ST. ANNE'S COLLEGE OF ENGINEERING AND TECHNOLOGY

(Accredited by NAAC, Approved by AICTE, New Delhi. Affiliated to Anna University, Chennai)

ANGUCHETTYPALAYAM, PANRUTI – 607 106.

DEPARTMENT OF MECHANICAL ENGINEERING

ME8691 COMPUTER AIDED DESIGN AND MANUFACTURING

OBJECTIVES:

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

To understand the application of computers in various aspects of Manufacturing viz., Design, Proper planning, Manufacturing cost, Layout & Material Handling system.

UNIT INTRODUCTION

Product cycle- Design process- sequential and concurrent engineering- Computer aided design – CAD system architecture- Computer graphics – co-ordinate systems- 2D and 3D transformation shomogeneous coordinates – Line drawing -Clipping- viewing transformation-Brief introduction to CAD and CAM – Manufacturing Planning, Manufacturing control- Introduction to CAD/CAM – CAD/CAM concepts ––Types of production – Manufacturing models and Metrics – Mathematical models of Production Performance

UNIT II GEOMETRIC MODELING

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 CAD STANDARDS

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

UNIT IV FUNDAMENTAL OF CNC AND PART PROGRAMING

Introduction to NC systems and CNC – Machine axis and Co-ordinate system- CNC machine tools- Principle of operation CNC- Construction features including structure- Drives and CNC controllers- 2D and 3D machining on CNC- Introduction of Part Programming, types – Detailed Manual part programming on Lathe & Milling machines using G codes and M codes- Cutting Cycles, Loops, Sub program and Macros- Introduction of CAM package.

UNIT V CELLULAR MANUFACTURING AND FLEXIBLE MANUFACTURING SYSTEM (FMS)

Group Technology(GT),Part Families–Parts Classification and coding–Simple Problems in Opitz Part Coding system–Production flow Analysis–Cellular Manufacturing–Composite part concept–Types of Flexibility – FMS – FMS Components – FMS Application & Benefits – FMS Planning and Control– Quantitative analysis in FMS

UNIT INTRODUCTION

Product cycle- Design process- sequential and concurrent engineering- Computer aided design – CAD system architecture- Computer graphics – co-ordinate systems- 2D and 3D transformation homogeneous coordinates – Line drawing -Clipping- viewing transformation-Brief introduction to CAD and CAM – Manufacturing Planning, Manufacturing control- Introduction to CAD/CAM – CAD/CAM concepts —Types of production – Manufacturing models and Metrics – Mathematical models of Production Performance

TWO MARK QUESTION AND ANSWERS

1. **Define computer aided design.**

Computer-aided design (CAD) is the use of computer systems to assist in the creation, modification, analysis, or optimization of a design. CAD software 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

2. **Define product design.**

Product design is the process of creating a new product to be sold by a business to its customers. It is essentially the efficient and effective generation and development of ideas through a process that leads to new products.

3. **What are the steps involved in design process? (MAY 2016)**

The steps/phases involved in design process are

1. Research
2. Feasibility assessment
3. Conceptualization
4. Establishing design
5. Preliminary design
6. Detailed design
7. Production planning and tool design
8. Production

4. **ist the functional areas of CAD design process.**

The four stages or functional areas of a CAD design process are:

1. Geometric modeling
2. Design analysis and optimization
3. Design review and evaluation, and
4. Documentation and drafting

5. **What is sequential Engineering?**

Sequential engineering is the term used to describe the method of production in a linear format. The different steps are done one after another, with all attention and resources focused on that one task. After it is completed it is left alone and everything is concentrated on the next task.

6. **What is concurrent engineering?**

In concurrent engineering, different tasks are tackled at the same time, and not necessarily in the usual order. This means that information found out later in the process can be added to earlier parts, improving them, and also saving a lot of time.

7. **What are the Advantages and Disadvantages of Sequential Engineering?**

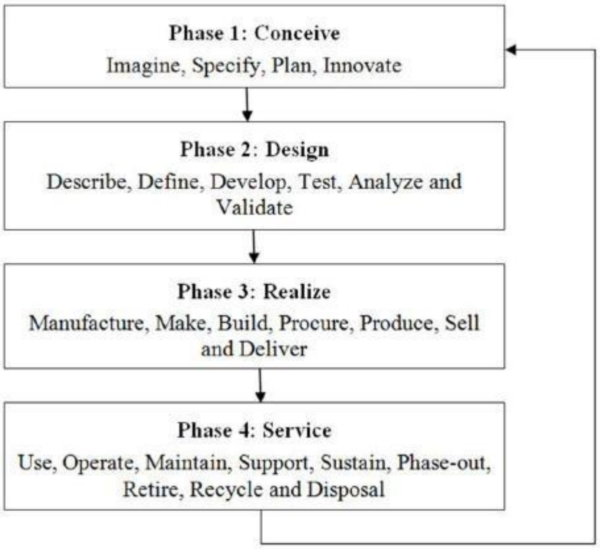
Advantages:

1. It is a simple, well-defined method and allows everyone to remain on the same page.
2. It is an enforced-discipline approach.

Disadvantages:

1. There is always some uncertainty at the beginning of a project. Changes that occur later in the projects are difficult and often times expensive.
2. Customer does not see a working product until the end, and if any changes need to be made at that point it can be costly and time-consuming.

8. Sketch product cycle model neatly.


Phase 1: Conceive
Imagine, Specify, Plan, Innovate
Phase 2: Design
Describe, Define, Develop, Test, Analyze and Validate
Phase 3: Realize
Manufacture, Make, Build, Procure, Produce, Sell and Deliver
Phase 4: Service
Use, Operate, Maintain, Support, Sustain, Phase-out, Retire, Recycle and Disposal

9. What are the advantages and limitations of CAD/CAM technology?

Advantages: 1. speed, particularly in terms of experimentation and communication 2. Accuracy (i.e. fewer mistakes), particularly when things are copied 3.Sharing of data 4. Increased productivity

Disadvantages: 1.Technology is expensive to buy and maintain. 2. Using computer technology requires a lot of training and experience.3. There is a lack of opportunity to experiment with real materials and 3D forms. 4. It is easy to lose data if files are not regularly ‘backed up’.

10. Define Computer graphics.

Computer graphics are pictures and movies created using computers - usually referring to image data created by a computer specifically with help from specialized graphic hardware and software.

11. Name the list of co-ordinate systems used in computer graphics systems.

1.Working Coordinate System (WCS) 2. Model (Object) Coordinate System (MCS) 3.Screen coordinate system (SCS)

12. Define working coordinate system.

It is a convenient user-defined system that facilitates geometric construction. It can be established at any position and orientation in space that the user desires.

A WCS can be thought of as a portable coordinate system. It is often used when the desired plane of sketching is not easily defined as one of the MCS planes.

13. Define model coordinate system.

MCS is the reference space of the model with respect to which all the model's geometrical data is stored. In CAD system MCS is generally shown by displaying the X, Y, and Z axes. The orthogonal planes formed by the axes of MCS are datum planes, sketch planes, or construction planes.

14. Define screen coordinate system.

An SCS is a 2D device-dependent coordinate system whose origin is located at the lower left corner of the screen. It is mostly used in view-related clicks such as definitions of view origin and select views for graphics operations.

15. What is 2D and 3D transformations.(NOV 2015)

Transformations are used to position objects, to shape objects, to change viewing positions, and even to change how something is viewed. 2D transformations, which uses 3x3 matrices, 3D transformations use 4X4 matrices (X, Y, Z, W)

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

For 2D

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For 3D

16. Distinguish between translation and scaling.

When every entity of a geometric model remains parallel to its initial position, the rigid body transformation of the model is defined as translation.

Scaling is used to change, increase or decrease the size of an entity or a model.

17. What is mirror (reflection) transformation?

Reflection (or mirror) transformation is useful in constructing symmetrical models. If, for example, a model is symmetric w.r.to a plane, and then only half of its geometry is created which can be copied by reflection to generate the full model.

18. Whatdoes Mapping mean?

Mapping means changing the description of a point (or set of point) belonging to an object from one coordinate system to the second one without altering the position and orientation in space w.r.to origins.

19. What is translational and rotational mapping?

When the axes of the two coordinates systems are parallel, the mapping is defined to be translational.

Rotational mapping between two coordinates system share the same origin and their orientations are different by the angle θ.

20. Whatis meant by concatenation?

In practice a series of transformations may be applied to a geometric model. Thus combining or concatenating transformations are quite useful. They are simply obtained by multiplying the [T] matrices of the corresponding individual transformations.

