

OBJECTIVES:

- To provide an overview of how computers are being used in mechanical component design

UNIT I FUNDAMENTALS OF COMPUTER GRAPHICS 9

Product cycle- Design process- sequential and concurrent engineering- Computer aided design – CAD system architecture- Computer graphics – co-ordinate systems- 2D and 3D transformations-homogeneous coordinates - Line drawing -Clipping- viewing transformation

UNIT II GEOMETRIC MODELING 9

Representation of curves- Hermite curve- Bezier curve- B-spline curves -rational curves- Techniques for surface modeling – surface patch- Coons and bicubic patches- Bezier and B- spline surfaces. Solid modeling techniques- CSG and B-rep

UNIT III VISUAL REALISM 9

Hidden – Line-Surface-Solid removal algorithms – shading – colouring – computer animation.

UNIT IV ASSEMBLY OF PARTS 9

Assembly modelling – interferences of positions and orientation – tolerance analysis- massproperty calculations – mechanism simulation and interference checking.

UNIT V CAD STANDARDS 9

Standards for computer graphics- Graphical Kernel System (GKS) - standards for exchangeimages-Open Graphics Library (OpenGL) - Data exchange standards - IGES, STEP, CALSetc. - Communication standards.

TOTAL : 45 PERIODS

OUTCOMES:

- Upon completion of this course, the students can able to use computer and CAD software's for modeling of mechanical components.

TEXT BOOKS:

- Ibrahim Zeid “Mastering CAD CAM” Tata McGraw-Hill Publishing Co.2007

REFERENCES:

- Chris McMahon and Jimmie Browne “CAD/CAM Principles”, "Practice and Manufacturing management “ Second Edition, Pearson Education, 1999.
- William M Neumann and Robert F.Sproul “Principles of Computer Graphics”, McGraw Hill Book Co. Singapore, 1989.
- Donald Hearn and M. Pauline Baker “Computer Graphics””. Prentice Hall, Inc, 1992.
- Foley, Wan Dam, Feiner and Hughes - "Computer graphics principles & practice" Pearson Education - 2003.

Need for this Subject

- ✓ To understand the fundamental concepts on which the CAD/CAM systems are built; Users become very efficient.

INTRODUCTION

Design

- ✓ A plan or drawing produced to show the look and function or workings of a building, garment, or other object before it is made.
- ✓ A specification of an object, manifested by an agent, intended to accomplish goals, in a particular environment, using a set of primitive components, satisfying a set of requirements, subject to constraints.

Computer Aided Design (CAD)

- ✓ CAD, or computer-aided design and drafting (CADD), is the use of computer technology for design and design documentation.
- ✓ CAD replaces manual drafting with an automated process.
- ✓ Computer-aided drafting (CAD) is the use of computer systems to aid in the creation, modification, analysis, or optimization of a design.
- ✓ CAD/CAM has been utilized in engineering practice in many ways including drafting, design, simulation, analysis and manufacturing.
- ✓ CAD is used to increase the productivity of the designer, improve the quality of design, improve communications through documentation, and to create a database for manufacturing.
- ✓ CAD/CAM includes geometric modeling, computer graphics, design applications, and manufacturing applications.

Need for CAD!!

- ✓ This century is known for rapid development particularly in the field of computers.
- ✓ Presently industries cannot survive worldwide competition unless they introduce new products with better quality (Q), at lower cost (C), and with shorter lead time (D).
- ✓ Accordingly, many industries have tried to use the computer's huge memory capacity, fast processing speed, and user-friendly interactive graphic capabilities to automate and tie together the cumbersome production tasks, thus reducing the time and cost of product development and production.
- ✓ Computer-aided design (CAD), computer-aided manufacturing (CAM), and computer-aided engineering (CAE) are the technologies used for this purpose during the product cycle.

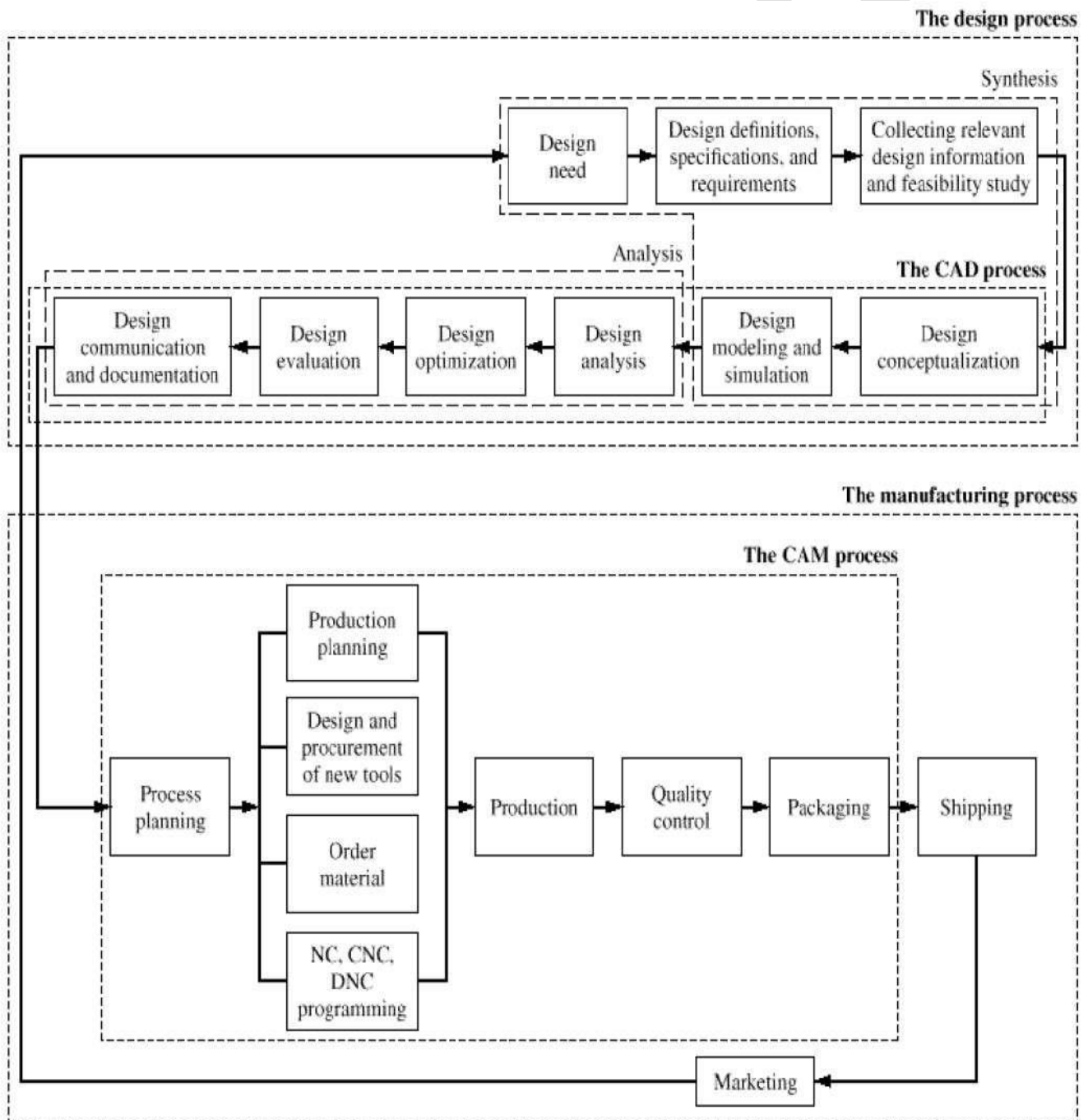
Product cycle- Design process- sequential and concurrent engineering- Computer aided design – CAD system architecture- Computer graphics – co-ordinate systems- 2D and 3D transformations-homogeneous coordinates - Line drawing -Clipping- viewing transformation

Product cycle

What is product cycle?

- ✓ In the design and manufacture of a product various activities and functions must be accomplished. These activities and functions are referred to as the "Product Cycle".

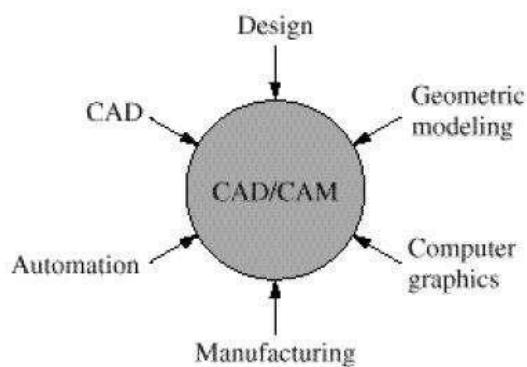
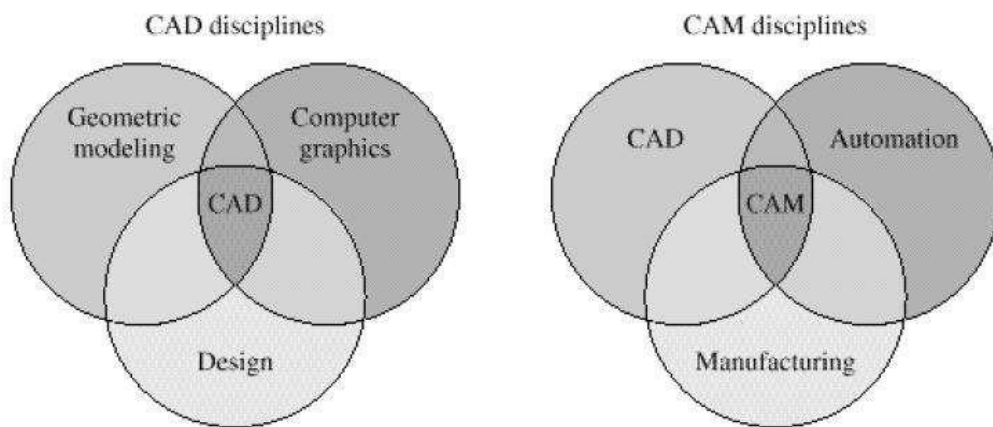
A typical product cycle



- ✓ Two main processes: Design process and the manufacturing process.
- ✓ The activities involved in the design process can be classified largely as two types: synthesis and analysis.
- ✓ The philosophy, functionality, and uniqueness of the product are all determined during synthesis.
- ✓ During synthesis, a design takes the form of sketches and layout drawings that show the relationship among the various product parts.
- ✓ Most of the information generated and handled in the synthesis sub process is qualitative and consequently it is hard to capture in a computer system.
- ✓ The analysis sub-process begins with an attempt to put the conceptual design into the context of engineering sciences to evaluate the performance of the expected product.
- ✓ This requires design modeling and simulation. An important aspect of analysis is the "What if" questions that help us to eliminate multiple design choices and find the best solution to each design problem.
- ✓ Bodies with symmetries in their geometry and loading are usually analyzed by considering a portion of the model.
- ✓ Example: stress analysis to verify the strength of the design, interference checking to detect collision between components while they are moving in an assembly, and kinematic analysis to check whether the machine to be used will provide the required motions.
- ✓ The quality of the results obtained from these activities is directly related to and limited by the quality of the analysis model chosen.
- ✓ Prototypes may be built for the design evaluation. Prototypes can be constructed for the given design by using software packages (CAM). The new technology called rapid prototyping is become popular now for constructing prototypes. This technology enables the construction of a prototype by depositing layers from the bottom to the top.
- ✓ The outcome of analysis is the design documentation in the form of engineering drawings (blueprints).
- ✓ The manufacturing process begins with process planning, using the drawings from the design process, and it ends with the actual products.
- ✓ Process planning is a function that establishes which processes—and the proper parameters for the processes—are to be used. (backbone)

- ✓ It also selects the most efficient sequence for the production of the product. The outcome of the process planning is a production plan, tools procurement, materials order, and machine programming. Other special requirements, such as design of jigs and fixtures, are also planned.
- ✓ The relationship of process planning to the manufacturing process is analogous to that of synthesis to the design process: It involves considerable human experience and qualitative decisions. This description implies that it would be difficult to computerize process planning.
- ✓ Once process planning has been completed, the actual product is produced and inspected against quality requirements. Parts that pass the quality control inspection are assembled, functionally tested, packaged, labeled, and shipped to customers.
- ✓ Market feedback is usually incorporated into the design process. With this feedback, a closed-loop product cycle results.

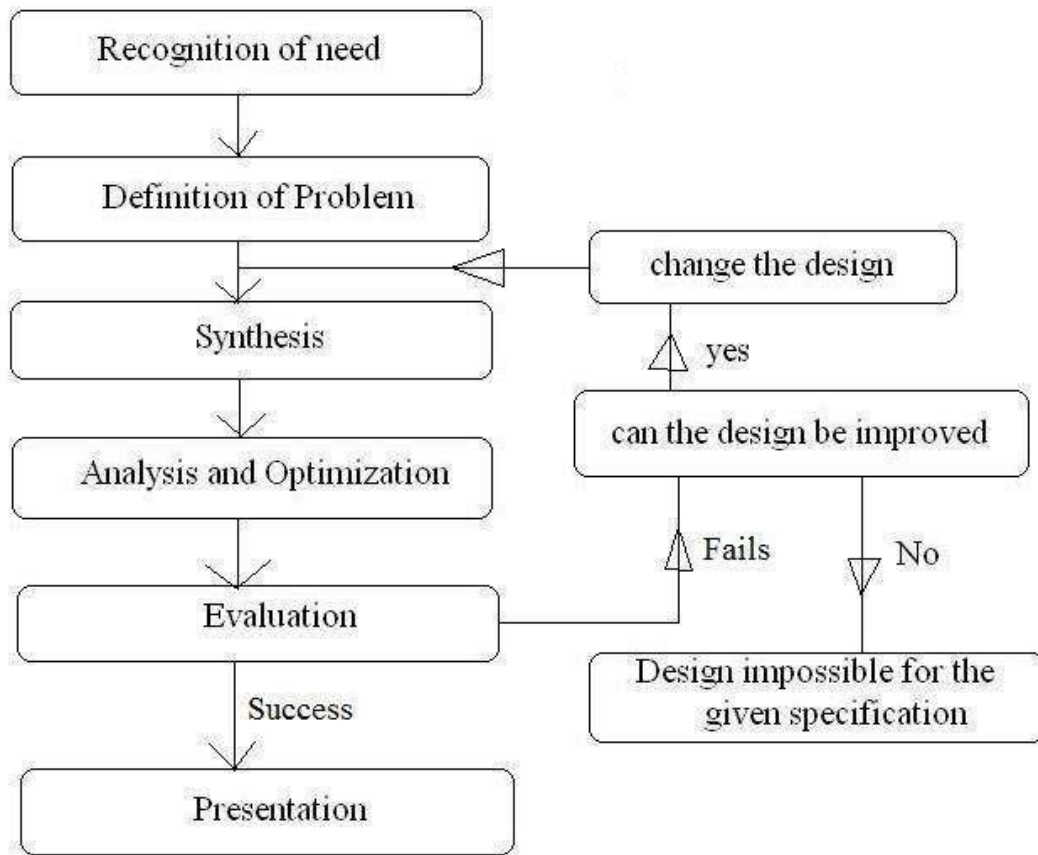
CAD and CAM Disciplines



Design Process

- ✓ In general it is agreed that the design progresses in a step by step manner from identification of need for the problem, a search for solutions and development of chosen solution to manufacture, test and use.
- ✓ These descriptions of design are often called models of the design process. Various models are namely
- ✓ Shigley, Pahl and Beitz, Ohsuga and Earle.

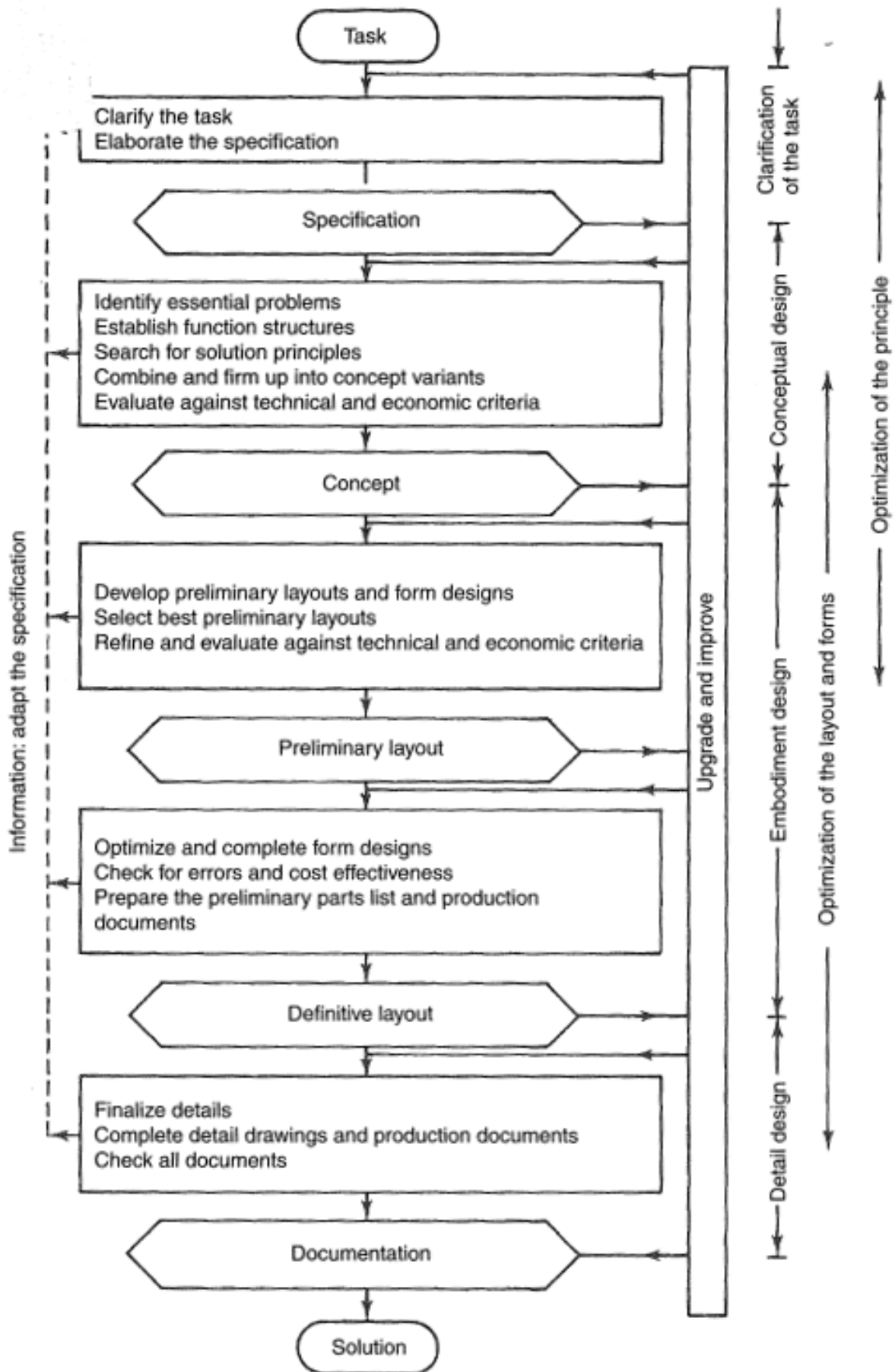
Shigley's Design Process



Pahl and Beitz model of the design process

- ✓ Clarification of the task
- ✓ Conceptual design
- ✓ Embodiment design
- ✓ Detail design
- ✓

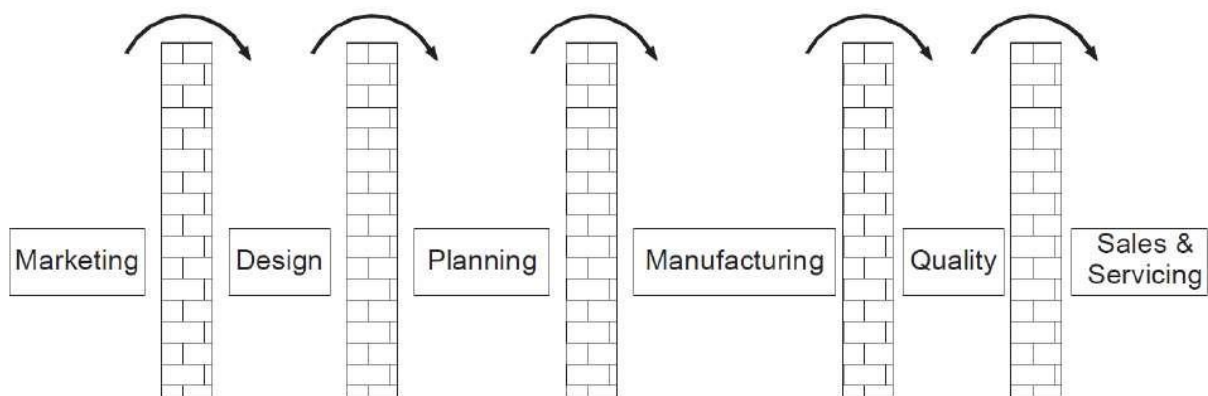
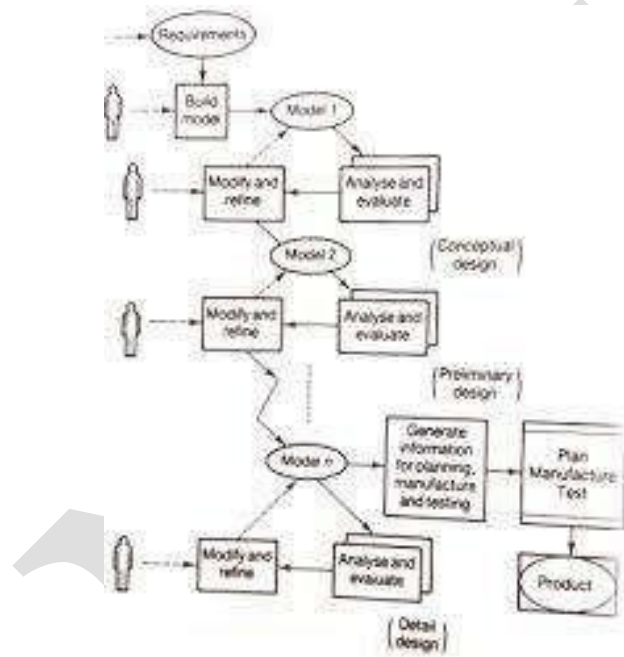
Pahl and Beitz model of the design process



Ohsuga and Earle model of the design process All the above models of the design process follow a traditional view which consists of various steps. However, the pressure to reduce product design and development time-scales (Lead time) demands for simultaneous/ concurrent/ parallel engineering. The design, development, analysis and the preparation of manufacturing information are done in parallel then it is termed as Simultaneous /concurrent engineering.

Sequential Engineering

- ✓ In the traditional product development process the product design, planning, manufacturing, supporting quality and testing, marketing activities are carried out one after another. All these phases are carried out sequentially hence it is called as sequential engineering.



Across the Wall Approach in Sequential Engineering

- ✓ This situation assumes that there is no interaction among the major departments involved in product manufacturing during the initial development process.
- ✓ Often the need for engineering changes is discovered during planning or manufacturing or assembly.
- ✓ In each phase/activities there is no interaction between them. The other name for sequential approach is ‘over wall’ approach.
- ✓ There is no interaction i.e. there is a communication barrier between each department.

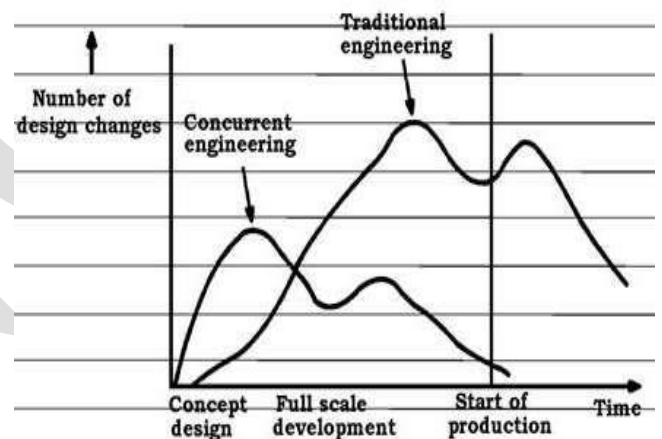
Limitations

- ✓ Decisions are taken by individuals.
- ✓ Product modifications/changes will be slow.
- ✓ Each activity is carried out sequentially.
- ✓ Because of the above reasons, time taken for product development is more. Lead to inevitable conflicts; each department sticking to their own decisions and may often require intervention of senior management to resolve conflicts or differences in opinion.

✓ Long lead time

If any modification to be made on the product by stream department, it has to be fed back and this often involves in additional expenditure and also results in unnecessary delay in product cycle.

✓ Lower quality



Concurrent Engineering

- ✓ Global competition pressurizes the firms to produce products with high performance, reliable, low cost with less lead time.
- ✓ To achieve this, in the product planning stage itself a co-operate work between design and manufacturing and other related activities has to be made.
- ✓ During product development process, bringing together a design team with the appropriate combination of specialized expertise is known as “Concurrent Engineering” or “Simultaneous Engineering”.
- ✓ A typical team consists of experts from

- 1) Material Manufacturing process
- 2) Assembly
- 3) Maintenance 4) Marketing

- 5) Service personnel 6) End users and
- 7) Specialists in the product related field.

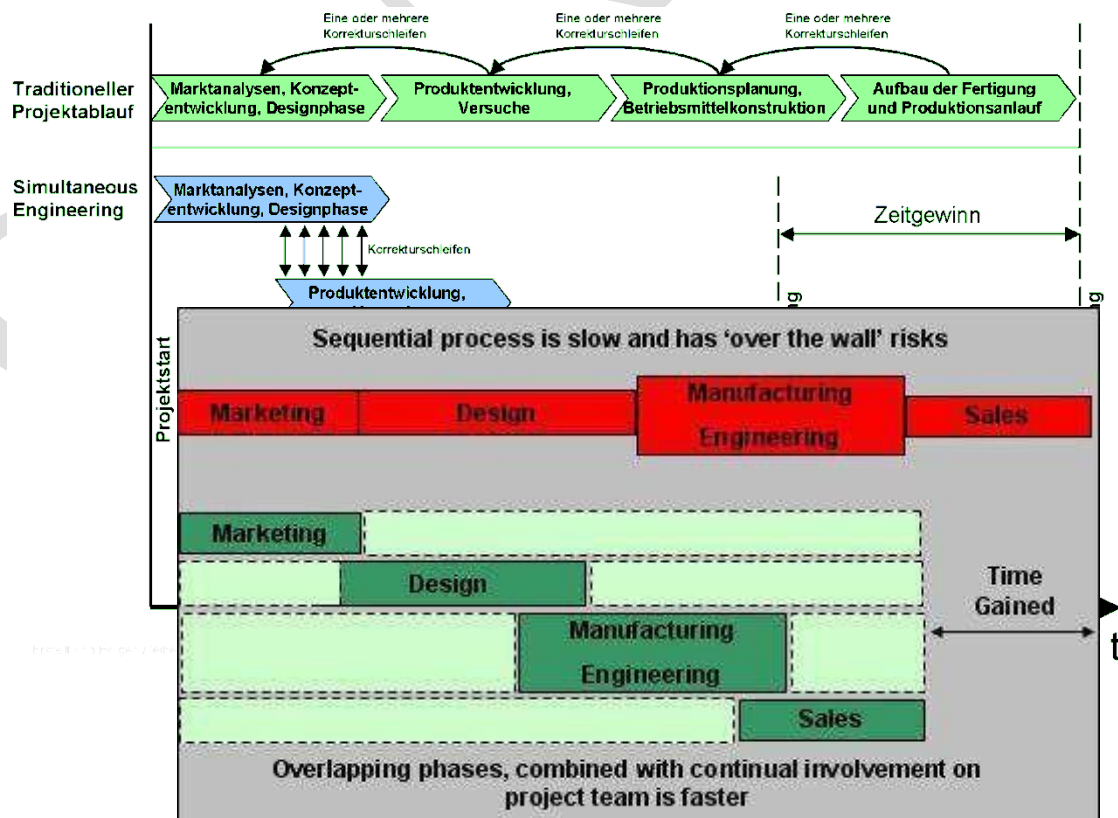
- ✓ The product development is the responsibility of this multi-disciplinary team which follows the product from inception to complete production.
- ✓ There are circumstances in which concurrent engineering is not necessarily the best approach, especially where there is very high uncertainty in the product development process.

Characteristics of Concurrent Engineering:

- ✓ Product responsibilities lie on team of multi-disciplinary group. Integration of design, process planning and production will be achieved.
- ✓ Product lead time will be less, because cross-functional activities are started simultaneously.
- ✓ Most of the modification charges are carried in the planning stage itself. Frequent review of design and development process.
- ✓ Rapid prototyping
- ✓ More attention will be given to satisfy the customer needs and to include newer technologies in product development.

Advantages of Con-Current Engineering:

- ✓ The design decisions are taken by a team of multi-disciplinary experts. Changes and modifications on the product design will be faster.
- ✓ Shorter lead times because all the activities related to product development are carried out simultaneously.
- ✓ Higher quality.



Software Technologies available for Concurrent Engineering

Design	Planning/Manufacture	Visualization/Simulation
Solid modeling	Process planning	Factory Simulation
Surface modeling	ERP	Simulation software for-
Assembly modeling	Generative machining	• Welding
Sheet metal design	Shop floor data collection	• Casting
Drafting	Human machine interface	• Forming
Tolerance analysis	Job tracking	• Forging
Mechanism design	Work in process inventory	• Plastic injection molding-
Finite Element analysis	tracking	• Robot operation
Harness design	PDM, VPDM and PLMSoftware for-	• Machining etc.
Mold design	• EDM	• Rapid Prototyping
Mold flow analysis	• Wire EDM	
Dynamic analysis	• Press brake	
Thermal analysis	• Grinding	
Composites design	• Turret Punch Press	
Piping design		
Optimization		
Tool design		
Standard part libraries		

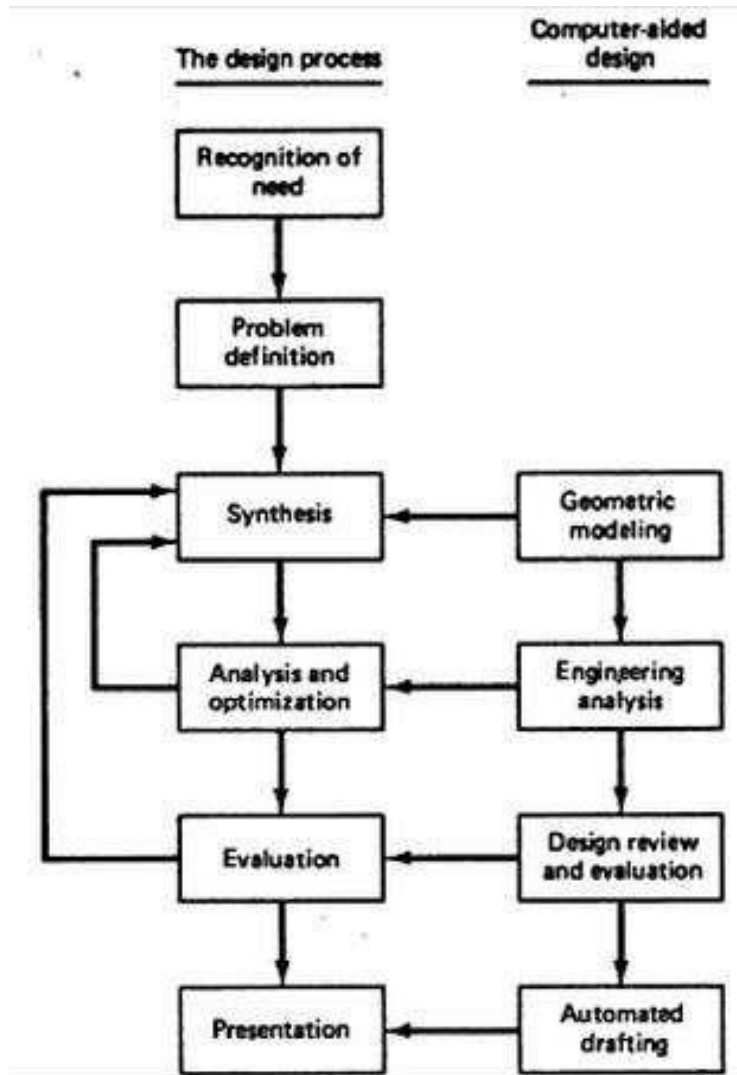
Soft and Hard Prototyping

Boeing Commercial Airplane Group – CATIA- Solid model – Interference problems – visualization benefits – subcontractors- Quotations are more accurate

Examples of Concurrent Engineering (CE)

- ✓ Hewlett Packard – Yokogawa HP – Cycle time decreased by 35 %, Manufacturing cost 42%, Inventory 64%, Field failure rates 60 %
- ✓ Neon Car in USA - Each team took 1 year – 4 months before launch team Scooty by TVS Motors Ltd. –
- ✓ Cross functional teams

COMPUTER AIDED DESIGN (CAD)



The CAD process

- ✓ The conventional design process has been accomplished on drawing boards with design being documented in the form of a detailed engineering drawing. This process is iterative in nature and is time consuming. The computer can be beneficially be used in the design process.
- ✓ The various stages related tasks which are performed by a modern computer aided design system can be grouped in four functional areas.
 1. Geometric Modeling
 2. Engineering analysis
 3. Design review and evaluation
 4. Automated drafting.

Geometric Modeling

- ✓ The geometric modeling is concerned with computer compatible mathematical description of geometry of an object. The mathematical description should be such that
 - The image of the object can be displayed and manipulated in the computer terminal,
 - Modification on the geometry of the object can be done easily,
 - It can be stored in the computer memory
 - Also can be retrieve back on the computer screen for review, analysis or alteration.

Geometric modeling is classified into

- ✓ Wireframe modeling Solid modeling
- ✓ Surface modeling

Engineering analysis

- ✓ The computer can be used to aid the analysis work such as stress-strain analysis, heat transfer analysis, etc. the analysis can be done by using specific program generated for it or by using general purpose software commercially available in the market.
- ✓ The geometric models generated can be used for the analysis by properly interfacing the modeling software with the analysis software.
- ✓ Two types of engineering analysis are
 - Analysis for mass properties
 - Finite element analysis (FEA)

Design review and evaluation

- ✓ The accuracy of the design can be checked and rectified if required in the screen itself.
- ✓ Layering feature available in software are very useful for design review purpose.
- ✓ Similarly, using the layer procedure, every stage of production can be checked.
- ✓ Suppose a new mechanism is to be designed, the same mechanism can be simulated in the computer.
- ✓ By animation, the working of the mechanism can be checked.
- ✓ These will relieve the designer from tedious conventional method of mechanism checking.
- ✓ Another advantage of animating the complete assembly of product is that whether any component fouls the other components in its working.

Automated drafting

- ✓ Automated drafting is the process of creating hard copies of design drawing. The important features of drafting software's are automated dimensioning, scaling of the drawing and capable of generating sectional views.

- ✓ Also the enlargement of minute part details and ability to generate different views of the object like orthographic, oblique, isometric and perspective views are possible.
- ✓ Thus, CAD systems can increase productivity on drafting.

Benefits of CAD

- ✓ Efficiency, effectiveness and creativity of the designer are drastically improved.
- ✓ Faster, Consistent and more accurate.
- ✓ Easy modification (copy) and improvement (Edit).
- ✓ Repeating the design drawing is not needed when modifying. Manipulation of various dimensions, attributes is easy.
- ✓ Parametric and posses parent-child relationship.
- ✓ Inspecting tolerance and interface is easy.
- ✓ Use of standard components from part library makes fast modeling. Excellent graphical representation.
- ✓ Co-ordination among the groups and sharing the design data is possible. Exchange of e-drawing and storage of several data are easily possible.
- ✓ Graphical Simulation and animation studies the real-time behavior. 3D visualization of model in several orientations eliminates prototype.
- ✓ Documentation at various design phases is efficient, easier, flexible and economical.
- ✓ Linkage to Manufacturing to carry out the production (NC/CNC programming).

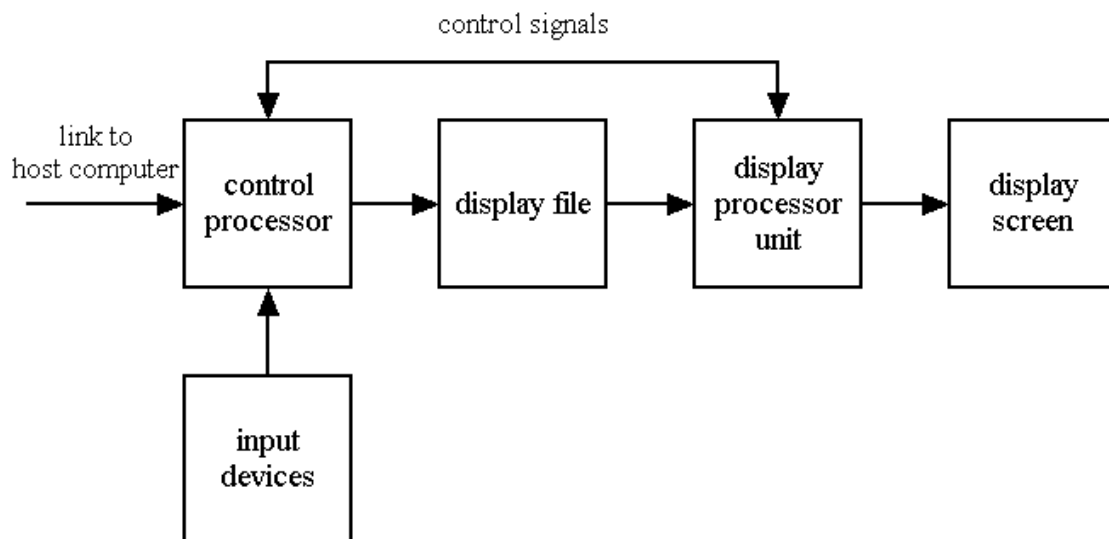
Engineering applications of CAD

This list includes following applications of CAD but not limited to: Structural design of

- ✓ Aircraft
- ✓ Aircraft simulation Real time simulation Automobile industries Architectural design
- ✓ Pipe routing and plan layout design Electronic industries
- ✓ Dynamic analysis of mechanical systems Kinematic analysis
- ✓ Mesh data preparation for finite element analysis

COMPUTER GRAPHICS

- ✓ Computer Graphics involves creation, display, manipulation and storage of pictures and experimental data for proper visualization using a computer.
- ✓ Typically a graphics system comprises of a host computer which must have a support of a fast processor, a large memory and frame buffer along with a few other crucial components.
- ✓ The first of them is the display devices. Color monitors are one example of such display device. There are other examples of output devices like LCD panels, laser printers, color printers, plotters etc.
- ✓ Set of input devices are also needed. Typical examples are the mouse, keyboard, joystick, touch screen, trackball etc.
- ✓ Through these input devices it is possible to provide input to the computer and display device is an output device which shows the image.

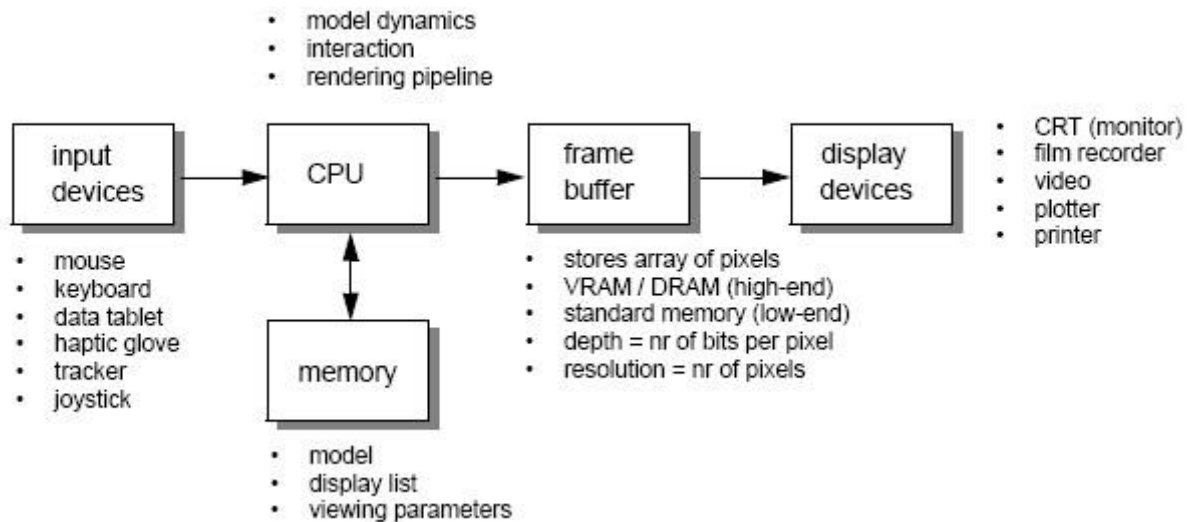


Block diagram of basic computer graphics layout

Few examples of application areas of Computer Graphics:

- ✓ The first and most important of them is the GUI as it is called. It has various components.
- ✓ A graphical interface is basically a piece of interface or a program which sits between the user and the graphics application program.
- ✓ It helps the graphics system to interact with the user both in terms of input and output. Let us see few examples.

- ✓ Typical components which are used in a Graphical User Interface are menus, icons, cursors, dialog boxes and scrollbars.

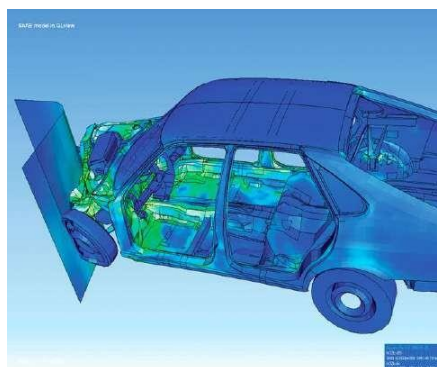
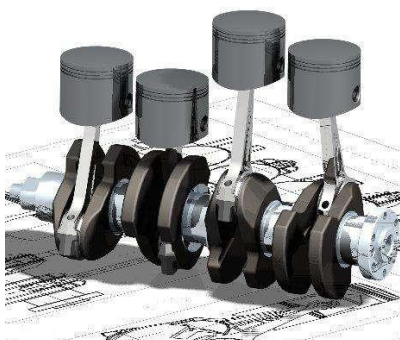


Overview of graphic design process

- ✓ Grids are used in two dimensional graphics packages to align the objects along a set of specific coordinates or positions. It can be switched on and off and displayed on the screen.
- ✓ Sketching is an example which is used to draw lines, arcs, poly lines and various other objects.
- ✓ The most difficult part of the GUI is three dimensional interfaces which is normally available at the bottom of screen.
- ✓ It is easy to interact and handle with two dimensional objects but for interacting with the three dimensional objects three dimensional interface is needed to pick up one of the 3D objects from a two dimensional screen.
- ✓ Essentially the computer monitor is just a two dimensional ray of pixels where the entire picture is projected and the picture could represent a three dimensional scene.
- ✓ Special facilities for 3D interface to handle or manipulate three dimensional objects are needed.

Other typical application areas of Computer Graphics

- ✓ A typical example of CAD/CAM is a vast scope in many fields of engineering and technology.
- ✓ VLSI: Very Large Scale Integration constructions in the area of architecture and civil engineering and



circuits.

3D graphics of camshaft with piston

3D structure of car when crash testing

SANVCET

- ✓ Designs in the area of electrical engineering: These are the areas of CAD/CAM designs where we actually manufacture a particular chip in VLSI or a printed circuit board.
- ✓ Another example, in a circuit or architecture, a bridge or road or it could be a building or a complex, a multi-storey complex.
- ✓ Simulation studies and simulators: Simulation studies and simulators include areas such as computation, fluid dynamics, heat and mass transfer, various other mechanical and chemical processors.
- ✓ It could include even studies about nuclear explosion and disaster management, damage prediction in a certain area, flight simulators.
- ✓ Flight simulators car racing, docking of a naval ship or a space shuttle. Training pilots and engineers before going on to particular mission simulators can be used to really test the skill of a particular person.
- ✓ It could be even in the field of sports one would like to use simulators to study the reaction time of a particular individual.
- ✓ Process monitoring in chemical and mechanical processors; sensors fitted at various points which monitor the health and the flow of a particular assembly line; online monitoring to find out that the entire team is working true and in a correct form.
- ✓ The sensors are giving feedback simultaneously from different locations and if there is any wrong signal coming out from the sensor you can generate an alarm using a Graphical User Interface and alert the person who is probably monitoring it. So GUI could be used for process management as well.
- ✓ Advertisement, film industry, computer industry.

Characteristics of computer graphics

- ✓ Based on the control the user has over the image
 - Passive computer graphics - the user has no control
 - Interactive graphics – the user may interact with the graphics Based on the way the image is generated
- ✓ image is generated
 - Vector graphics – the image comprises of number of lines (.eps)
 - Raster graphics –manipulation of the color and intensity of points; pixels
- ✓ Based on the space
 - Image-space graphics – image itself is directly manipulated to create a picture.
 - Object-space graphics – Separate model is manipulated.

COORDINATE SYSTEM

Three types of coordinate systems are generally used in CAD/CAM operations Model Coordinate System

✓ (MCS) or Database CS/ World CS / Master CS Working Coordinate System (WCS)

✓ Screen Coordinate System (SCS) or Device CS

Model Coordinate System (MCS)

✓ It is the reference space of the model with respect to which all the model geometrical data is stored.

✓ It is a Cartesian system which forms the default coordinate system used by a particular software program.

✓ The X, Y, and Z axes of the MCS can be displayed on the computer screen. The choice of origin is arbitrary.

✓ The three default sketch planes of a CAD/CAM system define the three planes of MCS, and their intersection point is the MCS origin.

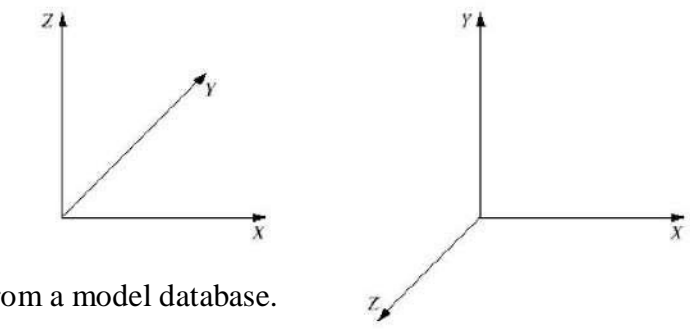
✓ When a CAD designer begins sketching, the origin becomes a corner point of the profile being sketched. The sketch plane defines the orientation of the profile in the model 3D space.

✓ Existing CAD/CAM software uses the MCS as the default WCS.

✓ The MCS is the only coordinate system that the software recognizes when

storing or retrieving graphical information from a model database.

