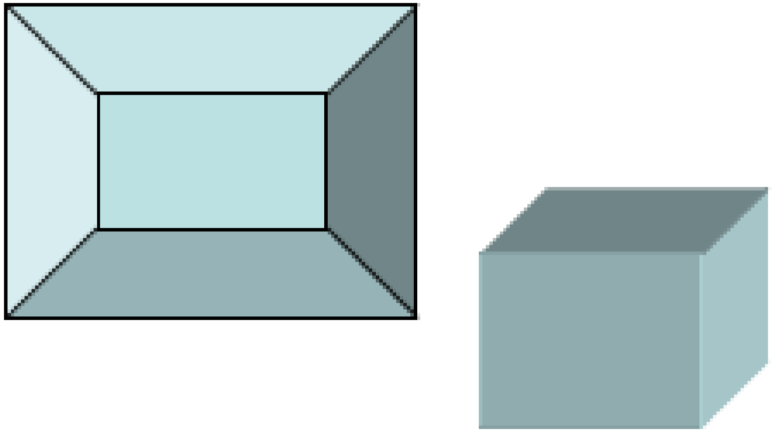
Same Topology and different geometry

Definition of a Solid Model

A solid model of an object is a more complete representation than its surface (wireframe) model. It provides more topological information in addition to the geometrical information which helps to represent the solid unambiguously.

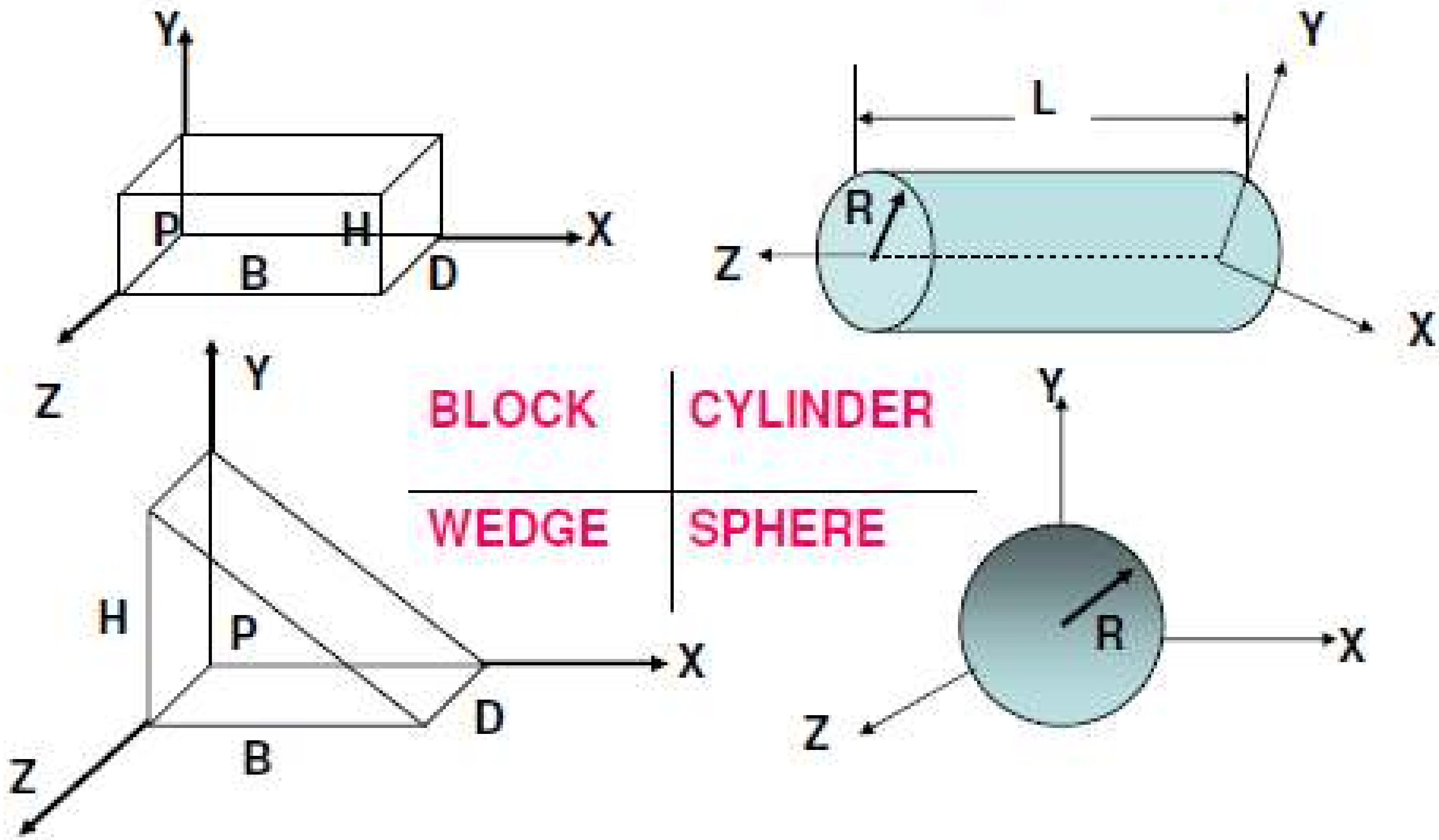


Wireframe Model



Solid Model

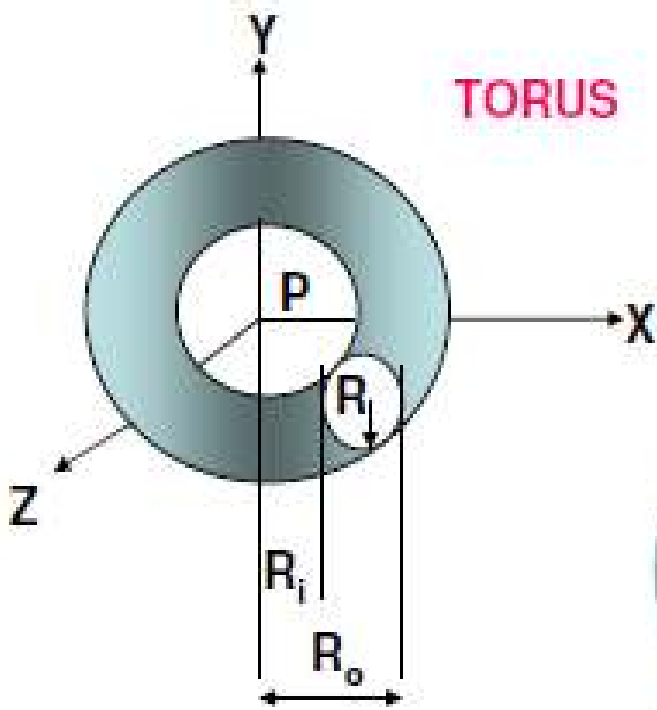
Solid Primitives: Ready starting objects

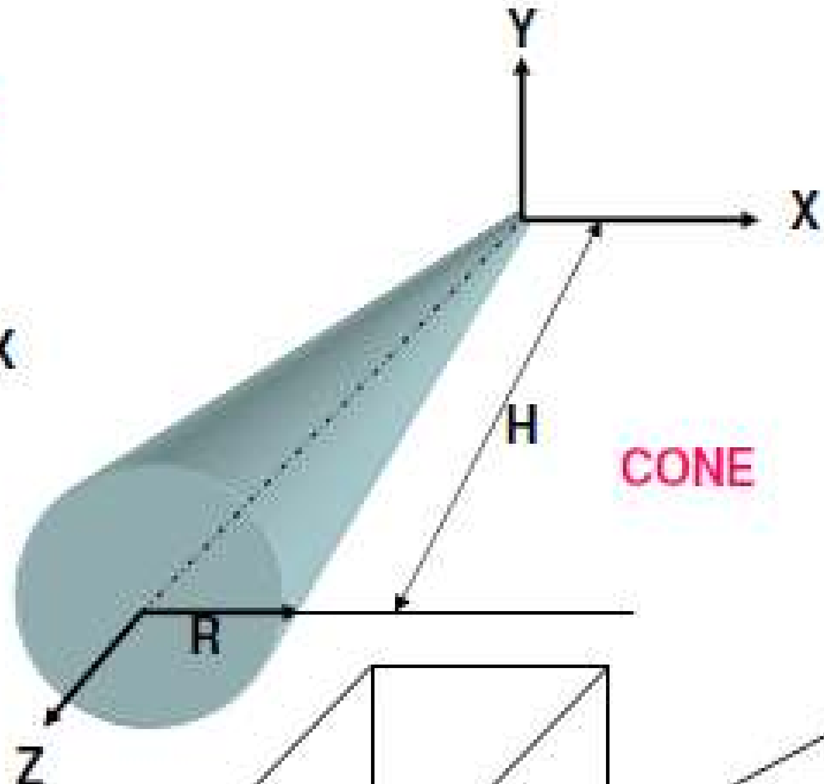


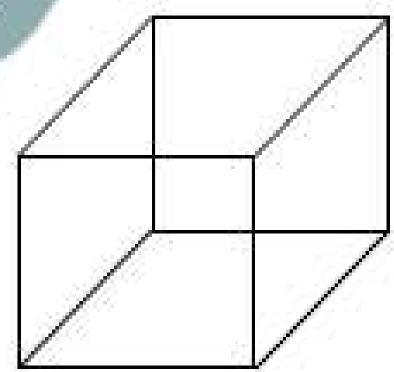
Origin of MCS: P   Orientation   Topology and shape

Dimension (size) L,B,H.D,R   Perspective on/off

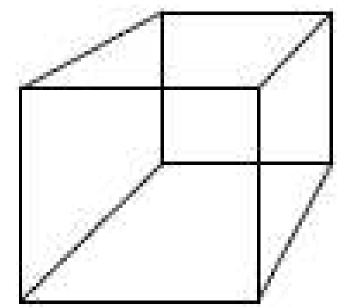
Solid Primitives: Ready starting objects







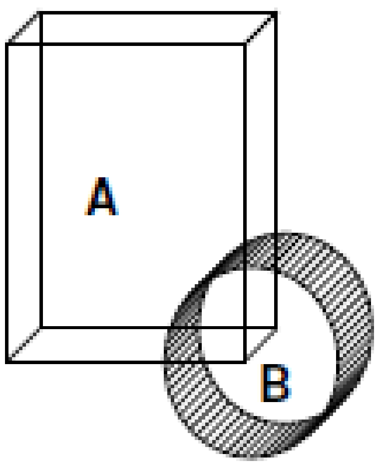
PERSPECTIVE OFF

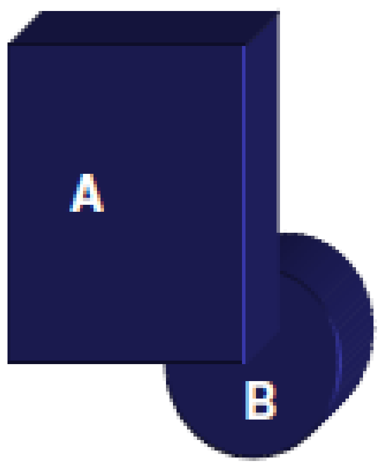


PERSPECTIVE ON

Operation on Primitives

- A desired solid can be obtained by combining two or more solids
- When we use Boolean (set) operations the validity of the third (rusulting) solid is ensured





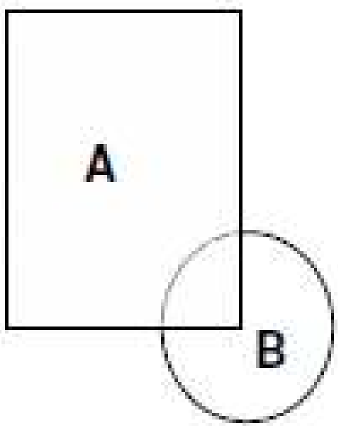
3 Dimensional

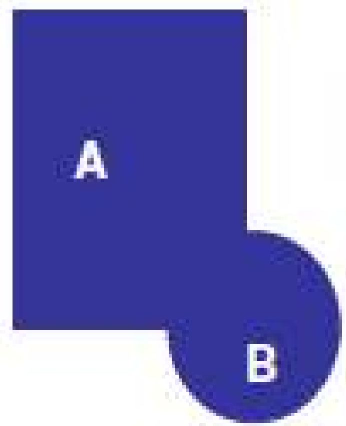
UNION: BLOCK ∪ CYLINDER

$A \cup B$

Operation on Primitives

- A desired solid can be obtained by combining two or more solids
- When we use Boolean (set) operations the validity of the third (rusulting) solid is ensured

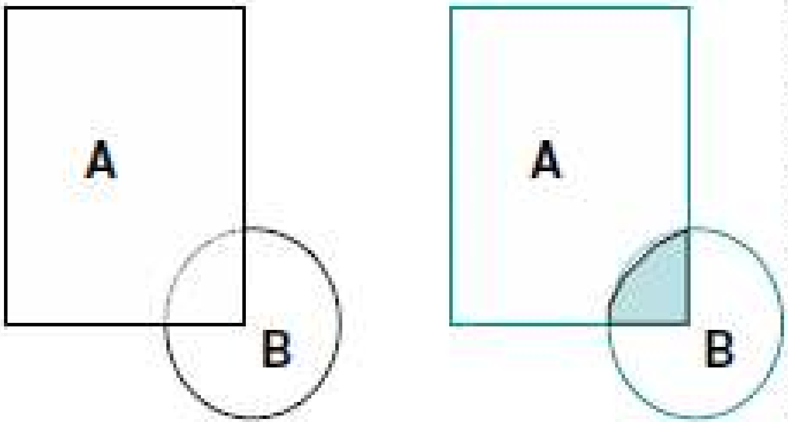




UNION: BLOCK ∪ CYLINDER

$A \cup B$

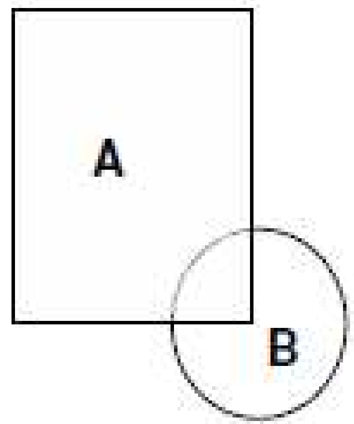
Operation on Primitives

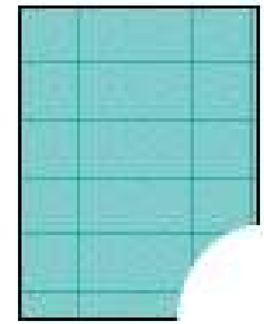


INTERSECTION:

BLOCK ∩ CYLINDER

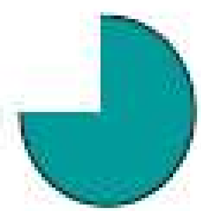
$A \cap B$





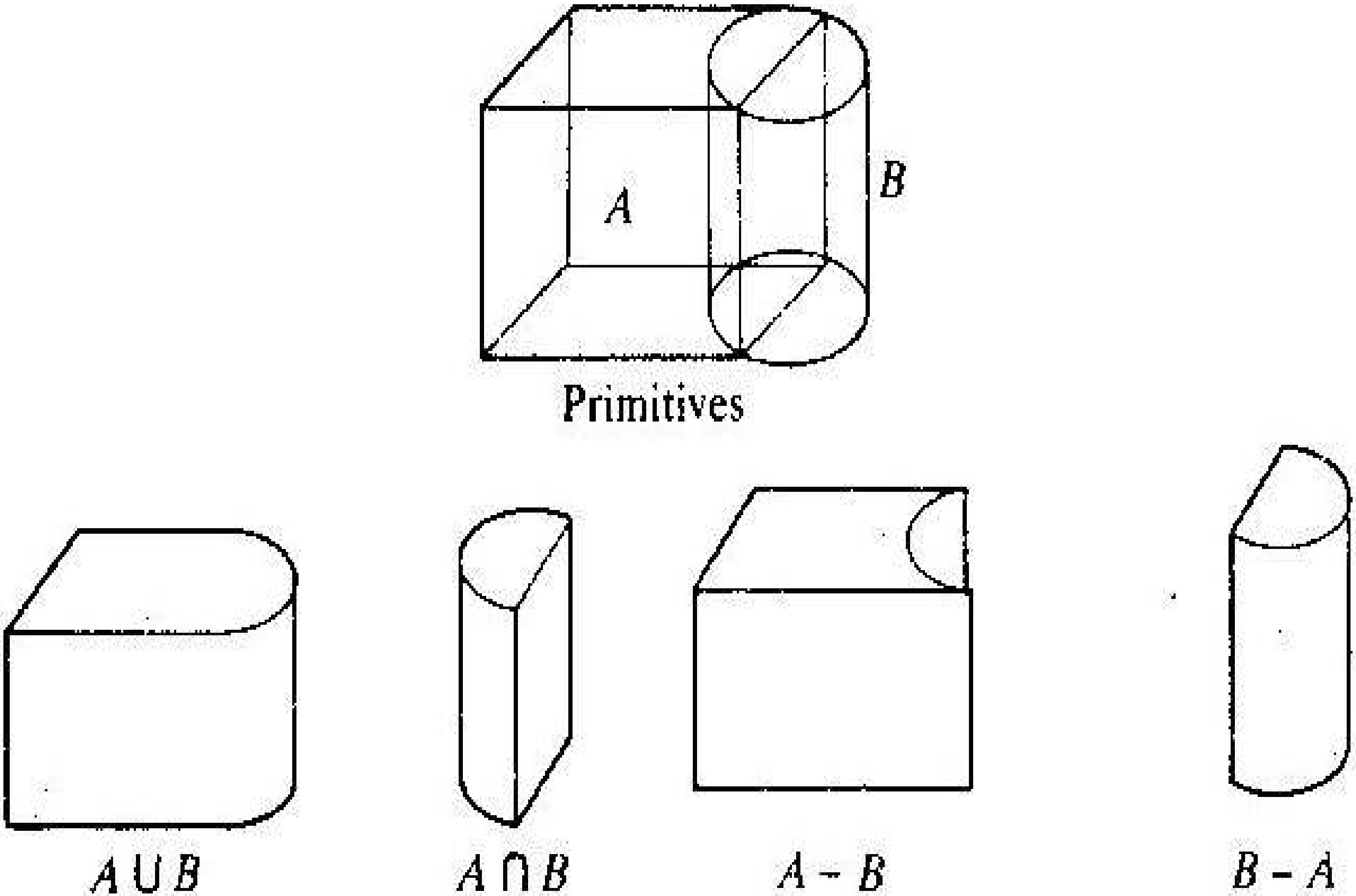
A - B

B - A



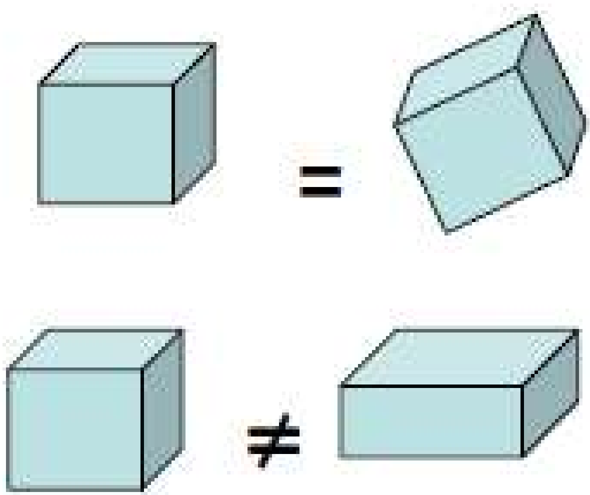
DIFFERENCE:

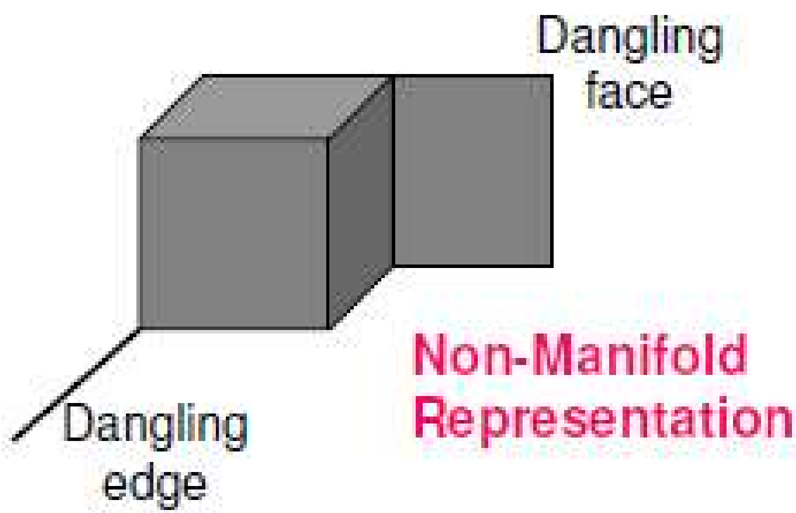
BLOCK - CYLINDER


A
B
Primitives
A ∪ B
A ∩ B
A − B
B − A

Properties of Solid Models

- **Rigidity:** Shape of the solid is invariant w.r.t location/orientation
- **Homogeneous 3-dimensionality**: The solid boundaries must be in contact with the interior. No isolated and dangling edges are permitted.





Properties of Solid Models

- **Finiteness and finite describability:** Size is finite and a finite amount of information can describe the solid model.
- **Closure under rigid motion and regularised Boolean operations**: Movement and Boolean operations should produce other valid solids.
- **Boundary determinism:** The boundary must contain the solid and hence determine the solid distinctively.
- **Any valid solid must be bounded, closed, regular and semi-analytic subsets of E^3**

Algorithms in Solid Modelers

- Three types of algorithms are in use in many solid modelers according to the kind of input and output

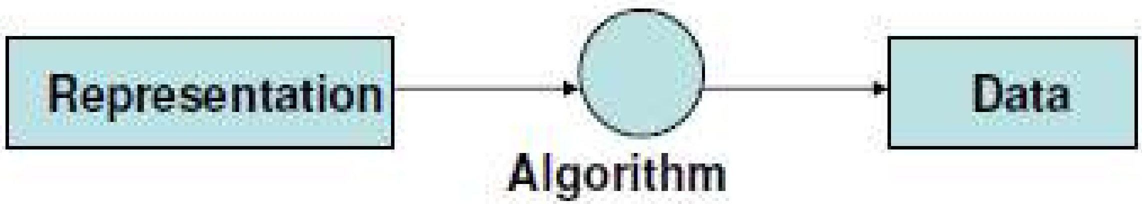
1. a; data -> representation

most common – build, maintain, manage representations

Algorithms in Solid Modelers

- Some algorithms operate on existing models to produce data.

2. a; representation -> data

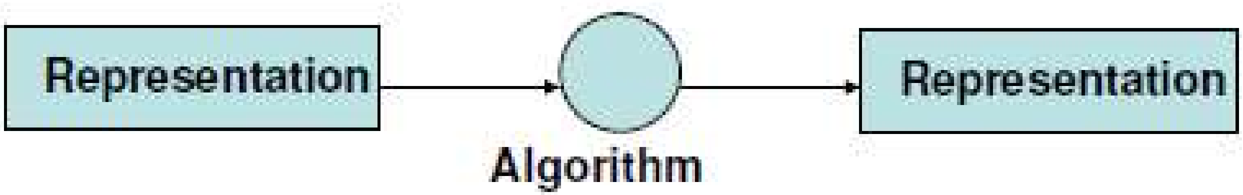


E.g.. Mass property calculation

Algorithms in Solid Modelers

- Some representations use the available representations to produce another representation.

3. a: representation -> representation



- E.g. Conversion from CSG to B-rep

Solid representation schemes

- **Constructive solid geometry (CSG)**
- **Boundary representation (B-Rep)**
- **Spatial occupancy enumeration**
- **Cell decomposition**
- **Sweep representation**

SWEEP REPRESENTATION

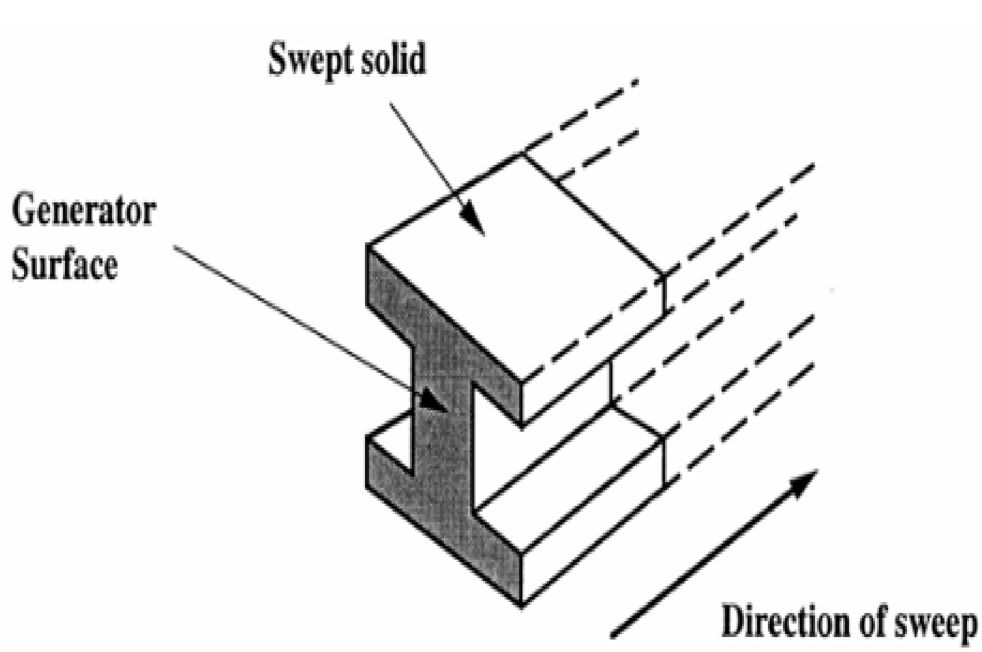
Sweep

- Useful in creating 3 D solid models that possess translational, rotational or other symmetries.
- Class includes,
  1) Solids of uniform thickness in given direction – Known as extruded solids and are created via linear or translational sweep
  2) Axisymmetric solids – Solids of revolution which can be created via rotational sweep

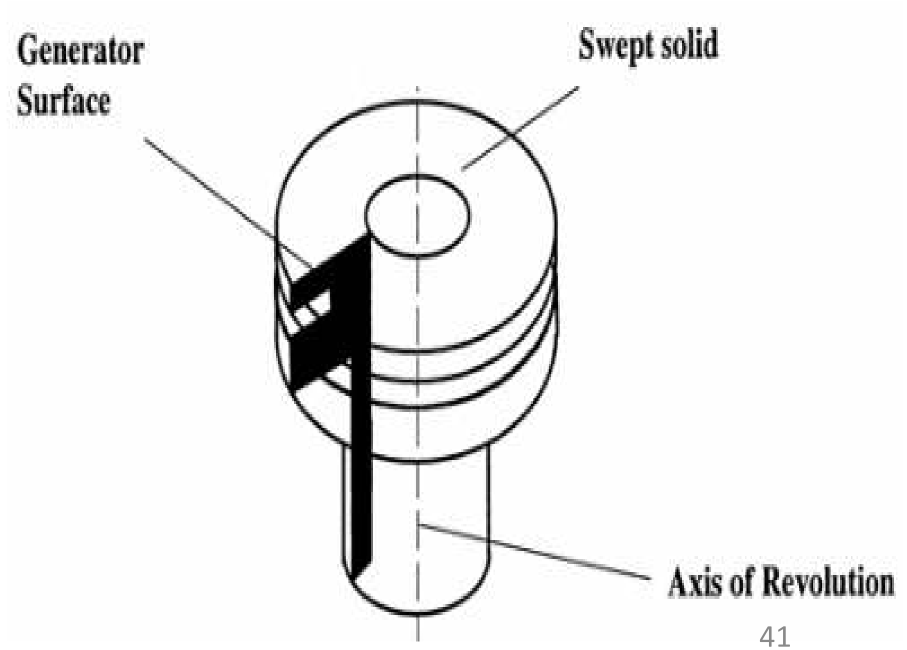
Solids that have a uniform thickness in a particular direction and axisymmetric solids can be created by what is called *Transitional (Extrusion) and Rotational (Revolution) Sweeping*

- Sweeping requires two elements – a surface to be moved and a trajectory, analytically defined, along which the movement should occur.

Extrusion



Revolution



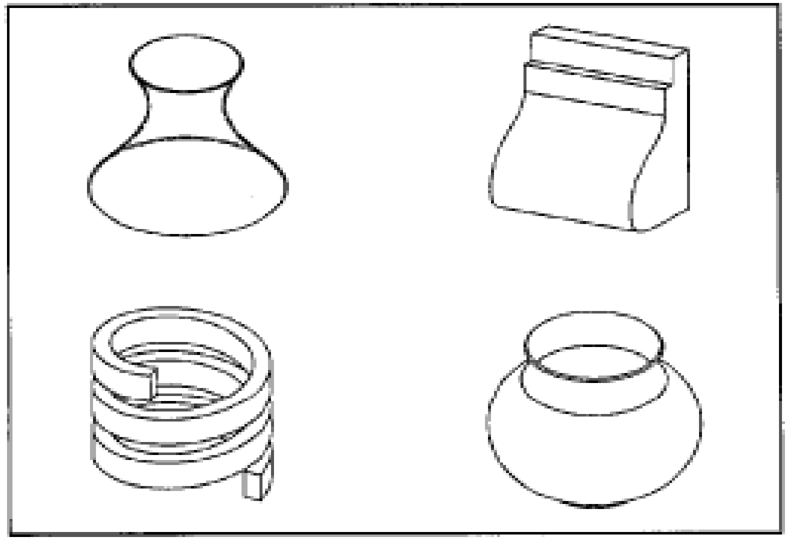
Sweeping

Sweeping can be carried out in two different forms:

- *Extrusion* - to produce an object model from a 2D cross-section shape, the direction of extrusion, and a given depth. Advanced applications include curved extrusion guideline and varying cross-sections.
- *Revolving* - to produce a rotation part, either in solid or in shell shape. Revolving a 2D cross-section that is specified by a closed curve around the axis of symmetry forms the model of an axially symmetric object.

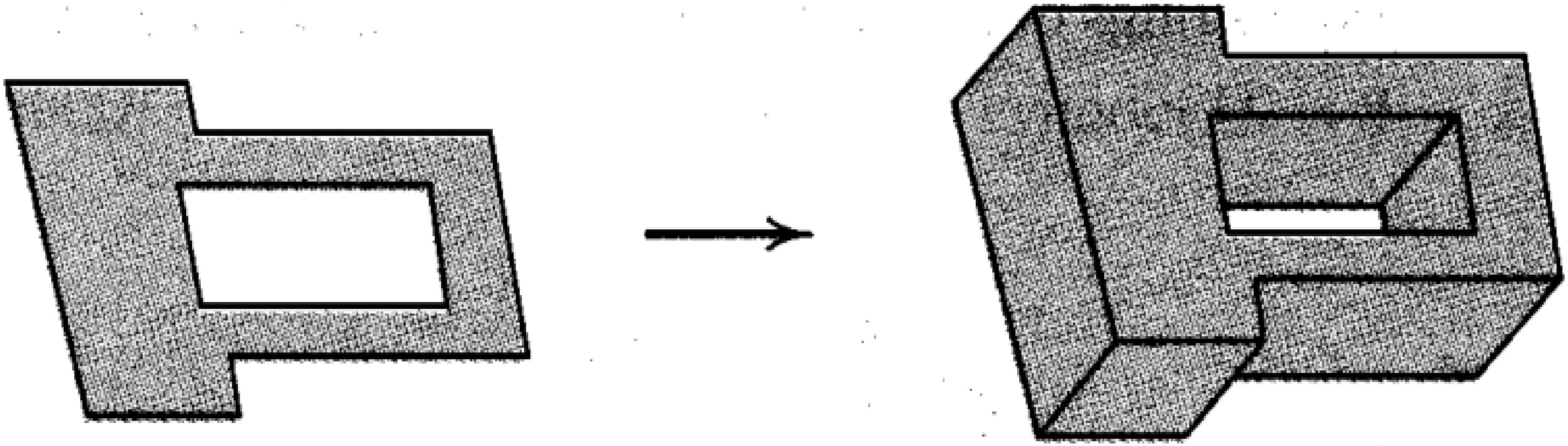
Sweeping is most convenient for solids with translational or rotational symmetry. Sweeping also has the capability to guarantee a closed object.

Advanced: spatial sweeping; & varying cross-section

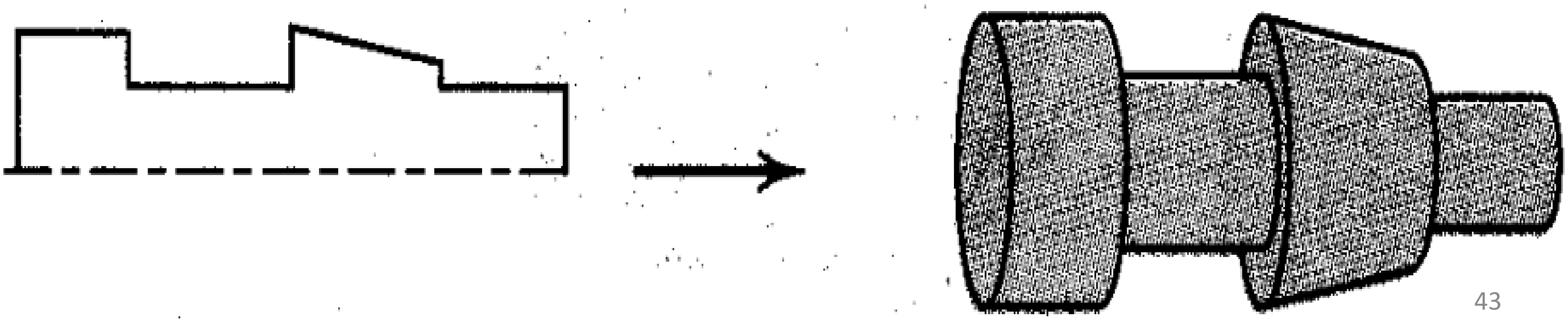


Sweep Representation

Extrusion (Transitional Sweeping)



Revolution (Rotational Sweeping)



Sweep

- Based on the notion of moving a point, curve, of a surface along a given path
- Three types,

  1) Linear

  2) Non linear and

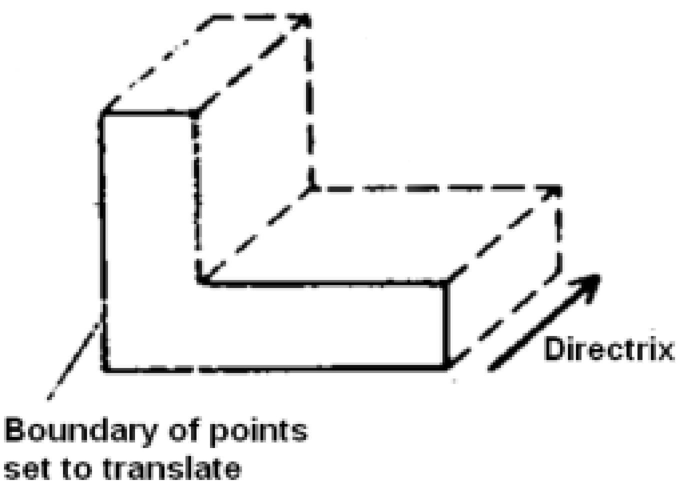
  3) Hybrid sweeps

- Linear – The path is a linear or circular vector described by a linear, most often parametric, equation
- Non-linear – The path is a curve described by a higher order equation

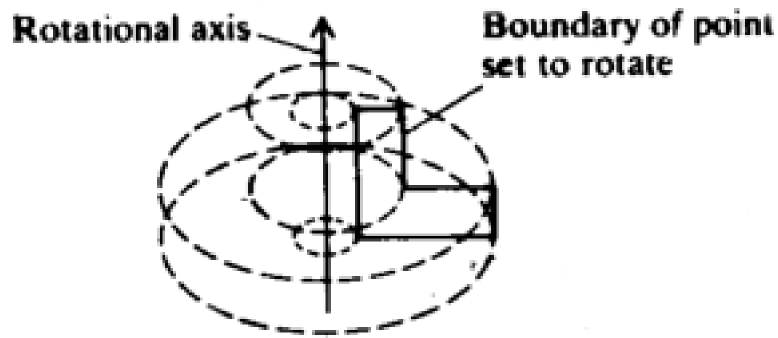
  (quadratic, cubic, or higher)

- Hybrid – Combines linear and/or non-linear sweep via set operations

- Linear divided further into
  1) Translational and
  2) Rotational

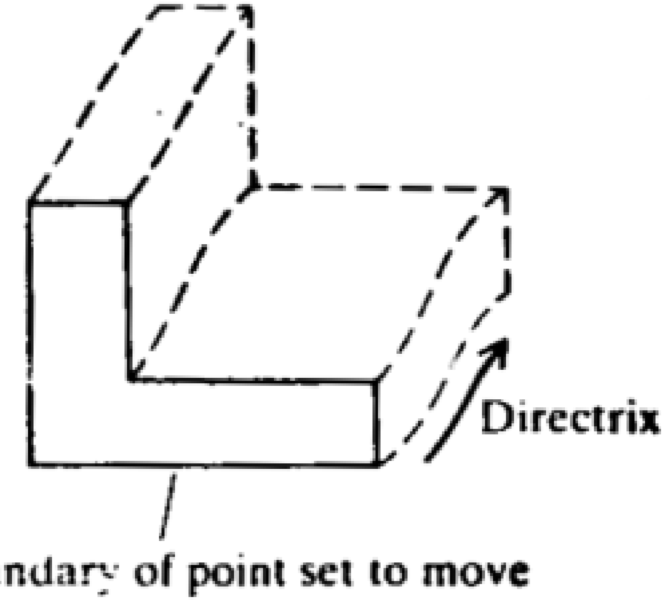


Translational sweep

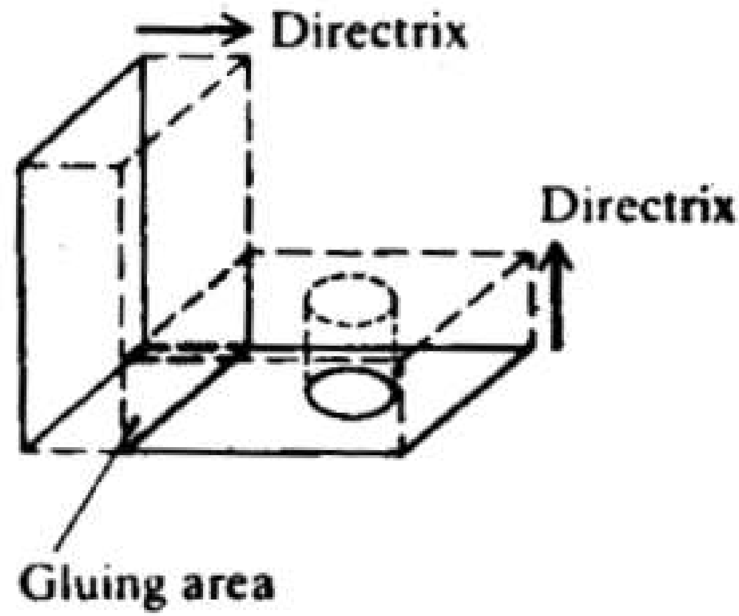


Rotational sweep

- **Directrix being a** curve

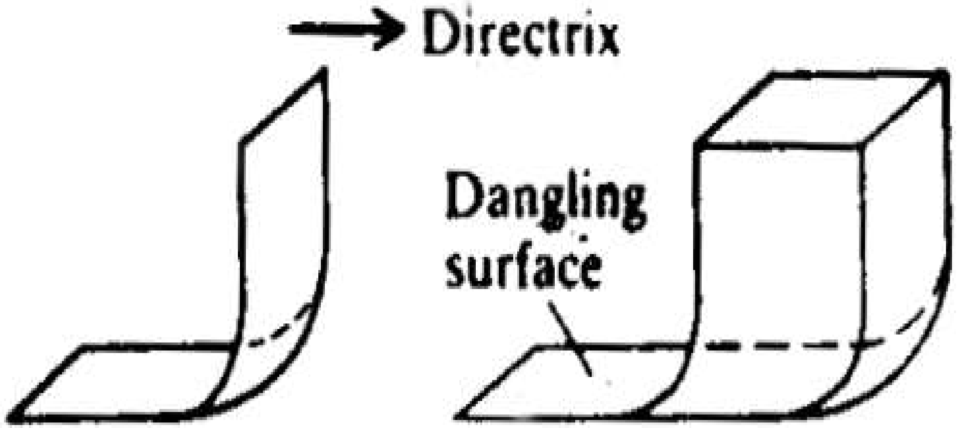


(b) Nonlinear sweep



(c) Hybrid sweep

Invalid sweep



(d) Invalid sweep

- **If sweeping direction is not chosen properly.**

- Basic element – Wire frame curves, both analytic and synthetic

- Building operation
  - For linear and non linear
    - » Generate the boundary and sweep it
  - For hybrid
    - » These operation extend to include Boolean operations

- One linear sweep can be converted to a B-rep or CSG
- Linear sweep to CSG conversion must be based on unbounded CSG primitives

Constructive Solid Geometry, CSG

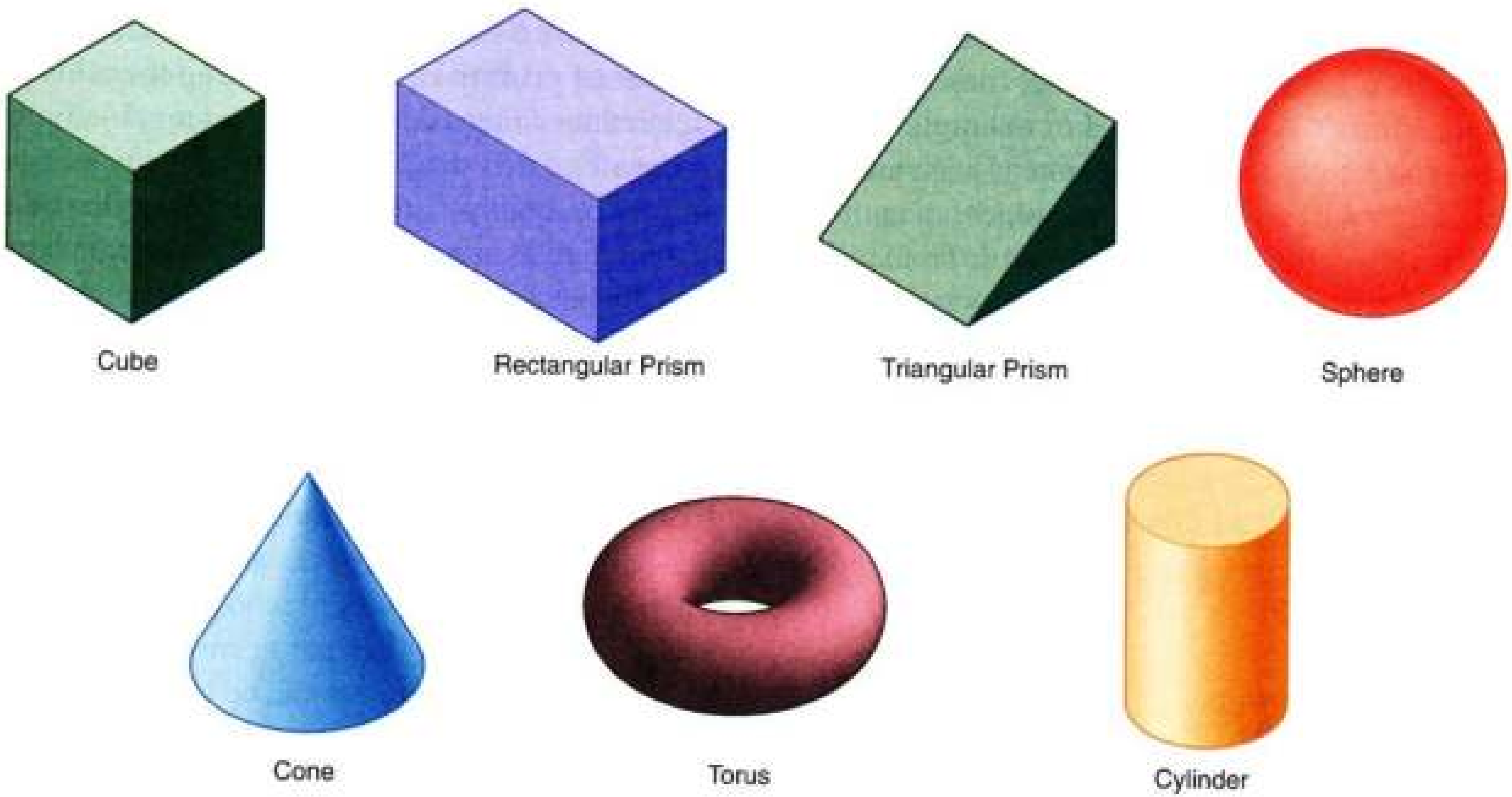
Constructive Solid Geometry, CSG

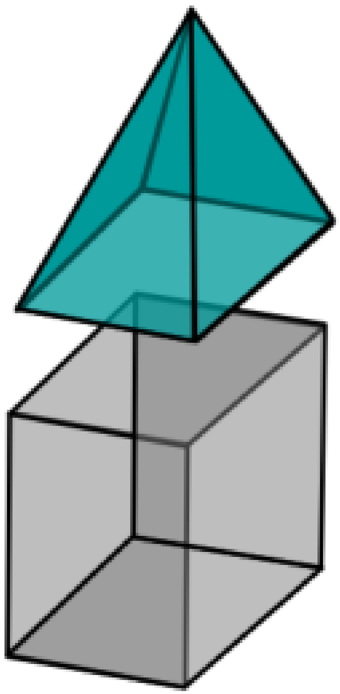
- This is a solid modeling method that combines simple solid primitives to build more complex models using **Boolean operators: union, difference and intersection.**

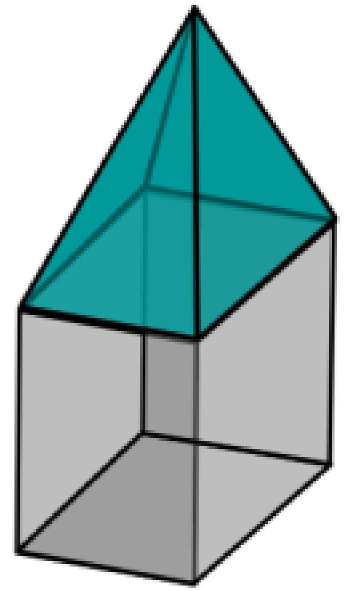
- The resulting model is a procedural model stored in the mathematical form of a binary tree where leaf nodes are solid primitives, correctly sized and positioned, and each branch node is a Boolean operator.