P^{*}= [Tn] [Tn-1]… [T2][T1]P

21. Why are homogenous coordinates often used?

Representing points by their homogenous coordinates provides an effective way to unify the description of geometric transformations as matrix multiplications.

It remove many anomalous situations encountered in Cartesian geometry such as representing points at infinity and no intersection of parallel lines. Also they greatly simplify the expressions defining rational parametric curves and surfaces.

22. Explainshort notes on viewing.

In the World Coordinate system, an area is selected which to be displayed is called Window. Further World coordinates are converted to viewing coordinates as per the viewing plane and the viewing direction of the user. Further it is transformed into normalized coordinate system which is a standard coordinate system having extent between 0 and 1. Y-extent is also between 0 and 1.Further the normalized coordinate system is converted to physical display device coordinate system on which an area is chosen where the image is displayed called viewport.

23. Define Product cycle.

Product cycle is the process of managing the entire lifecycle of a product from starting, through design and manufacture, to repair and removal of manufactured products.

24. List out fundamentals of product life cycle management. (MAY 2016)

1. Customer Relationship Management (CRM)
2. Supply Chain Management (SCM)
3. Enterprise resource planning (ERP)
4. Product Planning and Development (PPD).

25. What is conceptualization in design process?

A Concept Study is the stage of project planning that includes developing ideas and taking into account the all features of executing those ideas. This stage of a project is done to reduce the likelihood of assess risks, error and evaluate the potential success of the planned project.

26. Differentiate preliminary design and detailed design.

Preliminary design and Detailed design

The **preliminary design** fills the gap between the design concept and the detailed design phase. The system configuration is defined, and schematics, diagrams, and layouts of the project will offer early project configuration.

In **detailed design** and optimization, the parameters of the part being produced will change, but the preliminary design focuses on creating the common framework to construct the project.

The next phase of preliminary design is the Detailed Design.

27. What are the advantages of concurrent engineering?

1. Both product and process design run in parallel and take place in the same time.
2. Process and Product are coordinated to attain optimal matching of requirements for effective quality and delivery.
3. Decision making involves full team involvement.

28. What is ‘Rendering’?

Rendering is the making of a two dimensional image from a three dimensional model by means of computer programs. A picture file has objects in a strictly defined data structure; it would have information of geometry, lighting, viewpoint, texture, and shading as a description of the scene.

29. Name some applications of computer graphics. (JAN 2017)

Computer graphics provides high resolution and high quality images. These applications includes medical imagining systems, analyzing high–terrain topology, recognizing and interpreting satellite images, and performing scientific visualization.

30. Name some scope of computer applications in design concept. (JAN 2016)

The sketches and drawings can be created using a CAD/CAM system. Computer provides users with functions to perform geometric modeling and construction, editing and manipulation of existing models. The typical modeling operations that users can engage in are model creation, cleanup, documentation, and printing/plotting.

31. List some activities of the CAD process.

Activities of the CAD process include mass properties, finite element analysis, dimensioning tolerance, assembly modeling, generating shaded images, and documentation and drafting

32. List some Mechanical Engineering Applications of CAD.

- ➢ Two Dimensional Drafting: This is the most common use of a CAD package. 2-D drawings are used for manufacturing a product.
- ➢ Report Generating: To generate reports and bill of materials. Spreadsheets and word-processors can be linked to provide a report writing facility.
- ➢ 3-D Modeling: To create the wireframe, surface and solid models.The 3-D models are for concept verification, manufacturing, FEA, etc.
- ➢ Finite Element Analysis: FEA package is used for pre-processing, analysis, and post-analysis of structures. For this application, a CAD package contains both the modeling and analysis modules.
- ➢ Manufacturing: manufacturing software is usually called CAM, and contains CAD software as one of the components. CAM software provides capabilities of carrying out 2 and 3-axes machining.

33. Discuss utilization of CAD/CAM.

CAD/CAM has been utilized in engineering practice in many ways including drafting, design, Simulation, Analysis and manufacturing.

34. List out the four major areas of CAD/CAM.

CAD/CAM spans four major areas: geometric modeling, computer graphics, design applications, and manufacturing applications.

35. What is a Transformation?

A transformation is the process of mapping points to other locations. Common transformations are Translation, Scaling and Rotation.

36. What is homogenous transformation? (JAN 2014& NOV 2016)

Homogeneous coordinates are used in computer graphics and geometry modeling to speed the execution of geometric transformations.

They are often used in projective geometry, mechanism analysis and design, and robotics for development and formulation.

37. What is clipping?

Clipping is the process of determining the portions of a primitive lying within a region called the 'clipping region'. Types of clipping are Point clipping, Line clipping, Polygon clipping.

38. What is the purpose of Clipping?

It is for preventing: Activity in one window from affecting pixels in other windows Mathematical overflow and underflow from primitives passing behind the eye point or at great distances (in 3D).

39. Define clipping transformations. (JAN 2014)

Clipping transformation is transformation of viewing window when clipping the segments. 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 wraparound.

40. What is Area Clipping?

Clipping a line segment yields at most one line segment, Clipping a polygon can yield multiple polygons (However, clipping a convex polygon can yield at most one other polygon), these types clipping with polygon are known as area clipping.

41. What is Rasterization?

Rasterization (scan conversion) is to determine which pixels that are inside a primitive specified by a set of vertices.

- To convert continuous geometry, inside viewing region, into discrete pixels.
- Fragments have a location (pixel location) and other attributes such color and texture coordinates that are determined by interpolating values at vertices.
- Pixel colors determined later using color, texture, and other vertex properties
-

42. Define co-coordinating system & types. (JAN 2014)

Coordinate system is used to input, store, and display model geometry and graphics. The three Coordinate systems are Model (Master) Coordinate system (MCS), Working Coordinate system (WCS), and Screen Coordinate system (SCS).

43. Classify the viewing operations on a CAD system.

Most of the viewing operations on a CAD system can be classified into three groups: view orientation, view modes, and view manipulations.

44. What is View orientation?

View orientation includes standard views such as front, top, right, and isometric. They basically align the model on-screen in a way that displays the selected view.
View modes allow us to change the display of the model to different types such as wireframe, hidden, or shaded

45. Discuss about View manipulations.

View manipulations allow as dynamically rotating, panning and zooming the model to gain better control over its viewing.

46. Classify the modeling approaches.

The three modeling approaches are primitives, features, and sketching.

47. Discuss on primitives approach and list some primitives.

The primitives approach views a solid model as a combination of simple, generic, and standard shapes that can be combined. Primitives include a block (box), cylinder, sphere, cone, wedge, and torus. These primitives are combined via the boolean operations union, subtractions, and intersection.

48. What is meant by feature approach?

The feature approach is similar to the primitives approach; it replaces primitives with features and embeds Boolean operation in features.

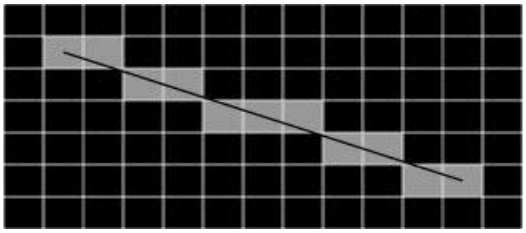
49. What is meant by sketching approach?

The sketching approach is similar to the features approaches, with one change. Instead of using predefined shapes only, such as holes and ribs, it allows CAD designers to create much more elaborate and more general features starting from a sketch.

50. Discuss on Line-Drawing Algorithms.

Bresenham’s line algorithm properties: -Only uses incremental integer calculations,

- Can be adapted to display circles and other curves
- Basic idea find next pixel from current one



Bresenham’s line algorithm

51. What are the sequences involved in 3D Rotations?

3D rotation sequence:

i. Translate the object so that the rotation axis passes through the coordinate origin.
ii. Rotate the object so that the axis of rotation coincides with one of the coordinate axes.
iii. Perform the specified rotation about 3. the selected axis.
iv. Apply inverse rotations to bring the rotation axis back to its original orientation.
v. Apply the inverse translation to bring the rotation axis back to its original spatial position.

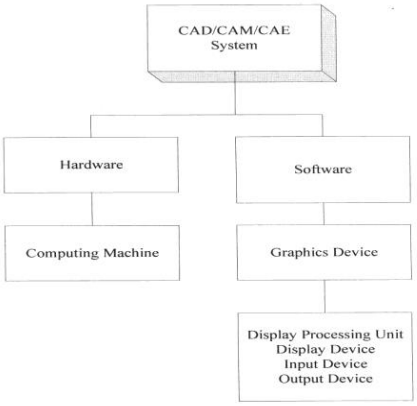
52. What is homogeneous representation?