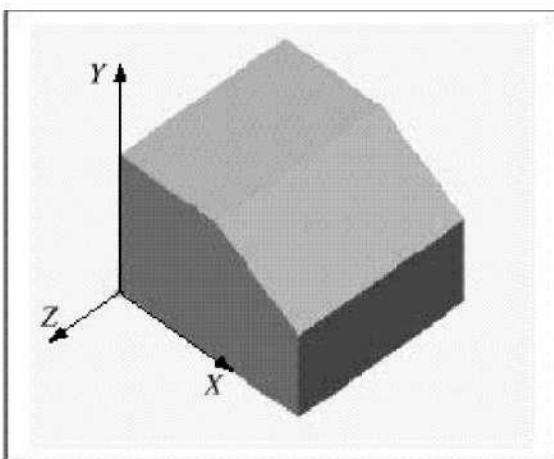
✓ Many existing software package allow the user to input Cartesian and cylindrical corodinales. This input information is transformed to (x, y, z) coordinates relative to the MCS before being stored in the database.



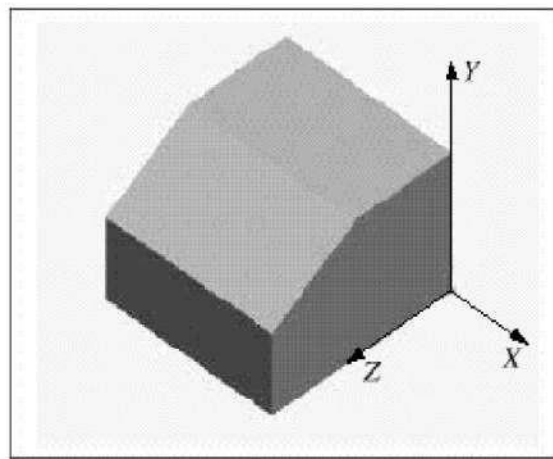
(a) XY plane defines model top view

(b) XY plane defines model front view

Possible orientations of MCS in 3D space.



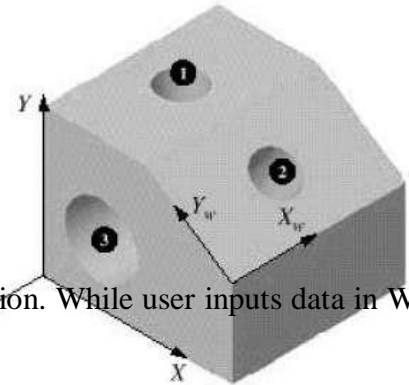
Orientation



Orientation

Working Coordinate System (WCS)

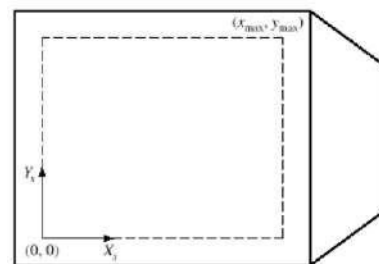
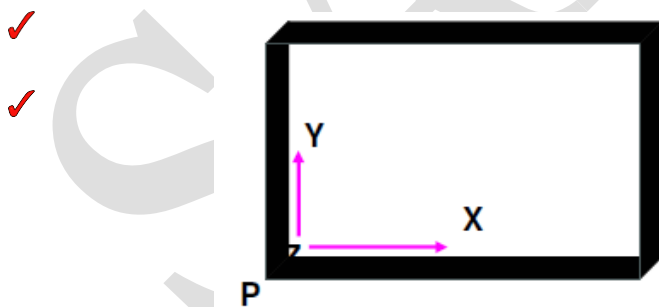
- ✓ This is basically an auxiliary coordinate system used in place of MCS.
- ✓ For convenience while we develop the geometry by data input this kind of coordinate system is useful.
- ✓ It is very useful when a plane (face) in MCS is not aligned (easily defined) along any orthogonal planes.
- ✓ It can be established at any position and orientation in space that the user desires.
- ✓ The user can define a cartesian coordinate system whose XY plane is coincident with the desired plane of construction. That new system is called as WCS.
- ✓ It is a user defined system that facilitates the geometrical construction. While user inputs data in WCS the software transforms it to MCS before storing the data.
- ✓ In Unigraphics: WCS; In Pro/E and Solidworks: Sketch/Construction plane.
- ✓ A WCS requires three noncollinear points to define its XY plane. The first defines the origin, the first and the second define the x-axis, and so.
- ✓ There is only one active WCS at any one time. If the user defines multiple WCSs in one session, the software recognizes only the last one.



Screen Coordinate System (SCS)

- ✓ In contrast to MCS and WCS, Screen Coordinate System is a two-dimensional device-independent system
- ✓ whose origin is usually located at the lower left corner of the graphic display (display screen).

The physical dimensions of the device screen and the type of device determine the range of the SCS.



A 1024 x 1024 display has an SCS with a range of (0,0) to (1024,1024). The SCS is important for display, screen input and digitizing tasks.

- ✓ A transformation operation from MCS coordinates to SCS coordinates is performed by the software
- ✓ before displaying the model views and graphics. For a geometric model, there is a data structure to
- ✓ store its geometric data (relative to MCS), and a display file to store its display data (relative to SCS).
- ✓ **Coordinate system for conventional practice** Cartesian Coordinates Polar Coordinates

TWO-DIMENSIONAL TRANSFORMATIONS

- ✓ Geometric transformations provide a means by which an image can be enlarged in size, or reduced, rotated, or moved. These changes are brought about by changing the co-ordinates of the picture to a new set of values depending upon the requirements.
- ✓ Geometric transformations have numerous applications in geometric modeling, e.g., manipulation of size, shape, and location of an object.
- ✓ In CAD, transformation is also used to generate surfaces and solids by sweeping curves and surfaces, respectively.
- ✓ The term 'sweeping' refers to parametric transformations, which are utilized to generate surfaces and solids.
- ✓ During sweep, a curve is transformed through several positions along or around an axis, generating a surface. The appearance of the generated surface depends on the number of instances of the transformation.
- ✓ A parameter t is varied from 0 to 1, with the interval value equal to the fraction of the parameter.
- ✓ For example, to generate 10 instances, the parameter will have a value $t/10$. Transformations of simple cases are only discussed. There are two types of transformations:

Modeling Transformation:

- ✓ This transformation alters the coordinate values of the object.
- ✓ Basic operations are scaling, translation, rotation and, combination of one or more of these basic transformations.
- ✓ Examples of these transformations can be easily found in any commercial CAD software.
- ✓ For instance, AutoCAD uses SCALE, MOVE, and ROTATE commands for scaling, translation, and rotation transformations, respectively.

Visual Transformation:

- ✓ In this transformation there is no change in either the geometry or the coordinates of the object.
- ✓ A copy of the object is placed at the desired sight, without changing the coordinate values of the object.
- ✓ In AutoCAD, the ZOOM and PAN commands are good examples of visual transformation.

BASIC MODELING TRANSFORMATIONS

There are three basic modeling transformations:

- ✓ Translation Scaling
- ✓ Rotation
- ✓ Other transformations, which are modification or combination of any of the basic transformations, are
- ✓ Shearing, Mirroring, copy, etc.

Translation Transformation

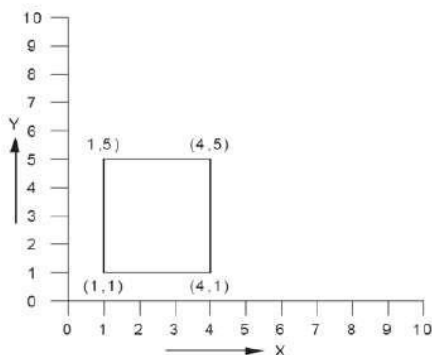
- ✓ Moving drawing or model across the screen is called translation.
- ✓ This is accomplished by adding to the co-ordinates of each corner point the distance through which the drawing is to be moved (translated).
- ✓ In translation, every point on an object translates exactly the same distance. The effect of a translation
- ✓

transformation is that the original coordinate values increase or decrease by the amount of the translation along the x, and y axes.

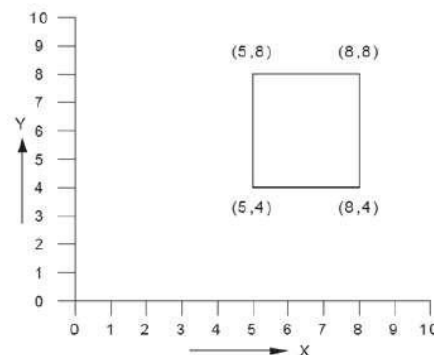
- ✓ The translation operation requires a translation vector, that is, a translation distance and a translation direction.
- ✓ If the layer modifier is used, a different layer from that of the existing one may be chosen for the copy.

Scaling

✓



Original Rectangle



Rectangle After Translation

To change the size of an object, scaling transformation is used.

- ✓ In the scaling process, the dimensions of the object can be enlarged or reduced.
- ✓ In scaling transformation, the original coordinates of an object are multiplied by the given scale factor.
- ✓ Alternative names of the operation used by CAD/CAM systems are magnify or zoom.
- ✓ There are two types of scaling transformations: uniform and non-uniform.
- ✓ In the uniform scaling, the coordinate values change uniformly along the x, y, and z coordinates, whereas, in non-uniform scaling, the change is not necessarily the same in all the coordinate directions.

Rotation

- ✓ In rotation, we rotate the object at particular angle θ (theta) from its origin.

Shearing

- ✓ A shearing transformation produces distortion of an object or an entire image. There are two types of shears: X-shear and Y-shear.
- ✓ A transformation that slants the shape of an object is called the shear transformation.
- ✓ One shifts X coordinates values and other shifts Y coordinate values. However; in both the cases only one coordinate changes its coordinates and other preserves its values.
- ✓ Shearing is also termed as Skewing.

- ✓ The X-Shear preserves the Y coordinate and changes are made to X coordinates, which causes the vertical lines to tilt right or left.
- ✓ The Y-Shear preserves the X coordinates and changes the Y coordinates which causes the horizontal lines to transform into lines which slopes up or down.

Mirroring/ Reflection

- ✓ In modeling operations, one frequently used operation is mirroring an object.
- ✓ Mirroring is a convenient method used for copying an object while preserving its features.
- ✓ Reflection is the mirror image of original object.
- ✓ In reflection transformation, the size of the object does not change.

HOMOGENEOUS TRANSFORMATIONS

- ✓ The above transformations with the exception of translation can be represented as a row vector X, Y and a 2 X 2 matrix.
- ✓ All the four transformations discussed above can be represented as a product of a 1 X 3 row vector and an appropriate 3 X 3 matrix.
- ✓ The conversion of a two-dimensional co-ordinate pair (X, Y) into a 3- dimensional vector can be achieved by representing the point as [X Y 1].
- ✓ After multiplying this vector by a 3 X 3 matrix, another homogeneous row vector is obtained [X1 Y1 1].
- ✓ The first two terms in this vector are the co-ordinate pair which is the transform of (X, Y).
- ✓ This/The three dimensional representation of a two dimensional plane is called homogeneous coordinates and the transformation using the homogeneous co-ordinates is called homogeneous transformation.

Translation:

$$[X1 \ Y1 \ 1] = [X \ Y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}$$

Rotation

$$[X1 \ Y1 \ 1] = [X \ Y \ 1] \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Scaling

$$[X_1 \ Y_1 \ 1] = [X \ Y \ 1] \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

X-shear

$$[X_1 \ Y_1 \ 1] = [X \ Y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Y-shear

$$[X_1 \ Y_1 \ 1] = [X \ Y \ 1] \begin{bmatrix} 1 & Sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Scaling With an Arbitrary Point

- ✓ In uniform scaling, all points and their coordinates are scaled by a factor s . Therefore, unless the fixed point is located at $(0, 0)$, it will be moved to a new location with coordinates s -times x and s -times y .
- ✓ To scale an object about a fixed point, the fixed point is first moved to the origin and then the object is scaled.
- ✓ Finally, the object is translated or moved so that the fixed point is restored to its original position. The transformation sequence is,

Combined Transformations

- ✓ Most applications require the use of more than one basic transformation to achieve desired results. Scaling with an arbitrarily fixed point involves both scaling and translation. And rotation around a given point, other than the origin, involves rotation and translation.
- ✓ Composite transformation can be achieved by concatenation of transformation matrices to obtain a combined transformation matrix.
- ✓ Sequences of transformations can be combined into a single transformation using the concatenation process.
- ✓ The change in the order of transformation would lead to different results, as in general matrix multiplication is not cumulative, that is $[A] \cdot [B] \neq [B] \cdot [A]$ and the order of multiplication.
- ✓ The basic purpose of composing transformations is to gain efficiency by applying a single composed transformation to a point, rather than applying a series of transformation, one after another.
- ✓ For example, to rotate a line AB about an arbitrary point (X_p, Y_p) , we have to carry out three steps

i) Translate point (X_p, Y_p) to the origin ii) Rotate it

about the origin

iii) Finally, translate the center of rotation back where it belonged.

THREE DIMENSIONAL TRANSFORMATIONS

✓ It is often necessary to display objects in 3-D on the graphics screen. The transformation

✓ matrices developed for 2-dimensions can be extended to 3-D.

i. **Scaling:** The scaling matrix in 3-D is:

$$\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ii. **Translation:** The translation matrix is:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix}$$

iii. **Rotation:** Rotation in 3-D can be about X - , Y - or Z axis.

Rotation about Z axis: $R_z =$

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

Similarly $R_x =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

similarly

and $R_y =$

$$\begin{bmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

GRAPHIC PRIMITIVES : LINE DRAWING

- ✓ A drawing is created by an assembly of points, lines, arcs, circles. In computer graphics drawings are created in a similar manner.
- ✓ Each of these is called an entity. The drawing entities that a user may find in a typical CAD package include
 - point
 - line
 - construction line,
 - multi-line,
 - polyline
 - circle
 - spline
 - arc
 - ellipse
 - polygon
 - rectangle

POINT PLOTTING

- ✓ The frame buffer display is an example of a point plotting device. The smallest unit accepted by such displays is a single pixel.
- ✓ To construct a useful picture on a point plotting device, a picture must be built out of several hundreds of pixels.

DRAWING OF LINES

- ✓ Straight line segments are used a great deal in computer generated pictures. The following criteria have been stipulated for line drawing displays :
 - i. Lines should appear straight
 - ii. Lines should terminate accurately
 - iii. Lines should have constant density
 - iv. Line density should be independent of length and angle
 - v. Line should be drawn rapidly
- ✓ The process of turning on the pixels for a line segment is called vector generation.

- ✓ If the end points of the line segment are known, there are several schemes for selecting the pixels between the end pixels.
- ✓ Important methods (algorithms) of generating a line segment are a symmetrical digital differential analyzer (DDA) and Bresenham's Algorithm.

DDA ALGORITHM

- ✓ The digital differential analyzer generates lines from the differential equations.
- ✓ DDA is scan conversion line-generating algorithm, based on the calculation of either Δy or Δx differences along the line path.
- ✓ A line can be generated by increasing unit interval in one coordinate and determining the corresponding integer values of interval in other coordinate nearest to the line path.
- ✓ In the real world of limited precision displays, addressable pixels only must be generated.
- ✓ This can be done by rounding to the next integer after each incremental step. After rounding, a pixel is displayed at the resultant X and Y locations. An alternative to rounding is the use of arithmetic overflow.
- ✓ X and Y are kept in registers that have integer and fractional parts.
- ✓ The incrementing values which are less than unity are repeatedly added to the fractional part and whenever the result overflows the corresponding integer part is incremented.
- ✓ The integer parts of X and Y are used to plot the line.
- ✓ This would normally have the effect of truncating. The DDA is therefore initialized by adding 0.5 in each of the fractional parts to achieve true rounding.
- ✓ This is because the points that are plotted must be pixel grid points and many of these may not lie on the actual line.
- ✓ For a line segment joining two points P1 and P2, a parametric representation is given by $P(u) = P1 + (P2 - P1)u$; When P(u) is a position vector, u is small increment and $0 \leq u \leq 1$.
- ✓ Once the address of all the pixels is determined, they are stored in a frame buffer.
- ✓ The display driver reads the array of addresses and it illuminates the corresponding pixels.
- ✓ To look a line in continuous manner, it is necessary to increment the parameter u according to the resolution of the display device.
- ✓ If too large increment is selected, the line may appear to be discontinuous. The symmetrical DDA generates reasonably accurate lines since a displayed pixel is never away from a true line by half the pixel unit.

- ✓ It can be noted that lines drawn on a raster display may have a jagged or staircase appearance unless the lines are vertical or horizontal.

Advantages

- ✓ It is the simplest algorithm. It is a faster method.
- ✓ It eliminates the multiplication of necessary increments.

Disadvantages

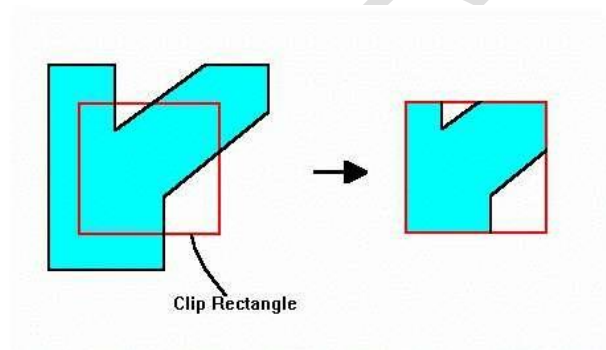
- ✓ Floating point arithmetic in DDA algorithm is time-consuming.
- ✓ The algorithm is orientation dependent. Therefore, accuracy is poor.

Bresenham's Line Drawing Algorithm

- ✓ Bresenham's algorithm selects optimum raster locations with minimum computation.
- ✓ This algorithm always increments by one unit in either X or Y depending upon the slope of the line.
- ✓ The increment in the other variable either zero or one is determined by examining the distance (error) between the actual line location and the nearest grid location.
- ✓ This distance is called decision variable or error. Only the sign of this error needs be examined.
- ✓ This is accurate and efficient method of raster generation algorithms to display lines, circles, ellipse and other curves incorporating only incremental integer calculation.
- ✓ Bresenham's line drawing algorithm rectifies the disadvantage of DDA Algorithm.
- ✓ The features are: The line generated by this technique uses standard equation of line ($y=mx + b$)

CLIPPING

- ✓ Various projections of an object's geometry can be defining views.
- ✓ A view requires a view window.
- ✓ If any part of the geometry is not inside the window, it is made invisible by the CAD software through a process known as clipping.
- ✓ Clipping is the process of determining the visible portions of a drawing lying within a window.
- ✓ Clipping, in the context of computer graphics, is a method to selectively enable or disable rendering operations within a defined region of interest.
- ✓ In clipping each graphic element of the display is examined to determine whether or not it is completely inside the window, completely outside the window or crosses a window boundary.
- ✓ Portions outside the boundary are not drawn.
- ✓ Any geometry lying wholly outside the view boundary is not mapped to the screen, and any geometry lying partially inside and partially outside is cut off at the boundary before being mapped.
- ✓ Typical clipping algorithm: Cohen-Sutherland clipping algorithm.
- ✓ If clipping is not done properly a CAD system will produce incorrect pictures due to an overflow of internal coordinate registers. This effect is known as wrap round.



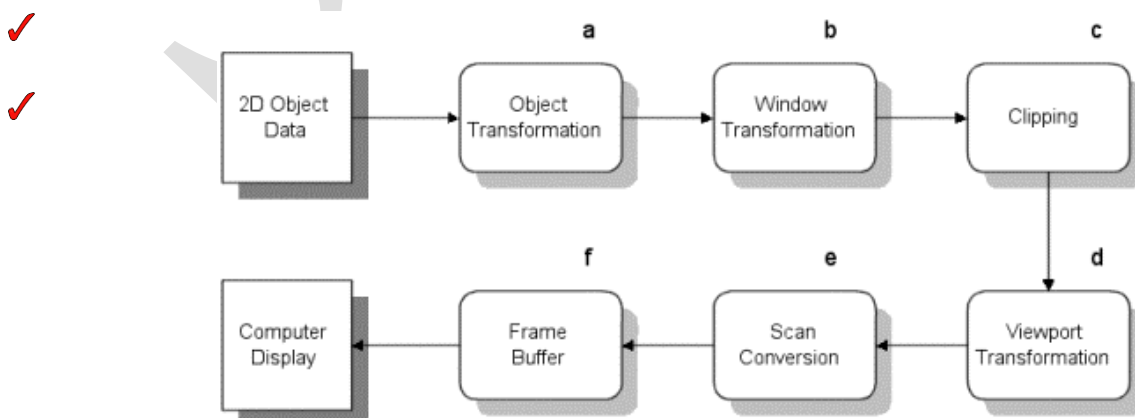
Applications of Clipping:

- ✓ Extracting part of defined scene for viewing
- ✓ Identifying visible surfaces in 3D Views
- ✓ Anti-aliasing line segments or object boundaries
- ✓ Creating objects using solid modelling procedures
- ✓ Drawing and painting operations

Viewing Transformation

For the complex drawings of an object/element, World Coordinate system become very difficult to visualize. In such cases a window is required to visualize portions of the drawing on the display device.

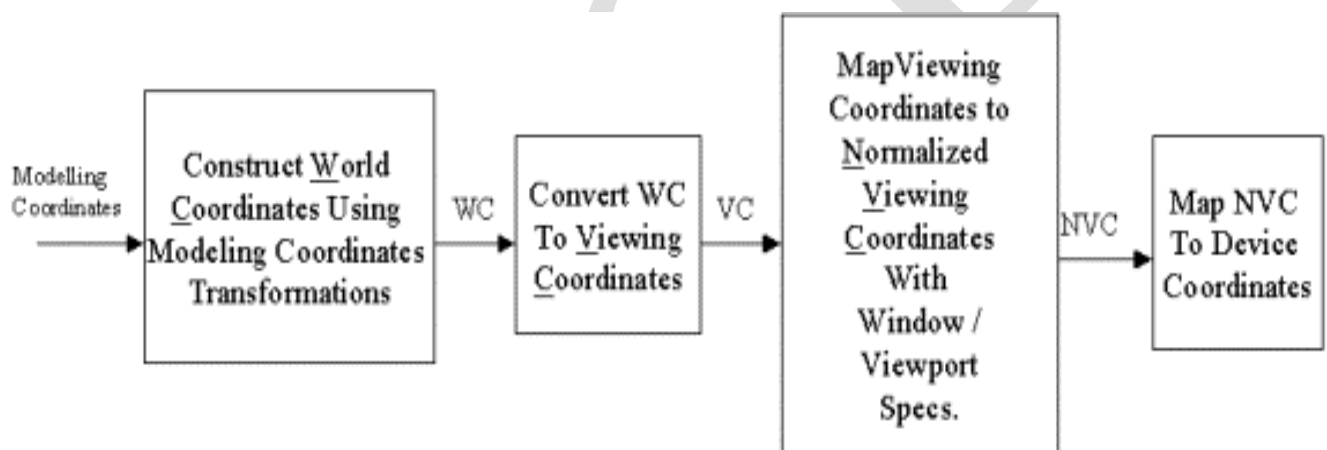
Viewing of an Image



Window – world coordinate area for display (what is viewed) A window defines a

rectangular area in world coordinates.

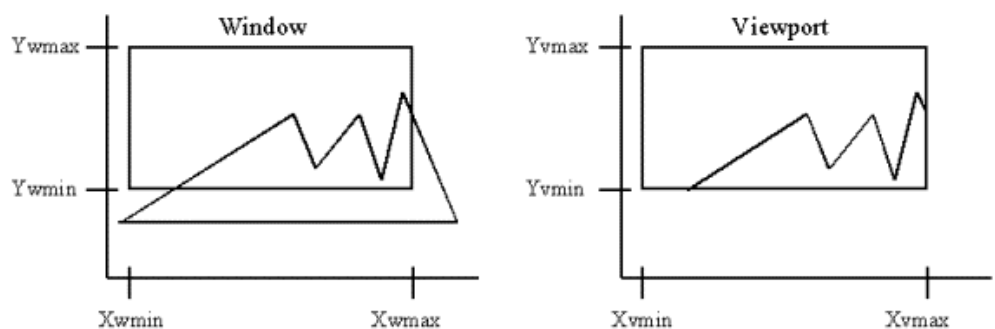
- ✓ Normally an image is modelled in World Device Coordinates (WDC).
- ✓ When the interest is a portion of it then the portion of interest which is a polygonal area can be specified in world coordinates, called the "window".
- ✓ Viewport—area on display device to which window is mapped (where it is displayed)
- ✓ A viewport is defined in normalized coordinates, a rectangular area on the display device where the image of the data appears.
- ✓ The user may want to create images on different parts of the screen. For that the viewport is defined in Normalized Device Coordinates (NDC).
- ✓ Using NDC allows for output device independence. Later NDC to Physical Device Coordinates (PDC) is possible.
- ✓ Viewing Transformations – mapping part of a world coordinate scene to device coordinates – i.e., Window to Viewport Mapping



Viewing Transformation of an Image

- ✓ Viewing transformation is the process of mapping coordinates/points from window onto the viewport.
- ✓ When we define an image in world coordinate system, to display that image we must map the image to the physical output device.

- ✓ Transformation operations are useful CAD/CAM productivity tools for manipulating existing entities.



- ✓ They can be used to transform coordinates from World Coordinates to Device Coordinates.
- ✓ These operations and their related input are based on the mathematical relations.

- ✓ A copy modifier is usually available with these operations, in which case a copy of the existing entity is made and acted upon by the operation, thus keeping the existing entity unchanged.
- ✓ The translation operation requires a translation vector, that is, a translation distance and a translation direction.
- ✓ The rotation operation requires a rotational vector, that is, an axis of rotation and an angle of rotation.
- ✓ The angle of rotation could be positive or negative. A positive angle is an angle represented by a vector whose direction is in positive direction of the axis of rotation.
- ✓ The mirror operation requires a mirroring axis. The existing entity is always mirrored onto the other side of the mirroring axis in reverse, that is, as if it were being viewed in a mirror.
- ✓ The scale operation requires a scale factor.
- ✓ Alternative names of this operation which are also used by CAD/CAM systems are zooming or magnify.
- ✓ The operation is usually used to adjust the size of displayed geometry within a view window.
- ✓ If the geometry is too small, the operation can enlarge it to fill the view window.

SANJEEV

Representation of curves- Hermite curve- Bezier curve- B-spline curves -rational curves- Techniques for surface modeling – surface patch- Coons and bicubic patches- Bezier and B- spline surfaces. Solid modeling techniques- CSG and B-rep

INTRODUCTION

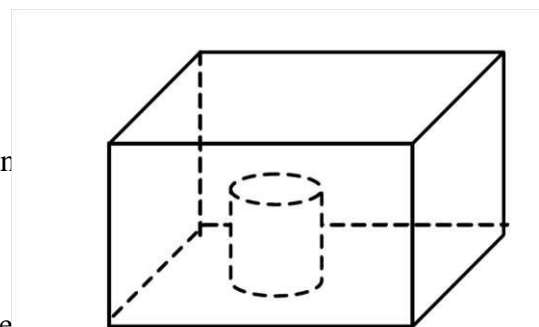
- ✓ In order to understand the significance of curves, we should look into the types of model representations that are used in geometric modeling.
- ✓ Curves play a very significant role in CAD modeling, especially, for generating a wireframe model, which is the simplest form for representing a model.
- ✓ There are three different forms in which a model can be represented in CAD:
 - Wireframe model
 - Surface Model
 - Solid model

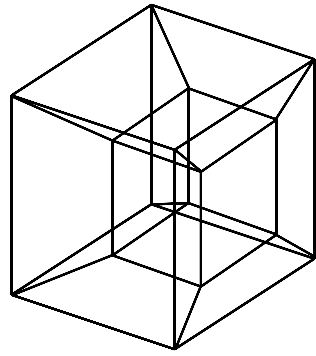
Wireframe model:

- ✓ Joining points and curves creates wireframe models.
- ✓ A model consist entirely lines, edges, points, curves that's called wire frame modelling.
- ✓ It looks as if it's made up with a bunch of wires. This is the simplest CAD model of an object.
- ✓ A wire-frame model consists of a finite set of points together with the edges connecting various pairs of these points, so that a drawing of these points and edges aids the visualization of the object.
- ✓ It represents the entire object, and it shows what it consist inside also. It consists of two types of information.
- ✓ The first is termed 'metric' or 'geometric' data which relate to the three- dimensional coordinate positions of the wire-frame node' points in space.
- ✓ The second is concerned with the 'connectivity' or 'topological' data, which relates pairs of points together as edges.
- ✓ The wireframe model creation is somewhat similar to drawing a sketch by hand to communicate or conceptualize an object.

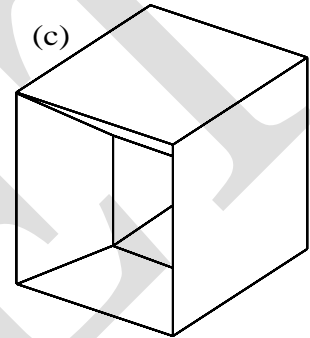
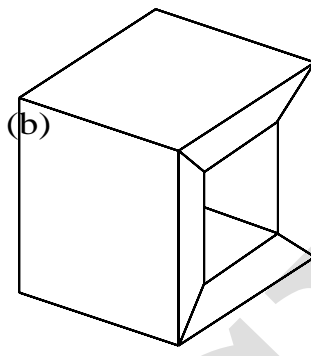
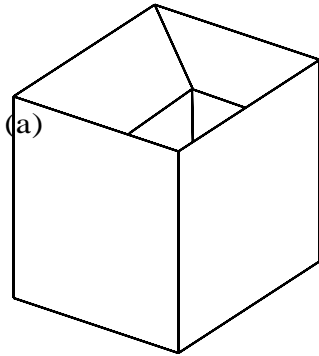
Advantages

- ✓ To represent the parts of model.
- ✓ Can be created quickly and easily to verify a concept of an object and design ideas.
- ✓ We can see the entire object at any view of that.
- ✓ Ease of creation and low level hardware and software requirements.
- ✓ The data storage requirement is low.





Wireframe
ambiguity: Is this object
(a), (b) or (c) ?



Limitations of a wireframe model

- ✓ It can be very confusing to visualize.
- ✓ Difficult to find the interferences.
- ✓ Complicate objects can be confused with many edges.
- ✓ Not possible to calculate volume and mass.
- ✓ Limited usage, it is not used in manufacturing and production.
- ✓ These models can be ambiguous and unable to provide mass property calculations, hidden surface removal, or generation of shaded images.

Surface Model:

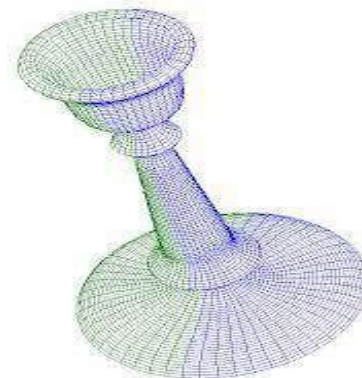
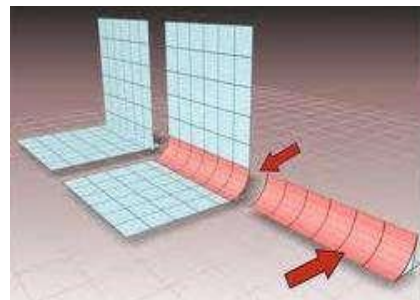
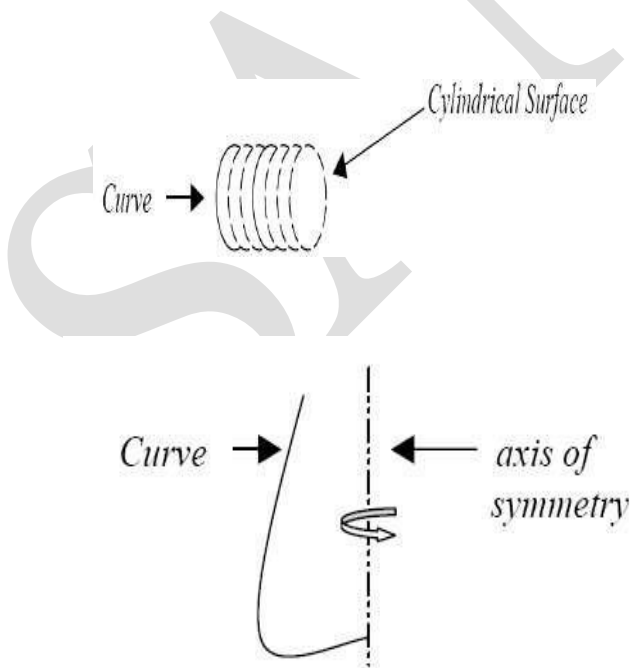
- ✓ It is an extension of a wire frame model with additional information.
- ✓ A wire frame model can be extracted from a surface model by deleting all surface entities (not the wireframe entities – point, lines, or curves!).
- ✓ Surface models can be shaded and represented with hidden lines.
- ✓ A surface of an object is more complete and less ambiguous representation than its wire frame model
- ✓ The appearance or resolution of a surface model depends on the number of sweeping instances we select.
- ✓ For a realistic looking model, we need to select a large number of instances, requiring a large computer memory, or, opt for a not-so realistic model by selecting a small number of instances, and save memory.
- ✓ In some commercial CAD packages we have the option of selecting the resolution of a model; other packages have a fixed value for resolution that cannot be changed by users.

- ✓ Surface models are used for modeling surfaces of engineering components. We can create the complicated shapes by using surface modelling.
- ✓ Like car, ships, airplane wing and castings.
- ✓ Surface models are useful for representing surfaces such as a soft-drink bottle, automobile fender, aircraft wing, and in general, any complicated curved surface.

Types of surfaces

- ✓ Simple and basic form of Surface is flat Surface. Complex Surface Known as Sculptured Surface.
- ✓ Surface produce from combining two Families of Curves that Intersect one Another in a criss-cross manner, Creating network of inter-connected Patches.

1. Plain Surface
2. Rules (lofted) Surface
3. Surfaces of Revolution
4. Translated surface
5. Coon Patch
6. Fillet Surface
7. Offset Surface
8. Bezier Surface
9. B-spline Surface



Advantages:

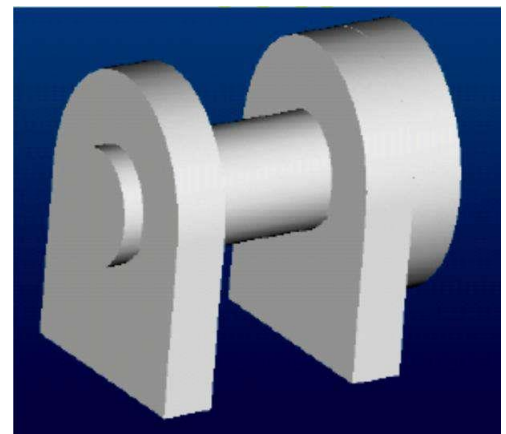
- ✓ Compare with the wire frame it gives more information. It gives hidden lines and shadings.
- ✓ It used to analysing the element(FEA).
- ✓ Support shading, Support volume and mass calculation, finite element modeling, NC path generation, cross sectioning, and interference detection. (when complete)

Limitations:

- ✓ Require more training and mathematical background of the users.
- ✓ Require more CPU time and memory
- ✓ Only skilled persons who handled this work.
- ✓ Still ambiguous; no topological information Awkward to construct

Solid Model:

- ✓ Solid models are the most preferred form of CAD models and represent unambiguous image of a component.
- ✓ Solid models represent objects in a very realistic and unambiguous form; however, they require a large amount of storage memory and high-end computer hardware.
- ✓ A solid model can be shaded and rendered in desired colors to give it a more realist appearance.
- ✓ Representation of an object by a solid model is relatively a new concept. There were only a couple of solids modeling CAD programs available in late 1980s, and they required mainframe computers to run on.
- ✓ However, in 1990s, due to the low cost and high speed, PCs have become the most popular solid modeling software platform, prompting almost all the CAD vendors to introduce their 3-D solid modeling software that will run on a PC.
- ✓ A solid model can be used to analyze the moment of inertia, mass, volume, sections of the model, etc.
- ✓ Solid models are mathematical models of objects in the real world that satisfy specific properties, listed below.



1. Bounded: The boundary must limit and contain the interior of the solid.

2. Homogeneously Three-Dimensional: No dangling edges or faces be present so that the boundary is always in contact with the interior of the solid.

Solid Model Creation Scheme

There are six common representations in solid modeling.

- i.** Spatial Enumeration: In this simplest form of 3D volumetric raster model, a section of 3D space is described by a matrix of evenly spaced cubic volume elements called voxels.
- ii.** Cell Decomposition: This is a hierarchical adaptation of spatial enumeration. 3D space is sub-

divided into cells. Cells could be of different sizes. These simple cells are glued together to describe a solid object.

- iii. **Boundary Representation:** 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.
 - iv. **Sweep Methods:** In this technique a planar shape is moved along a curve. Translational sweep can be used to create prismatic objects and rotational sweep could be used for axisymmetric components.
 - v. **Primitive Instancing:** This modeling scheme provides a set of possible object shapes which are described by a set of parameters. Instances of object shape can be created by varying these parameters.
 - vi. **Constructive Solid Geometry (CSG):** Primitive instances are combined using Boolean set operations to create complex objects.
- ✓ In most of the modeling packages, the approach used for modeling uses any one of the following three techniques:
- i. Constructive solid geometry (CSG or C-Rep)
 - ii. Boundary representation (B-Rep)
 - iii. Hybrid method which is a combination of B-Rep and CSG.

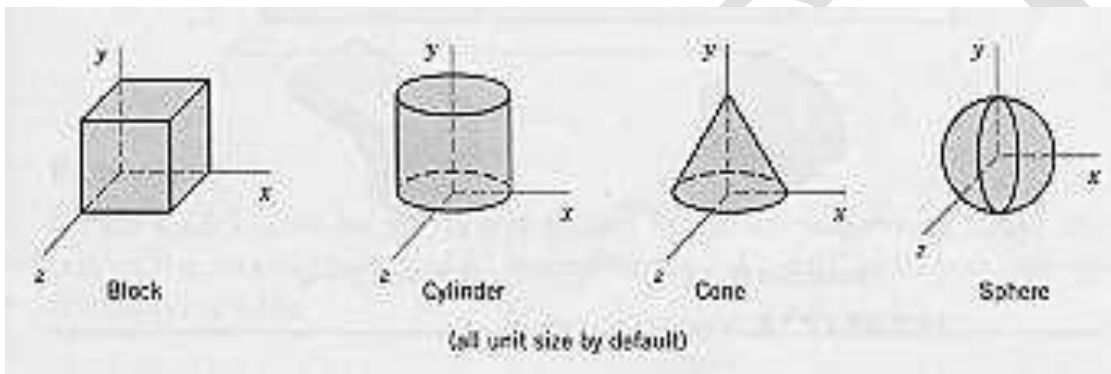
CONSTRUCTIVE SOLID GEOMETRY (CSG)

- ✓ This scheme is based on the principle that two primitives can be combined to produce a new solid model.
- ✓ Introduced: Ian Braid (Cambridge University, ~74) This method is also known as 'Building Block' method.
- ✓ The scheme uses the Union, Intersection, and Subtraction techniques to create three-dimensional models, which are based on the Boolean operation.
- ✓ In a CSG model, physical objects are created by combining basic elementary shapes known as primitives like blocks, cylinders, cones, pyramids and spheres.
- ✓ The steps involved in generating a solid model are:
 1. Select the primitives from a library
 2. Go through the scaling, dimension modification, and any other transformations.
 3. Combine the primitives to create the desired solid model.
- ✓ Since CSG method uses solid primitives, internal details of the object are automatically contained in the model.
- ✓ The model can be sectioned to reveal internal details and can be used for calculating mass, volume, moment of inertia, etc.
- ✓ New solid models can be created from the primitives or other solid models by the following Boolean

operations like union (\cup), difference ($-$) and intersection (\cap).

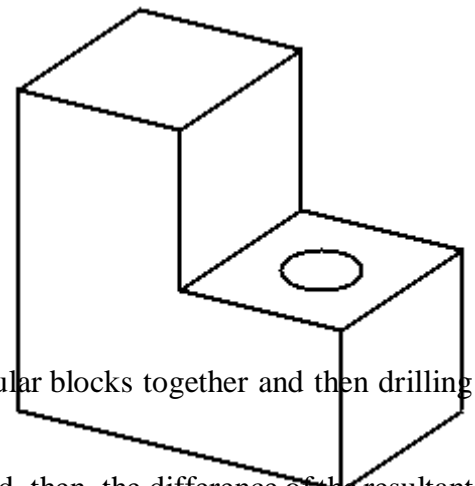
- ✓ Union (\cup): Two solids are joined and the common volume of one of the primitives is neglected in the resulting solid.
- ✓ Subtraction or Difference ($-$): One solid is subtracted from the other and the resultant solid retains only the uncut portion of the solid.
- ✓ Intersection (\cap): When two solids are combined, the resultant solid represents the common volume of the two solids.
- ✓ Note that the above definitions are not rigorous and have to be refined to define the Regularized Boolean Set Operations to avoid impossible solids being generated.

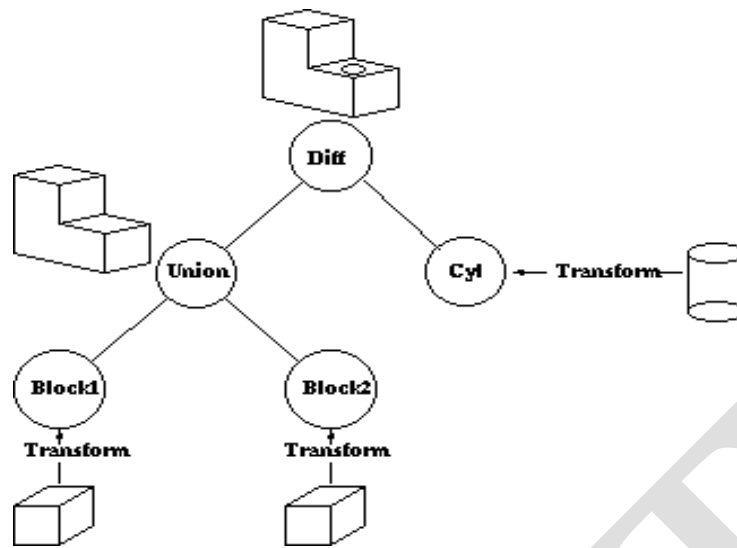
Examples of primitives used in CSG



Typical standard primitives are: Cone, cylinder, sphere, torus, block, closed spline surface, right angular Wedge and Swept solids (A revolution or linear sweep of a planar face which may contain holes).

- ✓ The method of Constructive Solid Geometry arose from the observation that many industrial components derive from combinations of various simple geometric shapes such as spheres, cones, cylinders and rectangular solids.
- ✓ In fact the whole design process often started with a simple block which might have simple shapes cut out of it, perhaps other shapes added on etc. in producing the final design.
- ✓ This simple component could be produced by gluing two rectangular blocks together and then drilling the hole.
- ✓ Or in CSG terms the union of two blocks would be taken and then the difference of the resultant solid and a cylinder would be taken.
- ✓ In carrying out these operations the basic primitive objects, the blocks and the cylinder, would have to be scaled to the correct size, possibly oriented and then placed in the correct relative positions to each other before carrying out the logical operations.
- ✓ A CSG model is then held as a tree structure whose terminal nodes are primitive objects together with an appropriate transformation and whose other nodes are Boolean Set Operations.
- ✓ This is illustrated for the objects which are constructed using cube and cylinder primitives.





Method of CSG approach

Advantages (For more points refer the text book)

- ✓ CSG methods are useful both as a method of representation and as a user interface technique.
- ✓ A user can be supplied with a set of primitive solids and can combine them interactively using the Boolean set operators to produce more complex objects.
- ✓ Editing a CSG representation is also easy, for example changing the diameter of the hole in the example above is merely a case of changing the diameter of the cylinder.

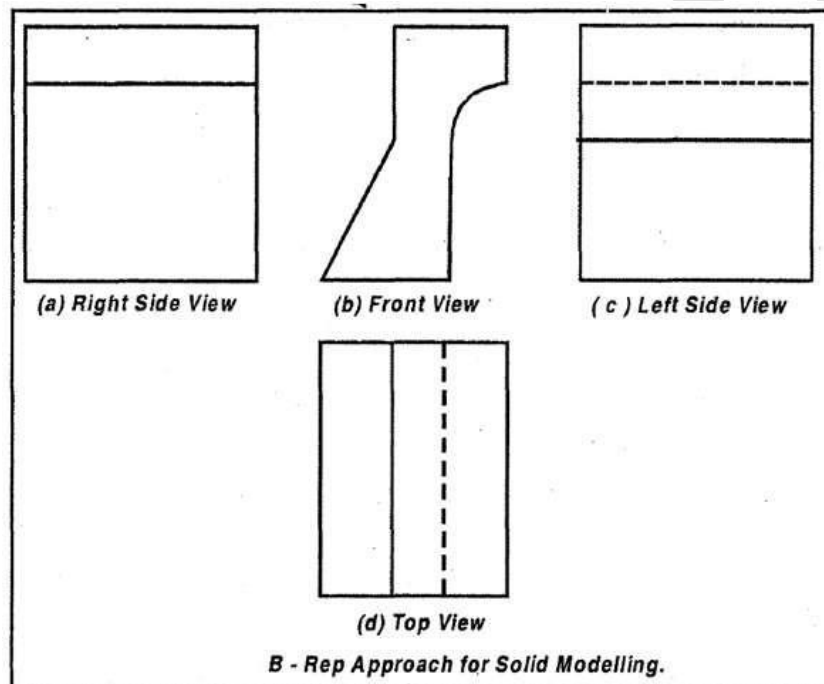
Limitations (For more points refer the text book)

- ✓ It is slow to produce a rendered image of a model from a CSG tree.
- ✓ This is because most rendering pipelines work on B-reps and the CSG representation has to be converted to this form before rendering.
- ✓ Hence some solid modelers use a B-rep but the user interface is based on the CSG representation.