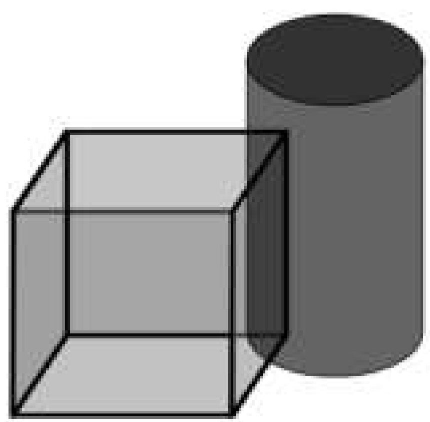
Primitive Solids and Boolean Operations

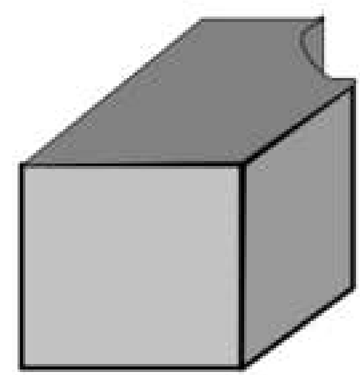
The basic primitive solid:







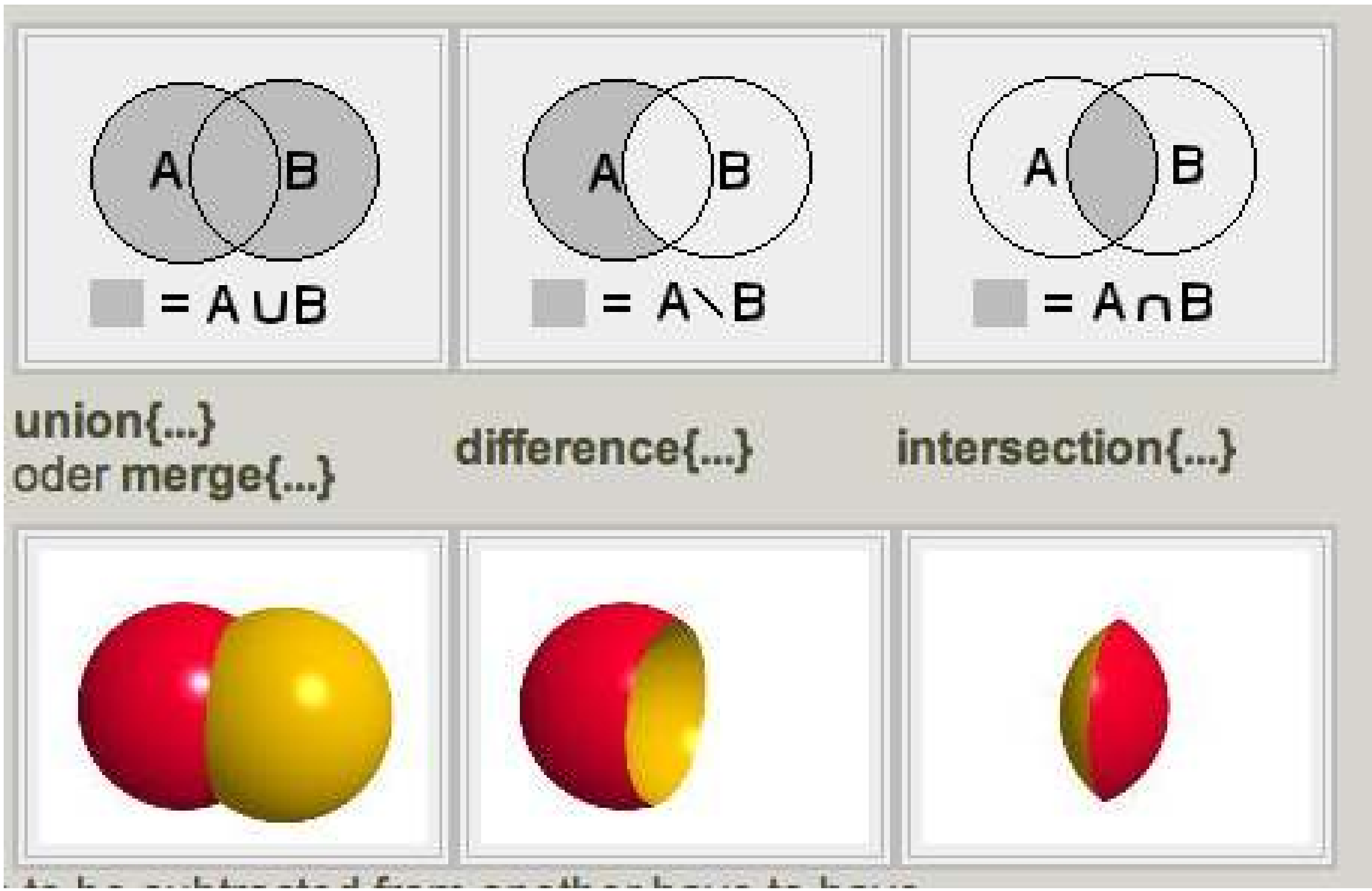




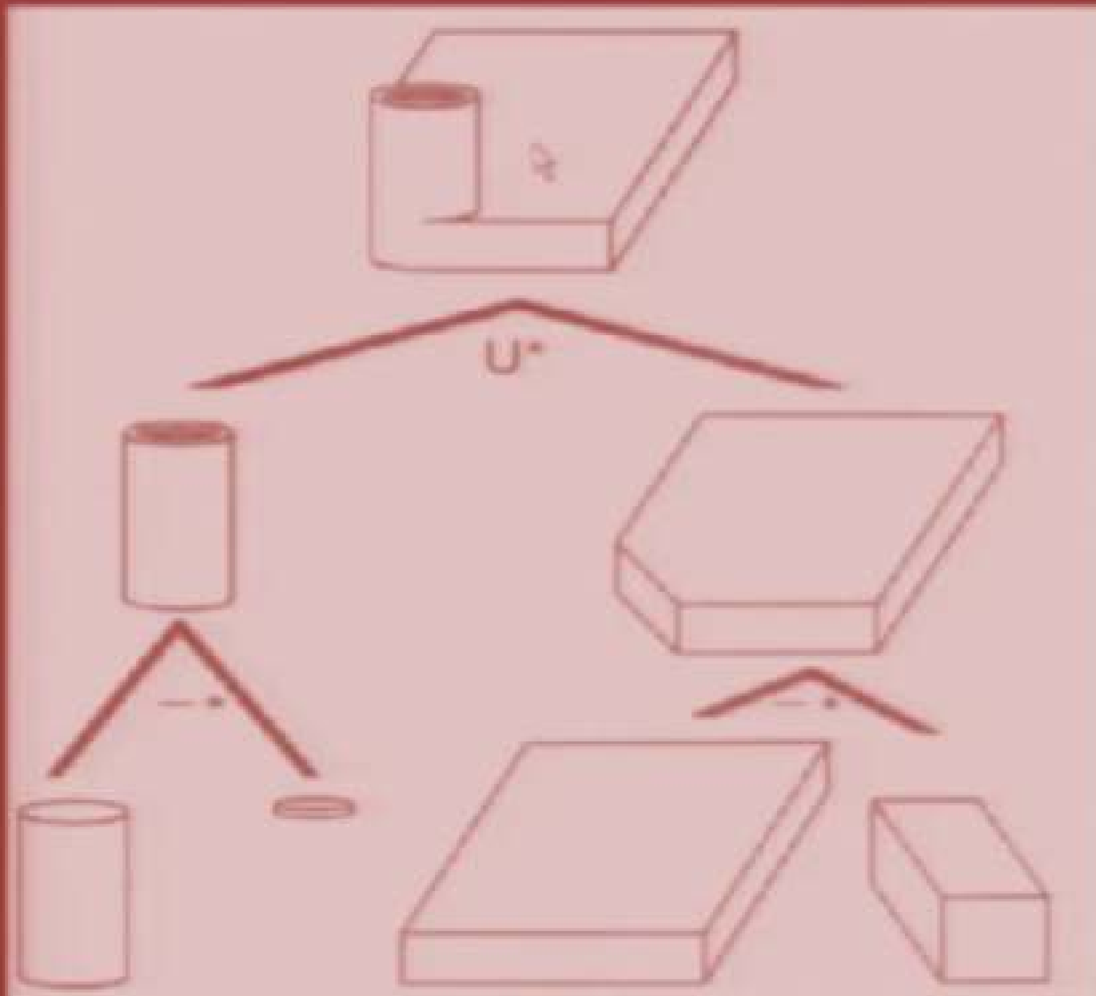
Constructive Solid Geometry, CSG

- CSG defines a model in terms of combining basic and generated (using extrusion and sweeping operation) solid shapes.
- CSG uses Boolean operations to construct a model (George Boole, 1815-1864, invented Boolean algebra).
- There are three basic Boolean operations:
  - ***Union*** **(Unite, join) - the operation combines two volumes included in the different solids into a single solid.**
  - ***Subtract*** **(cut) - the operation subtracts the volume of one solid from the other solid object.**
  - ***Intersection*** **- the operation keeps only the volume common to both solids**

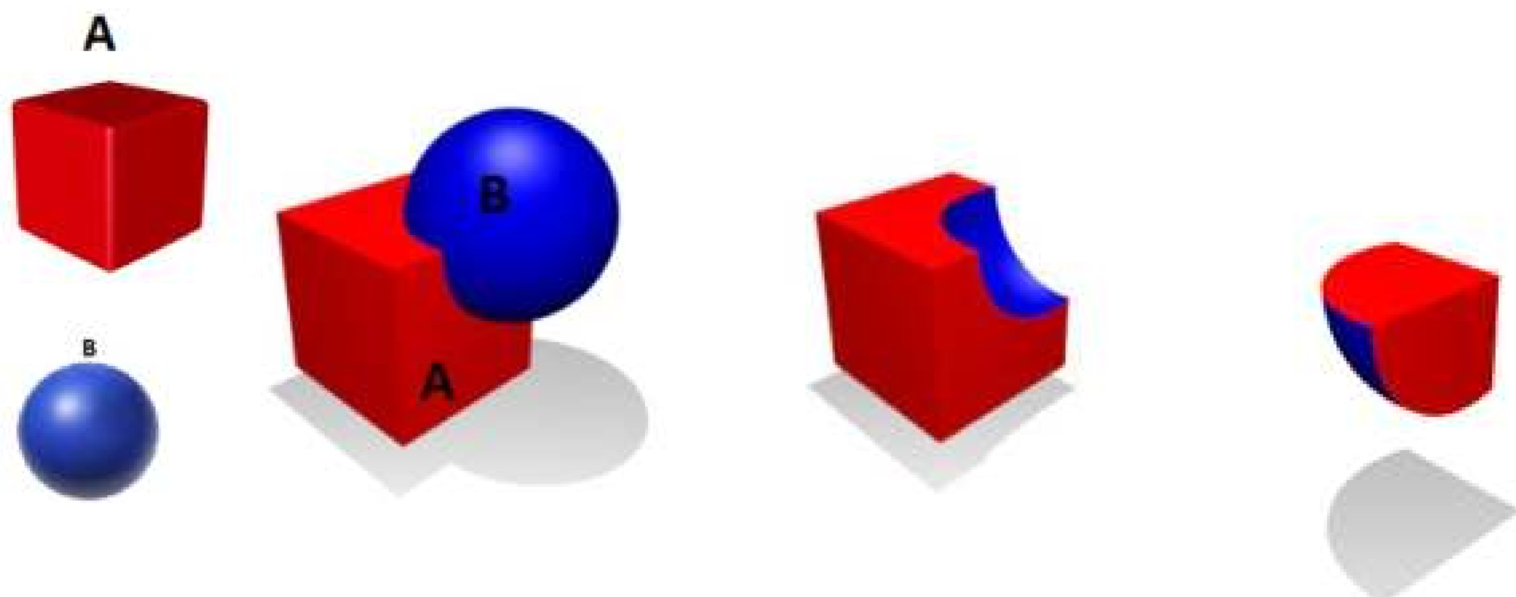
Constructive Solid Geometry (CSG)

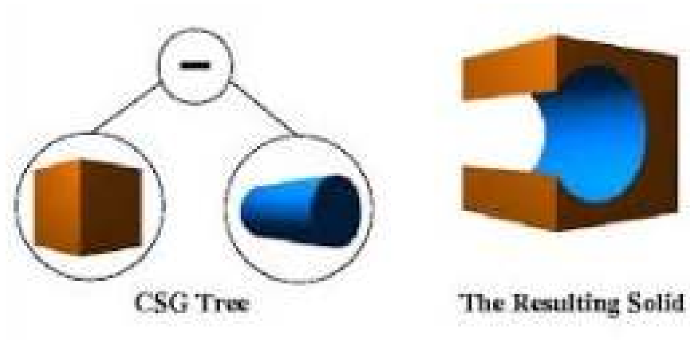


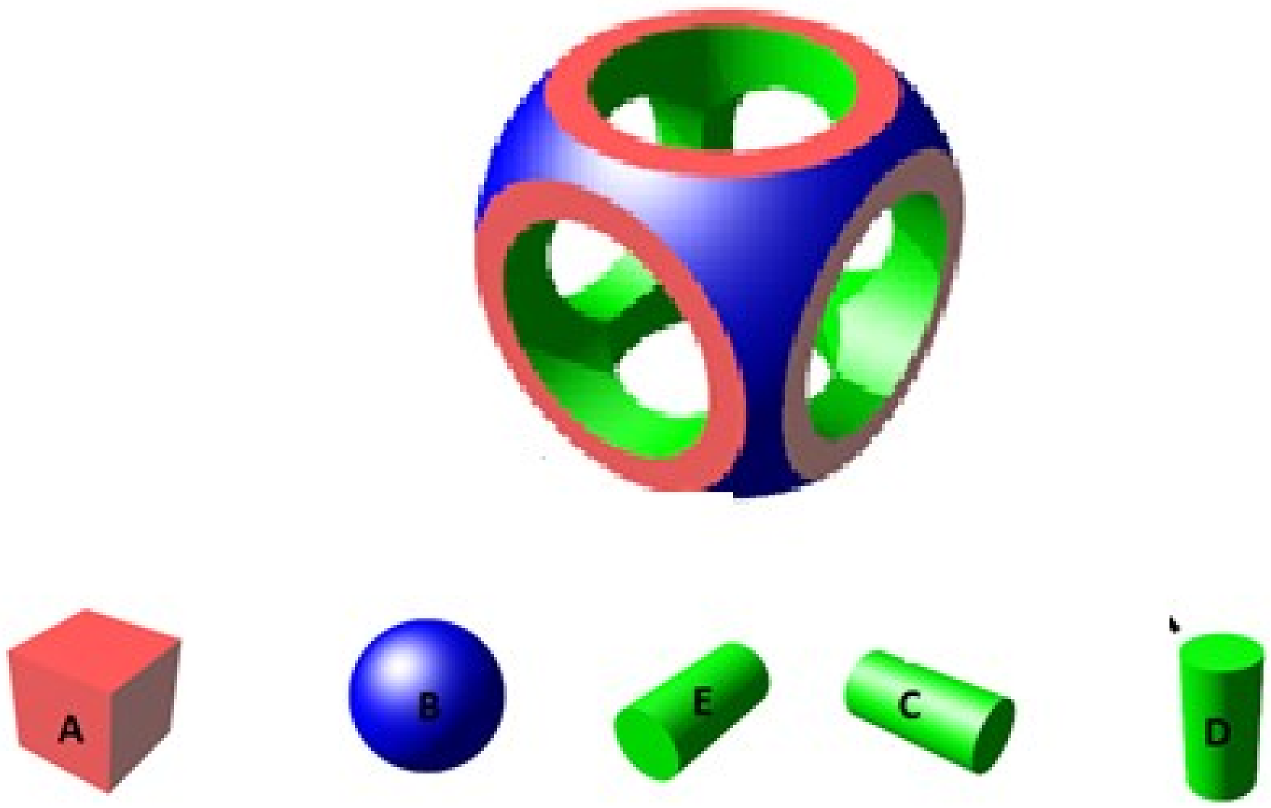
Constructive Solid Geometry

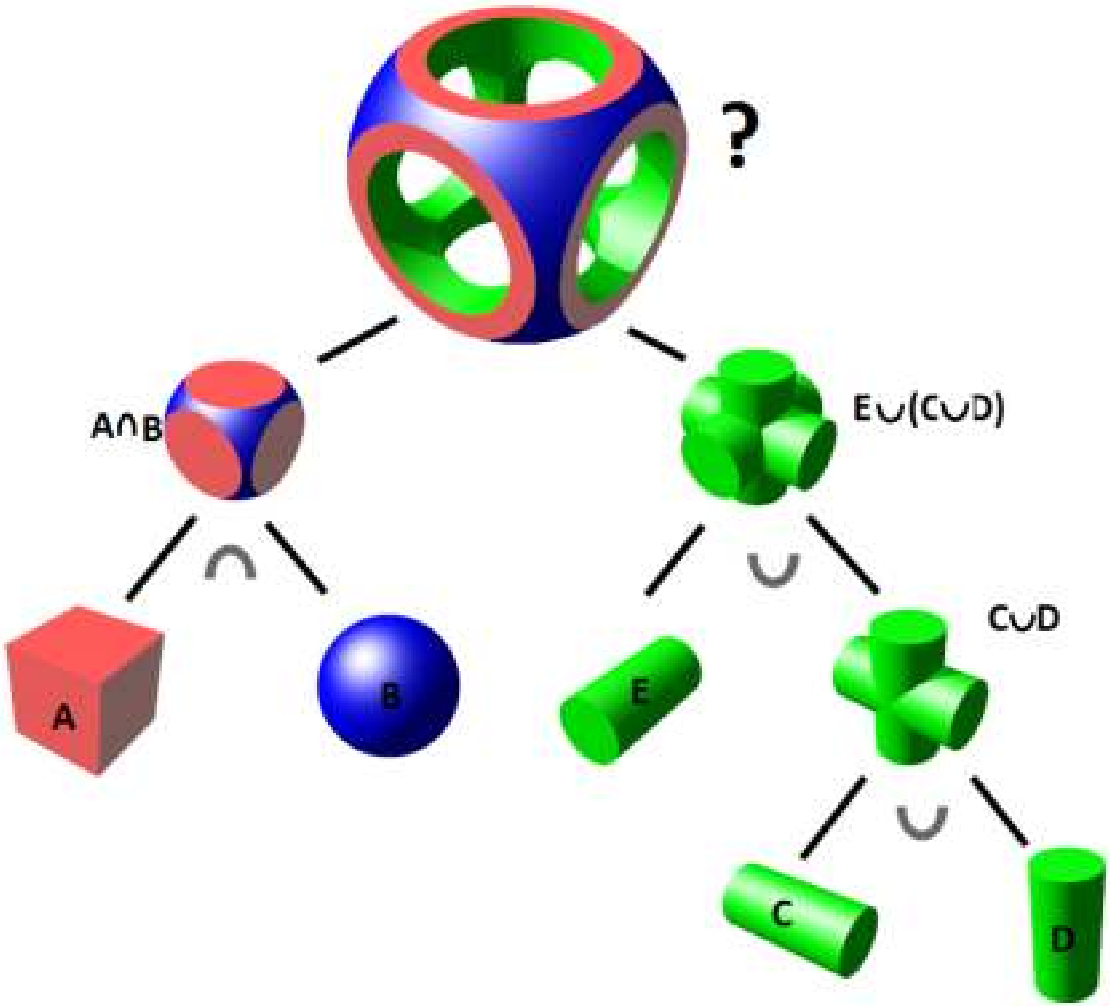


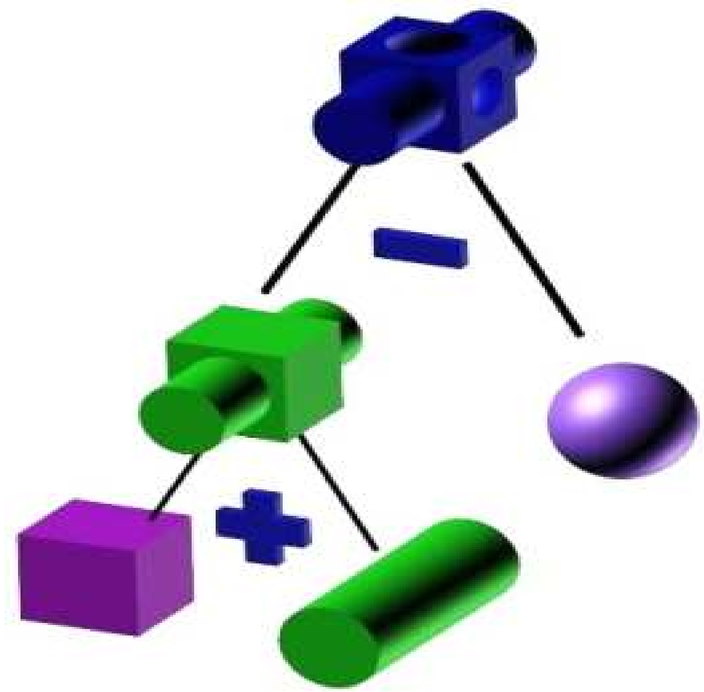
Boolean Operations in CSG





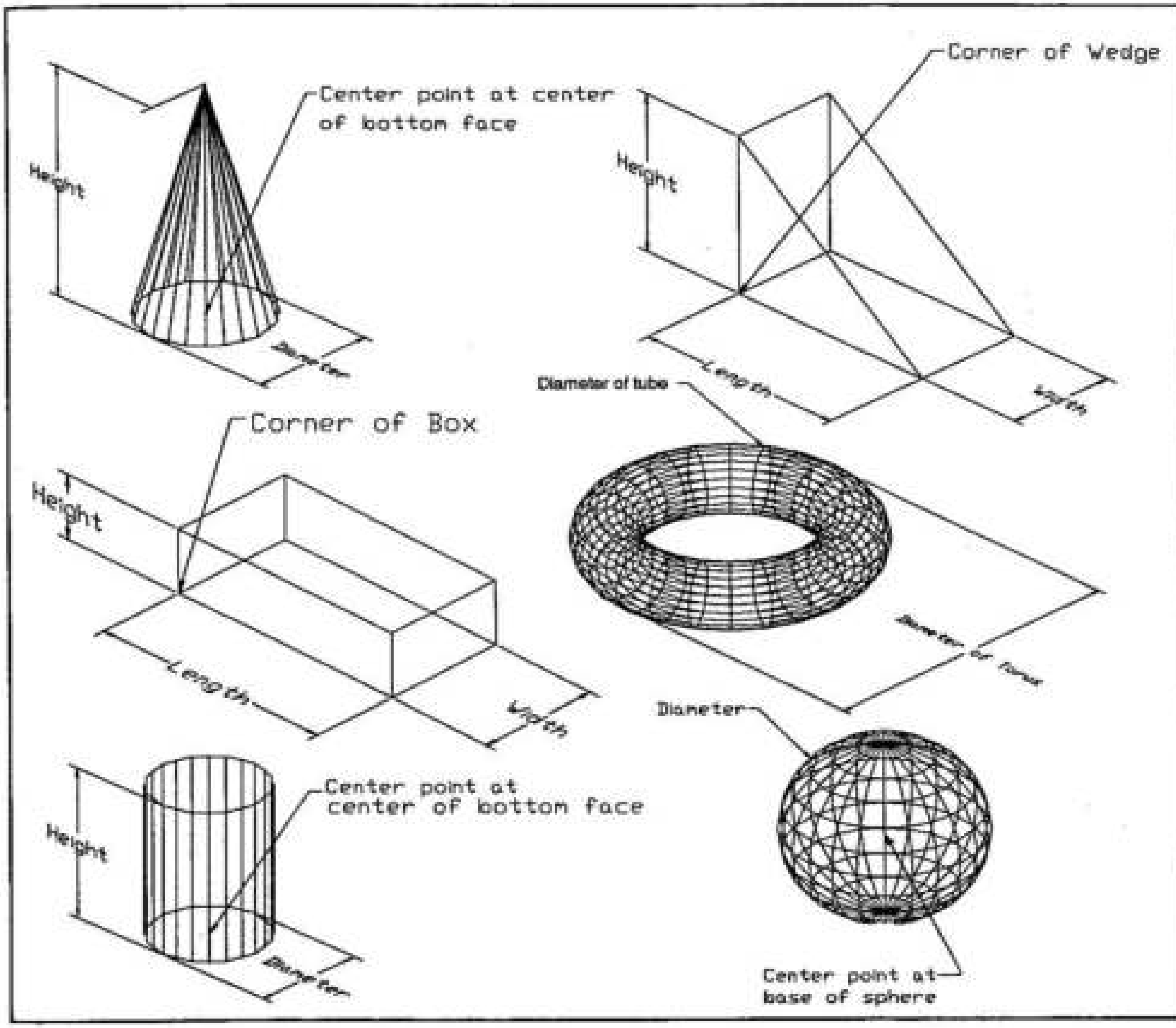

A
B
E
C
D


?
A∩B
E∪(C∪D)
∩
∪
A
B
E
C∪D
∪
C
D

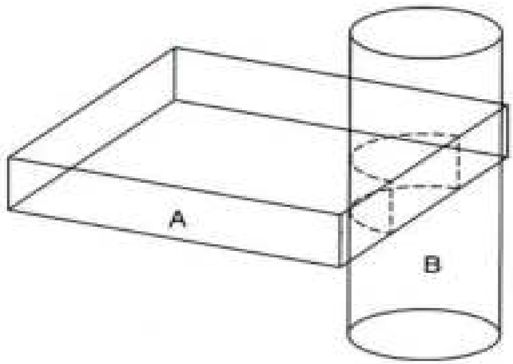


Primitive Solids

The location of the insertion base or base point and default axes orientation.

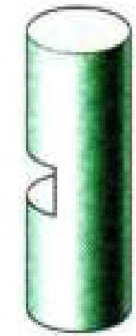


Boolean Operations

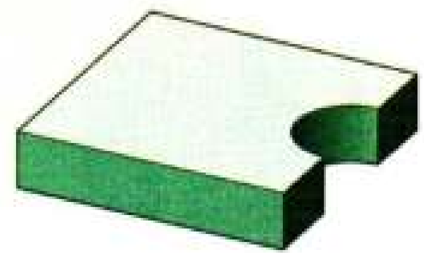


Subtract

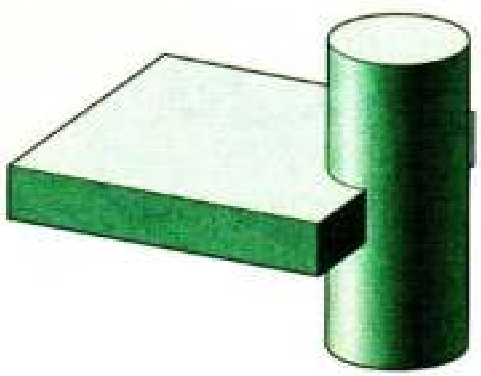
B − A



A − B



A ∪ B



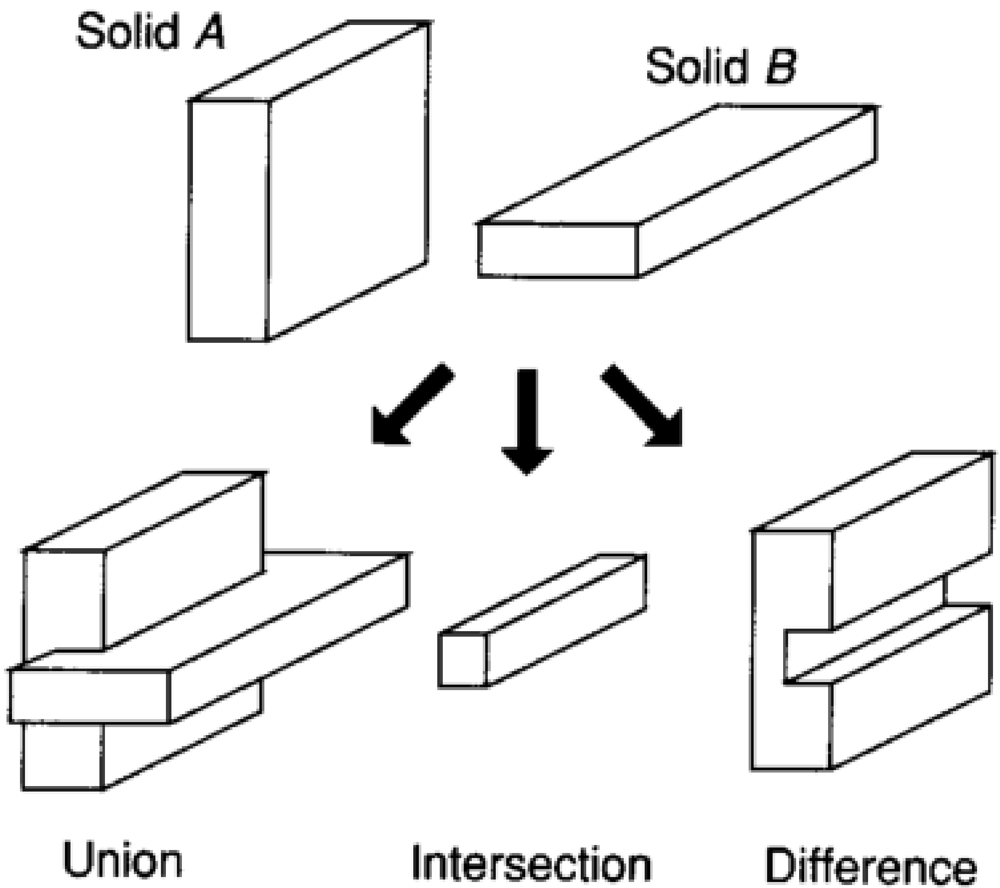
A ∩ B

Intersection

Union

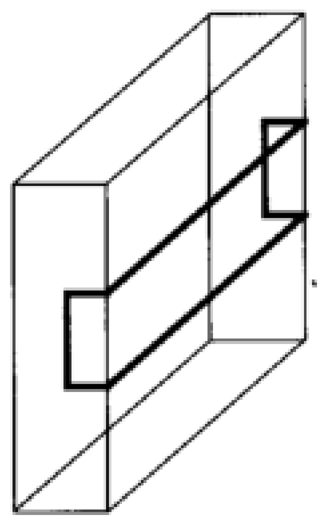
Implementing Boolean Operation

Consider solids *A* and *B*.

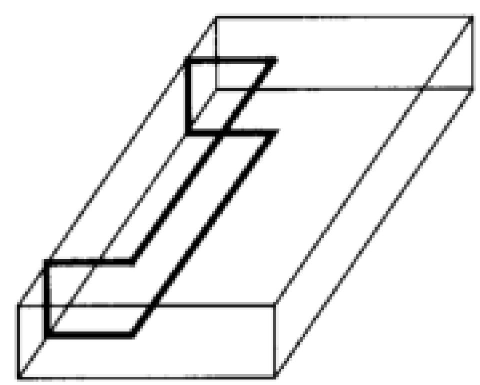


Boolean Operation

The intersection curves of all the faces of solid *A* and *B* are calculated. These intersections are inscribed on the associated faces of the two solids.



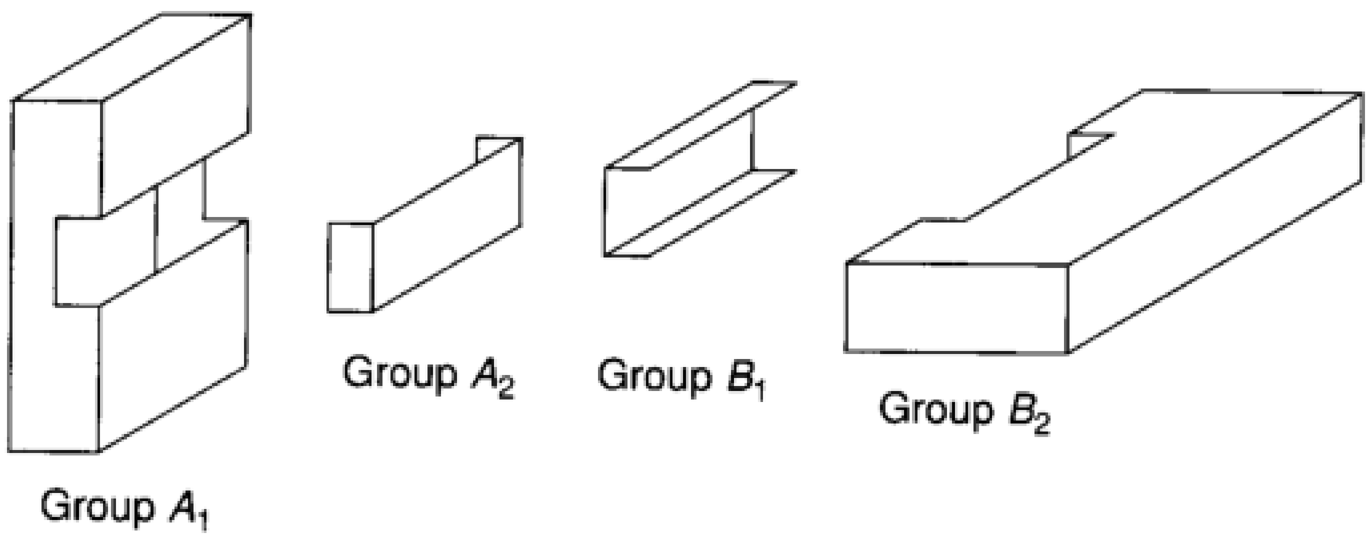
Solid *A* after modifiction



Solid *B* after modifiction

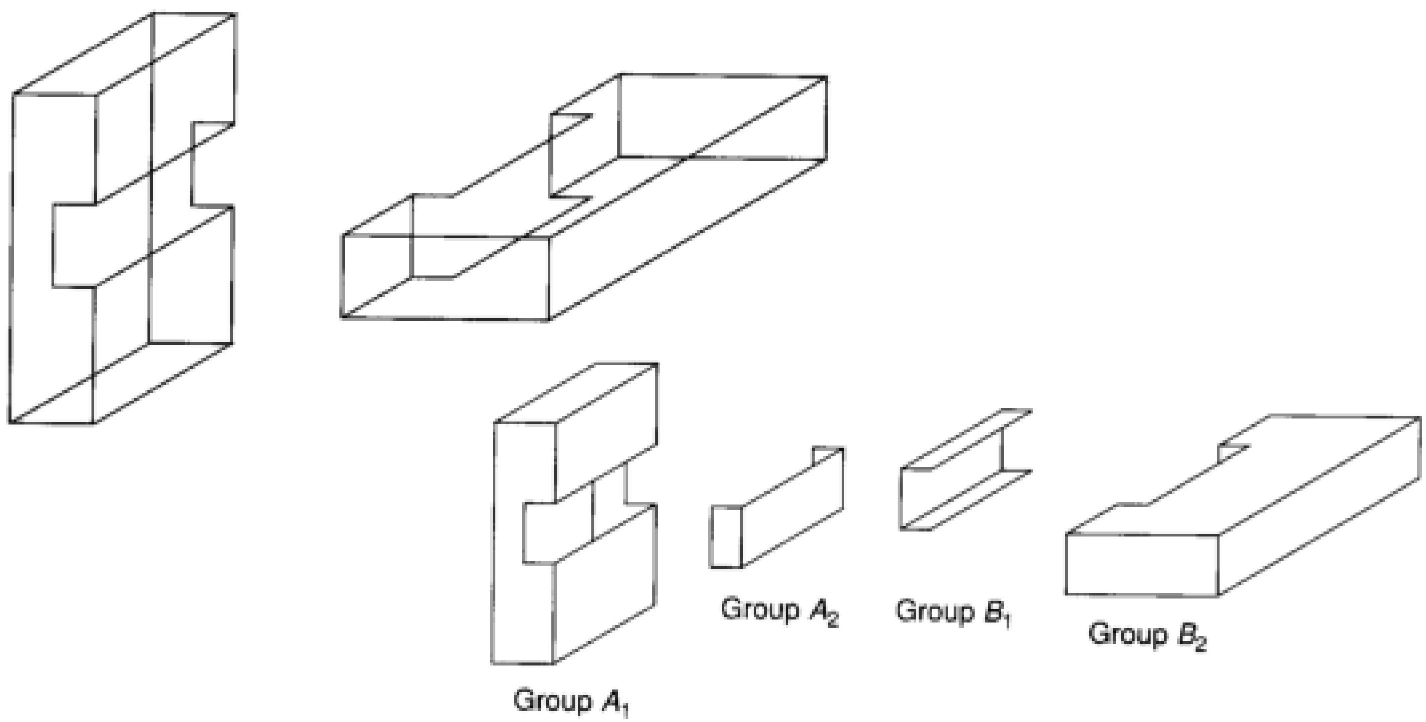
Boolean Operation

The faces of solid *A* are classified according to their relative location with respect to solid *B*. Each face is tested to determine whether it is located inside, outside, or on the boundary surface of solid B.
The faces in group A_1 are outside solid *B*, and those of group B_1 are inside solid *A*.



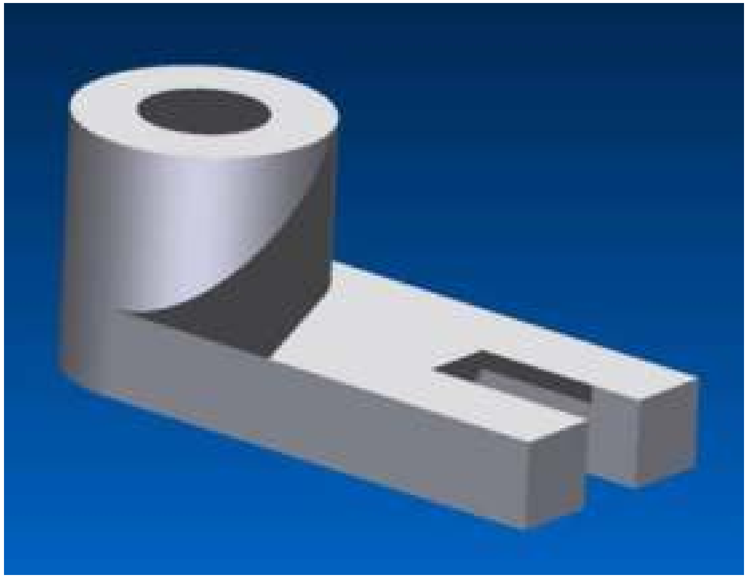
Boolean Operation

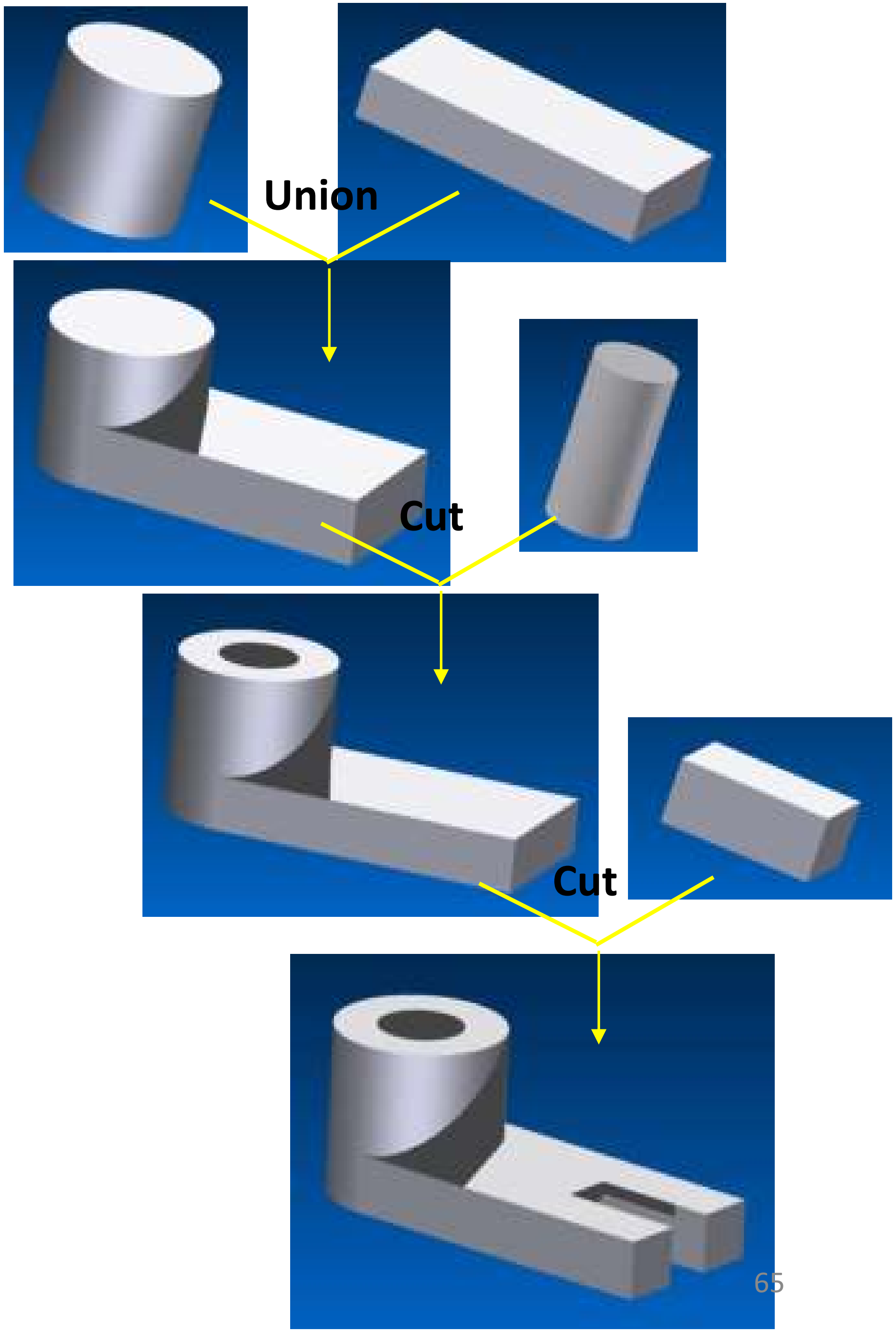
Groups of faces are collected according to the specific Boolean operation and the unnecessary face groups are eliminated. For example, for union operation, group A_1 and B_2 are collected and A_2 and B_1 are eliminated.



Solid Modeling Example Using CSG

Plan your modeling strategy before you start creating the solid model



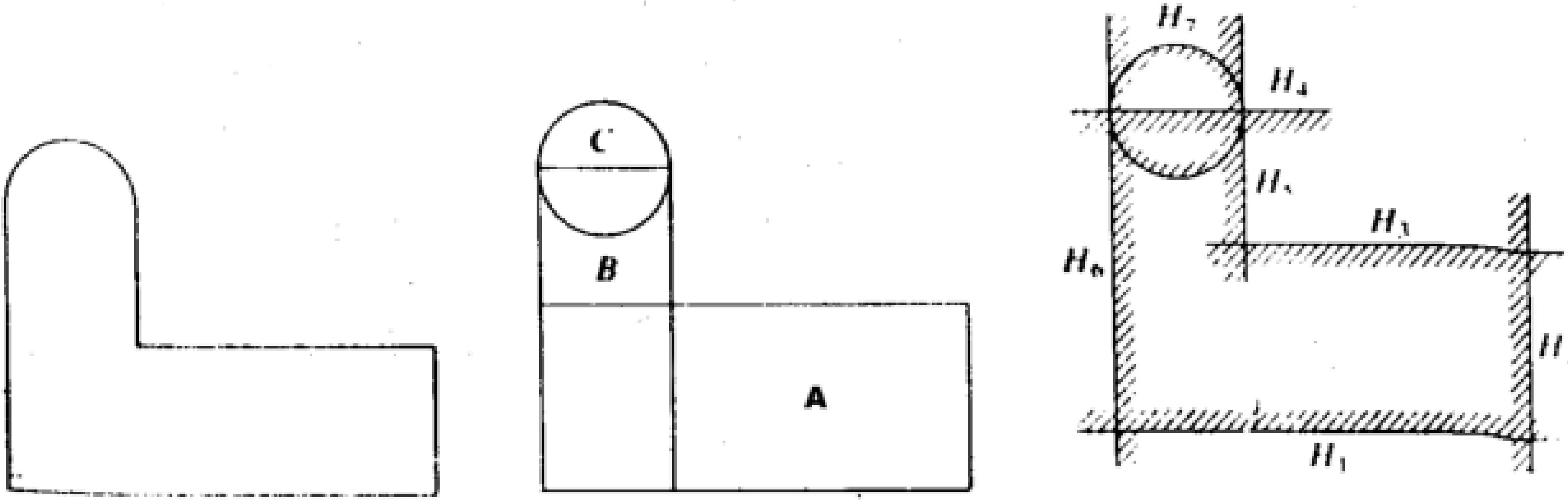


- Offers representations that are succinct
- Easy to create and store
- Easy to check for validity
- Based on topological notation;
  - Physical object can be divided into a set of primitives
  - Combined by a set of rules
- Primitives themselves are considered valid CSG models
- Each primitive is bounded by a set of surfaces
- Two main types of CSG schemes

  1) Most popular, based on bounded solid primitives – r-sets

  2) Less popular, generally unbounded half-spaces – non r-sets
- Bounded primitives, more concise

Bounded and unbounded primitives

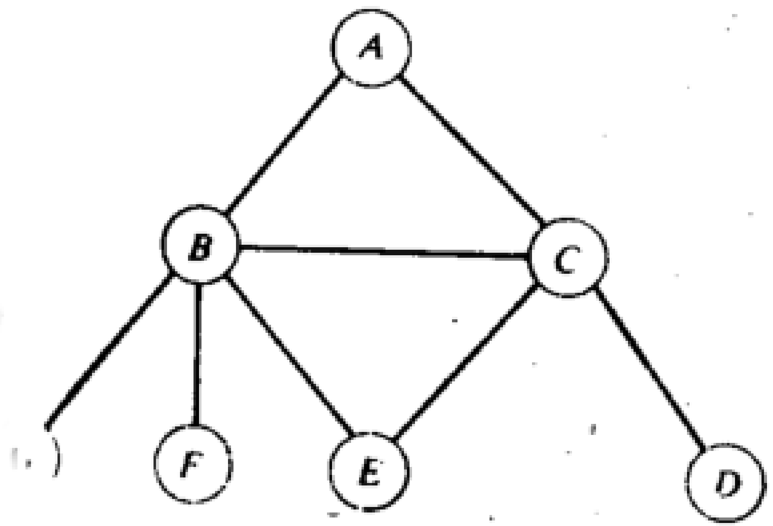


(*a*) Solid (*b*) Bounded primitives (c) Unbounded half-spaces

- Database stores its topology and geometry
- Topology is created via regularized set operations
- Validity checks of the used primitives, syntactically correct
- Geometry stored includes configuration parameters of its primitives and rigid motion and transformation.

- Data structures of most CSG representation are based on the concept **of graphs and trees.**
- **Graph** : set of nodes connected by a set of branches of lines.
- Each branch in a graph is specified by a pair of nodes.

Graph

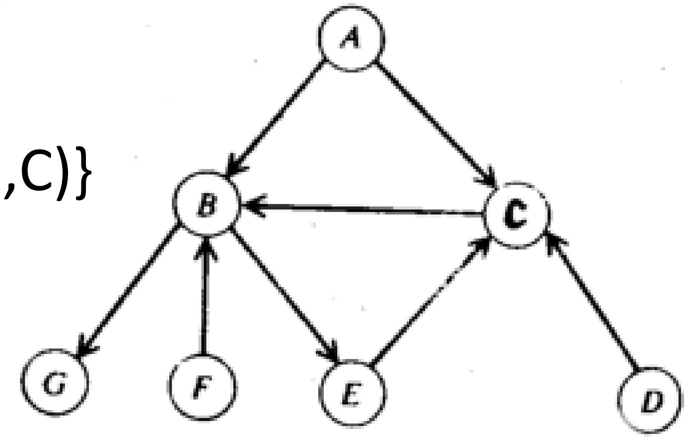


(a) Graph

- The set of nodes is {A,B,C,D,E,F,G}
- Set of branches or set of pairs is {{A,B},{A,C},{B,C},{B,E},{B,F},{B,G},{C,D},{C,E}}

DIGRAPH

- The set of ordered pairs {(A,B),(A,C),(C,B),(B,E),(F,B),(B,G),(D,C),(E,C)}

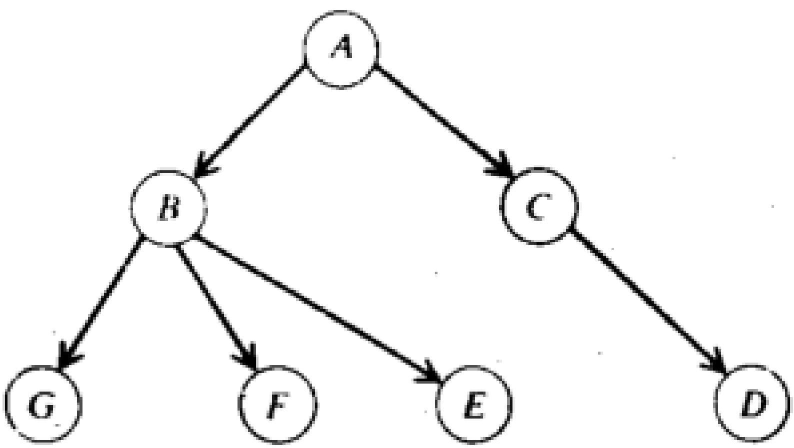


(b) Digraph

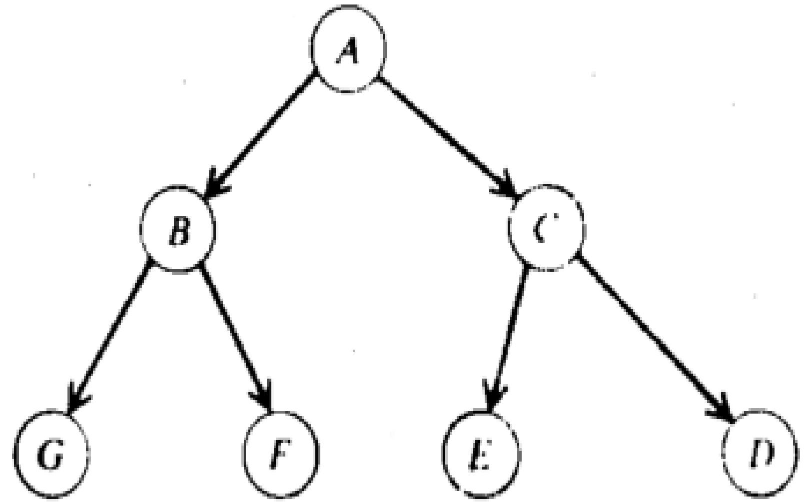
- Each node in digraph has an indegree and outdegree
- Indegree : Number of arrow heads entering the node
- Outdegree : Number of arrow tails leaving the node
- Cyclic : Start and end node of a path are the same
- If not, acyclic

Trees

- Tree : An acyclic digraph in which only a single node, called the root has a zero indegree and every other node has an indegree of 1
- A graph need not be a tree but a tree must be a graph
- Ancestor or predecessor, descendants
- **Binary tree** : Each node of an ordered tree has two descendants (left and right)

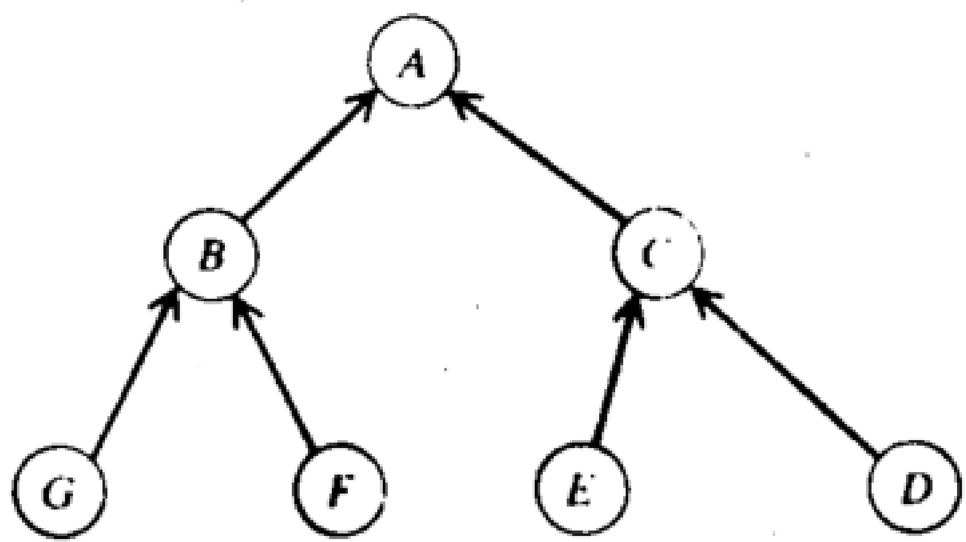


(a) Tree



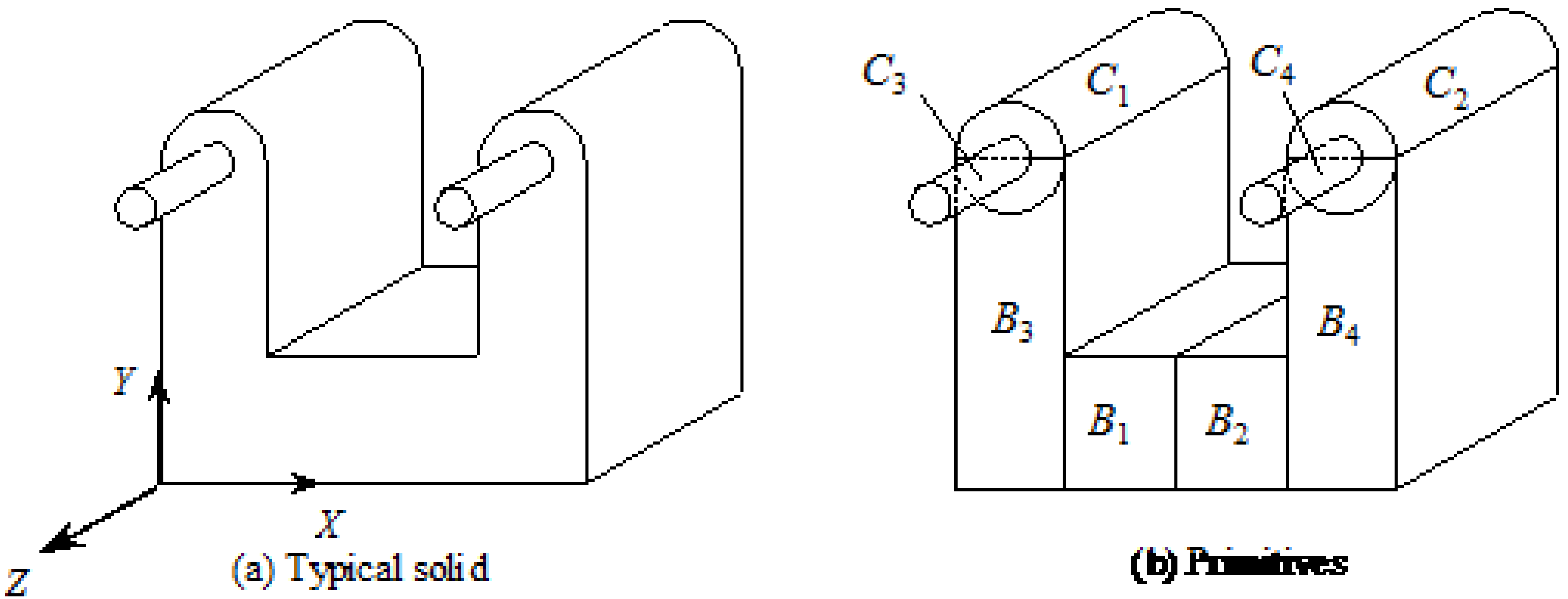
(b) Binary tree

- Inverted binary tree : Arrow directions in a binary tree are reversed
- Inverted binary tree is very useful to understand the data structure of CSG models (Boolean models)
- A CSG tree is defined as an inverted ordered binary tree whose leaf nodes are primitives and interior nodes are regularized set operations