It is desirable to express all geometric transformations (translation, rotation, rotation, scaling, and mirroring) in the form of matrix multiplications. Representing points by their homogeneous coordinates provides an effective way to unify the description of geometric transformations as matrix multiplications is known as homogeneous representation.

53. List some of the advantages of CAD/CAM systems.

Greater flexibility, Reduced lead times, Reduced inventories, Increased Productivity, Improved customer service, Improved quality, Improved communications with suppliers, Better product design, Greater manufacturing control, Supported integration, Reduced costs, Increased utilization, Reduction of machine tools, Less floor space.

54. Give the general configuration of a CAD computer system.



55. **In what ways CAD can help manufacturing activity? Discuss. (May/June 09)**

CAD helps manufacturing process from the process planning, the outcome of the process planning is a production plan, tools procurement, material order, and machine programming. CAD also helps manufacturing activities by sharing part diagrams, drafts with the DC, CNC, and DNC programming process.

56. **CAD helps in integrating CAM- Justify this statement. (May/June 09)**

CAD helps in integrating CAM because the combination of CAD, Automation, and Manufacturing only leads to CAM. CAM disciplines only satisfy with the complete contribution of the CAD.

57. **How do you specify a plotter for graphics application?**

A plotter is often used to produce large size drawings and assemblies, whereas, a laser jet printer is adequate to provide a 3-D view of a model.

58. **Briefly describe the types of storage devices used in computers. (Nov 08,Jun 09)**

Common types of mass storage devices are floppy discs, hard discs, CD-ROMS and removable storage devices. Another type of mass storage device is the cartridge tape.

59. **In design, what do you understand by synthesis and engineering analysis?**

In Computer-Aided Design Process, synthesis and analysis are two major activities. Synthesis is largely qualitative and hard to capture on computer. Engineering Analysis can be greatly enhanced with computers, once analysis is complete; design evaluation and rapid prototyping are done.

60. **Explain how CAD helps to synthesize a product design and do engineering analysis for getting optimal design.(June 09)**

During synthesis, adesign takesthe formof sketchesand layout drawings that show the relationship among the various product parts. These sketches and drawings can be created using a CAD/CAM system or simply hand drawn on paper. They are used during brainstorming discussions among various design teams and for presentation purposes. The analysis subprocess begins with an attempt to put the conceptual design into the context of engineering sciences to evaluate the performance of the expected product.

61. **What do you understand by the CPU? (Nov 08)**

Like a computer this is the brain of a PLC.

It has several sub-parts.

(a) microprocessor to carry out arithmetic and logic operations.

(b) Memory to store data and programs.

(c) rocess image memory.

(d) Internal timers and counters.

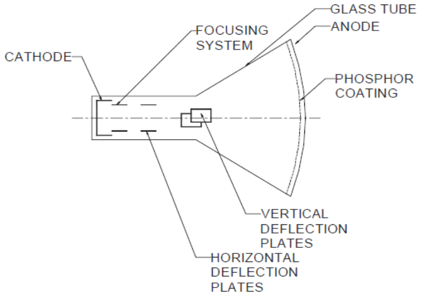
(e) Flags.

62. Bring out clearly the difficulties a design engineer has to face at each of the design stages if they are carried out manually.(Nov 08)

Design engineer has to draft manually in the form of sketches and layout drawings in simple hand-drawn on a paper, if Designer has to change the design slightly; he has to draft it entirely. Similarly in all the design stages the manual works are more difficult to change and may leads to misunderstanding of the drawing from one stages too another.

63. How does a CRT work?

In a color CRT, there are three electron guns, one each for red, green and blue. The phosphor dots for red, green and blue are arranged in a triangular pattern. The individual beams intersect at a shadow mask which directs a red beam to a red phosphor dot and so on.

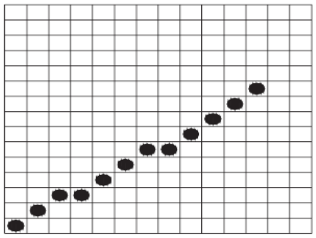


Working principle of CRT

64. Define Raster Scan Technique.

The screen of a monitor can be considered to consist of a large number of minute subdivisions called picture elements (pixels, in short). For example, in a typical SVGA monitor, the screen is divided horizontally into 1024 units and vertically into 768 units.

The resolution of this monitor is then 1024 x 768. A raster CRT graphics device can be considered a matrix of discrete cells each of which can be made bright. Thus it is basically a point plotting device. If a line is to be drawn, it can be approximated by a series of dots close to the path of the line. Figure illustrates this concept.



Line drawing

65. Write briefly on the secondary storage devices used in CAD System. (Nov 08)

The volume of data required by a manufacturing company is so large that secondary storage devices such as hard discs, tapes, CD-ROMs, and other digital storage devices of several gigabyte capacities will be used.

66. What are various output primitives?

A drawing is created by an assembly of points, lines, arcs, circles. For example, drawing consists of several entities. In computer graphics also drawings are created in a similar manner. Each of these is called an entity.

67. What is transformation? How many types of transformations are there to change the geometry?

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.

68. What is a projection? Explain different types of projections?

In drawing practice, a 3-dimensional object is represented on a plane paper. Similarly in computer graphics a 3-dimensional object is viewed on a 2-dimensional display. A projection is a transformation that performs this conversion. Three types of projections are commonly used in engineering practice: parallel, perspective and isometric.

69. What are entities?

A drawing is created by an assembly of points, lines, arcs, circles. For example, drawing consists of several entities. In computer graphics also drawings are created in a similar manner. Each of these is called an entity

70. What do you mean by geometric modeling?

Computer representation of the geometry of a component using software is called a geometric model. Geometric modeling is done in three principal ways. They are:

i. Wire frame modeling

ii. Surface modeling

iii. Solid modeling

71. What are the steps involved in viewing transformation?

Part B -16 MARKS

1. Explain Product life cycle with design process. (Or) Briefly explain the conventional Process of the product cycle in the conventional manufacturing environment. (Nov 08, MAY 2016)

Product Life Cycle:

- ➢ Every product goes through a cycle from birth, followed by an initial growth stage, a relatively stable matured period, and finally into a declining stage that eventually ends in the death of the product as shown schematically in figure 1.1.
- ➢ **Introduction stage:** In this stage the product is new and the customer acceptance is low
  hence the sales are low.
- ➢ **Growth stage:** Knowledge of the product and its capabilities reaches to a growing
  number of customers.
- ➢ **Maturity stage:** The product is widely acceptable and sales are now stable, and it grows
  with the same rate as the economy as a whole grows.
- ➢ **Decline stage:** At some point of time the product enters the decline stage. Its sales start
  decreasing because of a new and a better product has entered the market to fulfill the
  same customer requirements.

- ➢ Figure 1.1.1.shows the life cycle of a typical product. The product begins with a need which is identified based on customers' and markets' demands.
- ➢ The product goes through two main processes from the idea conceptualization to the finished product: The design process and the manufacturing process.

Design process:

- ➢ The product goes through two main processes from inception to a finished product: the design process and the manufacturing process.
- ➢ Synthesis and analysis are the two main sub processes of the design process.

Synthesis:

- ➢ The philosophy, functionality, and uniqueness of the product arc 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.
- ➢ These sketches and drawings can be created using a CAD/CAM system or simply hand-drawn on paper. They are used during brainstorming discussions among various design teams and for presentation purposes.

Analysis:

- 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. The outcome of analysis is the design documentation in the form of engineering drawings (also known as **blueprints**).

Manufacturing Process:

- The manufacturing process begins with the process planning and ends with the actual product.
- Process planning is considered the backbone or the manufacturing process since it attempts to determine the most efficient sequence in which to produce the product.
- A process planner must be aware of the various aspects of manufacturing to plan properly. The planner typically works with the blueprints and may communicate with the design team to clarify or request changes in the design to fit manufacturing requirements.