BOUNDARY REPRESENTATION (B-REP) SCHEME

- ✓ B-rep scheme is closely related to the traditional drafting method.
- ✓ In this method, front view, top view, bottom view, side view of an object is sketched and connected by means of lines to create a relationship.
- ✓ The B-rep is built on the idea that a physical object is enclosed by a set of faces, which themselves belong to closed and orientable surfaces.
- ✓ The data base for boundary model has topology and geometry. The boundary of an object includes faces, vertices and edges.
- ✓ A solid model is created by combining faces and contains vertices, edges, loops, and bodies.
- ✓ Euler operations are utilised for creating topology i.e. to create, manipulate and edit the faces, edges and vertices.
- ✓ Geometry is created by euclidean calculations. However, boolean operations are employed for manipulating and editing the model.
- ✓ Computation of the volumetric properties is possible in B-rep. In B-rep, a model is made up of the following primitives:

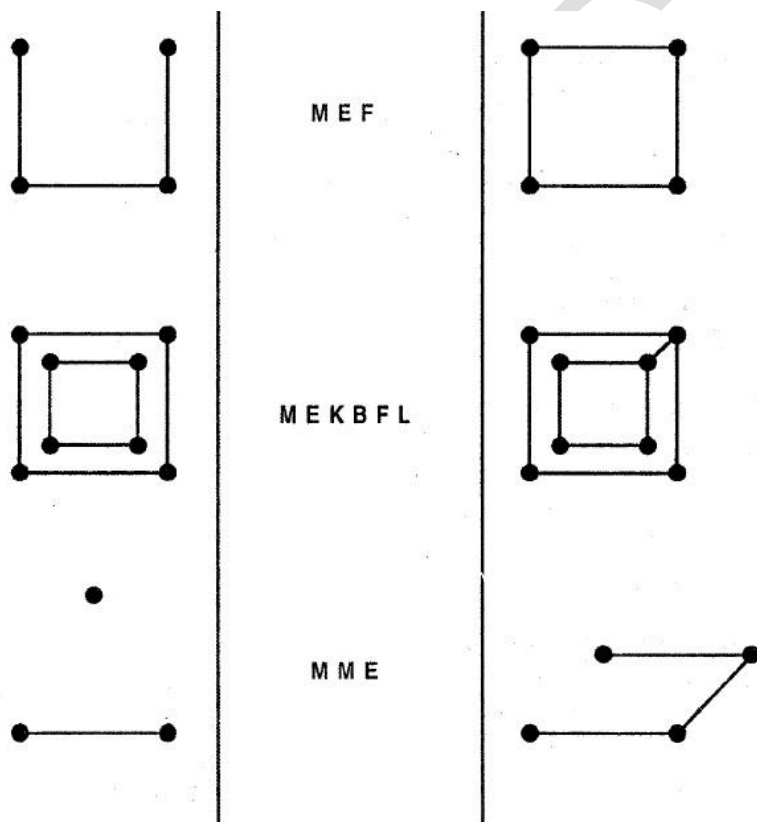
- Vertex: A point in space
- Edge: A finite, no-intersecting space curve bounded by two vertices that are not necessarily distinct.
- Face: A finite connected, non-self-intersecting, region of a closed oriented surface, bounded by one or more loops.
- Loop: An ordered alternating sequence of vertices and edges. A loop defines a non-self-intersecting closed space curve, which may be a boundary of a face.
- Body: Entity that has faces, edges and vertices. A minimum body is a point.



- ✓ The solid geometry will represent the rigid motion, transformation and other metric information's such as angles, volumes, distances.
- ✓ Boundary representation system requires explicit representation of boundary of solid.
- ✓ The translation object can be created by three types of operations.
 - Linear • Rotational • Compound
- ✓ The major operations carried out in B - rep system are:

<ul style="list-style-type: none"> • Extrusion • Revolution • Chamfer 	<ul style="list-style-type: none"> • Tweaking • Blending • Swaging
--	---

Operator	Euler operations	Description of operator
MBFV	→	Make Body Face Vertex
MEV	→	Make Edge, Vertex
MEKL	→	Make Edge, Kill Loop
MEF	→	Make Edge, Face
MEKBFL	→	Make Edge, Kill Body, Face, Loop
MFKLG	→	Make Face Kill Loop, Genus
MME	→	Make Multiple Edges
KVE	→	Kill Vertex, Edge



Only the boundary surfaces of the model are stored and the volumetric properties are calculated by the Gauss Divergence theorem, which relates volume integral to surface integrals.

- ✓ This scheme can model a variety of solids depending on the primitive surfaces (planar, curved, or sculptured).
- ✓ There are two types of solid models in this scheme:
 1. Polyhedral solids
 2. Curved solids

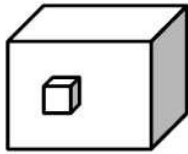
Polyhedral Solids:

- ✓ Polyhedral models consist of straight edges, e.g., a non-cylindrical surface: box, wedge, combination of two or more non-cylindrical bodies, etc.
- ✓ Polyhedral solids can have blind or through holes, and two or three- dimensional faces, with no dangling edges.
- ✓ A valid polyhedral abides by the Euler's equation:

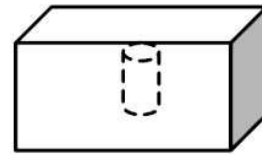
$$F - E + V - L = 2(B - G)$$

Where, F = Face; E = Edge; V = Vertices; L = Inner Loop; B = Bodies; G = through holes

- ✓ A simple polyhedral has no holes; each face is bounded by a single set of connected edges (bounded by one loop of edges).
- ✓ Euler's equation for a simple polyhedral can be reduced to: $F - E + V = 2$ Example: For the box shown, $F = 6$, $E = 12$, and $V = 8$



Polyhedral with two loops



Polyhedral with a blind hole

Curved Solids:

- ✓ A curved solid is similar to a polyhedral object but it has curved faces and edges. Spheres and cylinders are examples of curved solids.

Advantages of B-rep(For more points refer the text book)

- ✓ This method is very powerful for creating complex shapes solid models. B-rep model can be easily converted into wire frame model system.
- ✓ B-rep system stores an explicit definition of the model boundaries. B-rep system is very much compatible with other systems.

Limitations of B-rep(For more points refer the text book)

- ✓ This requires more storage space.
- ✓ This concept cannot be applied for tool path generation.

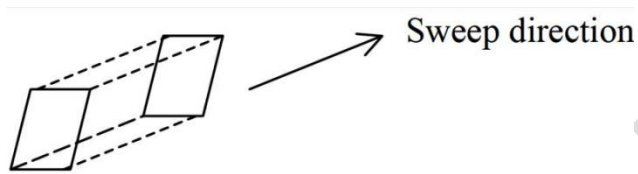
SWEEPING SCHEME (Not in the scope of the syllabus)

- ✓ Sweeping can create a solid model.
- ✓ The method is useful for creating 2 ½ – dimension models. The generated models are axisymmetric and have uniform thickness (i.e., extruded models).
- ✓ There are two types of sweeps: linear and rotational.
- ✓ In linear sweep, a closed 2- D sketch is extruded through the desired length, creating a homogeneous

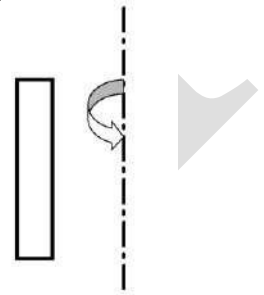
and axisymmetric model, as shown in the figure.

- ✓ In rotational sweep, a closed sketch is rotated around an axis. The generated model is always axisymmetric.
- ✓ In addition to the two sweeps described above, a model can also be created by a non-linear sweep.
- ✓ In this type of sweep, a closed sketch is swept along a non-linear path.

Linear sweep



Creating a box by Sweeping a rectangle



Techniques for surface modeling

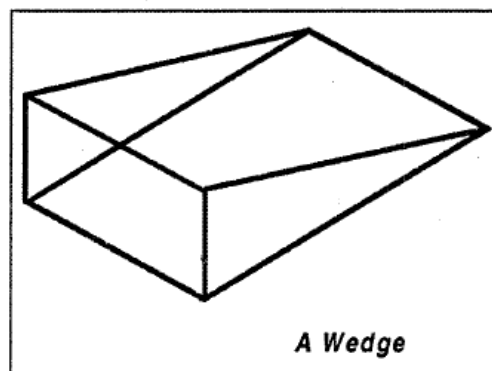
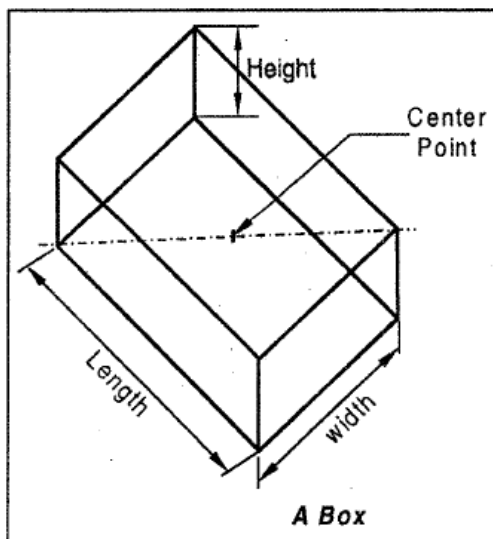
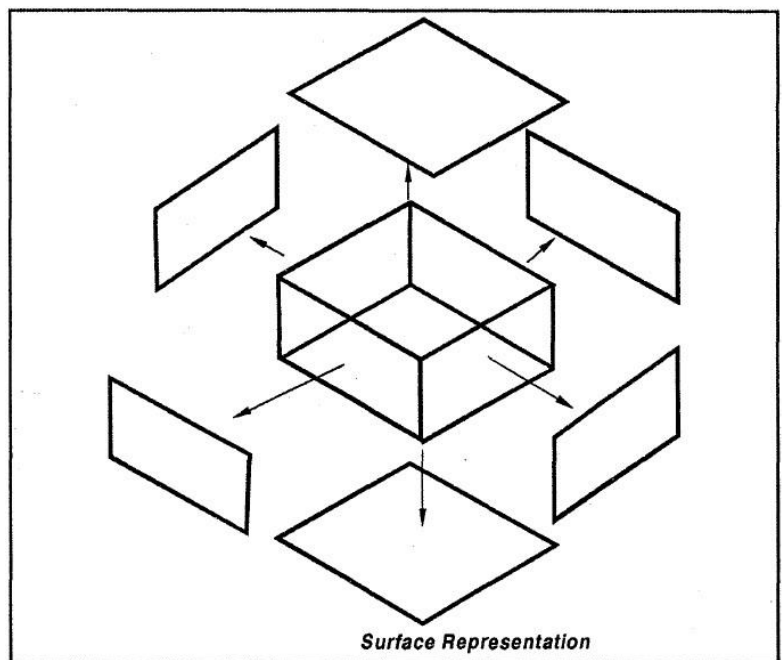
- ✓ The technique of representation of objects (or) components by surfaces is called surface modelling.
- ✓ In this surface modelling, objects (solid objects) can be clearly interpreted by the user (viewer).
- ✓ The major drawback is that, no data is available about the interior of solid. Moreover, mass property and inertia property calculation would be difficult.
- ✓ This surface modelling application is for modelling car-bodies, ships, aerospace structure, dies, etc.
- ✓ The advanced surface modelling techniques like B-splines and NURBS (Non-Uniform Rational B-Splines) are used for creating surfaces.

- ✓ The standard surfaces for modelling include

- (a) Box
- (b) Pyramid wedge
- (c) Sphere
- (d) Dome
- (e) Torus
- (f) Mesh
- (g) Cones

- ✓ Box is a three-dimensional solid specified by length, center and corner

- ✓ Wedge: A 3D solid with a sloped face tapering along the x-axis.

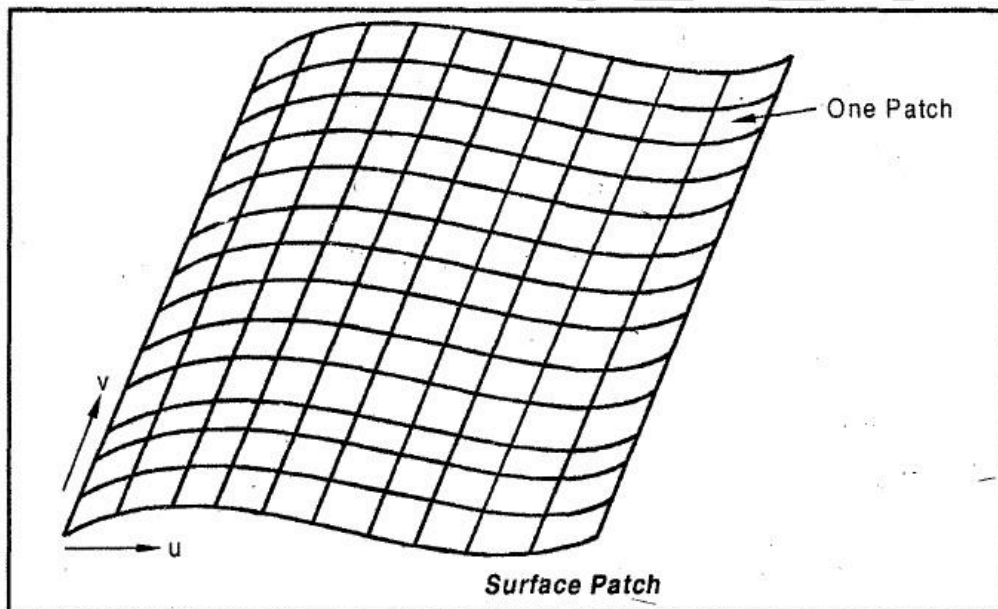


The techniques available for surface modelling are:

- ✓ (a) Surface patch
- ✓ (b) Coons Patch
- ✓ (c) Bicubic Patch
- ✓ (d) Bezier surfaces
- ✓ (e) B-spline surfaces.

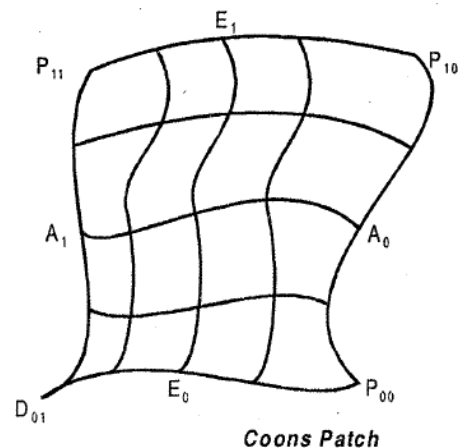
Surface Patch

- ✓ The building blocks of the surfaces are known as patch.
- ✓ Generally u and v are the two variables used for representing a patch. A patch is bi-parametric, values ranging between 0 and 1.
- ✓ For creating Curved patches, one of the parametric variable is fixed.

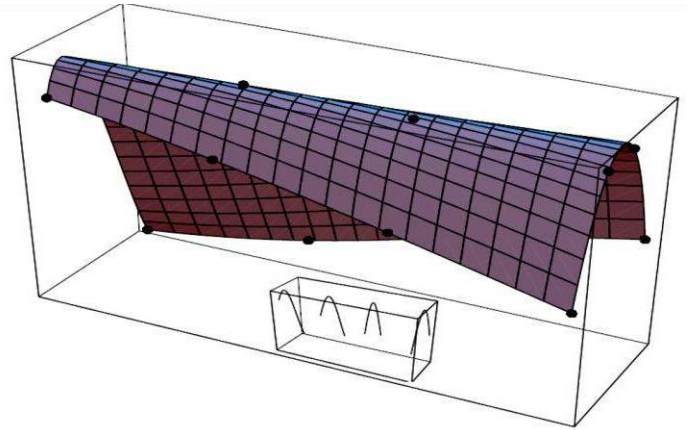
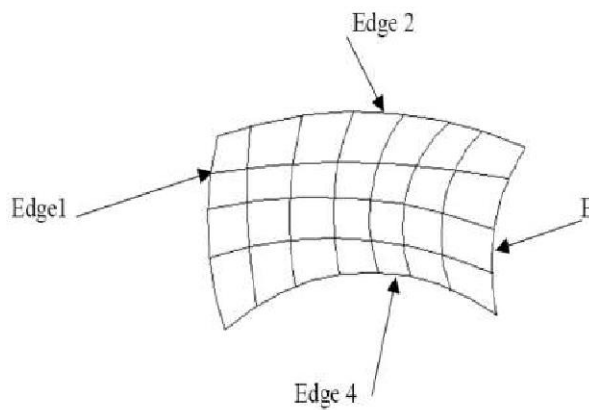


The Coons Patch

- ✓ This type of surface is based on the pioneering work of Steven Anson Coons at MIT in the 1960s.
- ✓ A coons patch is a result of interpolation done on intersecting mesh of curves, resulting in rectangular grid of patches.
- ✓ Coons patch is used to create a surface using Curves that form Closed boundaries.
- ✓ The Patches May be Triangular or Rectangular type .
- ✓ The simplest of coons patch is the linearly lended coons patch.
- ✓ Linear Coons surface, is a generalization of lofted surfaces.
- ✓ This patch is defined by 4 boundary



curves and corner points of patch.

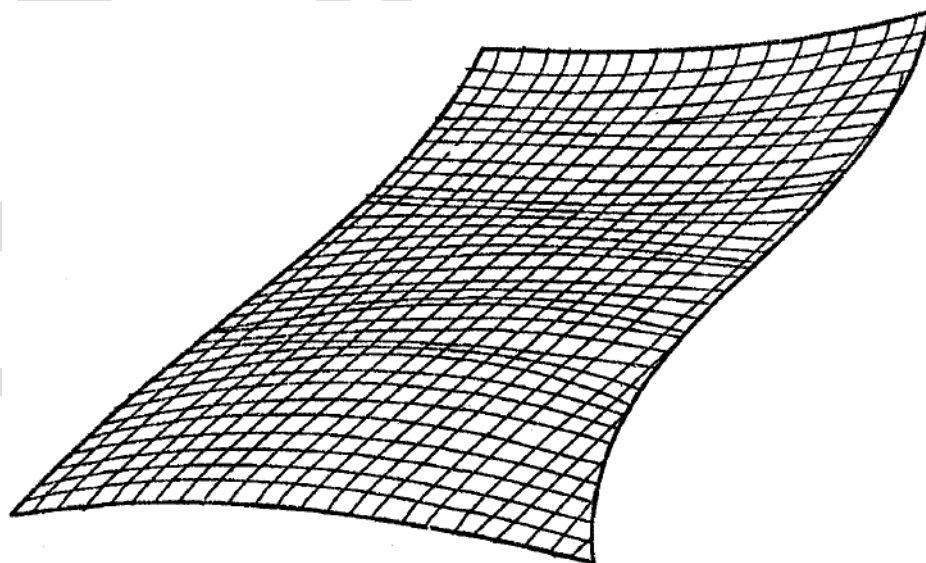


- ✓ All four boundary curves are given, and none has to be a straight line. Naturally, the boundary curves have to meet at the corner points, so these points are implicitly known.
- ✓ Coons decided to search for an expression $P(u,w)$ of the surface that satisfies (1) it is symmetric in u and w and (2) it is an interpolation of $P(u, 0)$ and $P(u, 1)$ in one direction and of $P(0, w)$ and $P(1, w)$ in the other direction.

THE BICUBIC SURFACE PATCH

- ✓ In order to define the surfaces using point and tangent, bicubic patch is referred.
- ✓ The General" form of bicubic Patch is

$$\vec{P}(ab) = \sum_{i=0}^3 \sum_{j=0}^3 \vec{K}_{ij} a^i b^j$$



Bi - Cubic Surface Patch.

BEZIER SURFACE

✓ This is a synthetic Surface that approximate data points.

✓ Bezier Surface is generated from the basics of Bezier Curve.

✓ It can permit twist in the Surface.

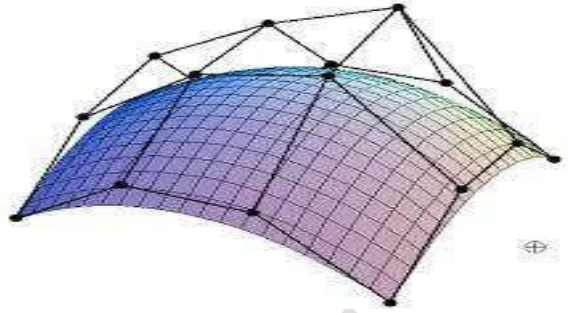
✓ The Surface Does not Pass Through all the Control points. A bezier surface is

✓ given by

$$\vec{P}(a,b) = \sum_{i=0}^m \sum_{j=0}^n \vec{B}_{i,m}(a) \vec{B}_{j,n}(b) \vec{P}_{i,j} \quad a, b, \in [0, 1]$$

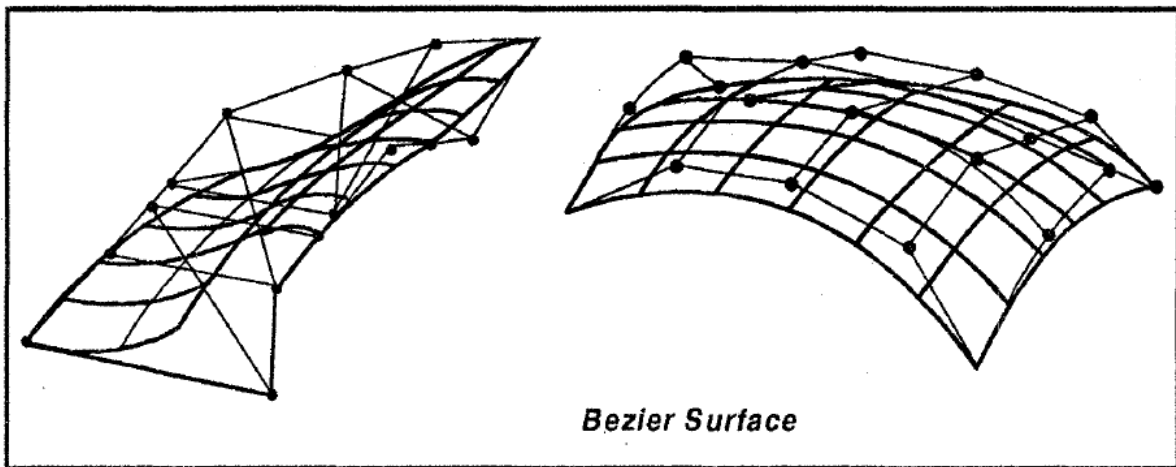
$\vec{P}_{i,j} \rightarrow$ Vertices of Polygon

$\vec{B}_{i,m}$ and $\vec{B}_{j,n} \rightarrow$ Blending Functions.



✓ In Bezier curves, it is possible to convert the straight surface to smooth surface between Patches.

The parametric cubic (PC) curve, Equation is useful, since it can be used when either four points, or



✓ two points and two tangent vectors, are known.

✓ The PC curve can easily be extended to a bicubic surface patch by means of the Cartesian product.

✓ The conclusion is that curves that lie on this surface in the u or in the w directions are parametric cubics.

✓ The four boundary curves are consequently also PC curves. Any change in any of

✓ them produces a different surface patch.

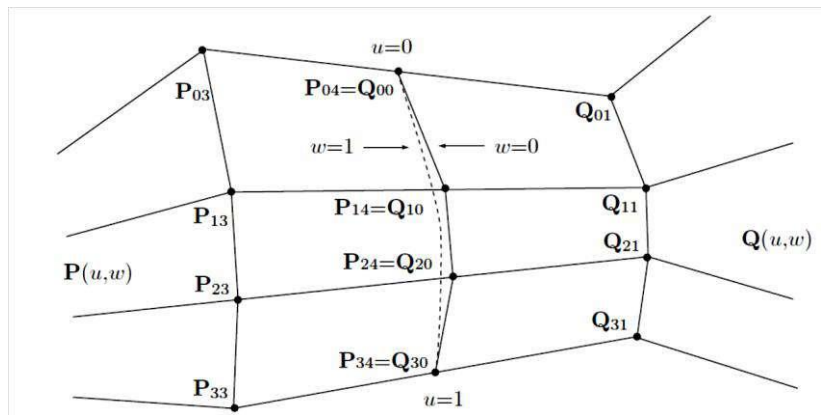
Rational Bezier Surfaces

✓ The rational Bezier curve is the principle of this type of curve can be extended to surfaces.

✓ When all the weights w_{ij} are set to 1, the Equation reduces to the original rectangular Bezier surface patch. The weights serve as additional parameters and provide fine, accurate control of the shape

of the surface.

- ✓ Figure can be pulled toward the center point [point (4, 1, 1)] by assigning $w_{22} = 5$, while keeping the other weights set to 1.

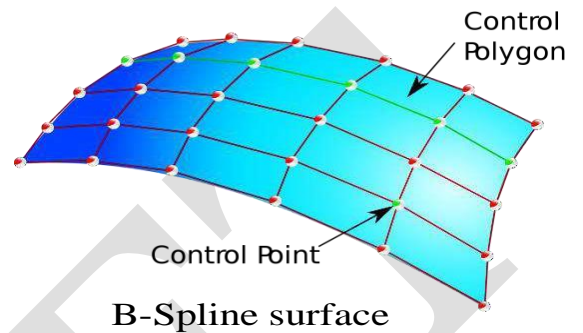


Smoothly joining of Bezier rectangular patch

SANVCET

B-SPLINE SURFACE

- ✓ It is a surface that can approximate or interpolate given input data.
- ✓ It is another synthetic Surface which can be generated through interpolation or approximation type.
- ✓ It is a general surface like a Bezier surface but with the advantage of permitting local control of the surface.
- ✓ In this surface, a characteristic polygon passes through centre points of polygon with its edges tangential to the polygon edge.
- ✓ There is another case, where the control polygon is closed feature.



- ✓ The general form for B-spline surface is:

$$\vec{P}(a, b) = \sum_{i=0}^m \sum_{j=0}^n N_{i,k}(a) N_{j,l}(b) \vec{P}_{i,j} \quad \text{where,}$$

$\vec{P}_{i,j} \rightarrow$ Vertices of Polygon

$N_{i,k}$ and $N_{j,l} \rightarrow$ Blending Functions

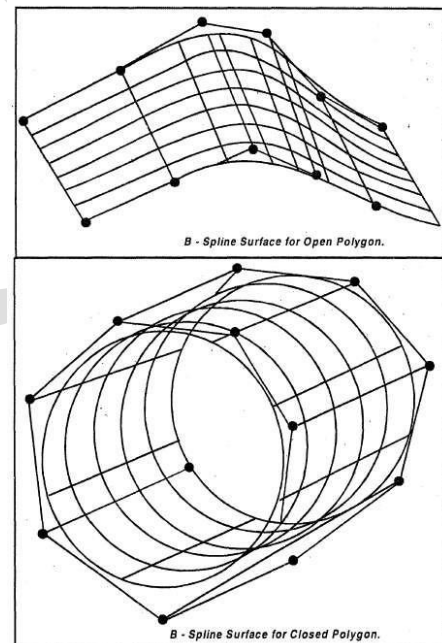
Uniform B-Spline Surfaces

- ✓ The uniform B-Spline surface patch is constructed as a Cartesian product of two uniform B-spline curves.
- ✓ Its four corner points are not the four extreme control points, but

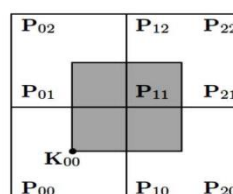
$$K_{00} = P(0, 0) = 1/4 (P_{00} + P_{01} + P_{10} + P_{11}), K_{01} = P(0, 1) = 1/4 (P_{01} + P_{02} + P_{11} + P_{12}),$$

$$K_{10} = P(1, 0) = 1/4 (P_{10} + P_{11} + P_{20} + P_{21}), K_{11} = P(1, 1) = 1/4 (P_{11} + P_{12} + P_{21} + P_{22}).$$

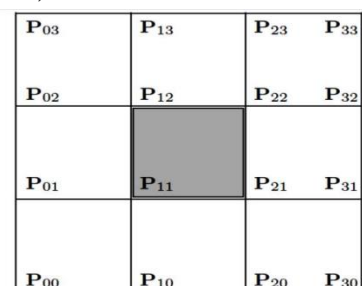
- ✓ This point is therefore located midway between points $(P_{00}+P_{01})/2$ and $(P_{10}+P_{11})/2$. Figure shows this location, as well as the locations of the other three corner points, for the case where the control points are equally spaced.



Idealized B-Spline surface patches



(a)



(b)

Curves –Introduction

- ✓ The most mathematically straight forward geometric entities are curves, which are an integral part of any design.
- ✓ It is necessary that an engineer has to draw one or other types of curves or curved surfaces which find its application in the automotive, aerospace and hydrospace industries.

REPRESENTATION OF CURVES

- ✓ Curves are one of the most essential objects to create high-resolution graphics.
- ✓ While using many small polylines allows creating graphics that appear smooth at fixed resolutions, they do not preserve smoothness when scaled and also require a tremendous amount of storage for any high-resolution image.
- ✓ Curves can be stored much easier, can be scaled to any resolution without losing smoothness, and most importantly provide a much easier way to specify real-world objects.
- ✓ All of the popular curves used in graphics are specified by parametric equations.
- ✓ Instead of specifying a function of the form $y = f(x)$, the equations $y = f_y(u)$ and $x = f_x(u)$ are used.
- ✓ Using parametric equations allows curves that can double back and cross themselves, which are impossible to specify in a single equation in the $y = f(x)$ case.
- ✓ Parametric equations are also easier to evaluate: changing u results in moving a fixed distance along the curve, while in the traditional equation form much work is needed to determine whether to step through x or y , and determining how large a step to take based on the slope.

Role of Curves in Geometric Modeling

- ✓ Curves are used to draw a wireframe model, which consists of points and curves; the curves are utilized to generate surfaces by performing parametric transformations on them.
- ✓ A curve can be as simple as a line or as complex as a B-spline.
- ✓ In general, sweeping a curve along or around an axis creates a surface, and the generated surface will be of the same type as the generating curve, e.g., a fixed form curve will generate a fixed form surface.
- ✓ As stated earlier, curves are used to generate surfaces.
- ✓ To facilitate the computer-language algorithm, curves are represented by parametric equations.
- ✓ Non-parametric equations are used only to locate a point of intersection on the curve, and not for generating them.
- ✓ In general, curves can be classified as follows:
 - Based on the points position
 - Plane Curve (Circle)
 - Space Curve
 - Based on forms
 - Known form Curves/ Analytical Curves

▪ Free form Curves/Special Curves/Synthetic Curves

Known form Curves/ Analytical Curves

- ✓ This type of curve can be represented by a simple mathematical equation, such as, a circle or an ellipse.
- ✓ They have a fixed form and cannot be modified to achieve a shape that violates the mathematical equations.

Advantages

- ✓ Evaluations of the intermediate points are precise and easy. Space taken for the storage of curve is compact.
- ✓ Evaluation of the curve properties such as slope and radius of curvature are easy.
- ✓ Drawing of curve is easy from the storage data.
- ✓ Any change in the curve can be easily modified in the design criteria.

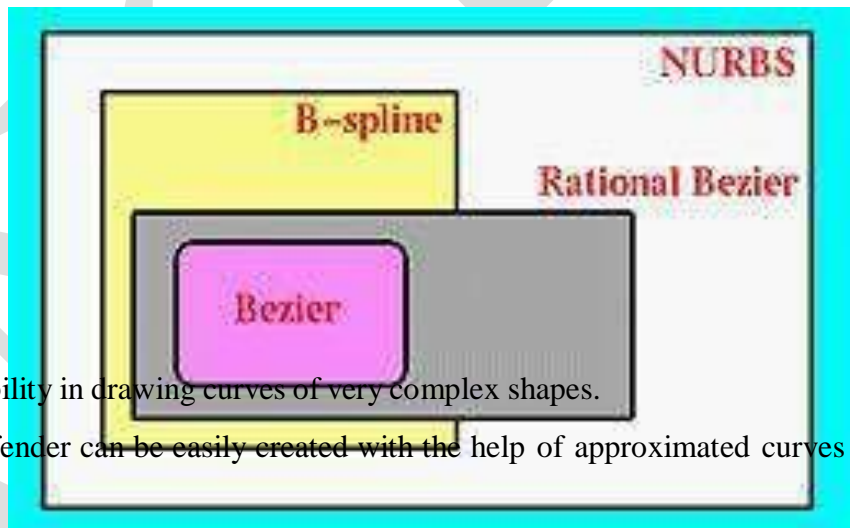
Free form Curves/Special Curves/Synthetic Curves

i) Interpolated curves (Hermite Curve, LeGrange)

- ✓ An interpolated curve is drawn by interpolating the given data points and has a fixed form, dictated by the given data points. These curves have some limited flexibility in shape creation, dictated by the data points.

ii) Approximated Curves (B-Spline, Bezier, NURBS)

- ✓ These curves provide the most flexibility in drawing curves of very complex shapes.
- ✓ The model of a curved automobile fender can be easily created with the help of approximated curves and surfaces.



A system that supports users to design curves must be:

1. Intuitive: We expect that every step.
2. Flexible: The system should provide the users with more control for designing and editing the shape of a curve.
3. Easy: The way of creating and editing a curve should be easy.
4. Unified Approach: The way of representing, creating and editing different types of curves (e.g., lines, conic sections and cubic curves) must be the same.
5. Invariant: The represented curve will not change its geometry under geometric transformation (translation, rotation, ...)

Bézier, B-spline and NURBS curves advantage:

1. User layouts a set of control points for the system.

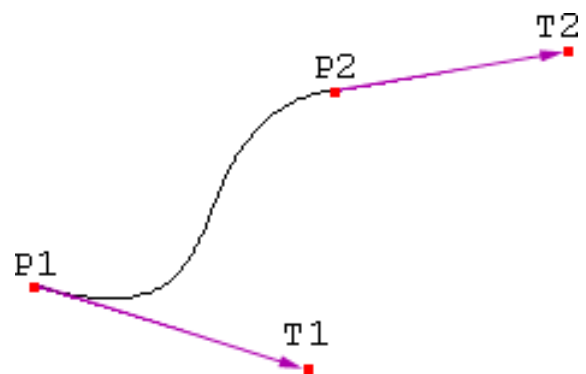
2. A user can change the positions of some control points and some other characteristics for modifying the shape of curve.
3. If necessary, a user can add control points.
4. They are very geometric
5. The transition from curve to surface will not cause much difficulty.

Limitations

1. The transformations of the curve becomes difficult to handle when there are large number of data points involved.
2. The interpolation techniques are used to obtain intermediate points, which do not actually lie on the curve.
3. Numerical differentiation method is used to obtain the slope of the curve, which is an inaccurate procedure.
4. Exact property of the curve cannot be obtained because of the irregularity in the shape.
5. A good quality circle requires a minimum of 32 points on its circumference; therefore, a huge storage is required as compared to the analytical representation of circle in which center and radius is sufficient to represent the circle.

Hermite Curves

- ✓ Hermite curve is geometrically determined by defining positions and tangents of end points.
- ✓ Hermite curve is also known as cubic curve.
- ✓ Hermite curve passes (interpolate) through the end points of the curve segment and possesses first order (slope) continuity.
- ✓ Hermite cubic, curve is an interpolating piecewise cubic polynomials.
- ✓ Unlike the natural cubic splines, Hermite curve segments are adjusted locally because shape of each curve segment is dependent only upon its endpoint constraints.
- ✓ 3D curve of polynomial bases
- ✓ In CG, often used as the trace for camera with Frenet frame, or rotation- minimizing frame



$$h1(s) = 2s^3 - 3s^2 + 1$$

$$h2(s) = -2s^3 + 3s^2$$

$$h3(s) = s^3 - 2s^2 + s$$

$$h4(s) = s^3 - s^2$$

$$P(s) = P1h1(s) + P2h2(s) + T1h3(s) + T2h4(s) \quad P'(s) = P1h1'(s)$$

$$+ P_2 h_2'(s) + T_1 h_3'(s) + T_2 h_4'(s)$$

$$h_1' = 6s^2 - 6s \quad h_2' = -6s^2 + 6s \quad h_3' = 3s^2 - 4s + 1 \quad h_4' = 3s^2 - 2s \quad P(0) = P_1,$$

$$P(1) = P_2;$$

$$P'(0) = T_1, \quad P'(1) = T_2$$

✓ The parametric equation of a cubic spline segment is given by,

$$P(S) = \sum_{i=0}^3 C_i S^i, \quad 0 \leq S \leq 1$$

Here,

$S \rightarrow$ parameter

$C_i \rightarrow$ the polynomial coefficient

In scalar form, the equation (2.1) can be written as,

$$\left. \begin{aligned} x(S) &= C_{3x} S^3 + C_{2x} S^2 + C_{1x} S + C_{0x} \\ y(S) &= C_{3y} S^3 + C_{2y} S^2 + C_{1y} S + C_{0y} \\ z(S) &= C_{3z} S^3 + C_{2z} S^2 + C_{1z} S + C_{0z} \end{aligned} \right\} \dots(2.2)$$

Equation (2.1) can be written in the expanded form as,

$$P(S) = C_3 S^3 + C_2 S^2 + C_1 S + C_0 \quad \dots(2.3)$$

The equation (2.3) can also be written in matrix form as,

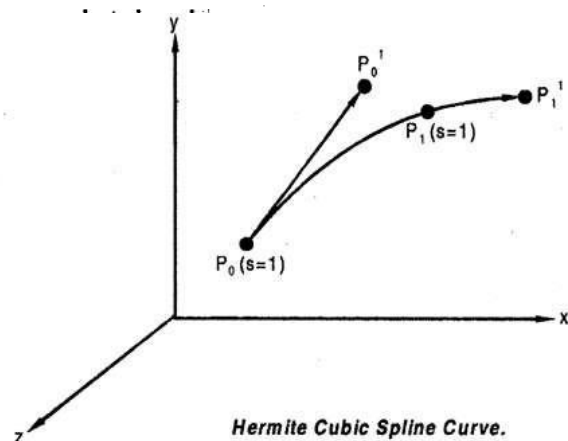
$$P(S) = S^T C \quad \dots(2.4)$$

where

$$S = (S^3 \ S^2 \ S)^T$$

$$P'(S) = \sum_{i=0}^3 C_i i S^{i-1}, \quad 0 \leq S \leq 1 \quad \dots(2.5)$$

The tangent vector of the curve at P_0 and P_1 is T_0 and T_1 respectively. To find the coefficients C_i consider the cubic spline curve with the two end points P_0 and P_1 as shown in Fig.



Hermite Cubic Spline Curve.

Applying the boundary conditions (P_0, P_0' at $S = 0$ and P_1, P_1' at $S = 1$) in equation (2.1) and (2.5) gives,

$$P_0 = C_0$$

$$P_0' = C_1$$

$$P_1 = C_3 + C_2 + C_1 + C_0 \quad \dots(2.6)$$

$$P_1' = 3C_3 + 2C_2 + C_1$$

Substituting equations (2.6) into (2.3) and rearranging the solution we get,

$$P(S) = (2S^3 - 3S^2 + 1)P_0 + (-2S^3 + 3S^2)P_1 \\ + (S^3 - 2S^2 + S)P_0' + (S^3 - S^2)P_1', \quad 0 \leq S \leq 1 \quad \dots(2.7)$$

where,

P_0, P_1, P_0' and $P_1' \rightarrow$ Geometric coefficients

The tangent vector becomes

$$P'(S) = (6S^2 - 6S)P_0 + (-6S^2 + 6S)P_1 + (3S^2 - 4S + 1)P_0' + (3S^2 - 2S)P_1', \quad 0 \leq S \leq 1 \quad \dots(2.8)$$

The function of S in equation (2.7) and (2.8) are called blending functions.

The first two functions blend P_0 and P_1 and the second two blend P_0' and P_1' to produce the left hand side in each equation.

Writing equation (2.7) in matrix form, we get,

$$P(S) = S^T [M_H] V, \quad 0 \leq S \leq 1 \quad \dots(2.9)$$

where,

$M_H \rightarrow$ Hermite matrix

$V \rightarrow$ Geometry matrix

Both of the coordinates are given by,

$$[M_H] = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \dots(2.10)$$

$$V = [P_0 P_1 P_0' P_1']^T$$

Shape control of Hermite curve:

There are three ways to modify/control the shape of Hermite (cubic) curves, they are:

- (i) Change in the location of the control points.
- (ii) Change in the magnitudes of the tangent vectors, keeping the same directions.
- (iii) Change in the directions of the tangent vectors, keeping the same magnitudes.

Limitations of Hermite curve:

- (i) It is always difficult to select the magnitude as well as angle of the tangent vectors at the two end points of the curve segment.
- (ii) Curves are difficult to control because of the global shape control characteristics.
- (iii) The cubic curves never reduce exactly to a conic section and poorly approximate asymptotic curves.

(iv) The computation time required to invert the tangent vector matrix can be excessive when the number of control points are large.

(v) A curve will be more flexible if more number of control points, could be added, thus creating more curves which are all still of cubic order.

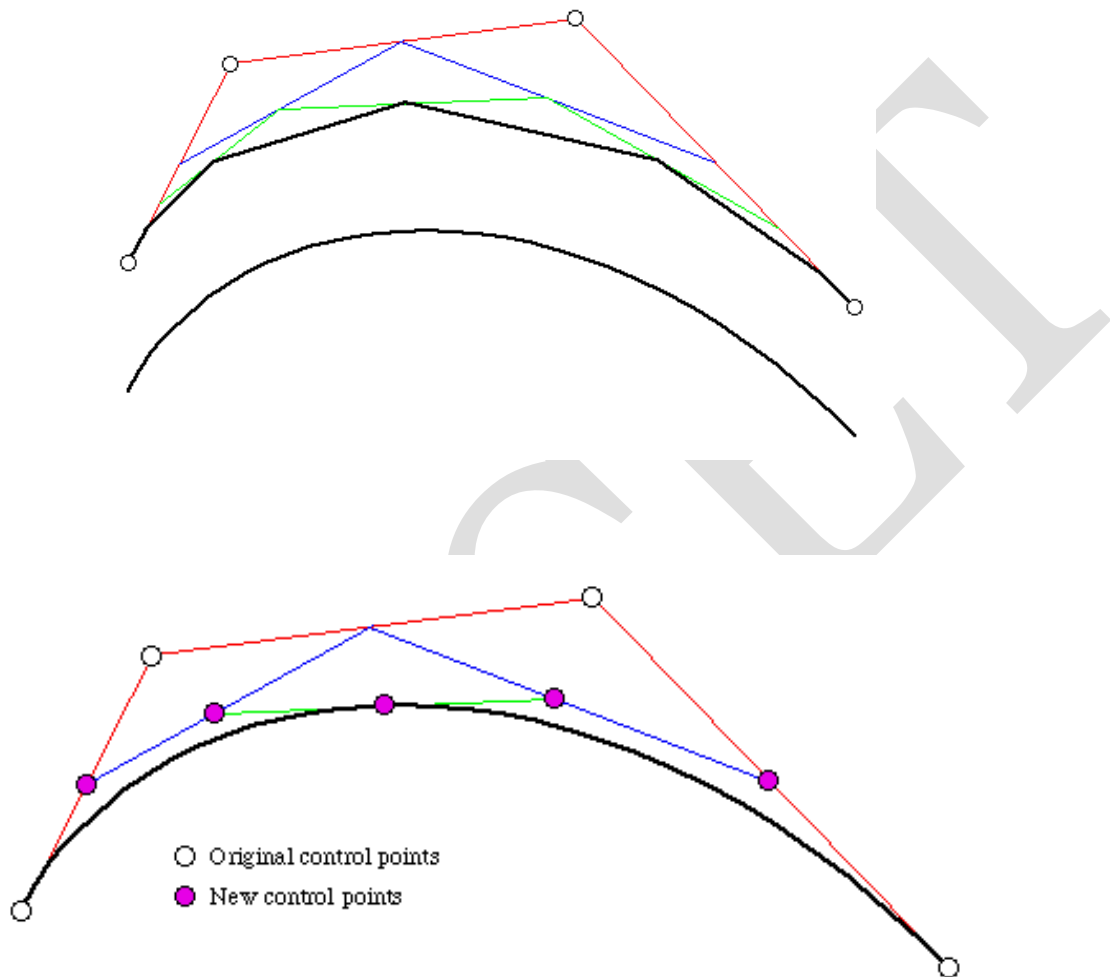
- ✓ Hermite curves are based on the interpolation technique which means the curve passes through the existing datapoints.
- ✓ This technique is suitable when the basic shape of the curve is obtained by experimental methods, or through the mathematical calculation.
- ✓ Bezier curves are another alternative to generate curves based on the approximation techniques, and to produce the curves that do not pass through the given data pointer.

Bezier curves

- ✓ A Bezier curve is defined by a set of data points. The curve may interpolate or extrapolate the data points.
- ✓ Some CAD systems offer both options; others offer the interpolation version only. In both cases, the data points are used to control the shape of the resulting curves.
- ✓ Bezier curves are another very popular curve type developed by P. Bezier of the French car firm Regie Renault.
- ✓ The nomenclature of a cubic (4 control points).
- ✓ In the Bezier curve the following observations are made
 1. The shape of the Bezier curve is obtained or determined by the number of control points.
 2. Generally the Bezier curve approximates the data points and interpolates the end points.
 3. The position of the control points and vertices of Bezier characteristics polygon is used to define the Bezier curve.
 4. The Bezier curve follows the shape of characteristics polygon.
 5. The Bezier curve is tangent to the first and last polygon segment of the characteristic polygon.
 6. The shape of the Bezier curve is defined uniquely by the characteristic polygon.
- ✓ Bezier curves, unlike B-splines, must have precisely n control points, where $n + 1$ is the degree of the Bezier polynomial.
- ✓ To create a longer curve, it is necessary to connect multiple Bezier curves.
- ✓ This is usually done by making the last control point of one curve the same as the first control point of the next curve.
- ✓ Bezier curves pass through the first and last control points of each curve segment, however, which makes them quite easy to work with and popular for use in interactive design programs.
- ✓ Bezier curves, like B-Spline curves, always lie within the convex hull of the control points, and always have the sum of the basis functions add to 1.
- ✓ The idea behind Bezier curves is quite simple. To create a smooth curve, one intuitive mechanism is

to first connect the four control points with lines.

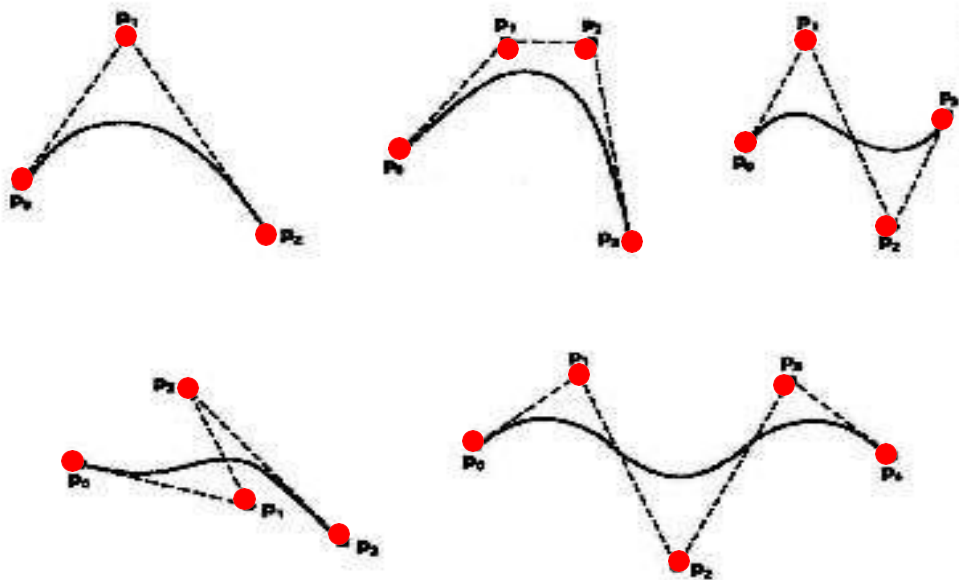
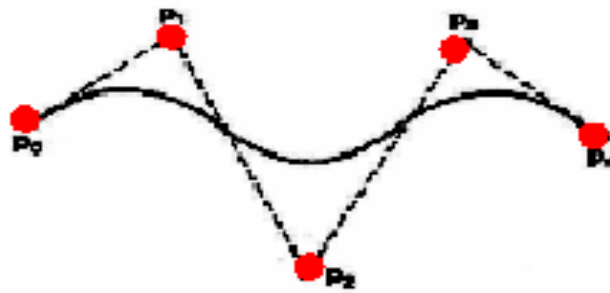
- ✓ Then draw new lines connecting the midpoints of those lines, and more new lines to connect the midpoints of the new lines, and so forth until the resulting curve is sufficiently smooth. This is illustrated below.