(c) Inverted binary tree

Primitives



(a) Typical solid

(b) Primitives

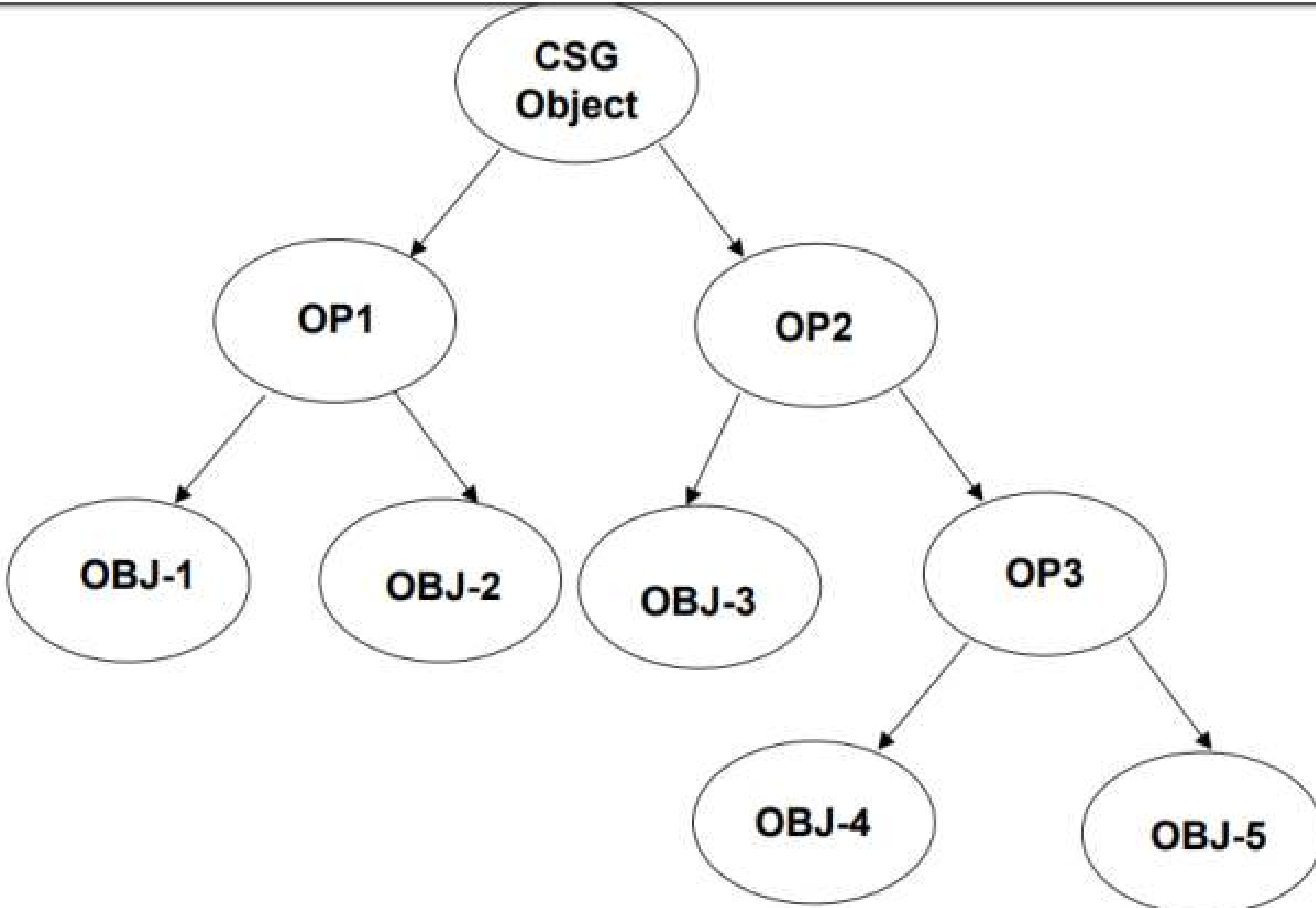
$$\left.\begin{aligned}
&B_1 = block \quad positioned \quad properly \\
&B_2 = block \quad positioned \quad properly \\
&B_3 = block \\
&B_4 = B_3 \quad moved \quad properly \quad in \quad the \quad X \quad direction \\
&C_1 = cylinder \quad positioned \quad properly \\
&C_2 = C_1 \quad moved \quad properly \quad in \quad the \quad X \quad direction \\
&C_3 = cylinder \quad positioned \quad properly \\
&C_4 = C_3 \quad moved \quad properly \quad in \quad the \quad X \quad direction
\end{aligned}\right\} \text{Primitives' definitions}$$

Boolean operation

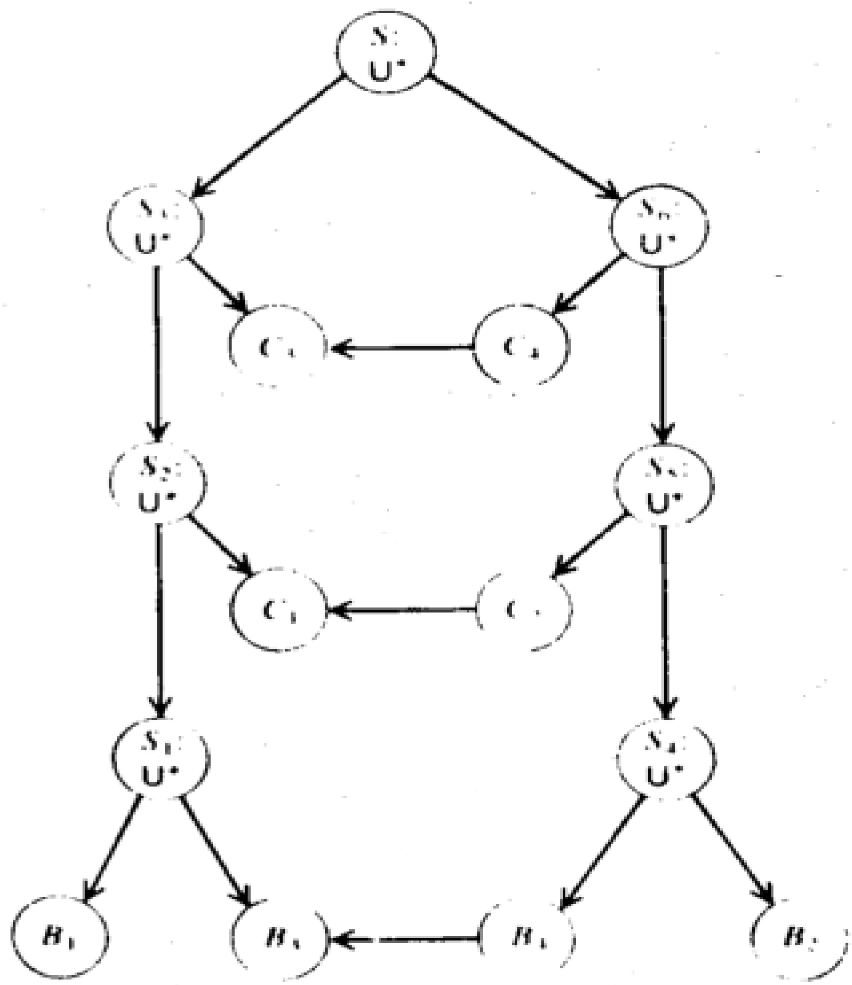
$$\left.\begin{aligned} S_1 &= B_1 \cup^* B_3 \\ S_2 &= S_1 \cup^* C_1 \\ S_3 &= S_2 \cup^* C_3 \end{aligned}\right\} \quad \text{Construct left half}$$

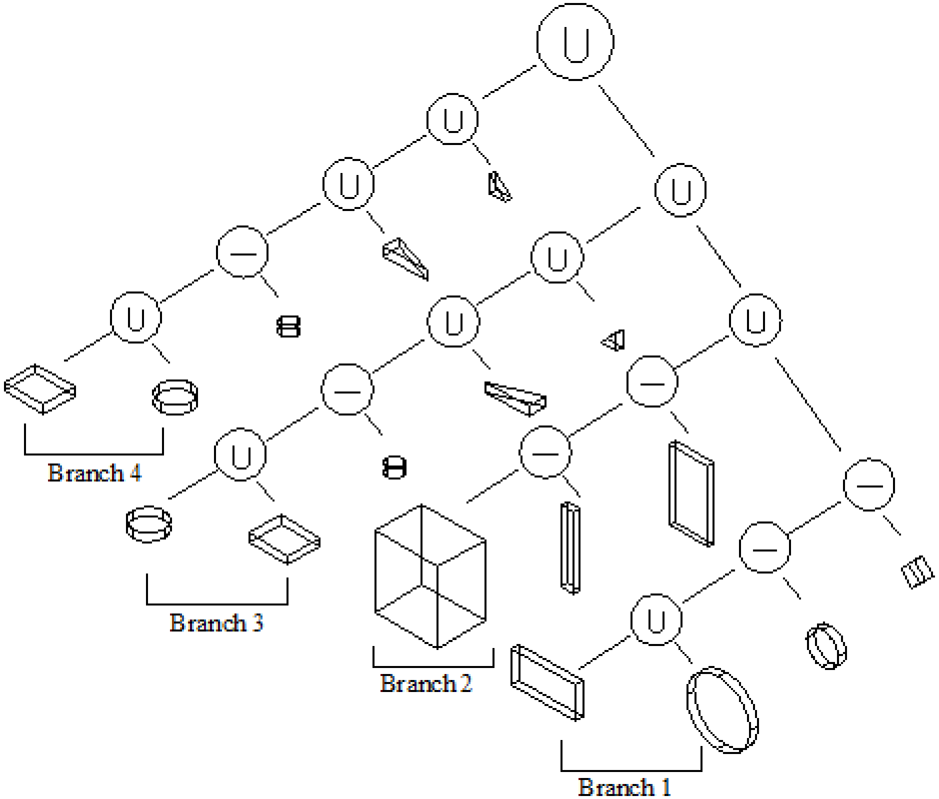
$$\left.\begin{aligned} S_4 &= B_2 \cup^* B_4 \\ S_5 &= C_2 \cup^* S_4 \\ S_6 &= C_4 \cup^* S_5 \end{aligned}\right\} \quad \text{Construct right half}$$

$$\left. S = S_3 \cup^* S_6 \right\} \quad \text{Model}$$

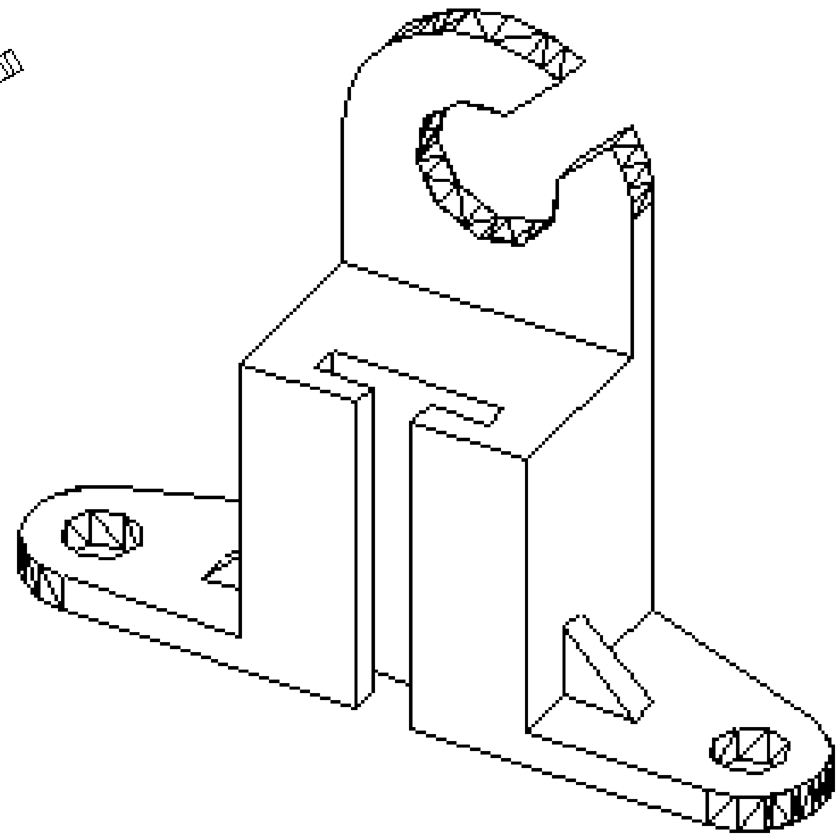

CSG
Object
OP1
OP2
OBJ-1
OBJ-2
OBJ-3
OP3
OBJ-4
OBJ-5

CSG graphs





(a) Construction tree


(c) Final solid model

BOUNDARY REPRESENTATION (B-rep)

BOUNDARY REPRESENTATION …(B-rep)

- A solid model is formed by defining the surfaces that form its boundary (edges and surfaces)

- The face of a B-rep represents an oriented surface, there are two sides to the surface; solid side (inside) and void side (outside), unlike faces in a wireframe.

- B-rep model is created using Euler operation

- Many Finite Element Method (FEM) programs use this method. Allows the interior meshing of the volume to be more easily controlled.

Boundary Representation (B-Rep)

- In this representation solid objects are represented as unions of their boundaries or enclosing surfaces. The enclosing surfaces can include planar polygons, quadrics and free-form surface patches.

- In this scheme topological and geometric information are explicitly defined

Boundary Representation

- Topological information provides the relationships among vertices, edges/curves and aces/patches. In addition to connectivity, topological information also includes orientation of edges and faces etc.

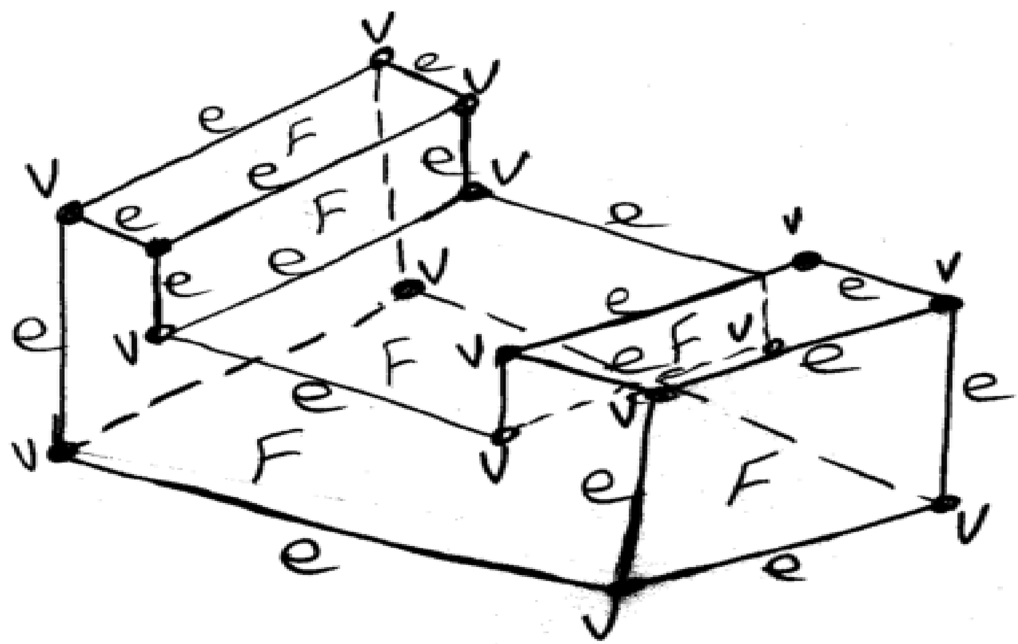
- Geometric information usually consists of equations of the edges/curves and faces/patches.

- One of the most widely used schemes
- Topological notion - Physical object is bounded by a set of faces, faces are regions or subsets of closed and orientable surfaces,closed surface is one that is continuous without breaks.
- Each face is bounded by edges
- Each edge is bounded by vertices
- Database contains both topology and geometry
- Topology is created by performing Euler operations
- Geometry is created by performing Euclidean calculations

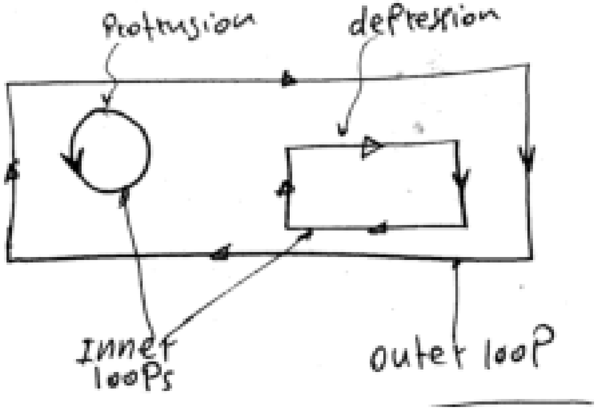
- Euler operations are used to create, manipulate, and edit the faces, edges and vertices of a boundary model
- Euler operators are considered to be lower level operators than Boolean operators
- Boolean operators are often employed as one of the means of creating, manipulating, and editing the model
- Stores only bounding surfaces of the solid

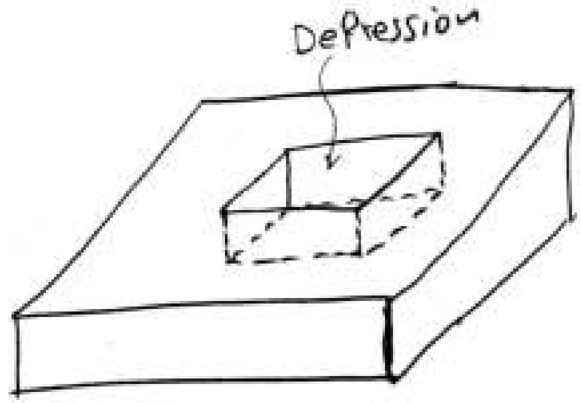
B-Rep Entities Definition

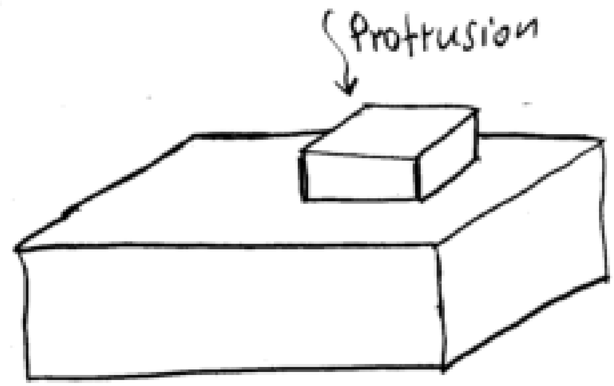
- **Vertex** is a unique point in space
- **An Edge** is a finite, non-self-intersecting, directed space curve bounded by two vertices
- **A Face** is defined as a finite connected, non-self-intersecting, region of a closed oriented surface bounded by one or more loops



- A **Loop** is an ordered alternating sequence of vertices and edges. A loop defines a non-self-intersecting, piecewise, closed space curve which, in turn, may be a boundary of a face.

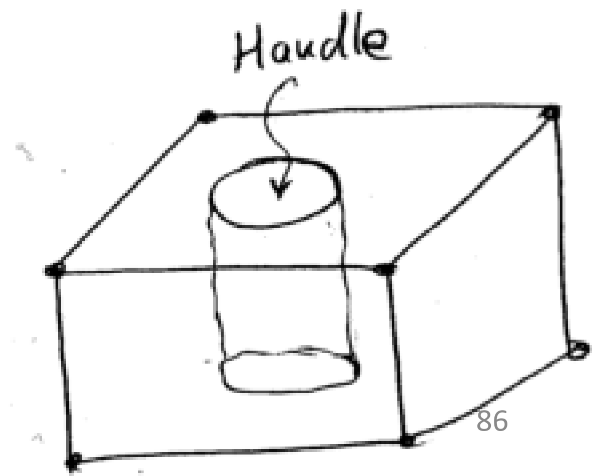






- A Handle (Genus or Through hole) is defined as a passageway that passes through the object completely.

- A Body (Shell) is a set of faces that bound a single connected closed volume. Thus a body is an entity that has faces, edges, and vertices

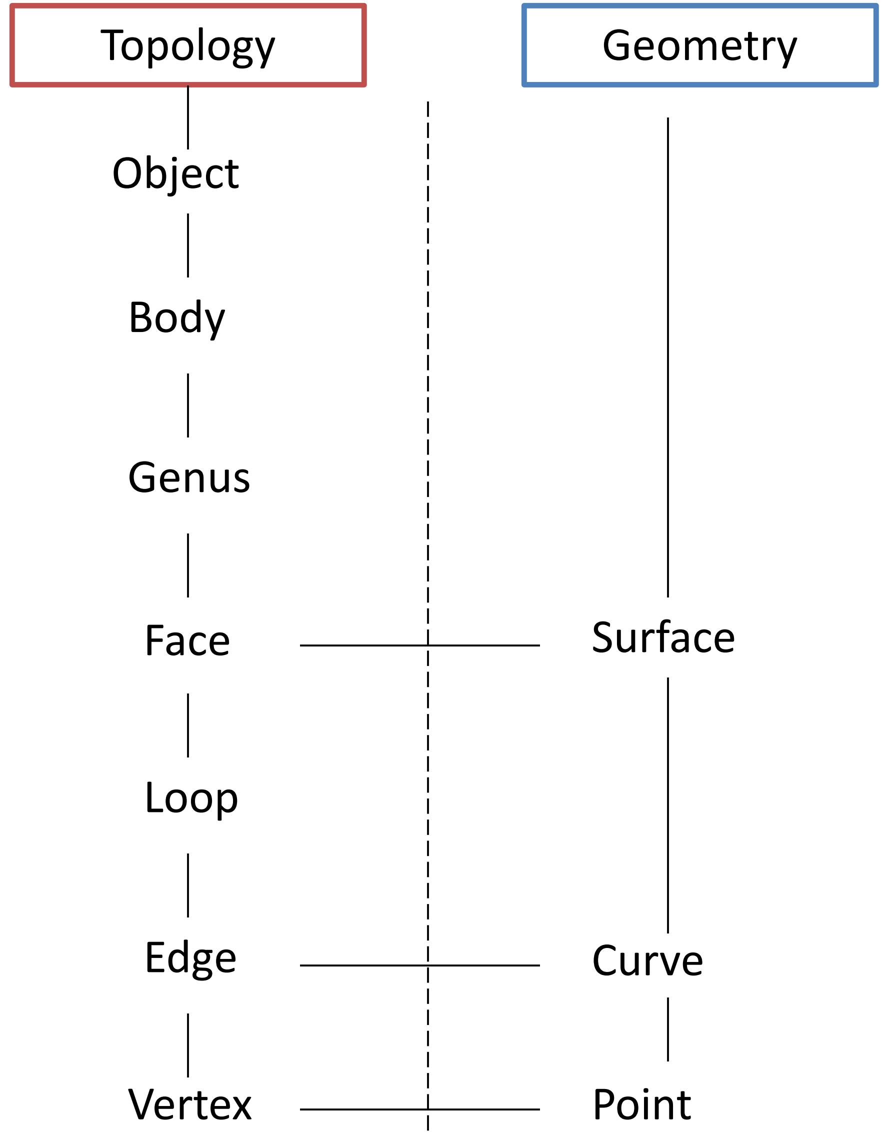


B-Rep Data Structure

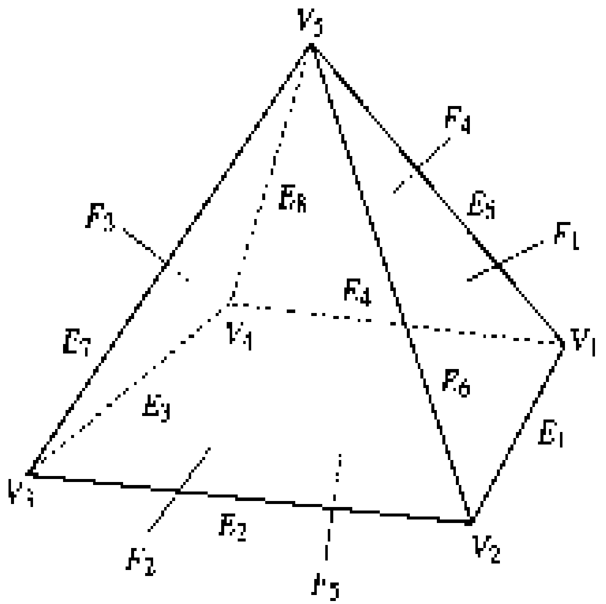
A general data structure for a boundary model should have both topological and geometrical information

- Geometry relates to the information containing shape defining parameters, such as the coordinates of the vertices

- Topology describes the connectivity among the various geometric components, that is, the relational information between the different parts of an object



B-Rep Data Structure



Face Table

Face	*Edges*
F_1	E_1, E_5, E_6
F_2	E_2, E_6, E_7
F_3	E_3, E_7, E_8
F_4	E_4, E_8, E_5
F_5	E_1, E_2, E_3, E_4

Edge Table

Edge	*Vertices*
E_1	V_1, V_2
E_2	V_2, V_3
E_3	V_3, V_4
E_4	V_4, V_1
E_5	V_1, V_5
E_6	V_2, V_5
E_7	V_3, V_5
E_8	V_4, V_5

Vertex Table

Vertex	*Coordinates*
V_1	x_1, y_1, z_1
V_2	x_2, y_2, z_2
V_3	x_3, y_3, z_3
V_4	x_4, y_4, z_4
V_5	x_5, y_5, z_5
V_6	x_6, y_6, z_6

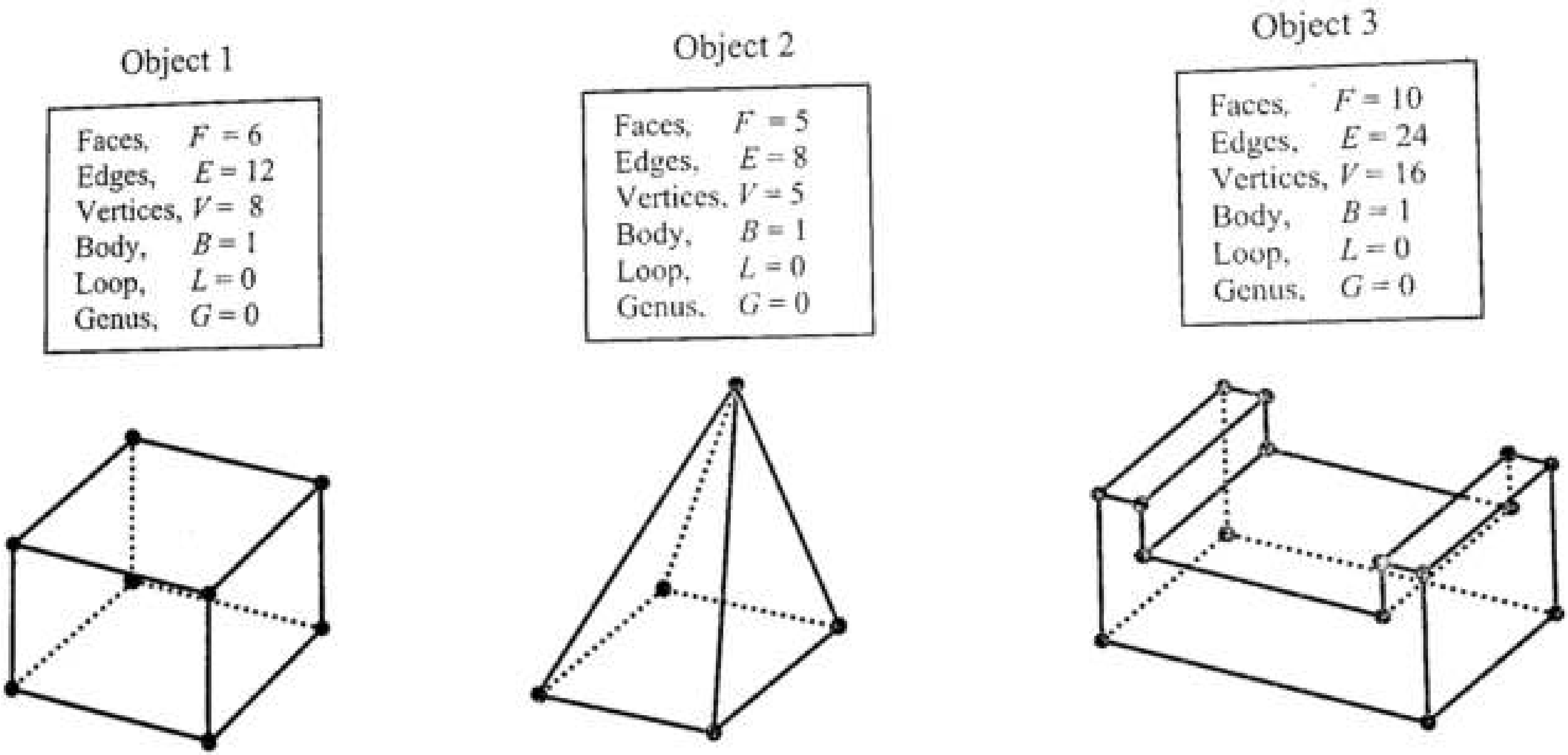
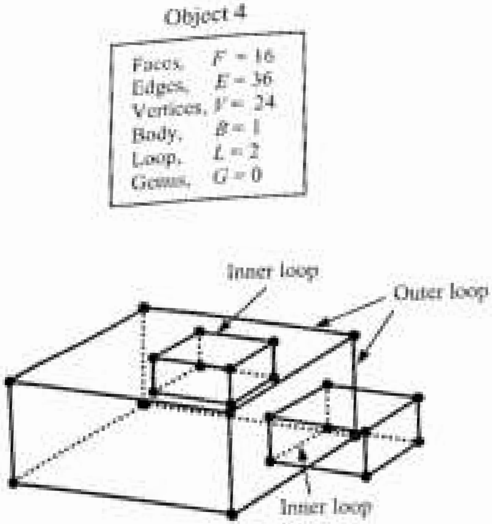
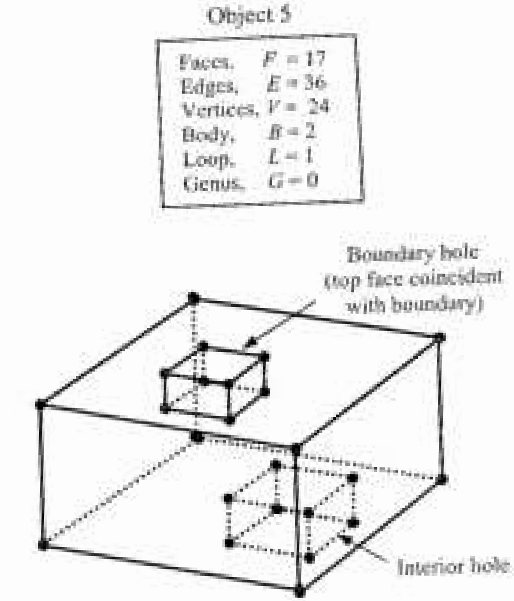


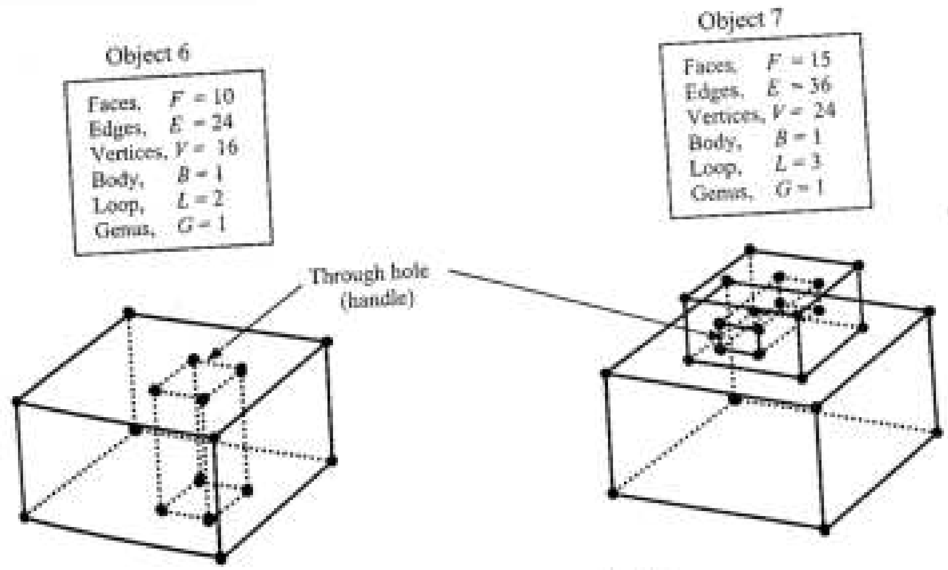
Fig. 9.31. Simple polyhedral objects bounded by loop of edges



(a) Polyhedral object with faces of inner loop



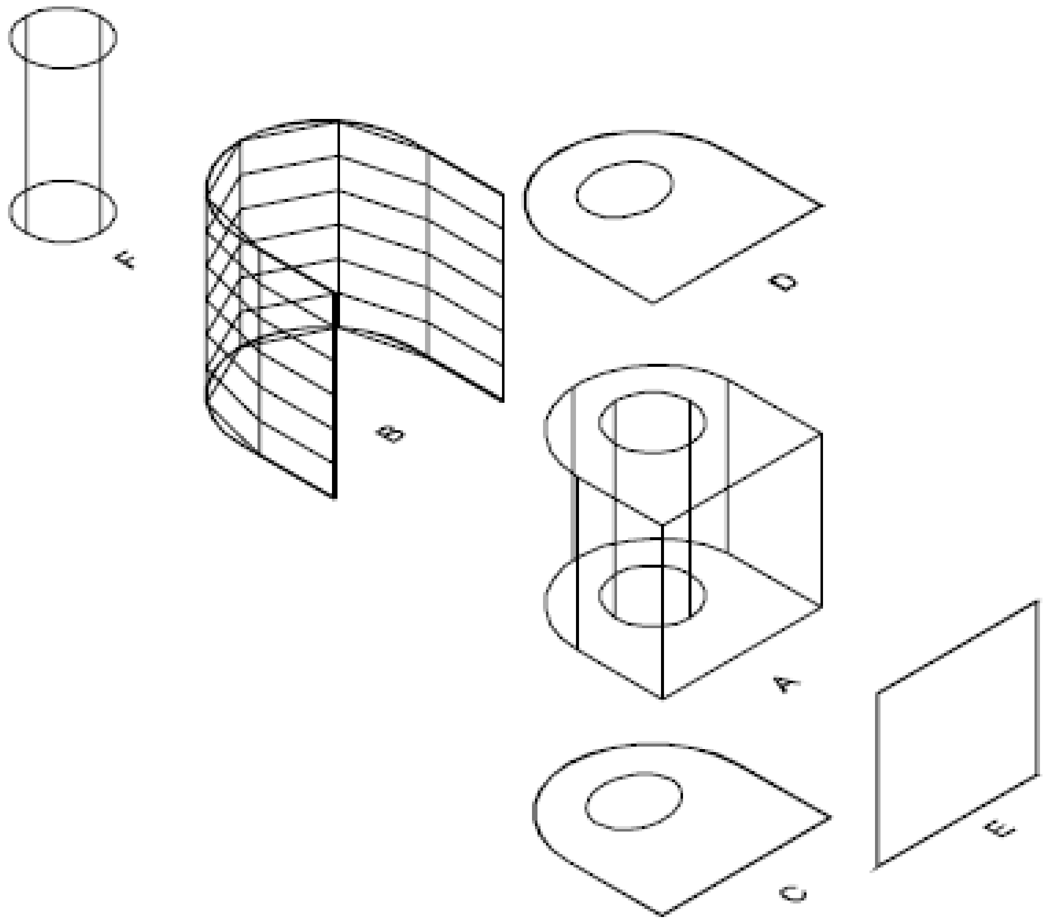
(b) Polyhedral object with not through hole



(c) Polyhedral object with through holes

Fig. 9.32. Polyhedral objects bounded by inner loop, interior hole and through holes

B-Rep Model



B-Rep Model

Solid modeling system

If a solid modeling system is to be designed,

1. The domain of its representation scheme must be defined
2. The basic elements must be identified
3. The proper operators that enable to build complex objects must be developed
4. Suitable date structure to store all relevant data
5. Other system and geometric utilities (such as intersection algorithms) may also need to be designed

Basic Elements

- Objects that are often encountered in engineering applications can be classified - polyhedral or curved objects
- Polyhedral object – consists of planar faces connected at straight edges which in turn, are connected at vertices
- Curved object – similar to a polyhedral object but with curved faces and edges instead.

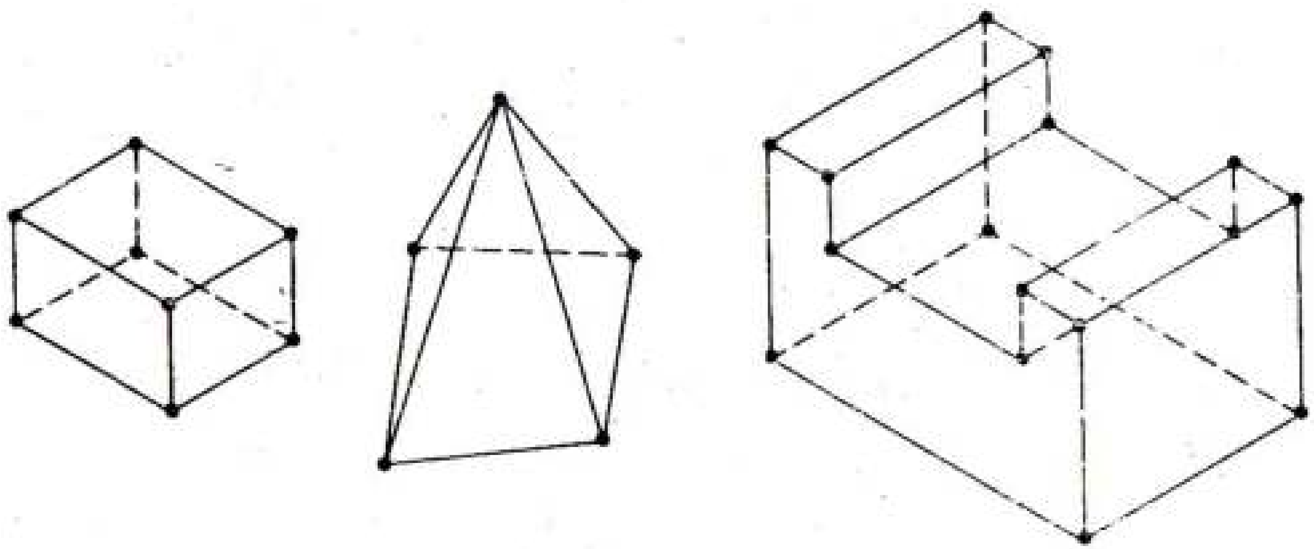
Polyhedral objects

Classified into 4 classes,

1. Simple polyhedra
2. With faces of inner loops
3. With not through holes
4. With handles

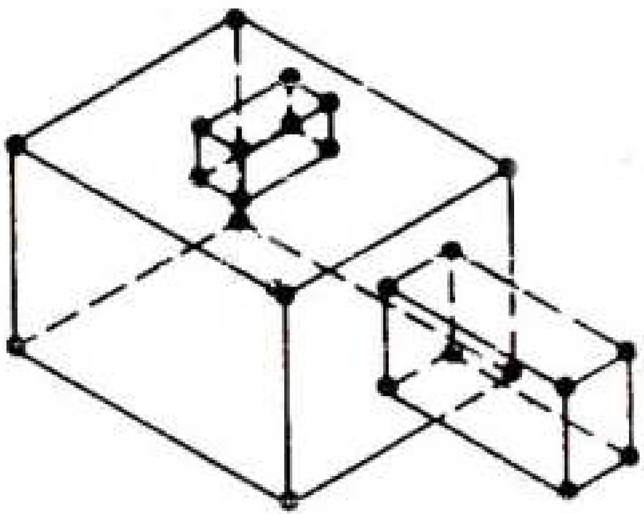
Simple polyhedra

- Do not have holes
- Each face is bounded by a single set of connected edges



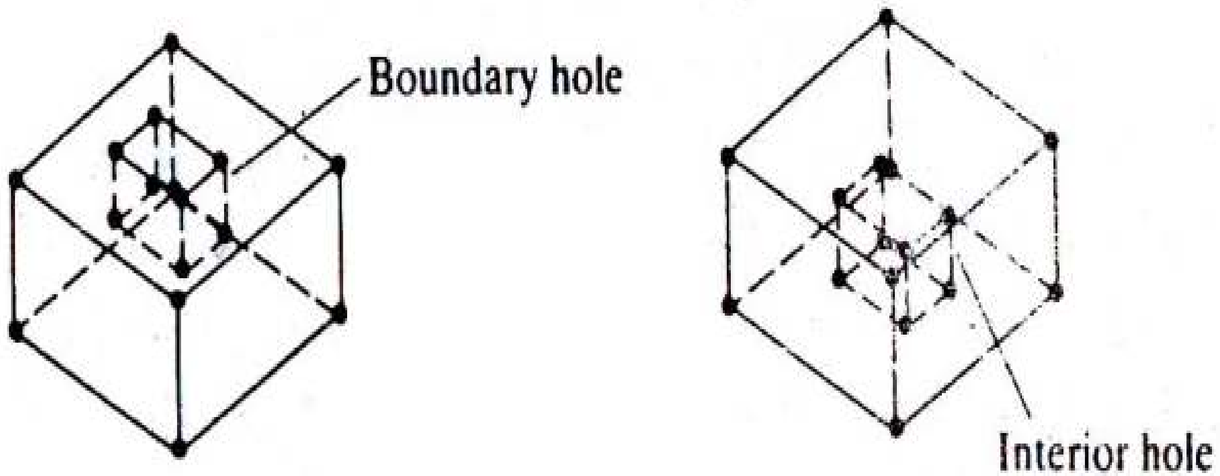
Simple polyhedra

With faces of inner loops



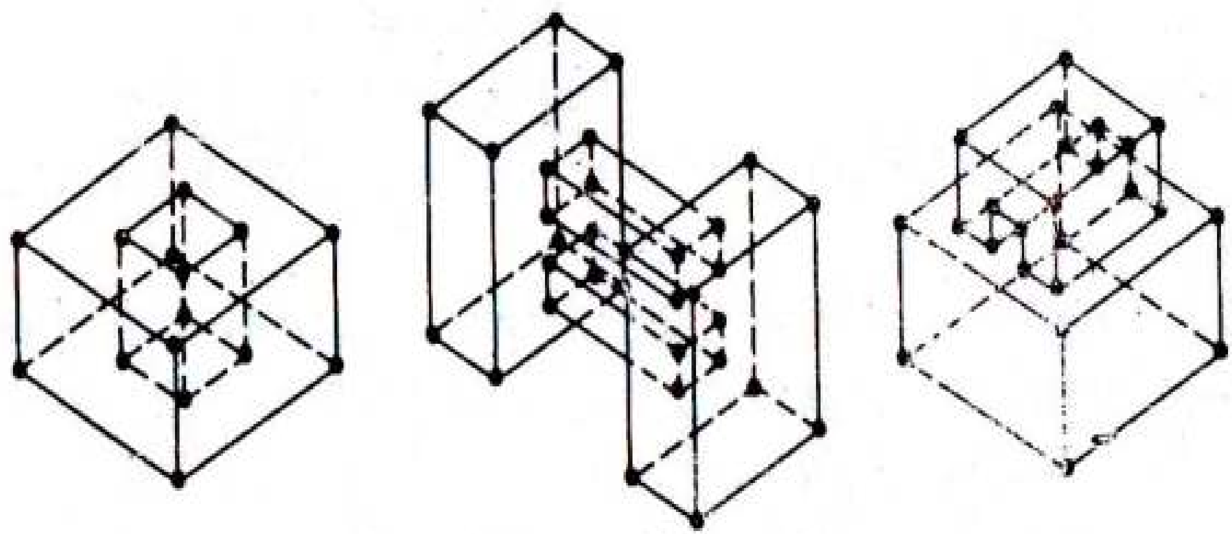
Polyhedra with faces of inner loops

With not through holes



Polyhedra with not through holes

With handles



Polyhedra with handles (through holes)

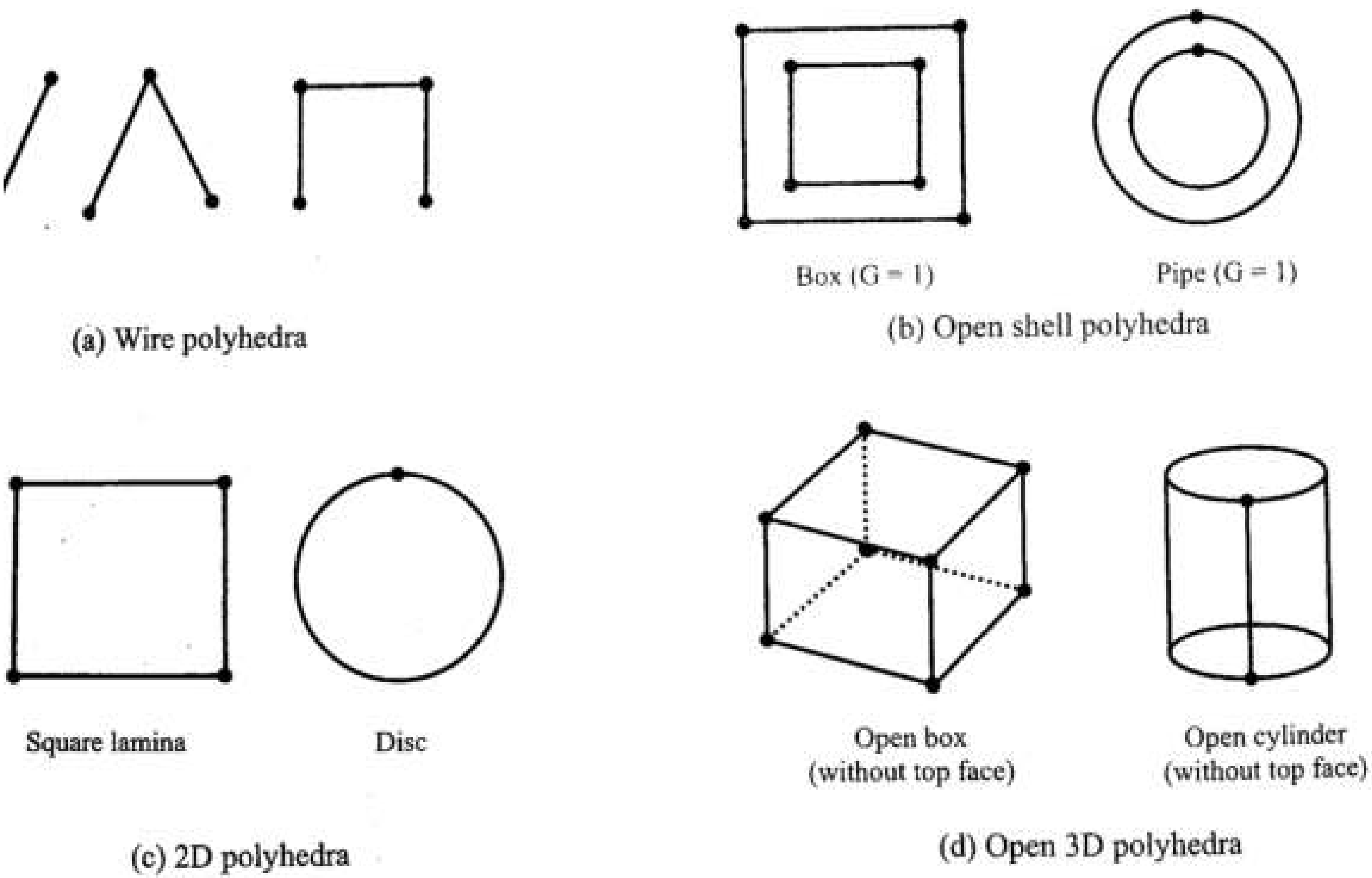


Fig. 9.33. Open polyhedral objects

Curved edges

- Representation is more complex
- Direct and indirect schemes
- Direct scheme – edge is represented by a curve equation and ordered endpoints
- Indirect scheme – edge is represented by the intersection of two surfaces

Building operations

- M – make and K – kill
- Euler equation forms the basis to develop building operations to create boundary models of complex objects
- MBFV, MEV, MEF, KFEVB, KEV, etc.
- In order to preserve topology
- Gluing operation can result in forming a genus or killing one body

Euler-Poincare law

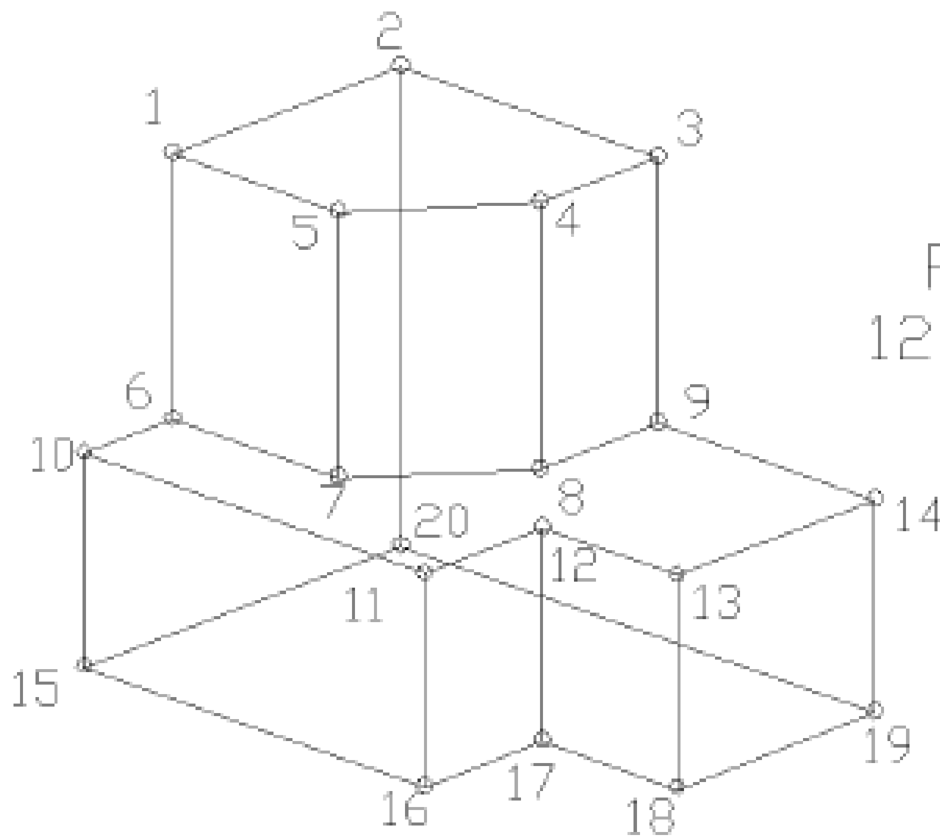
- Topologically valid if they satisfy the following equations:

$$F - E + V - L = 2\ (B - G)$$

- Above equation applies to closed polyhedral objects only

- For open objects,

$$F - E + V - L = B - G$$



$V=20$

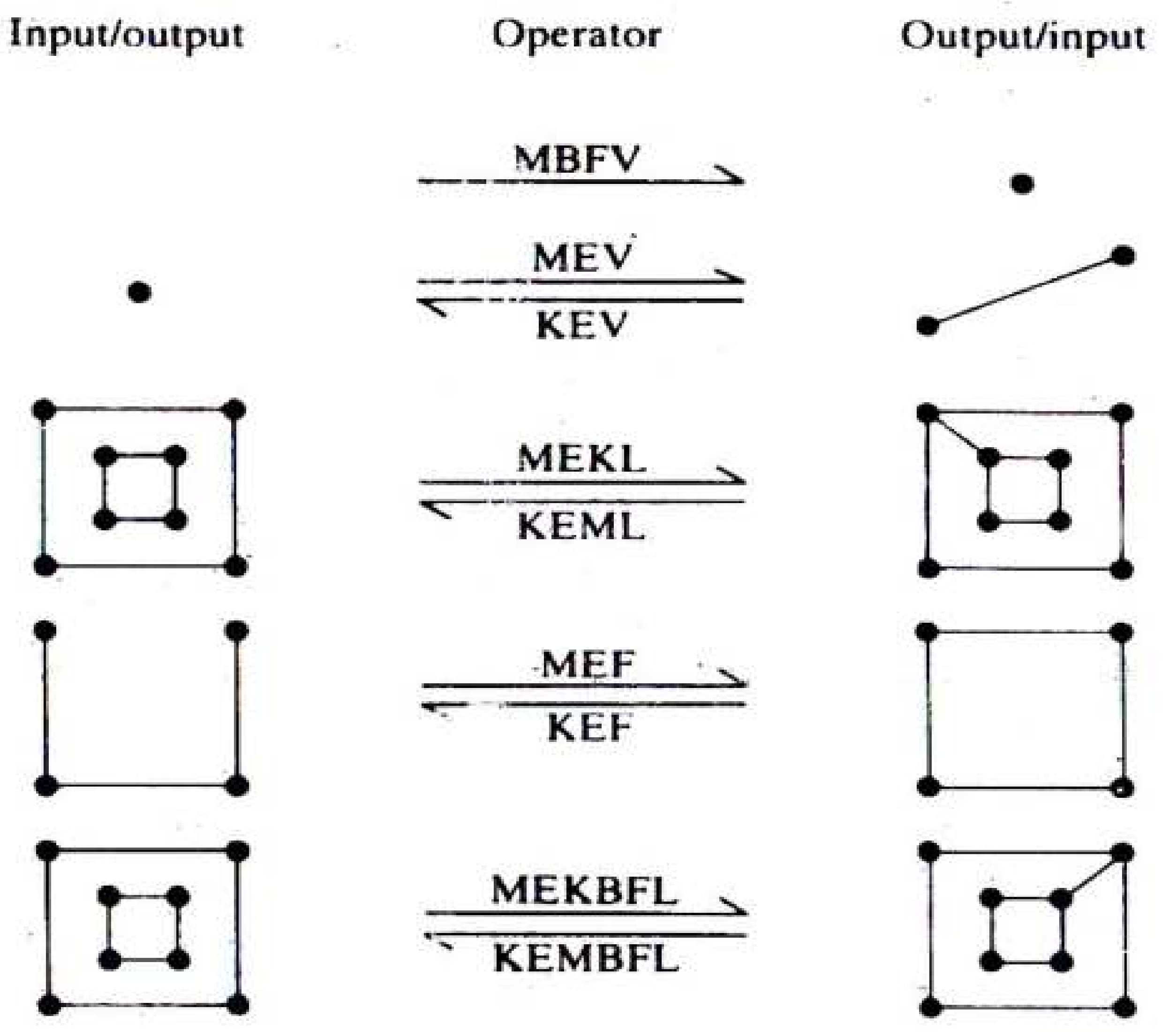
$E=30$

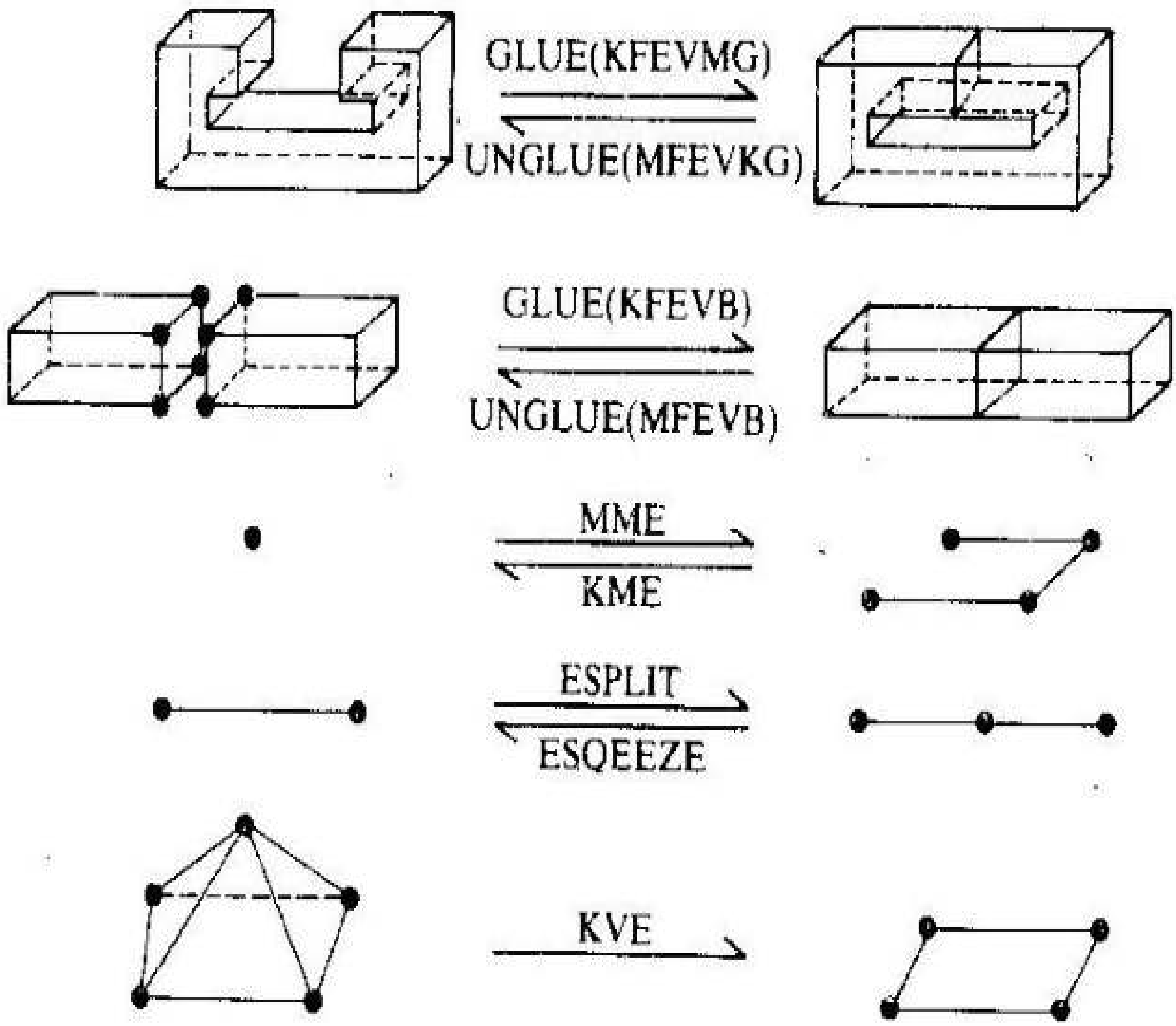
$F=12$

$F-E+V=2$

$12-30+20=2$

Building operations



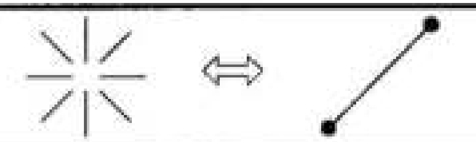
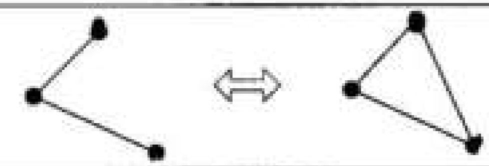
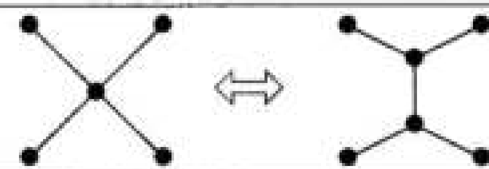
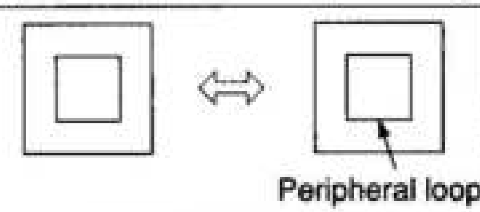