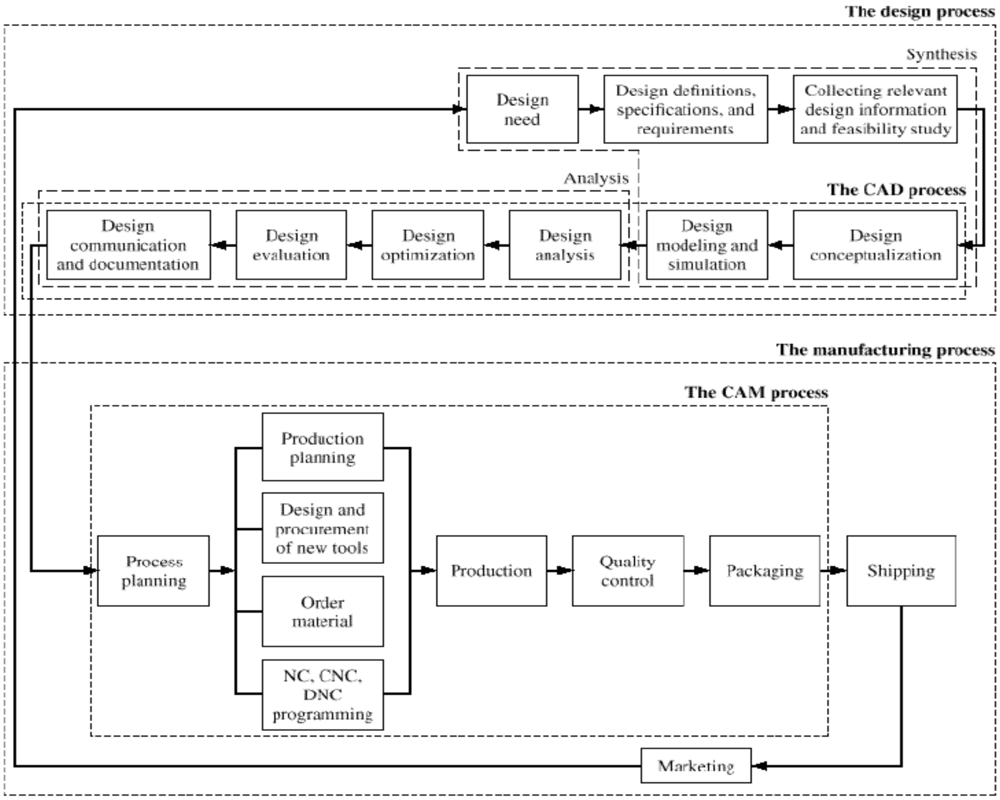


Figure 1.1.Typical product life cycle

- The outcome of the process planning is a production plan, tools procurement, material order, and machine programming. Other special manufacturing needs such as design or jigs and fixtures or inspection gages are planned.

Product Cycle Model:

There are several Product cycle models in industry to be considered, one of the possible product cycle is given below (**Figure 1.2**):

Step 1: Conceive-Imagine, Specify, Plan, Innovate

- The first step is the definition of the product requirements based on company, market and customer. From this requirement, the product's technical data can be defined.
- In parallel, the early concept design work is performed defining the product with its main functional features.
- Various media are utilized for these processes, from paper and pencil to clay mock-up to 3D Computer Aided Industrial Design.

Step 2: Design-Describe, Define, Develop, Test, Analyze and Validate

- This is where the completed design and development of the product begins, succeeding to prototype testing, through pilot release to final product.
- It can also involve redesign and ramp for
- Improvement to existing products as well as planned obsolescence. The main tool used for design and development is CAD.
- This can be simple 2D drawing / drafting or 3D parametric feature based solid/surface modeling. This step covers many engineering disciplines including: electronic, electrical, mechanical, and civil. Besides the actual making of geometry there is the analysis of the components and assemblies.

Optimization, Validation and Simulation activities are carried out using Computer Aided Engineering (CAE) software. These are used to perform various tasks such as: Computational Fluid Dynamics (CFD); Finite Element Analysis (FEA); and Mechanical Event Simulation (MES). Computer Aided Quality (CAQ) is used for activities such as Dimensional tolerance analysis.

Step 3: Realize-Manufacture, Make, Build, Procure, Produce, Sell and Deliver

- Once the design of the components is complete the method of manufacturing is finalized. This includes CAD operations such as generation of CNC Machining instructions for the product’s component as well as tools to manufacture those components, using integrated Computer Aided Manufacturing (CAM) software.
- It includes Production Planning tools for carrying out plant and factory layout and production simulation. Once details components are manufactured their geometrical form and dimensions can be verified against the original data with the use of Computer Aided Inspection Equipment (CAIE). Parallel to the engineering tasks, sales and marketing work take place. This could consist of transferring engineering data to a web based sales configuration.

Figure 1.2.Product Cycle Model

Step 4: Service-Use, Operate, Maintain, Support, Sustain, Phase-out, Retire, Recycle and Disposal

The final step of the lifecycle includes managing of information related to service for repair and maintenance, as well as recycling and waste management information. This involves using tools like Maintenance, Repair and Operations Management software.

2. Briefly explain about design process. (AU-NOV-2016)

Design Process

- The design process includes series of steps that engineers apply in making functional products and processes.
- The parts of the process often need to be repeated many times before production of a product can start.
- The parts that get iterated and the number of such design cycles in any given project can be highly changeable.
- One method of the engineering design process focuses on the following common aspects:

1. Research

- A considerable amount of time is used on research, or finding information. Consideration should be given to the available applicable literature, issues and successes linked with available solutions, and need of marketplaces.
- The basis of information should be significant, including existing results. Reverse engineering can be a successful technique if other solutions are available in the market. Added sources of information include the trade journals, available government documents, local libraries, vendor catalogs and personal organizations.

2. Feasibility assessment

- The feasibility study is an analysis and assessment of the possible of a proposed design which is based on detail investigation and research to maintain the process of decision creation.
- The feasibility assessment helps to focus the scope of the project to spot the best situation.
- The purpose of a feasibility assessment is to verify whether the project can continue into the design phase.

3. Conceptualization

- A Concept Study is the stage of project planning that includes developing ideas and taking into account the all features of executing those ideas.
- This stage of a project is done to reduce the likelihood of assess risks, error and evaluate the potential success of the planned project.

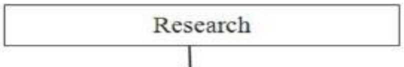


Fig.2.1. Design Process

4. Establishing the design requirements

- Establishing design requirements is one of the most essential elements in the design practice, and this task is usually performed at the same time as the feasibility analysis.
- The design requirements control the design of the project all over the engineering design process. A few design requirements comprise maintainability, hardware and software parameters, availability, and testability.

5. Preliminary design

- The preliminary design fills the gap between the design concept and the detailed design phase. During this task, the system configuration is defined, and schematics, diagrams, and layouts of the project will offer early project configuration.
- In detailed design and optimization, the parameters of the part being produced will change, but the preliminary design focuses on creating the common framework to construct the project.

6. Detailed design

- The next phase of preliminary design is the Detailed Design which may include of procurement also.
- This phase builds on the already developed preliminary design, aiming to further develop each phase of the project by total description through drawings, modeling as well as specifications.
- The advancement CAD programs have made the detailed design phase more competent. This is because a CAD program can offer optimization, where it can shrink volume without compromising the part's quality.
- It can also calculate displacement and stress using the FEM to find stresses throughout the part. It is the responsibility of designer to find whether these stresses and displacements are acceptable, so the part is safe.

7. Production planning and tool design

- The production planning and tool design is more than planning how to mass-produce the project and which tools should be used in the manufacturing of the component.
- Tasks to complete in this stage include material selection, identification of the production processes, finalization of the sequence of operations, and selection of jigs, fixtures, and tooling. This stage also includes testing a working prototype to confirm the created part meets qualification standards.
- With the finishing of qualification testing and prototype testing, the design process is completed.

Design process---Shigley Model

It involves six steps

Recognition of need

Definition of problem

Synthesis

Change the design

Analysis
and
Optimizatio
n

Yes

Can the design be improved?

Evaluation

No

No

Presentation

Design impossible for
the given specifications

Recognition of need

- Realization of problem exists in the design or in the product
- Identification of some defect in a current machine design
- New product opportunity

Definition of problem

- Specification of the item to be designed
- Functional characteristics, cost, quality, performance, etc.

Synthesis

- Preliminary ideas are developed through research of similar product or designs in use.

Analysis and Optimization

- Suitability for the specified design constraints
- If not suitable or design fails to satisfy the constraints
- Then redesign or modified iteration continues until the proposed design meet the specifications (or) until feasibility is achieved.
- Then components, sub-assemblies or sub-systems are then synthesized into the final overall system in a similar iterative manner.