- ✓ These Bézier curves can then be broken into smaller Bézier curves, and so on, until the control points of the Bézier curve are very close to a straight line.
- ✓ When this happens, a straight line is drawn between the first and last control points, and the recursion pops back up.
- ✓ While it is possible to define a curve type based on connecting the last third of one segment to the first third of the next, or the last fourth of one segment to the first fourth of the next, and so on, using midpoints allows for a very efficient subdivision algorithm to be derived.
- ✓ Spline approximation method;
- ✓ Useful and convenient for curve and surface design; Easy to implement;
- ✓ Available in Cad system, graphic package, drawing and painting packages.
- ✓ In general, a Bézier curve section can be fitted to any number of control points.

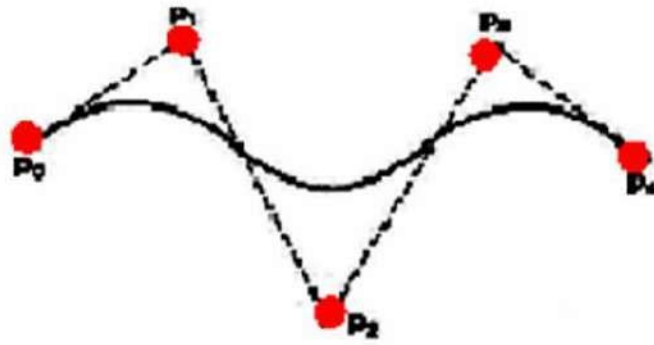
- ✓ The number of control points to be approximated and their relative position determine the degree of the Bézier polynomial.
- ✓ Given $n+1$ control point positions:
- ✓ These coordinate points can be blended to produce the following position vector $C(u)$, which describes the path of an approximating Bézier polynomial function between P_0 and P_n .

$$C(u) = \sum_{k=0}^n p_k B_{k,n}(u), \quad 0 \leq u \leq 1$$

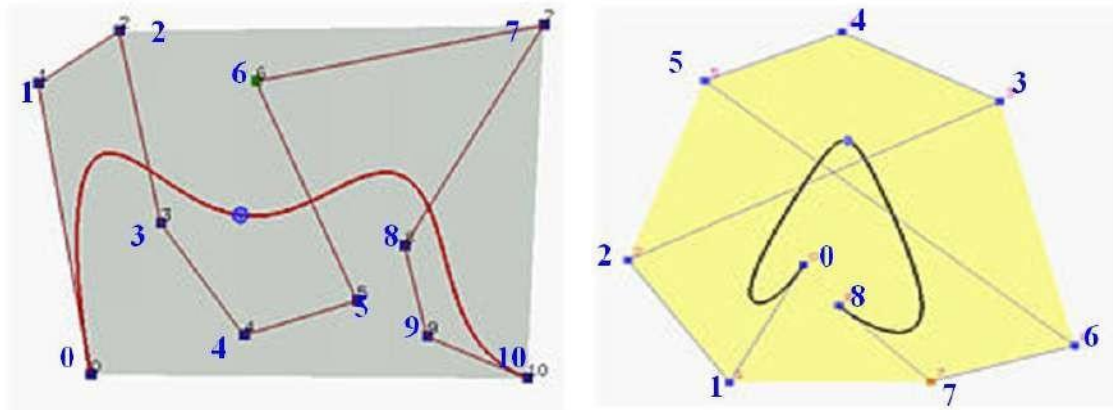


Properties of Bézier Curves.

1. The degree of a Bézier curve defined by $n+1$ control points is n :
2. The curve passes through the first and the last control point $C(u)$ passes through P_0 and P_n

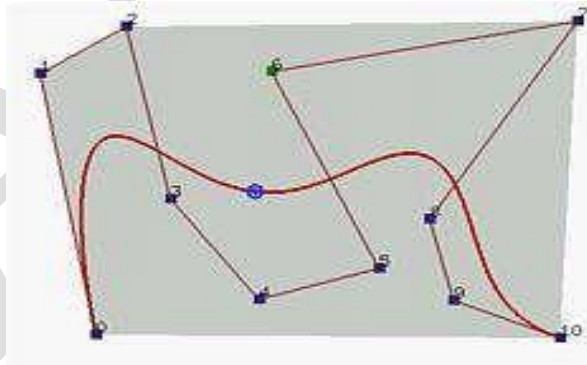


3. Bézier curves are tangent to their first and last edges of control polyline



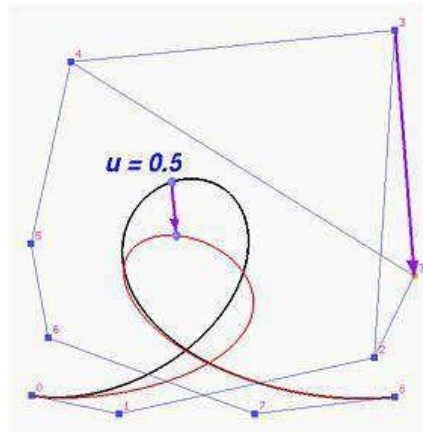
The Bézier curve lies completely in the convex hull of the given control points.

Note that not all control points are on the boundary of the convex hull. For example, control

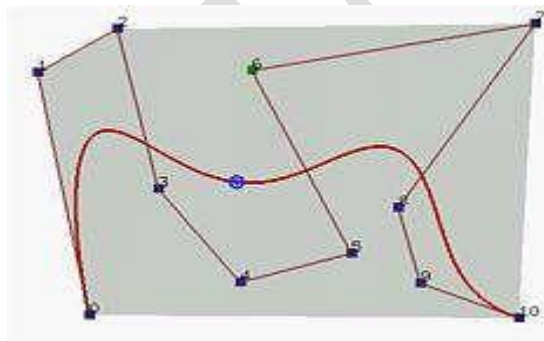
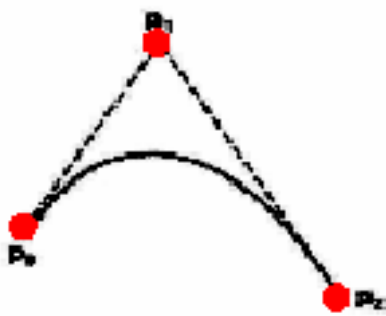


points 3, 4, 5, 6, 8 and 9 are in the interior. The curve, except for the first two endpoints, lies completely in the convex hull.

4. Moving control points:

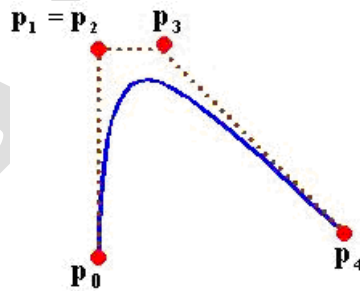


5. The point that corresponds to u on the Bézier curve is the "weighted" average of all control points, where the weights are the coefficients $B_{k,n}(u)$.

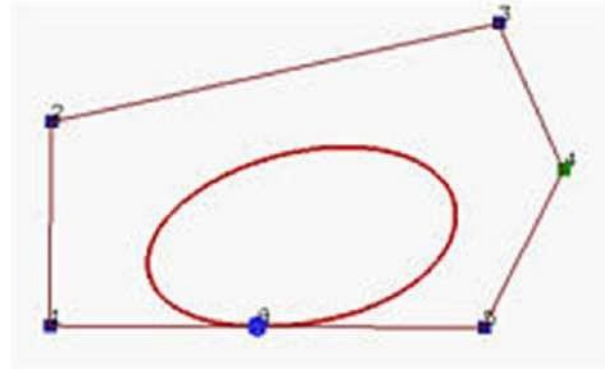
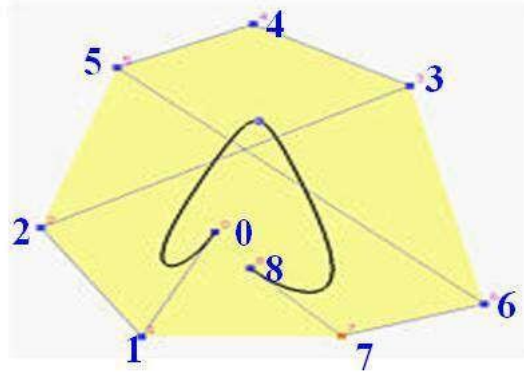


$$C(u) = \sum_{k=0}^n \mathbf{p}_k B_{k,n}(u), \quad 0 \leq u \leq 1$$

6. Multiple control points at a single coordinate position gives more weight to that position.



7. Closed Bézier curves are generated by specifying the first and the last control points at the same position.



Note: Bézier curves are polynomials which cannot represent circles and ellipses.

8. If an affine transformation is applied to a Bézier curve, the result can be constructed from the affine images of its control points.

✓ Given $n+1$ control point positions:

$$P_k \quad (x_k, y_k, z_k) = \quad 0 \leq k \leq n$$

$$C(u) = \sum_{k=0}^n p_k B_{k,n}(u), \quad 0 \leq u \leq 1$$

✓ The Bézier blending functions are the Bernstein polynomials:

$$B_{k,n}(u) = C(n, k) u^k (1-u)^{n-k}$$

✓ Disadvantages of Bezier curves

- (i) The curve does not pass through the control points which may be inconvenient to the designer.
- (ii) The Bezier curve lacks local control. It only has the global control nature.
- (iii) If one control point is changed, the whole curve changes. Thus the designer cannot selectively change parts of the curve.

Difference between the Bezier curve and Hermite curve

1. Its defining points control the shape of the Bezier curve. It allows relationship between the input and output parameters.
2. Hermite curve is cubic in nature so the degree of polynomial are not related to the number of control points. Where as the Degree of polynomial for the Bezier curve is variable and related to the number of control points defining the curve.
3. Bezier curve is smoother than Hermite curve because it has higher order derivatives.

B-SPLINE CURVES

- ✓ B - Spline curves give another effective method, besides that of Bezier curves.
- ✓ It has the ability to interpolate or approximate a set of given data points. Moreover B - spline curves are the proper and powerful generalization of Bezier curves.
- ✓ In addition to Bezier curves, the B -spline curve separates the degree of the resulting curve from the number of the given control points.
- ✓ The flexibility in the degree of the resulting curve is achieved by choosing the blending function of B - spline curves with an additional degree of freedom that does not exist in Bernstein polynomials.
- ✓ Finally, the final curve always lies within the convex hull of the control points.

Similar to E $N_{i,k}(S)$ where, $N_{i,k}(S)$ is defined by

$N_{i,k}(S) \rightarrow$ B - spline function

The B - spline functions have the following properties.

Partition of unity \rightarrow Ensures the relationship between the curve and its defining control points is invariant under transformation.

$$\sum_{i=0}^n N_{i,k}(S) = 1$$

Positivity: \rightarrow This property ensures that the curve segment lies completely within the convex hull of P_i .

$$N_{i,k}(S) = 0$$

Local support: \rightarrow indicates that each segment of a B - spline curve is influenced by only k control points affect any k curve segments.

$$N_{i,k}(S) = 0 \text{ if } S \notin [S_i, S_{i+k+1}]$$

Continuity:

$N_{i,k}(S)$ is $(k-2)$ time continuously differentiable.

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

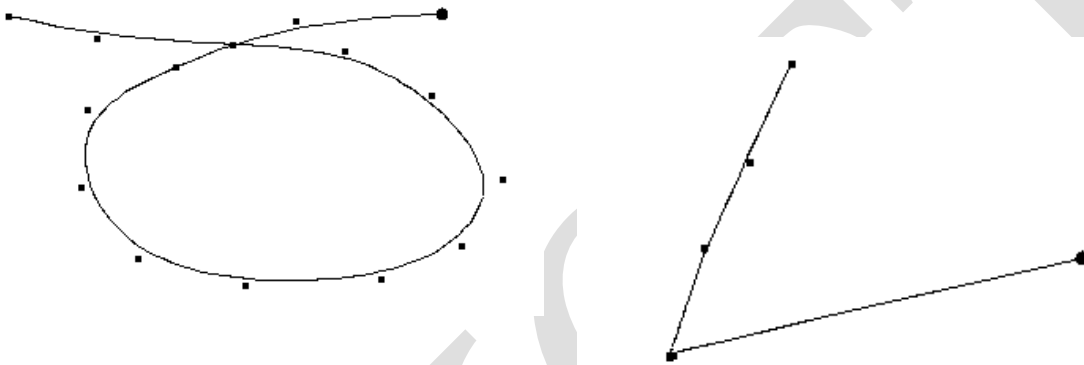
$$N_{i,k}(S) = (S - S_1) \frac{N_{i,k-1}(S)}{S_{i+k-1} - S_i} + (S_{i+k} - S) \frac{N_{i+1,k-1}(S)}{S_{i+k} - S_{i+1}}$$

where,

$$N_{i,1} = \begin{cases} 1, & S_i \leq S \leq S_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

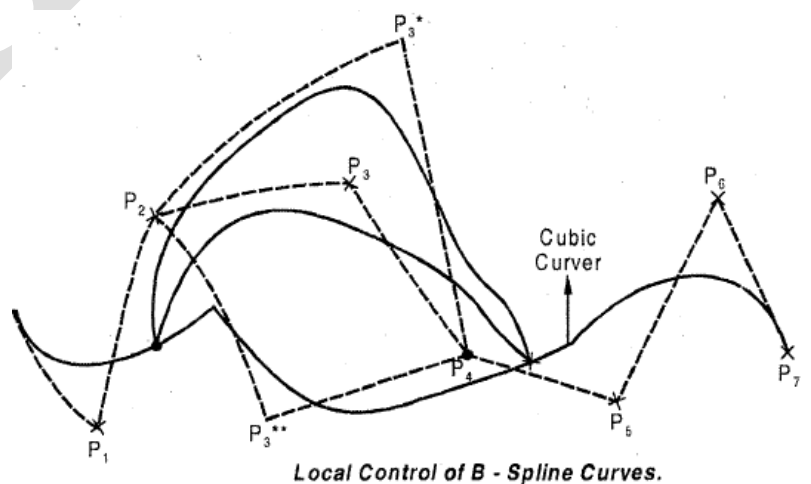
Note: The parameter k controls the degree ($k-1$)

- ✓ While B-splines can be of higher order, usually cubic B-splines are used. Cubic functions allow for enough curve flexibility for most tasks, and are easier to work with than higher order functions since they behave more predictably.
- ✓ The cubic B-Spline blending functions work over four control points. The blending functions appear below.
- ✓ Note that special blending functions are needed for the first and last two sections of the B-Spline so that it passes through the first and last points.



- ✓ B-Splines are designed to eliminate sharp corners, as seen above. A sharp corner can be obtained, however, by using duplicate sequential control points.
- ✓ The characteristics of B - spline-curves that are useful in design can be summarized as follows:

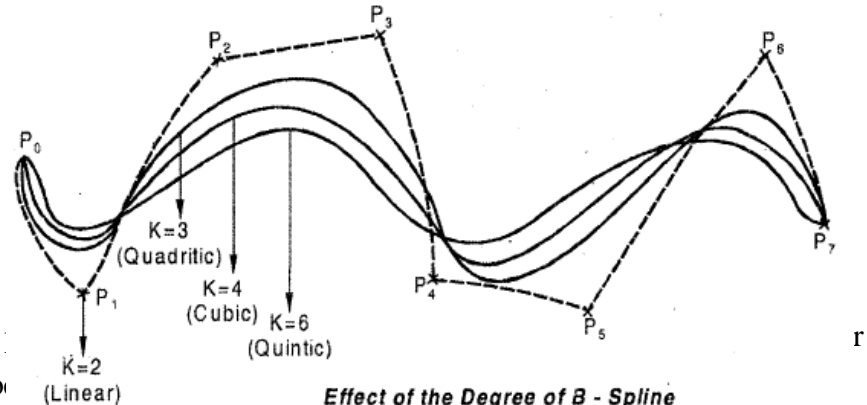
1. The local control of the curve can be achieved by changing the position of control points, using multiple control points by placing several points at the same location, or by choosing a different degree ($k - 1$). Changing one control point affects only k segments. The local control for a cubic B - spline curve by moving P_3 to P_3^* and P_3^{**} is shown in the below Figure. Only the four curve segments surrounding P_3 change.



2. A non periodic B-spline curve passes through the first and last control points P_0 and P_{n+1} and it is tangent to the first $(P_1 - P_0)$ and last $(P_{n-1} - P_n)$ segment of control polygon, similar to the Bezier curve, as shown in the above Figure.

3. Generally the lesser the degree, the closer the curve gets to the control points as shown in the below Figure. When $k = 1$ a zero-degree curve results. The curve then becomes the control points themselves. When $k = 2$ the curve becomes the polygon segments themselves.

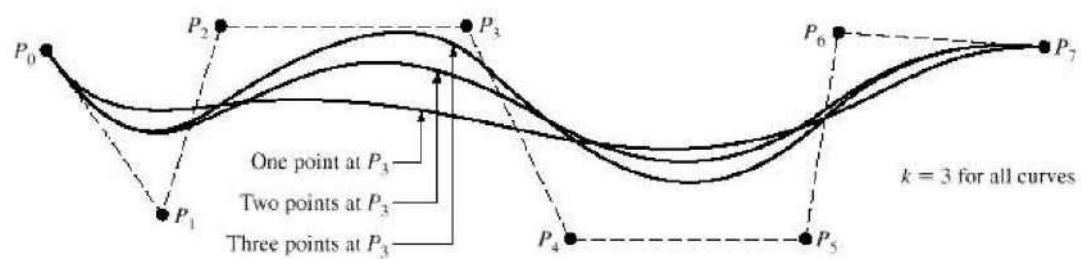
4. The second-degree curve is always tangent to the mid points of all the internal polygon segments as shown in the above Figure. This is not the case for other degrees.



5. If K equals the number of control curve. In this case the range of 'u' b

6. Multiple control points induce the regions of high curvature of a B-spline curve, which is the curve is pulled more toward a control point by increasing its multiplicity. This multiplicity is useful for creating sharp corners in the curve.

7. Increasing the degree of the curve makes it more difficult to control and to calculate accurately.



Effect of point multiplicity of B-spline curve on its shape.

Therefore, a cubic B-spline is sufficient for a large number of applications.

Open or non-periodic B-spline curves have been discussed. The same theory can be extended to cover closed or periodic B-spline curves. The only difference between open and closed curves is in the choice of the knots and the basic functions.

Advantages of B - spline curve

- ✓ They provide local control of the curve shape as opposed to global control by using a special set of blending (base) functions that provide local influence.
- ✓ B-Spline curves also provide the ability to add control points without increasing the degree of the curve.

RATIONAL CURVES

- ✓ Rational curve is defined by the algebraic ratio of two polynomials while a nomenclature curve is defined

by one polynomial.

- ✓ A parametric curve in homogeneous form is referred to as a rational curve. These curves draw their theories from projective geometry.
- ✓ Rational Bezier curves, rational B-spline and βspline curves, rational ionic sections, rational cubics and rational surfaces have been formulated.
- ✓ The most widely used rational curves are non uniform rational B-splines (NURBS).
- ✓ The formulation of rational curves requires the introduction of homogeneous space and the homogeneous coordinates.
- ✓ The homogeneous space is A - D space.
- ✓ A rational B-spline curve defined by $n + 1$ control points P_i is given by,

$$P(S) = \sum_{i=0}^n P_i R_i k(S), \quad 0 \leq S \leq S_{\max}$$

$R_i, k(S)$ are the rational B - spline basis function are given by

$$R_i, k(S) = \frac{h_i N_i, k(u)}{\sum_{i=0}^n h_i N_i, k(S)}$$

- ✓ The rational basis function $R_i, k(s)$ have nearly all the analysis and geometric characteristics of their non rational B - spline counter parts.
- ✓ A rational B - spline is considered as unified representation that can define a variety of curves and surfaces.
- ✓ This curve can represent all wireframe, surface and solid entities.
- ✓ This permits unification and conversion from one modeling technique to another. Consider a circular cylinder (hole) represented by a B - spline, some data on the specific curve type may be lost unless it is carried along.
- ✓ Data including the fact that the part feature is a cylinder - would be useful for manufacturing to identify it as a hole to be drilled or bored rather than a surface to be milled.

Non-uniform rational B-spline

- ✓ NURBS (non uniform rational B-splines) are almost exclusively used by CAD/CAM systems as the internal representation of all geometric entities that these systems provide.
- ✓ It is a mathematical model commonly used in computer graphics for generating and representing curves and surfaces.

- ✓ It offers great flexibility and precision for handling both analytic and synthetic shapes.
- ✓ NURBS provide a unified approach to formulate and represent curves and surfaces.
- ✓ NURBS provide a convenient design tool to create smooth curves and surfaces interactively.
- ✓ The development of NURBS dates back to the late 1970s when Boeing developed the Tiger CAD system based on rational B-splines.
- ✓ These splines were developed by integrating the representations of B-splines with rational Bezier curves.

NURBS have many advantages

- ✓ Any curve or surface can be formulated using NURBS.
- ✓ NURBS are considered a unified canonical representation that can define both synthetic (Bezier, B-splines) and analytic (circles and conics) curves and surfaces.
- ✓ The premise is that they can represent all curve, surface, and solid entities, allowing unification and conversion from one CAD system to another via exchange standards (STEP and IGES).
- ✓ Their related algorithms are stable and accurate.
- ✓ They enhance manufacturing and machining accuracy and speed.
- ✓ They are intuitive and flexible to use in design and geometric modeling.
- ✓ NURBS are also invariant under affine and perspective transformations such as translation, rotation, and projections.
- ✓ They reduce the memory consumption when storing shapes (compared to simpler methods).
- ✓ They can be evaluated reasonably quickly by numerically stable and accurate algorithms.

NURBS have some problems

- ✓ The definition of simple curves such as arcs, circles, and conics is verbose: they require more data to define as NURBS than the traditional way. For example, the traditional definition of a circle is a center (x, y, z) , radius and the circle plane defined by a normal vector (n_x, n_y, n_z) that is a total of seven numbers.
- ✓ The NURBS definition requires 38 numbers. The loss of information on simple shapes is another problem. For example, if a circular cylinder (hole) is represented by a B-Spline, some data on the specific curve type may be lost unless it is carried along. Data, including that the part feature was a cylinder, would be useful in manufacturing to identify it as a hole to be drilled or bored rather than a surface to be milled.
- ✓ The improper use of the extra flexibility that NURBS offer (such as weights) can produce ill-behaved NURBS.
- ✓ Moreover, some algorithms, such as surface/surface intersection and inverse-point solution, work better under the non-NURBS representation.

Hidden – Line-Surface-Solid removal algorithms – shading – colouring – computer animation.

VISUAL REALISM - INTRODUCTION

What is Visual Realism?

- ✓ Visual Realism is a method for interpreting picture data fed into a computer and for creating pictures from difficult multidimensional data sets.
- ✓ Visualization can be classified as :
 - Visualization in geometric modeling
 - Visualization in scientific computing.
- ✓ Visualization in geometric modeling is helpful in finding connection in the design applications.
- ✓ By shading the parts with various shadows, colors and transparency, the designer can recognize undesired unknowninterferences.
- ✓ In the design of complex surfaces shading with different texture characteristics can use to find any undesired quick modifications in surface changes.
- ✓ Visualization in computing is viewed as a technique of geometric modeling. It changes the data in numerical form into picture display, allowing users to view their simulations and computations.
- ✓ Visualization offers a process of seeing the hidden.
- ✓ Visualization in scientific computing is of great interest to engineers during the design process.
- ✓ Existing visualization methods are:
 - Parallel projections
 - Perspective projection.
 - Hidden line removal
 - Hidden surface removal
 - Hidden solid removal
 - Shaded models
- ✓ Hidden line and surface removal methods remove the uncertainty of the displays of 3D models and is accepted the first step towards visual realism.
- ✓ Shaded images can only be created for surface and solid models.
- ✓ In multiple step shading process, the first step is removing the hidden surfaces / solids and second step is shades the visible area only.
- ✓ Shaded images provide the maximum level of visualization.
- ✓ The processes of hidden removal need huge amounts of computing times and also upper end hardware services.
- ✓ The creation and maintenance of such a models are become complex.
- ✓ Hence, creating real time images needs higher end computers with the shading algorithms embedded into the hardware.

Visibility Techniques

- ✓ In general these techniques attempt to establish relationships among polygons and edges in the viewing plane. The techniques normally check for overlapping of polygons in the viewing plane. If overlapping occurs, depth comparisons are performed.
- Minimax Test
- Containment Test
- Surface Test/back face/depth test
 - Computing Silhouettes
 - Edge Intersections
- Segment (Scanline) Comparisons
 - Homogeneity Test
- ✓ Object-space algorithms are more accurate than image-space algorithms.
- ✓ The former perform geometric calculations using the floating-point precision of the computer hardware, while the latter perform calculations with accuracy equal to the resolution of the display screen used to render the image.
- ✓ The enlargement of an object-space image does not degrade its quality of display as does the enlargement of an image-space image.
- ✓ As the complexity of the scene increases (large number of objects in the scene), the computation time grows more quickly for object-space algorithms than for image-space algorithms.

Sorting

- ✓ Sorting is an operation that orders a given set of records according to a selected criterion.
- ✓ The time required to perform the sort depends on the number of records to be processed, the algorithm that performs the sort and the properties of the initial ordering of the records (whether it is random or semi ordered).
- ✓ Many visibility algorithms (hidden-line, hidden-surface, and hidden-solid algorithms) make extensive use of sorting operations.
- ✓ Sorting and searching operate on the records of the scene data base. These records typically contain geometrical, topological and viewing information about the polygons and faces that make the scene.

Coherence

- ✓ Naturally, the elements of a scene or its image have some interrelationships, known as coherence.
- ✓ Hidden-line algorithms that utilize coherence in their sorting techniques are more effective than other algorithms that do not.
- ✓ Coherence is a measure of how rapidly a scene or its image changes.
- ✓ It describes the extent to which a scene or its image is locally constant.

- ✓ The coherence of a set of data can improve the speed of its sorting significantly.
- ✓ The gradual changes in the appearance of a scene or its image from one place to another can reduce the number of sorting operations greatly.
- ✓ Several types of coherence can be identified in both the object space and the image space.
- ✓ Edge coherence: The visibility of an edge changes only when it crosses another edge.
- ✓ Face coherence: If a part of a face is visible, the entire face is probably visible. Moreover, the penetration of faces is a relatively rare occurrence, and therefore it is not usually checked by hidden-removal algorithms.
- ✓ Geometric coherence: Edges that share the same vertex or faces that share the same edges have similar visibilities in most cases.
- ✓ For example, if three edges (the case of a box) share the same vertex, they may be all visible, all invisible, or two visible and one invisible. The proper combination depends on the angle between any two edges (less or greater than 180°) and on the location of any of the edges relative to the plane defined by the other two edges.
- ✓ Frame coherence: A picture does not change very much from frame to frame.
- ✓ Scanline coherence: Segments of a scene visible on one scan line are most probably visible on the next line.
- ✓ Area coherence: A particular element (area) of an image and its neighbors are all likely to have the same visibility and to be influenced by the same face.
- ✓ Depth coherence: The different surfaces at a given screen location are generally well separated in depth relative to the depth range of each.
- ✓ The first three types of coherence are object space based, while the last four are image space based.
- ✓ If an image exhibits a particular predominant coherence, the coherence would form the basis of the related hidden-line removal algorithm.

HIDDEN-LINE REMOVAL

- ✓ For a given 3D scene, a given viewing point, and a given direction, eliminate from a 2D projection of the scene all parts of edges and faces which the observer cannot see.
- ✓ For orthographic projections, the location of the viewing point is not needed.

The Priority Algorithm

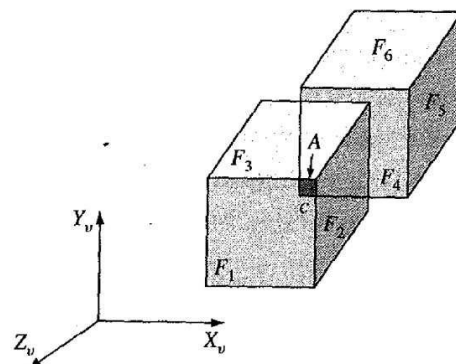
- ✓ This algorithm is also known as the depth or Z algorithm.
- ✓ The algorithm is based on sorting all the faces (polygon) coordinate value of each. This step is sometimes known as sorting.
- ✓ If a face intersects more than one face, other visibility tests are used to resolve any ambiguities.
- ✓ This step constitutes determination of coverings.

Example

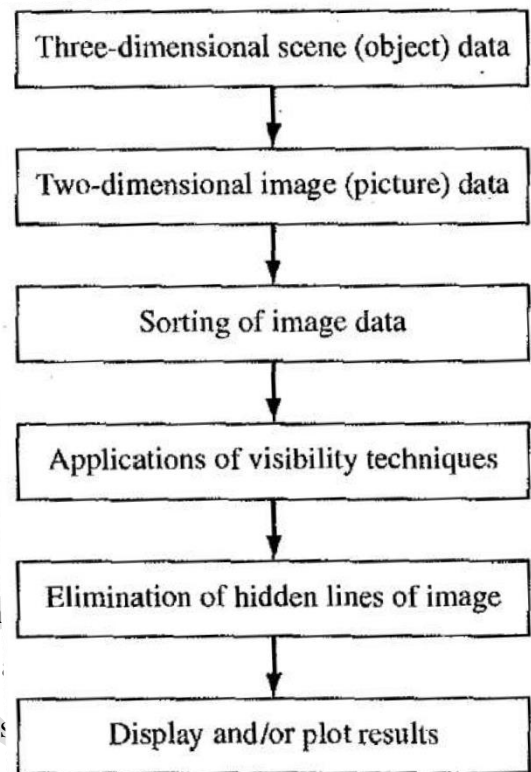
- ✓ Consider a scene of two boxes as shown in Figure. It shows the scene in the standard VCS where the viewing eye is located at infinite distance -on the positive Z_v direction. The following steps provide guidance for implementing the algorithm:

Step 1: Utilize the proper orthographic projection to obtain the desired view (whose hidden lines are to be removed) of the scene.

- ✓ This results in a set of vertices with coordinates of (X_v, Y_v, Z_v) .



(a) Scene of two boxes



Step 2: Utilize the surface test to remove back faces to improve the efficiency of the priority algorithm.

- ✓ To enable one to perform the depth test, the plane equation of any face (polygon) in the image can be obtained using Eq.

- ✓ Given three points that lie in one face, can be written as:

$$z_v = Ax_v + By_v + C$$

where

$$A = \frac{(z_{v1} - z_{v3})(y_{v2} - y_{v3}) - (z_{v2} - z_{v3})(y_{v1} - y_{v3})}{D}$$

$$B = - \frac{(z_{v1} - z_{v3})(x_{v2} - x_{v3}) - (z_{v2} - z_{v3})(x_{v1} - x_{v3})}{D}$$

$$C = z_{v1} - Ax_{v1} - By_{v1}$$

and

$$D = (x_{v1} - x_{v3})(y_{v2} - y_{v3}) - (x_{v2} - x_{v3})(y_{v1} - y_{v3})$$

- ✓ Any two edges of a given face can be used to calculate the face normal. Steps 1 and 2 result in a face list which will be sorted to assign priorities.
 - ✓ For this example, six faces F1 - F6 form such a list.
 - ✓ The order of the faces in the list is immaterial.
- Step 3: Assign priorities to the faces in the face list.
- ✓ The priority assignment is determined by comparing two faces at any one time.
 - ✓ The priority list is continuously changed, and the final list is obtained after few interactions. Here is how priorities can be assigned.
 - ✓ The first face in the face list (F1) is assigned the highest priority 1. F1 is intersected with the other faces in the list, that is, F2- F6.
 - ✓ The intersection between F1 and another face may be an area, A, as in the case of F1 and F4 shown in Figure, an edge as for faces F1 and F2, or an empty set (no intersection) as for faces F1 and F6.
 - ✓ In the case of an area of intersection, the (x_v, y_v) coordinates of a point c inside A can be computed Utilizing the above equation.
 - ✓ For both faces F1 and F4, the two corresponding Z_v values of point c can be calculated and compared.
 - ✓ The face with the highest Z_v values is assigned the highest priority.
 - ✓ In the case of an edge of intersection, both faces are assigned the same priority.
 - ✓ They obviously do not obscure each other, especially after the removal of the back faces.
 - ✓ In the case of no face intersection, no priority is assigned.

Face list	Priority list
F_1	1
F_2	1
F_3	1
F_4	2
F_5	
F_6	

Iteration 1

Face list	Priority list
F_2	1
F_3	1
F_4	2
F_5	
F_6	
F_1	

Iteration 2

Face list	Priority list
F_3	1
F_4	2
F_5	
F_6	
F_1	
F_2	

Iteration 3

Face list	Priority list
F_4	X 2
F_5	X 2
F_6	X 2
F_1	1
F_2	1
F_3	1

Iteration 4

(b) Assignment of priorities

- ✓ In the above example F_1 intersects F_2 and F_3 in edges.
- ✓ Therefore both faces are assigned priority 1. F_1 and F_4 intersect in an area. Using the depth test, and assuming the depth of F_4 is less than that of F_1 .
- ✓ F_4 is assigned priority 2. When we intersect faces F_1 and F_5 , we obtain an empty set, that is, no priority assignment is possible.
- ✓ In this case, the face F_1 is moved to the end of the face list, and the sorting process to determine priority starts all over again.
- ✓ In each iteration, the first face in the face list is assigned priority 1. The end of each iteration is detected by no intersection.
- ✓ Four iterations that yield the final priority list. In iteration 4, faces F_4 to F_6 are assigned the priority 1 first.
- ✓ When F_4 is intersected with F_1 the depth test shows that F_1 has higher priority. Thus, F_1 is assigned priority 1 and the priority of F_4 to F_6 is dropped to 2.

Step 4: Reorder the face and priority lists so that the highest priority is on top of the list.

- ✓ In this case, the face and priority lists are $[F_1, F_2, F_3, F_4, F_5, F_6]$ and $[1, 1, 1, 2, 2, 2]$, respectively.

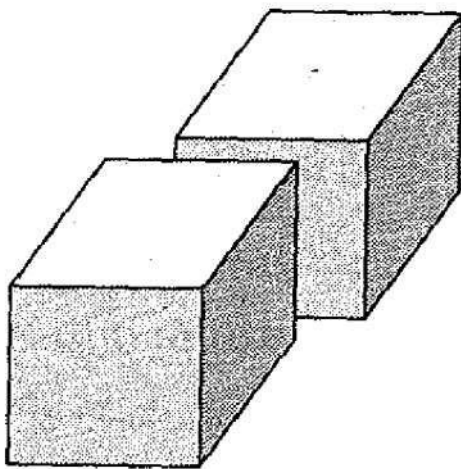
Step 5: In the case of a raster display hidden line removal is done by the hardware.

- ✓ We simply display the faces in reverse order of their priority.
- ✓ Any faces that would have been hidden by the others would thus first be displayed, but would be covered later either partially or entirely by faces of higher priority.

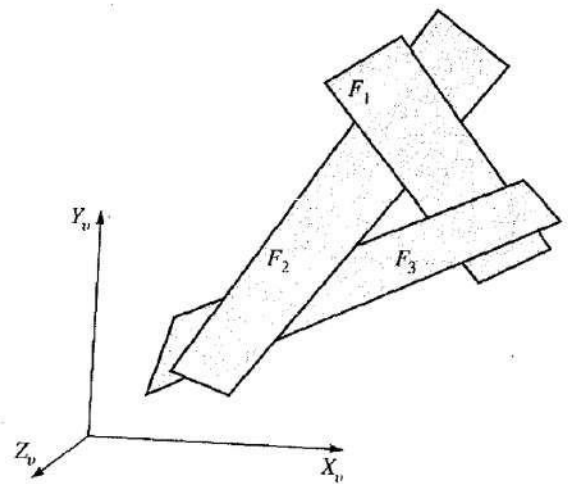
Step 6: In the case of a vector display the hidden line removal must be done by the software by determining coverings.

- ✓ For this purpose, edges of a face are compared with all other edges of higher priority.

- ✓ An edge list can be created which maintains a list of all line segments that will have to be drawn as visible.
- ✓ Visibility techniques such as the containment test and edge intersection are useful in this case.
- ✓ In Some scenes, ambiguities may result after applying priority test. The figure shows a case in which the order of faces is cyclic.
- ✓ Face F1 covers F2, F2 covers F3, and F3 covers F1. It is very difficult to find the priority list that produces this cyclic ordering and coverage.
- ✓ To rectify this ambiguity, additional criteria to determine coverage must be added to the priority algorithm.



(c) Image with hidden lines removed



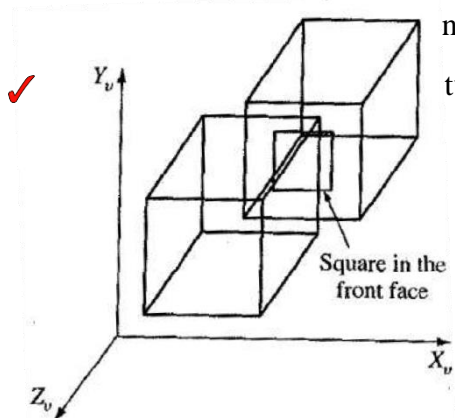
Cyclic faces.

Area Oriented Algorithm

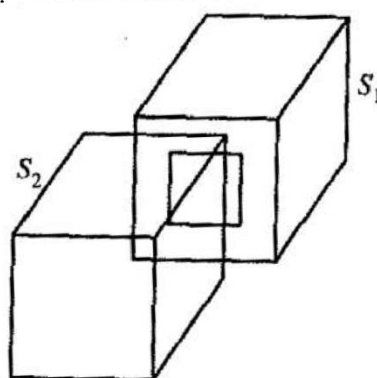
- ✓ The area oriented algorithm described here subdivides the data set of a given scene in a stepwise fashion until all visible area in the scene are determined and displayed.
- ✓ In this algorithm as well as in the priority algorithm, no penetration of faces is allowed.

Step 1. Identify Silhouette polygons:

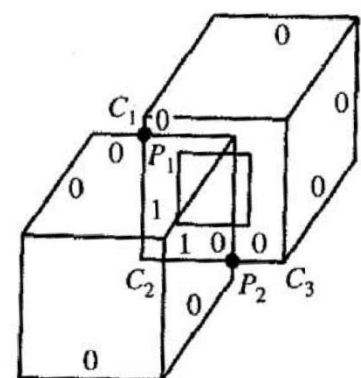
- ✓ Silhouette polygons are polygons whose edges are silhouette edges.
- ✓ First silhouette edges in the scene are recognized.
- ✓ Second, the connection of Silhouette edges to form closed silhouette polygons can be achieved by



(a) Scene of two boxes



Step 1. Silhouette polygons

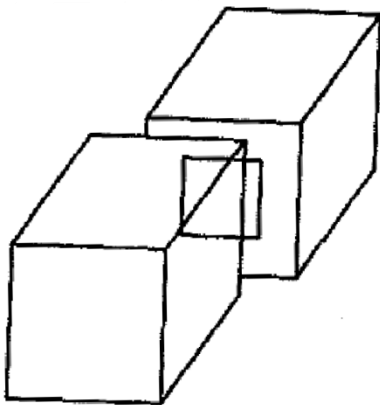


Step 2. Quantitative hiding

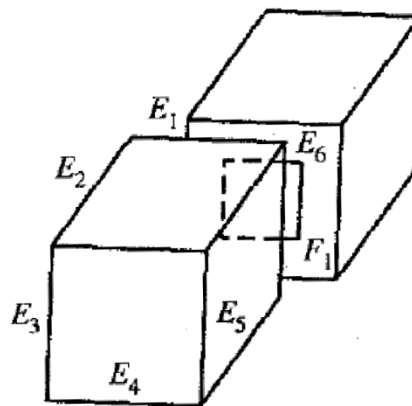
- ✓ Step 2. Assign quantitative hiding (QH) values to edges of silhouette polygons: This is achieved by intersecting the polygons (the containment test can be utilized first as a quick test).
- ✓ The intersection points define the points where the value of QH may change. Applying the depth test to the points of intersection (P1 and P2 Figure), we determine the segments of the silhouette edges that are hidden.
- ✓ For example, if the depth test at P1 shows that Z_v at P1 of S1 is smaller than that of S2, edge C1C2 is partially visible.
- ✓ Similarly, the depth test at P2 shows that edge C2C3 is also partially visible. To determine which segment of an edge is visible, the visibility test can be used.
- ✓ Determination of the values of QH at the various edges or edge segments of silhouette polygons is based on the depth test.
- ✓ A value of 0 indicates that the edge or segment is visible, and a value of 1 indicates that the edge or segment is invisible.

Step 3. Determine the visible silhouette segments:

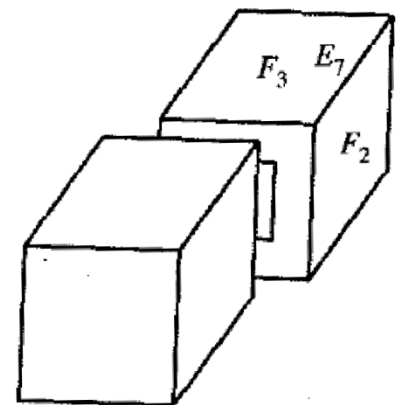
- ✓ From the values of QH, the visibility of silhouette segments can be determined with the following rules in mind.
- ✓ If a closed silhouette polygon is completely invisible, it need not be considered any further.
- ✓ Otherwise, its segments with the lowest QH values are visible.



Step 3. Visibility of silhouette segments



Step 4. Visibility of internal edges



Step 5. Display of visible areas

(b) Steps of the algorithm

- ✓ Step 4. Intersect the visible silhouette segments with partially visible faces: This step is used to determine if the silhouette segments hide or partially hide nonsilhouette edges in partially visible faces.
- ✓ Edges E1 to E6 of S2 are intersected with the internal edges (edges of the square in the face) of F1, and the visible segments of the internal edges are determined,
- ✓ By accessing only the silhouette edges of the covering silhouette polygon only and the partially visible

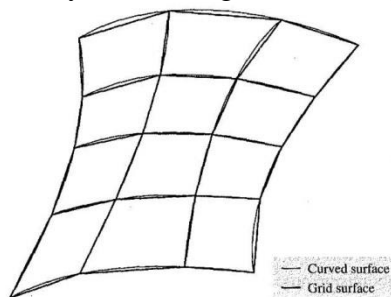
face only, the algorithm avoids any unnecessary calculations.

Step 5: Display the interior of the visible or partially visible polygons:

- ✓ This step can be achieved using a stack and simply enumerates all faces lying inside a silhouette polygon.
- ✓ The stack is initialized with a visible face which has a silhouette edge.
- ✓ We know this face belongs to a visible area.
- ✓ A loop begins with popping a face (F2) from the stack.
- ✓ We examine all the edges of the face. If an edge (E7) is not fully invisible, the neighboring face (F3) also has visible edges and, therefore, is pushed into the stack if it has not already been pushed.
- ✓ The edge itself or its visible segments are displayed.
- ✓ The loop is repeated and the algorithm stops when the stack is empty.
- ✓ The area-oriented algorithm is more efficient than the priority algorithm because it hardly involves any unnecessary edge/face intersection.

Hidden-Line Removal for Curved Surfaces

- ✓ The above algorithms described for flat faces are extendable to curved polyhedral by approximating them by planer polygons.
- ✓ The u-v grid offered by parametric surface representation offers such an approximation.
- ✓ This grid can be utilized to create a “grid surface” consisting of straight- edged regions, as shown in figure by approximating the u-v grid curves by line segments.
- ✓ The overlay hidden-line algorithm is suitable for curved surfaces.
- ✓ The algorithm begins by calculating the u-v grid using the surface equation. It then creates the grid surface with linear edges.
- ✓ Various criteria can be utilized to determine the visibility of the grid surface. There is no best hidden-line algorithm. Many algorithms exist and some are more efficient and fast in rendering images than others for certain applications.
- ✓ Firmware and parallel processing computations of hidden-line algorithms is making it possible to render images in real time.
- ✓ This adds to the difficulty of deciding on a best algorithm.



Approximation of a curved surface.

Hidden-Surface Removal

- ✓ Hidden-surface removal and hidden-line removal are one problem. Most of the line algorithms are applicable here and vice versa.
- ✓ The following are image-space algorithms only for hidden-surface removal. A wide variety of these algorithms exist.
- ✓ They include the z-buffer algorithm, Watkin's algorithm, Warnock's algorithm, and Painter's algorithm.
- ✓ The Watkin's algorithm is based on scanline coherence, while the Warnock's algorithm is an area-coherence algorithm.
- ✓ The Painter's algorithm is a priority algorithm for raster displays.

The z-Buffer Algorithm

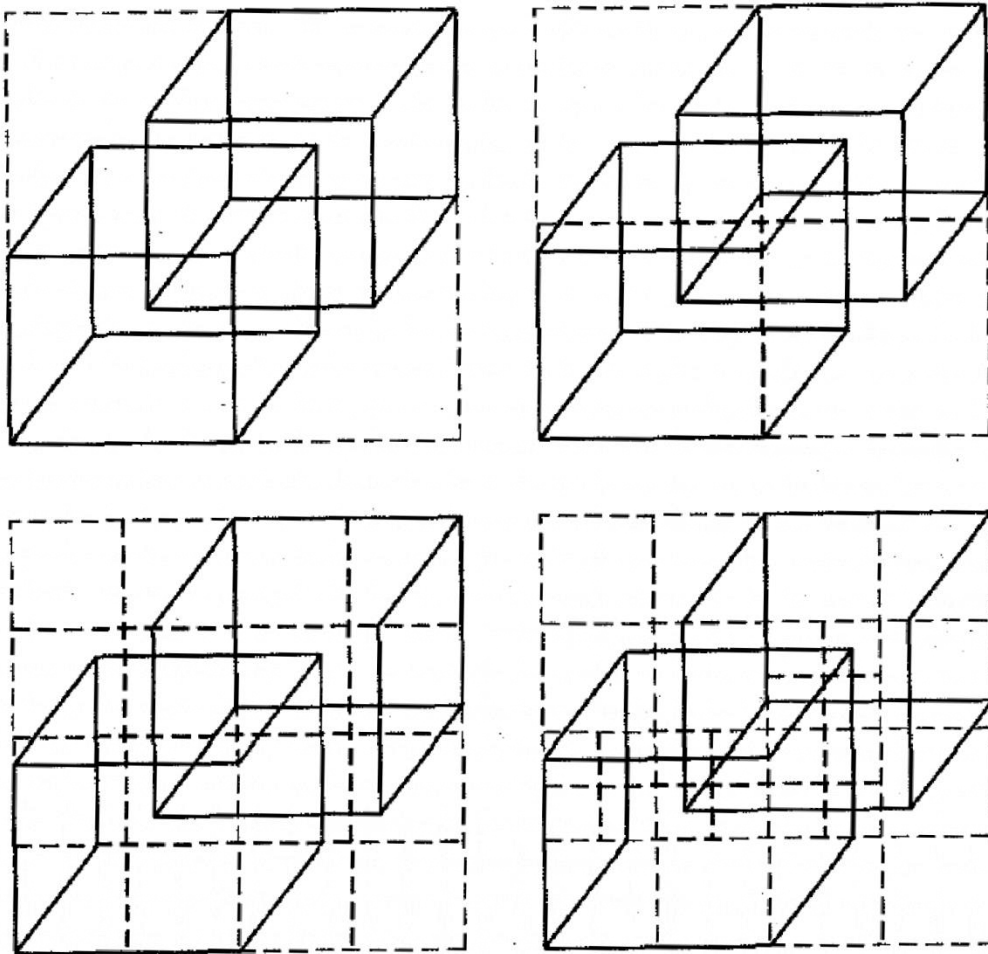
- ✓ This is also known as the depth-buffer algorithm.
- ✓ In addition to the frame buffer, this algorithm requires a z-buffer in which z values can be sorted for each pixel.
- ✓ The z-buffer is initialized to the smallest z-value, while the frame buffer is initialized to the background pixel value. Both the frame- and z-buffers are indexed by pixel coordinates (x, y) .
- ✓ These coordinates are actually screen coordinates.
- ✓ The z-buffer algorithm works as follows. For each polygon in the scene, find all the pixels (x, y) that lie inside or on the boundaries of the polygon when projected onto the screen.
- ✓ For each of these pixels, calculate the depth z of the polygon at (x, y) .
- ✓ If $z > \text{depth}(x, y)$, the polygon is closer to the viewing eye than others already stored in the pixel.
- ✓ In this case, the z-buffer is updated by setting the depth at (x, y) to z . Similarly, the intensity of the frame-buffer location corresponding to the pixel is updated to the intensity of the polygon at (x, y) .
- ✓ After all the polygons have been processed, the frame buffer contains the solution.