GLUE(KFEVMG)
UNGLUE(MFEVKG)
GLUE(KFEVB)
UNGLUE(MFEVB)
MME
KME
ESPLIT
ESQEEZE
KVE

Building operations

Some Euler operations

Operation	Operator	Complement	Description of operator
Initialize database and begin creation	MBFV	KBFV	Make Body, Face, Vertex
Create edges and vertices	MEV	KEV	Make Edge, Vertex
Create edges and faces	MEKL	KEML	Make Edge, Kill Loop
	MEF	KEF	Make Edge, Face
	MEKBFL	KEMBFL	Make Edge, Kill Body, Face, Loop
	MFKLG	KFMLG	Make Face, Kill Loop, Genus
Glue	KFEVMG	MFEVKG	Kill Face, Edge, Vertex, Make Genus
	KFEVB	MFEVB	Kill Face, Edge, Vertex, Body
Composite operations	MME	KME	Make Multiple Edges
	ESPLIT	ESQUEEZE	Edge-Split
	KVE		Kill Vertex, Edge

Euler Operators

MEVVLS (KEVVLS)	make (kill) edge, two vertices, loop, shell	
MEL (KEL)	make (kill) edge, loop	
MEV (KEV)	make (kill) edge, vertex	
MVE (KVE)	make (kill) vertex, edge	
MEKH (KEMH)	make (kill) edge, kill (make) hole	
MZEV (KZEV)	make (kill) zero length edge, vertex	
MPKH (KPMH)	make(kill) peripheral loop, kill (make) hole loop	

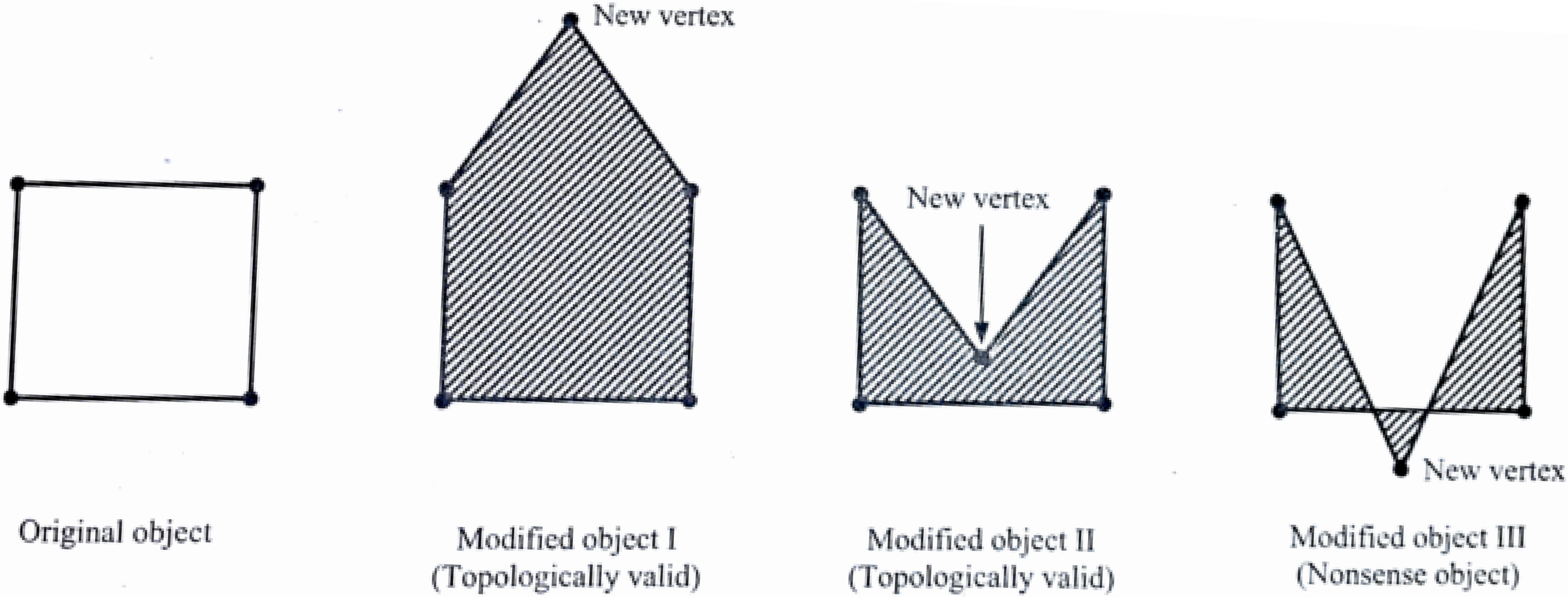


Fig. 9.30. Modified shapes of boundary model by introducing a new vertex

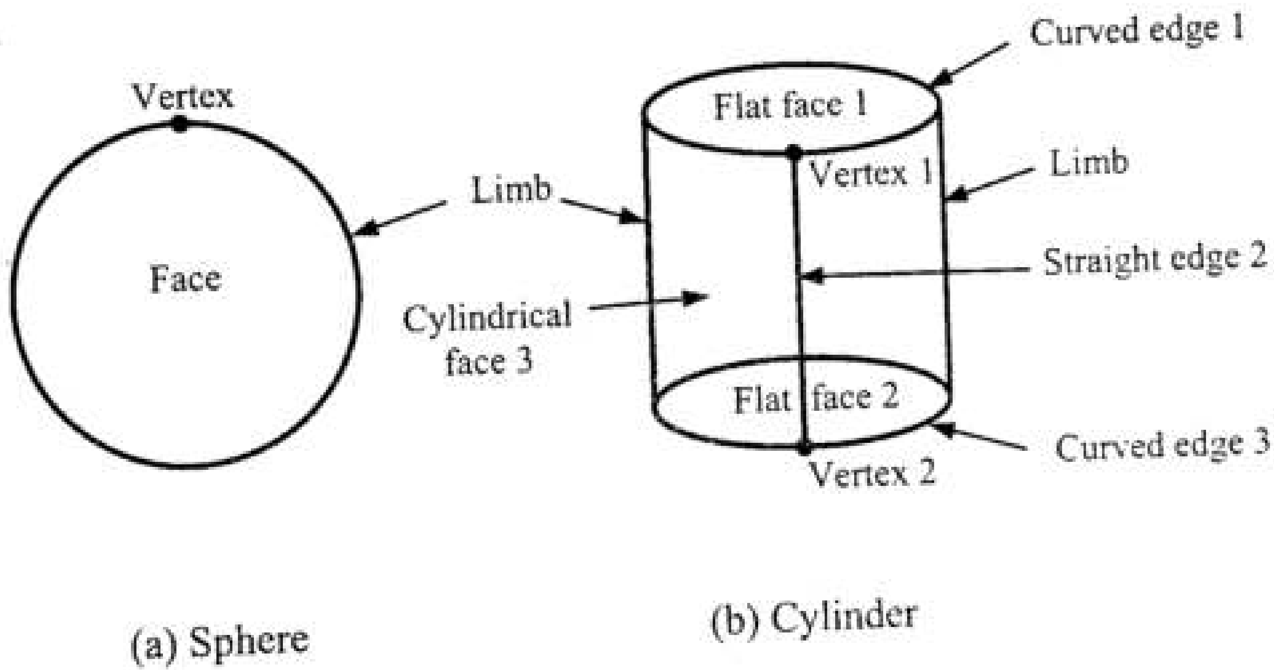


Fig. 9.34. Boundary models of curved polyhedral objects

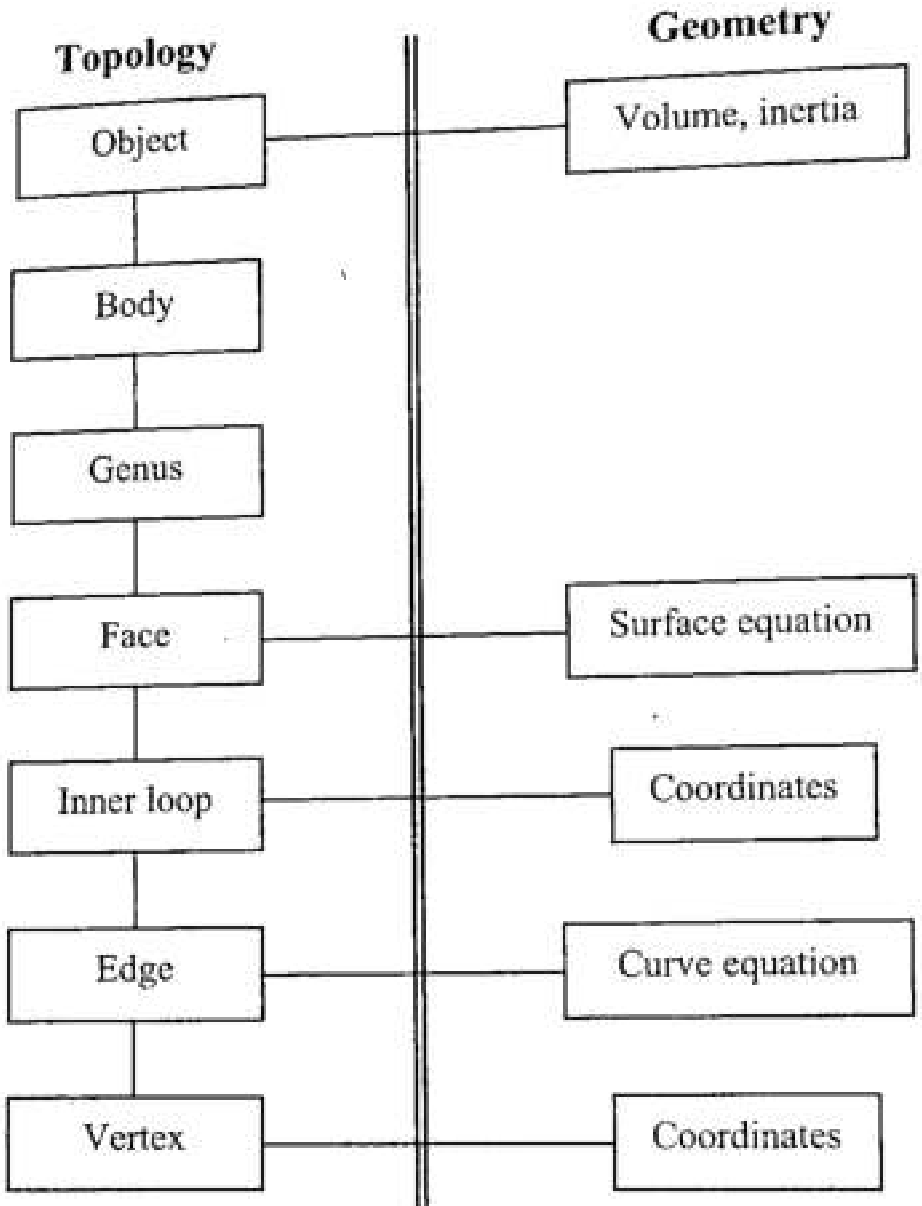


Fig. 9.35. Data structure of a boundary model

Table 9.1 Some Make group Euler's operators

Operator	Operation	*V*	*E*	*F*	*L*	*B*	*G*	Change in Euler-Poincaré law
MEV	Make Edge, Vertex	+1	+1	–	–	–	–	0
MFE	Make Face, Edge	–	+1	+1	–	–	–	0
MBFV	Make Body, Face, Vertex	+1	–	+1	–	+1	–	0
MME	Make Multiple Edges	–	+n	–	–	–	–	0
MFEVB	Make Face, Edge, Vertex, Body	+1	+1	+1	–	–	–	0
MEKL	Make Edge, Kill Loop	–	+1	–	–1	–	–	0
MEKBFL	Make Edge, Kill Body, Face, Loop	–	+1	–1	–1	–1	–	0
MFKLG	Make Face, Kill Loop, Genus	–	–	+1	–1	–	–1	0
MFEVKG	Make Face, Edge, Vertex, Kill Genus	+1	+1	+1	–	–	–1	0

Table 9.2 Some Kill group Euler's operators

Operator	Operation	*V*	*E*	*F*	*L*	*B*	*G*	Change in Eu Poincaré la
KEV	Kill Edge, Vertex	–1	–1	–	–	–	–	0
KFE	Kill Face, Edge	–	–1	–1	–	–	–	0
KBFV	Kill Body, Face, Vertex	–1	–	–1	–	–1	–	0
KME	Kill Multiple Edges	–	–n	–	–	–	–	0
KFEVB	Kill Face, Edge, Vertex, Body	–1	–1	–1	–	–	–	0
KEML	Kill Edge, Make Loop	–	–1	–	+1	–	–	0
KEMBFL	Kill Edge, Make Body, Face, Loop	–	–1	+1	+1	+1	–	0
KFMLG	Kill Face, Make Loop, Genus	–	–	–1	+1	–	+1	0
KFEVMG	Kill Face, Edge, Vertex, Make Genus	–1	–1	–1	–	–	+1	0

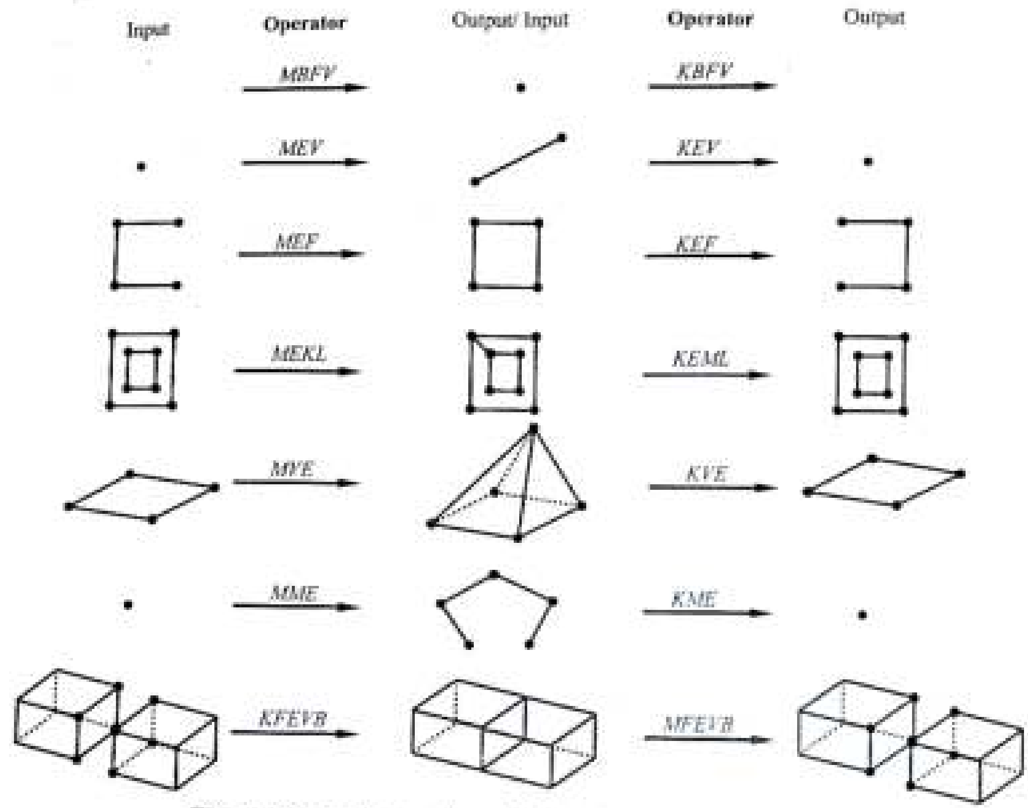


Fig. 9.36. Topology creations of objects through Euler's operators

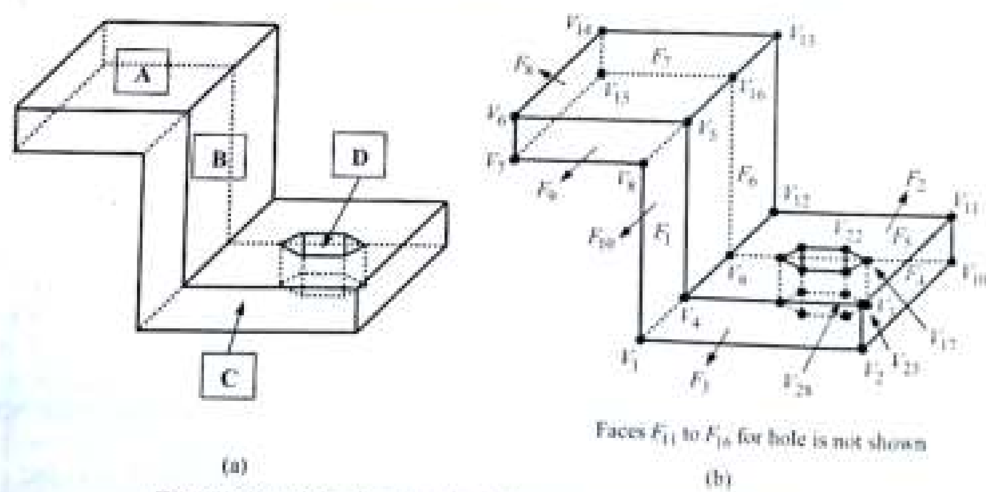


Fig. 9.37. (a) Simple polyhedral object (b) boundary model of solid S

Table 9.3 *Faces and vertices of boundary model of solid S*

Description of Faces	**Face**	**Vertices**
Front face	F_1	V_1 to V_8
Back face	F_2	V_9 to V_{16}
Bottom face of Part C	F_3	$V_1 - V_2 - V_{10} - V_9$
Right side face of Part C	F_4	$V_2 - V_{10} - V_{11} - V_3$
Top face of Part C	F_5	$V_4 - V_3 - V_{11} - V_{12}$
Right side face of Part B	F_6	$V_4 - V_{12} - V_{13} - V_5$
Top face of Part A	F_7	$V_5 - V_{13} - V_{14} - V_6$
Left side face of Part A	F_8	$V_6 - V_7 - V_{15} - V_{14}$
Bottom face of Part A	F_9	$V_7 - V_8 - V_{16} - V_{15}$
Left side face of Part B	F_{10}	$V_1 - V_9 - V_{16} - V_8$
First face of hexagonal hole	F_{11}	$V_{17} - V_{18} - V_{24} - V_{23}$
Second face of hexagonal hole	F_{12}	$V_{18} - V_{19} - V_{25} - V_{24}$
Third first of hexagonal hole	F_{13}	$V_{19} - V_{20} - V_{26} - V_{25}$
Fourth face of hexagonal hole	F_{14}	$V_{20} - V_{21} - V_{27} - V_{26}$
Fifth face of hexagonal hole	F_{15}	$V_{21} - V_{22} - V_{28} - V_{27}$
Sixth face of hexagonal hole	F_{16}	$V_{22} - V_{17} - V_{23} - V_{28}$

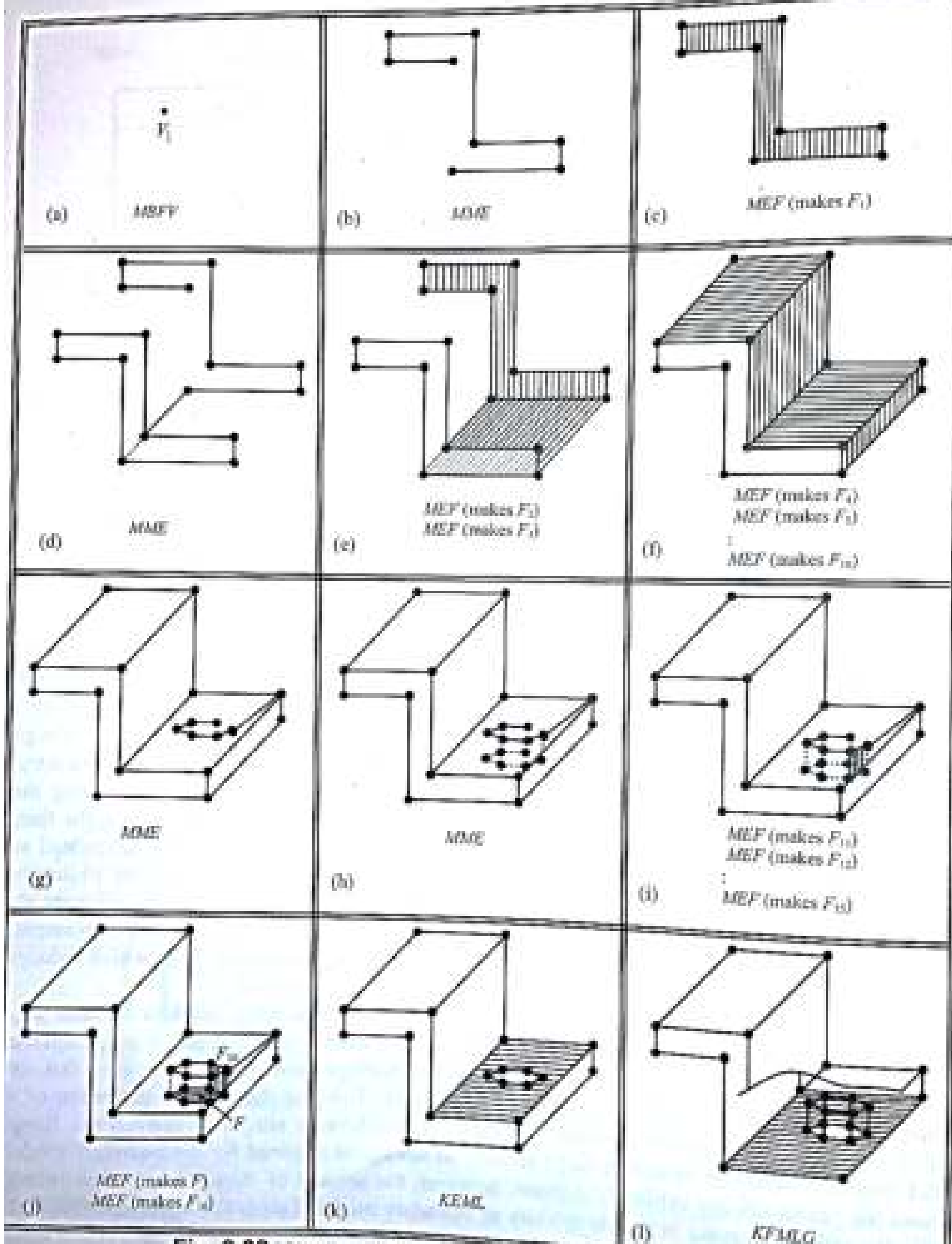


Fig. 9.38. Steps (a – l) for creation of boundary model of solid S.

Spatial-partitioning representations

Describe objects as collections of adjoining nonintersecting solids

- Spatial-partitioning creates collections of solids that,
  - may or may not be the same type as the original object
  - are like building blocks
  - can vary in type, size, position, parameterization, and orientation

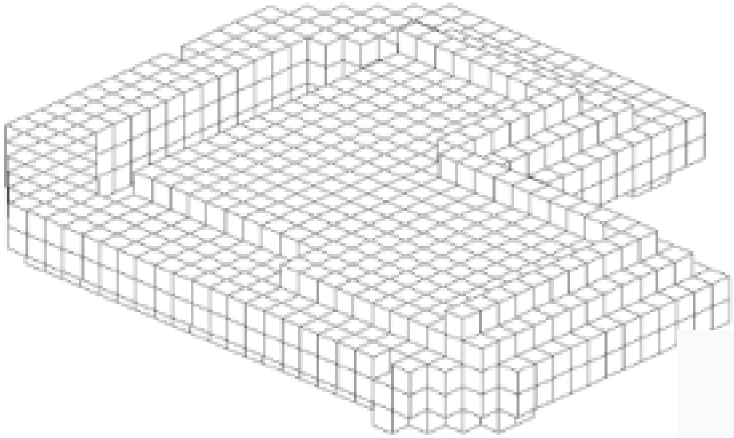
- Solid objects can be formed with spatial-partitioning using,
  - cell decomposition
  - spatial-occupancy enumeration
  - octrees and quadtrees, etc.

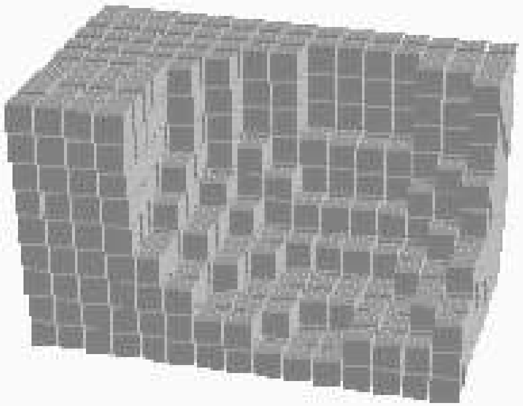
Spatial-occupancy enumeration

- It’s a special case of decomposition method

- In decomposition methods, a solid is decomposed into a collection of adjoining, non-intersecting solid primitives.
- Two of the important methods under this category are:

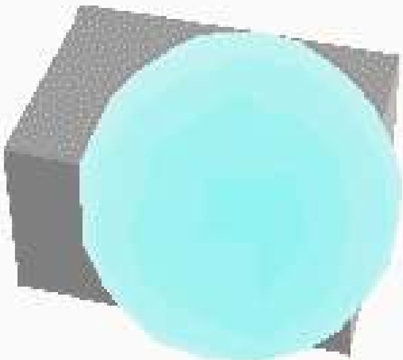
    Octree Based Modeling
    Voxel Based Modeling





Spatial Enumeration model of a prismatic block with a spherical shape removed.

Below are the parent shapes



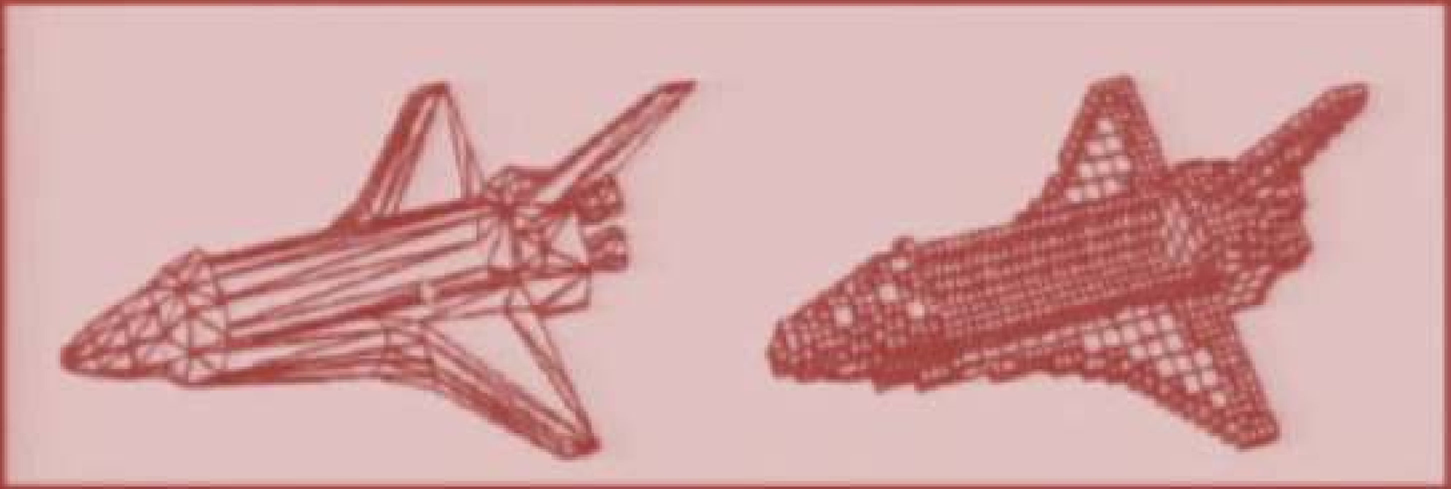
Voxel based Spatial-occupancy enumeration

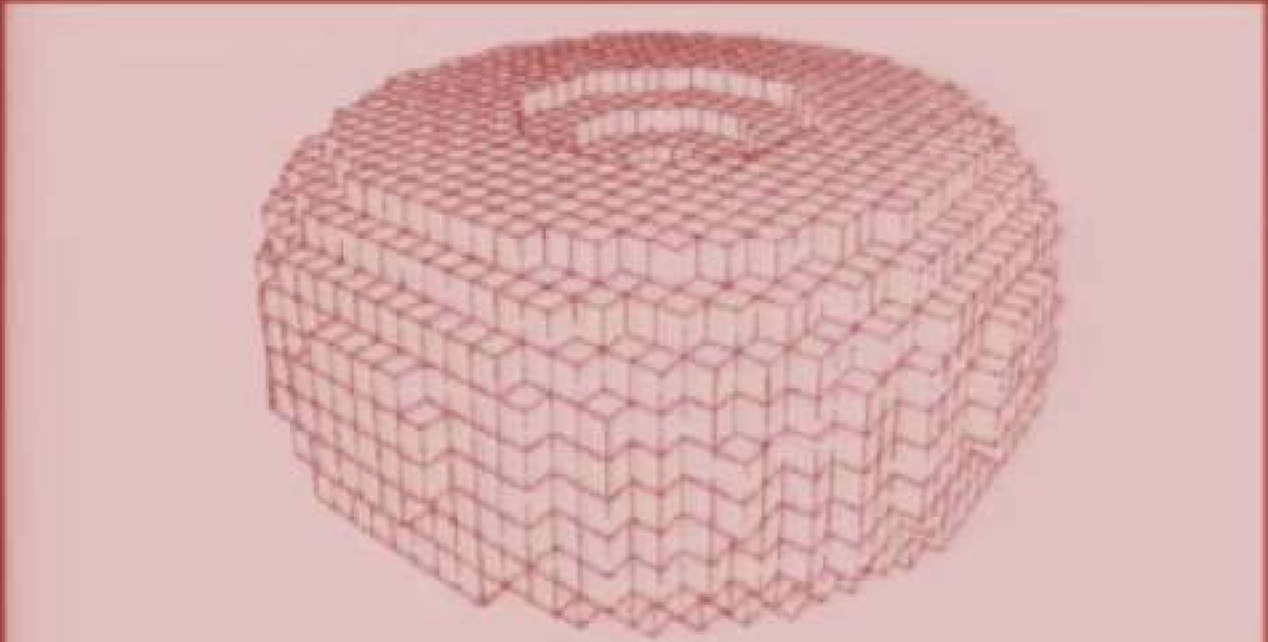
- Solid point set represented by collection of non-overlapping “blocks”
- Blocks are “pasted” together to create geometry.
- 3d space decomposed into a set of identical cells.
  - Most commonly used geometry for cells is the cube.
- Cells are located by their centers, within a fixed 3-dimensional grid (xyz space).

Voxel based Spatial-occupancy enumeration

- Defines objects using identical cells arrayed in a fixed and regular grid (called *voxels*)
- Most commonly uses a cube cell type
- Only controls whether or not a cell is present or absent in every cell in a grid
- No other controls are defined
- Creates unique and unambiguous list of occupied cells

DECOMPOSITION METHODS



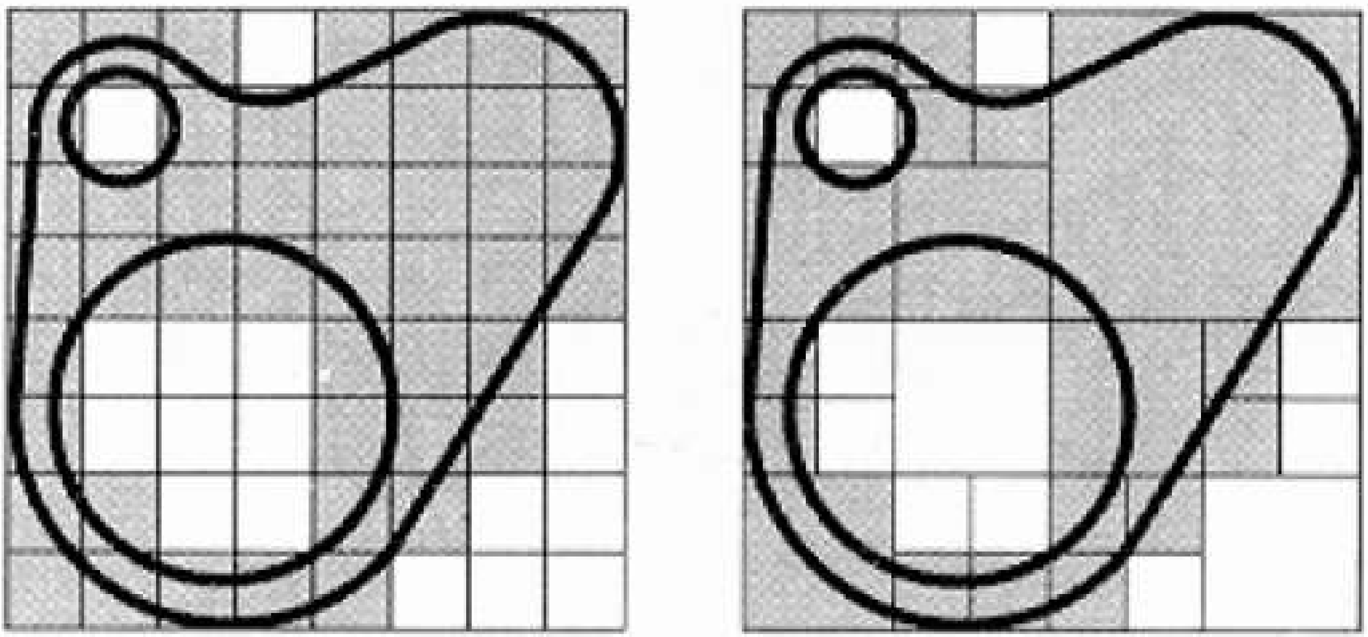


Quadtrees based Spatial-occupancy enumeration

- Quadtrees
  - Successively subdivide a 2D plane in both dimensions
  - Where each quadrant is full, partly full, or empty depending on how much of the complex object intersects the area
  - Where partly full quadrants are recursively subdivided
  - And subdivision continues until all cells are full or empty

a) 2D spatial enumeration of shape shown.

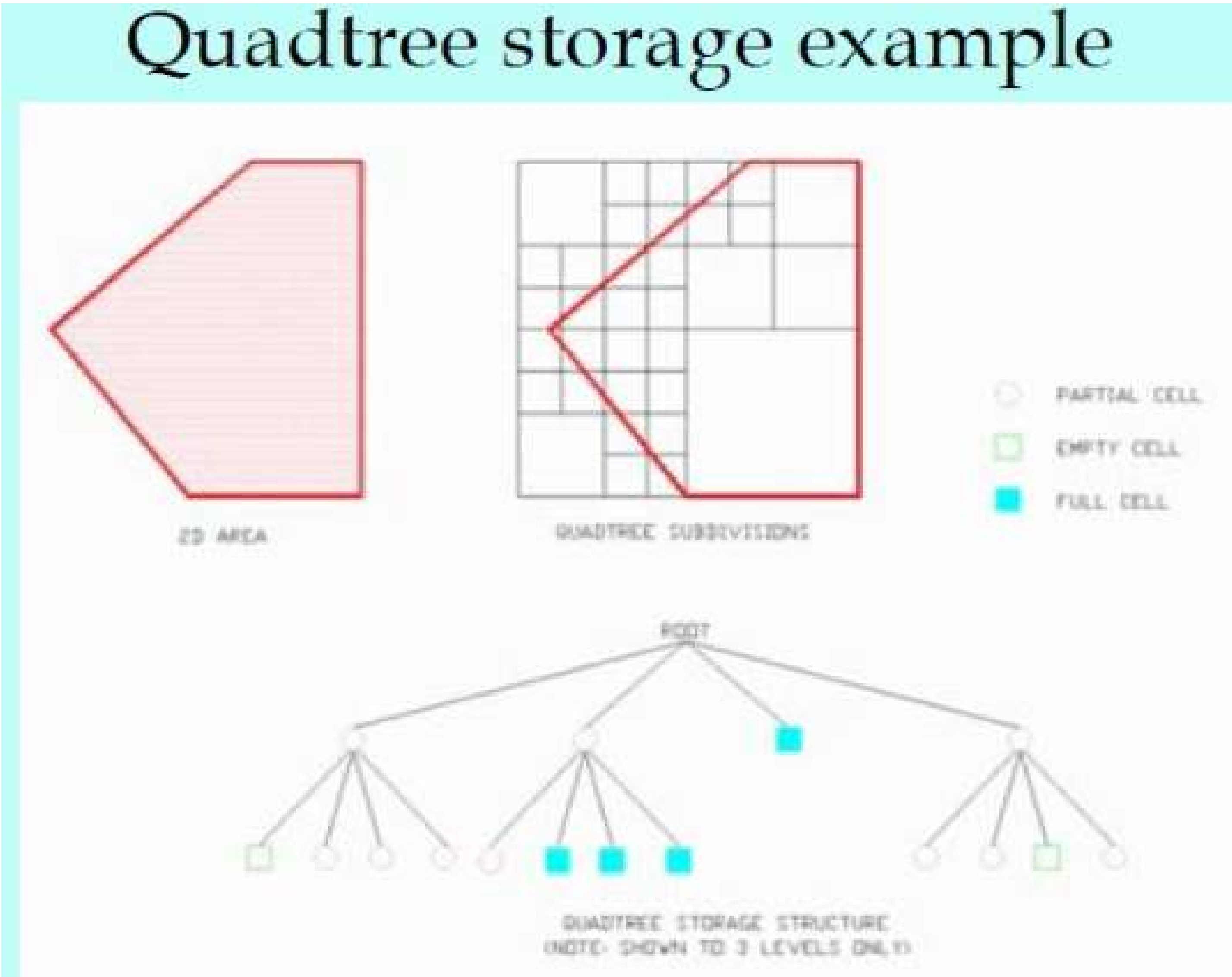
b) Quadtree representation of the same.



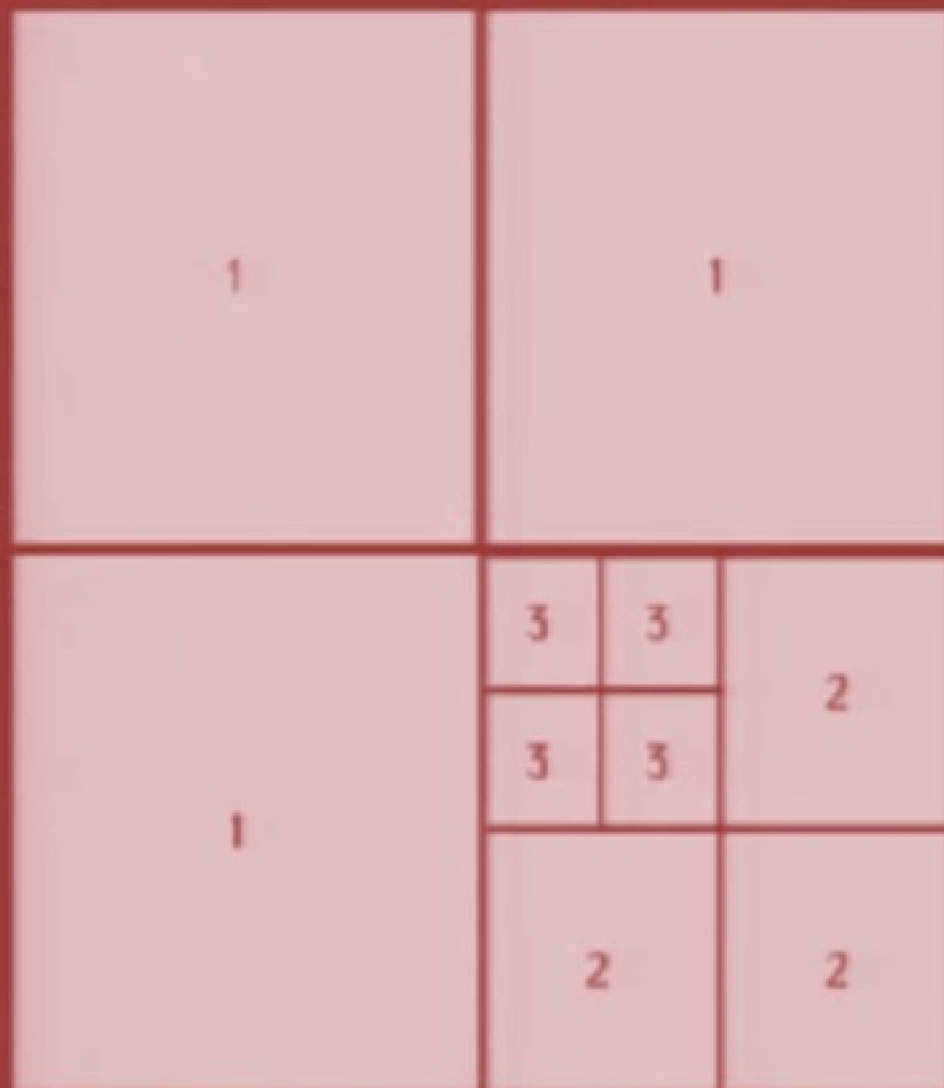
(a) (b)

Voxel Quadtrees

Quadtrees


Quadtree storage example
2D AREA
QUADTREE SUBDIVISIONS
PARTIAL CELL
EMPTY CELL
FULL CELL
ROOT
QUADTREE STORAGE STRUCTURE
(NOTE: SHOWN TO 3 LEVELS ONLY)

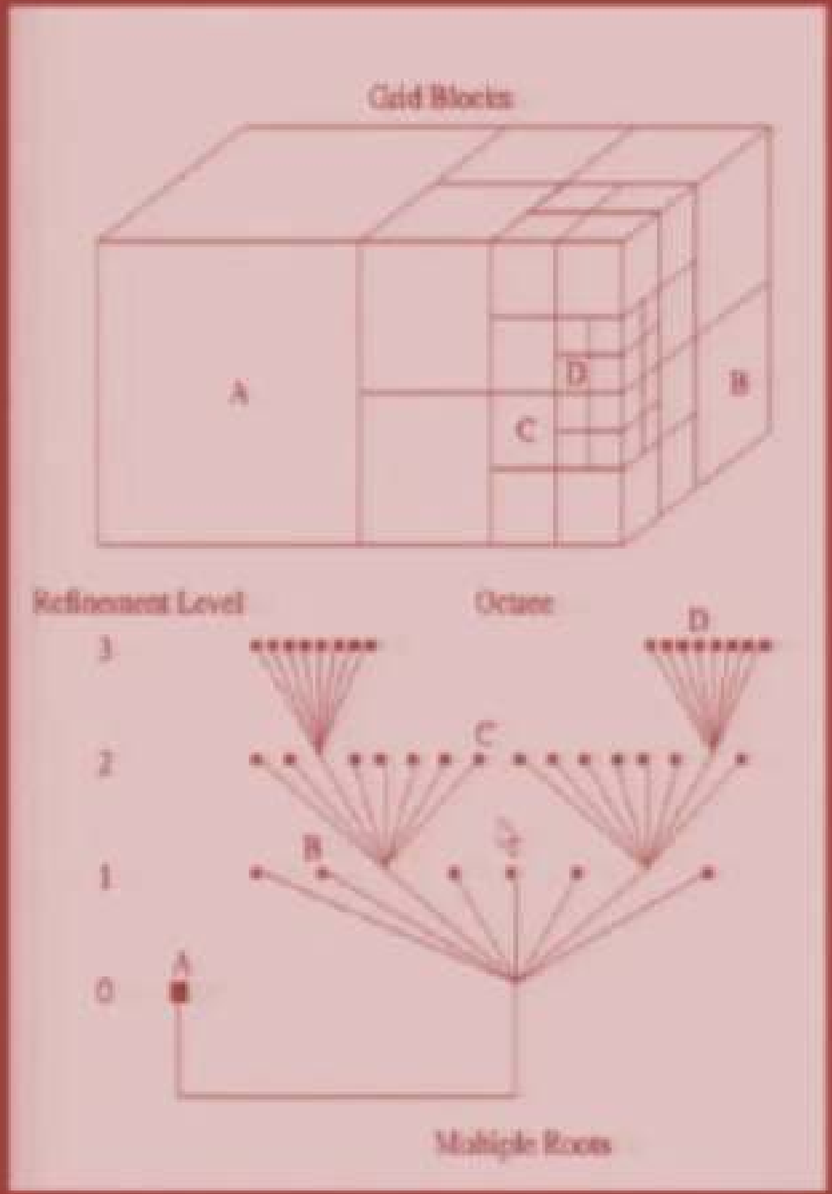
DECOMPOSITION METHODS



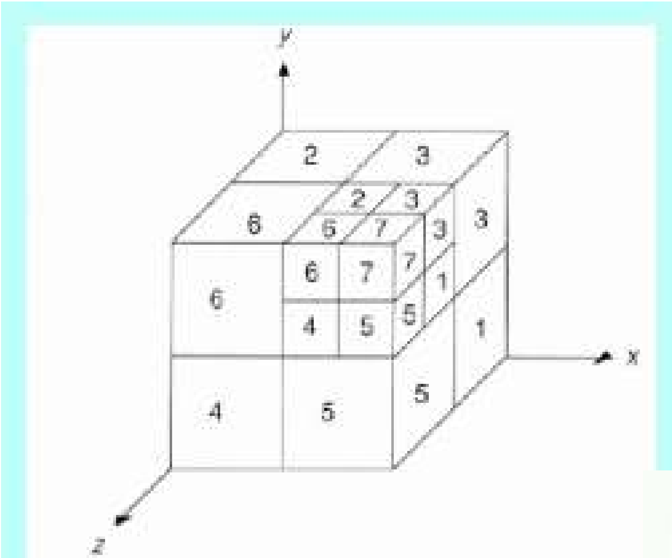
Octrees
based Spatial-occupancy enumeration

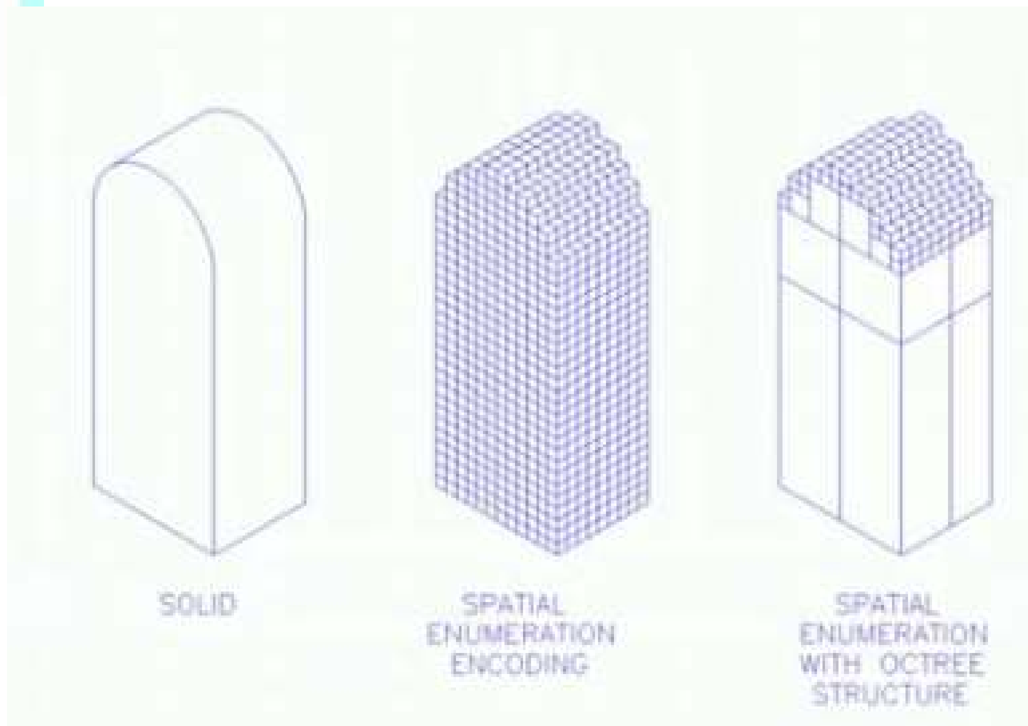
- Octrees
  - Are a hierarchical way to use voxels
  - Octrees represent the 3 dimensional extension of the quadtree concept.
  - Expand quadtree (2d) to 3d space
  - Spatial volumes are sub-divided into a set of eight cells or octants.
  - Are designed to reduce the storage requirements of the spatial-occupancy enumeration approach
  - Are derived from 2D quadtrees...And expanded to 3d octrees

DECOMPOSITION METHODS



octree


y
x
z
2
3
6
2
3
6
7
3
3
6
7
7
1
6
4
5
5
1
4
5
5


SOLID
SPATIAL ENUMERATION ENCODING
SPATIAL ENUMERATION WITH OCTREE STRUCTURE

Advantage of decomposition

- Volume calculation
- Solid intersection or collision checking
- Convenient method for storage

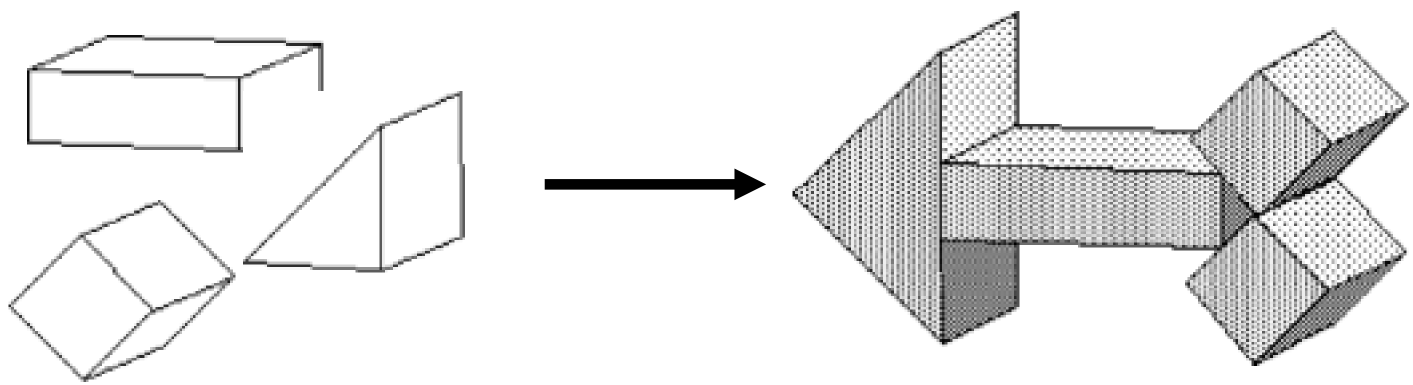
Disadvantage of decomposition

- Not a Convenient method for construction of solid
- Approximation method

Cell decomposition

- Special decomposition method.
- In cell decomposition method , the object divided In the form of a cell.
- All cells are made of box like objects, but need not be same size and shape.
  - In octree cells are of different size but same shape
  - In voxel cells are of same size and shape
- But in cell decomposition method , cells can be of different shape and size.