Evaluation

- Prototyping
- Testing
- Quality
- Reliability testing

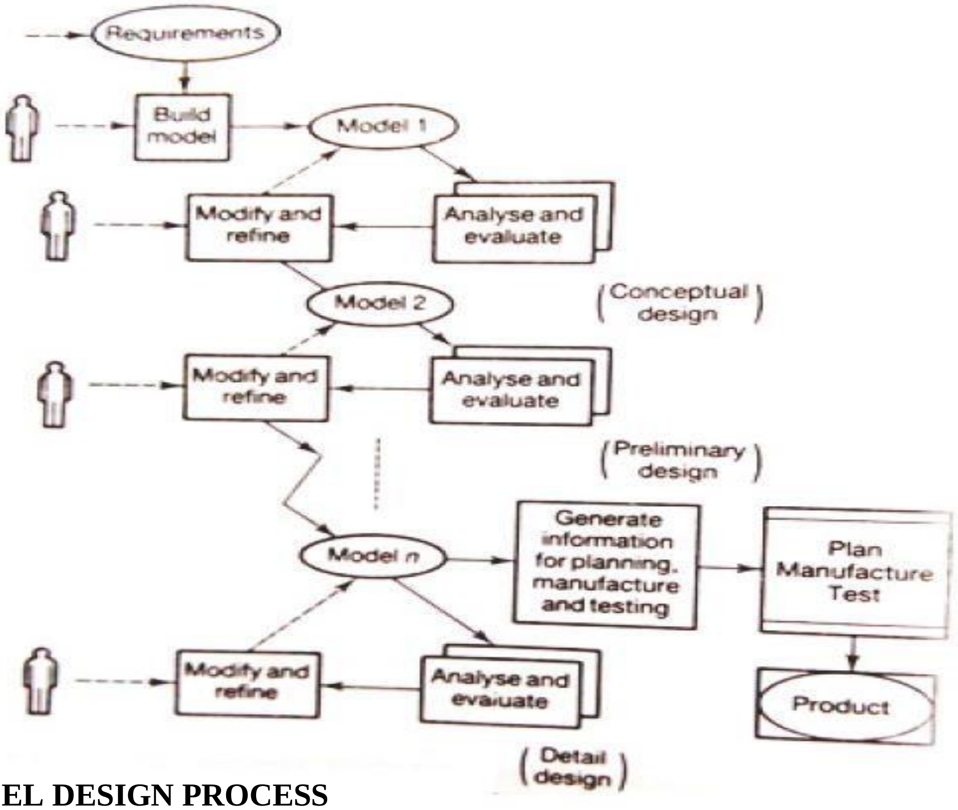
Presentation

- Documentation of the design through drawings, material specifications, assembly lists, etc.

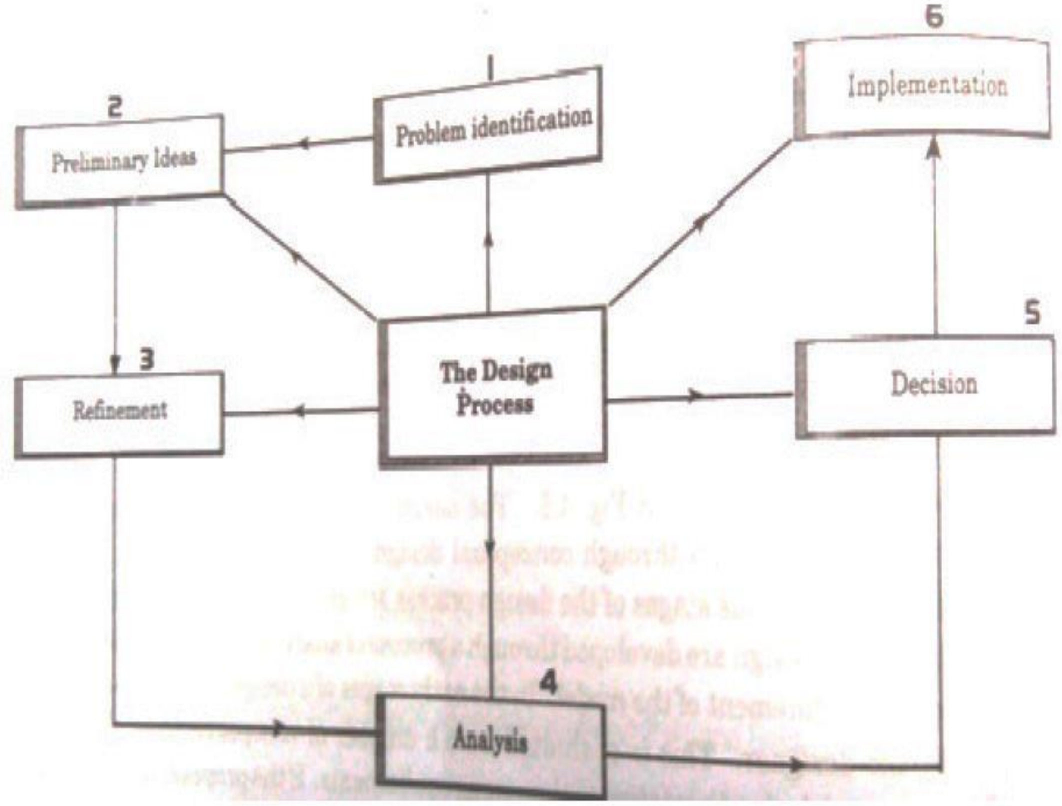
Pahl and Beitz Model:

1. Clarification of the task: Collection of information, constraints on the design.
2. Conceptual design: establishment of the functions to be included in the design,
3. Embodiment design: problems are resolved and weak aspects are eliminated
4. Detail design: The dimensions, tolerance, materials and form of individual components of the design are specified in design for subsequent manufacturing.

OHSUGA MODEL DESIGN PROCESS

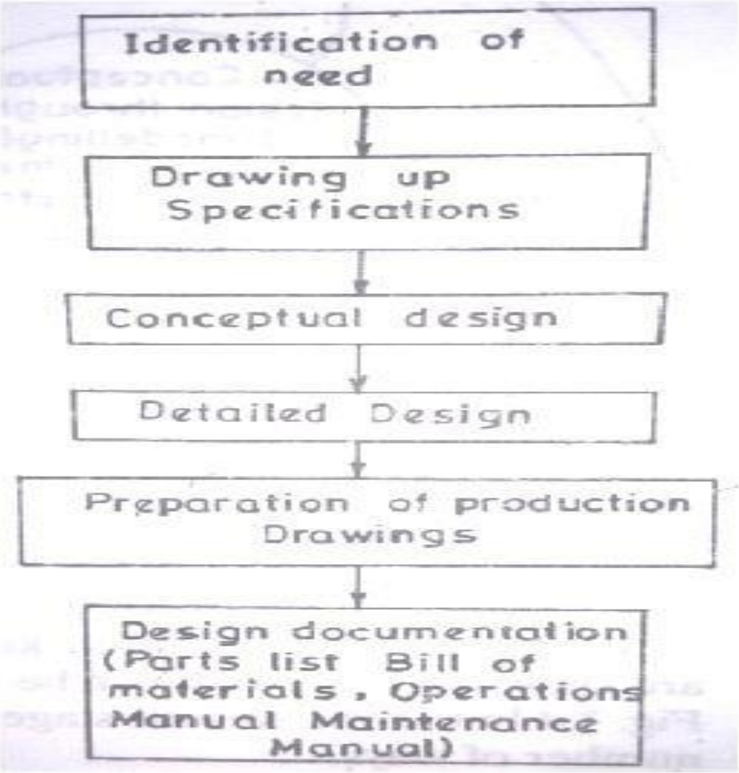


EARLE MODEL DESIGN PROCESS



3. Write briefly on Sequential Engineering.

Sequential Engineering:


Identification of need
Drawing up Specifications
Conceptual design
Detailed Design
Preparation of production Drawings
Design documentation (Parts list Bill of materials, Operations Manual Maintenance Manual)

- ➢ With today's marketplace becoming more and more competitive, there is an ever-increasing pressure on companies to respond quickly to market needs, be cost effective, reduce lead-times to market and deliver superior quality products.
- ➢ Traditionally, design has been carried out as a sequential set of activities with distinct non overlapping phases. In such an approach, the life-cycle of a product starts with the identification of the need for that product. These needs are converted into product requirements which are passed onto the design department.
- ➢ The designers design the product's form, fit, and function to meet all therequirements, and pass on the design to the manufacturing department.
- ➢ After the product is manufactured it goes through the phases of assembly, testing, and installation. This type of approach to life-cycle development is also known as `over the wall' approach, because the different life-cycle phases are hidden or isolated from each other.
- ➢ Each phase receives the output of the preceding phase as if the output had been thrown over the wall. In such an approach, the manufacturing department, for example, does not know what it will actually be manufacturing until the detailed design of the product is over.
- ➢ There are a lot of disadvantages of the sequential engineering process. The designers are responsible for creating a design that meets all the specified requirements. They are usually not concerned with how the product will be manufactured or assembled. Problems and in consistencies in the designs are therefore, detected when the product reaches into the later phases of its life-cycle.
- ➢ At this stage, the only possible option is to send the product back for a re-design. The whole process becomes iterative and it not until after a lot of re-designs has taken place that the product is finally manufactured.
- ➢ Because of the large number of changes, and hence iterations, the product's introduction

to market gets delayed.