Warnock's Algorithm

- ✓ This is one of the first area-coherence algorithms.
- ✓ Essentially, this algorithm solves the hidden-surface problem by recursively subdividing the image into sub images.
- ✓ It first attempts to solve the problem for a window that covers the entire image.
- ✓ Simple cases as one polygon in the window or none at all are easily solved.
- ✓ If polygons overlap, the algorithm tries to analyze the relationship between the polygons and generates the display for the window. If the algorithm cannot decide easily, it subdivides the window into four smaller windows and applies the same solution technique to every window. If one of the four windows is

still complex, it is further subdivided into four smaller windows.

- ✓ The recursion terminates if the hidden-surface problem can be solved for all the windows or if the window becomes as small as a single pixel on the screen.



- ✓ In this case, the intensity of the pixel is chosen equal to the polygon visible in the pixel.
- ✓ The subdivision process results in a window tree.
- ✓ One would devise a rule that any window is recursively subdivided unless it contains two polygons.
- ✓ In such a case, comparing the z depth of the polygons determines which one hides the other.
- ✓ While the subdivision of the original window is governed by the complexity of the scene, the subdivision of any window into four equal windows makes the algorithm inefficient.
- ✓ A better way would be to subdivide a window according to the complexity of the scene in the window.
- ✓ This is equivalent to subdividing a window into four unequal sub windows.

Hidden-Solid Removal

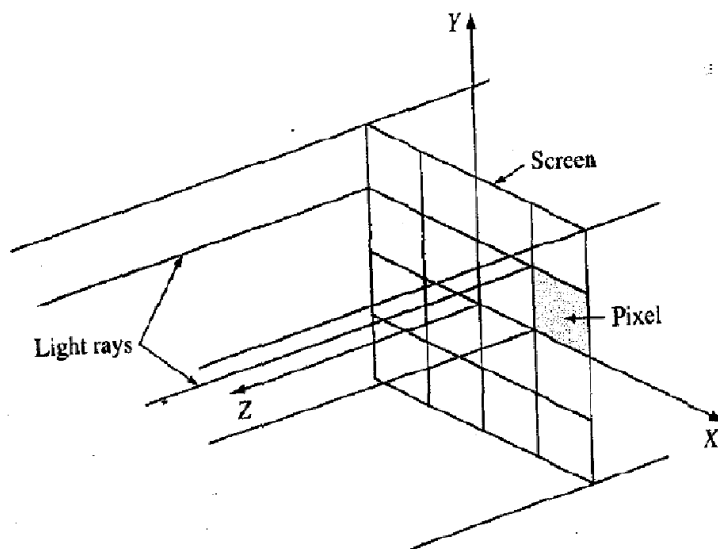
- ✓ The hidden-solid removal problem involves the display of solid models with hidden lines or surfaces removed.
- ✓ Due to the completeness and unambiguity of solid models, the hidden-solid removal is fully automatic.
- ✓ CAD systems provide users with menu choices to display models including shaded, hidden lines removed,

or wireframe (no hidden lines removed).

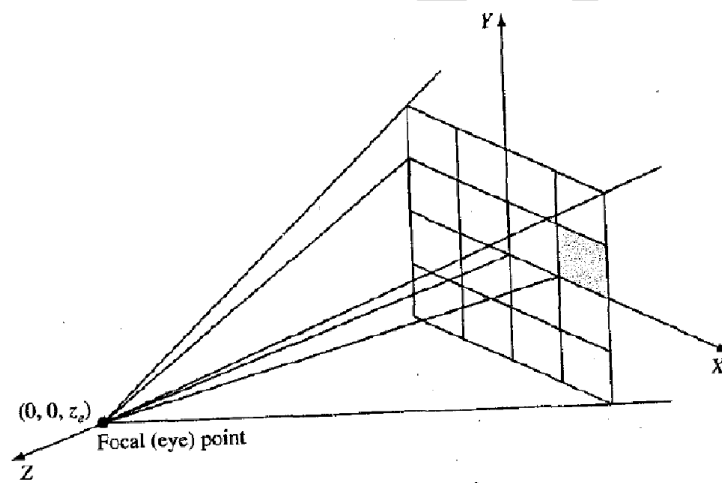
- ✓ For displaying CSG models, both the visibility problem and the problem of combining the primitive solids into one composite model have to be solved. There are three approaches to displaying CSG models.
- ✓ The first approach converts the CSG model into a boundary model that can be rendered with the standard hidden-surface algorithms.
- ✓ The second approach utilizes a spatial subdivision strategy.
- ✓ The third approach uses a CSG hidden-surface algorithm, which combines the CSG evaluation with the hidden-surface removal on the basis of ray classification.

Ray-Tracing Algorithm

- ✓ The virtue of ray tracing is its simplicity and reliability.
- ✓ The most complicated numerical problem of the algorithm is finding the points at which lines (rays) intersect surfaces.
- ✓ Therefore a wide variety of surfaces and primitives can be covered.
- ✓ Ray tracing has been used to enhance the visual realism of solids by generating line drawings with hidden solids removed, animating solids, and shading pictures.
- ✓ It has also been utilized in solid analysis, mainly calculating mass properties. The idea of using ray tracing to generate shaded images of solids is to emulate the photographic process in reverse.
- ✓ For each pixel in the screen, a light ray is cast through it into the scene to identify the visible surface.
- ✓ The first surface intersected by the ray, found by "tracing" along it, is the visible one.
- ✓ At the ray/surface intersection point, the surface normal is computed and knowing the position of the light source, the brightness of the pixel can be calculated.
- ✓ Ray tracing is considered a brute force method for solving problems. The basic ray-tracing algorithm is simple, but slow.
- ✓ The CPU usage of the algorithm increases with the complexity of the scene under consideration.
- ✓ Various alterations and refinements have been added to the algorithm to improve its efficiency.
- ✓ Moreover, the algorithm has been implemented into hardware (ray-tracing firmware) to speed its execution.
- ✓ The basics of ray tracing stems from light rays and camera models.
- ✓ The geometry of a simple camera model is analogous to that of projection of



(a) Parallel view



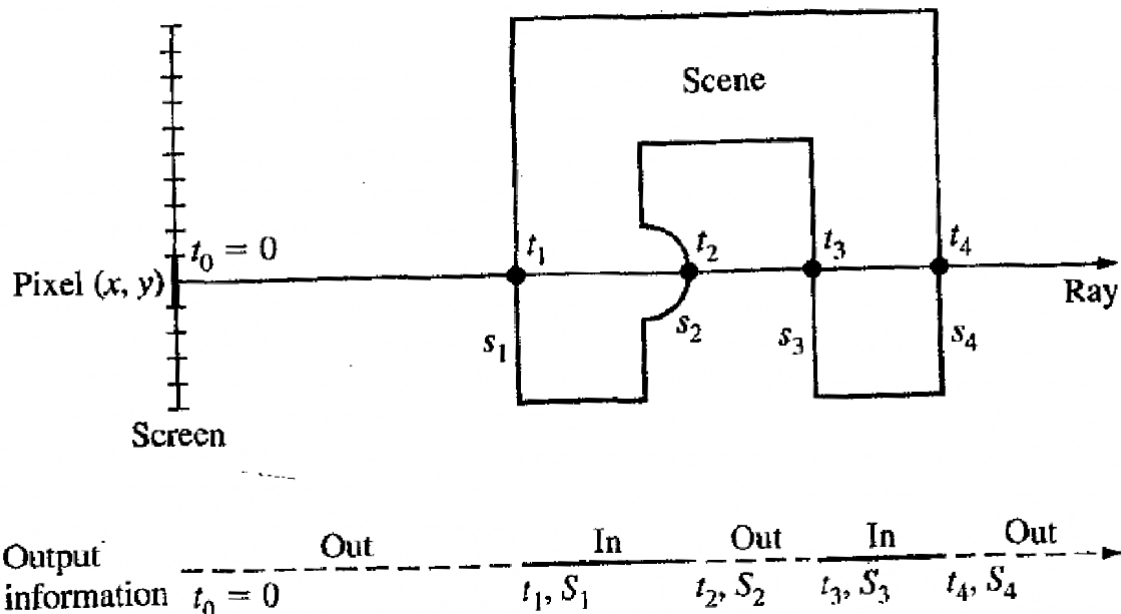
(b) Perspective view

geometric models.

- ✓ Referring to Figure, the center of projection, projectors, and the projection plane represent the focal point, light rays, and the screen of the camera model, respectively.
- ✓ We assume that the camera model uses the VCS. For each pixel of the screen, a straight light ray passes through it and connects the focal point with the scene.
- ✓ When the focal length, the distance between focal point and screen, is infinite, parallel views result, and all light rays become parallel to the Z_v axis and perpendicular to the screen (the $X_v Y_v$ plane).
- ✓ A ray is a straight line which is best defined in a parametric form as a point (X_0, Y_0, z_0) and a direction vector $(\Delta x, \Delta y, \Delta z)$.
- ✓ Thus, a ray is defined as $[(X_0, Y_0, Z_0) (\Delta x, \Delta y, \Delta z)]$.
- ✓ For a parameter t , any point (x, y, z) on the ray is given by:

$$\begin{aligned} x &= x_0 + t\Delta x \\ y &= y_0 + t\Delta y \\ z &= z_0 + t\Delta z \end{aligned}$$

- ✓ A ray-tracing algorithm takes the ray definition as an input and output information about how the ray intersects the scene.
- ✓ Knowing the camera model and the solid in the scene, the algorithm can find where the given ray enters and exits the solid as shown in Figure for a parallel view.
- ✓ The output information is an ordered list of ray parameters, t_i which denotes the enter/exit points, and a list of pointers, S_i to the surfaces (faces) through which the ray passes.
- ✓ The ray enters the solid at point t_1 , exits at t_2 enters at t_3 and finally exits at t_4 .
- ✓ Point t_1 is closest to the screen and point t_4 is farthest.
- ✓ The lists of ray parameters and surface pointers suffice for various applications.



- ✓ While the basics of ray tracing are simple, their implementation into a solid modeler is more involved and depends largely on the representation scheme of the modeler.
- ✓ When boundary representation is used in the object definition, the ray-tracing algorithm is simple.
- ✓ For a given pixel, the first face of the object intersected by the ray is the visible face at that pixel.
- ✓ When the object is defined as a CSG model, the algorithm is more complicated because CSG models are compositions of primitive solids. Intersecting the primitive solids with a ray yields a number of intersection points, which requires additional calculations to determine which of these points are intersection points of the ray with the composite solid.
- ✓ A ray-tracing algorithm for CSG models consists of three main modules: ray/primitive intersection, ray/primitive classification, and ray/solid classification.
- ✓ The ray tracing algorithm to generate line drawings of hidden solids has advantages.

- ✓ It eliminates finding, parameterizing, classifying, and storing the curved edges formed by the intersection of surfaces.
- ✓ The silhouettes of curved surfaces are by-products, and they can be found whenever the view changes.
- ✓ The main drawbacks of the algorithm are speed and aliasing.
- ✓ Aliasing causes edges to be jagged and surface "slivers" may be overlooked. Speed is particularly important to display hidden-solid line drawings in an interactive environment.
- ✓ If the user creates a balanced tree of the solid in the scene, the efficiency of ray tracing improves.
- ✓ The coherence of visible surfaces (surfaces visible at two neighboring pixels are more likely to be the same than different) can also speed up the algorithm.

Shading

- ✓ Line drawings, still the most common means of communicating the geometry of parts, are limited in their ability to portray intricate shapes.
- ✓ Shaded images can convey complex shape information.
- ✓ The visible faces of an object are found by hidden surface algorithms, but further information is required to display the colour of the object.
- ✓ Shading refers to the process of altering the colour of an object surface polygon in the 3D scene based on its angle on lights and its distance from lights to create a photorealistic effect.
- ✓ They also can convey features other than shape such as surface finish or material type (plastic or metallic look).
- ✓ Shaded-image rendering algorithms filter information by displaying only the visible surface.
- ✓ Many spatial relationships that are unresolved in simple wireframe displays become clear with shaded displays.
- ✓ Shaded Images are easier to interpret because they resemble the real objects.
- ✓ Shaded images have viewing problems not present in wireframe displays. Solids of interest may be hidden or partially obstructed from view, in which case various shaded images may be obtained from various viewing points.
- ✓ Critical geometry such as lines, arcs, and vertices are not explicitly shown. Well-known techniques such as shaded-Image/wireframe overlay, transparency, and sectioning can be used to resolve these problems.
- ✓ In shading a scene (rendering an image), a pinhole camera model is almost universally used.
- ✓ Rendering begins by solving the hidden-surface removal problem to determine which objects and/or portions of objects are visible in the scene.

- ✓ As the visible surfaces are found, they must be broken down into pixels and shaded correctly.
- ✓ This process must take into account the position and color of the light sources and the position, orientation, and surface properties of the visible objects.
- ✓ Shading models simulate the way visible surfaces of objects reflect light. They determine the shade of a point of an object in terms of light sources, surface characteristics, and the positions and orientations of the surfaces and sources.
- ✓ Two types of light sources can be identified: point light source and ambient light.
- ✓ Objects illuminated with only point light source look harsh because they are illuminated from one direction only.
- ✓ This produces a flashlight-like effect in a black room. Ambient light is a light of uniform brightness and is caused by the multiple reflections of light from the many surfaces present in real environments.
- ✓ Shading models are simple. The inputs to a shading model include intensity and color of light source (S), surface characteristics at the point to be shaded, and the positions and orientations of surfaces and light sources.
- ✓ The output from a shading model is an intensity value at the point.
- ✓ Shading models are applicable to points only. To shade an object, a shading model is applied many times to many points on the object.
- ✓ These points are the pixels of the display. To compute a shade for each point on a 1024 x 1024 raster display, the shading model must be calculated over one million times.
- ✓ These calculations can be reduced by taking advantage of shading coherence, that is, the intensity of adjacent pixels is either identical or very close.
- ✓ Particularly, we consider point light sources shining on the surfaces of objects. (Ambient light adds a constant intensity value to the shade at every point.)
- ✓ The light reflected off a surface can be divided into two components: diffuse and specular.
- ✓ When light hits an ideal diffuse surface, it is reradiated equally in all directions, so that the surface appears to have the same brightness from all viewing angles. Dull surfaces exhibit diffuse reflection.
- ✓ Examples of real surfaces that radiate mostly diffuse light are chalk, paper, and flat paints.
- ✓ Ideal specular surfaces reradiate light in one direction only, the reflected light direction.
- ✓ Examples of specular surfaces are mirrors and shiny surfaces. Physically, the difference between these two components is that diffuse light penetrates the surface of an object and is scattered internally before emerging again, while specular light bounces off the surface.
- ✓ The absence of diffuse light makes a surface look shiny.
- ✓ The light reflected from real objects contains both diffuse and specular components, and both must be

modeled to create realistic images.

- ✓ A basic shading model that incorporates both a point light source and ambient light can be described as follows:
- ✓ $I_p = I_d + I_s + I_b$; where I_p , I_d , I_s , and I_b are respectively the resulting intensity (the amount of shade) at point p , the intensity due to the diffuse reflection component of the point light source, the intensity due to the specular reflection component, and the intensity due to ambient light.
- ✓ The above Equation is written in a vector form to permit the modeling of colored surfaces.
- ✓ For the common red, green and blue color system, Eq. represents three scalar equations, one for each color.
- ✓ Specular reflection is a characteristic of shiny surfaces. Highlights visible on shiny surfaces are due to specular reflection, while other light reflected from these surfaces is caused by diffuse reflection.
- ✓ The location of a highlight on a shiny surface depends on the directions of the light source and the viewing eye. If you illuminate an apple with a bright light, you can observe the effects of specular reflection. Note that at the highlight the apple appears to be White (not red), which is the color of the incident light.
- ✓ The specular component is not as easy to compute as the diffuse component. Real objects are nonideal specular reflectors, and, some light is also reflected slightly off axis from the ideal light direction (defined by vector r in Figure).
- ✓ This is because the surface is never perfectly flat but contains microscopic deformations.
- ✓ For ideal (perfect) shiny surfaces (such as mirrors), the angles of reflection and incidence are equal.
- ✓ This means that the viewer can only see specular reflected light when the angle Q is zero.
- ✓ For non ideal (non perfect) reflectors, such as an apple, the intensity of the reflected light drops sharply as Q increases.
- ✓ One of the reasonable approximations to the specular component is an empirical approximation and takes the form.
- ✓ Most surfaces, including those that are curved, are described by polygonal meshes when the visible-surface calculations are to be performed by the majority of rendering algorithms.
- ✓ The majority of shading techniques are therefore applicable to objects modeled as polyhedra.
- ✓ Shading Methods:
 - With a single intensity – Flat Shading
 - Using a interpolation scheme – Smooth Shading

Flat Shading	Smooth Shading
Single intensity is calculated for each polygon and hence looks less realistic.	Intensity at each point of a surface can be obtained using an interpolation scheme.
Uses the same colour for every pixel in a face.	Smooth shading uses linear interpolation of colours between vertices.
Edges appear more pronounced than they would on a real object.	The edges disappear with this technique.
Individual faces are visualized.	Underlying surface is visualized.
Same colour for any point of the face.	Each point of the face has its own colour.
Not suitable for smooth objects.	Suitable for any objects.
Less computationally expensive.	More computationally expensive.
Used for high speed rendering.	Used for more realistic rendering.

Types of Smooth shading

Gouraud Shading or First derivative shading

- ✓ Gouraud shading, computes an intensity for each vertex and then interpolates the computed intensities across the polygons.
- ✓ Gouraud shading performs a bi-linear interpolation of the intensities down and then across scan lines.
- ✓ It thus eliminates the sharp changes at polygon boundaries

The algorithm is as follows:

- Compute a normal N for each vertex of the polygon.
- From N compute an intensity I for each vertex of the polygon.
- From bi-linear interpolation compute an intensity I_i for each pixel.
- Paint pixel to shade corresponding to I_i .

Advantages of Gouraud shading:

- ✓ Gouraud shading gives a much better image than faceted shading It is not too computationally expensive

Disadvantages to Gouraud shading:

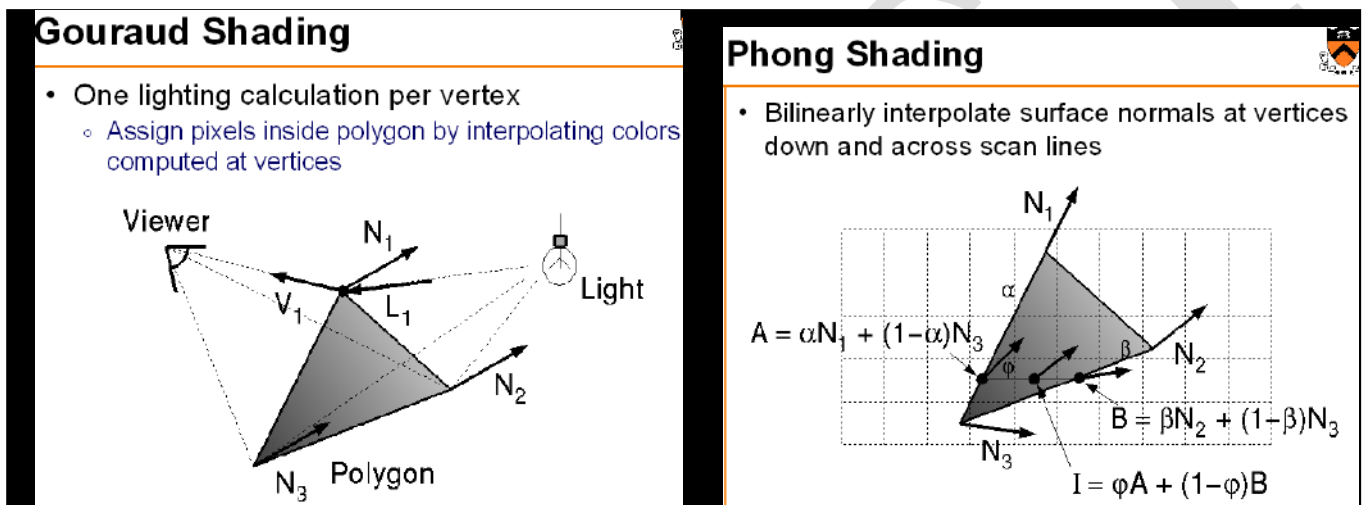
- ✓ It eliminates creases that you may want to preserve, e.g. in a cube.

Phong Shading or Second derivative shading

- ✓ Phong shading, is similar to Gouraud shading except that the Normals are interpolated.
- ✓ It interpolates normal vectors themselves across polygons and then applies the shading model at each pixel in the image.

- ✓ Thus, the specular highlights are computed much more precisely than in the Gouraud shading model.
- ✓ The algorithm is as follows:
 - Compute a normal N for each vertex of the polygon.
 - From bi-linear interpolation compute a normal, N_i for each pixel. (This must be renormalized everytime)
 - From N_i compute an intensity I_i for each pixel of the polygon.
 - Paint pixel to shade corresponding to I_i .
- ✓ Note that this method is much more computationally intensive than Gouraud shading:

Shading Enhancements



The visual realism of images can be enhanced by including shading effects such as:

Transparency, Shadows, Surface details, Texture

Transparency :

- ✓ It is used to shade translucent materials which allows some of the back pixels to show through producing a screen – door effects.
- ✓ The light emission from a transparent material is generally a combination of reflected and transmitted light.

Shadows :

- ✓ Important elements of visual realism. They give important cues about light and object positions.
- ✓ Two types of shadow algorithm are available: shadow volumes and shadow mapping.

Surface details :

- ✓ It refers to the patterns contained by some surfaces such as logo of an object, paintings on a vessel, dividing lines on a highway, etc that must be taken into account in the rendering process.

Texture :

- ✓ It is another important element that enhances the realism of an image.
- ✓ It is an approach where the texture pattern may either be defined in a rectangular array or as a procedure

that modifies surface intensity values.

- ✓ It can also contain other surface properties such as wrinkled surfaces.

COLOURING

- ✓ The two main ingredients of shaded images are colours and textures. The display of realistic scenes is mostly in colour.
- ✓ Colours are used for realism, aesthetics and to distinguish the different areas in the geometry of an object.
- ✓ They help the designer to classify components in an assembly or highlight the sectional views and dimensions of the model.
- ✓ Neutral colours such as black, white or grey are called achromatic colours, provided by black and white raster displays.
- ✓ The only attribute of achromatic light is its intensity or amount which is assigned a value between 0 and 1.
- ✓ Colour is created by taking advantage of the fundamental trichromacy of the human eye.
- ✓ A typical colour CRT uses three electron beams and a triad of colour dots on the phosphor screen to provide each of the three colours, red, green and blue.
- ✓ In the other hand, colour raster can terminate superior colour graphics capabilities and are usually available with 1024 X 1024 resolution but require large memory for the refreshbuffer.
- ✓ Three colour parameters are
 - Hue
 - Saturation or Purity
 - Brightness
- ✓ The combination of frequencies present in the reflected light from an object is perceived as the colour of the object.
- ✓ The Hue or simply the colour is the dominant wavelength or dominant frequency.
- ✓ Saturation is the purity of a colour. It describes how washed out or how pure the colour appears.
- ✓ It defines a range from pure colour (100%) to gray(0%) at a constant lightness level.
- ✓ Brightness represents the perceived intensity of light. It refers to the lightness or darkness of a colour.
- ✓ Dark value with black added are called 'shades' and light value with white added are called 'tints' by adding both black and white pigments, 'tones' of the colour are produced.

Colour Models

- ✓ A colour model is an abstract mathematical model describing the way colours can be presented as tuples of numbers, typically as three or four colour components.
- ✓ It is used to describe colour as accurately as possible.

✓ The range of colours that can be described by a combination of other colours is called a colour gamut.

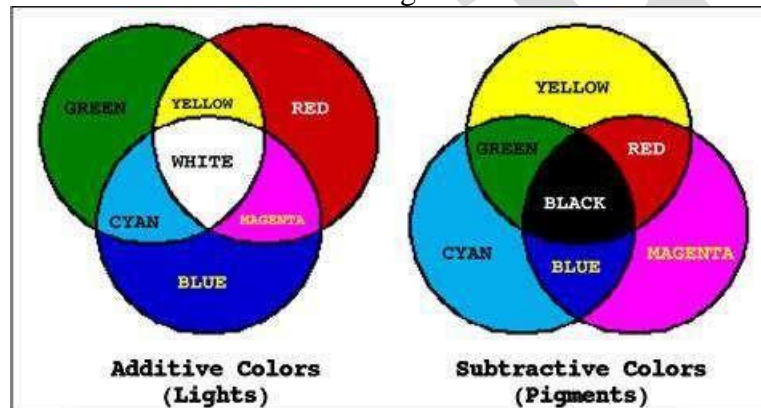
- Additive colour models
- Subtractive colour models

Additive Color models

- ✓ Additive Color models are based on the principle of transmitted light.
- ✓ It is created by mixing a number of different colours, the primary colours red, blue and green being normally used.
- ✓ This system includes monitors, liquid crystal displays, digital projectors and televisions.
- ✓ Each pixel on a monitor screen starts out as black but when red, green and blue phosphors of the pixel are illuminated simultaneously, that pixel becomes white.

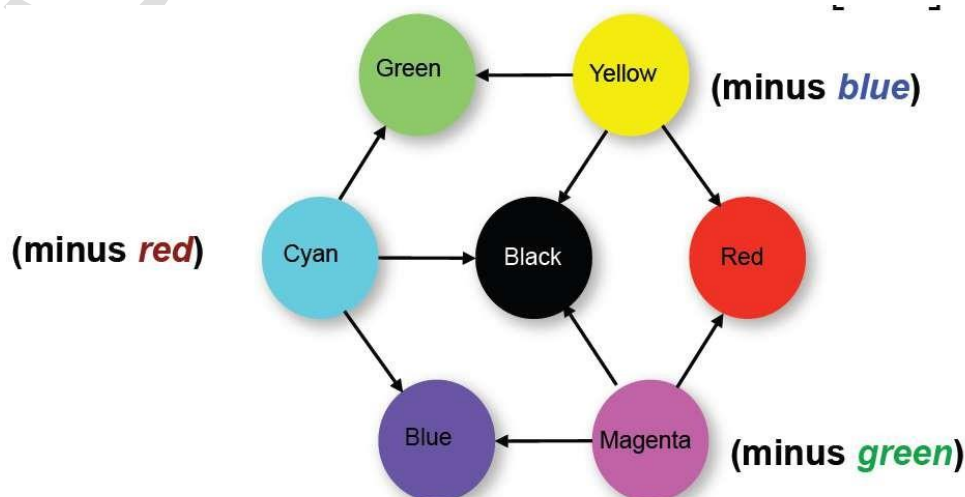
Subtractive Color models

- ✓ This model perceives colour as a result of reflected light.



The colour that a surface displays depends on which parts of the visible spectrum are not absorbed and therefore remain visible.

- ✓ If an object absorbs (subtracts) all the illuminating light, i.e., no light is reflected back to the viewer it appears black.
- ✓ Subtractive colour models filter the red, green and blue components of the image from white light.
- ✓ Colour paintings, photography and printing processes use the subtractive process to reproduce colour.
- ✓ In printing, black is added to improve the contrast. Printing processes use colour inks that act as filters.
- ✓ These inks are transparent – it is the paper that reflects the unabsorbed light back to the viewer.



Different Colour Models

- ✓ RGB Colour model – Additive color model; Black at origin (0,0,0) CMY (or) CMYK colour model; Subtractive; White at origin (0,0,0)
- ✓ YIQ Colour model; Adopted by the National Television System Committee (NTSC)

Y-axis contains the brightness information

- I stands for In-phase
- Q stands for Quadrature, both representing the hue and purity. HSV Colour model; Hue, Saturation and Value
- ✓ HSL Colour model; Hue, Saturation and Lightness

COMPUTER ANIMATION

- ✓ To 'animate' literally means 'to give life to'.
- ✓ The process of moving something which can't move by itself is called 'animation'.
- ✓ It is a technique for creating the illusion of motion with a series of static images. Animation adds to graphics the dimension of time which vastly increases the amount of information which can be transmitted.
- ✓ Conventional animation is defined as a technique in which the illusion movement is created by photographing a series of individual drawings on successive frames of film.
- ✓ Computer animation is the use of computer to create animation.
- ✓ Virtual entities may contain and be controlled by attributes such as location, orientation and scale.
- ✓ Animation is the change of an attribute over time. It generally refers to any time sequence of visual changes in a scene.
- ✓ To create the illusion of movement, an image is displayed on the computer screen and then quickly replaced by a new image that is similar to the previous image but shifted slightly.
- ✓ To trick the eye and brain into thinking they are seeing a smoothly moving object, the images should be drawn at around 12 frames per second or faster.
- ✓ A typical animation sequence for animated cartoons is:
 - Storyboard layout
 - Object definitions
 - Key-frame specifications
 - Generation of in-between frames
 - Recording animation sequence
- ✓ Computer animation demands higher frame rates as it produces more realistic images.

- Computer assisted and Computer generated
- ✓ Computer assisted animation is mostly 2-dimensional that computerize the traditional animation process.
- ✓ Computer generated or modeled animation or 3D animation utilizes available computer graphics and CAD techniques to create images, scenes and movements.
- ✓ Applications of Computer animation
 - Engineering, Educational
 - Entertainment, Advertising, Art
 - Architecture, Forensics, Medicine
 - Military and Space exploration.
 - In CAD :
 - Kinematic Simulation
 - Analysis of Linkage mechanism
 - Planning of a robotic work cycle

✓ Animation

- Frame buffer Animation
 - It provides the illusion animation for a variety of applications.
 - Limited real-time animation
 - Uses Static pictures which are stored in the image memory of a digital frame buffer.
 - The image once created and stored in the memory of the frame buffer remains completely unchanged.
 - Dynamics are added by modifying the pattern in which pixels are read from memory.
 - Techniques: Colour table animation, Zoom-pan-scroll animation, cross-bar animation
- Real-Time Playback
 - Frames are generated in advance at non-real time rates and stored on a file.
 - The frames are then later displayed to create the animation.
 - Frame rate determines how smoothly the animation plays back.
 - The frames are recorded and then played back at the rate required for real time presentation.
- Real Time Animation

- Creating animations in real-time is called real-time/live.
- This animation is limited by the capabilities of the computer and data transfer rates.
- Very complex animation is possible in a short time with the development in parallel processing and multiprocessors.

✓ Computer Animation Techniques

○ Keyframing

- A keyframe is a detailed drawing of the scene at a certain time in the animation sequence.
- Animator specifies the critical keypoints.
- Then the computer automatically generated the intermediate frames, called inbetweens.
- Using interpolation techniques huge amount of time can be saved.
- The animator has direct control over the position, shapes and motions of models at any moment in the animation.

○ Simulation/Procedural

- Also called algorithmic animation or simulation.
- The computer procedurally follows a set of rules to generate the motion.
- The animator specifies the parameters and initial conditions and runs simulation.
- The effect of changing a parameter is often unpredictable and the animator has to run a simulation to see the result.
- It is easy to generate a family of similar motions.
- It can be used for complex systems.

○ Motion Capture

- Also called as Mocap/performance animation.
- Special sensors called trackers, record the motions of a human or animal in three dimensions.
- This data is then used by the computer to generate the motion for an animation.
- Special puppets with joint angle sensors can also be used in the place of human performers.
- It's a popular technique with animators.
- Motion capture enables famous athletes to supply the

actions for characters in sports video games.

- Combinations

- ✓ Computer animation hardware and software

- The quality of the computer animation produced also depends on the hardware and software used.
- Commonly used hardware are Silicon Graphics Inc (SGI), PCs, Macintosh and Amiga.
- Popular softwares are 3D Studio Max, Light Wave 3D, Adobe Premiere, Alias wavefront, Animator Studio, Soft Image etc.

- ✓ Animation problems

- Frame to frame flicker

- The blank period between erasing and generating the contents of the pixels of a graphics display caused a blinking effect called flicker.

- Frame to frame discontinuity

- If the time sampling rate is not adequate, it results in discontinuous and jerky motion.

- Spatial aliasing

- This results in jagged edges.
- Anti-aliasing techniques can be used to solve the problem.

- Object interactions

- The problem of detecting and controlling object interactions is encountered when several objects are animated at once in a scene.
- In some systems, animator visually inspects the scene for object interaction.
- This is a time consuming and difficult process.
- Collision detection and collision response algorithms are available.

TOLERANCE ANALYSIS

- ✓ The tolerance is the permissible variation in the size of a dimension and it is the difference between the upper and lower acceptable limits.
- ✓ The tolerance produces a range of acceptable values for the dimension.
- ✓ If a part size and shape are not within the maximum and minimum limits defined by the part tolerances, the part is not acceptable: it is rejected during inspection and becomes scrap.

Need for tolerances:

- ✓ It is impossible to manufacture a component with exact size.

- ✓ Hence the designer decides how close to the real size is satisfactory.
- ✓ As a result of which most of the manufacturer produces interchangeable parts.
- ✓ The benefits of the interchangeable parts are listed below:

(a) Assembly of parts requires less skilled operators. So it is cheaper and quality is easier to maintain.

(b) The damaged components can be easily replaced by another component with the same range of dimensions.

(c) Parts can be made in large quantities with less demand on the skill of operator.

Tolerance Vs Cost:

- ✓ If the tolerances are too small (tight), the individual parts will cost more to make. If the tolerances are too large (loose), an unacceptable percentage of assemblies may be scrapped (rejected) or require rework. The three main characters that affect the cost of the given tolerance are:

- (i) Tolerance size,
- (ii) Component size,
- (iii) Detail of the features.

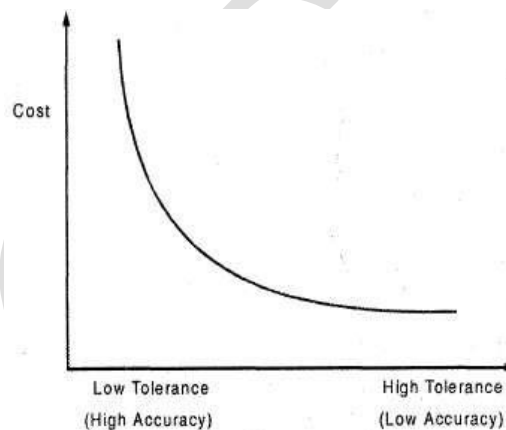


Fig. 4.8. Tolerance vs Cost Graph.

TERMINOLOGY

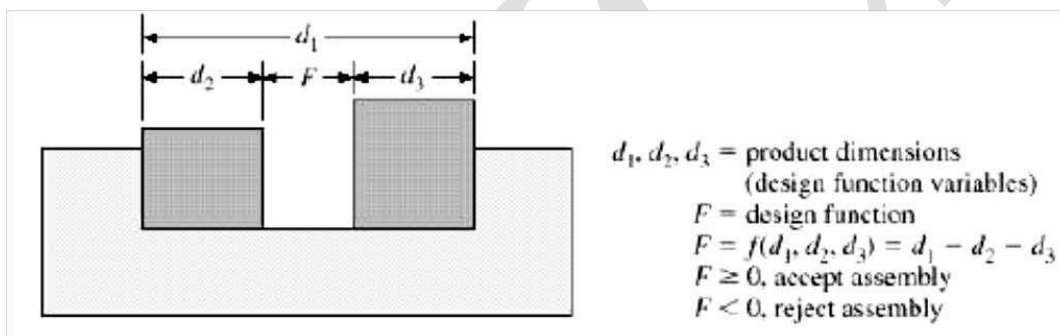
(ANSI (American National Standards Institute) definitions of tolerance terms)

- ✓ Nominal size is the designation used for the purpose of general identification.
 - o It is the dimension that results from design calculations such as stress or heat transfer analysis. It is usually expressed in common fractions or decimals.
- ✓ Basic size is the theoretical size from which limits of size are derived by the application of allowances and tolerances.
 - o It is the decimal equivalent of the nominal size.
 - o The number of decimal places determines the precision or accuracy required.
 - o For example, if the nominal size of a pipe is 3" the sizes of 3.5", 3.50", and 3.500" are three different sizes when we consider manufacturing and producing these dimensions in the machine shop.
- ✓ Actual size is the measured size of the finished part.
 - o It is a combination of the basic size and its assigned tolerance.
 - o It is the size that results from producing the part including all the manufacturing imperfections.
 - o The actual size must lie within the maximum and minimum allowable values for the size (known as limits); otherwise the part is rejected during inspection, therefore becoming scrap.

- ✓ Tolerance is the total amount by which a dimension may vary.
 - It is used to determine the permissible limits (maximum and minimum) of the dimensions.
- ✓ Allowance is the difference between the maximum material limits of mating parts.
 - It is the minimum clearance (positive allowance) or maximum interference (negative allowance) between mating parts.
 - It is also known as the tightest fit between mating parts-the smallest clearance or the largest interference.

Tolerance Analysis

- ✓ Tolerance analysis is defined as the process of checking the tolerances to verify that all the design constraints are met while manufacturing the part.
- ✓ Tolerance analysis is sometimes known as design assurance.
- ✓ The objective of tolerance analysis is to determine the variability of any quantity that is a function of product dimensions and are called design functions. Most often, these quantities are themselves dimensions.
- ✓ Product dimensions and variables that control the behaviour of a design function are called design function variables.



- ✓ The variability of design functions is used to assess the suitability of a particular tolerance specification.
- ✓ Tolerance analysis methods can be divided into two types. In the simple type, dimensions have conventional tolerances, and the result of tolerance analysis is the nominal value of the design function () and its upper () and lower () limits.
- ✓ This type of analysis is sometimes called worst-case analysis. This means that all possible combinations of in-tolerance parts must result in an assembly that satisfies the design constraints.
- ✓ The upper and lower limits of the design function represent the worst possible combination of the tolerances of the design function variables.
- ✓ The other type of tolerance analysis is performed on a statistical basis. Tolerance analysis methods of this type allow statistical tolerances and output a statistical distribution for the design function. This allows for more realistic analysis.

Worst-Case Arithmetic Method

- ✓ It uses the limits of dimensions to carry out the tolerance calculations. The actual or expected distribution of dimensions is not taken into account.
- ✓ Arithmetic tolerances require greater manufacturing accuracy. It is used in job shop production (very few parts are produced) and in cases where totally or 100% interchangeable assembly is required.
- ✓ Let us assume a closed-loop (meaning the resulting dimension is obtained by adding and/or subtracting the given dimensions) dimension set {d} of n elements such that the design function (resultant dimension) F is obtained by adding the first m elements (called increasing dimensions) and subtracting the last (n - m) elements (called decreasing dimensions).
- ✓ Using this method all tolerance information about F is obtained by adding and/or subtracting the corresponding information about the individual dimensions. Thus, we can write:

Nominal dimension:

$$F_N = \sum_{i=1}^m d_{iN} - \sum_{i=m+1}^n d_{iN}$$

Maximum dimension:

$$F_{\max} = \sum_{i=1}^m d_{i\max} - \sum_{i=m+1}^n d_{i\min}$$

Minimum dimension:

$$F_{\min} = \sum_{i=1}^m d_{i\min} - \sum_{i=m+1}^n d_{i\max}$$

Tolerance on F:

$$T_F = F_{\max} - F_{\min} = \sum_{i=1}^m T_i + \sum_{i=m+1}^n T_i = \sum_{i=1}^n T_i$$

Upper tolerance on F:

$$\begin{aligned} T_{uF} = F_{\max} - F_N &= \sum_{i=1}^m d_{i\max} - d_{iN} - \sum_{i=m+1}^n d_{i\min} - d_{iN} \\ &= \sum_{i=1}^m T_{u(i)} - \sum_{i=m+1}^n T_{L(i)} \end{aligned}$$


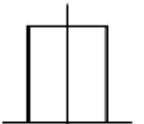
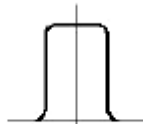

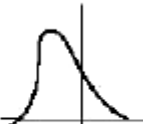
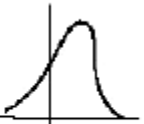
Lower tolerance on F:

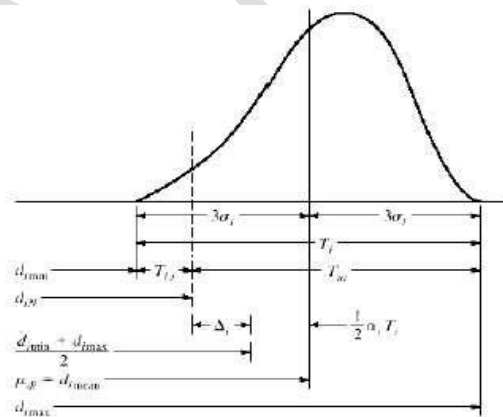
$$\begin{aligned} T_{LF} = F_{\min} - F_N &= \sum_{i=1}^m d_{i\min} - d_{iN} - \sum_{i=m+1}^n d_{i\max} - d_{iN} \\ &= \sum_{i=1}^m T_{L(i)} - \sum_{i=m+1}^n T_{u(i)} \end{aligned}$$

- ✓ where and are the upper and lower tolerances on dimension , respectively. For unilateral tolerances. One of these variables is zero.

Worst-Case Statistical Method

- ✓ This method, like the arithmetic method, uses the limits of dimensions to perform tolerance analysis.
- ✓ However, unlike the arithmetic method, it takes into consideration the fact that dimensions of parts of an assembly follow a probabilistic distribution curve. Consequently, the frequency distribution curve of the dimensions of the final assembly follows a probabilistic distribution curve.
- ✓ This method is used in both batch and mass production. It allows for variabilities in manufacturing conditions such as tool wear, machine conditions and random errors.
- ✓ It increases the manufacturing efficiency by increasing tolerance limits and, therefore, reducing the required accuracy of manufacturing.
- ✓ This method is applied to a closed-loop dimension set {d} with each element of the set having a probability distribution curve. The design function F is obtained in the same way as in the arithmetic method.
- ✓ A normal distribution is considered the basis of the analysis. Parameters relating other distributions to the normal distribution are shown in Table

Distribution	Normal	Uniform	Quasi-uniform	Triangle	Left skew	Right skew
Shape						
	0	0	0	0	0.26	0.26
K	1.0	1.73	1 1.5	1.22	1.17	1.17



Probabilistic distribution curve of a dimension.

- ✓ When the elements in the dimension set become large enough, the distribution of the design function F (the resulting dimension) will be asymptotically normal and independent of the distributions of the individual dimension.

$$F_N = \sum_{i=1}^m d_{iN} - \sum_{i=m+1}^n d_{iN}$$

$$T_F = \sqrt{\sum_{i=1}^n K^2 T_i^2}$$

$$F = \sqrt{\sum_{i=1}^n \frac{T_i^2}{6}} = \frac{T_F}{6}$$

$$T_{uF} = +\frac{1}{2} T_F$$

$$T_{LF} = -\frac{1}{2} T_F$$

$$= \sum_{i=1}^m \left(i + \frac{1}{2} \right) T_i - \sum_{i=m+1}^n \left(i + \frac{1}{2} \right) T_i$$

Monte Carlo Simulation Method

- ✓ The previous two methods are only applicable to conventional tolerances- mainly for closed-loop dimensional sets with linear design functions.
- ✓ Then these functions become more complex or nonlinear, applying these methods becomes less obvious if not impossible.
- ✓ Consider the simple example of a box that has sides of lengths a, b and c with 1% tolerances on each dimension.
- ✓ To calculate the resulting tolerances on a diagonal of the box, the design function F is the length of the diagonal and given by the relation:

$$F = \sqrt{a^2 + b^2 + c^2}$$

- ✓ To calculate the tolerances on the diagonal using the worst-case arithmetic method, we reduce (or increase) each dimension by 1%, which gives a tolerance on the diagonal of 0.01 .
- ✓ In addition, geometric tolerances must be considered in the tolerance analysis. To bring these tolerances into the analysis, 2D (such as area) or 3D (such as volume) design functions may have to be formulated instead of the 1D (dimension) functions used in the previous methods.
- ✓ These 2D and 3D functions can be written first in terms of nominal dimensions and then perturbed using the geometric tolerances.
- ✓ The Monte Carlo method is applicable whether the design variables (these are the design function variables used in the previous two methods) are linear or not.
- ✓ The method operates by generating (with computer random generation) a large sample of assembly instances of the assembly model to be analyzed. Each assembly instance consists of parts instances, each of which corresponds to a set of dimensions that are generated randomly using the statistical distributions assumed for nominal dimensions.
- ✓ Each generated assembly instance is checked to determine whether it meets the specified design constraints or not. Thus a statistical distribution can be generated for each design constraint.

Probabilities of accepting or rejecting assemblies can, therefore be estimated.