- Cell decomposition...
  - is a popular form of spatial-partitioning
  - composes complex objects from simple primitives in a bottom-up fashion by gluing them together! (like a union but without objects intersecting)
  - composes objects from cells, where any two cells must share a single point, edge, or face



Three simple primitives: called cells

Keep in mind with this method, that the complex object can be created using cells in more than one way...

Coordinate system for Solid modeling

- World Coordinate system
- User Coordinate system

MODULE 5

Introduction to

Finite Element Analysis

SYLLABUS

V	Introduction to finite element analysis - steps involved in FEM-Preprocessing phase – discretisation - types of elements	2
	Formulation of stiffness matrix (direct method, 1-D element) - formulation of load vector - assembly of global equations - implementation of boundary conditions - solution procedure - post processing phase	3
	Simple problems with axial bar element (structural problems only)	2

Objective

To introduce the concepts of finite element analysis procedures

Outcome

Students will have the knowledge about mathematical background of finite element analysis

- CAE COMPUTER AIDED ENGINEERING (CAE) CAE is a technology concerned with the use of computer systems to analyze CAD geometry, allowing designers to simulate and study how the product will behave so that the design can be validated, refined and, optimized.
- A typical CAE process comprises of pre-processing, solving, and post processing steps.
- In the pre-processing phase, engineers model the geometry and the physical properties of the design, as well as the environment in the form of applied loads or constraints.
- Next the model is solved using an appropriate mathematical formulation of the underlying physics.
- In the post-processing phase, the results are presented to the engineer of review.

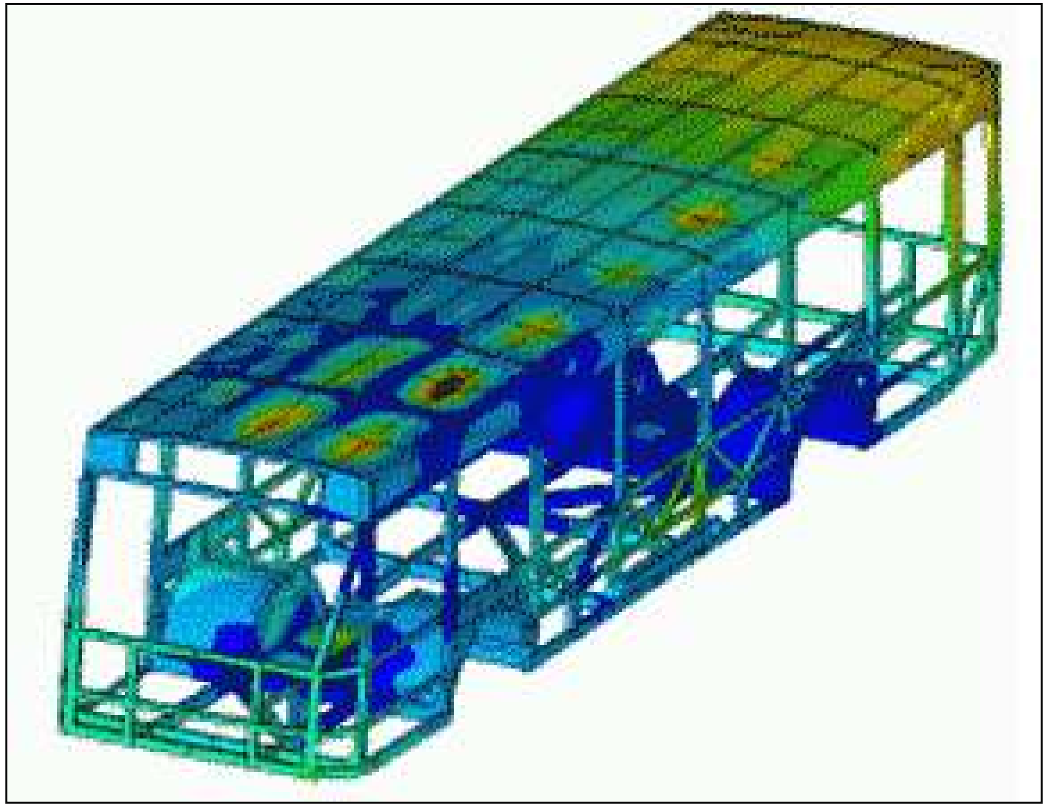
BENEFITS OF CAE

- Designs decision can be made based on their impact on performance.
- Designs can be evaluated and refined using computer simulation rather than physical prototype testing saving money and time.
- CAE can provide performance insights earlier in the development process, when design changes are less expensive to make.

CAE APPLICATION

- Stress and dynamic analysis on components and assemblies and finite element analysis (FEA)
- Thermal and fluid analysis using computational fluid dynamics (CFD)
- Kinematics and dynamic analysis of mechanisms (multibody dynamics)
- Simulation of manufacturing processes like casting,molding and die press forming
- Optimization of the product or process

INTRODUCTION TO FEM



Methods to Solve Any Engineering Problem

Analytical Method	Numerical Method	Experimental Method
Classical Approach 100% Accurate Results Applicable only for Simple problems like Cantilever , simply supported beams and Cylinders etc.. Complete in itself	**Mathematical Approach** **Approximate, Assumptions Made** **Applicable to real life complicated problems** **Results can not be believed blindly and must be verified by experimental methods and Hand Calculations.**	Actual Measurement Time Consuming , Needs expensive setup Applicable only if physical prototype is available Results can not be believed blindly and Minimum 2 or more prototypes must be tested.
Although applicable to simple shaped geometries only , Analytical methods are considered as Closed form solutions i.e. 100% Accurate	**Finite Element Method:** Linear , Nonlinear ,Buckling ,Thermal, Dynamics & Fatigue analysis. **Boundary Element Method:** Acoustics / NVH analysis **Finite Volume Method:** CFD (Computational Fluid Dynamics) & Computational Electromagnetic. **Finite Difference Method:** Thermal & Fluid Flow analysis (in combination with FVM)	-Strain Gauge - Photo elasticity - Vibration measurement (accelerometers) - Sensors for Temp & pressure etc… - Fatigue test

Different Numerical Methods

Finite Element Method (FEM) :

- Very Popular Method based upon discretization of component into blocks (elements) of Finite dimensions.
- Applications : Linear, Nonlinear, Thermal, Dynamics , Buckling and Fatigue Analysis

Boundary Element Method (BEM) :

- It's a very powerful and efficient technique to solve acoustics and NVH problems
- Just like Finite Element Method, it also requires Nodes and Elements but as the name suggest, it considers only the outer boundary of the domain.

Finite Volume Method (FVM) :

- All Computational Fluid Dynamics (CFD) softwares are based upon FVM.
- Unit Volume is considered in Finite Volume Method (similar to Elements in FEM)
- Variable properties at nodes are Pressure , Velocity , Area , Mass etc.
- It is based on Navier – Stoke equations (Mass ,Momentum and Energy Conservation equations)

Finite Difference Method (FDM) :

- Finite Element and Finite Difference share many common things.
- In general, FDMis described as a way to solve differencial equation.
- Discretizations used for solving differential equations by approximating them with difference equations
- It uses Taylor’s series to convert differential equation into algebraic equation. Higher order terms neglected.

Is it possible to use all the above listed methods (FEA ,BEM , FVM, FDM) to solve same problem (say Cantilever problem)?

- Answer : YES ! But the difference is in Accuracy achieved , programming ease and time required to obtain the solution are different.

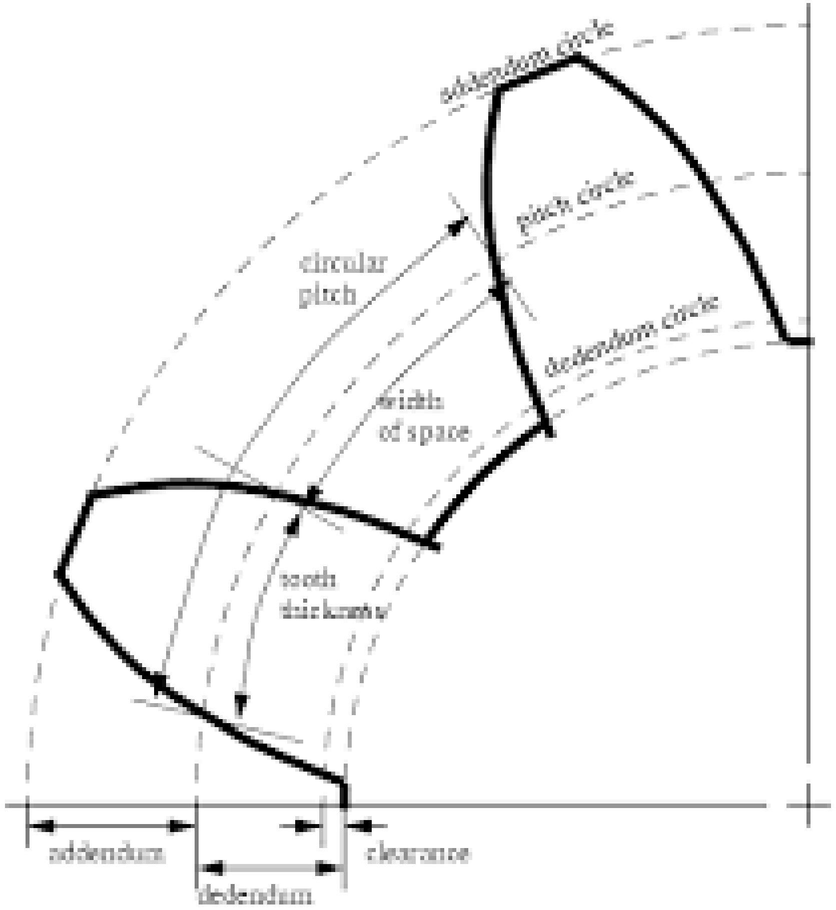
Are FEA and FEM different ?

- Finite Element Analysis (FEA) and Finite Element Method (FEM) both are one & the same.
- FEA is a method/process based upon FEM
- Term “FEA” is more popular in industries while “FEM” at Education centers.

- Finite element method is a **numerical method** for solving problems of Engineering and Mathematical Physics.
- In this method, a body or a structure in which the analysis to be carried out is **subdivided into smaller elements** of finite dimensions called **finite elements**. Then the body is considered as an assemblage of these elements connected at a finite number of joints called **'Nodes' or Nodal points.** The properties of each type of finite element is obtained and assembled together and solved as whole to get solution.
- In other words, in the finite element method, instead of solving the problem for the entire body in one operation, we formulate the equations for **each finite element and combine them to obtain the solution of the whole body.**
- Finite element method is used to solve physical problems involving complicated geometrics: loading and material properties which cannot be solved by analytical method.

INTRODUCTION TO FEM

- This method is extensively used in the field of structural mechanics, fluid mechanics, heat transfer, mass transfer, electric and magnetic fields problems.



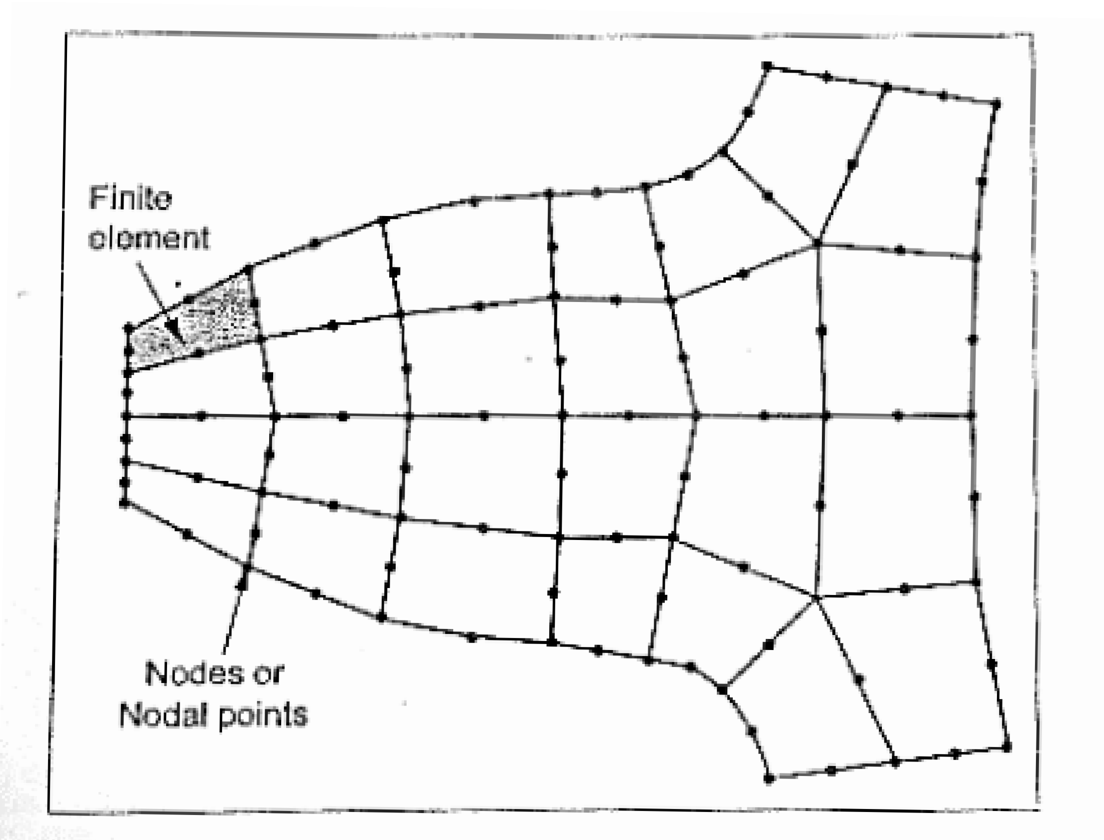
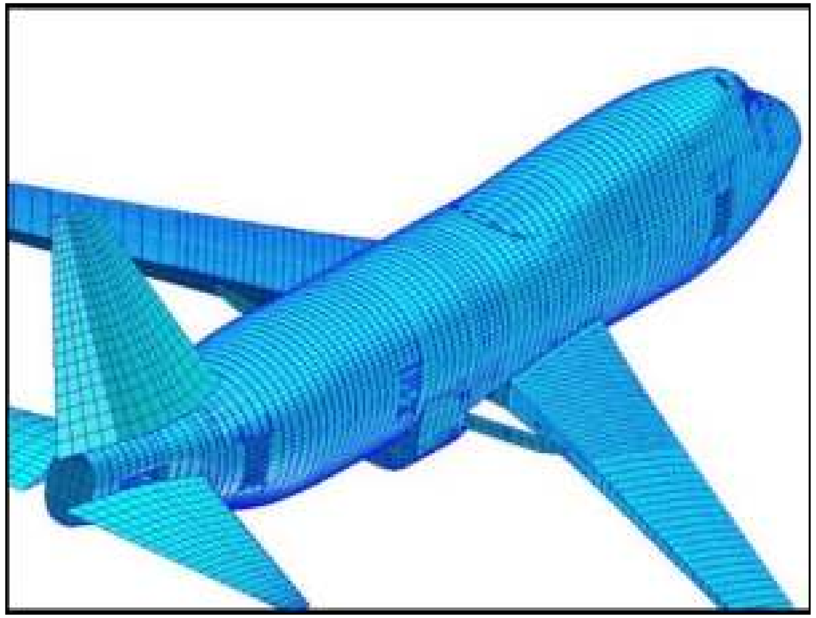
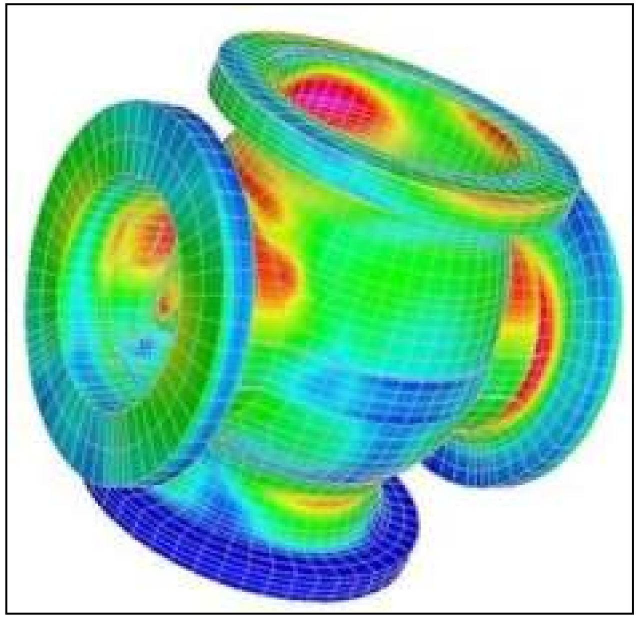


Fig. 1.1. Finite element discretization of spur gear teeth

Why Finite Element Method ?

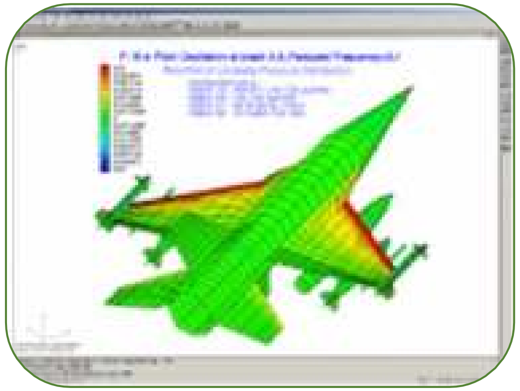
- FEA is the most widely applied **computer simulation** method in Engineering.
- It is very closely integrated with CAD/CAM applications.
- It is very well proven , tested and validated method for simulating any complex practical scenario in the area of Structural ,Thermal ,Vibration etc..

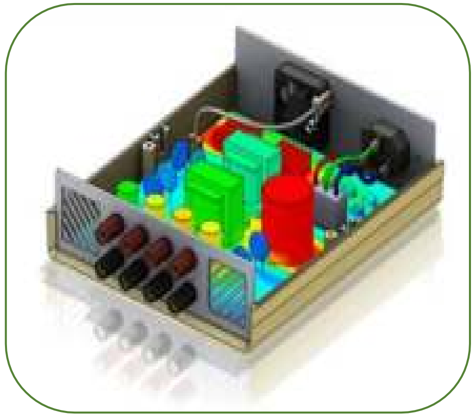




Application of FEM in Engineering

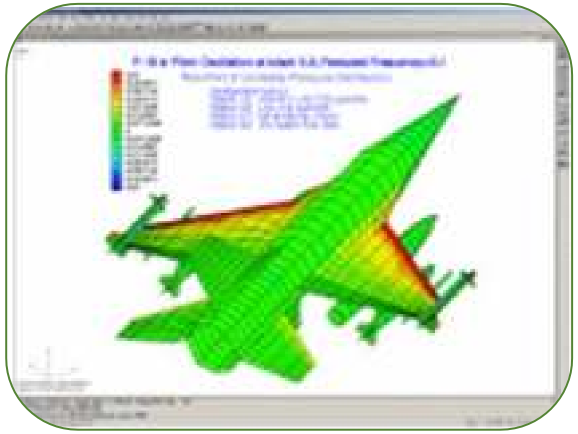
- Mechanical / Aerospace / Civil Engineering / Automobile Engineering
- Structural Analysis (Static / Dynamic , Linear / Non-Linear)
- Thermal Analysis (Steady State / Transient)
- Electromagnetic Analysis
- Geomechanics
- Biomechanics



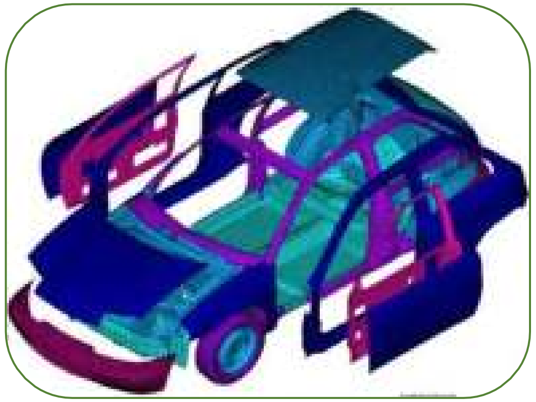


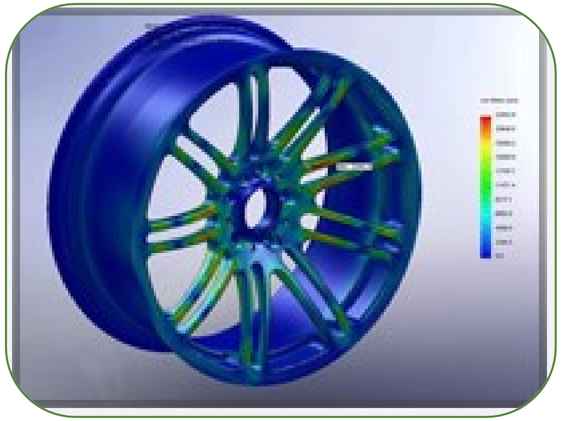
Practical Applications of FEA

- Aerospace Domain



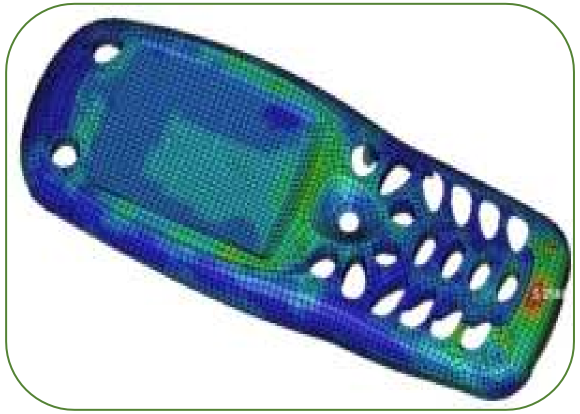
- Automotive Domain

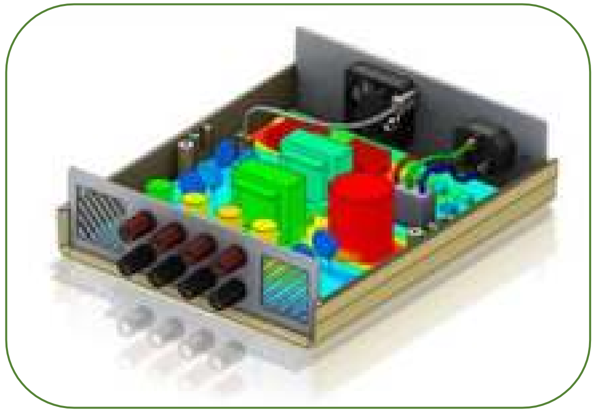




INTRODUCTION TO FEM

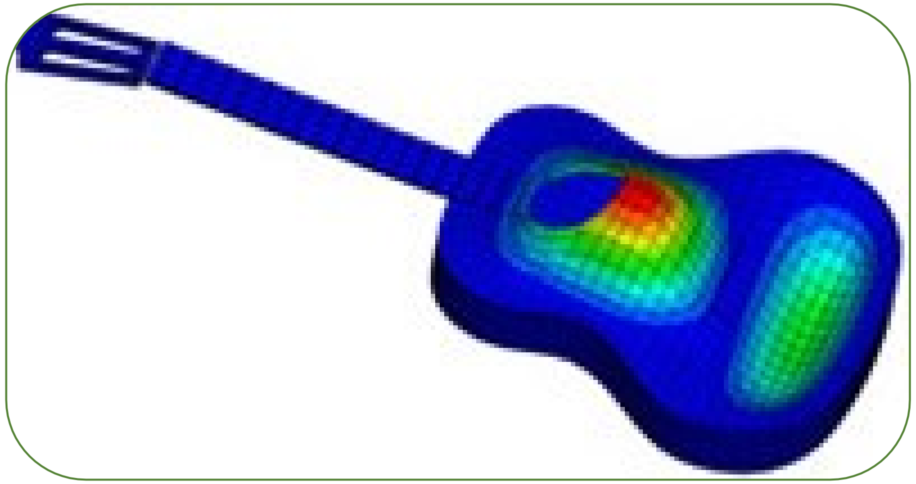
- Hi-Tech /Electronics

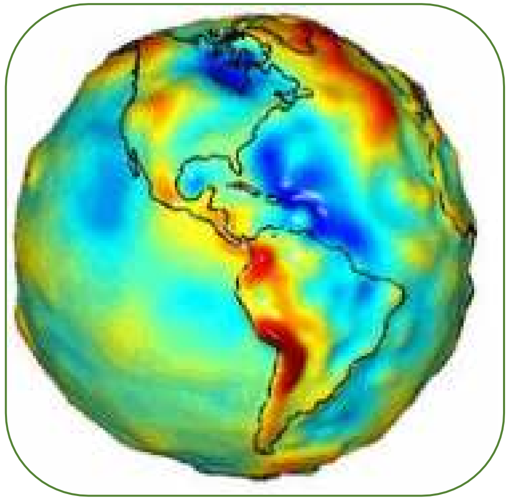




- Medical Devices

and many more ….





APPLICATIONS OF FEA

The heart and power of the FEA is that it can **readily handle very complex geometry**. The general nature of its theory makes it applicable to a wide variety of problems in engineering.

- Static analysis of trusses, beams, frames, plates, bridges. machine **structures.**
- Structural analysis of **aircraft wings**. missile and **rocket** structures.
- **Natural frequencies** and modes of structures, linkages, gears, flywheels. and cams.
- **Dynamic response** of structures subjected to a periodic loads and random loads.
- **Stress analysis** of pressure vessels, flywheels, crankshafts, cams, linkages, gears, machine members. etc.

APPLICATIONS OF FEA

- **Thermal analysis** in IC engines, turbine blades, steam pipes, and rocket nozzles.
- **Analyses of fluid flow** and wave propagation problems.
- Steady state analysis of synchronous and induction machines, **eddy current** and core losses in electric machines.
- Stress analysis of **bones and teeth**, load bearing capacity of implant and prosthetic systems, and **mechanics of heart values**.
- Analysis of casting, forming. welding and machining processes.
- **Analysis of earthquakes**.

- Based on the application, the finite element problems are classified as,
  - Structural problem
  - Non- structural problem

Structural problems: In structural problems, displacement at each nodal point is obtained. By using these displacement solutions, stress and strain in each element can be calculated.

Non-structural problems: In non-structural problems, temperature or fluid pressure at each nodal point is obtained. By using these values, properties such as heat flow, fluid flow etc., for each element can be calculated.

Advantages of FEA

- As previously indicated, the FEM has been applied to numerous problems, both **structural and nonstructural.**
- The ability to model **complex shaped bodies** quite easily
- **Handle several load conditions** without difficulty
- Handle different kinds of **boundary conditions**
- Model bodies composed of **several different materials**
- **Discretize the bodies with combination of different elements** because the element equations can be evaluated individually.

- **Cost** ↓
- **Design Cycle time** ↓
- **No. of Prototypes** ↓
- **Testing** ↓
- **Design Optimization** ↑

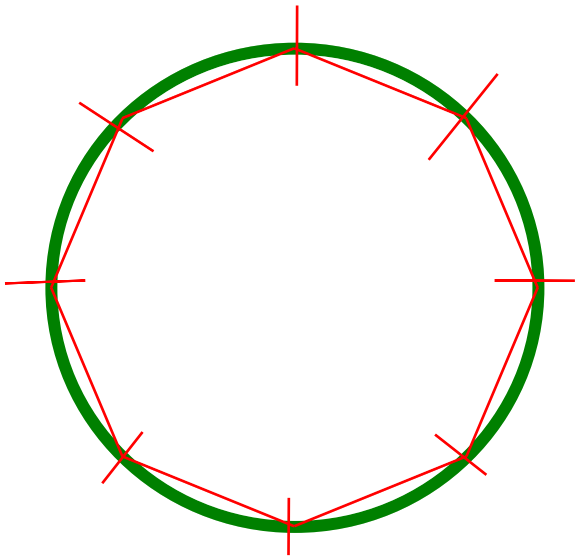
Disadvantages of the FEA

- A specific numerical result is obtained for a specific problem. A general closed-form solution which would permit one to examine system response to changes in various parameters is not produced.
- The FEA is applied to an approximation of the mathematical model of a system. so it obtains only "**approximate" solutions.**
- **Experience and judgment** are needed in order to construct a good finite element model.
- **Mistakes by users** while analyzing the problem in software can be fatal.
- A **powerful computer** and reliable FEA software are essential.
- **Input and output data** may be large and tedious to prepare and interpret.

Available Commercial FEA Tools/Software Packages

- *ANSYS* (General purpose, PC and workstations)
- *SDRC/I-DEAS* (Complete CAD/CAM/CAE package)
- *NASTRAN* (General purpose FEA on mainframes)
- *ABAQUS* (Nonlinear and dynamic analyses)
- *COSMOS* (General purpose FEA)
- *ALGOR* (PC and workstations)
- *PATRAN* (Pre/Post Processor)
- *HyperMesh* (Pre/Post Processor)
- *Dyna-3D* (Crash/impact analysis)

- The Philosophy of FEA can be explained with a small example such as " **Measuring the Perimeter of a Circle**"
- If one need to evaluate the perimeter of a circle without using the conventional formula (2ΠR), FEA approach is analogous to **Dividing the circle into a number of segments and joining the points using Straight lines**
- Since it is very easy to measure the length of straight line. Measure the length of one line and multiply it by No. of lines to get the perimeter.



Approximate results….isn't it ?

What if we want to achieve more accurate result?

Steps involved in FEA(structural problems)

- There are two general methods associated with FEA structural problems
  - Force method
  - Displacement method or stiffness method
- In force method, internal forces are considered as unknowns of the problem
- In displacement method, displacements of the nodes are considered as unknowns of the problem.
- **But displacement formulation is commonly used for solving structural problems**

FEM STEPS

1. Discretization of domain or continuum or structure(Mesh formation)
2. Identification of variables
3. Selection of Displacement or interpolation or shape function
4. Define the material behavior by using strain-displacement or stress-strain relationship
5. Define the stiffness matrix and equilibrium equations of element
6. Assembling of element equations to obtain the global or total equation
7. Applying boundary conditions

Pre Processing

8. Solution (Solver , Sub step / Time step , Nonlinearity etc)

Solution

9. In-Depth study & interpretation of Analysis Results (Sanity Checks)
10. Post processing of Results (Deflection , Stress , Strain etc..)
11. Report Preparation
12. Observation and Conclusion from the Analysis (MoS Calcs, Design ok)
13. Suggestion and Recommendation for Design Changes, if required.

Post Processing

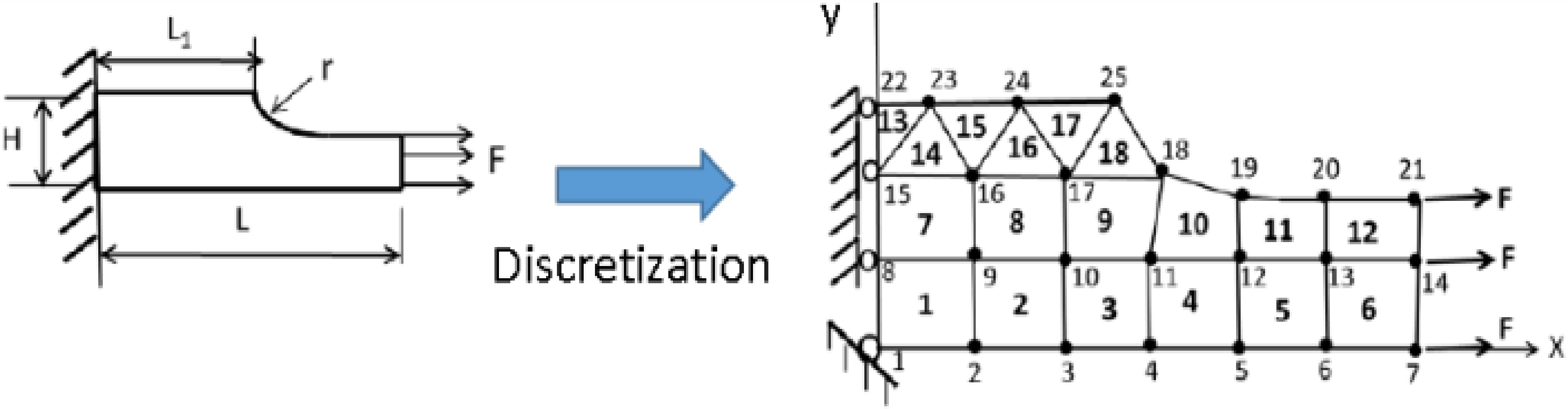
Step 1: discretization of the domain

- Discretization of the domain (Mesh formation) The first step involves dividing (discretization) the structure (domain or solution region) into subdivision elements with associated nodes.
- The art of subdividing a structure into a convenient number of smaller components is known as discretization
- The process of uniting the various elements together is called as assemblage
- The elements must be made small enough to give usable results and yet large enough to reduce computational effort. Small elements are desirable where the results are changing rapidly, such as where changes in geometry occur. Large elements are desirable where the results are relatively constant.
- The simple equations that model these finite elements are then assembled into a larger system of equations that models the entire problem.

Concept of Discretization (Meshing)

- Any continuum/domain can be divided into a number of pieces with very small dimensions.
- These small pieces of finite dimension are called ‘Finite Elements.
- Elements have definite shape
- A field quantity in each element is allowed to have a simple spatial variation which can be described by polynomial terms.
- Thus the original domain is considered as an assemblage of number of such small elements. These elements are connected through number of joints which are called ‘Nodes’.
- Loads are acting only at the nodes
- Nodes are placed where connection is made to another element

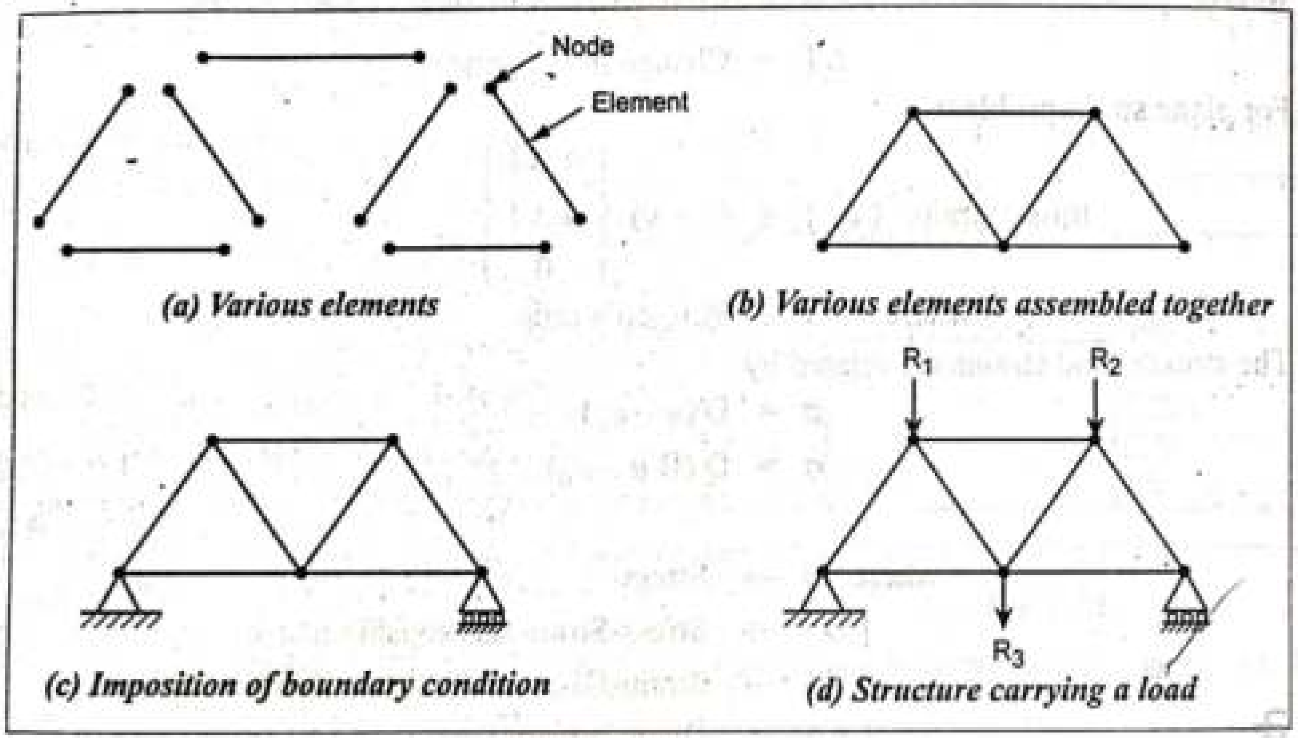
DISCRETIZATION

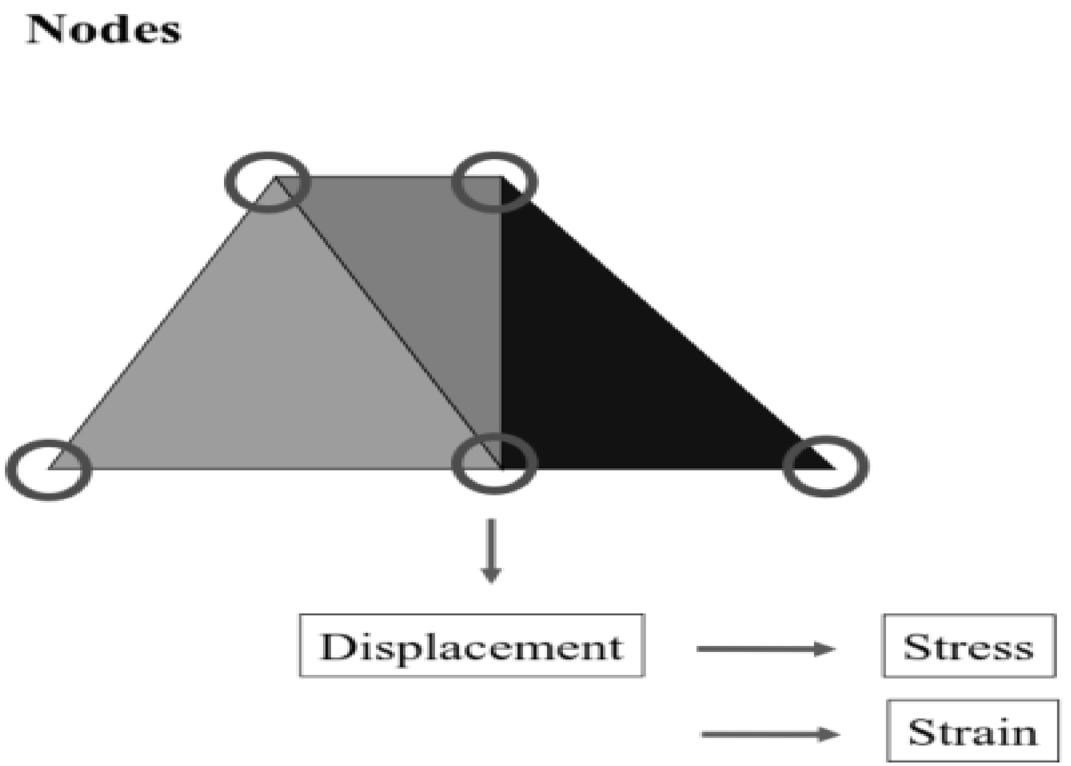


- Use bar elements for 1-D, Triangular and quadrilateral elements for 2-D plane and axisymmetric structures, and tetrahedron and hexahedron elements for 3-D media.
- Assign element and nodal numbers
- Enter nodal coordinates
- Enter element descriptions, e.g. Nodes 9,1017, 16 for Element 8;Node 16, 17, 24,24 for Element 16, etc.
- Enter nodal actions and constraints
- Make sure to place many more but smaller elements in regions with drastic change of geometry
- Many commercial FE codes offer "automatic mesh generation." Make use for this option.
- Many commercial packages offer automatic transformation of media profiles by CAD to FE analysis with automatic mesh generations. Again make use of this option.

DISCRETIZATION

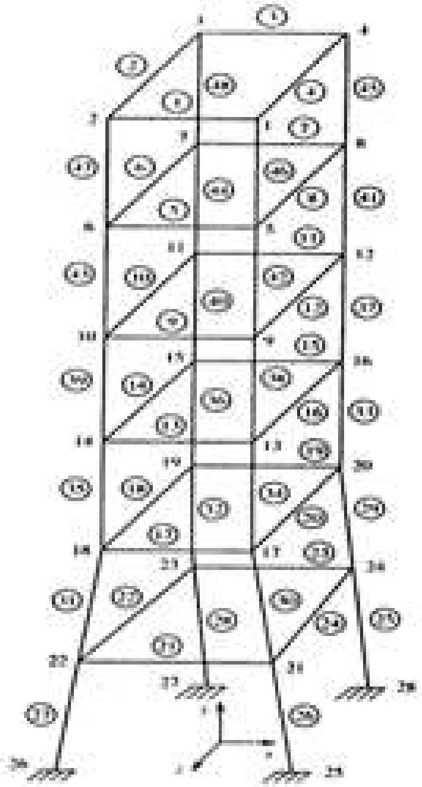
Fig. 1.11. Truss element



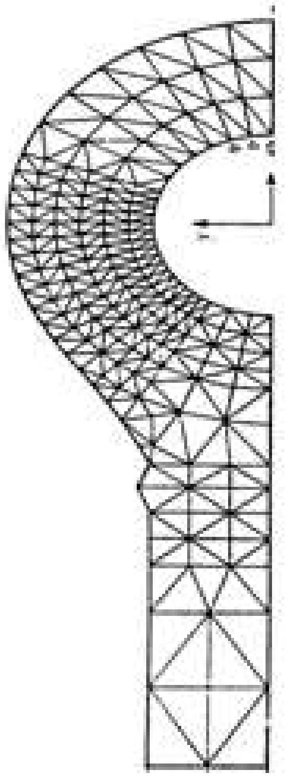


DISCRETIZATION

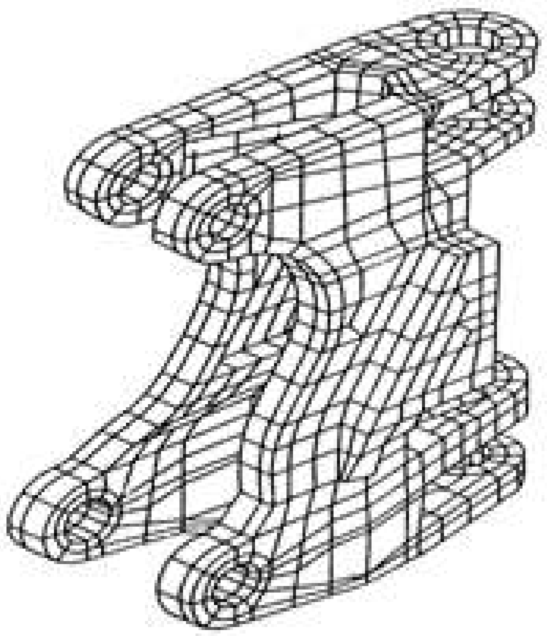
Discretization Examples



One-Dimensional Frame Elements



Two-Dimensional Triangular Elements



Three-Dimensional Brick Elements

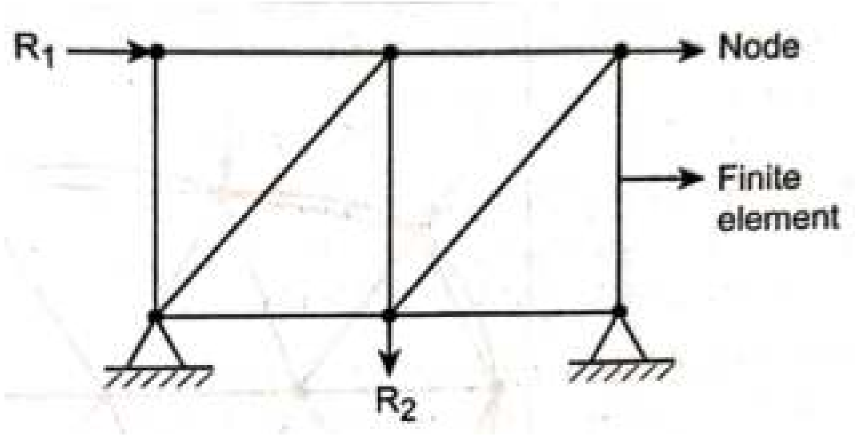
DISCRETIZATION

Discretization can be classified into

- Natural Discretization
- Artificial Discretization (Continuum)

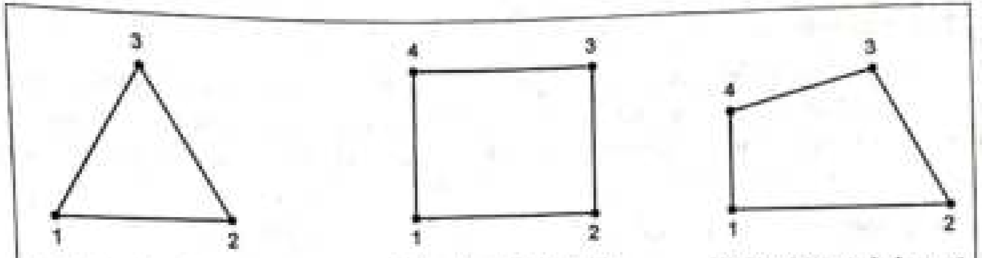
Natural Discretization

- In structural analysis , a truss is considered as a Natural Discretization
- The various member in the truss is constitute the elements.

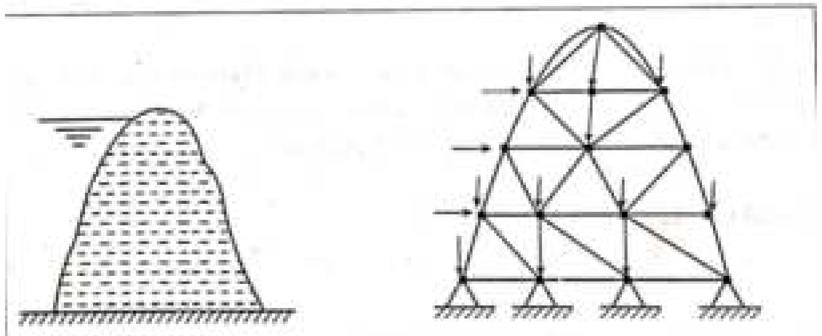


Artificial Discretization (Continuum)

- Continuum is considered to be a single mass of material as found in forging, concrete dam, plate…
- Triangular, Rectangular, Quadrilateral Elements are used for discretization



1. Triangular element 2. Rectangular element 3. Quadrilateral element



(a) Gravity dram (b) Discretization using triangular elements

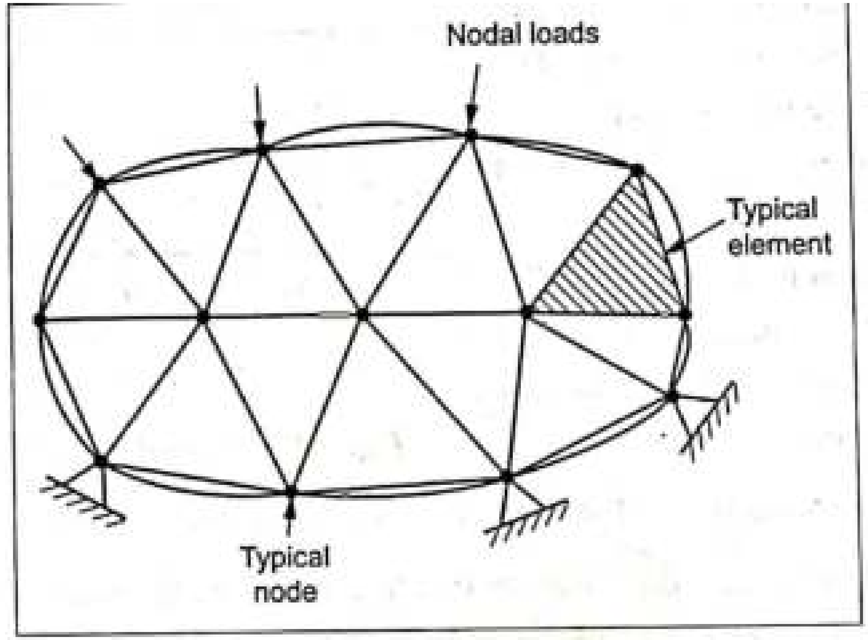


Fig. 1.15. Discretization using triangular elements

DISCRETIZATION

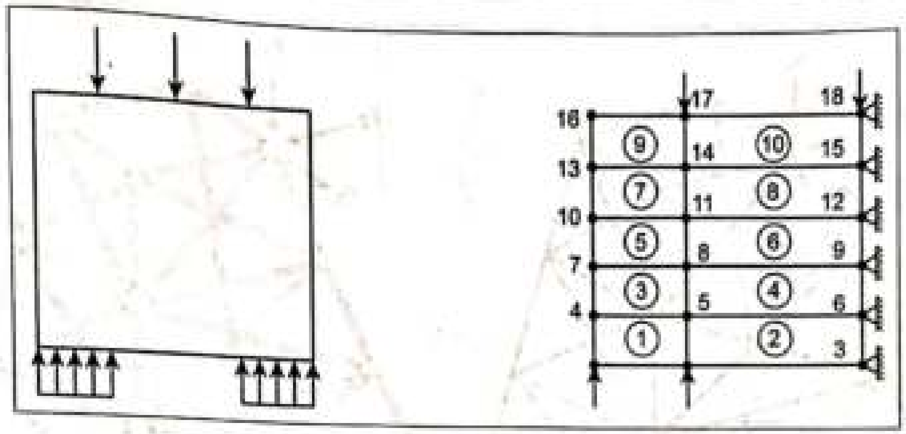


Fig. 1.17. Deep beam

Fig. 1.18. Deep beam discretization using rectangular elements

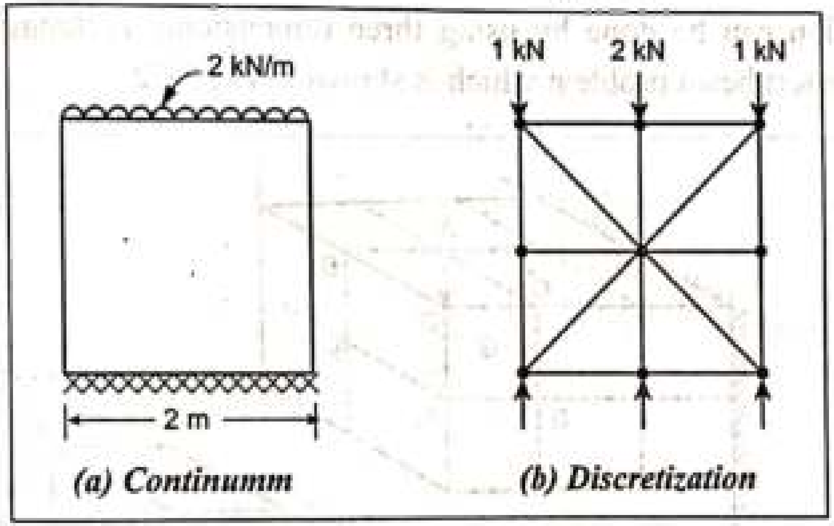


Fig. 1.19.