- In addition, each re-design, re-work, re-assembly etc. in curscost, and therefore the resulting product is costlier than what it was originally thought to be.
- The market share is lost because of the delay in product's introduction to market, and customer faith islost. All this is undesirable.

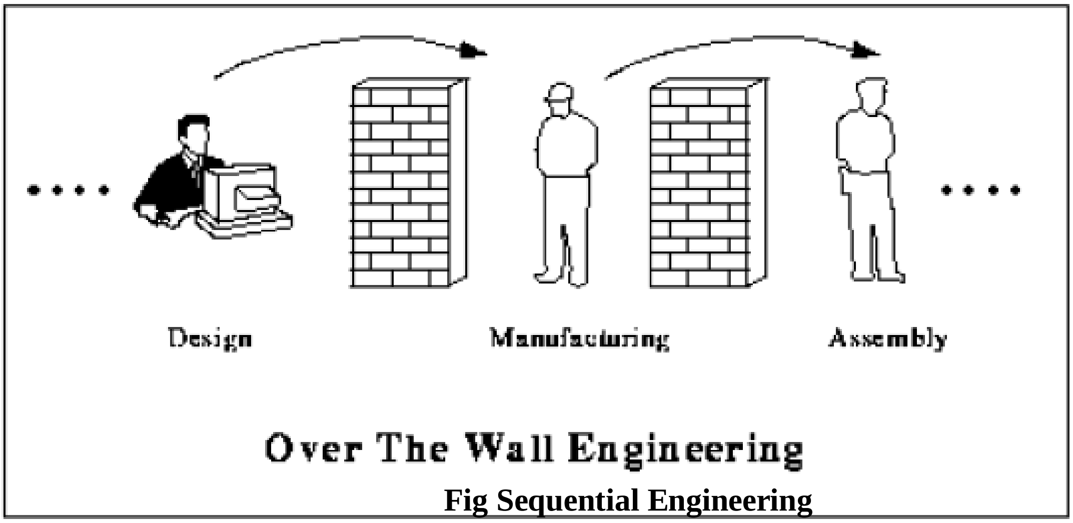
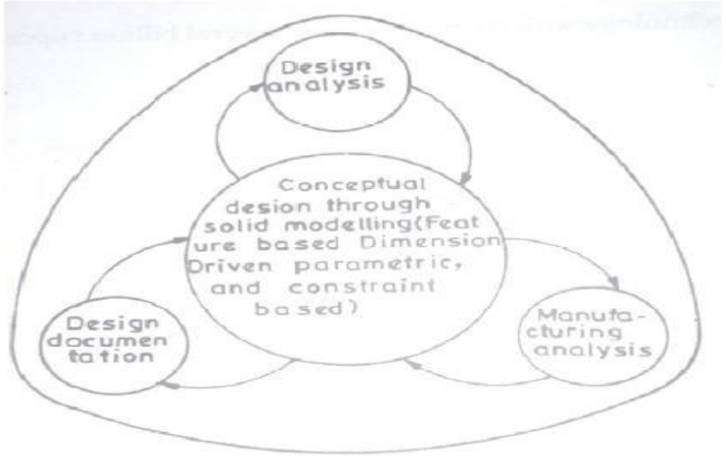


Fig Sequential Engineering

4. Write briefly on concurrent Engineering. (AU NOV 2015, MAY 2016)

- Concurrent Engineering is a dramatically different approach to product development in which various life-cycle aspects are considered simultaneously right from the early stages of design.
- These life-cycle aspects include product's functionality, manufacturability, testability, assimilability, maintainability, and everything else that could be affected by the design.
- In addition, various life-cycle phases overlap each other, and there in no "wall" between These phases.
- The completion of a previous life-cycle phase is not a pre-requisite for the start of the next life-cycle phase. In addition, there is a continuous feedback between these life-cycle phases so that the conflicts are detected as soon as possible.



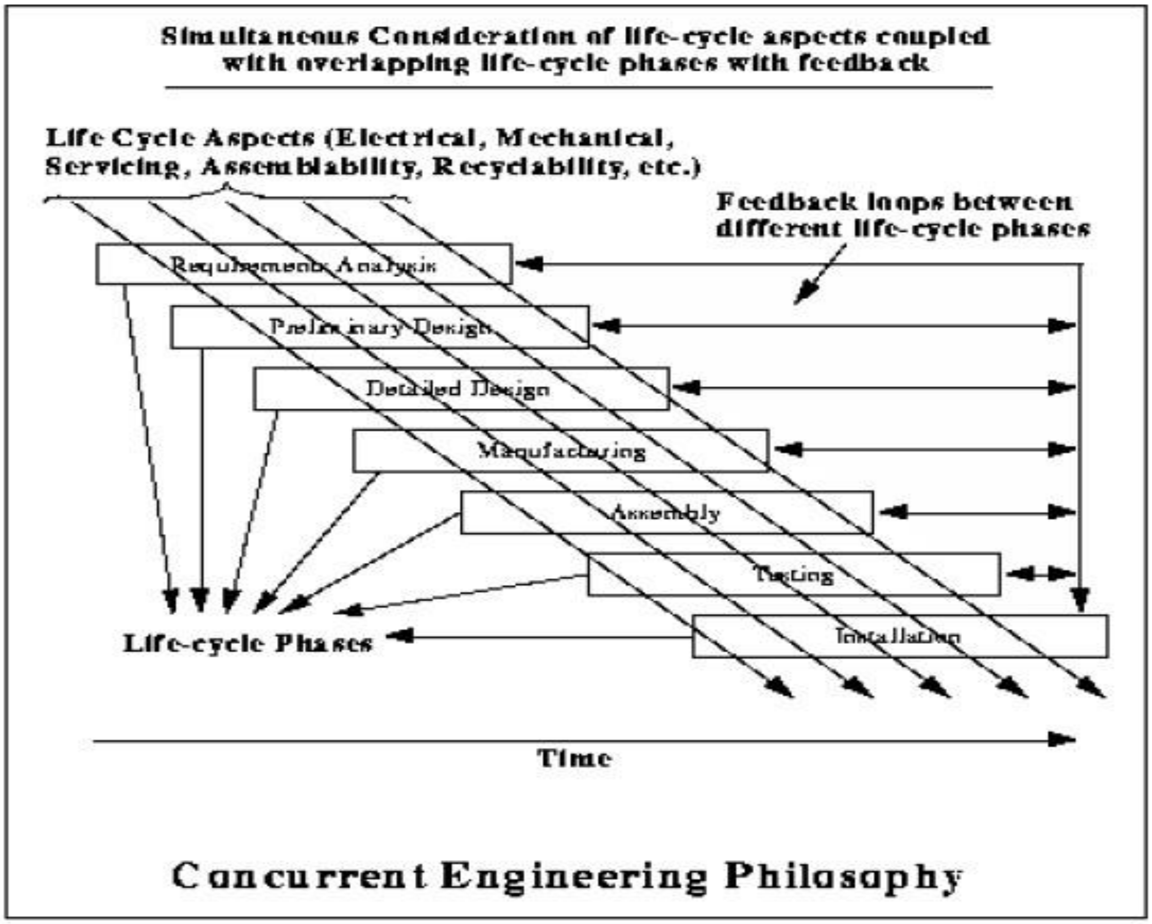
Design analysis
Conceptual desion through solid modelling(Feature based Dimension Driven parametric, and constraint based)
Design documentation
Manufacturing analysis

Concurrent engineering

- The concurrent approach results in less number of changes during the later phases of product life-cycle, because of the fact that the life-cycle aspects are being considered all through the design.
- The benefits achieved are reduced lead times to market, reduced cost, higher quality,

greater customer satisfaction, increased market share etc. Sequential engineering is the term used to describe the method of production in a linear format. The different steps are done one after another, with all attention and resources focused on that one task. After it is completed it is left alone and everything is concentrated on the next task.