- ✓ An algorithm based on the Monte Carlo method and implemented into a solid model can be described as follows:
- ✓ 1. Generate a candidate instance of an as-manufactured part using a normal (or other) distribution random number generator to perturb the vertices of the part within the specified size tolerance zone.
- 2. Check if the part instance meets the specified form tolerances. This is needed because form tolerances may be tighter than size tolerances, and because normal distribution may, in rare cases, generate perturbations with standard deviations beyond the size tolerance zone.
- 3. If one or more of the vertices of the part instance are found outside the zone of form tolerance, the assembly instance is rejected. If all vertices are inside the zone, the assembly instance may be accepted.
- 4. Repeat steps 1 to 3 for all other parts in the assembly.
- 5. Use the solid modeller to create the assembly instance using all the instances created in step 4. These instances are positioned relative to datums established by part features.
- 6. Check if the assembly instance from step 5 satisfies the design constraints. If it does, the assembly is accepted. Otherwise it is rejected.
- 7. Repeat steps 1 to 6 as many times as is needed by the desired sample size (number of assembly instances) for calculating the statistics. The larger the samples size the better, and the more confidence we have in the results. Generating an instance of a part (step 1) depends primarily on whether the solid modeller is a B-rep or CSG.
- ✓ For B-rep models, where faces, edges, and vertices are stored, size and form tolerances are applied as variations to the surface equations of the part faces.
- ✓ For CSG models, since part geometry is stored in unevaluated form as an ordered sequence of features (primitive or compound half-spaces), each feature is perturbed first using proper tolerances, then combined to generate a unique instance of the part.
- ✓ Assuming a normal distribution for x , and using $n = 6$, the density distribution function of the variation component x is given by:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

- ✓ If vertices are perturbed as mentioned in step 1 of the preceding algorithm, the solid model must be triangulated before the algorithm begins.
- ✓ This may be unacceptable. In this case, tolerances may be applied directly to equations of surfaces in B-rep or of half-spaces in CSG.

Assembly modelling – interferences of positions and orientation – tolerance analysis mass property calculations – mechanism simulation and interference checking.

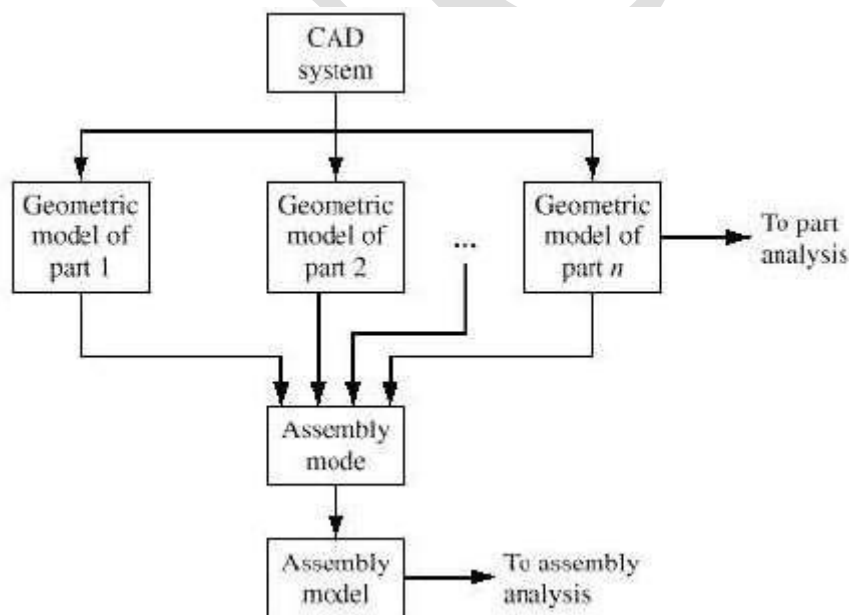
INTRODUCTION

- ✓ Design For Manufacture (DFM) guidelines have been developed for practically every aspect of manufacture, but they may be broadly divided into four groups relating to the general approach to DFM, to selection of manufacturing processes, to design for particular processes, and to assembly.

Assembly

- ✓ An assembly is a collection of independent parts.
- ✓ It is important to understand the nature and the structure of dependencies between parts in an assembly to be able to model the assembly properly.
- ✓ Assembly modeling is considered an extension of part modeling.
- ✓ CAD/CAM systems act assembly modelers. In addition to being geometric modelers, these systems provide users with an assembly mode to create assemblies
- ✓ CAD systems establish a link between an assembly and its individual parts such that designers need only change individual parts for design modification, and the system updates the assembly model automatically.

Creating an assembly model



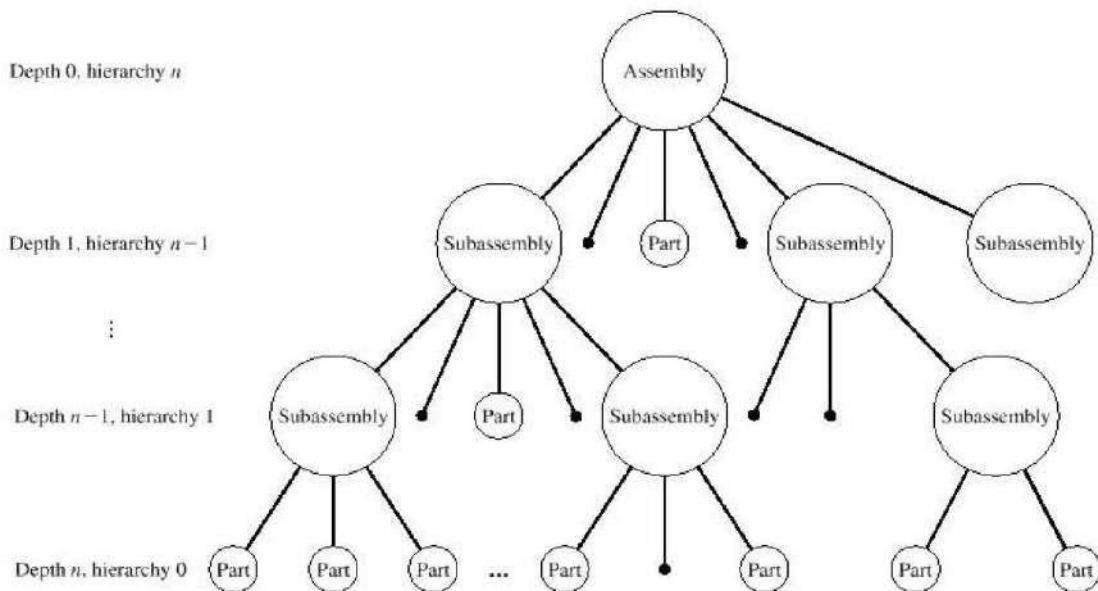
ASSEMBLY MODELING

The need for assembly arises due to the following reasons:

- ✓ The need for dismantling the assembled component during its service life. It is needed when
- ✓ various material combinations are used.
- ✓ Sometimes the physical size needs the assembly modeling.
- ✓ Certain manufacturing process needs the assembly modeling like casting in which pressing needs assembly.

Assembly Tree

- ✓ The most natural way to represent the hierarchical relationships between the various parts of an assembly is an assembly tree.
- ✓ An assembly is divided into several subassemblies at different levels (as the tree depths)
- ✓ Each subassembly at depth $(n-1)$ is composed of various parts. The leaves of the tree represent individual parts or subassemblies.
- ✓ The nodes of the tree represent parts and/or subassemblies, and its root represents the assembly itself.
- ✓ The assembly is located at the top of the tree at depth '0' or at the highest hierarchy 'n' of the assembly sequence.
- ✓ The assembly tree is not unique; other valid assembly sequences can be generated.



Assembly Processes & Planning

✓ The assembly modeling affects the execution of assembly process with certain considerations in designing products related directly to assembly process. These considerations can be classified as follows:

1. Design for ease of handling:

✓ Parts handling may consume almost 80% of assembly time. For handling procedures in general, materials and surfaces should be chosen which are adequate for handling, and all-parts should be dimensionally stable.

2. Design for ease of labelling:

✓ Installing a label on an assembly will be more difficult and labour intensive than installing a part, and there are frequently many unnecessary labels on assemblies.

✓ Consequently, significant labour and cost reduction can be achieved by examining the application of labels for redundancy and repetition.

3. Design for ease of testing:

✓ In general, designers should attempt to move testing of products back as close to their assembly as possible, and reduce the number of tests needed with the overall guidelines of simplification and standardization again.

Before we begin to create an assembly, we should consider the following issues: Assembly planning is a key to creating

✓ successful assemblies, especially the large ones that are typically encountered in practice.

✓ The important issue is not only creating the assembly, but also updating it in the future when design changes are made to the individual parts.

✓ These updates should be done automatically and correctly.

✓ As with the models of individual parts, an assembly model should be fully parametric and flexible.

1. Identify the dependencies between the components of an assembly.

2. Identify the dependencies between the features of each part.

3. Analyze the order of assembling the parts.

Three assembly approaches exist: bottom-up, top-down or a combination of both

Bottom-Up Assembly Approach

✓ The bottom-up approach is most common as it is the traditional and most logical approach.

✓ In this approach, we create the individual parts independently, insert them into an assembly and use the mating conditions to locate and orient them in the assembly as required by the assembly design.

✓ The assembly modeling process itself begins with creating a blank assembly model (file) using the assembly mode of a CAD system.

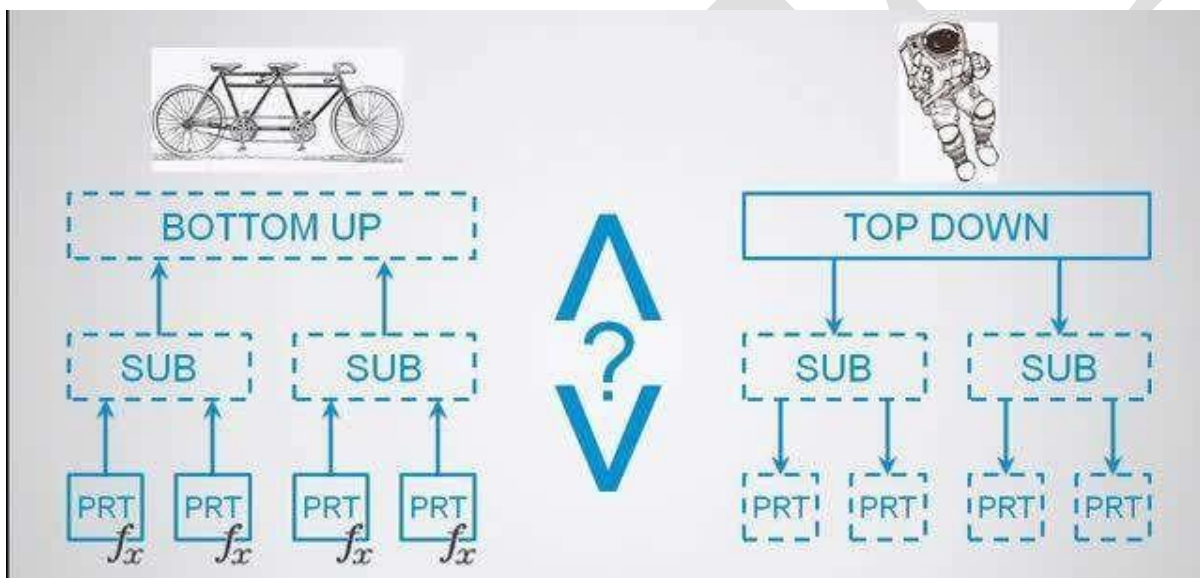
✓ We import (insert) the assembly parts into this model, one at a time.

✓ The first part we insert is known as the base part or the host, on top of which other parts are assembled.

- ✓ We use the proper mating conditions to place and orient each inserted part correctly in the assembly model.

Top-Down Assembly Approach

- ✓ The top-down approach, while good for any size assembly, is ideal for large assemblies consisting of tens of thousands of components.
- ✓ It provides an effective tool and a well-organized approach to managing the design of large assemblies.
- ✓ The top-down assembly approach fosters a systems engineering approach to product design, in which the assembly layout communicates design criteria to subsystem developers, including suppliers.
- ✓ The top-down assembly approach begins with an assembly layout sketch. The assembly layout sketch defines skeletal, space claim, and other physical properties that may be used to define the geometry of and the relationships between components (parts or subassemblies).



INTERFERENCE OF POSITION AND ORIENTATION:

- ✓ The choice of the material and the degree of interference determines the permanency of the joint.
- ✓ Interference fits can be used for assemblies which are required to transmit torque across the joint.
- ✓ Consider an example, in which a pulley would be fitted to a solid shaft without a keyway by using an interference fit.
- ✓ In those cases, the designer has to ensure that the degree of interference is sufficient to avoid any slippage between the components.
- ✓ This is achieved by considering,
 - (i) The pressure should transmit the torque.
 - (ii) Change in shaft outside diameter.
 - (iii) Change in pulley inside diameter.
 - (iv) The required interference.

Assembly mates

- ✓ Individual parts of an assembly are usually created separately using a CAD/CAM system and then merged (assembled) together using mating conditions.
- ✓ Mating conditions or constraints refer to the geometric or mathematical relationships between assembly components to restrict the location of parts in the assembly model.
- ✓ As the mates are defined, the allowable directions of linear or rotational motion of the components are applied.
- ✓ The components can be moved within its degrees of freedom, visualizing the assembly's behaviour.

Inference of Position and Orientation of a part in assembly modelling:

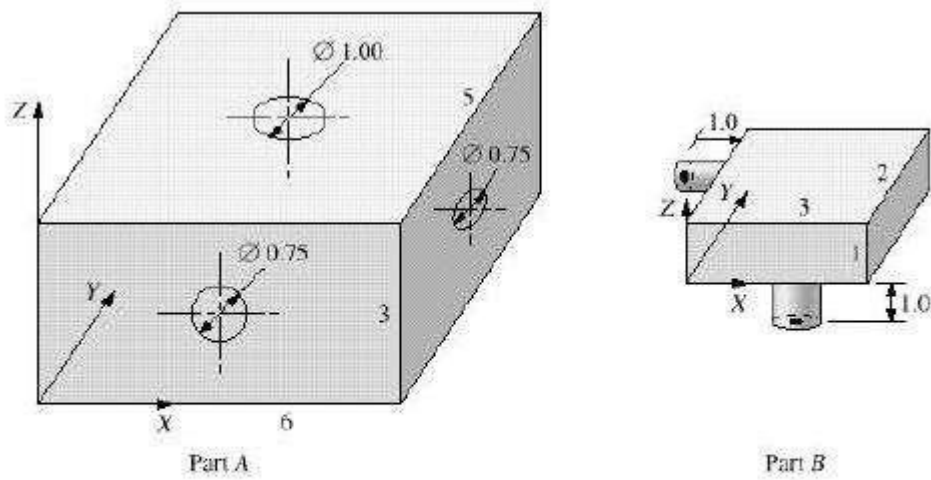
WCS Method

- ✓ Each part has its own database with its own MCS.
- ✓ Typically, the user selects one of the parts as a base part (host) and merges the other parts into it.
- ✓ Alternatively, the user can begin with a blank part as the host.
- ✓ The MCS of the host becomes the global coordinate system that is, the MCS of the assembly.
- ✓ A part MCS becomes a local coordinate system for this part.
- ✓ The final correct position of each part in the assembly is obtained by locating and orienting its MCS properly with respect to the global coordinate system of the assembly.
- ✓ The inference of the position (location) and orientation of a part in an assembly from mating conditions requires computing a 4×4 homogeneous transformation matrix from these conditions.
- ✓ This matrix relates the part's local coordinate system (part MCS) to the assembly's global coordinate system (assemblyMCS).
- ✓ The matrix transforms the coordinates of the geometric entities of the part from its MCS to the assembly MCS.

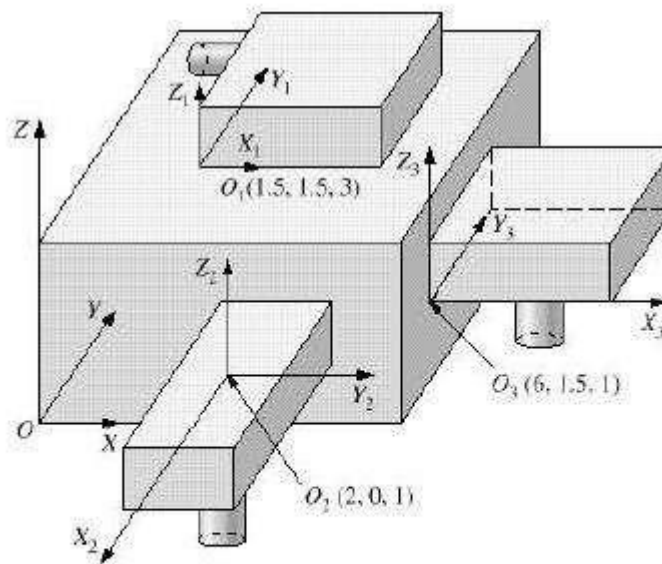
Example: The assembly consists of two parts A and B.

- ✓ Three instances of Part B are used in the assembly. The user first creates the two parts with the MCS of each part.
- ✓ To create the assembly we insert an instance of Part A into a blank assembly file.
- ✓ We use the instance as the assembly base. It is usually beneficial to assign a separate layer for each instance for ease of managing the assembly.

- ✓ To merge the instance of B on top of A, the X1 Y1 Z1 WCS is defined by the user as shown in Figure and then the instance is merged. Similarly, the X2 Y2 Z2 and X3 Y3 Z3 WCSs are defined and shown in the Figure.



(a) Individual parts of the assembly



(b) Assembly

Assembly creation via the WCS method.

- ✓ The transformation matrices to merge the three instances of B into A are given by, respectively:

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & | & 1.5 \\ 0 & 1 & 0 & | & 1.5 \\ 0 & 0 & 1 & | & 3 \\ \hline 0 & 0 & 0 & | & 1 \end{bmatrix}$$

$$T_2 = \left[\begin{array}{ccc|c} 0 & 1 & 0 & 2 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

$$T_3 = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 6 \\ 0 & 1 & 0 & 1.5 \\ 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

Mate Method

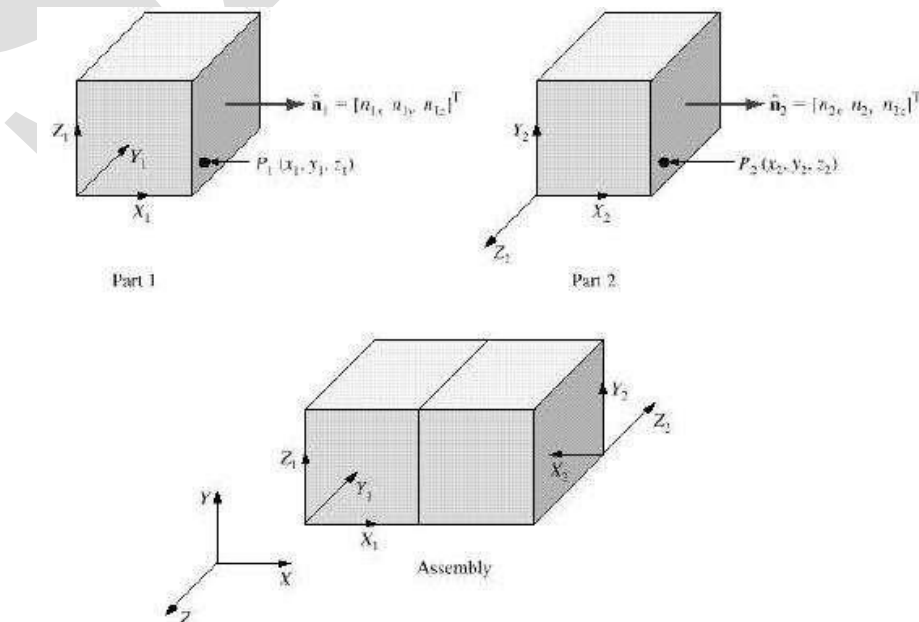
- ✓ In a typical assembly, the mating conditions between two components are not enough by themselves to completely constraint the two components.
- ✓ Locating and orienting parts in their assembly is achieved by specifying mating conditions among them.
- ✓ The mating conditions specify the spatial relationships among the parts. The most common mating conditions are ,
 - Coincident
 - Concentric
 - Tangent
 -
 - Coplanar
 - parallel
 - perpendicular

- ✓ There are six degrees of freedom: three translations along the axes of the assembly MCS and three rotations about the same axes.
- ✓ A part should be fully constrained to allow the creation of the correct assembly, for example, if we mate two faces using the coincident condition only, they can still rotate relative to each other.
- ✓ Mating conditions can be provided interactively with ease because they use simple geometric entities such as faces and centerlines.
- ✓ Let [T1] and [T2] be the transformation matrices from the X1 Y1 Z1 and X2 Y2 Z2 coordinate systems, respectively, to the MCS of the assembly.
- ✓ The unit normals and the two points specifying the mating conditions can be expressed in terms of the MCS (XYZ) system as follows :
- ✓ The mating conditions along with the properties of [T] provide the necessary equations to solve for the 12N - 1 variables.

$$\begin{bmatrix} n_{1x}^a \\ n_{1y}^a \\ n_{1z}^a \\ 0 \end{bmatrix} = T_1 \begin{bmatrix} n_{1x} \\ n_{1y} \\ n_{1z} \\ 0 \end{bmatrix} \quad \begin{bmatrix} x_1^a \\ y_1^a \\ z_1^a \\ 1 \end{bmatrix} = T_1 \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} n_{2x}^a \\ n_{2y}^a \\ n_{2z}^a \\ 0 \end{bmatrix} = T_2 \begin{bmatrix} n_{2x} \\ n_{2y} \\ n_{2z} \\ 0 \end{bmatrix}$$

Coincident

- ✓ The coincident mating condition is applied between two planar faces, or between a planar face and a cylindrical face.
- ✓ Example: The hatched faces are the faces to be mated. Each face is specified by its unit normal vector and any one point on the face with respect to the part MCS.
- ✓ The planar face of Part 1 is specified by the unit normal vector \hat{n}_1 and by the point P1 with respect to the X1 Y1 Z1 coordinate system. Similarly, the planar face of Part 2 is specified by \hat{n}_2 and P2 with respect to the X2 Y2 Z2 coordinate system.



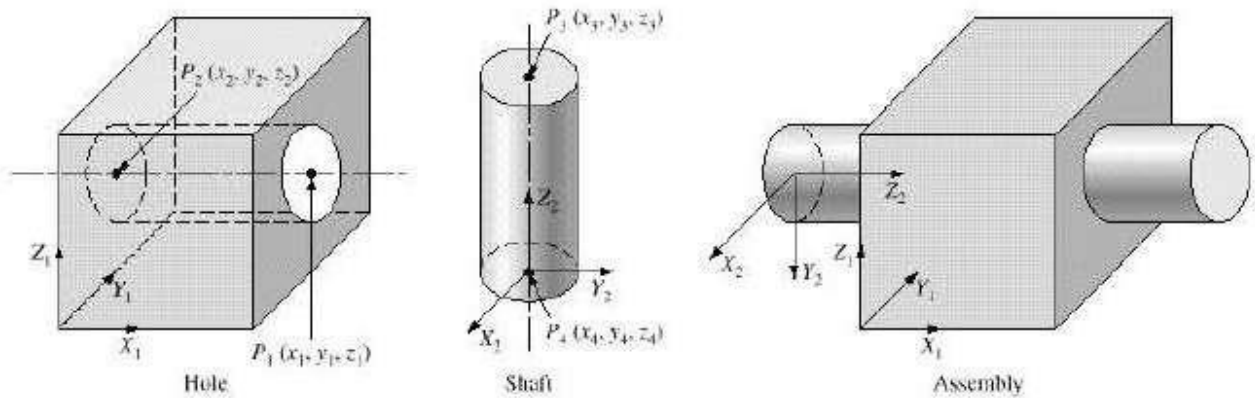
Concentric

- ✓ The concentric mating condition holds between two cylindrical faces: a shaft cylindrical face and a hole cylindrical face.

The mating condition is achieved by forcing the shaft and hole axes to be collinear.

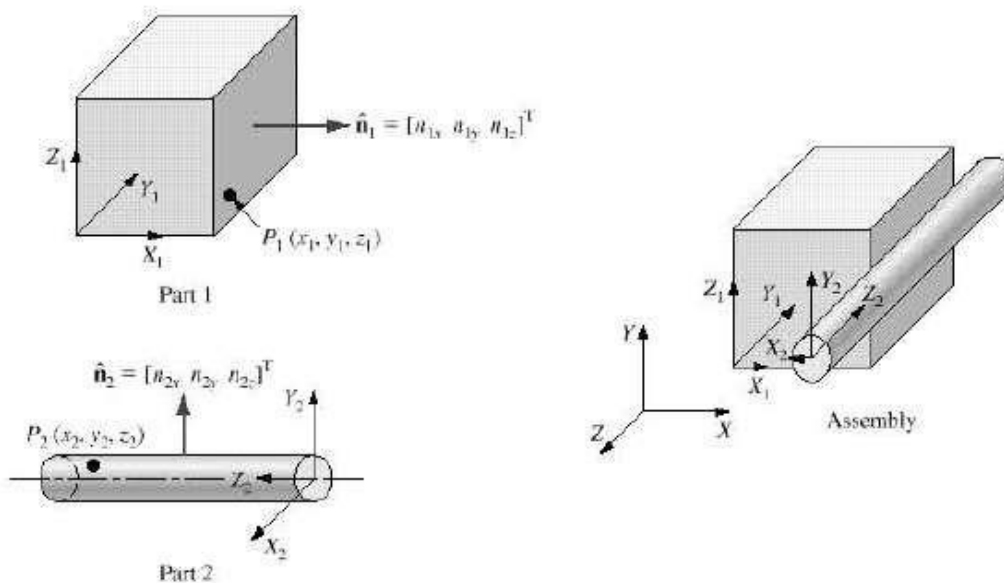
- ✓ Each axis is specified by two points. The hole axis is specified by the two points P1 and P2 defined with respect to the X1 Y1 Z1 MCS.
- ✓ Similarly, the shaft axis is specified by the two points P3 and P4 with respect to the X2 Y2 Z2 MCS.

The tangent mating condition is applicable between two planar/cylindrical or cylindrical/cylindrical



- ✓ faces.

- ✓ The tangent mating condition is achieved by forcing a cylindrical face to be tangent to a planar (flat) face.
- ✓ The difference between the tangent and coincident mating conditions is that the former uses at least one cylindrical face; While the latter uses TWO planar (flat) faces.



- ✓ The coplanar mating condition holds between two planar faces when they lie in the same plane.

- ✓ It is similar to the coincident condition except that the points P1 and P2 are chosen to lie on the two edges to mate.

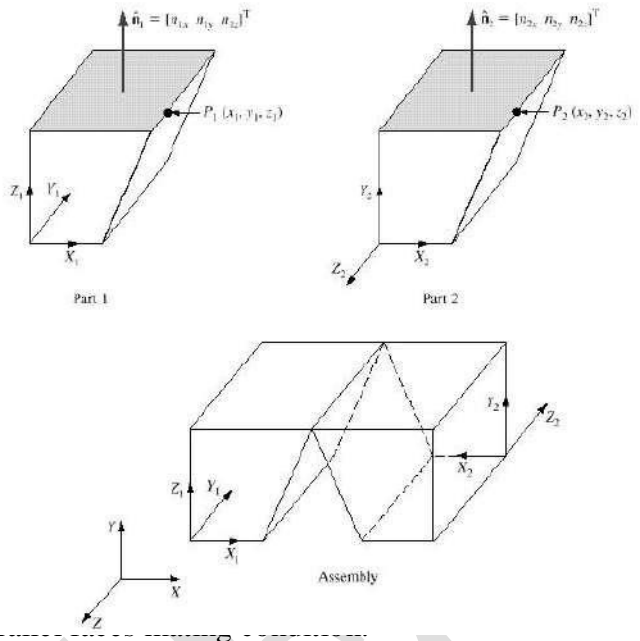
- ✓ The coplanar condition is the compliment (opposite) of the coincident condition and is satisfied by forcing the two normals $n1^{\wedge}$ and $n2^{\wedge}$ to be in the same direction.

- ✓ The parallel faces mating condition is similar to mating faces are not in contact with one another

- ✓ This condition is achieved by forcing the normal

- ✓ If we separate the two mating faces we have a p

- ✓ The perpendicular faces mating condition is specified by requiring two faces to be perpendicular to each other.



VO

TOLERANCE ANALYSIS

- ✓ The tolerance is the permissible variation in the size of a dimension and it is the difference between the upper and lower acceptable limits.

- ✓ The tolerance produces a range of acceptable values for the dimension.

- ✓ If a part size and shape are not within the maximum and minimum limits defined by the part tolerances, the part is not acceptable: it is rejected during inspection and becomes scrap.

Need for tolerances:

- ✓ It is impossible to manufacture a component with exact size.

- ✓ Hence the designer decides how close to the real size is satisfactory.

- ✓ As a result of which most of the manufacturer produces interchangeable parts.

- ✓ The benefits of the interchangeable parts are listed below:

(a) Assembly of parts requires less skilled operators. So it is cheaper and quality is easier to maintain.

(b) The damaged components can be easily replaced by another component with the same range of dimensions.

(c) Parts can be made in large quantities with less demand on the skill of operator.

Tolerance Vs Cost:

- ✓ If the tolerances are too small (tight), the individual parts will cost more to make. If the tolerances are too large (loose), an unacceptable percentage of assemblies may be scrapped (rejected) or require rework. The three main

- ✓

characters that affect the cost of the given tolerance are:

- (i) Tolerance size,
- (ii) Component size,
- (iii) Detail of the features.

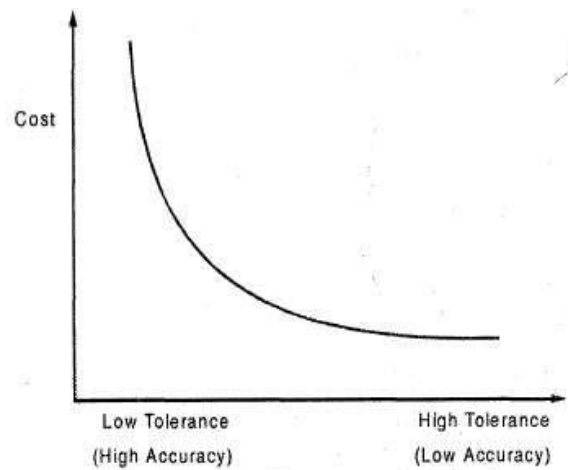


Fig. 4.8. Tolerance vs Cost Graph.

TERMINOLOGY

(ANSI (American National Standards Institute) definitions of tolerance terms)

- ✓ Nominal size is the designation used for the purpose of general identification.
 - o It is the dimension that results from design calculations such as stress or heat transfer analysis. It is usually expressed in common fractions or decimals.
- ✓ Basic size is the theoretical size from which limits of size are derived by the application of allowances and tolerances.
 - o It is the decimal equivalent of the nominal size.
 - o The number of decimal places determines the precision or accuracy required.
 - o For example, if the nominal size of a pipe is 3" the sizes of 3.5", 3.50", and 3.500" are three different sizes when we consider manufacturing and producing these dimensions in the machine shop.
- ✓ Actual size is the measured size of the finished part.
 - o It is a combination of the basic size and its assigned tolerance.
 - o It is the size that results from producing the part including all the manufacturing imperfections.
 - o The actual size must lie within the maximum and minimum allowable values for the size (known as limits); otherwise the part is rejected during inspection, therefore becoming scrap.
- ✓ Tolerance is the total amount by which a dimension may vary.
 - o It is used to determine the permissible limits (maximum and minimum) of the dimensions.
- ✓ Allowance is the difference between the maximum material limits of mating parts.
 - o It is the minimum clearance (positive allowance) or maximum interference (negative allowance) between mating parts.
 - o It is also known as the tightest fit between mating parts—the smallest clearance or the largest interference.

Tolerance presentation:

- ✓ Each and every engineering drawing is subjected to tolerances, with critical dimensions. The tolerances are presented in three ways,

1. Direct tolerance:

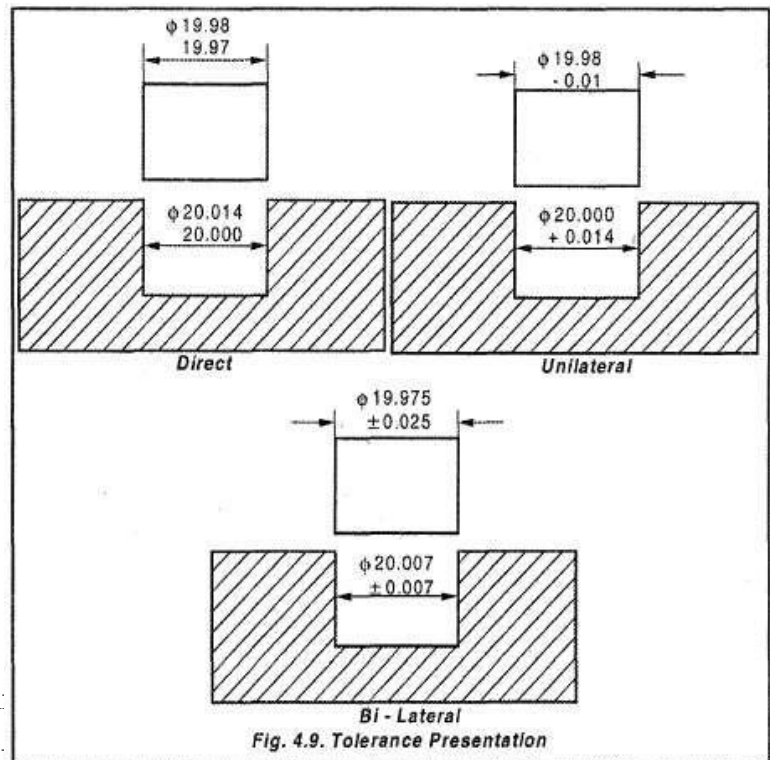
The maximum and minimum acceptable values are given in dimension.

2. Unilateral tolerance:

A unilateral tolerance is a tolerance in which variation is permitted only on one side of the basic size.

3. Bilateral tolerance:

A bilateral tolerance is specified as plus or minus deviation from the basic size.



Fits and Limits

- ✓ Fits between mating parts specify the range of tightness or looseness that may result from the application of specifications.
- ✓ Fits can be identified as cylindrical or location fits.
- ✓ Cylindrical fits apply when an internal member fits in an external member as a shaft in a hole.
- ✓ Location fits are intended to determine only the location of the mating parts. Both cylindrical and location fits can be divided into three types: clearance fits, interference fits, and transition fits.
- ✓ In clearance fit, one part is always loose relative to the other, that is, a shaft is loose in a hole or two stationary parts can be freely assembled or disassembled.
- ✓ In interference fit, one part is forced tight into the other during assembly and an internal pressure between the two results.
 - Interference fits which can transmit torques or forces between mating parts (e.g., a shaft and a pulley) are usually referred to as force fits.
- ✓ A transition fit is a fit which may result in either clearance or interference condition.
- ✓ When utilizing a given fit to calculate the limits and toleranced dimensions of mating parts, a basis or the system for calculating it is adopted and there exist two systems :
 - Hole basis system
 - Shaft basis system.
- ✓ Both systems assume unilateral tolerances h and s for the hole and the shaft, respectively.

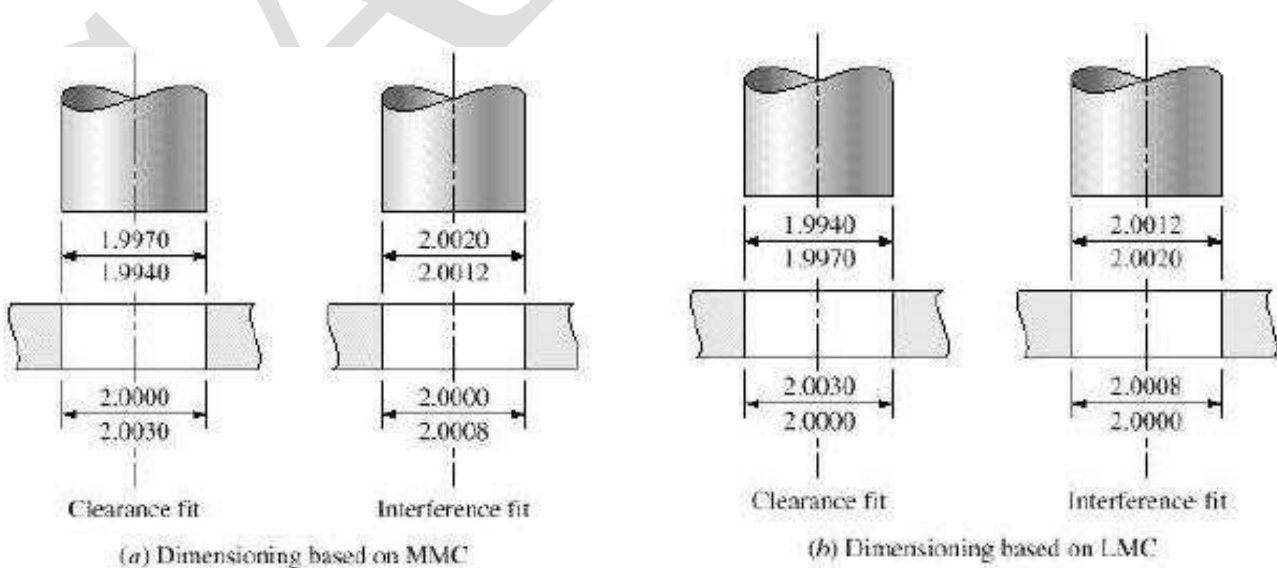
Hole basis system vs shaft basis system:

- ✓ In the Hole basis system, the minimum hole is taken as the basic size, and the allowance and tolerances are applied accordingly.
- ✓ In the shaft basis system, the maximum shaft is taken as the basic size, and the allowance and tolerances are applied accordingly.
- ✓ The basic hole system is the most widely used and recommended system in practice.

- ✓ This is due to manufacturing considerations. It is usually easier to machine shafts for any desired size than holes.
- ✓ Holes are often produced by using standard reamers. Broaches, and other standard tools and standard plug gages are used to check the actual sizes.
- ✓ Shaft basis system is advantageous when several parts having different fits, but one basic size, are mounted on a single shaft.
- ✓ Typically, the textile industry uses the shaft basis system.

Approaches in dimensioning a drawing:

- ✓ The Maximum material condition (MMC) is defined as the condition where the maximum amount of material is preserved in the shaft-hole system.
- ✓ The LMC is the condition where the least amount of material is preserved in the shaft-hole system.
- ✓ The MMC produces the least possibility of assembly because it produces the most dangerous condition (tightest fit).
- ✓ However, once assembled; the resulting assembly best meets its functional requirement.
- ✓ The LMC, on the other hand, produces the best possibility of assembly because it produces the least dangerous condition (loose fit). However, once assembled, the resulting assembly least meets its functional requirement.
- ✓ Thus MMC is usually preferred over the LMC and also while manufacturing, the machinist aims at the principal dimension which is the one shown above the dimensioning line, that is, the maximum shaft or minimum hole in the MMC.
- ✓ Though a machinist produce an oversized hole or an undersized shaft, the parts might still be acceptable, providing the dimensions are within the limits specified by the drawing.
- ✓ In effect, the MMC reduces the amount of scrap, which is a valuable economic gain.

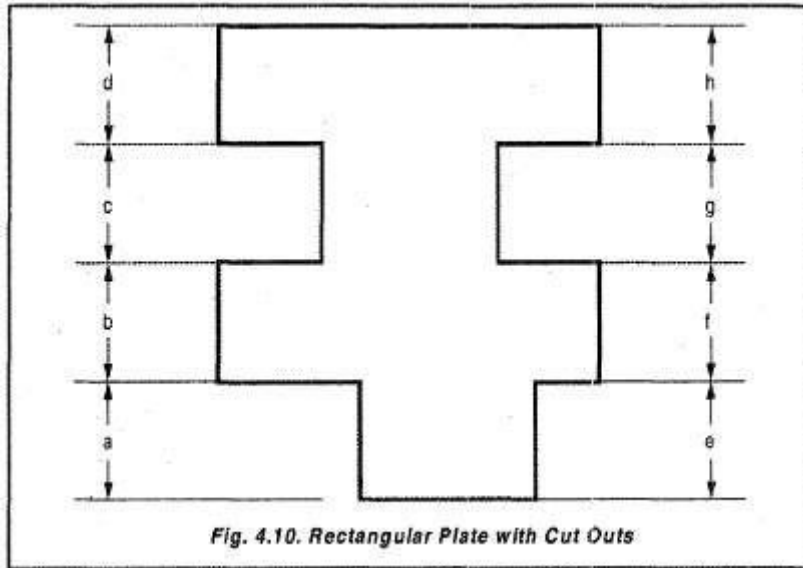


Tolerance allocation:

- ✓ It is very much essential that the component should be fully defined and also, it should not be over

dimensioned.

- ✓ Over dimensioned drawing arises confusion particularly when tolerances are involved.
- ✓ Example: Consider a rectangular plate with cut-outs on two sides as shown in Figure.
- ✓ Consider a general case where the dimensions are from letters 'a' to 'e'.
- ✓



$$e = a + b + c + d - f - g - h$$

giving the limits.

$$e_1 = a_1 + b_1 + c_1 + d_1 - f_2 - g_2 - h_2$$

$$e_2 = a_2 + b_2 + c_2 + d_2 - f_1 - g_1 - h_1$$

Subtract e_1 from e_2

$$e_1 - e_2 = a_1 - a_2 + b_1 - b_2 + c_1 - c_2 + d_1 - d_2 + f_1 - f_2 + g_1 - g_2 + h_1 - h_2$$

Now the Results are:

$$e_1 - e_2 \rightarrow \text{tolerance of } e$$

$$a_1 - a_2 \rightarrow \text{tolerance of } a$$

$$b_1 - b_2 \rightarrow \text{tolerance of } b$$

The above equation can be written as

$$\text{tolerance } (e) = \text{tol } (a) + \text{tol } (b) + \text{tol } (c) + \text{tol } (d) + \text{tol } (f) + \text{tol } (g) + \text{tol } (h)$$

The limits of each dimensions are represented by subscripts, 1 being the maximum and 2 being the minimum values.

The dependent dimension cannot be specified directly. We consider 'e' as the dependent dimension.

✓ Thus the above equation satisfies the definition of tolerance.

Tolerance grade:

✓ A tolerance grade relates the actual tolerance to the component size. The ISO (BS4500) uses 18

✓ tolerance grades which are listed below.

IT16	Sand casting (approx), flame cutting.
IT15	Stamping (approx).
IT14	Die casting, rubber moulding.
IT13	Press work, tube rolling.
IT12	Light press work, tube drawing.
IT11	Drilling, milling, slotting, planing, metal rolling (or) extrusion.
IT10	Engineering works in General.
IT9	Worn capstan or automatic, horizontal or vertical boring machine.
IT8	Centre lathe turning and boring, reaming, capstan or automatic in good condition.
IT7	High quality turning, broaching, honing.
IT6	Grinding, fine honing.
IT5	Ball bearings, machine lapping, diamond or fine boring, fine grinding.
IT4	Gauges, fits of extreme precision produced by lapping.
IT3	Good quality gauges, gap gauges.
IT2	High quality gauges, plug gauges.
IT1	Slip blocks, reference gauges.
IT0	Very fine reference gauges, horology.
IT01	Extremely fine reference gauges, precision horology.

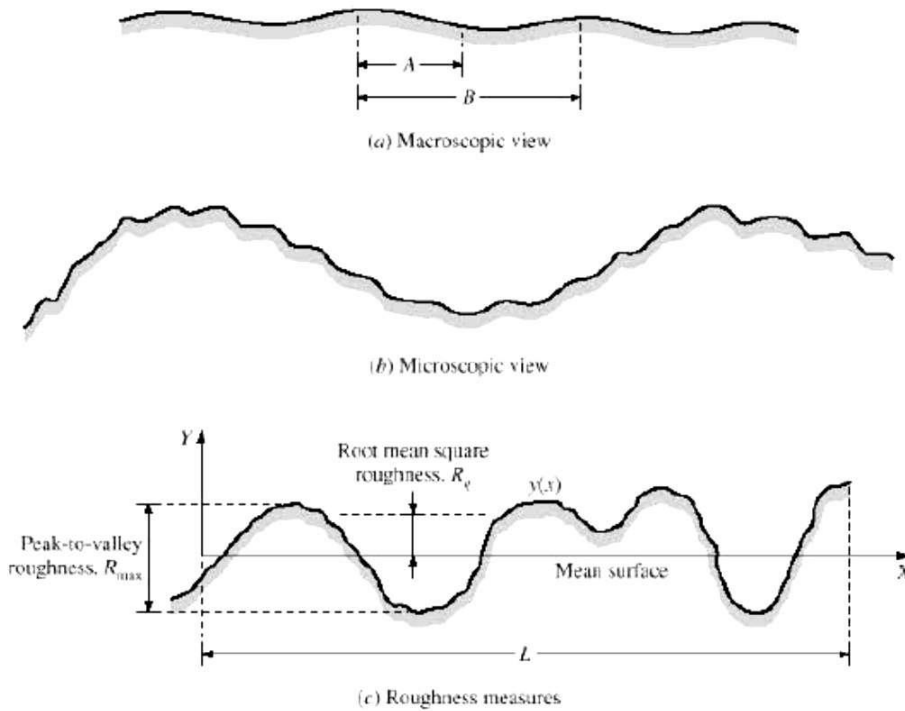
Surface Quality

✓ Surface quality is an important factor that affects the performance of mating parts relative to each other as well as the choice of the manufacturing processes.

✓ Tolerances and surface quality are interrelated in the sense that they are both a direct outcome of manufacturing processes.

- ✓ A manufacturing process, such as lapping and honing, that produces small tolerances (small tolerance grades) also produces smooth surfaces.
- ✓ Therefore, in specifying tolerances, a designer should consider the requirements of surface finish in addition to functional and assembly requirements.

Geometric tolerances



Geometric tolerance specifies the maximum variation of form or position by defining a tolerance zone within which the feature is to be constrained.

- ✓ They are known as traditional or coordinate tolerances methods.
- ✓ They typically control the variability of linear dimensions that describe location, size, and angle.

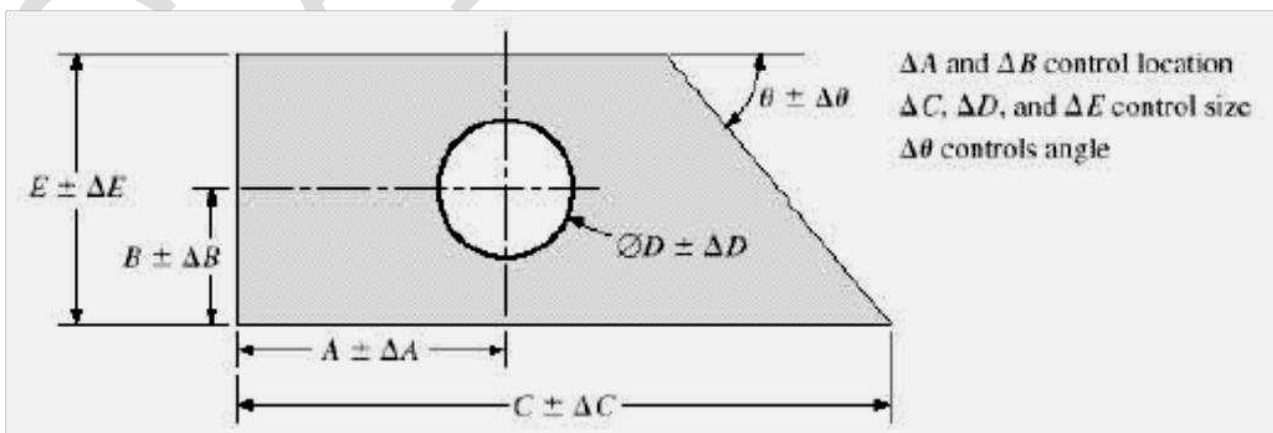


Table 20.5 ANSI symbols for geometric tolerancing.