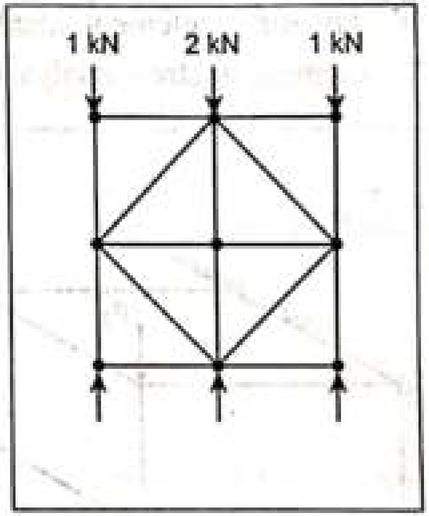
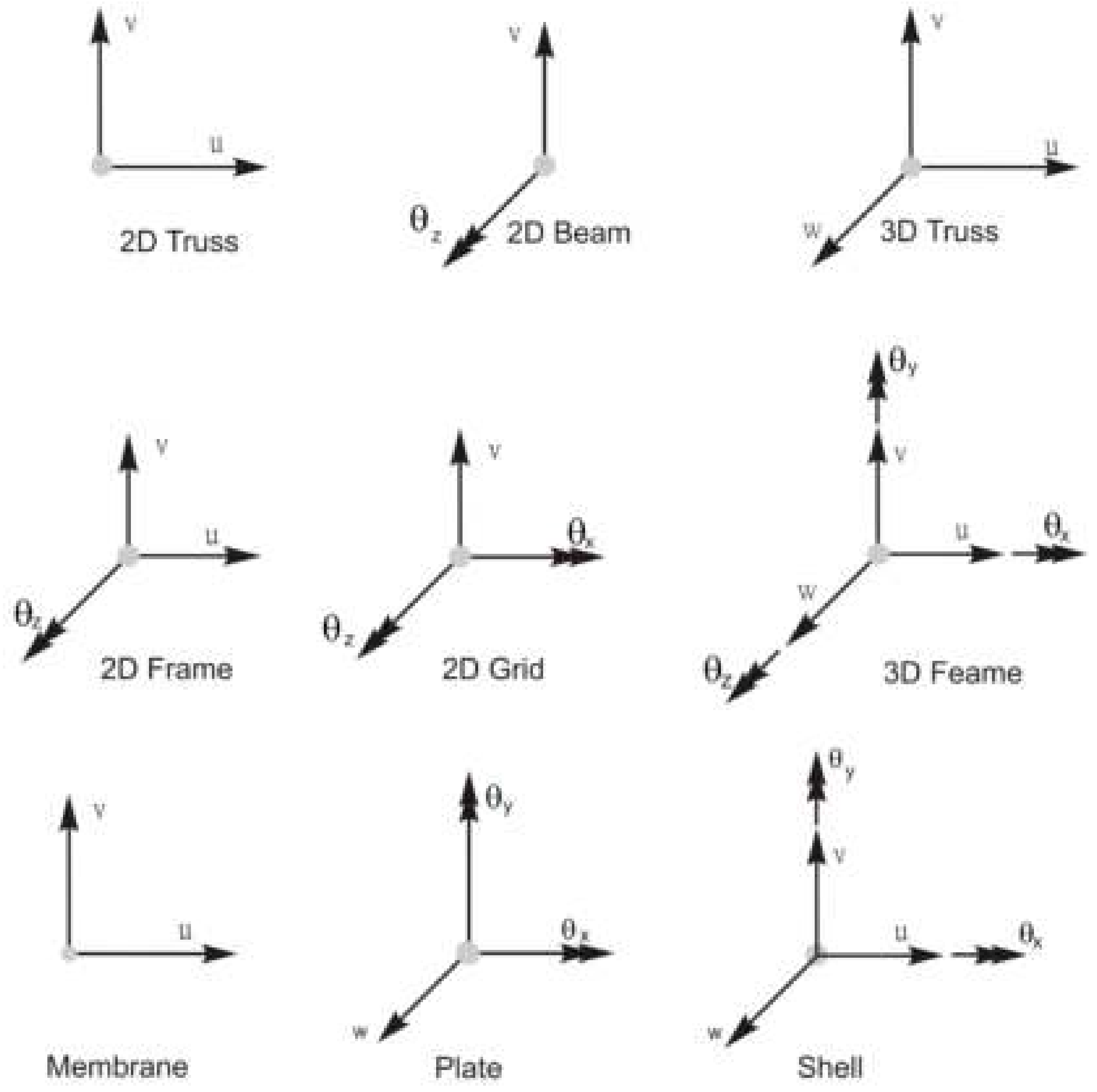


Fig. 1.20. Alternative way of discretization

DISCRETIZATION

Degrees of Freedom

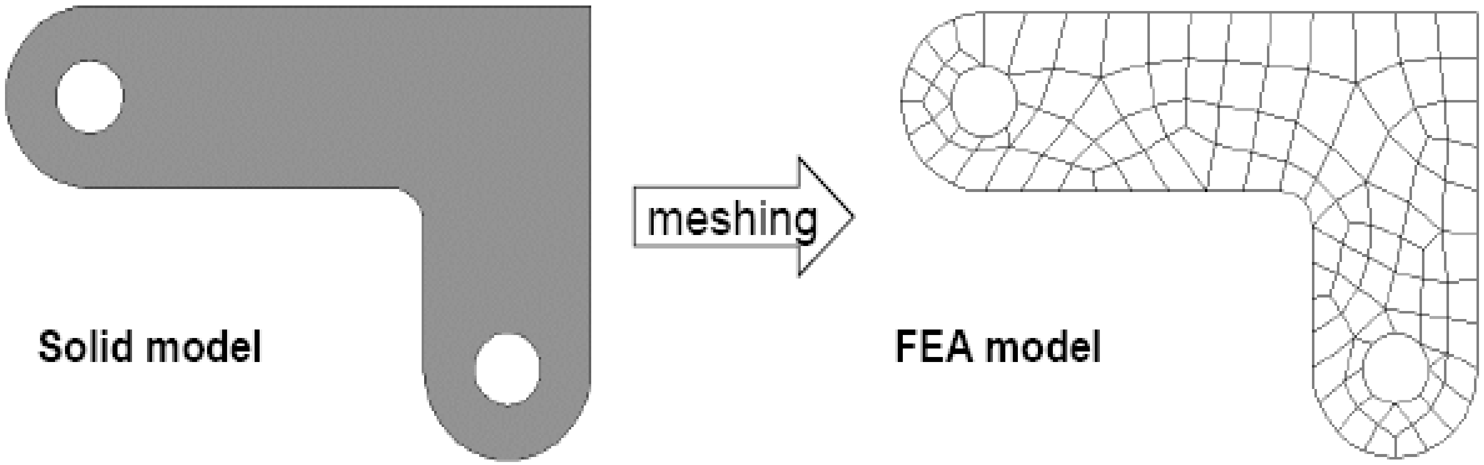
- Nodes are subject to deformation due to nodal forces.
- This includes displacements, rotations, strains. Collectively called as nodal displacements .
- This finite number of displacements is the number of degrees of freedom of the structure.



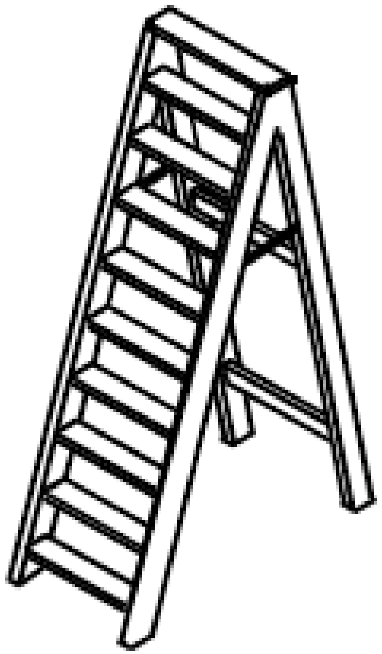
DISCRETIZATION

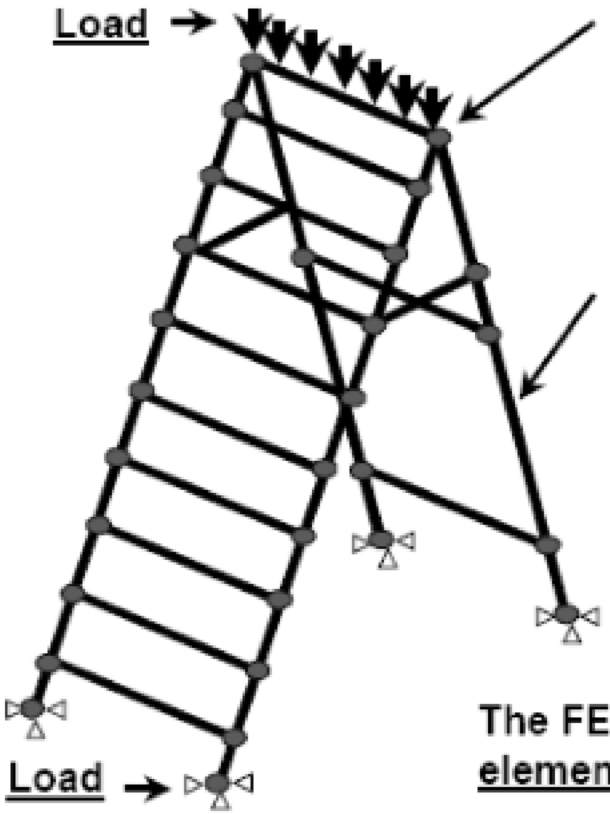
Meshing is the process used to "fill" the solid model with nodes and elements, i.e, to create the FEA model.

– Remember, you need nodes and elements for the finite element solution, not just the solid model. The solid model does NOT participate in the finite element solution.



DISCRETIZATION





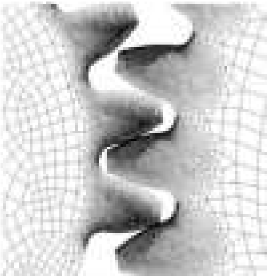
Node: Coordinate location in space where degrees of freedom and actions of the physical system exist.

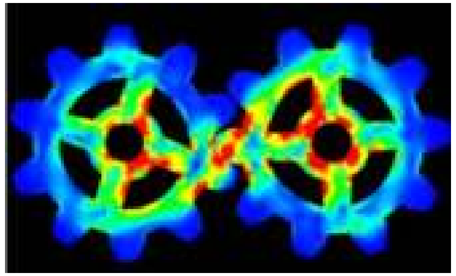
Element: Mathematical, matrix representation (called stiffness or coefficient matrix) of the interaction among the degrees of freedom of a set of nodes. Elements may be line, area, or solid, and two or three dimensional.

The FEA model consists of a number of simply shaped elements, connected to nodes, subjected to loads.

Physical System

FE Model

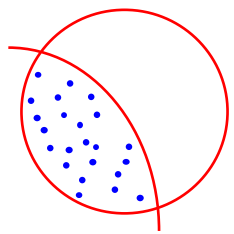


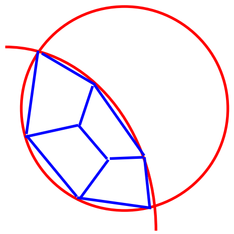


DISCRETIZATION

Concept of FEM is all about Discretization (Meshing) i.e. Dividing a big structure/component into small discrete Blocks (Nodes and Element concept)

But why do we do this Meshing ???





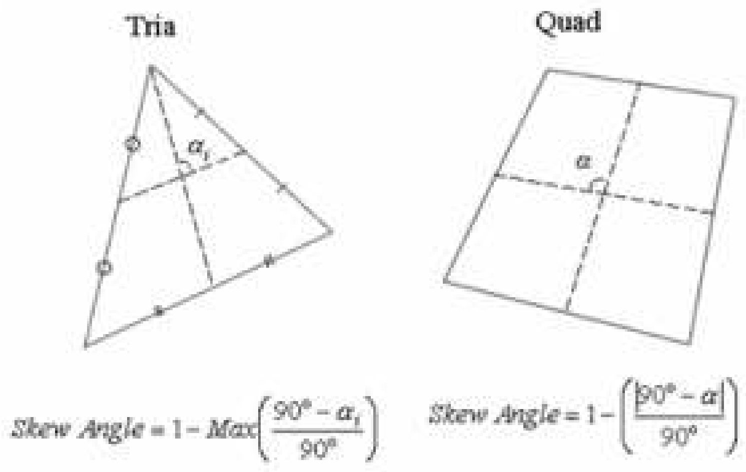
No. of Points = ∞
DoF per point = 6
Total No of Equations to be solved = $\infty * 6 = \infty$

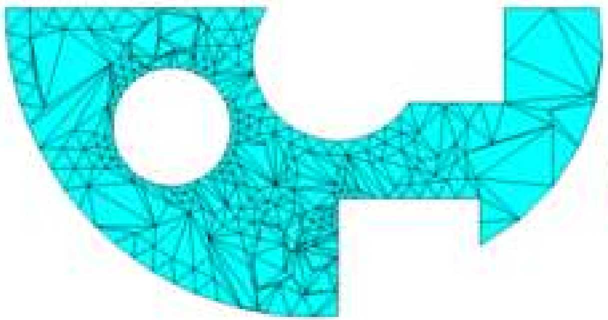
No. of Points = 8
DoF per point = 6
Total No of Equations to be solved = $8 * 6 = 48$

From Infinite to Finite…Hence the Term “Finite Element Method”

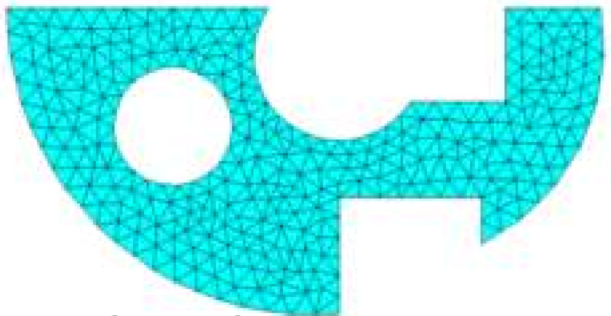
Parameters deciding the "Quality" of Mesh :

- Aspect ratio
- Skew / Warpage
- Element internal Angles
- Location of nodes

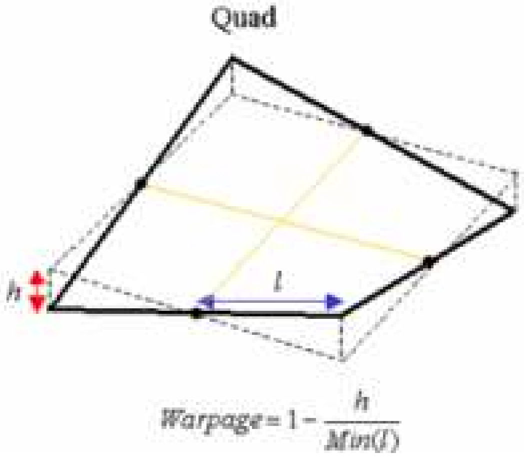




Bad Quality FEA

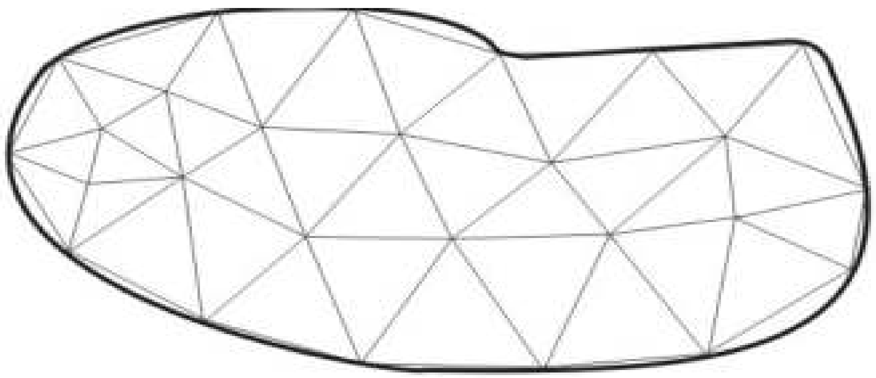


Good Quality FEA

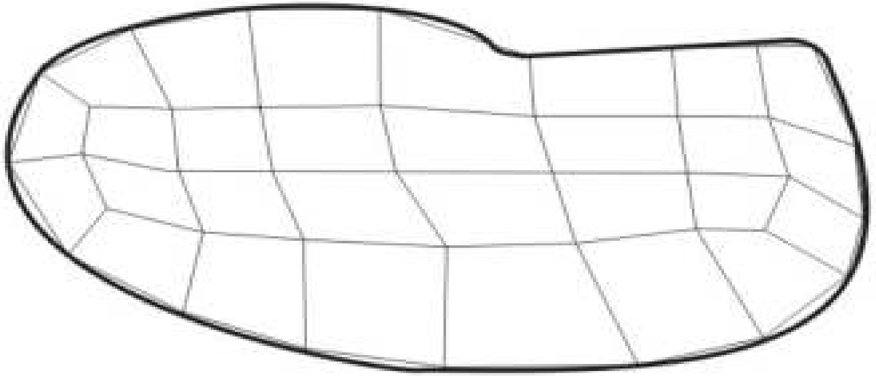


Better the Mesh Quality , Better the Accuracy

Types of 2D mesh – for continuum



(a) Triangular mesh



(b) Quadrilateral mesh

Numbering of nodes and elements

- The nodes and elements should be numbered after discretization process.
- It decided the size of the stiffness matrix
- While numbering following condition should be satisfied

$$\left\{\begin{matrix}\text{Maximum}\\ \text{node number}\end{matrix}\right\} - \left\{\begin{matrix}\text{Minimum}\\ \text{node number}\end{matrix}\right\} = \text{Minimum}$$

It is explained in the Fig.1.6(*a*) and (*b*).

Longer Side Numbering Process:

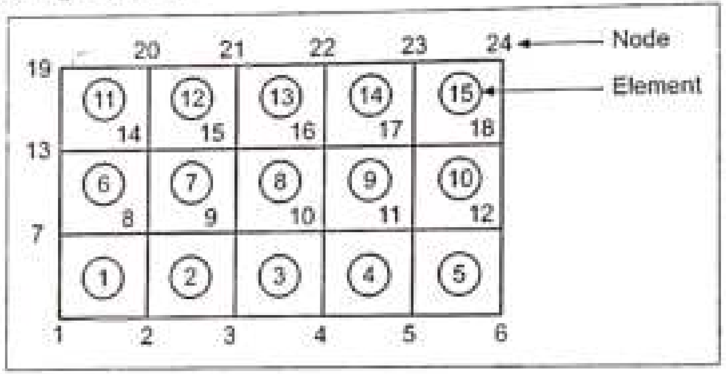
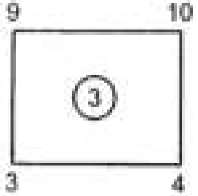


Fig. 1.6. (a)

[*Note:* Number with circle denotes element.
Number without circle denotes node]

Considering element (3),



Maximum node number = 10

Minimum node number = 3

Difference = 7 ... (1.1)

DISCRETIZATION

Shorter Side Numbering Process:

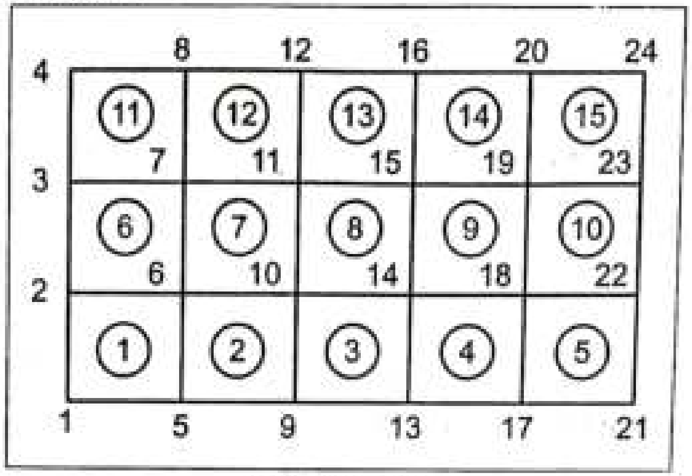
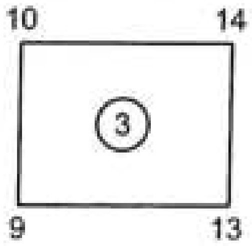


Fig. 1.6. (b)

Considering the same element (3).



Maximum node number = 14

Minimum node number = 9

Difference = 5 ... (1.2)

From equation (1.1) and (1.2), we came to know, shorter side numbering process is followed in the finite element analysis and it reduces the memory requirements.

Discretization process

Following factors are considered during discretization

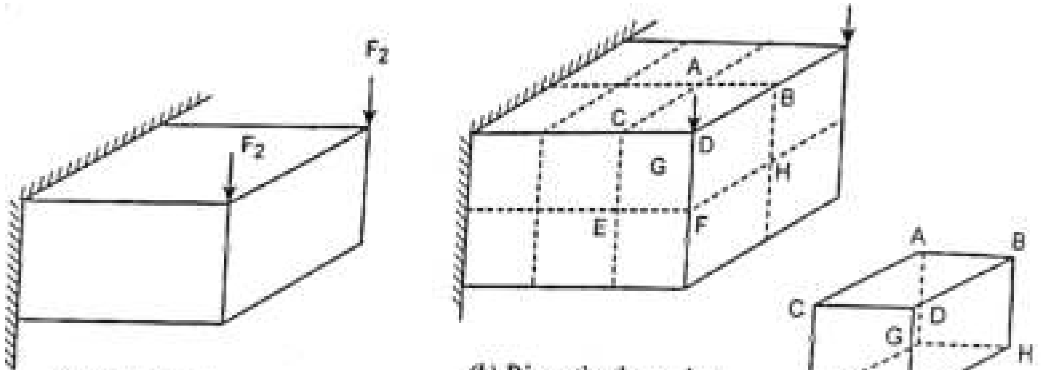
- Types of elements
- Size of the element
- Location of nodes
- Number of elements

DISCRETIZATION

1. Types of elements

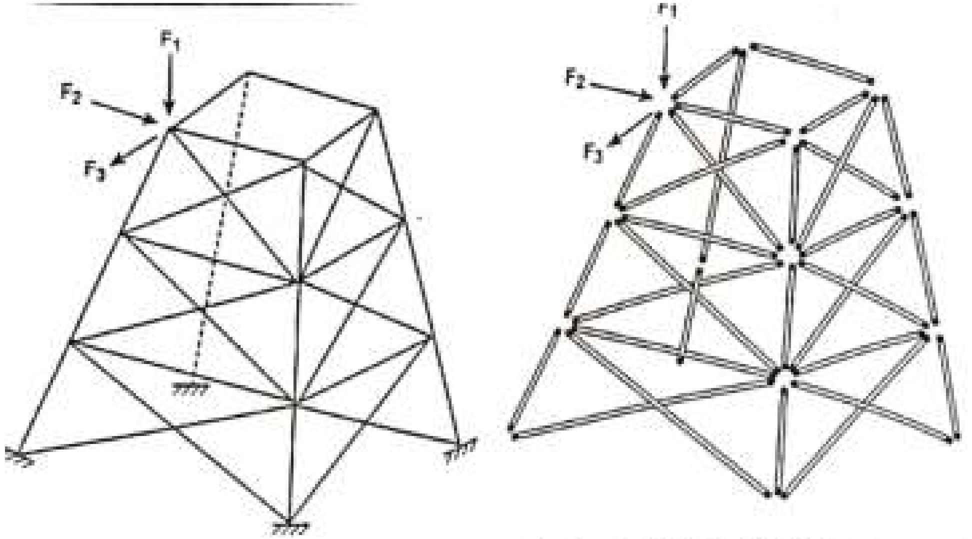
Depends on following factors

- Number of degrees of freedom needed
- Expected accuracy
- Necessary equations required



(a) Short beam

(b) Discretization using three-dimensional elements



(a) Original structure

(b) Discretization using bar elements

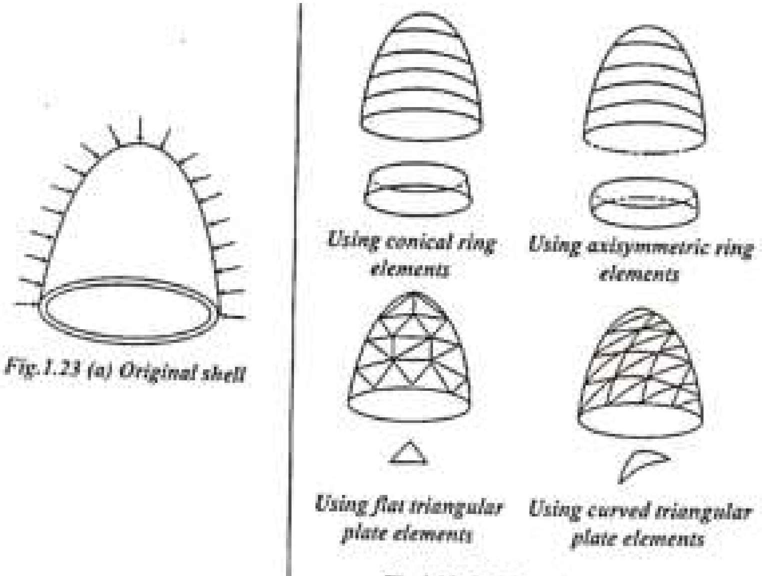


Fig. 1.23 (a) Original shell

Fig. 1.23 (b) Discretization using different types of elements

DISCRETIZATION

2. Size of elements

- For proper convergence of the solution
- Small size elements give accurate result but computational time is more
- Aspect ratio of the element influence the accuracy of the result

3. Location of Nodes

- If the structure has no geometric, load, boundary conditions and material properties, the structure can be divided into equal subdivisions.
- If the structure has any discontinuity in geometric, load, boundary conditions and material properties, the structure Nodes should be introduced at these discontinuity.

DISCRETIZATION

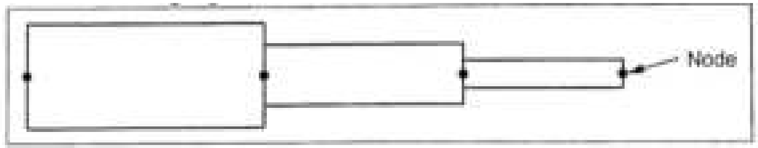


Fig. 1.24. Geometric discontinuities

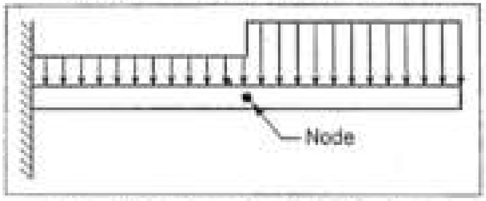


Fig. 1.25. Discontinuity in loading

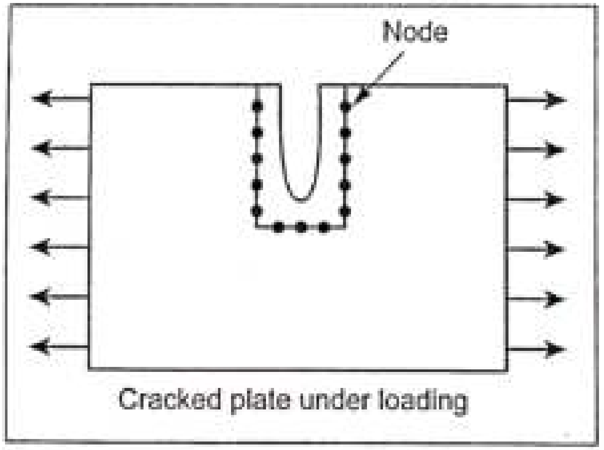


Fig. 1.26. Discontinuity of boundary conditions

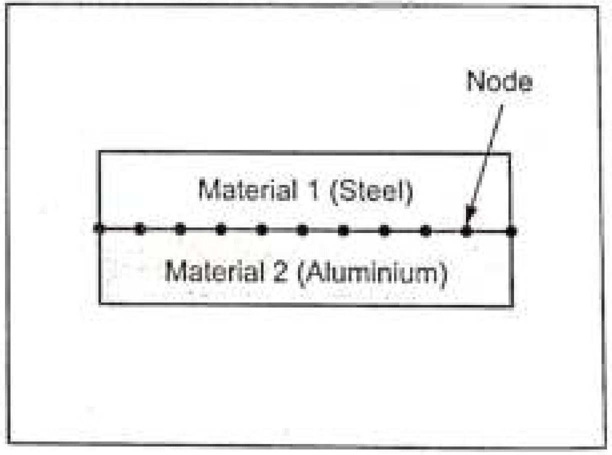


Fig. 1.27. Material discontinuity

3. Number of elements

The number of elements to be selected for discretization depends,

- Accuracy desired
- Size of the element
- Number of degrees of freedom involved

TYPES OF ELEMENTS

Types of Finite Elements

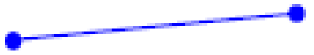
Linear - 1st Order Element

linear

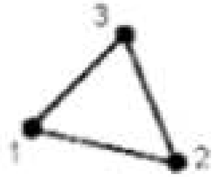
quadratic

Quadratic 2nd Order Element

1-D (Line) Element

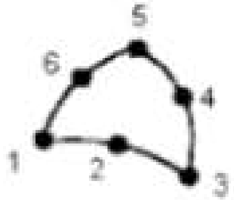


(Spring, truss, beam, pipe, etc.)



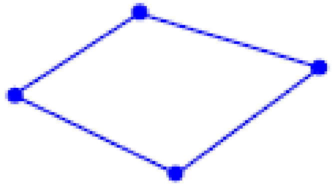
SHEL3

Triangular Element

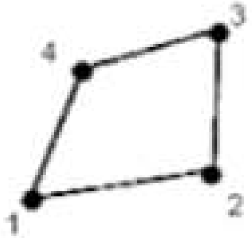


SHEL6

2-D (Plane) Element

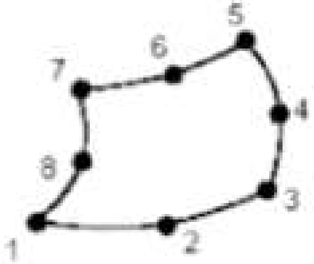


(Membrane, plate, shell, etc.)



SHEL4

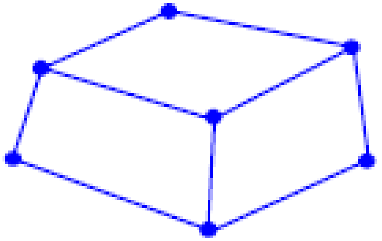
Quadrilateral Element



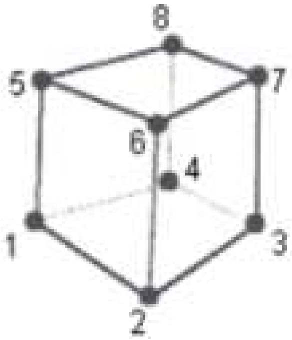
SHEL8

2D

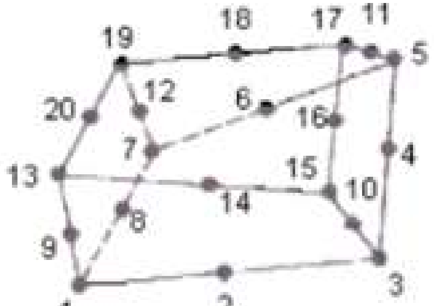
3-D (Solid) Element



(3-D fields - temperature, displacement, stress, flow velocity)

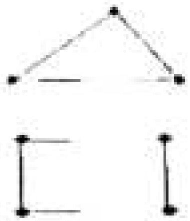
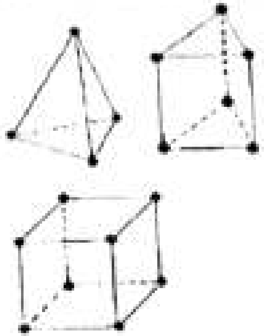
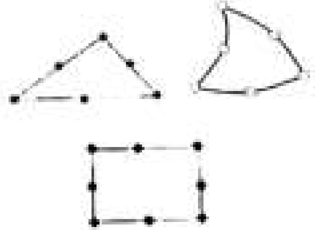
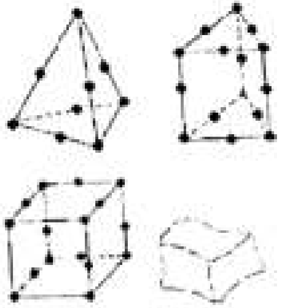
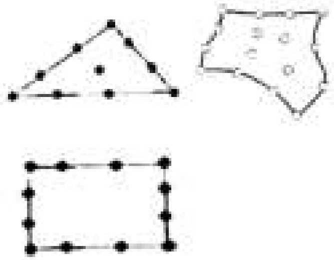
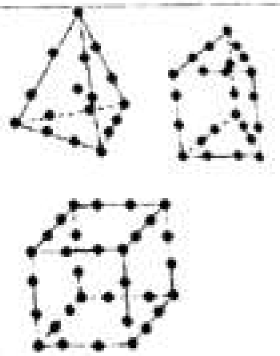


Hexahedral Element

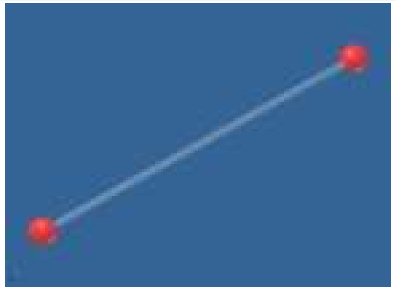


3D

FEM STEPS

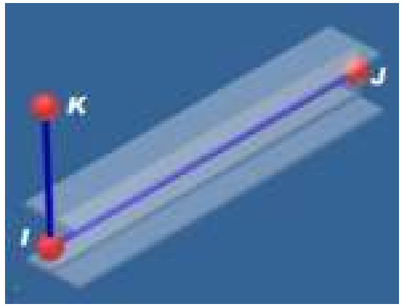
	1D Elements	2D Elements	3D Elements
Linear Elements (First Order)			
Quadratic Elements (second Order)			
Cubic Elements (Third Order)			

Type of Elements



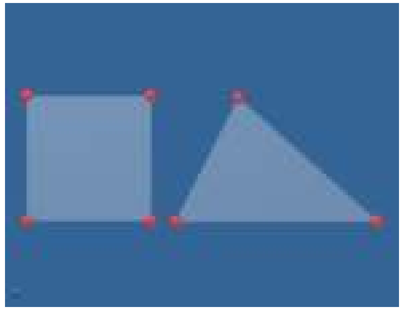
Truss Element

- Truss elements are long and slender, have 2 nodes, and can be oriented anywhere in 3D space. Truss elements transmit force axially only and are 3 DOF elements which allow translation only and not rotation.
- Trusses are normally used to model towers, bridges, and buildings. A constant cross section area is assumed and they are used for linear elastic structural analysis.



Beam Element

- Beam elements are long and slender, have three nodes, and can be oriented anywhere in 3D space
- Beam elements are 6 DOF elements allowing both translation and rotation at each end node.
- The i,j nodes define element geometry, the K node defines the cross sectional orientation.
- A constant cross section area is assumed.

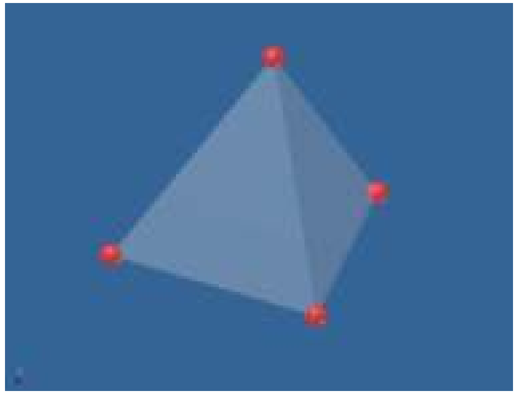


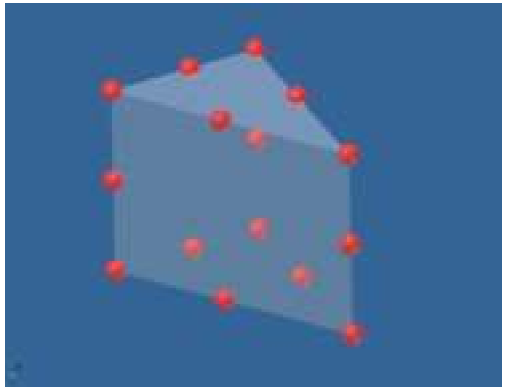
2D Element (2D Planar)

- 2D Elements are 3 or 4 node elements with only 2 DOF, Y and Z translation, and are normally created in the YZ plane. They are used for Plane Stress or Plane Strain analyses.
- Plane Stress implies no stress normal to the cross section defined - strain is allowed - suitable to model the 2D cross section of a body of revolution.
- Plane Strain implies no strain normal to the cross section defined - stress is allowed - suitable to model the 2D cross section of a long dam.

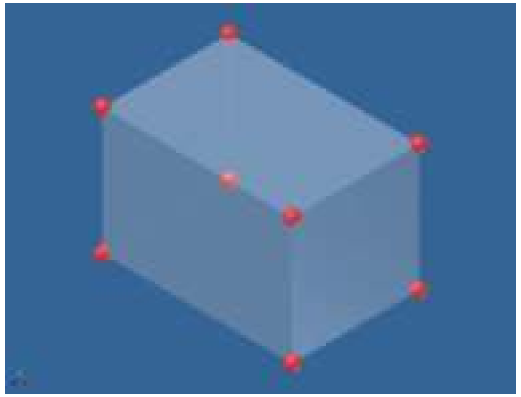
TYPES OF ELEMENTS

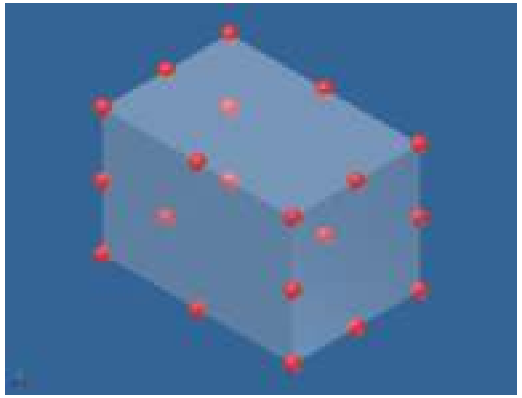
Different Type of 3D Elements





Tetrahedron





Hexahedral

Step 2: identification of variables

The elements are assumed to be connected at their intersecting points referred to as **nodal points**. At each nodes, unknown are to be prescribed.
Identify primary unknown quantity:

- Element displacements for stress analysis
- Element temperature for heat conduction analysis
- Element velocities for fluid dynamic analysis

Step 3: Select a displacement function or interpolation function

- This step involves choosing a displacement function **within each element**. The function is defined, **within the element** using the nodal values of the element. This function represents the **variation of the displacement within the element**. Linear, quadratic and cubic **polynomials** are frequently used functions depending upon the type of element.

FEM STEPS

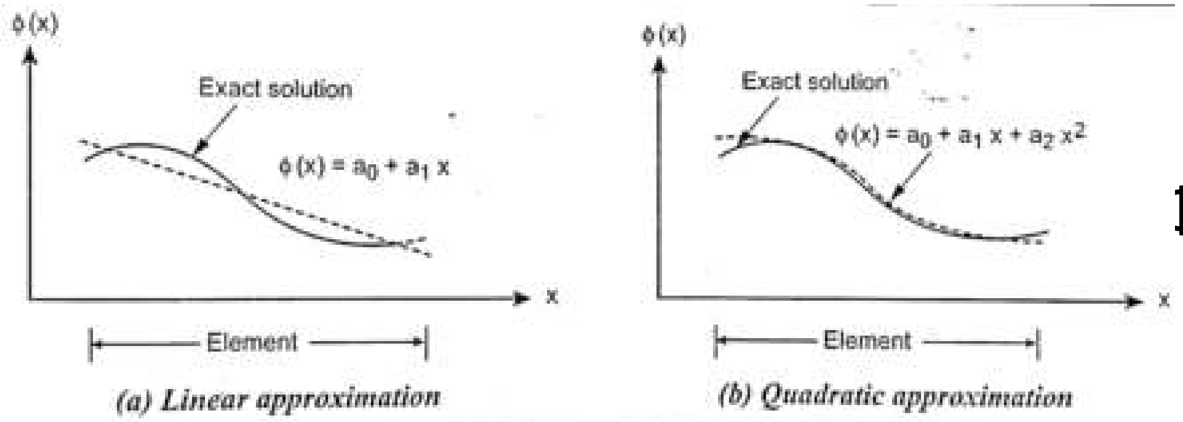


Fig. 1.7. Polynomial approximation in one dimension

Ɖ$_x$ is the field variable.

Case (i): Linear Polynomial:

$$\text{One dimensional problem } \phi(x) = a_0 + a_1 x$$

$$\text{Two dimensional problem } \phi(x, y) = a_0 + a_1 x + a_2 y$$

$$\text{Three dimensional problem } \phi(x, y, z) = a_0 + a_1 x + a_2 y + a_3 z$$

Case (ii): Quadratic Polynomial:

$$\text{One dimensional problem } \phi(x) = a_0 + a_1 x + a_2 x^2$$

$$\text{Two dimensional problem } \phi(x, y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 y^2 + a_5 xy$$

$$\text{Three dimensional problem } \phi(x, y, z) = a_0 + a_1 x + a_2 y + a_3 z + a_4 x^2 + a_5 y^2 + a_6 z^2 + a_7 xy + a_8 yz + a_9 xz$$

Step 3: Interpolation functions and derivation of Interpolation functions

- a very important step

Because primary unknown quantities in FEA are for those in the elements, but elements are interconnected at nodes, so it is important to establish relationship for the quantities in the elements with the associated nodes. This is what interpolation functions are defined. Mathematical expressions of interpolation functions:

$$\{\phi(\mathbf{r})\} = \{N(\mathbf{r})\}\{\phi\}$$

where {Φ(r)} = Element quantity, {Φ} = nodal quantity, N(r) = interpolation function with r = coordinates

Interpolation functions may be expressed to relate the corresponding nodes in the following way:

- $Element\,Quantity\ \phi(x,y,z) = Interpolation\ Function\ \{N_1(x,y,z) \quad N_2(x,y,z) \quad N_3(x,y,z) \quad N_4(x,y,z)\}\times$ $Nodal\,Qunatity,\ \{\phi\}^T = \{\phi_1 \quad \phi_2 \quad \phi_3 \quad \phi_4\}$ for tetrahedron elements with 4 nodes
- $Element\ Quantity\ \phi(x,y) = Interpolation\ Function\ \{N_1(x,y) \quad N_2(x,y) \quad N_3(x,y)\}\times Nodal\,Quantity,\ \{\phi\}^T = \{\phi_1 \quad \phi_2 \quad \phi_3\}$ for plate elements with 3 nodes
- $Element\,Quantity\ \phi(x) = Interpolation\ function\ \{N_1(x) \quad N_2(x)\}\times Nodal\,Quantity,\ \{\phi\}^T = \{\phi_1 \quad \phi_2\}$ for bar elements with 2 nodes

Step 4: Define the material behavior by using strain-displacement or stress-strain relationship

- Strain-displacement or stress-strain relationship are necessary for deriving the equations for each finite element.
- In case of one dimensional deformation ,say in X direction

$$\text{Strain , } \varepsilon_X = \frac{du}{dx}$$

Where 'u' is the displacement or deformation.

$$\text{Stress , } \sigma_x = E\varepsilon_X$$

E is modulus of elasticity

Step 5: Define the stiffness matrix and equilibrium equations of element

- In this step, the stiffness matrix and equilibrium equations for one, two or three dimensional elements are obtained based on the following methods

This equation can be derived by any one of the following methods.

- Direct Equilibrium Method: This method is much easier to apply for line or one dimensional elements.
- Variational Method: This method is most easily adaptable to the determination of element equations for complicated elements (i.e., element having large number of degrees of freedom) like axisymmetric stress element, plate bending element and two or three dimensional solid stress element.
- Weighted Residual Method: This method is (Galerkin's method) useful for developing the element equations in thermal analysis problems. They are especially useful when a functional such as potential energy is not readily available.

- **Stiffness matrix** represents the system of linear equations that must be solved in order to ascertain an approximate solution to the differential equation.
- A matrix which relates the force vector to the displacement vector'. In mathematical term.
- If you think of a spring, then the deflection of a spring 'd' can be related to the applied force 'P' by
  P=K*d
- where K is the stiffness. From this, you can say stiffness is the amount of force required to cause unit displacement. The same concept is valid for stiffness matrix also. If you think of a structures which has multi degrees of freedom, then you will have many stiffness term associated with these degrees of freedom. In FEM, theses are written in matrix form. This is called stiffness matrix.

- For a structural problem the equation given below is the equilibrium equation of an element

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ \vdots \\ F_n \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & \cdots & k_{1n} \\ k_{21} & k_{22} & k_{23} & \cdots & k_{2n} \\ k_{31} & k_{32} & k_{33} & \cdots & k_{3n} \\ \vdots & & & & \vdots \\ k_{n1} & \cdots & \cdots & \cdots & k_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_n \end{Bmatrix}$$

In compact matrix form as,

$$\{F^e\} = [k^e]\{u^e\}$$

where, e is a Element, $\{F\}$ is the vector of element nodal forces, $[k]$ is the element stiffness matrix and $\{u\}$ is the element displacement vector.

Step 6: Assembling of element equations to obtain the global or total equation

- After the element stiffness matrices are formed for all elements, they are assembled to form overall stiffness matrix or global equilibrium equations using the method of superposition called direct stiffness method.

$$\{F\} = [K]\{u\}$$

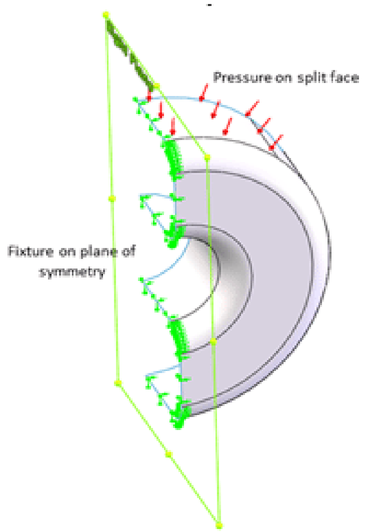
where, $\{F\} \rightarrow$ Global force vector.

$[K] \rightarrow$ Global stiffness matrix.

$\{u\} \rightarrow$ Global displacement vector.

Step 7 : Applying boundary conditions

- The boundary conditions are to be imposed in the global equilibrium equations, which may result in the reduction of size of the global stiffness matrix and equation.
- Global stiffness matrix [K] is a singular matrix because its determinant is equal to zero. In order to remove this singularity problem, certain boundary conditions are applied so that the structure remains in place instead of moving as a rigid body.



Solution Phase

Step 8: Solution for the unknown displacements

- A set of simultaneous algebraic equations formed in step 6 can be written in expanded matrix form as follows:

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ \vdots \\ \vdots \\ F_n \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & \cdots & k_{1n} \\ k_{21} & k_{22} & k_{23} & \cdots & k_{2n} \\ k_{31} & k_{32} & k_{33} & \cdots & k_{3n} \\ k_{41} & k_{42} & k_{43} & \cdots & k_{4n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ k_{n1} & k_{n2} & k_{n3} & \cdots & k_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \vdots \\ \vdots \\ u_n \end{Bmatrix}$$

- These equations can be solved and unknown displacements {u} are calculated by using Gaussian elimination method or Gauss-Seidel method

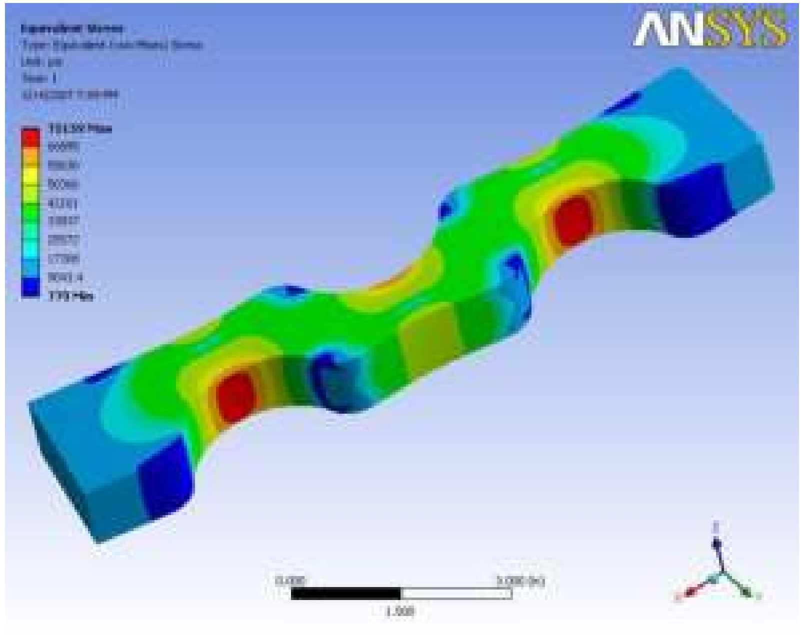
Post Processing Phase

Step 9 : Compute element stress and strains

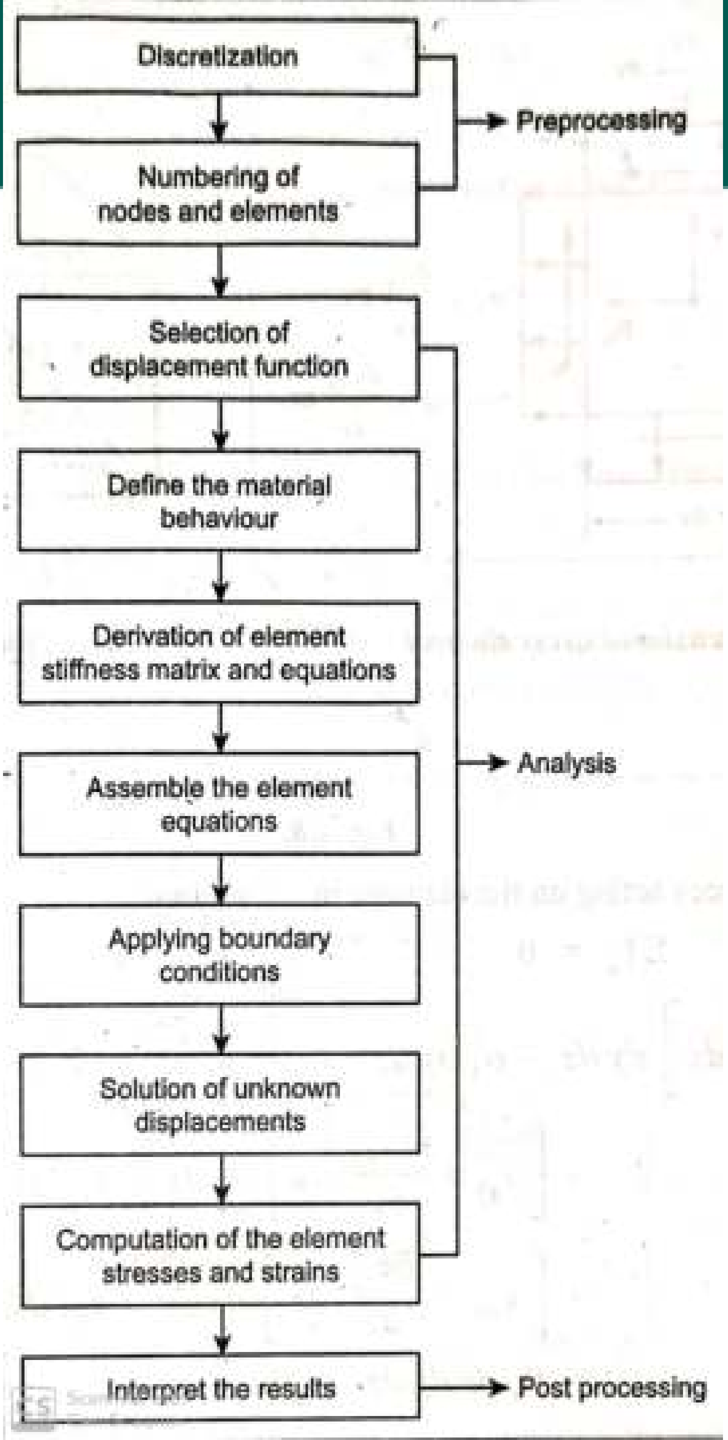
- Computation of the element strains and stresses from the nodal displacements {u},
- In structural stress analysis problem, stress and strain are important factors. From the solution of displacement vector { u }, stress and strain value can be calculated.