- In concurrent engineering, different tasks are tackled at the same time, and not necessarily in the usual order. This means that info found out later in the process can be added to earlier parts, improving them, and also saving a lot of time.
- Concurrent engineering is a method by which several teams within an organization work simultaneously to develop new products and services and allows a more stream lined approach.
- The concurrent engineering is a non-linear product or project design approach during which all phases of manufacturing operate at the same time -simultaneously. Both product and process design run in parallel and occur in the same time frame.
- Product and process are closely coordinated to achieve optimal matching of requirements for effective cost, quality, and delivery. Decision making involves full team participation and involvement.
- The team often consists of product design engineers, manufacturing engineers, marketing Personnel, purchasing, finance, and suppliers.


Simultaneous Consideration of life-cycle aspects coupled with overlapping life-cycle phases with feedback
Life Cycle Aspects (Electrical, Mechanical, Servicing, Assemblability, Recyclability, etc.)
Feedback loops between different life-cycle phases
Requirements Analysis
Preliminary Design
Detailed Design
Manufacturing
Assembly
Testing
Installation
Life-cycle Phases
Time
Concurrent Engineering Philosophy

5. Write briefly on Computer Aided Design.

- CAD is the intersection of Computer Graphics, Geometric modeling and Design tools (Fig.4.1.). The concepts of computer graphics and geometric modeling and must be used innovatively to serve the design process. CAD is the function of computer systems to support in the creation, modification, analysis, or optimization of a design.

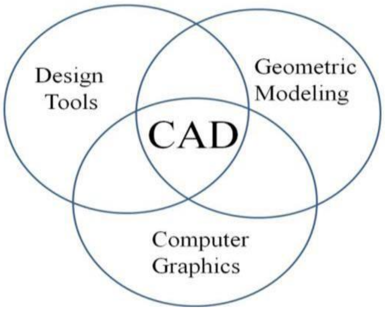


Fig.4.1.CAD

- CAD software for design uses either vector-based graphics to explain the objects of traditional drafting, or may also develop raster graphics showing the overall look of designed objects. During the manual drafting of engineering drawings, the output of CAD must convey information, like dimensions, materials, processes, and tolerances.
- CAD is a significant industrial art used in many purposes, including industrial and architectural design, shipbuilding, automotive, and aerospace industries, and many more. CAD is also extensively used to create computer animation for special effects in movies, and technical manuals, frequently called as Digital Content Creation.
- CAD software packages provide the designer with a multi window environment with animation which is regularly used in Digital Content Creation. The animations using wire frame modeling helps the designer to see into the interior of object and to observe the behaviors of the inner components of the assembly during the motion.

CAD Tools

- The CAD tools are mainly using for graphics applications and modeling. Aids such a color, grids, geometric modifiers and group facilitate structural geometric models.
- Visualization is achieved through shaded components and animation which focus design conceptualization, communication and interference detection. FEM packages provide optimization in shape and structure.
- Adding tolerances, tolerance analysis and investigating the effect of manufacturing on the design can perform by utilizing CAD tools.

6.Write short notes on Scope and applications of CAD/CAM Scope

of CAD/CAM

- At the core of the CAD and CAM processes is a geometric model of the product under design. Activities of the CAD process include mass properties, finite element analysis, dimensioning, tolerance, assembly modeling, generating shaded images, and documentation and drafting.
- Activities of the CAM process include CAPP (Computer Aided Process Planning), NC (numerical control) programming, design of injection mold, CMM (coordinate measuring machine) verification, inspection, assembly via robots, and packaging.
- The CAD process and its tools utilize three disciplines: geometric modeling, computer graphics, and design. The CAM process utilizes the disciplines of CAD itself,

manufacturing, and automation.

CAD/CAM Applications

- There are considerable numbers of applications for the various existing CAD/CAM systems. For example, there are mechanical, electrical, and architectural CAD and CAM products.
- An inspection of these various systems reveals that they have a generic structure and common modules. Awareness of such structure and modules enables users to better understand system functions for both evaluation and training purposes. The major

available modules are

- The **geometric engine (module)** is the heart of a CAD/CAM system. It provides users with functions to perform geometric modeling and construction. Editing and manipulation of existing geometry, drafting and documentation. The typical modeling operations that users can engage in arc model creation, cleanup, documentation, and printing/plotting. Shaded images can be generated as part of model documentation.

- The **applications module** varies from one software system to another. However, there are common applications shared by most CAD/CAM systems. Mechanical applications include mass property calculations, assembly analysis, tolerance analysis and synthesis, sheet metal design, finite element modeling and analysis, mechanisms analysis, animation techniques, and simulation and analysis of plastic injection molding. Manufacturing applications include CAPP, NC, CIM, robot simulation, and group technology.
- The **programming module** allows users to customize systems by programming them to fit certain design and manufacturing tasks. A CAD/CAM system requires advanced knowledge of the system architecture, its database format, and a high-level programming language such as C, C++, Java, Scheme, or others.
- The **communications module** is crucial if integration is to be achieved between the CAD/CAM system, other computer systems, and manufacturing facilities. It is common to network the system to transfer the CAD database of a model for analysis purposes or to transfer its CAM database to the shop floor for production. This module also serves the purpose of translating databases between CAD/CAM systems using graphics standards such as IGES and STEP.
- The **collaborative module** is emerging as an outcome of the widespread of the World Wide Web and the Internet. This module supports collaborative design. Various design teams in different geographical locations can work concurrently on the same part, assembly, or drawing file in real time over the Web. One team can make changes that other teams can view and accept or reject.
- The various CAD/CAM systems support engineering design, analysis, and manufacturing applications. They all have a flexible pricing structure that allows customers to add on applications as needed.

7. Explain the benefits of CAD and reason for implementation of CAD. Benefits of CAD

1. Productivity Improvement in Design

- Depends on Complexity of drawing,
- Degree of repetitiveness of features in the designed parts,
- Degree of symmetry in the parts,
- Extensive use of library of user

2. Shorter Lead Times
3. Flexibility in Design
4. Design Analysis
5. Fewer Design Error
6. Standardization of Design, Drafting and Documentation
7. Drawings are more understandable
8. Improved Procedures of Engineering Changes
9. Benefits in Manufacturing:
    - Tool and fixture design for manufacturing
    - Computer Aided process planning
    - Preparation of assembly lists and bill of materials
    - Computer aided inspection
    - Coding and classification of components
    - Production planning and control
    - Preparation of numerical control programs for manufacturing the parts on CNC machines
    - Assembly sequence planning

Reason for implementation of CAD

- To increase the productivity of the designer
- To improve the Quality of Design
- To improve Documentation
- To create a Database for manufacturing

8. Write short notes on: CAD, CAD process, components of CAD system, and advantages of CAD systems. (Or) CAD system Architecture (MAY 2016).

Computer-Aided Design (CAD)

- CAD can be defined as the use of computer systems to perform certain functions in the design process. Use of computer systems to assist in the creation, modification, analysis, and optimization of a design
- In general, a Computer Aided Design (CAD) package has three components: a) Design, b) Analysis, and c) Visualization. A brief description of these components follows.

a) **Design**: Design refers to geometric modeling, i.e., 2-D and 3-D modeling, including, drafting, part creation, creation of drawings with various views of the part, assemblies of the parts, etc.

b) **Analysis**: Analysis refers to finite element analysis, optimization, and other number crunching engineering analyses. In general, a geometric model is first created and then the model is analyzed for loads, stresses, moment of inertia, and volume, etc.

c) **Visualization**: Visualization refers to computer graphics, which includes: rendering a model, creation of pie charts, contour plots, shading a model, sizing, animation, etc.

Each of these three areas has been extensively developed in the last 30 years Several books are written on each of these subjects and courses are available through the academic institutions and the industry.