Feature	Tolerance type	Description	ANSI symbol
Individual features	Form	Straightness	—
		Flatness	
		Circularity (roundness)	
		Cylindricity	
Individual or related feature	Profile	Profile of a line	
		Profile of a surface	
Related features	Orientation	Angularity	
		Perpendicularity	
		Parallelism	
	Location	Position	
		Concentricity	
	Runout	Runout	Circular runout
Total runout			

Table 20.6 ANSI modifying symbols.

Modifying term	ANSI symbol	Modifying term	ANSI symbol
At maximum material condition		Spherical diameter	S \varnothing
Regardless of feature size		Radius	R
At least material condition		Spherical radius	SR
Projected tolerance zone		Reference	()
Diameter	\varnothing	Arc length	

Table 20.7 ANSI form tolerances.

Form tolerance	Specification on a drawing	Interpretation
Straightness		<p>Control surface straightness</p>
		<p>Control axis straightness</p>
Flatness		
Roundness (circularity)		
Cylindricity		

Form tolerance	Specification on a drawing	Interpretation
Profile		<p>Control profile of a line</p>

Table 20.7 ANSI form tolerances. (Continued)

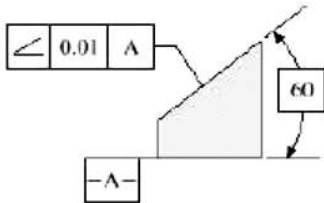
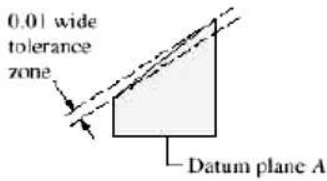
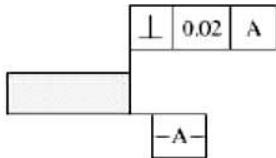
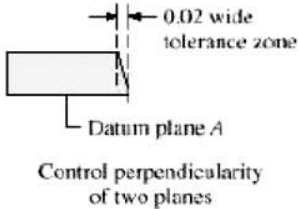
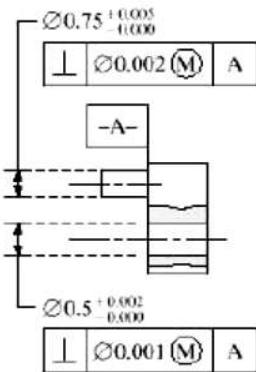
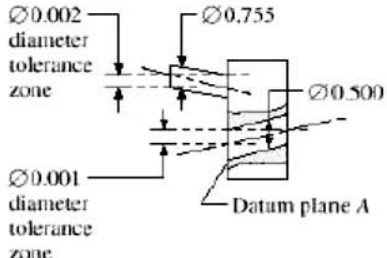

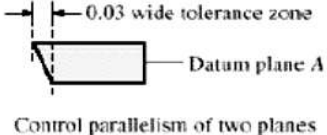
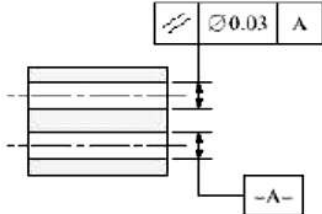
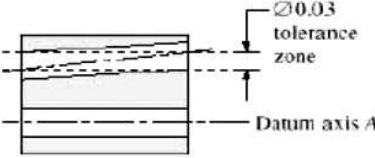
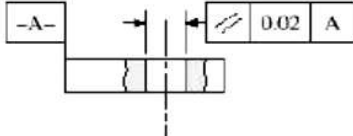
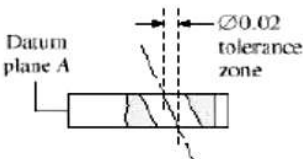
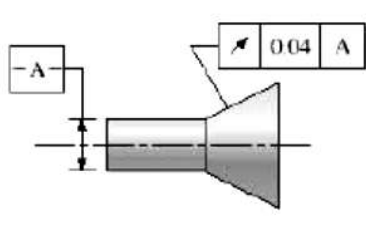
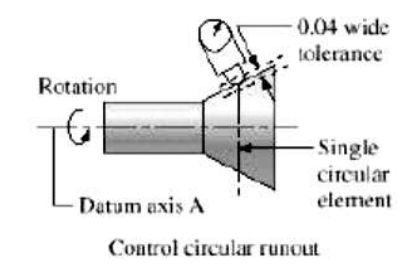
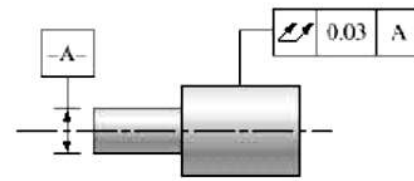
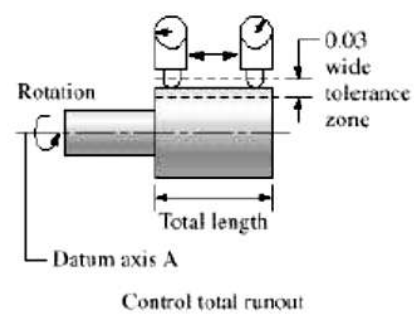
Form tolerance	Specification on a drawing	Interpretation
Angularity		
Perpendicularity		 <p>Control perpendicularity of two planes</p>
		 <p>Control perpendicularity of an axis and a plane</p>
Parallelism		 <p>Control parallelism of two planes</p>
		 <p>Control parallelism of two axes</p>
		 <p>Control parallelism of an axis and a plane</p>

Table 20.7 ANSI form tolerances. (Continued)

Form tolerance	Specification on a drawing	Interpretation
Runout		 <p>Control circular runout</p>
		 <p>Control total runout</p>

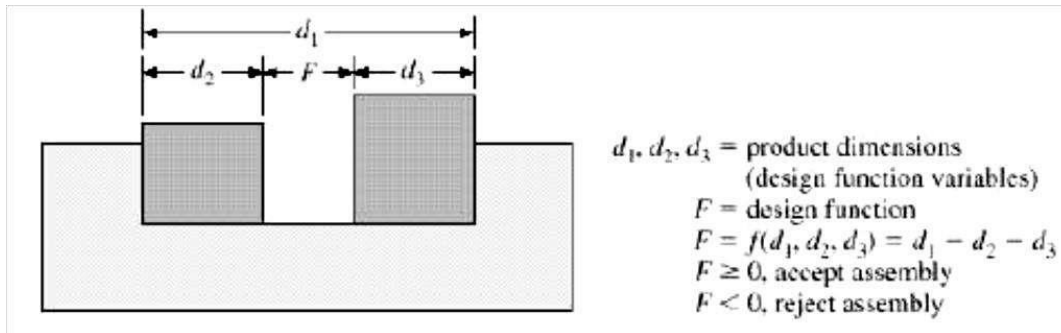
Need for geometrical tolerances:

- ✓ Adding geometric tolerances to a drawing is a very efficient manner of specifying closely the requirements of the designer.
- ✓ It would, be easy to over-tolerance a component; the result will be an increase in the cost of manufacture.
- ✓ They should be applied for all requirements critical to the function and interchangeability of parts, except where it is known that the machinery and manufacturing techniques can be relied upon to achieve the required standards of accuracy.
- ✓ So drawing for any subsequent machining options does not need to include the geometric tolerances (unless they are directly affected by machining process).
- ✓ Geometric tolerances should be applied only where essential, to avoid unnecessary cost increases.
- ✓ In all cases, the tolerance should be as large as possible, subject to meeting the design requirements.
- ✓ It should be emphasized here that geometric tolerancing is quite crucial to achieve the automation of design and manufacturing demanded from the CAD/CAM technology.

Tolerance Analysis

- ✓ Tolerance analysis is defined as the process of checking the tolerances to verify that all the design constraints are met while manufacturing the part.
- ✓ Tolerance analysis is sometimes known as design assurance.
- ✓ The objective of tolerance analysis is to determine the variability of any quantity that is a function of product dimensions and are called design functions. Most often, these qualities are themselves dimensions.

- ✓ Product dimensions and variables that control the behaviour of a design function are called design function variables.
- ✓ The variability of design functions is used to assess the suitability of a particular tolerance specification.



MASS PROPERTY CALCULATIONS

- ✓ The mass properties of an object are a set of useful properties used in various engineering applications.
- ✓ These properties include mass, centroid, first moments, and second moments of inertia.
- ✓ The main difference between mass and geometric properties is the inclusion of the density of the object material in the former.
- ✓ Formally, an object can have a centroid (of its volume), a center of mass (of its mass), and a center of gravity (of its weight) that may differ from each other if the acceleration of gravity and/or the density of the object material is not constant.
- ✓ The study is extended assuming the acceleration due to gravity 'g' and density of a given material 'p' are constants.
- ✓ Mass property algorithm that utilizes these methods are fully automatic and require no additional input except mass attributes.

Mass:

- ✓ Mass of an object is similar to formulating its volume. Once the volume of an object is calculated, it is multiplied by its density to obtain its mass.
- ✓ Mass = density x volume

$$m = \rho \times v$$

Cross sectional area:

- ✓ A cross sectional area is a planar region bounded by a closed boundary. Boundary consists of set of continuous curves joined together.
- ✓ Three properties of a planar region such as its length, its area and its centroid are most important.

From above figure,

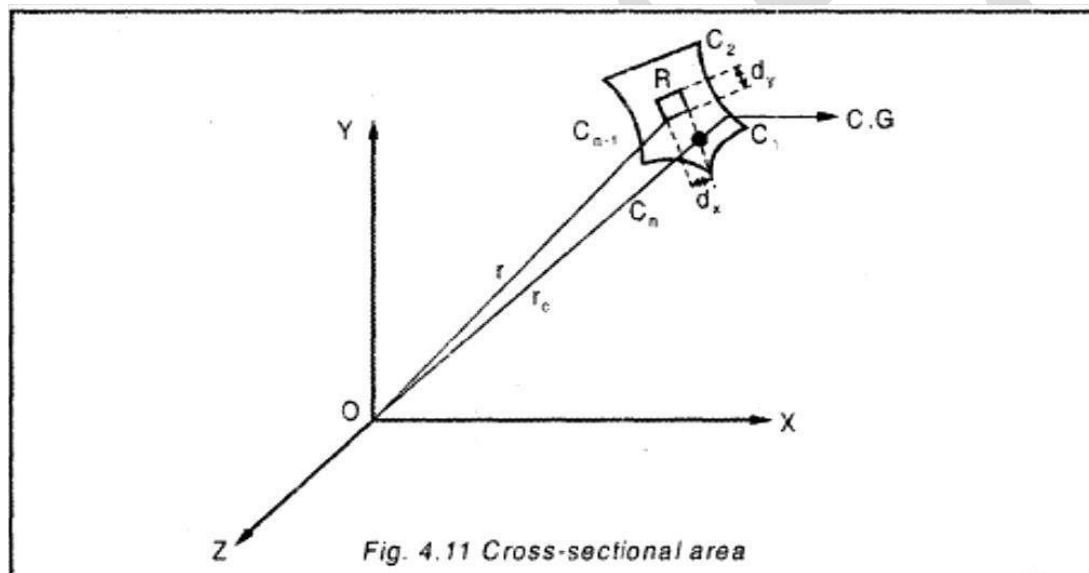
$R \rightarrow$ Object in space

$C_1, C_2, C_{n-1} \rightarrow$ Continuous curves

$X, Y, Z \rightarrow$ Co-ordinate axis

Therefore, length of contour is the sum of lengths of the continuous curves.

$$L = \sum_{i=1}^n L_i$$



To calculate the area A of region R , consider an area element dA of sides dx_L and dy_L

$$A = \iint_R dA$$

$$A = \iint_R dx_L \cdot dy_L \quad \dots(1)$$

Evaluation of integral requires mapping this equation from the local co-ordinate system of region R .

$$\begin{bmatrix} x_L \\ y_L \\ z_L \\ 1 \end{bmatrix} = [T] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \dots(2)$$

End points of curves c_i can be used for this purpose.

We obtain,

$$\begin{bmatrix} x_L \\ y_L \\ z_L \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11}x + r_{12}y + r_{13}z + x_d \\ r_{21}x + r_{22}y + r_{23}z + y_d \\ 0 \\ 1 \end{bmatrix} \quad \dots(3)$$

$Z_6 = 0$ due to region R lies in $X_L Y_L$ plane

$$\Rightarrow d_{xL} = r_{11} dx + r_{12} dy + r_{13} dz \quad \dots(4)$$

$$d_{yL} = r_{21} dx + r_{22} dy + r_{23} dz \quad \dots(5)$$

Substituting equation (4) and (5) in (1), A becomes

$$A = \alpha_1 \iint_R dx dx + \alpha_2 \iint_R dy dy + \alpha_3 \iint_R dz dz + \alpha_4 \iint_R dx dy$$

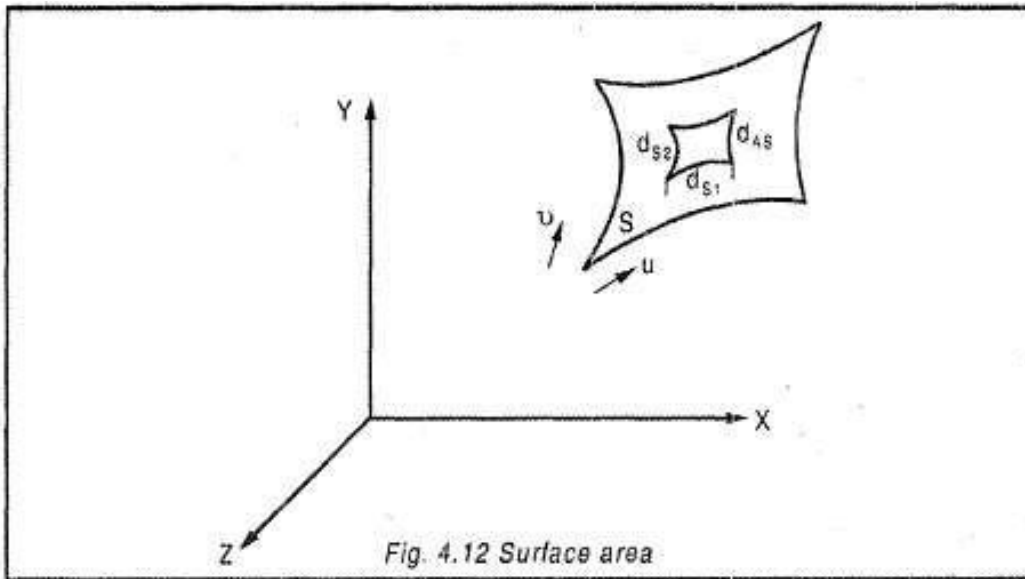
$$+ \alpha_5 \iint_R dx dz + \alpha_6 \iint_R dy dz \quad \dots(6)$$

$$\alpha_1 = r_{11} r_{21}, \alpha_2 = r_{12} r_{22}; \alpha_3 = r_{13} r_{23}$$

$$\alpha_4 = r_{11} r_{22} + r_{12} r_{21}; \alpha_5 = r_{11} r_{23} + r_{13} r_{21}$$

$$\alpha_6 = r_{12} r_{23} + r_{13} r_{22}$$

Surface area



Surface area A_s of surface is given by,

$$A_s = \int_s dA_s$$

To evaluate the integral, dA_s of a rectangle, it must be written as product of length of two perpendicular sides. These sides are taken along u and v direction as ds_1 and ds_2 .

$$A_s = \iint_s ds_1 ds_2 \quad \dots(1)$$

$$ds_1 = \sqrt{p_u \cdot p_u} du \quad \dots(2)$$

$$ds_2 = \sqrt{p_v \cdot p_v} dv \quad \dots(3)$$

substituting equation (2) and (3) in (1) we get

$$A_s = \int_{v_1}^{v_2} \int_{u_1}^{u_2} \sqrt{(p_u \cdot p_u) (p_v \cdot p_v)} du \cdot dv \quad \dots(4)$$

For an object consisting of multiple surface, its total surface area is equal to sum of its individual surface areas,

$$A_s = \sum_{i=1}^n A_{si} \quad \dots(5)$$

Volume:

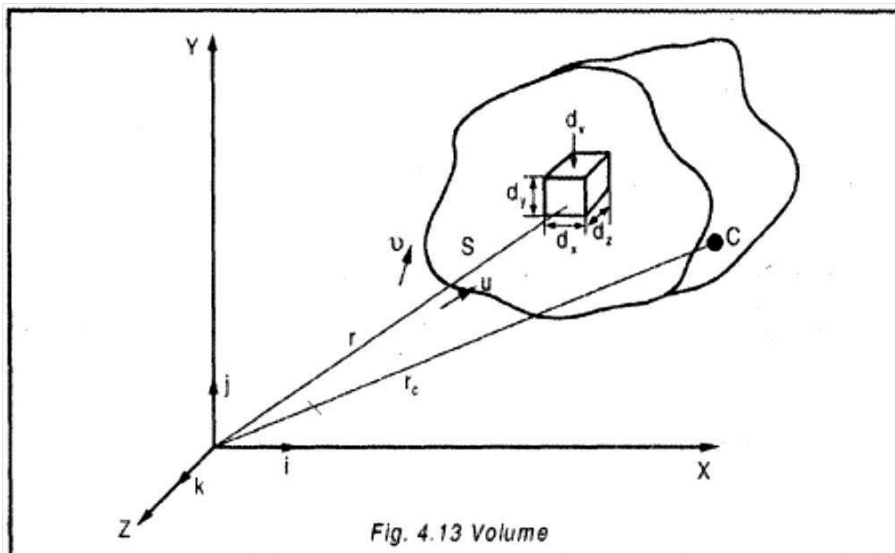


Fig. 4.13 Volume

The volume V can be expressed as triple integral by integrating the volume element dv ,

$$V = \iiint_v dv = \iiint_v dx dy dz$$

This volume can be changed into a surface integral using Gauss divergence theorem.

$$\iiint_v \nabla \cdot F dv = \iint_s F \cdot \hat{n} ds \quad \hat{n} \rightarrow \text{unit normal vector}$$

$$\nabla = \frac{\partial}{\partial x} \hat{i} + \frac{\partial}{\partial y} \hat{j} + \frac{\partial}{\partial z} \hat{k}$$

$$v = \iiint_v dv = \iint_s z (\hat{k} \cdot \hat{n}) dA_s$$

$$v = \int_{u_1}^{u_2} \int_{v_1}^{v_2} z (\hat{k} \cdot \hat{n}) \sqrt{(p_u \cdot p_u) (p_v \cdot p_v)} du dv$$

Centroid of the object is located by vector r_c

$$r_c = \frac{\iiint_v r dv}{\iiint_v dv} = \frac{\iiint_v r dv}{v}$$

Centroid:

From the above equation, the center of mass of an object by replacing the volume v by the mass m ,

$$r_c = \frac{\iiint r dm}{m}$$

By substituting ρ_v for m in above equation, the center of mass is coincident with center of volume.

Calculation of center of gravity location

- ✓ It is also called the "center of mass" of the object.
- ✓ The single point where the static balance moments about three mutually perpendicular axes are all zero.
- ✓ Centre of gravity is the centroid of the volume of object, if the object is homogeneous.
- ✓ The point about which the object rotates in free space.
- ✓ The point through which the force of gravity (centre of gravity of composite masses) is calculated from moments taken about the origin. Component distances for centre of gravity location may be either positive or negative and in fact their polarity depends on the choice of reference axis location.

Calculating moment of inertia:

- ✓ Moment of inertia is applied to rotational motion, whereas inertia is applied to linear motion. Inertia is the tendency to remain at rest or moving in a straight line at same velocity.
- ✓ Moment of inertia depends on distribution of mass in an object. The greater the distance from the center of rotation, greater is moment of inertia.

Choosing the reference Axis location in calculating MOI:

- ✓ Reference axis is important to define Centre of gravity (C.G) of an object. Only one axis is necessary to define moment of inertia.
- ✓ Axis of rotation is also chosen to define it.
- ✓ If the object flies in space, then the axis is called "principal axis".

Polarity of moment of inertia:

- ✓ Values of centre of gravity can either be positive or negative but the value for moment of inertia is always positive.

$$I = Mr^2$$

where

$I \rightarrow$ Moment of Inertia. (MI)

$M \rightarrow$ Mass of element

$r \rightarrow$ Distance from point mass to reference axis

First moment of inertia

It is defined as the moment of an object property with respect to a given plane. For a lumped mass, the first moment of mass about a given plane is equal to the product of the mass and its perpendicular distance from the plane.

Therefore,

$$M_{xy} = \iiint_m z \, dm = \rho \iiint_v z \, dv$$

$$M_{xz} = \iiint_m y \, dm = \rho \iiint_v y \cdot dv$$

$$M_{yz} = \iiint_m x \, dm = \rho \iiint_v x \, dv$$

Substituting

$$A_s = \int_{v_1}^{v_2} \int_{u_1}^{u_2} \sqrt{(p_u \cdot p_u)(p_v \cdot p_v)} \, du \, dv \text{ in above equation}$$

$$M_{xy} = \rho V_{zc} = mz_c$$

$$M_{xz} = \rho V_{yc} = my_c$$

$$M_{yz} = \rho V_{xc} = mx_c$$

$$\therefore M = \rho \cdot v$$

Second moment of Inertia

It is the product of mass and the square of the perpendicular distance between the mass and the axis. Therefore second moment of inertia of a distributed mass about its axis can be written as

$$I_{xx} = \iiint_m (y^2 + z^2) dm = \rho \iiint_v (y^2 + z^2) dv \quad \dots(1)$$

$$I_{yy} = \iiint_m (x^2 + z^2) dm = \rho \iiint_v (x^2 + z^2) dv \quad \dots(2)$$

$$I_{zz} = \iiint_m (x^2 + y^2) dm = \rho \iiint_v (x^2 + y^2) dv \quad \dots(3)$$

With the help of Gauss Divergence theorem, inertia integrals can be changed to surface integrals.

$$\text{For } I_{xx} \quad \mathbf{F} = \left(y^2 z + \frac{z^3}{3} \right) \mathbf{k}$$

$$I_{yy} \quad \mathbf{F} = \left(x^2 z + y^2 z \right) \mathbf{k}$$

$$I_{zz} \quad \mathbf{F} = \left(x^2 z + \frac{z^3}{3} \right) \mathbf{k}$$

Substituting this in equation (1), (2), (3) we get

$$I_{xx} = \rho \iint_s z \left(y^2 + \frac{z^2}{3} \right) (\hat{\mathbf{k}} \cdot \hat{\mathbf{n}}) dA_s$$

$$I_{yy} = \rho \iint_s z \left(x^2 + \frac{z^2}{3} \right) (\hat{\mathbf{k}} \cdot \hat{\mathbf{n}}) dA_s$$

$$I_{zz} = \rho \iint_s z (x^2 + y^2) (\hat{\mathbf{k}} \cdot \hat{\mathbf{n}}) dA_s$$

Using $dA_s = ds_1 ds_2$ along with above equation we get,

$$I_{xx} = \rho \int_{v_1}^{v_2} \int_{u_1}^{u_2} zk \left(y^2 + \frac{z^2}{3} \right) (\hat{k} \cdot \hat{n}) du dv \quad (\text{where}$$

$$ds_1 = \sqrt{p_u \cdot p_u} \cdot du$$

$$I_{yy} = \rho \int_{v_1}^{v_2} \int_{u_1}^{u_2} zk \left(x^2 + \frac{z^2}{3} \right) (\hat{k} \cdot \hat{n}) du dv \quad ds_2 = \sqrt{p_v \cdot p_v} \cdot dv)$$

$$I_{zz} = \rho \int_{v_1}^{v_2} \int_{u_1}^{u_2} zk (x^2 + y^2) (\hat{k} \cdot \hat{n}) du \cdot dv$$

For closed object with multiple surfaces, integrals are summed to give,

$$I_{xx} = \rho \sum_{i=1}^n \int_{v_1}^{v_2} \int_{u_1}^{u_2} z k_i \left(y^2 + \frac{z^2}{3} \right) (\hat{k} \cdot \hat{n}_i) du \cdot dv$$

$$I_{yy} = \rho \sum_{i=1}^n \int_{v_1}^{v_2} \int_{u_1}^{u_2} z k_i \left(x^2 + \frac{z^2}{3} \right) (\hat{k} \cdot \hat{n}) du \cdot dv$$

$$I_{zz} = \rho \sum_{i=1}^n \int_{v_1}^{v_2} \int_{u_1}^{u_2} z k_i (x^2 + y^2) (\hat{k} \cdot \hat{n}) du \cdot dv$$

Properties of composite objects:

- ✓ For an object with multiple sub objects, properties can easily be calculated. The original object is referred to as composite object.
- ✓ Therefore the volume of the object is the sum of the volume of its sub objects.

$$V = \sum_{i=1}^n V_i$$

where V , V_i and n are the volume and number of the object respectively.

$$V = \sum_{i=1}^n V_i$$

where V , V_i and n are the volume and number of the object respectively.

The centroid is given by,

$$r_c = \frac{\sum_{i=1}^n r_{ci} V_i}{\sum_{i=1}^n V_i} = \frac{\sum_{i=1}^n r_{ci} V_i}{V}$$

$$x_c = \frac{\sum_{i=1}^n x_{ci} V_i}{V}; \quad y_c = \frac{\sum_{i=1}^n y_{ci} V_i}{V}; \quad z_c = \frac{\sum_{i=1}^n z_{ci} V_i}{V}$$

where $r_{ci} \rightarrow$ position vector of centroid

Therefore first moment,

$$M_{xy} = \sum_{i=1}^n (M_{xy})_i$$

$$M_{xz} = \sum_{i=1}^n (M_{xz})_i$$

$$M_{yz} = \sum_{i=1}^n (M_{yz})_i$$

Second moment is written as

$$I_{xx} = \sum_{i=1}^n (I_{xx})_i$$

$$I_{yy} = \sum_{i=1}^n (I_{yy})_i$$

$$I_{zz} = \sum_{i=1}^n (I_{zz})_i$$

Product of inertia is written as,

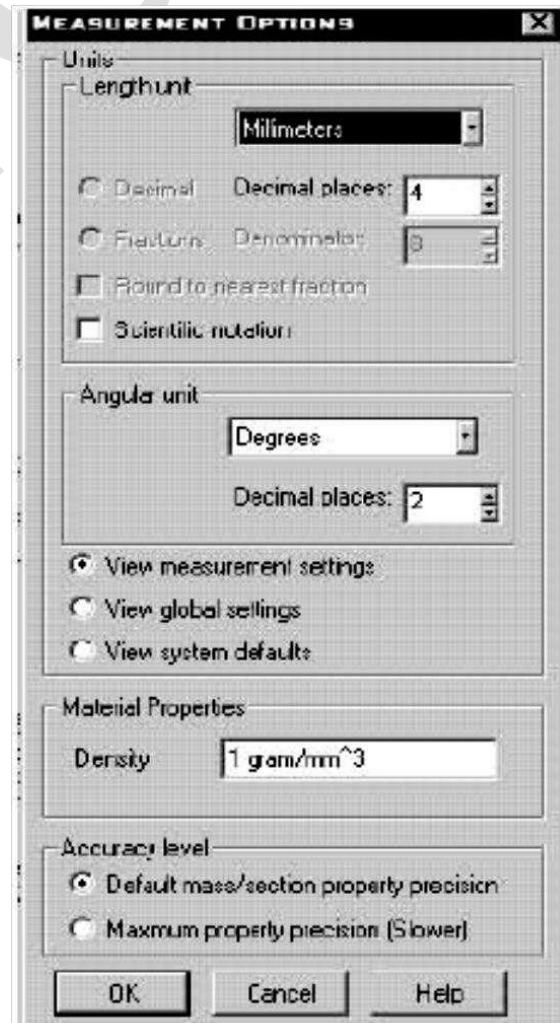
$$I_{xy} = \sum_{i=1}^n (I_{xy})_i$$

$$I_{xz} = \sum_{i=1}^n (I_{xz})_i$$

$$I_{yz} = \sum_{i=1}^n (I_{yz})_i$$

Mass Properties on CAD/CAM Systems

- ✓ CAD systems provide users with options to control the calculations of the mass properties of CAD models.
- ✓ These options include the choice of measurement units for length (mm or inches) and angles (radians or degrees), the choice of decimal places (two, three, four, and so forth), and the choice of density units (lb/in³ or g/mm³).
- ✓ It is the user's responsibility to ensure the correctness of units of the density so the proper units for the mass and inertial properties are produced.



MECHANISM SIMULATION

- ✓ Simulation is a key technology for designing new products, developing new manufacturing processes and evaluating the performance of mechanisms.
- ✓ Mechanisms are represented by connected or assembled joints of 3D solids with defined constraints, actuators, springs, dampers or forces.
- ✓ In addition to pure geometric and connection information, mass, moment of inertia, gravity, force and torque information are also modeled and used to predict the motion.
- ✓ Simulation models are sets of mathematical equations representing the behaviour of the system in a physical domain of interest.
- ✓ Recent advances in Computer Aided Design (CAD) have allowed creation of virtual prototypes that represent mechanical systems at any scale and complexity.
- ✓ By using CAD/CAE systems for virtual prototyping, the user has to be a skilful engineer because these systems only allow the visualisation of the simulation results as 2D drawings, plots or graphs and require a mental transformation of 2-D objects into dynamic 3-D objects.
- ✓ Virtual Reality technologies facilitate the development of new industrial applications by providing advanced visualization capabilities and multi-sensorial human interaction interfaces.
- ✓ Mechanical systems are described by means of representing them as systems of rigid bodies connected by various kinematic pairs and force elements.
- ✓ Linkage is computer aided design software used for quick prototyping of linkage mechanisms.
- ✓ The number of operations needed to add a link and get it connected to other links in the mechanism has been minimized to the lowest number possible, making this program ideal for "throwing together" a working machine.
- ✓ Motion simulation provides complete, quantitative information about the kinematics - including position, velocity, and acceleration, and the dynamics – including joint reactions, inertia forces, and power requirements, of all the components of a moving mechanism.
- ✓ In additional importance, the results of motion simulation can be obtained virtually at no additional time expense, because everything needed to perform motion simulation has been defined in the CAD assembly model already, and just needs to be transferred to the motion simulation program.
- ✓ A common mistake made in creating motion simulation models is selecting improper joints to connect bodies.
- ✓ Introducing improper joints creates an invalid or inaccurate model that does not simulate the true behavior of the mechanical system.

Advantage of mechanism simulation:

1. Models behavior of a real hardware system using an approximation in software.
2. Mathematical formulae are used to emulate and predict physical phenomena.
3. Primary method used for evaluation of systems before manufacture.
4. Identification and tuning of critical components (optimization).

5. Time and money are saved by removing faults before manufacture.

Disadvantages of mechanism simulation:

1. Building hardware prototypes is impractical for large systems.
2. Takes too long and costs too much while simulating.

INTERFERENCE CHECKING

- ✓ Component interference, misaligned and mismatched holes and fasteners and incorrect tolerancing of manufactured parts are the main causes of high rework and scrap costs on the assembly floor.
- ✓ Interference avoidance modification seeks to access any interference that may exist as a result of basic design and then make proper modifications to remove the interference at the design stage.
- ✓ Thus, interference avoidance modification can be divided into two phases:
 1. Interference checking to discover existing interference, and then
 2. Making modifications to remove the interference, if interference does exist.
- ✓ Generally, there are three types of interference in fixture design:
 1. Interference among fixture components, including locators, clamps, and supports.
 2. Interference between fixture components and a workpiece.
 3. Interference between fixture components and a machining envelope.
- ✓ Once the interference has been detected with interference checking functions, proper modifications should be applied to the basic design to avoid or eliminate interference.
- ✓ Possible solutions include changing the size of fixture components, changing the shape of fixture components, changing the orientation of fixture components, changing the distance between interfering parts in the assembly, combining fixture components.

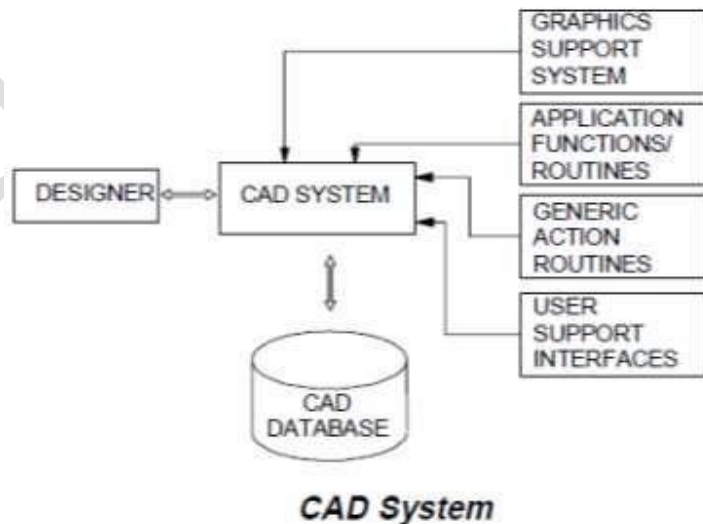
Example

- ✓ When the interference occurs on the side of the support where the support collides with the protrusion of the workpiece, the solution may be decreasing the side dimension of the support.
- ✓ If the interference occurs on the bottom of the support, there are several possible solutions. When the interference area is not large, the low cross portion of the support can be cut to eliminate the interference.

Standards for computer graphics - Graphical Kernel System (GKS) - Standards for exchange images - Open Graphics Library (Open GL) - Data exchange standards - IGES, STEP, CALS, etc - Communication standards.

INTRODUCTION

- ✓ The heart of any CAD model is the component database.
- ✓ It includes the graphics entities like points, lines, arcs, circles etc. and the co- ordinate points, which define the location of these entities.
- ✓ In designing a data structure for CAD database the following factors are to be considered:
 - The data must be neutral
 - The data structure must be user-friendly
 - The data must be portable.
- ✓ In order to achieve the above requirements, some type of standardization has to be followed by the CAD software designers.
- ✓ The basic elements associated with a CAD system are:
 - Operator (user)
 - Graphics support system
 - Other user interface support system
 - Application functions
 - Database
- ✓ The reasons for evolving a graphic standard thus include:
 - Need for exchanging graphic data between different computer systems.
 - Need for a clear distinction between modeling and reviewing aspects.



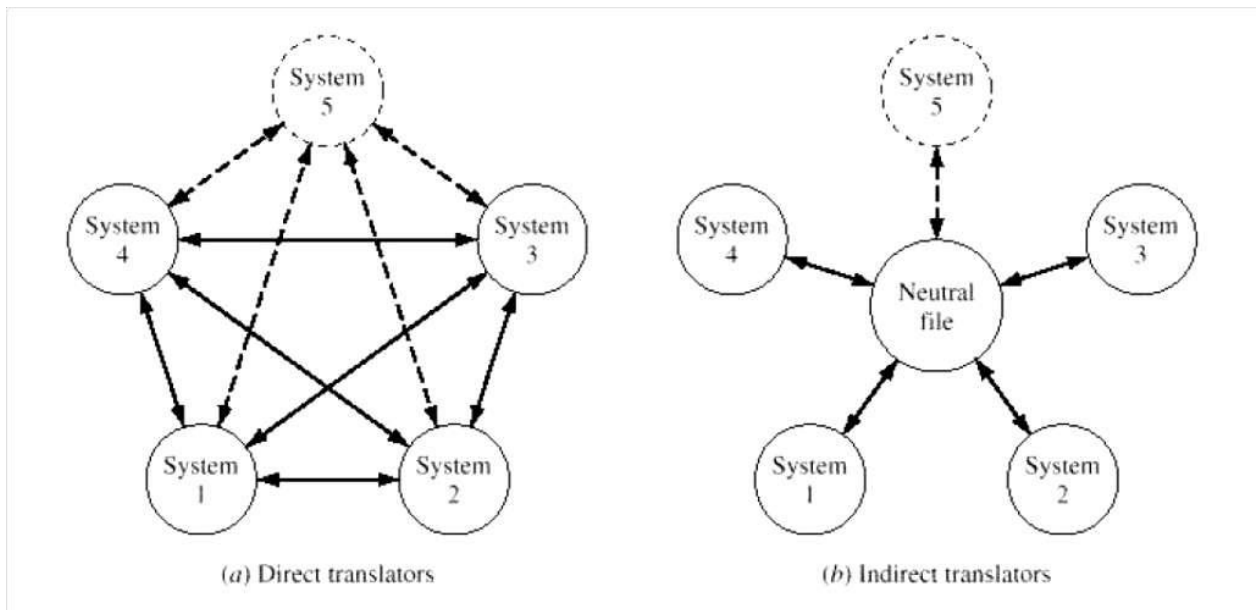
- ✓ Fundamental incompatibilities among entity representations greatly complicate exchanging modeling data among CAD/CAM systems.
- ✓ Even simple geometric entities such as circular arcs are represented by incompatible forms in many systems.
- ✓ Some systems use NURBS (Non-uniform rational Basis spline) to represent them, while others use the usual parametric representation.
- ✓ The transfer of data between dissimilar CAD/CAM systems must embrace the complete description of a product stored in its database.
- ✓ Four types of modeling data make up this description.
 - shape,
 - non-shape,
 - design, and
 - manufacturing data.
- ✓ Shape data consists of both geometrical and topological information.
- ✓ Non-shape data includes shaded images and measuring units of the database.
- ✓ Design data includes FEM/FEA. Manufacturing data includes tolerancing and bill of materials.
- ✓ When similar CAD/CAM systems are operated by both parties, no difficulty of exchange exists because the files that store modeling data are compatible. However many dissimilar CAD/CAM systems are in existence and thus data communication problems arise.

Translators

The two methods of exchanging the data among the different CAD/CAM systems:

Direct Translators

- ✓ Direct translators convert data directly in one step.
- ✓ The direct translator entails translating the modeling data directly from one CAD/CAM system format to another usually in one step.
- ✓ This solution converts the data (database) format from one native format to another.
- ✓ It requires knowledge of both the native formats.
- ✓ Direct translators provide a satisfactory solution when only a small number of systems are involved, but as this number increases the number of translator programs that need to be written becomes prohibitive.



Indirect translators

- ✓ It adopts the philosophy of creating a neutral database structure (also called a neutral file) which is independent of any existing or future CAD/CAM system.
- ✓ This structure acts as an intermediary and a focal point of communication among the dissimilar database structures of CAD/CAM systems.
- ✓ This solution converts native formats to a neutral format that all CAD/CAM systems can interpret and understand.
- ✓ Indirect translators utilize some neutral file format, which reflects the neutral database structure, with each system having its own pair of processor to transfer data to and from this neutral format.

Comparison between direct and indirect translators

- ✓ For 'direct' translators, eight additional translators (shown dashed) are needed to accommodate the addition of System 5.
- ✓ Direct translators run more quickly than the indirect ones, and the data files they produce are smaller than the neutral files created by indirect translators.
- ✓ As for indirect translators, they do not suffer from the increasing number of programs that have to be written as in the case of direct ones.
- ✓ Moreover, indirect translator philosophy provides stable communication between CAD/CAM systems, protects against system obsolescence, and eliminates dependence on 1 single systemsupplier.
- ✓ A side benefit of neutral files is that they can potentially be archived.
- ✓ Some companies in the aerospace industry for example need to keep CAD/CAM databases for 20 to 50 years.
- ✓ Indirect translators based on a standard neutral file format are now the common practice, while direct translators are seldom used.

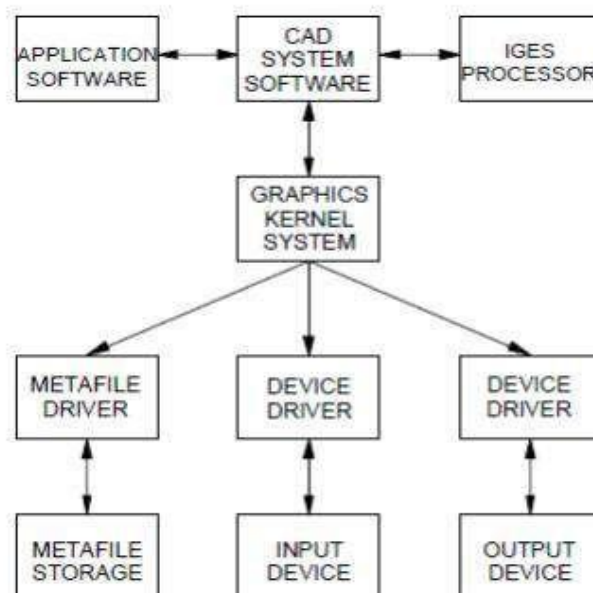
Timeline

Various attempts to develop a graphics standard resulted in the following developments.

- ✓ A Graphic Standards Planning Committee (GSPC) was formed in 1974 by ACM-SIGGRAPH (Association of Computing Machinery's Special Interest Group on Graphics and Interactive Techniques).
- ✓ A committee for the development of computer graphics standard was formed by DIN in 1975.
- ✓ IFIP organized a workshop on Methodology in Computer Graphics in 1976.
- ✓ A significant development in CAD standards is the publication of Graphical Kernel System (GKS) in 1982.

Graphical Kernel System (GKS)

- ✓ GKS (Graphical Kernel System) is an ANSI and ISO standard.
- ✓ GKS standardizes two dimensional graphics functionality at a relatively low level.
- ✓ The primary purposes of the standard are:
 - To provide for portability of graphics application programs.
 - To aid in the understanding of graphics method by application programmers.
 - To provide guidelines for manufacturers in describing useful graphics capabilities.
- ✓ The GKS (ANSI X3.124-1985) consists of three basic parts:
 - i. An informal exposition of contents of the standard, which includes such things as positioning of text, filling of polygons etc.
 - ii. A formalization of the expository material outlined in (i) by way of abstracting the ideas into functional descriptions (input/output parameters, effect of each function etc.
 - iii. Language bindings, which are the implementations of the abstract functions, described in (ii) in a specific computer language like FORTRAN, Ada or C.



GKS Implementation in a CAD Workstation

✓ The features of GKS include:

- i. Device independence: The standard does not assume that the input or output devices have any particular features or restrictions.
- ii. Text/Annotations: All text or annotations are in a natural language like English.
- iii. Display management: A complete suite of display management functions, cursor control and other features are provided.
- iv. Graphics Functions: Graphics functions are defined in 2D or 3D. The drivers in GKS

✓ also include metafile drivers.

✓ Metafiles are devices with no graphic capability like a disc unit.

✓ The GKS always works in a rectangular window or world coordinate system. The window also defines a scaling factor used to map the created picture into the internal co-ordinate system of GKS called normalized device co-ordinates.

✓ Windows and view ports can then work in this co-ordinate system. GKS offers two routines to define the user created pictures.

✓ o Primitive functions and attribute functions. Examples of primitive functions are:

- POLYLINE to draw a set of connected straight-line vectors
- POLYMARKER to draw a set of markers or shapes
- FILL AREA to draw a closed polygon with interior fill
- TEXT to create characters
- GDP (Generalized Drawing Primitive) to specify the standard drawing entities like circle, ellipse etc.

The syntax of commands for these primitives is:

POLYLINE	(n, X, Y)
POLYMARKER	(n, X, Y)
FILLAREA	(n, X, Y)
TEXT	$(x, y, \text{'test string'})$

where,

$n \rightarrow$ number of data points

$x \rightarrow$ is a single and X an array of x - coordinates

$y \rightarrow$ is a single and Y an array of y - coordinates

'test string' \rightarrow The text plotted

- ✓ The attribute functions define the appearance of the image e.g. color, line- type etc. Current level of GKS is GKS-3D, which is an extension to GKS that allows the creation of 3-Dobjects.

Graphical Kernal System' draws graphical elements into a window defined using a real-valued user

Example

If $X = (2, 5, 3.5, 2)$ and $Y = (2, 2, 5, 2)$

Then the POLYLINE is $(4, X, Y)$

Triangle vertices at,

$$x_1 = 2, \quad y_1 = 2$$

$$x_2 = 5, \quad y_2 = 2$$

$$x_3 = 3.5, \quad y_3 = 5$$

coordinate system and transformed into a viewport defined rising Normalized device coordinates (NDCs) in which coordinates values are defined to lie within the range $0 < x < 1$ and $0 < y < 1$.

- ✓ Display characteristics for primitives, such as line-style and thickness, colour, text font and text angle, are defined by attributes, the values of which are set using the SET command.
- ✓ GKS also allows attributes to be bundled, that is grouped together and modified as a single entity.
- ✓ The other features of the GKS standard include the handling of user interaction and a wide range of levels of operation for input and output.
- ✓ The other graphics standard system which was widely used in Europe is Graphics Kernal System (GKS).
- ✓ Graphics Kernal System implementations have been made by many hardware manufacturers, for many languages.
- ✓ On the other side, GKS is not satisfactory for dynamic graphics, nor as a tool for programming large graphics applications, and hence a variety of alternative approaches have been developed.
- ✓ The alternative approach for GKS is Programmer's Hierarchial Interactive Graphics System (PHIGS).
- ✓ The PHIGS standard itself has evolved from GKS and CORE, and has the features that are derived from each of the earlier standards.

(Metafile - a piece of graphical information stored in a format that can be exchanged between different systems or software.)

OTHER STANDARDS FOR COMPUTER GRAPHICS

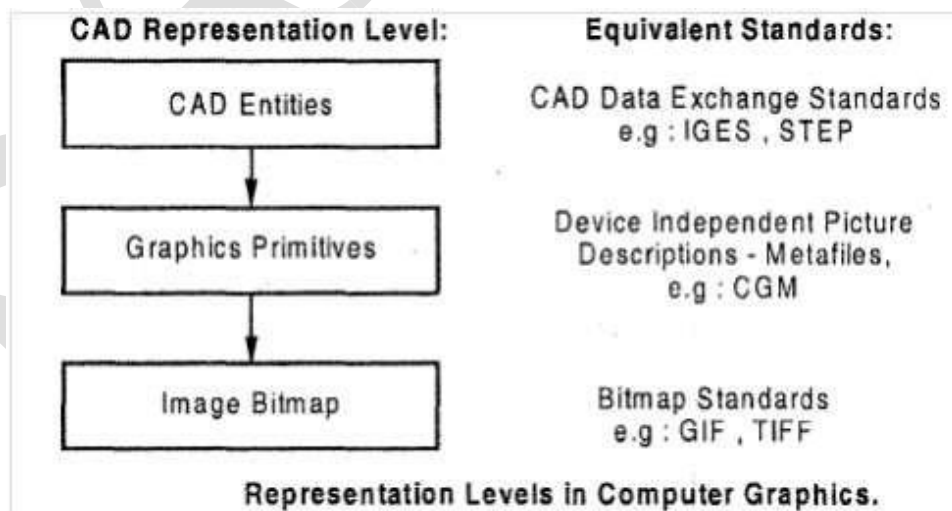
- ✓ To achieve the portability on a wide scale an industry graphics standard was required.
- ✓ So, the graphics standard planning committee of the Siggraph of the association for computing machinery (CACM) in 1977 introduced the CORE (core graphics system), and modified it in 1979.
- ✓ CORE provided a standardized set of commands to control the construction and display of graphics images and was independent of hardware or of language.