Step 10 : Interpret the results

- The results obtained are analyzed to determine the locations in the structure where large deformations are large stress occur and imported design decision are made .
- Results of FEA usually are presented in the following forms: (1) tabulations, (2) graphics: static and animation Interpretation of results



Flow chart

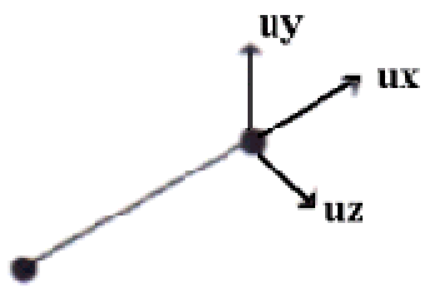


Discretization
Numbering of nodes and elements
Preprocessing
Selection of displacement function
Define the material behaviour
Derivation of element stiffness matrix and equations
Assemble the element equations
Analysis
Applying boundary conditions
Solution of unknown displacements
Computation of the element stresses and strains
Interpret the results
Post processing

STRUCTURAL PROBLEMS (One Dimensional)

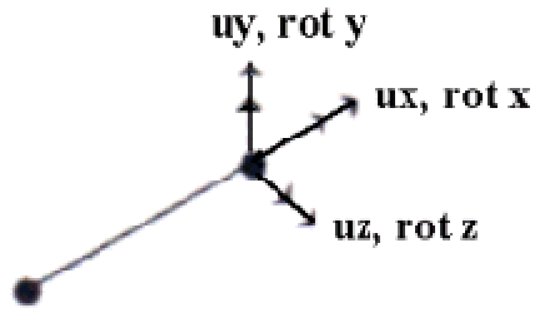
- Bar and Beams are considered as one dimensional problem.
- These elements are often used to model trusses and frame structures.
- A bar has longitudinal dimension much larger than the other dimensions.
- A bar is a member which resist only axial loads, whereas a beam can resist axial, lateral and twisting loads.
- A truss is an assemblage of bars with pin joints and a frame is an assemblage of beam elements.

Bar element



only displacements ux, uy, uz

Beam element



displacements ux, uy, uz and also rotational degrees rot x, rot y, rot

Stress, strain ,Displacement and Loading

In one dimensional problems, stress (σ), strain (e), displacement (u) and loading depends only on the variable x. So, the vectors u, σ and e can be written as,

$$u = u(x)$$
$$\sigma = \sigma(x)$$
$$e = e(x)$$

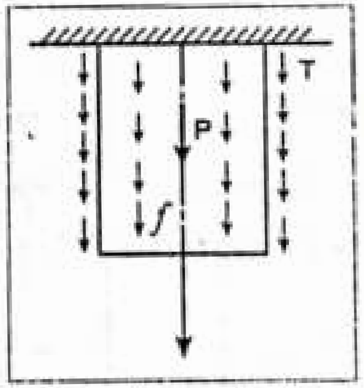


Fig. 2.1. A bar is subjected to loading

The stress-strain relationship is given by,

$$\sigma = E\, e$$

where, $\sigma \rightarrow$ Stress, N/mm^2.

$e \rightarrow$ Strain.

$E \rightarrow$ Young's modulus, N/mm^2.

The strain-displacement relationship is given by,

$$e = \frac{du}{dx}$$

The differential volume can be written as,

$$dV = A\, dx$$

There are three types of loading acts on the body. They are:

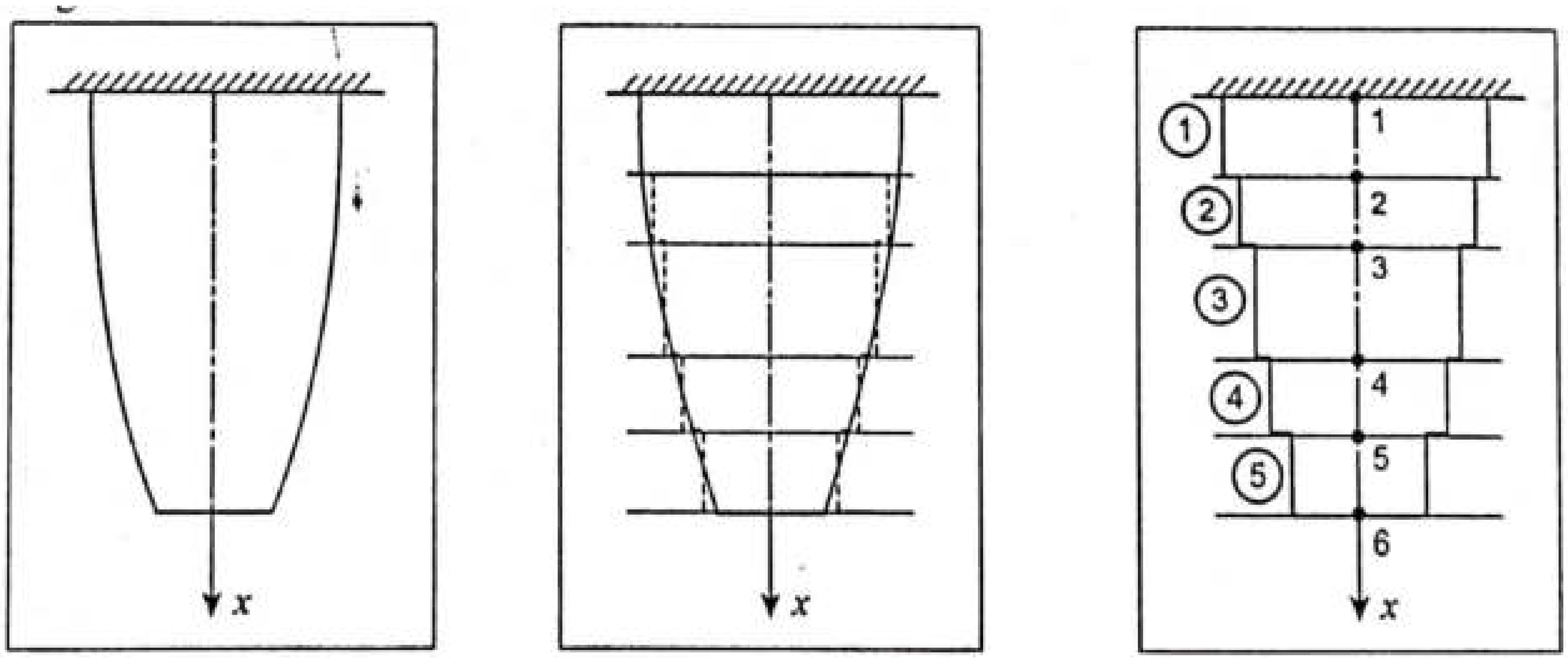
(*i*) Body force (f).

(*ii*) Traction force (T).

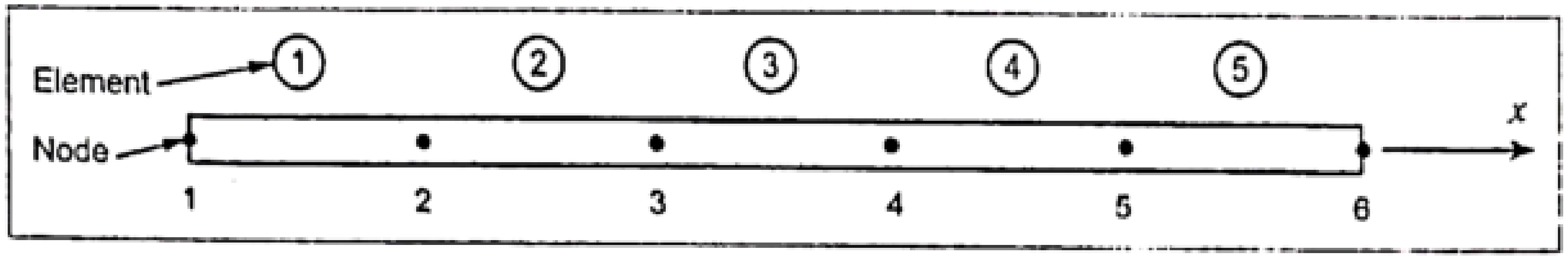
(*iii*) Point load (P).

Finite Element Modeling

- Finite element modeling consist of,
  - Discretization of the structure
  - Numbering of nodes



- In one dimensional problem, each node is allowed to move only in x direction.
- Each node has one degrees of freedom. (Degrees of freedom is nothing but a nodal displacement).



Co-ordinates

- Global Co-ordinates
  - The points in the entire structure are defined using Global co-ordinates

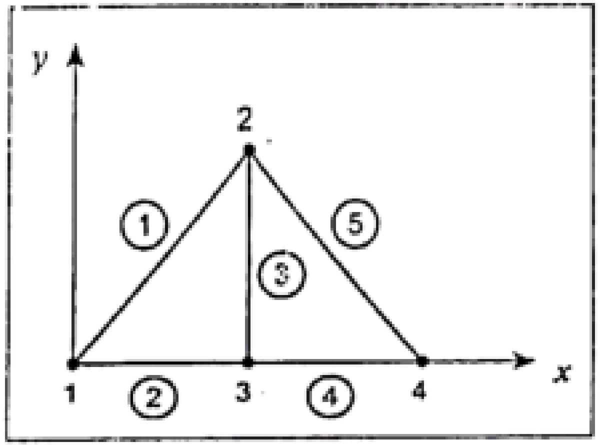
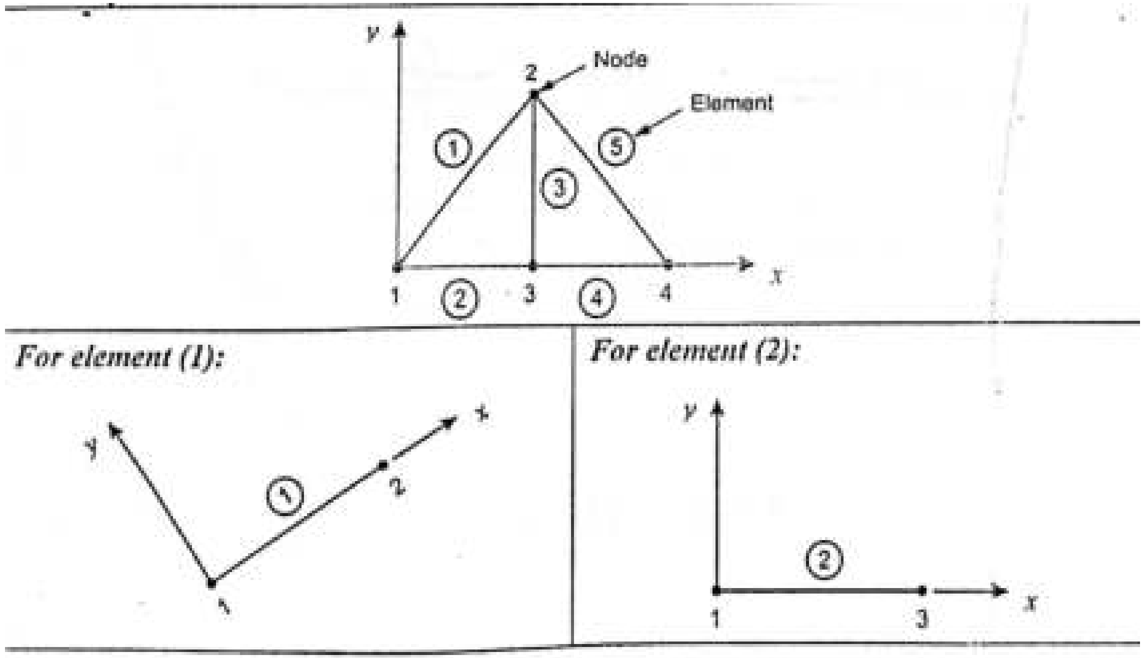
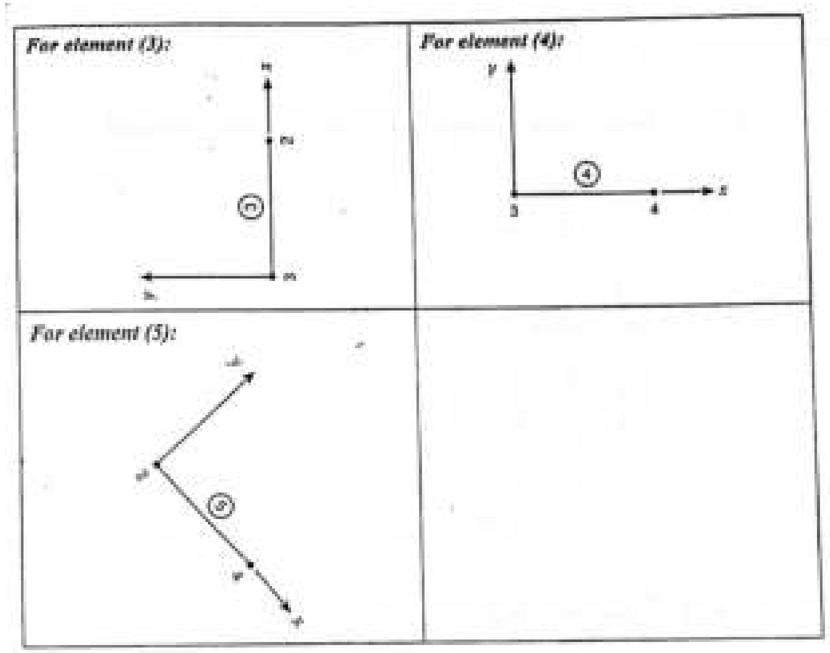


Fig. 2.8. Two dimensional triangular element

- Local Co-ordinates
  - In finite element method, separate co-ordinate is used for each element.
  - It is very useful for deriving element properties. But the final equations are to be formed only by global co-ordinate systems.





- Natural Co-ordinates
  - A natural co-ordinate system is used to define any point inside the element by a set of dimensionless numbers whose magnitude never exceeds unity.

(1) Natural Co-ordinates in One Dimension

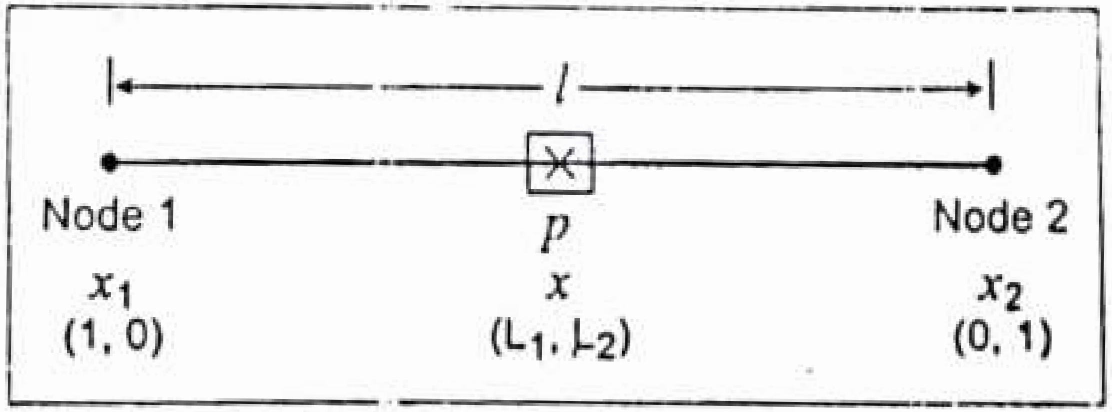


Fig. 2.10. Natural co-ordinates for a line element

Consider a two noded line element as shown in Fig.2.10. Any point p inside the line element is identified by two natural co-ordinates L_1 and L_2 and the cartesian co-ordinate x. Node 1 and node 2 have the cartesian co-ordinates x_1 and x_2 respectively.

We know that,

Total weightage of natural co-ordinates at any point is unity.

i.e., $$L_1 + L_2 = 1 \quad \text{... (2.1)}$$

Any point x within the element can be expressed as a linear combination of the nodal co-ordinates of nodes 1 and 2 as,

$$L_1 x_1 + L_2 x_2 = x \quad \text{... (2.2)}$$

Arrange equation (2.1) and (2.2) in matrix form,

$$\begin{bmatrix} 1 & 1 \\ x_1 & x_2 \end{bmatrix} \begin{Bmatrix} L_1 \\ L_2 \end{Bmatrix} = \begin{Bmatrix} 1 \\ x \end{Bmatrix}$$

$$\Rightarrow \quad \begin{Bmatrix} L_1 \\ L_2 \end{Bmatrix} = \begin{bmatrix} 1 & 1 \\ x_1 & x_2 \end{bmatrix}^{-1} \begin{Bmatrix} 1 \\ x \end{Bmatrix}$$

$$= \frac{1}{(x_2 - x_1)} \begin{bmatrix} x_2 & -1 \\ -x_1 & 1 \end{bmatrix} \begin{Bmatrix} 1 \\ x \end{Bmatrix}$$

$$\left[\textit{Note:} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}^{-1} = \frac{1}{(a_{11} \cdot a_{22}) - (a_{12} \cdot a_{21})} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \right]$$

$$= \frac{1}{x_2 - x_1} \begin{Bmatrix} x_2 - x \\ -x_1 + x \end{Bmatrix}$$

$$= \frac{1}{x_2 - x_1} \begin{Bmatrix} x_2 - x \\ x - x_1 \end{Bmatrix}$$

$$= \frac{1}{l} \begin{Bmatrix} x_2 - x \\ x - x_1 \end{Bmatrix} \quad [\because x_2 - x_1 \text{ is the length of the element, } l]$$

$$\begin{Bmatrix} L_1 \\ L_2 \end{Bmatrix} = \begin{Bmatrix} \dfrac{x_2 - x}{l} \\ \dfrac{x - x_1}{l} \end{Bmatrix}$$

The variation of L_1 and L_2 is shown in Fig.2.12 and Fig.2.13. L_1 is one at node 1 and it is zero at node 2 whereas L_2 is one at node 2 and it is zero at node 1.

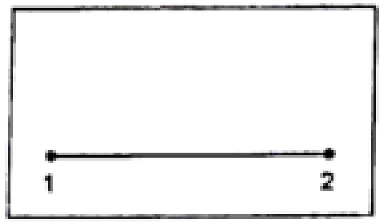


Fig. 2.11.

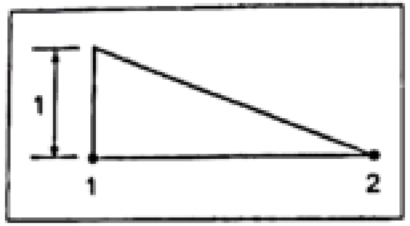


Fig. 2.12. Variation of L_1

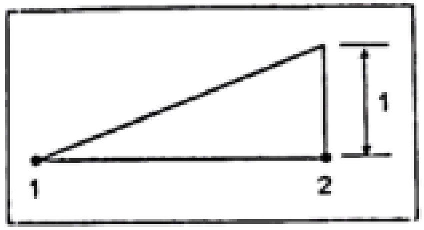


Fig. 2.13. Variation of L_2

Shape Function

If the values of the field variable are computed only at nodes, how are values obtained at other nodal points within a finite element? This is a most important point of finite element analysis.

The values of the field variable computed at the nodes are used to approximate the values at non-nodal points by interpolation of the nodal values.

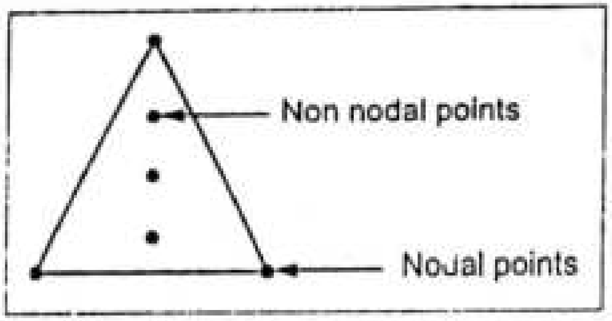


Fig. 2.17.

Consider the three noded triangular element as shown in Fig.2.17.

The nodes are exterior and at any point within the element the field variable is described by the following approximate relation.

$$\phi(x, y) = N_1(x, y)\,\phi_1 + N_2(x, y)\,\phi_2 + N_3(x, y)\,\phi_3$$

where ϕ_1, ϕ_2, ϕ_3 are the values of the field variable at the nodes, and N_1, N_2 and N_3 are the interpolation functions. N_1, N_2 and N_3 are also called as shape functions because they are used to express the geometry or shape of the element. Shape function has unit value at one nodal point and zero value at other nodal points.

In one dimensional problem, the basic field variable is displacement.

So, $$u = \Sigma N_i u_i \quad \text{where } u \rightarrow \text{Displacement.}$$

For two noded bar element, the displacement at any point within the element is given by,

$$u = \Sigma N_i u_i = N_1 u_1 + N_2 u_2$$

where, u_1 and u_2 are nodal displacements.

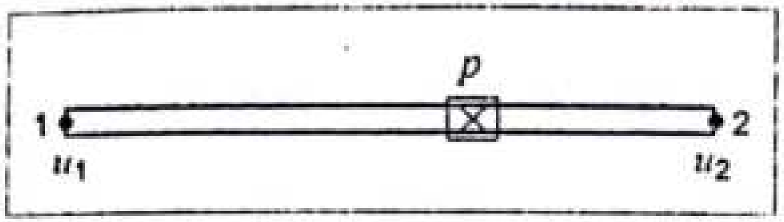


Fig. 2.18.

In two dimensional stress analysis problem, the basic field variable is displacement.

So, $$u = \Sigma N_i u_i$$

$$v = \Sigma N_i v_i$$

For three noded triangular element, the displacement at any point within the element is given by,

$$u = \Sigma N_i u_i = N_1 u_1 + N_2 u_2 + N_2 u_3$$

$$v = \Sigma N_i v_i = N_1 v_1 + N_2 v_2 + N_3 v_3$$

where, u_1, u_2, u_3, v_1, v_2 and v_3 are nodal displacements.

In general, shape functions need to satisfy the following:

1. First derivatives should be finite within an element.
2. Displacement should be continuous across the element boundary.

The characteristics of shape function are:

1. The shape function has unit value at its own nodal point and zero value at other nodal points.
2. The sum of shape function is equal to one.
3. The shape functions for two dimensional elements are zero along each side that the node does not touch.
4. The shape functions are always polynomials of the same type as the original interpolation equations.

2.6.2. Polynomial Shape Functions

Polynomials are generally used as shape function due to the following reasons.

1. Differentiation and integration of polynomials are quite easy.
2. It is easy to formulate and computerize the finite element equations.
3. The accuracy of the results can be improved by increasing the order of the polynomial.

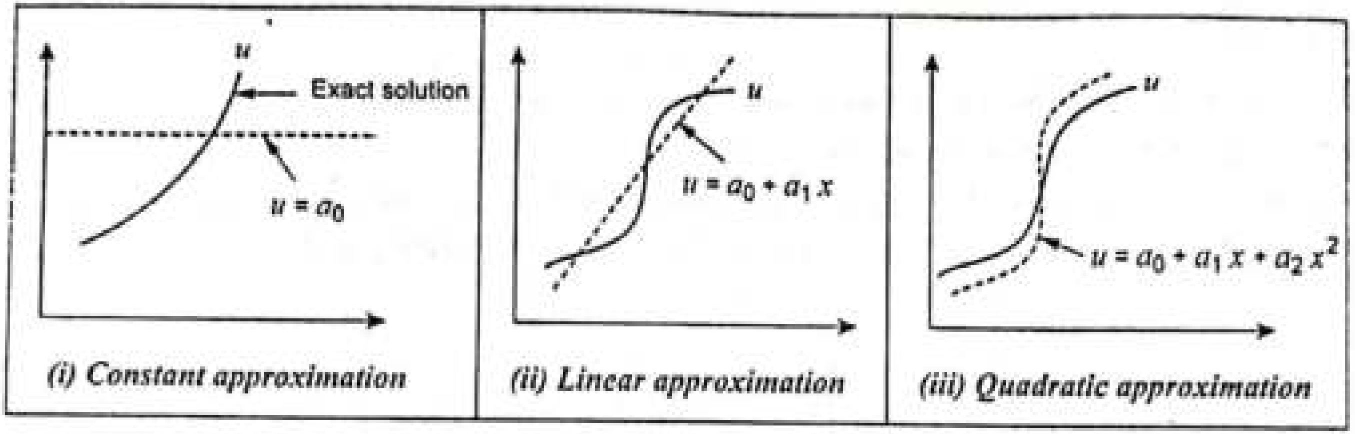


Fig. 2.19. Approximation of a function by polynomials of different order

Let us consider displacement u is a field variable.

Case (i): Linear polynomial

For one dimensional problem, $u = a_0 + a_1 x$

For two dimensional problem, $u(x, y) = a_0 + a_1 x + a_2 y$

For three dimensional problem, $u(x, y, z) = a_0 + a_1 x + a_2 y + a_3 z$

Case (ii): Quadratic polynomial

For one dimensional problem,

$$u = a_0 + a_1 x + a_2 x^2$$

For two dimensional problem,

$$u(x, y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 y^2 + a_5 xy$$

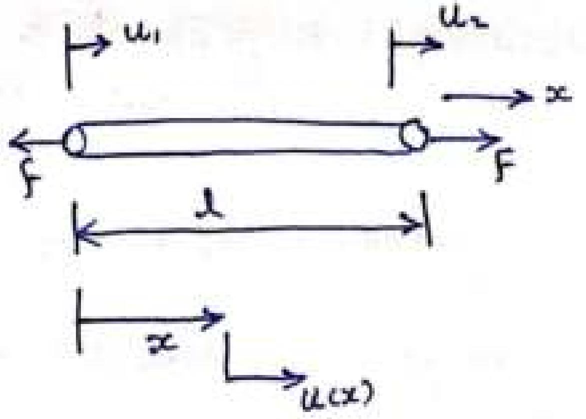
For three dimensional problem,

$$u(x, y, z) = a_0 + a_1 x + a_2 y + a_3 z + a_4 x^2 + a_5 y^2 + a_6 z^2 + a_7 xy + a_8 yz + a_9 xz$$

Shape function of a Bar Element

Consider a bar element with nodes 1 and 2 as shown in Fig.2.20. u_1 and u_2 are the displacements at the respective nodes. So, u_1 and u_2 are considered as degrees of freedom of this bar element.

[*Note:* Degrees of freedom is nothing but nodal displacements.]



Consider an elastic bar element (spar or link) element acted upon by a tensile force 'F' with following assemption.

1. The bar is geometrically stright
2. The material obeys Hook's law
3. Forces are applied only at the ends of the bar
4. The bar supports axial loading only bending, torsion and shear are not transmitted to the element.

The forces excerted on the ends of the element are colinear, equal in mangnitude and opposite in sense.

The bar of length 'L' is affixed a uniaxial co-ordinate system x with its origin ~~Ploted~~ placed at left end.

Axial displacement at any position along the length of the bar is $u(x)$.

Defining nodes 1 and 2 at each end, the nodal displacement are,

$$u = u_1 \quad \text{at} \quad x = 0$$

$$u = u_2 \quad \text{at} \quad x = l$$

Selecting a polynomial function as displacement function to represent the linear displacement of bar under load,

$$u(x) = a_0 + a_1 x \quad \text{---} (1)$$

In general, the total number of co-efficient in the function is equal to the total number of degrees of freedom associated with the element. Here there are two degrees of freedom.

ie, axial displacement at each of the two nodes,

eq. ① in matrix form,

$$u(x) = [1, x]\begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix}$$

Expressing $u(x)$ as a function of nodal displacement u_1 & u_2.

ie, by applying boundary conditions

when $x = 0$;

$$u(0) = u_1 = a_0$$

when $x = l$, $u(l) = u_2 = a_0 + a_1 l$

$$u_2 = u_1 + a_1 l$$

Solving for a_1; $a_1 = \dfrac{u_2 - u_1}{l}$

Substituting values of a_0 & a_1 in (1)

$$u(x) = \left(\frac{u_2 - u_1}{l}\right)x + u_1 \quad \text{--- (2)}$$

Equ ② in matrix form

$$u(x) = \begin{bmatrix} 1-\frac{x}{l} & \frac{x}{l} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

$$u(x) = \begin{bmatrix} N_1 , & N_2 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

where

$$\left.\begin{aligned} N_1 &= 1-\frac{x}{l} \\ N_2 &= \frac{x}{l} \end{aligned}\right\}$$ are called Shape function.

$$u(x) = N_1 u_1 + N_2 u_2$$

$$u = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = [u_1 , u_2]^T$$

is element displacement vector

where, Shape function, $N_1 = \frac{l-x}{l}$; Shape function, $N_2 = \frac{x}{l}$

We may note that N_1 and N_2 obey the definition of shape function, *i.e.*, the shape function will have a value equal to unity at the node to which it belongs and zero value at other nodes.

Checking: At node 1, $x = 0$.

$$\Rightarrow N_1 = \frac{l-x}{l} = \frac{l-0}{l}$$

$$\boxed{N_1 = 1}$$

$$\Rightarrow N_2 = \frac{x}{l} = \frac{0}{l}$$

$$\boxed{N_2 = 0}$$

At node 2, $x = l$

$$\Rightarrow N_1 = \frac{l-x}{l} = \frac{l-l}{l}$$

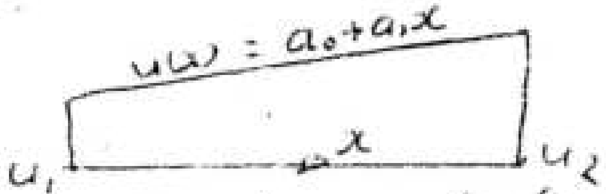
$$\boxed{N_1 = 0}$$

$$\Rightarrow N_2 = \frac{x}{l} = \frac{l}{l}$$

$$\boxed{N_2 = 1}$$

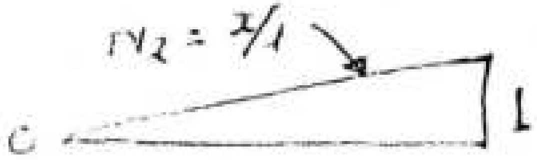
Characteristics of shape functions of bar element

1. shape functions N_1 & N_2 represent the shape of the assumed displacement function over the domain. N_1 & N_2 are linear functions that have properties $N_1 = 1$ at node 1 ; $N_1 = 0$ at Node 2, where as $N_2 = 1$ at node 2 ; $N_2 = 0$ at Node 1



Displacement function

N_1 over the domain



N_2 over the domain

2. $u(x) = N_1 u_1 + N_2 u_2$, The field variable $u(x)$ within the element depends upon the nodal values u_1 & u_2 and the actual magnitude depends on the shape functions N_1 & N_2.

3. Each shape function has a value of unity at its own node and zero at the other node.

4. Sum of all shape functions is equal to one

ie:- $N_1 + N_2 = 1 - x/l + x/l = 1$

or generally $\Sigma N_i = 1$, for $i = 1, 2 \ldots n$

where n = total number of nodes in an element.

5. The shape functions are always polynomials of the same degree as the original displacement function.

6. The sum of the derivatives of shape functions with respect to x will be equal to zero.

Stiffness Matrix

For a structural finite element, the stiffness matrix is a matrix which represents the primary characteristics of the element. The stiffness matrix contains informations regarding the geometric and material behaviour. It indicates the resistance of the element to change when subjected to external influences.

2.7. STIFFNESS MATRIX [K]

In order to get an expression for the stiffness matrix in finite element method, let us review the strain energy expression in structural mechanics.

Consider $\omega_1, \omega_2 \ldots\ldots \omega_n$ are nodal displacement parameters or otherwise known as degrees of freedom, $W_1, W_2, \ldots\ldots W_n$ are the corresponding nodal loads acting at degrees of freedom. $\{\omega\}$ and $\{W\}$ are column matrix.

$$\{W\} = \begin{Bmatrix} W_1 \\ W_2 \\ W_3 \\ \vdots \\ W_n \end{Bmatrix}$$

$$\{\omega\} = \begin{Bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \vdots \\ \omega_n \end{Bmatrix}$$

We know that, $\{W\} = [K]\{\omega^*\}$ ... (2.22)

where, W = Nodal loads.

K = Stiffness matrix.

ω^* = Degrees of freedom.

Strain Energy

The energy which is absorbed in a body, when strained within elastic limit is known as strain energy. Strain energy is always capable of doing work.

Strain Energy = Work Done

Strain energy is defined as the energy absorbed by a material upto elastic limit i. e.

Strain energy is an enery storing parameter which absorbs energy of a material upto thier elastic limit

The formula of strain enery can be given by

Strain energy (U) =1/2*load*deflection

From equation (2.22), we know that, nodal loads and the corresponding degrees of freedom are linked through stiffness matrix.

We know that,

$$\text{Work done, } P = \text{Strain energy}$$

$$\Rightarrow P = \frac{1}{2} W_1 \omega_1 + \frac{1}{2} W_2 \omega_2 + \frac{1}{2} W_3 \omega_3 + \ldots\ldots\ldots + \frac{1}{2} W_n \omega_n$$

We can write this equation in matrix form,

i.e.,

$$P = \frac{1}{2} [W_1 \; W_2 \; W_3 \; \ldots\ldots \; W_n] \begin{Bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \vdots \\ \omega_n \end{Bmatrix}$$

$$P = \frac{1}{2} \{ W \}^T \{ \omega^* \} \qquad \ldots (2.23)$$

[*Note:* [ ] → Row matrix; { } → Column matrix]

Substitute equation (2.22) in equation (2.23),

$$\Rightarrow P = \frac{1}{2}\left\{[K]\{\omega^*\}\right\}^T \{\omega^*\}$$

$$= \frac{1}{2}\{K\}^T \{\omega^*\}^T \{\omega^*\}$$

$$\boxed{P = \frac{1}{2}\{\omega^*\}^T [K] \{\omega^*\}} \qquad \ldots (2.24)$$

[$\because$ K is a symmetric matrix. So, $[K]^T = [K]$]

Equation (2.24) is a strain energy equation for a structure.

Our aim is to find the expression for stiffness matrix [K]. Let us consider one dimensional element. $u_1, u_2, u_3 \ldots\ldots, u_n$ are the degrees of freedom of that element.

We know that,

$$\text{Strain, } \{e\} = [B]\{u^*\} \qquad \ldots (2.25)$$

$$\Rightarrow \boxed{\{e\}^T = [B]^T \{u^*\}^T} \qquad \ldots (2.26)$$

where, $\{e\}$ is a strain matrix [Column matrix].

$[B]$ is a strain-displacement matrix [Row matrix].

$\{u^*\}$ is a degree of freedom [Column matrix]

We know that,

$$\text{Stress } \{\sigma\} = [E]\{e\}$$

$$\boxed{\{\sigma\} = [D]\{e\}} \qquad \dots (2.27)$$

where, $[E] = [D] =$ Young's modulus.

Strain energy expression is given by,

$$U = \int_v \frac{1}{2} \{e\}^T \{\sigma\}\, dv \qquad \dots (2.28)$$

Substitute $\{e\}^T$ and $\{\sigma\}$ values,

$$\Rightarrow U = \int_v \frac{1}{2} [B]^T \{u^*\}^T [D] \{e\}\, dv$$

$$= \frac{1}{2} \{u^*\}^T \int_v [B]^T [D] \{e\}\, dv$$

When stress σ is proportional to strain ϵ, the strain energy

U = **1 / 2 V σ ϵ**

Where,

σ = stress,

ϵ = strain,

V = volume of body

Substitute $\{ e \}$ value,

$$\Rightarrow U = \frac{1}{2} \{ u^* \}^T \int_v [B]^T [D] [B] \{ u^* \} \, dv$$

$$U = \frac{1}{2} \{ u^* \}^T \left[\int_v [B]^T [D] [B] \, dv \right] \{ u^* \} \qquad \dots (2.29)$$

From equation (2.24), we know that,

$$P = \frac{1}{2} \{ \omega^* \}^T [K] \{ \omega^* \} \qquad \dots (2.24)$$

Comparing equation (2.29) and (2.24),

$$\Rightarrow \{ \omega^* \}^T = \{ u^* \}^T$$

$$\{ \omega^* \} = \{ u^* \}$$

$$[K] = \int_v [B]^T [D] [B] \, dv$$

So, $\boxed{\text{Stiffness matrix, } [K] = \int_v [B]^T [D] [B] \, dv}$ $\qquad \dots (2.30)$

where, $[B] \rightarrow$ Strain displacement relationship matrix.

$[D] \rightarrow$ Elasticity matrix or Stress-strain relationship matrix.

In one dimensional problem,

$$\text{Strain, } e = \frac{du}{dx}$$

where, $u \rightarrow$ Displacement function.

$$[D] = [E] = E = \text{Young's modulus.}$$

In Beam problem,

$$\text{Strain, } e = \text{Curvature} = \frac{d^2u}{dx^2}$$

$$[D] = [EI] = \text{Flexural rigidity.}$$

2.7.1. Properties of Stiffness Matrix

1. It is a symmetric matrix.
2. The sum of elements in any column must be equal to zero.
3. It is an unstable element. So, the determinant is equal to zero.
4. The dimension of the global stiffness matrix [K] is N × N, where N is the number of nodes. This follows from the fact that each node has only one degree of freedom.
5. The diagonal coefficients are always positive and relatively large when compared to the off-diagonal values in the same row.

2.7.2. Derivation of Stiffness Matrix for One Dimensional Bar Element

Consider a one dimensional bar element with nodes 1 and 2 as shown in Fig.2.21. Let u_1 and u_2 be the nodal displacement parameters or otherwise known as degrees of freedom.

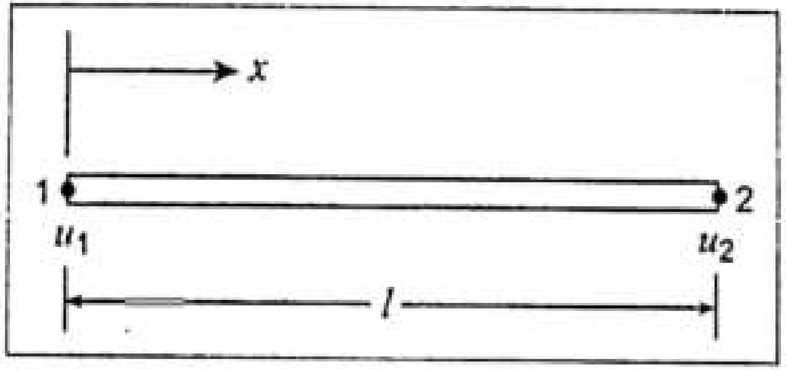


Fig. 2.21. A bar element with two nodes

We know that,

$$\text{Stiffness matrix } [K] = \int_v [B]^T [D] [B] \, dv \quad \text{[From equation no.(2.30)]}$$

In one dimensional bar element,

$$\text{Displacement function, } u = N_1 u_1 + N_2 u_2 \quad \text{[From equation no.(2.21)]}$$

$$\text{where, } N_1 = \frac{l - x}{l}$$

$$N_2 = \frac{x}{l}$$

We know that,

$$\text{Strain-Displacement matrix, } [B] = \begin{bmatrix} \frac{dN_1}{dx} & \frac{dN_2}{dx} \end{bmatrix}$$

$$[B] = \begin{bmatrix} \frac{-1}{l} & \frac{1}{l} \end{bmatrix} \quad \ldots (2.31)$$

$$\Rightarrow [B]^T = \begin{Bmatrix} \frac{-1}{l} \\ \frac{1}{l} \end{Bmatrix} \quad \ldots (2.32)$$

In one dimensional problems, [D] = [E] = E = Young's modulus ... (2.33)

Substitute [B], [B]^T and [D] values in stiffness matrix equation. [Limit is 0 to l].

$$\Rightarrow [K] = \int_0^l \begin{Bmatrix} \frac{-1}{l} \\ \frac{1}{l} \end{Bmatrix} \times E \times \begin{bmatrix} \frac{-1}{l} & \frac{1}{l} \end{bmatrix} dv = \int_0^l \begin{bmatrix} \frac{1}{l^2} & \frac{-1}{l^2} \\ \frac{-1}{l^2} & \frac{1}{l^2} \end{bmatrix} E\, dv$$

[$\because$ Matrix multiplication $(2 \times \underbrace{1) \times (1}_{} \times 2) = (2 \times 2)$]

$$= \int_0^l \begin{bmatrix} \frac{1}{l^2} & \frac{-1}{l^2} \\ \frac{-1}{l^2} & \frac{1}{l^2} \end{bmatrix} E\, A\, dx \qquad [\because dv = A\, dx]$$

$$= A\, E \begin{bmatrix} \frac{1}{l^2} & \frac{-1}{l^2} \\ \frac{-1}{l^2} & \frac{1}{l^2} \end{bmatrix} \int_0^l dx = A\, E \begin{bmatrix} \frac{1}{l^2} & \frac{-1}{l^2} \\ \frac{-1}{l^2} & \frac{1}{l^2} \end{bmatrix} [x]_0^l$$

$$= A\, E \begin{bmatrix} \frac{1}{l^2} & \frac{-1}{l^2} \\ \frac{-1}{l^2} & \frac{1}{l^2} \end{bmatrix} (l - 0) = A\, E\, l \begin{bmatrix} \frac{1}{l^2} & \frac{-1}{l^2} \\ \frac{-1}{l^2} & \frac{1}{l^2} \end{bmatrix}$$

$$= \frac{AEl}{l^2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$\boxed{[K] = \frac{AE}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}} \qquad \ldots (2.34)$$

The properties of a stiffness matrix are satisfied.

1. It is symmetric.
2. The sum of elements in any column is equal to zero.

2.8. DERIVATION OF FINITE ELEMENT EQUATION FOR ONE DIMENSIONAL BAR ELEMENT

We know that, General force equation is,

$$\{F\} = [K]\{u\} \qquad \ldots (2.35)$$

where, $\{F\}$ is a element force vector [Column matrix].

$[K]$ is a stiffness matrix [Row matrix].

$\{u\}$ is a nodal displacement [Column matrix].

For one dimensional bar element, stiffness matrix [K] is given by,

$$[K] = \frac{AE}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad \text{[From equation no.(2.34)]}$$

For two noded bar element,

$$\{F\} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix}$$

$$\{u\} = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

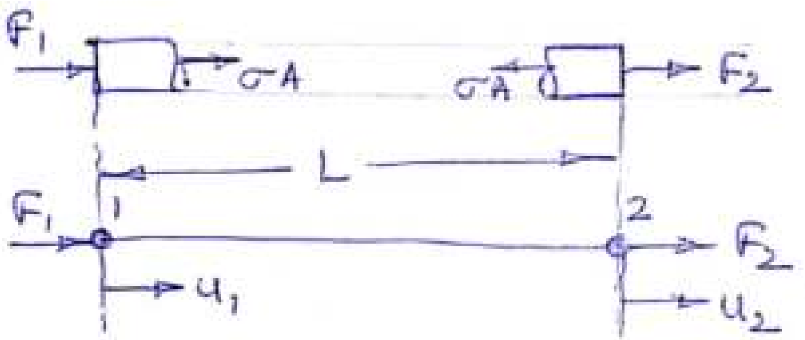
Substitute [K] { F } and { u } values in equation (2.35),

$$\Rightarrow \boxed{\begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \frac{AE}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}} \qquad \dots (2.36)$$

This is a finite element equation for one dimensional two noded bar element.

Another Method

Stiffness matrix of a 1D bar element in local co-ordinates – Direct method



Consider a uniform prismatic elastic bar element of length L and elastic modulus E, and having a cross sectional area of A. The axial displacements and node 1 and 2 are u_1 and u_2. The internal axial stress σ can be related to nodal forces F_1 and F_2 by free body diagram.

We can write

$F_1 + A\sigma = 0$ — (1) and $F_2 - A\sigma = 0$ — (2)

$\sigma = E\varepsilon$ and $\varepsilon = \dfrac{u_2 - u_1}{L}$

$\therefore$ (1) $\rightarrow F_1 + AE\dfrac{u_2 - u_1}{L} = 0$

(2) $\rightarrow F_2 - AE\dfrac{u_2 - u_1}{L} = 0$

or $F_1 = \dfrac{AE}{L}(u_1 - u_2)$

$F_2 = \dfrac{AE}{L}(u_2 - u_1)$

or $\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$

where $k = \dfrac{AE}{L}$

We can write

$$\boxed{[F] = [K_e][u]}$$

where $[K_e] = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix}$ is defined as the element stiffness matrix in local co-ordinates

$[F]$ – is known as nodal force vector.

$[u]$ – is known as displacement vector.

The ~~nodal~~ element stiffness matrix for a bar element is symmetric, singular and of the order 2x2 in correspondence with two nodal displacement or degrees of freedom.

Assembling the stiffness equations for global equation

2.9. ASSEMBLING THE STIFFNESS EQUATIONS OR GLOBAL EQUATIONS

Consider a bar as shown in Fig.2.22(*a*) This bar can be equally divided into 4 elements as shown in Fig.2.22(*b*).

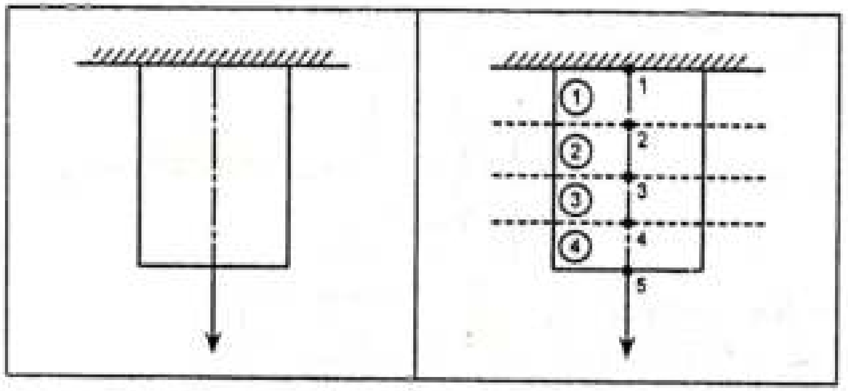


Fig. 2.22. (a) *Fig. 2.22. (b)*

Now the bar has 4 elements with 5 nodes.

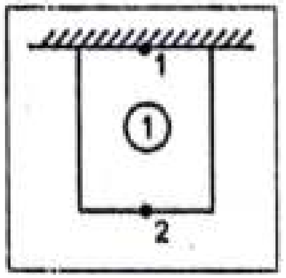
[*Note:* A number with circle denotes element and without circle denotes nodes]

We know that,

Finite element equation for two noded bar element is,

$$\begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \frac{AE}{l} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

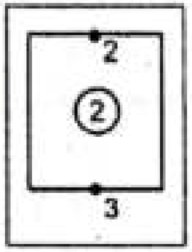
For element (1) (Nodes 1, 2):



Finite element equation is,

$$\begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \frac{AE}{l} \begin{bmatrix} \overset{a_{11}}{1} & \overset{a_{12}}{-1} \\ \overset{a_{21}}{-1} & \overset{a_{22}}{1} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \qquad \dots (2.37)$$

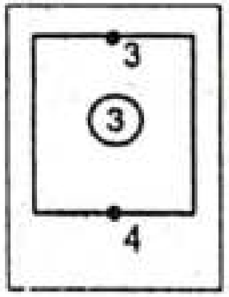
For element (2) (Nodes 2, 3):



Finite element equation is,

$$\begin{Bmatrix} F_2 \\ F_3 \end{Bmatrix} = \frac{AE}{l} \begin{bmatrix} \overset{a_{22}}{1} & \overset{a_{23}}{-1} \\ \overset{a_{32}}{-1} & \overset{a_{33}}{1} \end{bmatrix} \begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix} \qquad \dots (2.38)$$

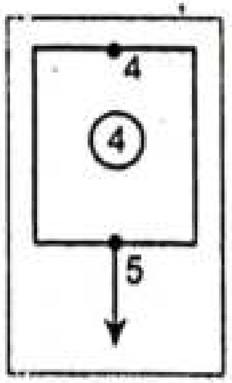
For element (3) (Nodes 3, 4):



Finite element equation is,

$$\begin{Bmatrix} F_3 \\ F_4 \end{Bmatrix} = \frac{AE}{l} \begin{bmatrix} \overset{a_{33}}{1} & \overset{a_{34}}{-1} \\ \overset{a_{43}}{-1} & \overset{a_{44}}{1} \end{bmatrix} \begin{Bmatrix} u_3 \\ u_4 \end{Bmatrix} \qquad \dots (2.39)$$

For element (4) (Nodes 4, 5):



Finite element equation is,

$$\begin{Bmatrix} F_4 \\ F_5 \end{Bmatrix} = \frac{AE}{l} \begin{bmatrix} \overset{a_{44}}{1} & \overset{a_{45}}{-1} \\ \overset{a_{54}}{-1} & \overset{a_{55}}{1} \end{bmatrix} \begin{Bmatrix} u_4 \\ u_5 \end{Bmatrix} \quad \ldots (2.40)$$

Assembling the equations (2.37), (2.38), (2.39) and (2.40),

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \end{Bmatrix} = \frac{AE}{l} \begin{bmatrix} \overset{a_{11}}{1} & \overset{a_{12}}{-1} & \overset{a_{13}}{0} & \overset{a_{14}}{0} & \overset{a_{15}}{0} \\ \overset{a_{21}}{-1} & \overset{a_{22}}{1+1} & \overset{a_{23}}{-1} & \overset{a_{24}}{0} & \overset{a_{25}}{0} \\ \overset{a_{31}}{0} & \overset{a_{32}}{-1} & \overset{a_{33}}{1+1} & \overset{a_{34}}{-1} & \overset{a_{35}}{0} \\ \overset{a_{41}}{0} & \overset{a_{42}}{0} & \overset{a_{43}}{-1} & \overset{a_{44}}{1+1} & \overset{a_{45}}{-1} \\ \overset{a_{51}}{0} & \overset{a_{52}}{0} & \overset{a_{53}}{0} & \overset{a_{54}}{-1} & \overset{a_{55}}{1} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{Bmatrix}$$

[*Note:* The bar has 5 nodes and each node has one degree of freedom. So, the global stiffness matrix size is 5×5]

$$[K]_{global} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Load Vector

- Consider a vertically hanging bar of length *l* , uniform cross-section area *A*, desity *ρ*, and youngs modulus *E*, this bar is subjected to self weight X_b

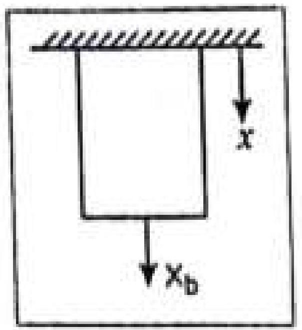


Fig. 2.23. Vertically hanging bar with self weight

The element nodal force vector is given by,

$$\{F\}_e = \int [N]^T X_b \qquad \dots (2.41)$$

We know that,

Self weight due to loading force, $X_b = \rho A \, dx \qquad \dots (2.42)$

For one dimensional bar element, the displacement function is given by,

$$u = N_1 u_1 + N_2 u_2 \qquad \text{[From equation no.(2.21)]}$$

where, $N_1 = \frac{l-x}{l}$

$$N_2 = \frac{x}{l}$$

$$\Rightarrow [N] = \begin{bmatrix} \frac{l-x}{l} & \frac{x}{l} \end{bmatrix}$$

$$\Rightarrow [N]^T = \begin{Bmatrix} \frac{l-x}{l} \\ \frac{x}{l} \end{Bmatrix} \qquad \dots (2.43)$$

Substitute X_b and $[N]^T$ values in equation (2.41),

$$\Rightarrow \{F\}_e = \int_0^l \begin{Bmatrix} \frac{l-x}{l} \\ \frac{x}{l} \end{Bmatrix} \rho A \, dx = \rho A \int_0^l \begin{Bmatrix} \frac{l-x}{l} \\ \frac{x}{l} \end{Bmatrix} dx$$

$$= \rho A \int_0^l \begin{Bmatrix} 1 - \frac{x}{l} \\ \frac{x}{l} \end{Bmatrix} dx = \rho A \int_0^l \begin{Bmatrix} dx - \frac{x \, dx}{l} \\ \frac{x \, dx}{l} \end{Bmatrix}$$

$$= \rho A \begin{Bmatrix} x - \dfrac{x^2}{2l} \\ \dfrac{x^2}{2l} \end{Bmatrix}_0^l = \rho A \begin{Bmatrix} l - \dfrac{l^2}{2l} \\ \dfrac{l^2}{2l} \end{Bmatrix} = \rho A \begin{Bmatrix} l - \dfrac{l}{2} \\ \dfrac{l}{2} \end{Bmatrix}$$

$$= \rho A \begin{Bmatrix} \dfrac{l}{2} \\ \dfrac{l}{2} \end{Bmatrix}$$

Force vector, $$[F]_e = \frac{\rho A l}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \qquad \ldots (2.44)$$

- Solved Problems(One dimensional)

Example 2.5 *A two noded truss element is shown in Fig.(i). The nodal displacements are $u_1 = 5$ mm and $u_2 = 8$ mm. Calculate the displacement at $x = \frac{l}{4}, \frac{l}{3}$ and $\frac{l}{2}$.*

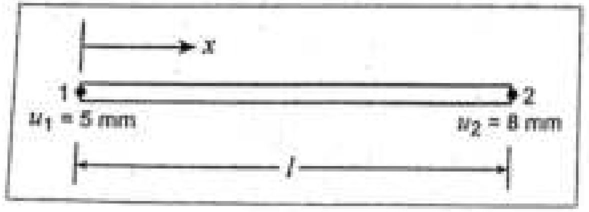


Fig. (i)

Given: Displacements, $u_1 = 5$ mm

$u_2 = 8$ mm

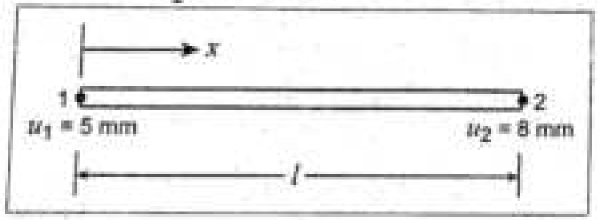


Fig. (ii)

To find: Displacement u at $x = \frac{l}{4}, \frac{l}{3}$ and $\frac{l}{2}$.

☺ ***Solution:*** Displacement function for two noded truss element is given by,

$$u = N_1 u_1 + N_2 u_2 \qquad \text{[From equation (2.21)]}$$

where, $N_1 = \frac{l-x}{l}$

$$N_2 = \frac{x}{l}$$

$$\Rightarrow \quad u = \left[\frac{l-x}{l}\right] u_1 + \left[\frac{x}{l}\right] u_2 \qquad \ldots (1)$$

Substitute $x = \frac{l}{4}$, $u_1 = 5$ and $u_2 = 8$ in equation (1),

$$\Rightarrow \quad u = \left[\frac{l - \frac{l}{4}}{l}\right] \times 5 + \left[\frac{\frac{l}{4}}{l}\right] \times 8$$

$$= \left[1 - \frac{1}{4}\right] \times 5 + \left[\frac{1}{4}\right] \times 8$$

$$\boxed{u = 5.75 \text{ mm}} \quad \text{at } x = \frac{l}{4}$$

Substitute $x = \frac{l}{3}$, $u_1 = 5$ mm and $u_2 = 8$ mm in equation (1)

$$(1) \Rightarrow \quad u = \left[\frac{l - \frac{l}{3}}{l}\right] \times 5 + \left[\frac{\frac{l}{3}}{l}\right] \times 8$$

$$= \left[1 - \frac{1}{3}\right] \times 5 + \left[\frac{1}{3}\right] \times 8$$

$$\boxed{u = 6 \text{ mm}} \quad \text{at } x = \frac{l}{3}$$

Substitute $x = \frac{l}{2}$, $u_1 = 5$ mm and $u_2 = 8$ mm in equation (1),

(1) $\Rightarrow$

$$u = \left[\frac{l - \frac{l}{2}}{l}\right] \times 5 + \left[\frac{\frac{l}{2}}{l}\right] \times 8$$

$$= \left[1 - \frac{1}{2}\right] \times 5 + \left[\frac{1}{2}\right] \times 8$$

$$\boxed{u = 6.5 \text{ mm}} \quad \text{at } x = \frac{l}{2}$$

Result:

$$u = 5.75 \text{ mm at } x = \frac{l}{4}$$

$$u = 6 \text{ mm at } x = \frac{l}{3}$$

$$u = 6.5 \text{ mm at } x = \frac{l}{2}$$

Example 2.6 *A one dimensional bar is shown in Fig.(i). Calculate the following:*

(i) Shape function N_1 and N_2 at point P.

(ii) If $u_1 = 3$ mm and $u_2 = -5$ mm, calculate the displacement u at point P.

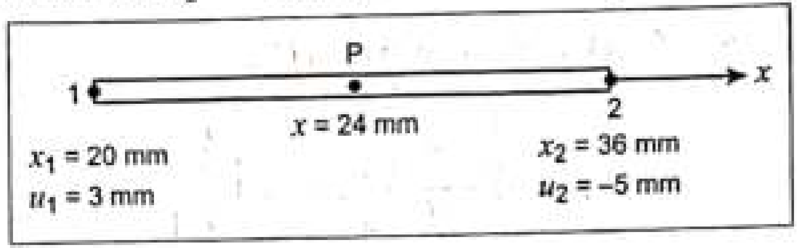


Fig. (i)

Given: $x_1 = 20$ mm, $x_2 = 36$ mm, $u_1 = 3$ mm, $u_2 = -5$ mm

$x = 24$ mm

To find:
1. Shape function N_1 and N_2 at point P.
2. Displacement u at point P.

☺ *Solution:*

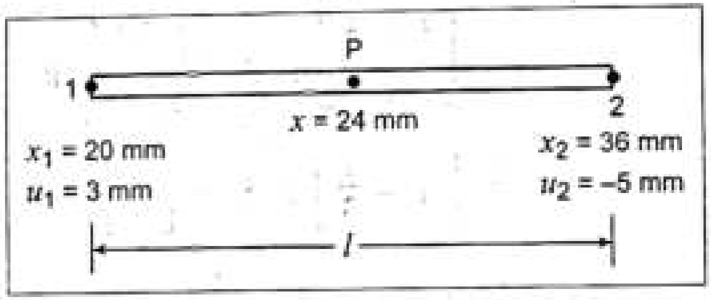


Fig. (ii)

We know that,

$$\text{Actual length of the bar, } l = x_2 - x_1 = 36 - 20$$

$$\boxed{l = 16 \text{ mm}}$$

The distance between point 1 and point P is,

$$x = 24 - 20$$

$$\Rightarrow \boxed{x = 4 \text{ mm}}$$

Displacement function for two noded bar element is given by,

$$u = N_1 u_1 + N_2 u_2 \qquad \ldots (1)$$

[From equation no.(2.21)]

$$\text{where, } N_1 = \frac{l - x}{l}$$

$$N_2 = \frac{x}{l}$$

$$\Rightarrow N_1 = \frac{16-4}{16}$$

$$\boxed{N_1 = 0.75 \text{ mm}}$$

$$\Rightarrow N_2 = \frac{4}{16}$$

$$\boxed{N_2 = 0.25 \text{ mm}}$$

Substitute N_1, N_2, u_1 and u_2 values in equation no.(1),

$$(1) \Rightarrow \quad u = N_1 u_1 + N_2 u_2$$

$$= (0.75)(3) + 0.25(-5)$$

$$\boxed{u = 1 \text{ mm}}$$

Result: 1. Shape function, $N_1 = 0.75$ mm

$N_2 = 0.25$ mm

2. Displacement u at point P is 1 mm.

Example 2.7 *Consider a bar as shown in Fig.(i). Cross-sectional area of the bar is 750 mm² and Young's modulus is 2×10^5 N/mm². If $u_1 = 0.5$ mm and $u_2 = 0.625$ mm, calculate the following:*

(i) Displacement at point, P — *(ii) Strain, e*

(iii) Stress, σ — *(iv) Element stiffness matrix, [K]*

(v) Strain energy, U

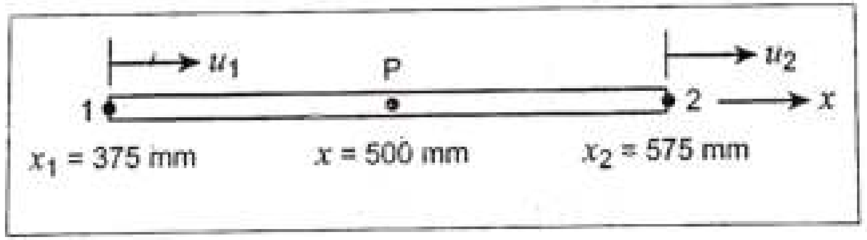


Fig. (i)

Given:

Area, $A = 750 \text{ mm}^2$

Young's modulus, $E = 2 \times 10^5 \text{ N/mm}^2$

Displacements, $u_1 = 0.5$ mm

$u_2 = 0.625$ mm

Distance, $x_1 = 375$ mm

$x_2 = 575$ mm

$x = 500$ mm

To find: (*i*) Displacement at point P, *i.e.*, u.

(*ii*) Strain, e

(*iii*) Stress, σ

(*iv*) Element stiffness matrix, [K]

(*v*) Strain energy, U

☺ ***Solution :***

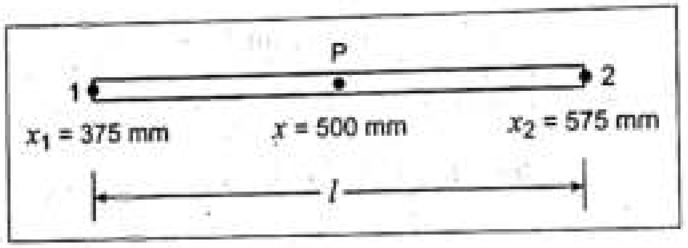


Fig. (ii)

We know that,

$$\text{Actual length of the bar, } l = x_2 - x_1 = 575 - 375$$

$$\boxed{l = 200 \text{ mm}}$$

The distance between point 1 and point P is,

$$x = 500 - 375$$

$$\boxed{x = 125 \text{ mm}}$$

We know that, Displacement function for two noded bar element is,

$$u = N_1 u_1 + N_2 u_2 \qquad \text{[From equation no.(2.21)]}$$

where, Shape function, $N_1 = \frac{l-x}{l}$

$$N_2 = \frac{x}{l}$$

$$\Rightarrow N_1 = \frac{200-125}{200}$$

$$\boxed{N_1 = 0.375}$$

$$\Rightarrow N_2 = \frac{x}{l} = \frac{125}{200}$$

$$\boxed{N_2 = 0.625}$$

Substitute N_1, N_2, u_1 and u_2 values in displacement equation,

$$\Rightarrow u = 0.375\,(0.5) + 0.625\,(0.625)$$

$$\boxed{\text{Displacement at point P, } u = 0.5781 \text{ mm}}$$

We know that, Strain, $e = [B]\{u^*\}$ [From equation no.(2.25)]

where, $[B]$ is a strain-displacement matrix.

$\{u^*\}$ is a degree of freedom.

$$\Rightarrow [B] = \begin{bmatrix} \frac{-1}{l} & \frac{1}{l} \end{bmatrix} \qquad \text{[From equation no.(2.31)]}$$

$$= \begin{bmatrix} \frac{-1}{200} & \frac{1}{200} \end{bmatrix}$$

$$\text{Strain, } e = [B]\{u^*\} = \begin{bmatrix} \frac{-1}{200} & \frac{1}{200} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

$$= \begin{bmatrix} \frac{-1}{200} & \frac{1}{200} \end{bmatrix} \begin{Bmatrix} 0.5 \\ 0.625 \end{Bmatrix}$$

$$= \left[\frac{-1}{200} \times 0.5 + \frac{1}{200} \times 0.625 \right]$$

$$\boxed{\text{Strain, } e = 6.25 \times 10^{-4}}$$

We know that, Stress, $\sigma = E\,e = 2 \times 10^5 \times 6.25 \times 10^{-4}$

$$\boxed{\sigma = 125 \text{ N/mm}^2}$$

For one dimensional bar element, stiffness matrix is given by,

$$[K] = \frac{AE}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \frac{750 \times 2 \times 10^5}{200}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$\boxed{[K] = 7.5 \times 10^5 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}}$$

We know that, Strain energy, $U = \frac{1}{2}\{u^*\}^T [K] \{u^*\}$

[From equation no.(2.24)]

$$= \frac{1}{2}[u_1 \;\; u_2] \times 7.5 \times 10^5 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

$$= \frac{1}{2}[0.5 \;\; 0.625] \times 7.5 \times 10^5 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} 0.5 \\ 0.625 \end{Bmatrix}$$

$$= \frac{1}{2}[0.5 \;\; 0.625] \times 7.5 \times 10^5 \begin{Bmatrix} 0.5 - 0.625 \\ -0.5 + 0.625 \end{Bmatrix} \qquad [\because (2 \times \underbrace{2) \times (2}) \times 1) = 2 \times 1]$$

$$= \frac{1}{2} \times 7.5 \times 10^5 \begin{bmatrix} 0.5 & 0.625 \end{bmatrix} \begin{Bmatrix} -0.125 \\ 0.125 \end{Bmatrix}$$

$$= \frac{1}{2} \times 7.5 \times 10^5 \; [0.5 \times (-0.125) + 0.625 \times 0.125]$$

$$\boxed{\text{Strain energy, } U = 5859.37 \text{ N-mm}}$$

Result:

(*i*) $u = 0.5781$ mm

(*ii*) $e = 6.25 \times 10^{-4}$

(*iii*) $\sigma = 125$ N/mm^2

(*iv*) $[K] = 7.5 \times 10^5$

(*v*) $U = 5859.37$ N-mm

Example 2.8 *A steel bar of length 800 mm is subjected to an axial load of 3 kN as shown in Fig. (i). Find the elongation of the bar, neglecting self weight.*

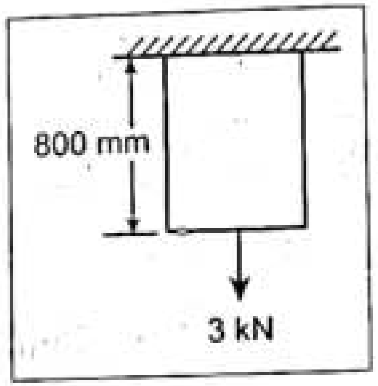


Fig. (i)

Take $E = 2 \times 10^5$ N/mm², $A = 300$ mm².

Given:

Length, l = 800 mm

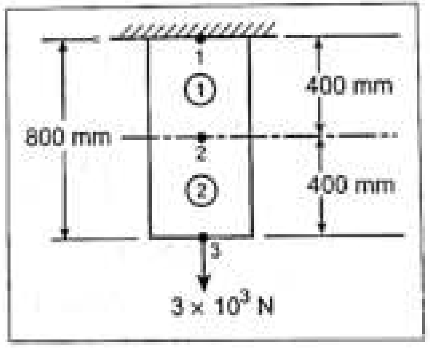
Load, F = 3 kN = 3×10^3 N

Young's modulus, E = 2×10^5 N/mm²

Area, A = 300 mm²

To find: Elongation, u

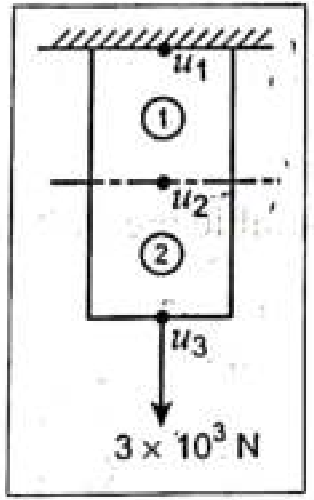
☺ *Solution:* We can divide the bar into two elements as shown in Fig.(*ii*).



Now the bar has 2 elements with 3 nodes.

[*Note:* Number with circle denotes Element & Number without circle denotes Node]

We can find the displacement at node 1, node 2 and node 3.



$u_1 = 0$

a_{11} a_{12} a_{13}

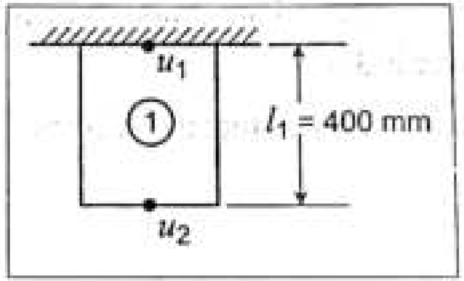
Displacement at node 1 is u_1, node 2 is u_2 and node 3 is u_3.

For one dimensional two noded bar element, the finite element equation is,

$$\begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \frac{AE}{l} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \quad \text{[From equation no.(2.36)]}$$

For element 1: (Nodes 1, 2):

Finite element equation is,



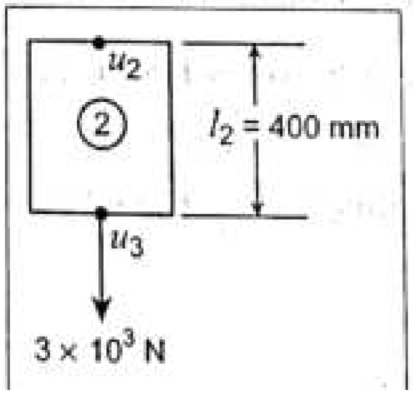
$$\frac{A_1 E}{l_1}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix}$$

$$\frac{300 \times 2 \times 10^5}{400}\begin{array}{c} \begin{array}{cc} a_{11} & a_{12} \end{array} \\ \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \\ \begin{array}{cc} a_{21} & a_{22} \end{array} \end{array}\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix}$$

$$150 \times 10^3 \begin{bmatrix} a_{11} & a_{12} \\ 1 & -1 \\ a_{21} & a_{22} \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} \qquad \text{...... (1)}$$

For element 2: (Nodes 2, 3):

Finite element equation is,



$$\frac{A_2 E}{l_2}\begin{array}{c} \begin{array}{cc} a_{22} & a_{23} \end{array} \\ \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \\ \begin{array}{cc} a_{32} & a_{33} \end{array} \end{array}\begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} F_2 \\ F_3 \end{Bmatrix}$$

$$\Rightarrow \qquad \frac{300 \times 2 \times 10^5}{400} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} F_2 \\ F_3 \end{Bmatrix}$$

$$\Rightarrow \qquad 150 \times 10^3 \begin{bmatrix} \overset{a_{22}}{1} & \overset{a_{23}}{-1} \\ \overset{a_{32}}{-1} & \overset{a_{33}}{1} \end{bmatrix} \begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} F_2 \\ F_3 \end{Bmatrix} \qquad \ldots (2)$$

Assemble the finite elements, *i.e.*, assemble the finite element equations (1) and (2).

$$\Rightarrow \qquad 150 \times 10^3 \begin{bmatrix} \overset{a_{11}}{1} & \overset{a_{12}}{-1} & \overset{a_{13}}{0} \\ \overset{a_{21}}{-1} & \overset{a_{22}}{1+1} & \overset{a_{23}}{-1} \\ \overset{a_{31}}{0} & \overset{a_{32}}{-1} & \overset{a_{33}}{1} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix}$$

$$\Rightarrow \qquad 150 \times 10^3 \underbrace{\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}}_{[K]_{\text{global}}} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} \qquad \ldots (3)$$

[*Note:* The rod has 3 nodes. Each node has single degree of freedom. So, the global stiffness matrix [K] size is 3 × 3.

It may be noted that the stiffness matrix properties are satisfied.

$$[K] = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

1. It is symmetric.
2. The sum of elements in any column is equal to zero.]

Applying Boundary Conditions

(*i*) Displacement at node 1 is zero. *i.e.*, $u_1 = 0$.

(*ii*) 3×10^3 N load is acting at node 3, *i.e.*, $F_3 = 3 \times 10^3$ N. Self-weight is neglected, so, $F_1 = F_2 = 0$.

Substitute u_1, F_1, F_2 and F_3 values in equation (3),

$$\Rightarrow \quad 150 \times 10^3 \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} 0 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 3 \times 10^3 \end{Bmatrix}$$

Here, $u_1 = 0$. So, neglect first row and first column of [K] matrix. Hence, the final reduced equation is,

$$\Rightarrow \quad 150 \times 10^3 \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 3 \times 10^3 \end{Bmatrix}$$

$$150 \times 10^3 (2u_2 - u_3) = 0 \qquad \ldots (4)$$

$$150 \times 10^3 (-u_2 + u_3) = 3 \times 10^3 \qquad \ldots (5)$$

Solving, $150 \times 10^3 (u_2) = 3 \times 10^3$

$$\boxed{u_2 = 0.02 \text{ mm}}$$

Substitute u_2 value in equation (4),

$$\Rightarrow \quad 150 \times 10^3 (2 \times 0.02 - u_3) = 0$$

$$\Rightarrow \quad 2 \times 0.02 - u_3 = 0$$

$$\Rightarrow \quad 2 \times 0.02 = u_3$$

$$\Rightarrow \quad \boxed{u_3 = 0.04 \text{ mm}}$$

Verification: We know that, Total elongation, $\delta L = \frac{pL}{AE}$

$$= \frac{3 \times 10^3 \times 800}{300 \times 2 \times 10^5}$$

$$\boxed{\delta L = 0.04 \text{ mm}}$$

Result:

1. Elongation or displacement at node 1, $u_1 = 0$

2. At node 2, $u_2 = 0.02$ mm

3. At node 3, $u_3 = 0.04$ mm

Module 6

Interpolation – selection of interpolation functions - CST element - isoparametric formulation (using minimum PE theorem) – Gauss-quadrature

Solution of 2D plane stress solid mechanics problems (linear static analysis)

Interpolation/shape function

If the values of the field variable are computed only at nodes, how are values obtained at other nodal points within a finite element? This is a most important point of finite element analysis.

The values of the field variable computed at the nodes are used to approximate the values at non-nodal points by interpolation of the nodal values.

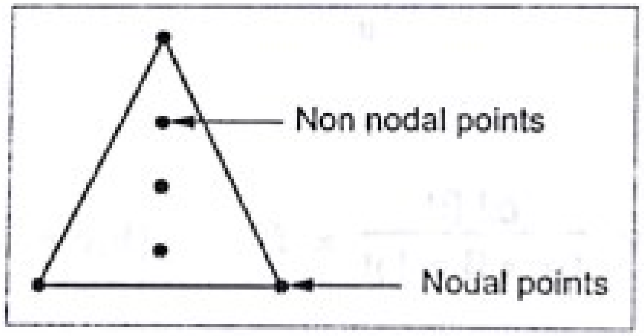


Fig. 2.17.

Consider the three noded triangular element as shown in Fig.2.17.

The nodes are exterior and at any point within the element the field variable is described by the following approximate relation.

$$\phi(x, y) = N_1(x, y)\,\phi_1 + N_2(x, y)\,\phi_2 + N_3(x, y)\,\phi_3$$

where ϕ_1, ϕ_2, ϕ_3 are the values of the field variable at the nodes, and N_1, N_2 and N_3 are the interpolation functions. N_1, N_2 and N_3 are also called as shape functions because they are used to express the geometry or shape of the element. Shape function has unit value at one nodal point and zero value at other nodal points.

In one dimensional problem, the basic field variable is displacement.

So, $$u = \Sigma N_1 u_i \quad \text{where } u \rightarrow \text{Displacement.}$$

For two noded bar element, the displacement at any point within the element is given by,

$$u = \Sigma N_i u_i = N_1 u_1 + N_2 u_2$$

where, u_1 and u_2 are nodal displacements.

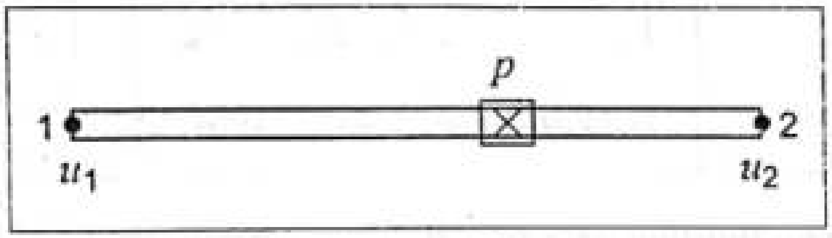


Fig. 2.18.

In two dimensional stress analysis problem, the basic field variable is displacement.

So, $$u = \Sigma N_i u_i$$
$$v = \Sigma N_i v_i$$

For three noded triangular element, the displacement at any point within the element is given by,

$$u = \Sigma N_i u_i = N_1 u_1 + N_2 u_2 + N_2 u_3$$
$$v = \Sigma N_i v_i = N_1 v_1 + N_2 v_2 + N_3 v_3$$

where, u_1, u_2, u_3, v_1, v_2 and v_3 are nodal displacements.

In general, shape functions need to satisfy the following:

1. First derivatives should be finite within an element.
2. Displacement should be continuous across the element boundary.

The characteristics of shape function are:

1. The shape function has unit value at its own nodal point and zero value at other nodal points.
2. The sum of shape function is equal to one.
3. The shape functions for two dimensional elements are zero along each side that the node does not touch.
4. The shape functions are always polynomials of the same type as the original interpolation equations.

2.6.2. Polynomial Shape Functions

Polynomials are generally used as shape function due to the following reasons.

1. Differentiation and integration of polynomials are quite easy.
2. It is easy to formulate and computerize the finite element equations.
3. The accuracy of the results can be improved by increasing the order of the polynomial.

The approximation of a non-linear one dimensional function by using polynomials of different order is shown in Fig.2.19.

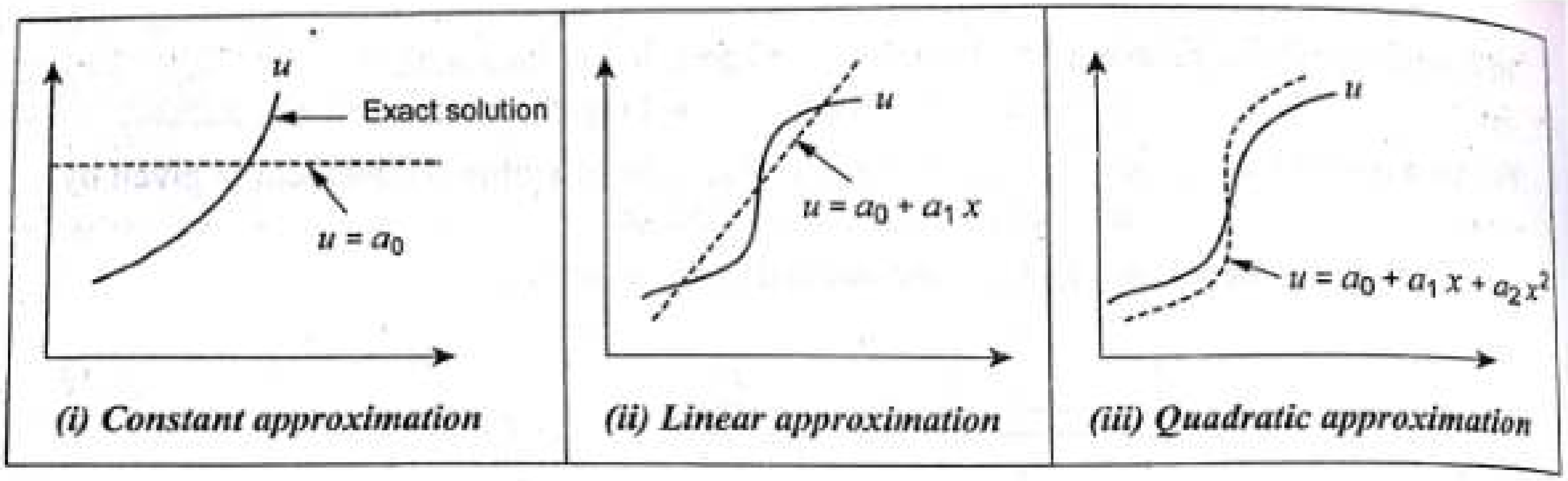


Fig. 2.19. Approximation of a function by polynomials of different order

Let us consider displacement u is a field variable.

Case (i): Linear polynomial

For one dimensional problem, $u = a_0 + a_1 x$

For two dimensional problem, $u(x, y) = a_0 + a_1 x + a_2 y$

For three dimensional problem, $u(x, y, z) = a_0 + a_1 x + a_2 y + a_3 z$

Case (ii): Quadratic polynomial

For one dimensional problem,

$$u = a_0 + a_1 x + a_2 x^2$$

For two dimensional problem,

$$u(x, y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 y^2 + a_5 xy$$

For three dimensional problem,

$$u(x, y, z) = a_0 + a_1 x + a_2 y + a_3 z + a_4 x^2 + a_5 y^2 + a_6 z^2 + a_7 xy + a_8 yz + a_9 xz$$

Interpolation/shape function

- In finite element analysis, the variations of displacement within an element are expressed by its Nodal displacement ($u = \Sigma N_i u_i$) with the help of interpolation function since the true variation of displacement inside the element is not known. Here, *u* is the displacement at any point inside the element and *ui* are the nodal displacements.
- This interpolating function is generally a polynomial with *n* degree which automatically provides a single-valued and continuous field.
- For linear interpolation, *n* will be 1 and for quadratic interpolation *n* will become 2 and so on.

Selection of interpolation functions

- Displacement function is the beginning point for the structural analysis by finite element method.
- This function represents the variation of the displacement within the element. On the basis of the problem to be solved, the displacement function needs to be approximated in the form of either linear or higher-order function.
- A convenient way to express it is by the use of polynomial expressions.

Selection based on,

- **Convergence criteria**
  - Accuracy is represented by a quality called convergence. By convergence, we mean that as we add more terms to the RR series, or as we add more nodes and elements into the mesh that replaces the original structure, the sequence of trial solutions must approach the exact solution
- **Compatibility**
- **Geometric invariance**

The convergence of the finite element solution can be achieved if the following three conditions are fulfilled by the assumed displacement function.

a. The displacement function must be continuous within the elements. This can be ensured by choosing a suitable polynomial. For example, for an n degrees of polynomial, displacement function in I dimensional problem can be chosen as:

$$u = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \alpha_4 x^4 + ... + \alpha_n x^n \tag{2.3.1}$$

b. The displacement function must be capable of rigid body displacements of the element. The constant terms used in the polynomial (α_0 to α_n) ensure this condition.

c. The displacement function must include the constant strains states of the element. As element becomes infinitely small, strain should be constant in the element. Hence, the displacement function should include terms for representing constant strain states.

2.3.1.2 Compatibility

Displacement should be compatible between adjacent elements. There should not be any discontinuity or overlapping while deformed. The adjacent elements must deform without causing openings, overlaps or discontinuous between the elements.

Elements which satisfy all the three convergence requirements and compatibility condition are called Compatible or Conforming elements.

2.3.1.3 Geometric invariance

Displacement shape should not change with a change in local coordinate system. This can be achieved if polynomial is balanced in case all terms cannot be completed. This 'balanced' representation can be achieved with the help of Pascal triangle in case of two-dimensional polynomial. For example, for a polynomial having four terms, the invariance can be obtained if the following expression is selected from the Pascal triangle.

$$u = \alpha_0 + \alpha_1 x + \alpha_2 y + \alpha_3 xy \tag{2.3.2}$$

The geometric invariance can be ensured by the selection of the corresponding order of terms on either side of the axis of symmetry.

$$
\begin{array}{ccccccccc}
 & & & & 1 & & & & \\
 & & & x & & y & & & \\
 & & x^2 & & xy & & y^2 & & \\
 & x^3 & & x^2y & & xy^2 & & y^3 & \\
x^4 & & x^3y & & x^2y^2 & & xy^3 & & y^4
\end{array}
$$

Fig. 2.3.1 Pascal'sTriangle

CST element

The triangular elements with different numbers of nodes are used for solving two dimensional solid members. The linear triangular element was the first type of element developed for the finite element analysis of 2D solids. However, it is observed that the linear triangular element is less accurate compared to linear quadrilateral elements. But the triangular element is still a very useful element for its adaptivity to complex geometry. These are used if the geometry of the 2D model is complex in nature. Constant strain triangle (CST) is the simplest element to develop mathematically. In CST, strain inside the element has no variation (Ref. module 3, lecture 2) and hence element size should be small enough to obtain accurate results. As indicated earlier, the displacement is expressed in two orthogonal directions in case of 2D solid elements. Thus the displacement field can be written as

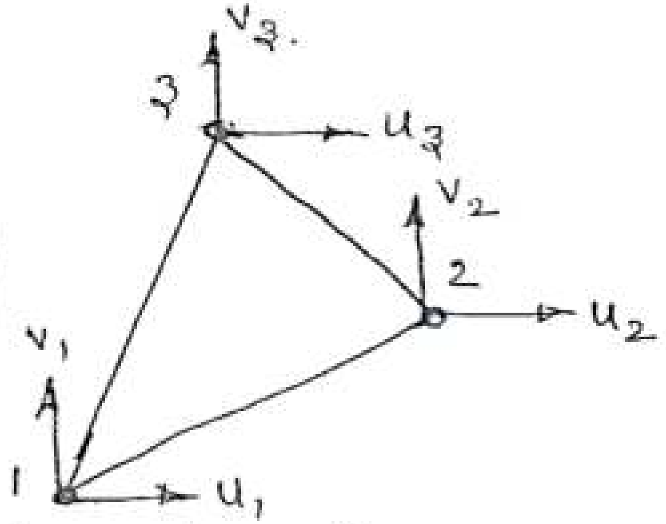
$$\{d\} = \begin{Bmatrix} u \\ v \end{Bmatrix} \tag{5.1.1}$$

Here, u and v are the displacements parallel to x and y directions respectively.

Constant Strain triangular Element (CST). (4)

A three noded triangular element is known as constant strain triangular element. It has six unknown displacement degrees of freedom ($u_1, v_1, u_2 v_2, u_3 v_3$). The element is called CST because it has constant strain throughout it.

$$\text{Displacements } [u] = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}$$



Since CST element has two degrees of freedom at each node (u, v) the total degrees of freedom is 6. Hence it has 6 generalized co-ordinates is 6. . Hence it has 6 generalized co-ordinates

$$u = a_1 + a_2 x + a_3 y$$

$$v = a_4 + a_5 x + a_6 y.$$

where a_1, a_2, a_3, a_4, a_5 and a_6 are global or generalized co-ordinates.

$$u_1 = a_1 + a_2 x_1 + a_3 y_1$$

$$u_2 = a_1 + a_2 x_2 + a_3 y_2$$

$$u_3 = a_1 + a_2 x_3 + a_3 y_3.$$

Because displacement functions are linear in in x and y, all lines in the element including its sides remain straight as the element deforms

The strain displacement relations ~~are~~ in 2D are (5

$$\varepsilon_x = \frac{\partial u}{\partial x}, \quad \varepsilon_y = \frac{\partial v}{\partial y} \quad \text{and} \quad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$$

(shear strain)

$$\therefore \quad \varepsilon_x = \frac{\partial}{\partial x}(a_1 + a_2 x + a_3 y) = a_2$$

$$\varepsilon_y = \frac{\partial}{\partial y}(a_4 + a_5 x + a_6 y) = a_6$$

$$\gamma_{xy} = \frac{\partial}{\partial y}(a_1 + a_2 x + a_3 y) + \frac{\partial}{\partial x}(a_4 + a_5 x + a_6 y)$$

Since strain is constant

$\gamma_{xy} = a_3 + a_5$. the element is called CST element

Isoparametric Element

If the shape function used for defining the geometry as well as displacement are the same, then the element is called isoparametric element. Equal number of nodes are used for defining the geometry as well as displacement is isoparametric element.

$$u = [N][\delta_e]_e \quad \text{— displacement field}$$

$$\left.\begin{aligned} x &= [N][x]_e \\ y &= [N][y]_e \end{aligned}\right\} \text{— geometry}$$

The finite element method is a powerful technique for analysing engineering proble involving complex and irregular geometrics. However, the two and three dimensio elements (triangle, rectangle, brick) discussed in previous chapters cannot be used efficien for irregular geometrics.

Consider a continuum shown in Fig.5.1(a) and it is discretized by using triangul elements which is shown in Fig.5.1(b).

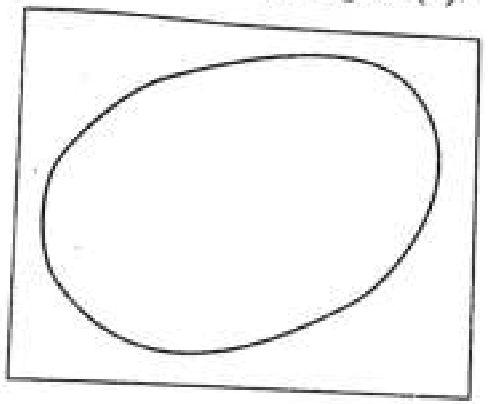


Fig. 5.1. (a) Continuum

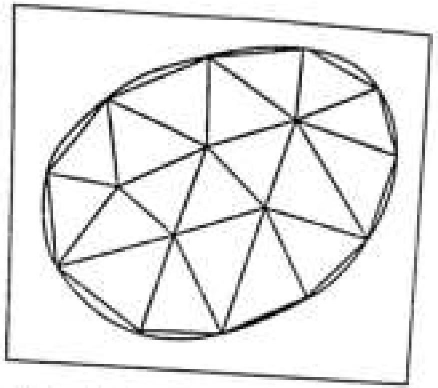
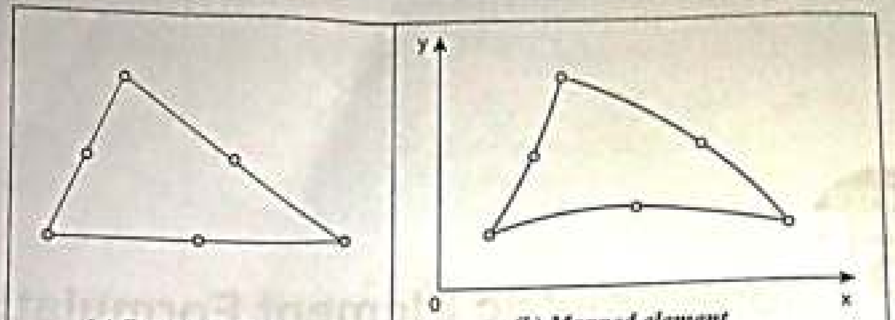


Fig. 5.1. (b) Continuum is discretized by triangular elements

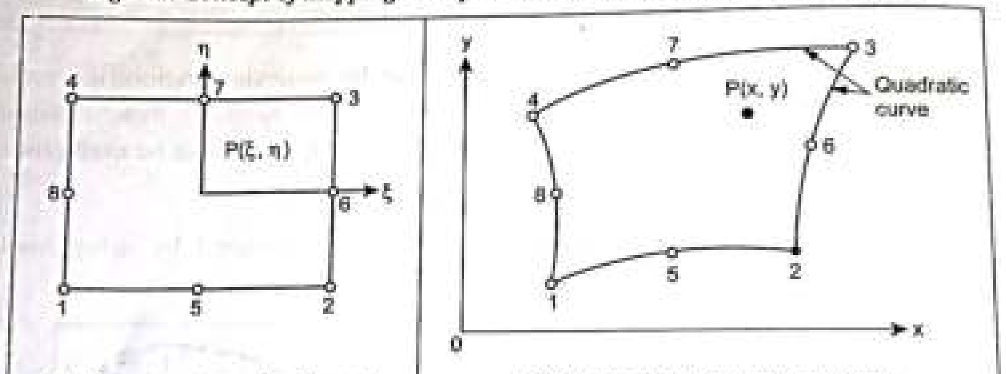
It is difficult to represent the curved boundaries by straight edges elements. A large number of elements may be used to obtain reasonable resemblance between original body and the assemblage. In order to overcome this drawback, isoparametric elements are used. *i.e.,* for problems involving curved boundaries, a family of elements known as "isoparametric elements" can be used.

The isoparametric concept was first brought out by ***Taig*** and latter on generalized by ***B.M. Irons*** for mapping the curved boundaries. They brought out the concept of mapping for regular triangular, rectangular elements and brick elements from natural co-ordinate system to global cartesian system as shown in Fig.5.2, 5.3 & 5.4.



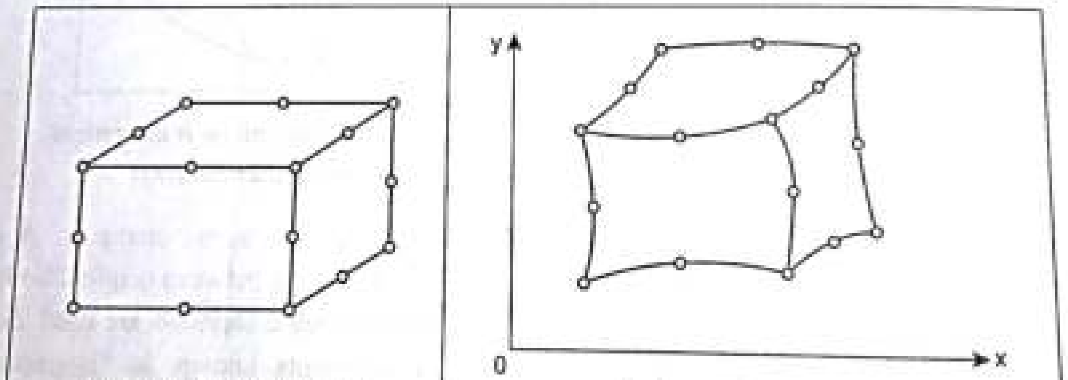
(a) Parent element (b) Mapped element

Fig. 5.2. Concept of mapping in isoparametric elements (Triangular element)



(a) Parent rectangular element (b) Mapped rectangular element (Quadrilateral element)

Fig. 5.3.



(a) Parent brick element (b) Mapped brick element

Isoparametric Element

We know that, shape functions are used for defining the geometry and displacements of the element. Consider a element shown in Fig.5.5.

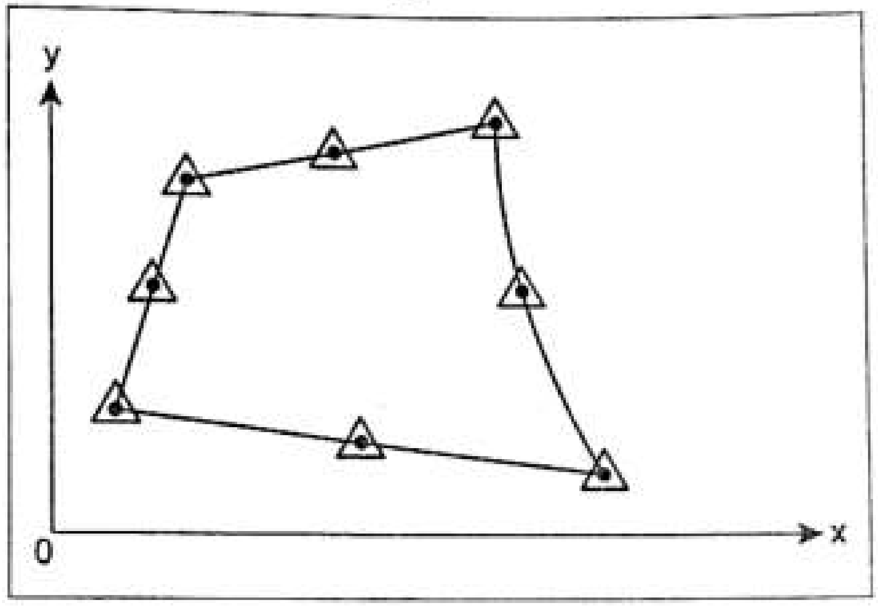


Fig. 5.5.

- Nodes used for defining geometry.

Δ Nodes used for defining displacements.

In this element, all the eight nodes are used in defining geometry as well as displacements.

If the number of nodes used for defining the geometry is same as number of nodes used for defining the displacements, then, it is known as isoparametric element.

uperparametric Element

Consider a element shown in Fig.5.6.

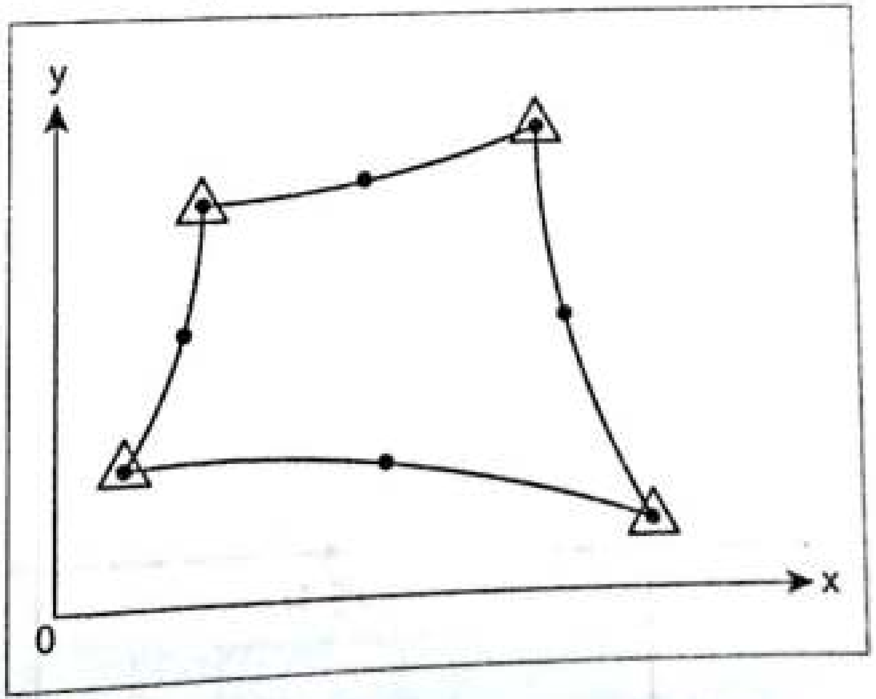


Fig. 5.6.

- ● Nodes used for defining geometry.
- Δ Nodes used for defining displacements.

In this element, eight nodes are used to define the geometry and four nodes are used to ine the displacements.

If the number of nodes used for defining the geometry is more than number of nodes used for defining the displacements, then, it is known as superparametric element.

Subparametric Element

Consider a element shown in Fig.5.7.

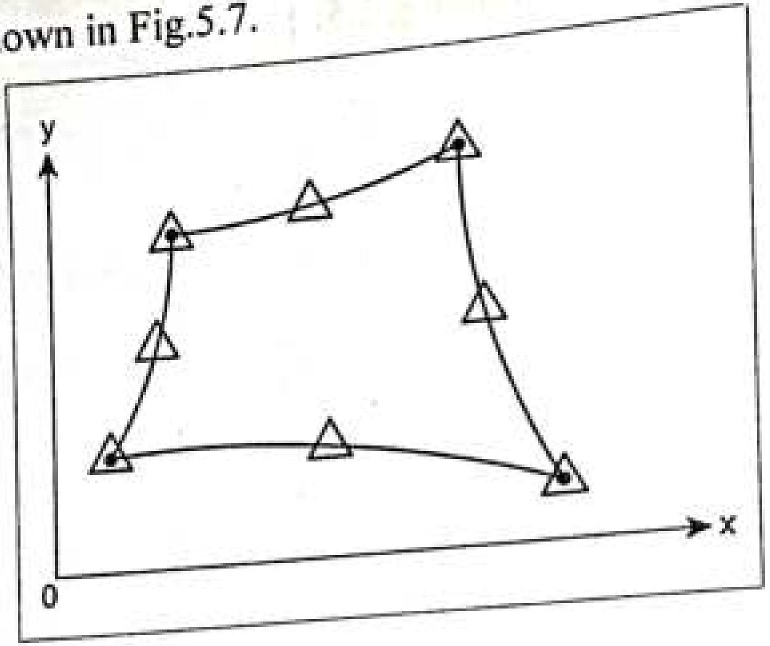


Fig. 5.7.

- Nodes used for defining geometry.

Δ Nodes used for defining displacements.

In this element, four nodes are used to define the geometry and eight nodes are used to define the displacements.

If the number of nodes used for defining the geometry is less than number of nodes used for defining the displacements, then it is known as subparametric element.

Gauss Quadrature Method

This method can be applied only when the integrand $f(x)$ is known explicitly so that the function can be evaluated at any desired value of x.

(i) Two Point Quadrature formula

Consider the integral $I = \int_{-1}^{1} f(x)\,dx$

Let $I = a_1 f(x_1) + a_2 f(x_2)$

where the co-effts a_1, a_2 and the function arguments x_1, x_2 are to be determined.

On deriving the formula we get.

$$I = \int_{-1}^{1} f(x)\,dx = f\left(\frac{1}{\sqrt{3}}\right) + f\left(-\frac{1}{\sqrt{3}}\right)$$

-1

or $\boxed{I = f(0.5773) + f(-0.5773)}$

It is known as Gauss two point quadrature formula. It gives an approximate value of the integral if $f(x)$ is any polynomial of degree 3 or less.

In deriving Gauss two point formula (11)
we have assumed that the integration
is from -1 to 1. If it is from a to b
then we shall apply a suitable change
of variable to bring the ~~to~~ in the range of -1 to 1

We replace the given variable x by
another variable t which are related by
the following formula.

$$x = \frac{(b-a)t + (b+a)}{2}$$

when $x = a$ $t = -1$

$x = b$ $t = 1$

and $dx = \left(\frac{b-a}{2}\right) dt$.

$$\therefore \int_a^b f(x)dx = \left(\frac{b-a}{2}\right)\int_{-1}^{1} f\left[\frac{(b-a)t+(b+a)}{2}\right]dt$$

Gauss three point quadrature formula

Gauss three point formula for $I = \int_{-1}^{1} f(x)dx$ is given by.

$$\boxed{\begin{aligned} I = {} & 0.55555555\, f(-0.77459667) + 0.88888889\, f(0) + \\ & 0.55555555\, f(0.77459667) \end{aligned}}$$

Find $\int_0^{\pi/2} \sin x \, dx$ by two and three point Gauss quadrature method.

Here $f(x) = \sin x$ $a = 0$ and $b = \pi/2$

for changing the integration limit to -1 - 1

$$x = \frac{(b-a)t + (b+a)}{2} = \frac{\pi}{4}(t+1)$$

$$\therefore I = \int_0^{\pi/2} \sin(x)\,dx = \frac{\pi}{4}\int_{-1}^{1} f\left[\frac{\pi}{4}(t+1)\right] dt.$$

$$= \frac{\pi}{4}\int_{-1}^{1} \sin\left[\frac{\pi}{4}(t+1)\right] dt. = \int_{-1}^{1} g(t)\,dt$$

where $g(t) = \frac{\pi}{4}\sin\left[\frac{\pi}{4}(t+1)\right]$

by two point method.

$$I = \int_{-1}^{1} g(t)\,dt = g(0.5773) + g(-0.5773)$$

$$\therefore I = \frac{\pi}{4} \sin\left[\frac{1.5773\pi}{4}\right] + \frac{\pi}{4} \sin\left[\frac{0.4227\pi}{4}\right]$$

$$= \frac{\pi}{4}\left[1.2712\right] = \underline{\underline{0.9985}}$$

by three point method.

$$I = 0.5555\,g(-0.7746) + 0.8889\,g(0) + 0.5555\,g(0.7746)$$

$$= 0.076841659 + 0.49265266 + 0.42951279$$

$$= \underline{\underline{1.000008116}}$$

Which is very close ~~to~~ the actual value 1