- ✓ So a program written using CORE could be run on any system that had implemented the standard.
- ✓ In earlier version, CORE provided for line drawing both two-dimensional and three-dimensional graphics, but the later version includes raster operation such as area fill.

Standards for Computer Graphics vs Images

- ✓ The purpose of GKS and other similar standards is to allow graphics to be drawn on a display device by an application program.
- ✓ For example, a CAD system might generate graphics primitives from the CAD model, and display them on the screen using GKS procedure calls.
- ✓ The CAD model itself comprises geometric, annotation and other entities stored in the CAD system data structure.
- ✓ This model is converted by the CAD system into a series of graphics primitives, and these are then displayed on the screen using the graphics procedures, typically by setting the values of the pixels in a rectangular raster array.
- ✓ The raster array is represented by a region of computer memory known as a bitmap.
- ✓ Images stored at the level of the graphics primitives are stored in graphics metafiles, which store graphical data containing device - independent description of pictures.
- ✓ Images stored at the level of bitmap are stored by making a structured, persistent representation of the bitmap.
- ✓ A graphics metafile is a formatted computer file containing graphics commands and data. It may be either a trail of the graphics commands used to generate a picture of the elements that make up the picture.
- ✓ A key feature of a metafile is that it is resolution independent, and can therefore be displayed at different sizes and scales and on devices of different resolution.

STANDARDS FOR EXCHANGING IMAGES



- ✓ There are set of important graphics standards concerning the storage and exchange of images produced using computer graphics.
- ✓ These standards may be divided into those concerned with images that are a collection of graphics

primitives.

- ✓ Those images are concerned and stored as bitmaps.
- ✓ The former includes the Computer Graphics Metafile (CGM), which establishes a format for device independent definition, capture, storage and transfer of vector graphics, images and the companion Computer Graphics Interface (CGI) provides a interface for the CGM primitives.
- ✓ Large number of bitmap storage formats including the Graphics Interchange Format (GIF), Tag image file format (TIFF or TIF), the windows Bitmap format (BMP) and many others.

Bitmaps

- ✓ At the lowest level in the computer graphics hierarchy is the pixel raster displayed on a graphics device.
- ✓ One bit per pixel allows only 'black-and-white' images.
- ✓ Colour bitmaps commonly assign 4, 8 or 24 bits/pixel to give 16 or 256 colours or 256 levels of each of the red, green and blue colour guns respectively.
- ✓ The simplest way of storing a bitmap is simply to write the numbers that the pixels represent to the file, together with a header giving information about the file.
- ✓ A large number of bitmap storage formats have been developed over the years for a variety of purposes.

OPEN GRAPHICS LIBRARY (OpenGL)

- ✓ Windows X offers a powerful, device independent means of generating graphical systems and user interfaces, but its device independence and distributed nature means that it can impose a high load on hardware.

(The X Window System (X11, or shortened to simply X, and sometimes informally X-Windows) is a windowing system for bitmap displays, common on UNIX-like computer operating systems.)

- ✓ For this reason, Silicon Graphics Inc. (SGI) developed the OpenGL application-programming interface (API) for the development of 2D and 3D graphics applications.
- ✓ Although it is possible for the API to be implemented entirely in software, it is designed to be implemented mostly or entirely in hardware.
- ✓ Open GL comprises a set of several hundred procedures and functions that allow a programmer to specify the objects and operations involved in the production of colour graphical images of three dimensional objects (SGI 1997). It is a low-level vendor-neutral software interface.
- ✓ It is often referred to as the assembler language of computer graphics.
- ✓ It provides enormous flexibility and functionality. It is used on a variety of platforms.
- ✓ OpenGL is a low-level graphics library specification. OpenGL makes available to the programmer a small set of geometric primitives - points, lines, polygons, images, and bitmaps.
- ✓ OpenGL provides a set of commands that allow the specification of geometric objects in two or three dimensions, using the provided primitives, together with commands that control how these objects are rendered into the frame buffer.
- ✓ It provides a means of drawing and rendering geometric objects like points, line segments, polygons for

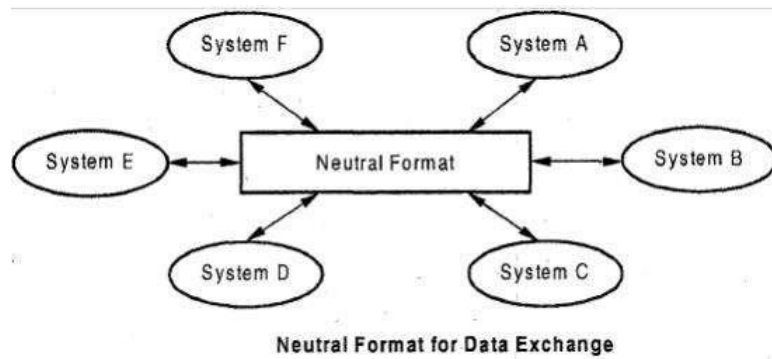
which specifying how they should be coloured, and how they should be mapped from the model space to the screen.

- ✓ Open GL does not require high performance display hardware to be present, but it does require a frame buffer - memory that stores the raster display bitmap.
- ✓ Open GL draws directly into the frame buffer but also allows the use of multiple buffers where, for example one buffer is displayed while second is being updated.
- ✓ Open GL integrates with X and may use the X server to execute commands, but if the program is running on the same machine as high-performance graphics hardware, the X server can be bypassed to obtain the highest performance.
- ✓ One of the restrictions of IrisGL (Earlier version of OpenGL) was that it only provided access to features supported by the underlying hardware. If the graphics hardware did not support a feature, then the application could not use it.
- ✓ OpenGL overcame this problem by providing support in software for features unsupported by hardware, allowing applications to use advanced graphics on relatively low-powered systems.
- ✓ History
 - In 1992, SGI led the creation of the OpenGL Architecture Review Board (OpenGL ARB), the group of companies that would maintain and expand the OpenGL specification in the future.
 - In 1994, SGI played with the idea of releasing something called "OpenGL++" which included elements such as a scene-graph API (presumably based on their Performer technology). The specification was circulated among a few interested parties – but never turned into a product.
 - Microsoft released Direct3D in 1995, which eventually became the main competitor of OpenGL.
 - On 1997, Microsoft and SGI initiated the Fahrenheit project, which was a joint effort with the goal of unifying the OpenGL and Direct3D interfaces (and adding a scene- graph API too).
 - In 1998, Hewlett-Packard joined the project. It initially showed some promise of bringing order to the world of interactive 3D computer graphics APIs, but on account of financial constraints at SGI, strategic reasons at Microsoft, and general lack of industry support, it was abandoned in 1999.
 - The latest version is OpenGL 4.5.

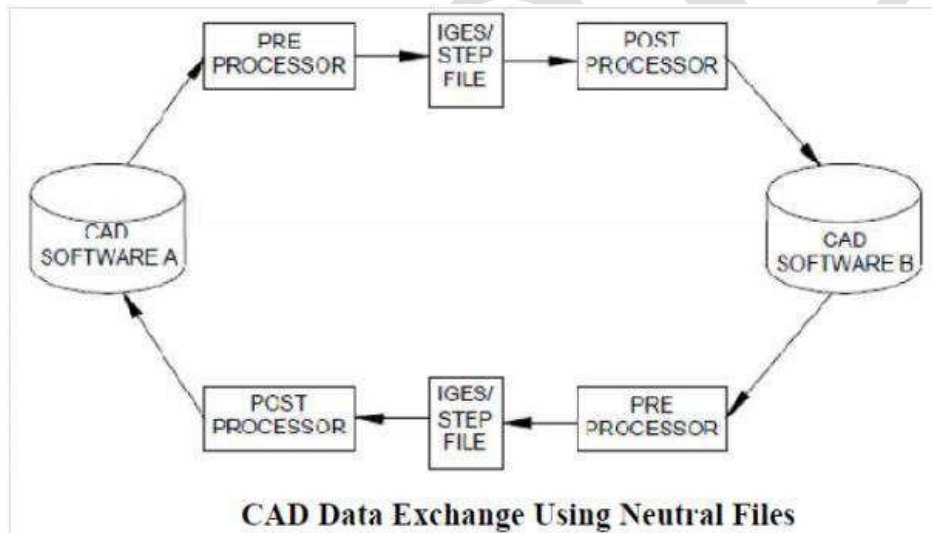
DATA EXCHANGE STANDARDS

- ✓ The increase in CAD applications in many parts of the engineering industry has been accomplished by growth in product variety and broadening of the range of companies involved in the design of a particular product.
- ✓ The easiest way for two companies to exchange data is to use the same CAD software, operating at the same revision level.

- ✓ The transfer of data between the systems has been made possible by the neutral format of data exchange.



- ✓ The neutral files will have standard formats and software packages can have pre-processors to convert drawing data to neutral file and postprocessors to convert neutral file data to drawingfile.
- ✓ Most commonly used types of neutralfiles:



- i. Drawing exchange files (DXF)
- ii. IGES files
- iii. STEP files

iv. CALS

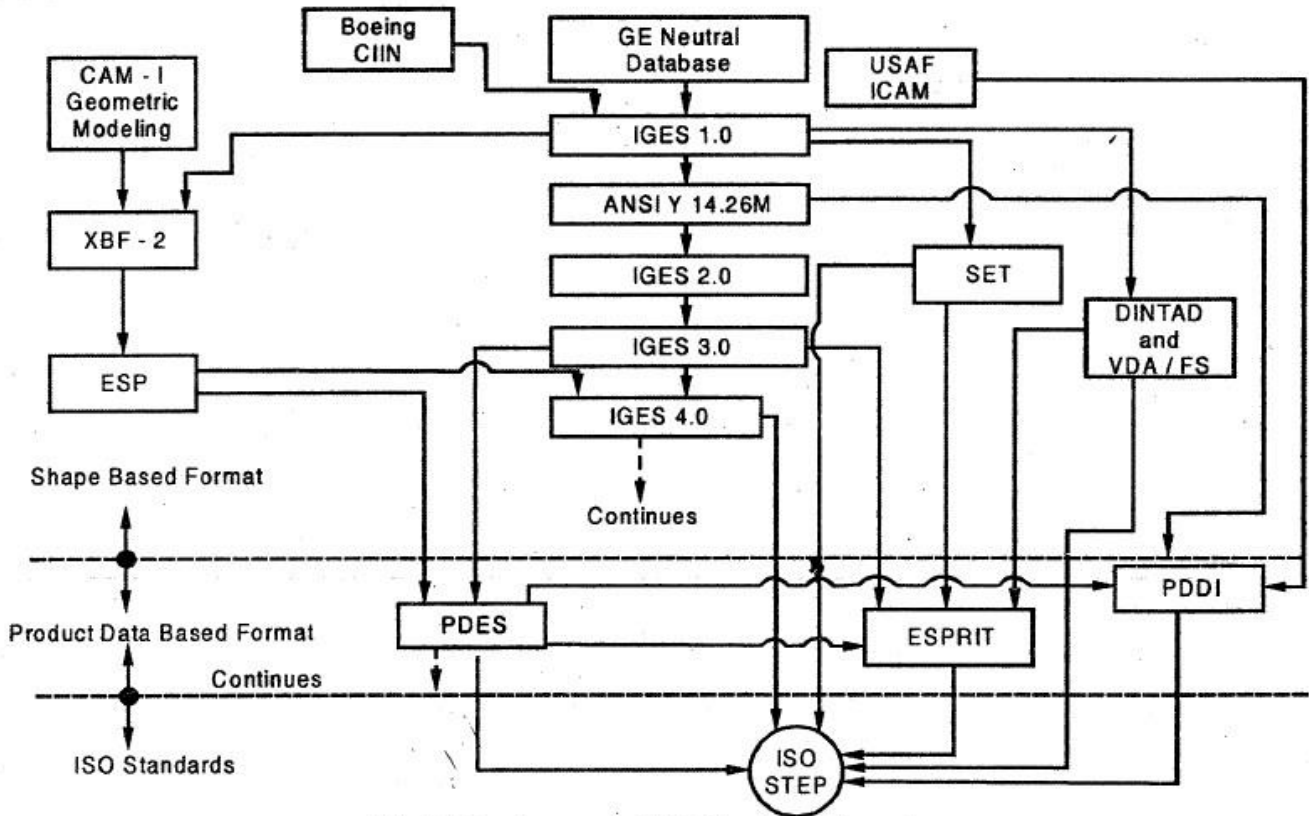


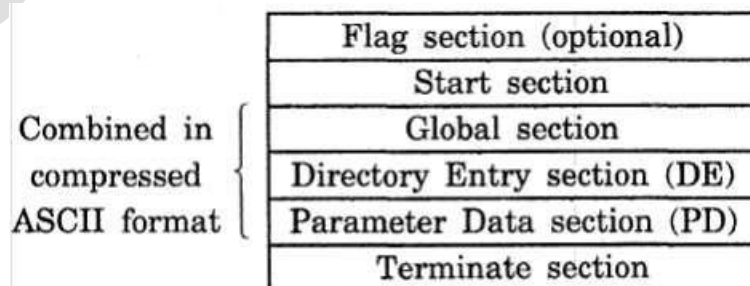
Fig:5.4 Development of Data Exchange Format

INITIAL GRAPHICS EXCHANGE SPECIFICATION (IGES)

- ✓ The IGES committee was established in the year 1979. The CAD/CAM Integrated Information Network (CIIN) of Boeing served as the preliminary basis of IGES. IGES version 1.0 was released in 1980.
- ✓ IGES continues to undergo revisions. IGES is a popular data exchange standard today. There are eight vertices (marked as PNT 0 - PNT 8), 12 edges and two circles that form the entities of the model.
- ✓ IGES (Initial Graphics Exchange Specification) is the first standard exchange format developed to address the concept of communicating product data among dissimilar CAD/CAM systems.
- ✓ IGES has gone through various revisions since its inception. Currently it supports solid modeling, including both B-rep and CSG schemes.
- ✓ IGES has three data types: geometric, annotation, and structure. The latter two are non geometric data types.
- ✓ Geometric entities define the product shape and include curves, surfaces, and solids.
- ✓ Non geometric entities provide views and drawings of the model to enrich its representation and include annotation and structure entities.
- ✓ Annotation entities include various types of dimensions (linear, angular, and ordinate), centerlines, notes, general labels, symbols, and cross-hatching.
- ✓ Structure entities include views, drawings, attributes (such as line and text fonts, colors, and layers), properties (e.g. mass properties), sub figures and external cross reference entities (for surfaces and assemblies), symbols (e.g.. mechanical and electrical symbols), and macros (to define parametric parts).
- ✓ IGES files can also be generated for:
 - i. Surfaces
 - ii. Datum curves and points IGES
- ✓ File structure
 - o An IGES file consists of six and the identification character, is also called the section code.

Flag Section

- ✓ The flag section is used only with compressed ASCII and binary format.



IGES file structure

Start section

- ✓ This section is setup manually by the person initiating the IGES file.
- ✓ This contains the information that may help the user at the destination, such as the features of the originating system.

Global section

- ✓ This section provides the 24 field parameters necessary to translate the file. It includes the delimiter characters (1 and 2), sender's identifier (3), file name (4), ID of the software which generate file (5), version of IGES processor (6), precision of integer (7 to 11), receiver's identifier (12), model space (13), units (14), name of the units (15), maximum number of line thickness (16, 17), time file generated (18), smallest distance (19), largest coordinate value (20), person and organization creating file (21 and 22), IGES version (23), drafting standards (24).

Directory section (DE)

- ✓ This section is generated by IGES pre-processor.
- ✓ This section also contains the entry for each entity in the file comprising a code representing the entity type and subtype.
- ✓ Two lines comprising 18 fields of eight characters are used for each entry.

Parameter data section (PD)

- ✓ This section contains the entity-specific data such as coordinate values, annotation text, number of spline data points and etc.
- ✓ The Parameter Data (PD) section contains the data that defines the entity. For example, if the entity is a circular arc, the center, start, and end points, unit normal, and etc... is required to define it. The Parameter Data section is identified by a "P" in column 73.
- ✓ The Directory section organizes and gives structure to the information in the Parameter Data section.
- ✓ There can be only one directory entry for each Parameter Data section entity.
- ✓ Directory section entries may reference other Directory section entries. This happens when we specify a transformation matrix and when we represent structures.
- ✓ The supported parameter data is the data being communicated.

Termination section

- ✓ This section marks the end of the data file.
- ✓ This section contains subtotals of records for data transmission check purposes.
- ✓ The Terminate section consists of one physical card image record.

Geometric Entities

- ✓ IGES uses two distinct but related Cartesian coordinate systems to represent geometric entity types. These are the MCS and WCS.
- ✓ IGES refers to the WCS as the definition space. The WCS plays a simplifying role in representing planar entities. In such a case, the XY plane of the WCS is taken as the entity plane, and therefore only x and y coordinates relative to the WCS are needed to represent the entity.
- ✓ To complete the representation, a transformation matrix is assigned (via a pointer) to the entity as one

of its parameters to map its description from WCS to MCS.

- ✓ This matrix itself is defined in IGES as entity type 124. Each geometric entity type in IGES has such a matrix.
- ✓ If an entity is directly described relative to the MCS, then no transformation is required.
- ✓ This is achieved in IGES by setting the value of the matrix pointer to zero to prevent unnecessary processing.
- ✓ As a general rule, all geometric entity types in IGES are defined in terms of a WCS and a transformation matrix.
- ✓ The case when MCS and WCS are identical is triggered by a zero value of the matrix pointer.
- ✓ IGES reserves entity numbers 100 to 199 inclusive for its geometric entities. Sample entity type numbers used by IGES are shown in below table. Specifications and descriptions of entities, including geometric entities, in IGES follow one pattern.
- ✓ Each entity has two main types of data: directory data and parameter data. The former is the entity type number, and the latter are the parameters required uniquely and completely define the entity. In addition, IGES specifies other parameters related to entity attributes and to IGES file structure.

Entity number	Entity description	Entity number	Entity description
100	Circular arc	132	Connect point
102	Composite curve	136	Finite element
104	Conic arc	138	Nodal display and rotation
106	Copious data	140	Offset surface
108	Plane	142	Curve on a parametric surface
110	Line	144	Trimmed parametric surface
112	Parametric spline curve	146	Nodal results
114	Parametric spline surface	148	Element results
116	Point	150	Block
118	Ruled surface	152	Right angular wedge
120	Surface of revolution	154	Right circular cylinder
122	Tabulated cylinder	156	Right circular cone
124	Transformation matrix	158	Sphere
125	Flask	160	Torus
126	Rational B-spline curve	162	Solid of revolution
128	Rational B-spline surface	164	Solid of linear extrusion
130	Offset curve	186	Ellipsoid

IGES geometric entities

Annotation Entities

- ✓ Drafting data are represented in IGES via its annotation entities.
- ✓ Many IGES annotation entities are constructed by using other basic entities that IGES defines, such as copious data (centerline, section, and witness line), leader (arrow), and a general note.
- ✓ An annotation entity may be defined in the modeling space (WCS) or in the drawing space (a given drawing).
- ✓ If a dimension is inserted by the user in model mode, then it requires a transformation matrix pointer when it is translated into IGES.

Entity number	Entity description	Entity number	Entity description
202	Angular dimension	216	Linear dimension
206	Diameter dimension	218	Ordinate dimension
208	Flag note	220	Point dimension
210	General label	222	Radius dimension
212	General note	228	General symbol
214	Leader (arrow)	230	Sectional area

IGES annotation entities

Structure Entities

- ✓ The previous two sections show how geometric and drafting data can be represented in IGES.
- ✓ Product definition includes much more information. IGES permits a valuable set of product data to be represented via its structure entities.
- ✓ These entities include associativity, drawing, view, external reference, property, subfigure, macro, and attribute entities. Attributes include line fonts, text fonts, and color definition.

Entity number	Entity description	Entity number	Entity description
302	Associativity definition	406	Property
304	Line font definition	408	Singular subfigure instance
306	Macro definition	410	View
308	Subfigure definition	412	Rect. array subfig. instance
310	Text font definition	414	Circular array subfig. instance
312	Text display template	416	External reference
314	Color definition	418	Nodal lead/constraint
320	Network subfigure definition	420	Network subfigure instance
402	Associativity instance	600 699	Macro instance (user defined)
404	Drawing	10000 99999	Macro instance (user defined)

IGES structure entities

Error handling

- ✓ While importing an IGES file, Error handling is very important.
- ✓ There are two major error sources when processing IGES file: Program errors in the processor and misinterpretation of the IGES standard itself.
- ✓ The way an IGES processor reports error is – the preprocessor should report the entity type, number of unprocessed entries, reasons for unprocessing and other relevant database information of these unprocessed entities.
- ✓ IGES should also report any invalid or missing data encountered in reading IGES files especially those that were edited.

STEP (Standard for the Exchange of Product Data)

- ✓ The Standard for the Exchange of Product Data (STEP) is the enabler for such seamless data exchange.
- ✓ It provides a worldwide standard for storing, sharing and exchanging product information among different CAD systems.
- ✓ Although STEP itself is the basis for Product Data Management System (PDM), it covers border functionalities.
- ✓ It includes methods of representing all critical product specifications such as shape information, materials, tolerances, finishes and product structure.
- ✓ Whereas the Initial Graphics Exchange Specification (IGES) standard has widespread use, it has its shortcomings. It does not convey the extensive product information needed in the design and manufacturing cycle.
- ✓ Often IGES translators are required to move design data from one CAD system to another.
- ✓ STEP is often viewed as a replacement for IGES, though IGES is still expected to be in active use for some more time in the future.
- ✓ Although the current focus of STEP is on mechanical parts, STEP is a data exchange standard that would apply to a wide range of product areas, including electronics, architectural, engineering and construction, apparel and shipbuilding.

STEP Architecture

STEP architecture has four main components: EXPRESS

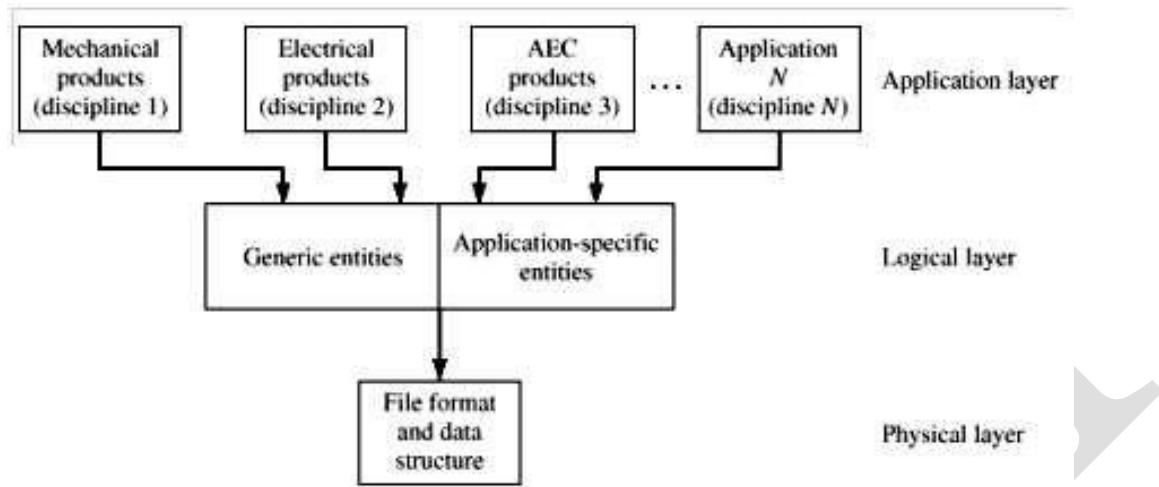
- ✓ modeling language
- ✓ Data schemes including attributes such as geometry, topology, features and tolerance.
- ✓ Application interface called Standard Data Access Interface (SDAI), which is a standard interface to enable applications to access and manipulate STEP data

STEP database, which has the following forms: ASCII format

- ✓ file for data exchange
- ✓ Working from file, usually in binary format, that can be shared by multiple systems

- ✓ Shared database, involving object oriented database management system or relational database system
- ✓ Knowledge base, with a database management system as a base coupled to an expert shell.

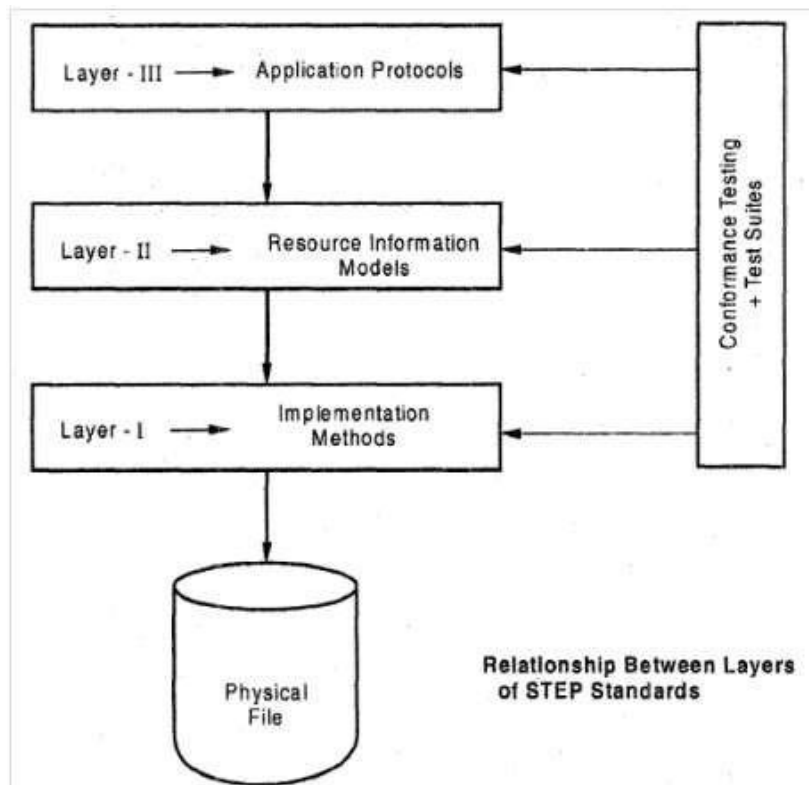
STEP has a three layer architecture as shown in that enables multiple application views and



STEP three-layer architecture.

implementations.

- ✓ The first layer comprises techniques for the implementation of STEP, in which the models are related to the EXPRESS language, and through this physical file.
- ✓ The second layer provides context – independent information such as the description of the geometry, topology or product structure. Resource models are so called because they provide Resource to the 3rd layer.
- ✓ The third layer contains information related to a particular application domain such as draughting or electrical product modeling.
- ✓ This layer describes constrained subsets of the STEP standard which should ensure that the implementations by different vendors are very much more compatible than IGES implementations.



✓ This standard itself is so large that it is being developed incrementally as a series of separate standards called parts.

✓ The parts, themselves are divided into seven classes, as follows:

1. Introductory (parts 1-9)

Consists of overview and general principles

2. Description methods (parts 11-19)

Consist of parts related to the express language

3. Implementation method (parts 21-29)

Describes how EXPRESS is mapped to physical file and other storage mechanisms.

4. Conformance Testing Methodology and Frame work (parts 31-9)

This provides the methods for testing implementations and test suites to be used during conformance testing.

5. Integrated Resource (parts 41-99, 101-99)

This part includes generic resources such as geometry and structure representation.

6. Application protocols (parts 201-99)

This part describes implementations of STEP specific to particular industrial applications.

7. Abstract test suites (parts 301-99)

This part provides test suites for each of the application protocols.

8. Application interpreted construct (parts 501)

This part describes various model entity constructs and specific modeling approaches.

STEP Enabler for Concurrent Engineering

STEP was released in early 1993 as a Draft International Standard (DIS).

The initial release of STEP has four basic parts. These include:

- EXPRESS modeling language
 - Two application protocols
 - Drafting and Configuration Control Design for three-dimensional product data.
 - Six application resources.
- ✓ Subsequent releases of STEP provided added functionality in terms of the kinds of product supported and the extent of the product life cycle.
 - ✓ While STEP is advancing towards maturity, it had been investigated for the feasibility of incorporation into framework system.
 - ✓ Both STEP and Concurrent Engineering share the common goal of influencing the product cycle from design, assembly, etc. to the disposal stages which have been realized in the CONSENS system under ESPRIT EP6896.
 - ✓ The object-oriented database for CONSENS has a schema with STEP definitions alongside company specific definitions.
 - ✓ A module called Product Information Archive (PIA) provides functionality for STEP data access via SDAI.
 - ✓ It is generic to be adopted for different domains.
 - ✓ For example it is used for product information by the Aircraft Company, Deutshces Aerospace and electronics manufacturing company, AEG.
 - ✓ STEP data export in a CAD modeling package has the following options:
 - (i) wire frame edges
 - (ii) Surfaces
 - (iii) Solids
 - (iv) Shells
 - (v) Datum curves and points

Difference between IGES and STEP

- ✓ IGES was not developed using formal specification language, it was not based on information modelling methodology.
- ✓ It does not clearly distinguish between the logical specification of the standard, the application requirements and physical specification for the storage data in exchange files.
- ✓ IGES does not provide sufficient conformance clauses or test sets for testing the implementations from

different CAD vendors.

- ✓ STEP improves on this in a number of respects.

CALS

- ✓ CALS (Continuous Acquisition and Life-cycle Support) is a United States Department of Defense (DoD) initiative for electronically capturing military documentation and linking related information.
- ✓ In the past, technical data such as engineering drawings, illustrations, and textual data for a weapon system was delivered to the Government in paper form.
- ✓ This made it necessary for DoD activities involved in managing the acquisition of a weapon system to orient their processes around handling paper-based documentation.
- ✓ These processes, however, were slow, error-prone and manpower intensive.
- ✓ In the mid 1980's the DoD sought to capitalize on advances in computer hardware and in the areas of computer-aided design, computer-aided engineering, and concurrent engineering.
- ✓ DoD structured a series of military specifications and standards that facilitated the handling of weapon system technical data in open, digital formats.
- ✓ This initiative grew into a joint DoD-industry Continuous Acquisition and Life-cycle Support (CALS) initiative and led to acquisition processes between defense contractors and DoD acquisition managers being conducted with technical data in digital formats.
- ✓ With this change, there came a need for data management systems that could receive, store, and manipulate technical data in its various formats.
- ✓ Additionally, many of the acquisition management processes required "reengineering" using concepts such as Business Process Reengineering in order to truly reap the benefits of receiving, handling, and managing technical data in digital formats.
- ✓ This CALS initiative has developed a number of standard specifications (protocols) for the exchange of electronic data with commercial suppliers.
- ✓ It was thought that the CALS initiative could help the DoD reduce its costs for acquiring technical documentation while making it more accurate, current and timely.
- ✓ In its beginnings, CALS primarily dealt with the logistics of support documentation.
- ✓ It was originally called Computer-Aided Logistics Support.
- ✓ As the benefits of the CALS initiative became better known, DoD acquisition managers sought to incorporate CALS concepts into weapon systems procurements.
- ✓ In 1988, CALS was renamed the Computer-aided Acquisition and Logistics Support initiative.
- ✓ This better reflected its use in managing the technical information associated with weapon system acquisition.
- ✓ Close integration among buyers and vendors or the different units of an enterprise, created and sustained through application of standard technologies (such as electronic data interchange or EDI), streamlining of business processes (business process engineering), and effective use of business and technical information.
- ✓ Developments in the field of Concurrent Engineering (CE) eventually led the CALS initiative to

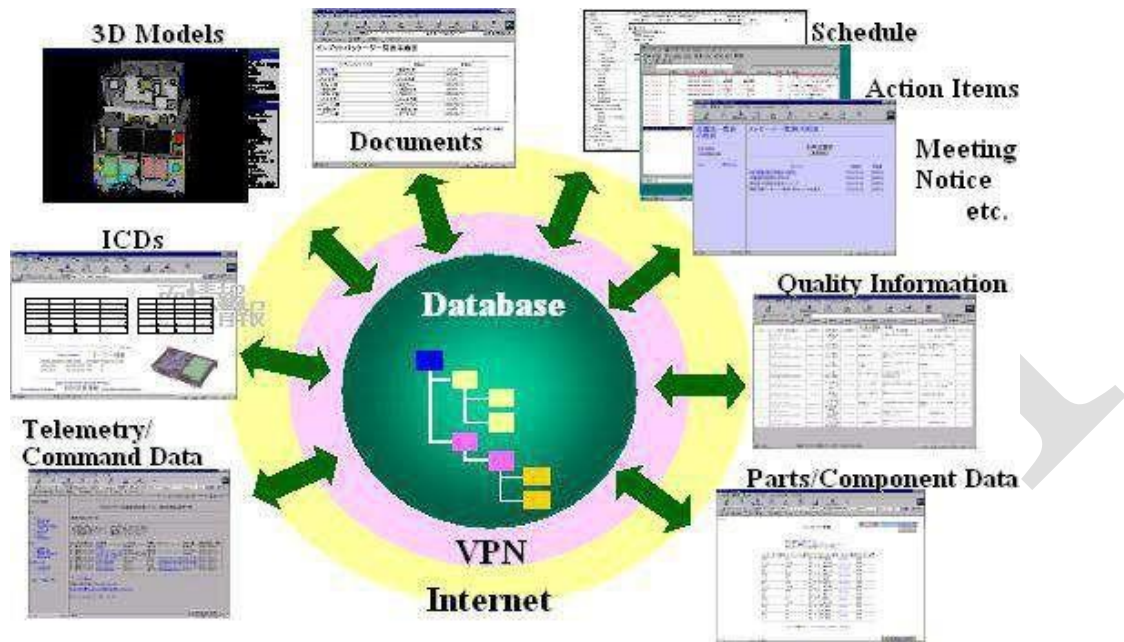
encompass all aspects of weapon system acquisition: design, production and logistics support processes.

- ✓ Similarly, advances in telecommunications such as enterprise networking and digital information exchange protocols led to more technical documentation to be exchanged between businesses.
- ✓ Terms such as electronic commerce and electronic data interchange soon became associated with CALS.
- ✓ Now renamed Continuous Acquisition and Life-cycle Support, the CALS initiative has been expanded from its roots in technical documentation and logistics support to CE and integrated business processes.
- ✓ These standards are often referred to as simply "CALS". CALS standards have been adopted by several other allied nations.
- ✓ It has gained acceptance outside the DoD and defense industries to become a joint DoD-industry managed initiative.
- ✓ The CALS initiative has also been accepted and implemented within international defense departments in Canada, Europe, Asia, and Australia.
- ✓ This data exchange standard was produced with an aim of applying computer technology to the process of specifying, ordering, operating, supporting and maintaining the weapons systems used by the US armed forces, although it can be adopted by any industry, not just defence industry.
- ✓ The CALS initiative has endorsed IGES and STEP as formats for digital data.
- ✓ CALS includes standards for electronic data interchange, electronic technical documentation, and guidelines for process improvement.
- ✓ The Vision is for all or part of a single enterprise (e.g., an original equipment manufacturer and its suppliers, or a consortium of public and private groups and academia), to be able to work from a common digital data base, in real time, on the design, development, manufacturing, distribution and servicing of products.
- ✓ The direct benefits would come through substantial reductions in product-to- market time and costs, along with significant enhancements in quality and performance.
- ✓ CALS has developed a further standard which defines subsets of IGES to be used for specific applications including technical illustrations, engineering drawings, electronic engineering data, and geometry for manufacture by numerical control machines.
- ✓ It is expected that CALS will use the STEP standards for product data, and will also extend into such areas as electronic hardware description and office document exchange.

✓

- ✓ The Spacecraft Industries CALS is to introduce a exchanging of interface data, which are applied in the collaborative manufacturing of spacecraft by the space industry.

The data is exchanged by way of CALS standard such as Word, Excel, PDF, TIFF and 3D-CAD etc.



through the network.

- ✓ Generally, several companies conduct manufacturing of spacecraft simultaneously, for which control documents named ICD (Interface Control Documentation and Drawing) with 3D-CAD are utilized to provide information on interface work.

Two "flagship" CALS systems:

- o Joint Computer-aided Acquisition and Logistics Support (JCALS) and the
- o Joint Engineering Data Management Information and Control System (JEDMICS).
- ✓ Joint Computer-aided Acquisition and Logistics Support
 - o JCALS concept originated from the US Army's Technical Information Management System (TIMS)
- ✓ JCALS is an information management system that will support acquisition, logistics support, engineering, manufacturing, configuration control and materiel management processes throughout the life-cycle of a weapon system.
- ✓ It uses multi weapon system IWSDBs and Global Data Dictionary and Directory (GDD/D) Services that are connected by a wide area computer network.
- ✓ The interface for users is the JCALS Workbench that provides an environment to access all of JCAL's functionality transparently to the user.

DXF

- ✓ DXF (Data exchange Format) is a de facto standard due to its popularity.
- ✓ DXF is an AutoCAD format. Auto Desk Inc., the maker of AutoCAD, publishes, supports, and maintains it.
- ✓ DXF 3D is a format that translates CAD models (part files), while DXF/DWG is a format that translates drawing files.
- ✓ DXF/DWG does not and cannot translate part files. DXF files come in two
- ✓

formats: ASCII and binary.

- ✓ The ASCII version is the most widely used in industry.

The DXF format is a tagged data representation of all the information contained in an AutoCAD drawing file. Tagged data means that each data element in the file is preceded by an integer number that is called a group code. A group code's value indicates what type of data element follows. This value also indicates the meaning of a data element for a given object (or record) type.

- ✓ Important sections of DXF file: Header, Tables, Blocks, and Entities.
- ✓ AutoCAD DXF (Drawing Interchange Format or Drawing Exchange Format) is a CAD data file format developed by Autodesk for enabling data interoperability between AutoCAD and other programs.
- ✓ DXF was originally introduced in December 1982 as part of AutoCAD 1.0, and was intended to provide an exact representation of the data in the AutoCAD native file format, DWG (Drawing), for which Autodesk for many years did not publish specifications.
- ✓ Because of this, correct imports of DXF files have been difficult. Autodesk now publishes the DXF specifications as a PDF on its website.
- ✓ Versions of AutoCAD from Release 10 (October 1988) and up support both ASCII and binary forms of DXF. Earlier versions support only ASCII.
- ✓ As AutoCAD has become more powerful, supporting more complex object types, DXF has become less useful. Certain object types, including ACIS solids and regions, are not documented.
- ✓ Other object types, including AutoCAD 2006's dynamic blocks, and all of the objects specific to the vertical market versions of AutoCAD, are partially documented, but not well enough to allow other developers to support them. For these reasons many CAD applications use the DWG format which can be licensed from Autodesk or non-natively from the Open Design Alliance.
- ✓ DXF coordinates are always without dimensions so that the reader or user needs to know the drawing unit or has to extract it from the textual comments in the sheets.

HEADER section: Contains General information about the drawing.

- ✓ The HEADER section of a DXF file contains the settings of variables associated with the drawing.
- ✓ System settings such as dimension style and layers.
- ✓ Each variable is specified by a 9 group code giving the variable's name, followed by groups that supply the variable's value.

CLASSES section

- ✓ Holds the information for application-defined classes whose instances appear in the BLOCKS, ENTITIES, and OBJECTS sections of the database.
- ✓ Generally does not provide sufficient information to allow interoperability with other programs.

✓ It is assumed that a class definition is permanently fixed in the class hierarchy.

TABLES section - contains definitions of named items.

✓ The TABLES section contains several tables, each of which can contain a variable number of entries.

✓ These codes are also used by AutoLISP and ObjectARX applications in entity definition lists.

✓ It includes line styles and user-coordinate systems. More tables:

- ✓ ○ Dimension Style (DIMSTYLE) table
- Layer (LAYER) table
- Linetype (LTYPE) table
- Text style (STYLE) table
- User Coordinate System (UCS) table
- View (VIEW) table

BLOCKS section

✓ This section contains an entry for each block reference in the drawing.

✓ The BLOCKS section of the DXF file contains all the block definitions, including anonymous blocks generated by the HATCH command and by associative dimensioning.

✓ Each block definition contains the entities that make up that block as it is used in the drawing.

✓ The format of the entities in this section is identical to those in the ENTITIES section.

ENTITIES section

✓ This section presents the group codes that apply to graphical objects. It includes entity definition and data.

✓ Example: Dimension entity definitions consist of group codes that are common to all dimension types, followed by codes specific to the type.

✓ More Ex.: Circle, Ellipse, Leader, Light

OBJECTS section

✓ This section presents the group codes that apply to non graphical objects. Also used by Auto LISP and Object ARX applications. ***THUMBNAILIMAGE section***

✓ Contains the preview image for the DXF file.

✓ This section exists only if a preview image has been saved with the DXF file.

Virtually all user-specified information in a drawing file can be represented in DXF format.

Advantages

✓ The DXF file format is the most compatible vector file type

✓ DXF files are used to exchange data between different CAD programs The DXF file

✓ format is easy to parse

✓ The DXF file specification is publicly available

Limitations

✓ DXF does not support application specific CAD elements Complex DXF

✓

files can become large in size

- ✓ Some applications can not deal with line widths in DXF Files

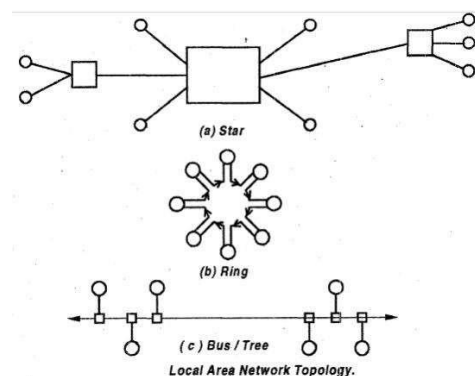
COMMUNICATION STANDARDS

- ✓ Data exchange depends not only on the compatibility of the applications data formats between the communicating systems, but also on compatibility of the physical means of communication. Computers are arranged to communicate with each other.
- ✓ Local connections are known as local area networks (LANs) and involves the connection of digital devices over distance from a few meters up to a few kilometers.
- ✓ Wide Area Networks (WANs) are used to connect the computers or machines of a number of university campuses, even if these sites are in different countries.

Local Area Networks

- ✓ A Local Area Network (LAN) is a network that is restricted to smaller physical areas e.g. a local office, school, or house. Approximately all current LANs whether wired or wireless are based on Ethernet. On a 'Local Area Network' data transfer speeds are higher than WAN and MAN that can extend to a 10.0 Mbps (Ethernet network) and 1.0 Gbps (Gigabit Ethernet).
- ✓ Computers and servers (provides services to other computers like printing, file storage and sharing) can connect to each other via cables or wirelessly in a same LAN.
- ✓ Wireless access in conjunction with wired network is made possible by Wireless Access Point (WAP).
- ✓ A WAP is able to connect hundreds or even more of wireless users to a network. Servers in a LAN are mostly connected by a wire since it is still the fastest medium for network communication.
- ✓ But for workstations (Desktop, laptops, etc.) wireless medium is a more suitable choice, since at some point it is difficult and expensive to add new workstations into an existing system already having complex network wiring.
- ✓ The most obvious area of difference between a WAN and a LAN is in the topology of network itself.
- ✓ LAN topology is generally rather simpler than the mesh arrangement.
- ✓ The method of controlling access to the network is also achieved in a number of different ways, of which perhaps the most important is the use of a control token, and carrier sense multiple access with collision detection (CSMA/CD).
- ✓ A wireless LAN or WLAN is a wireless Local Area Network, which is the linking of two or more computers without using wires.
- ✓ It uses radio communication to accomplish the same functionality that a wired LAN has. WLAN utilizes spread-spectrum technology based on radio waves.

To enable communication between devices in a

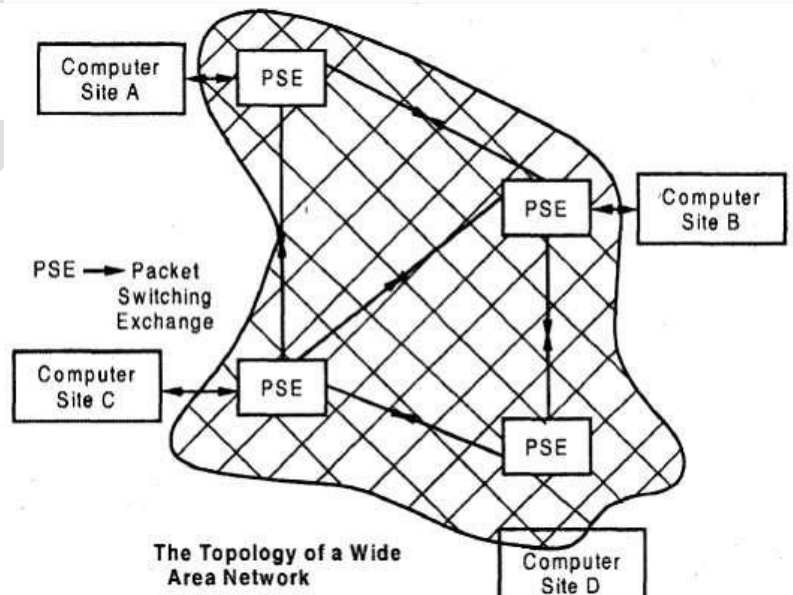


limited area, also known as the basic service set.

- ✓ This gives users the mobility to move around within a broad coverage area and still be connected to the network.

Wide Area Networks

- ✓ Wide Area Network is a computer network that covers relatively larger geographical area such as a state, province or country.
- ✓ Contrast with personal area networks (PAN's), local area networks (LAN's) or metropolitan area networks (MAN's) that are usually limited to a room, building or campus.
- ✓ WAN's are used to connect local area networks (LAN's) together, so that users and computers in one location can communicate with users and computers in other locations.
- ✓ It provides a solution to companies or organizations operating from distant geographical locations who want to communicate with each other for sharing and managing central data or for general communication.
- ✓ WAN is made up of two or more Local Area Networks (LANs) or Metropolitan Area Networks (MANs) that are interconnected with each other, thus users and computers in one location can communicate with users and computers in other locations.
- ✓ In 'Wide Area Network', Computers are connected through public networks, such as the telephone systems, fiber-optic cables, and satellite links or leased lines.
- ✓ The largest and most well-known example of a WAN is the Internet.
- ✓ WANs are mostly private and are building for a particular organization by 'Internet Service Providers (ISPs)' which connects the LAN of the organization to the internet.
- ✓ WANs are frequently built using expensive leased lines where with each end of the leased line a router is connected to extend the network capability across sites.
- ✓ For low cost solutions, WAP is also built using a 'circuit switching' or 'packet switching' methods.



- ✓ The data carrying networks operated by the public utilities are known as public switched data networks.
- ✓ When digital transmission is extended to subscriber equipment then at high-bit- rate switched transmission path will be available at each subscriber outlet, and network will be capable of supporting both digitized voice and data transmission without modems, as an integrated services digital network (ISDN).
- ✓ In addition, WAN's also refer to Mobile Data Communications, such as GSM, GPRS and 3G.

Fiber optic links

- ✓ It is an alternative to copper connecting wires in LANs.
- ✓ Optical communication offers the advantage of large transmission bandwidth and relatively immunity from electromagnetic interference or corrosion problems which make them ideal for industrial environments.
- ✓ Network based specifically on fiber optics new standard is fibre distributed data interference.
- ✓ As the bandwidth capabilities of fibre optics are further developed significantly higher rates are possible.