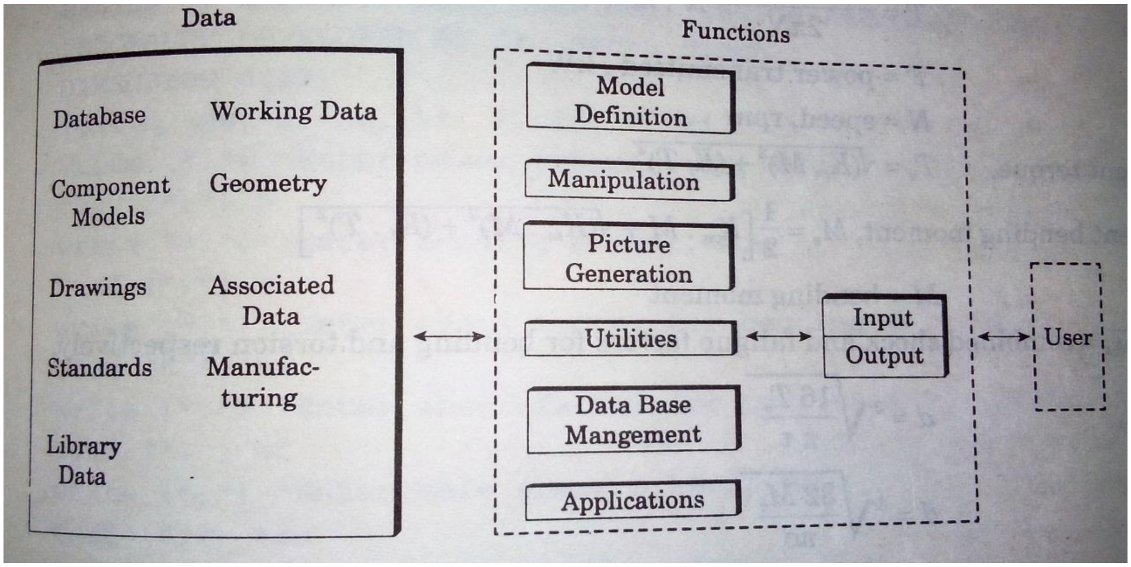
- Typical tools:
    - Tolerance analysis
    - Mass property calculations
    - Finite-element modeling and visualization

- Defines the geometry of the design

CAD system Architecture:

CAD system consists of

- Hardware
- Software
- Data
- Human knowledge and activities



Model definition: add geometry elements to a model of the form of components.

Model Manipulation: to move, copy, delete, edit or otherwise modify elements in the components.

Picture generation: To generate images of the design model on a computer screen or on design model.

User interaction: To handle commands input by the user and to present output to the user about the operation of the system.

Data base management: Fr the management of the files that make up the database.

Applications: To generate information for evaluation, analysis or manufacturing.

Computer-Aided Design Process

Two types of activities: synthesis and analysis

- Synthesis is largely qualitative and hard to capture on computer.
- Analysis can be greatly enhanced with computers.
- Once analysis is complete, design evaluation- rapid prototyping are done.

Fig 6.1.Components of CAD/CAM/CAE Systems

CAD Hardware

There are basically two types of devices that constitute CAD hardware: a) Input devices, and b) Output devices.

Input Devices

These are the devices that we use for communicating with computer, and providing our input in the form of text and graphics. The text input is mainly provided through keyboard. For graphic input, there are several devices available and used according to the work environment. A brief description of these devices is given here.

Mouse: This is a potentiometric device, which contains several variable resistors that send signals to the computer. The functions of a mouse include locating a point on the screen, sketching, dragging an object, entering values, accepting a software command, etc. Joystick and trackballs are analogous to a mouse device, and operate on the same principle.

Digitizers: Digitizers are used to trace a sketch or other 2-D entities by moving a cursor over a flat surface (which contains the sketch). The position of the cursor provides a feedback to the computer connected with the device. There are electrical wires embedded in orthogonal directions that receive and pass signals between the device and the computer. The device is basically a free moving pen shaped stylus, connected to a tablet.

Light Pens: Lockheed's CADAM software utilized this device to carry out the graphic input. A light pen looks like a pen and contains a photocell, which emits an electronic signal. When the pen is pointed at the monitor screen, it senses light, which is converted to a signal. The signal is sent to the computer, for determination of the exact location of the pen on the monitor screen.

Touch Sensitive Screens: This device is embedded in the monitor screens, usually, in the form of an overlay. The screen senses the physical contact of the user. The new generation of the Laptop computers is a good example of this device.

Other Graphic Input Devices: In addition to the devices described above, some CAD software will accept input via Image Scanners, which can copy a drawing or schematic with a camera and light beam assembly and convert it into a pictorial database.

The devices just described are, in general, independent of the CAD package being used. All commercial CAD software packages contain the device drivers for the most commonly used input devices. The device drivers facilitate a smooth interaction between our input, the software, and the computer. An input device is evaluated on the basis of the following factors:

- Resolution
- Accuracy
- Repeatability
- Linearity

Output Devices

After creating a CAD model, we often need a hard copy, using an output device. Plotters and printers are used for this purpose.

A **plotter** is often used to produce large size drawings and assemblies, whereas, a laser jet printer is adequate to provide a 3-D view of a model. Most CAD software require a plotter for producing a shaded or a rendered view.

CAD Software

- ➢ CAD software is written in FORTRAN and C languages. FORTRAN provides the number crunching, whereas, C language provides the visual images.
- ➢ The modern CAD software utilizes the open architecture system, i.e., software vendors do not design and manufacture their own hardware. Third party software can be used to augment the basic CAD package.
- ➢ Most popular CAD package will facilitate integration of the Finite Element Analysis and other CAD software from more than one vendor. For example, IDEAS preprocessor can work with almost all the FEA packages for pre and post analyses.
- ➢ Networking is an important consideration in applications of CAD software. A model created by one engineer must be readily accessible to others in an organization, which is linked by a LAN or other means. The designer, analyst, management, marketing, vendor, and others generally share a model. This is the concurrent engineering in action, mentioned earlier.

CAD Platform

- ➢ In general, we can run CAD software on three different CAD platforms: Mainframe, Workstation, and PC.

CAD PLATFORMS

MAINFRAME WORKSTATIONS PCs

MAINFRAME	**WORKSTATIONS**	**PCs**
Large Data storage	Medium size data storage	Limited data storage
Networked	Networked	Can be networked
Expensive	Relatively inexpensive	Inexpensive

Need interface language	Runs on Unix	Run on MS-Windows
No user control	No user control	User controlled Good for
Large corporations	Good for small companies	Good for all users

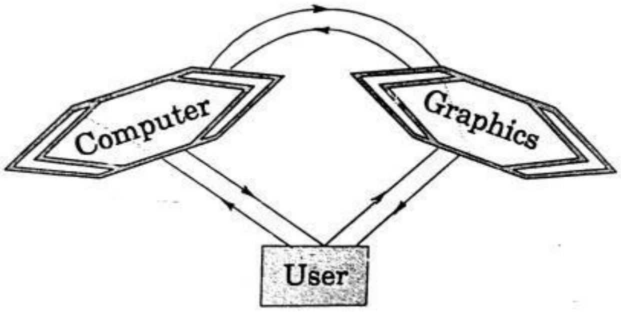
Advantages of CAD/CAM systems

- ➢ Greater flexibility, Reduced lead times, Reduced inventories, Increased Productivity, Improved customer service, Improved quality, Improved communications with suppliers,

  Better product design, Greater manufacturing control, Supported integration, Reduced costs, Increased utilization, Reduction of machine tools, Less floor space

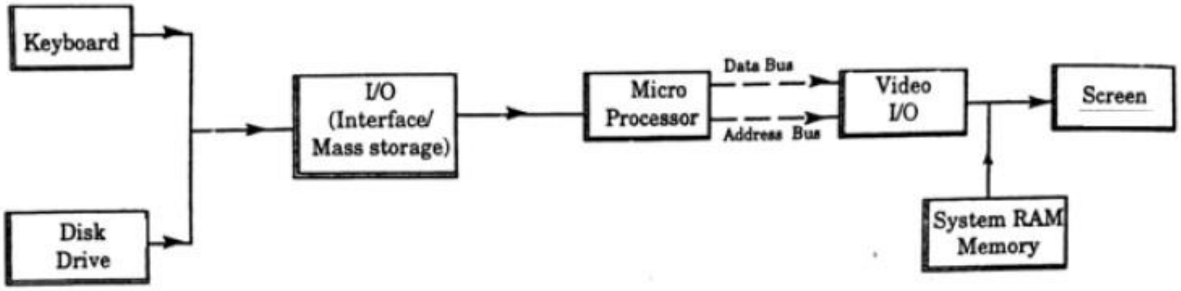
9. Explain computer graphics. (Or) What is meant by Interactive Computer Graphics? Explain its various elements.(Nov08)

- ➢ Computer Graphics is defined as creation, storage, and manipulation of pictures and Drawings by means of a digital computer
- ➢ It is an extremely effective medium for communication between people and computers
- ➢ Computer graphics studies the manipulation of visual and geometric information
- ➢ using computational techniques
- ➢ It focuses on the mathematical and computational foundations of image generation and processing rather than purely aesthetic issues.



Concept of Interactive computer Graphics

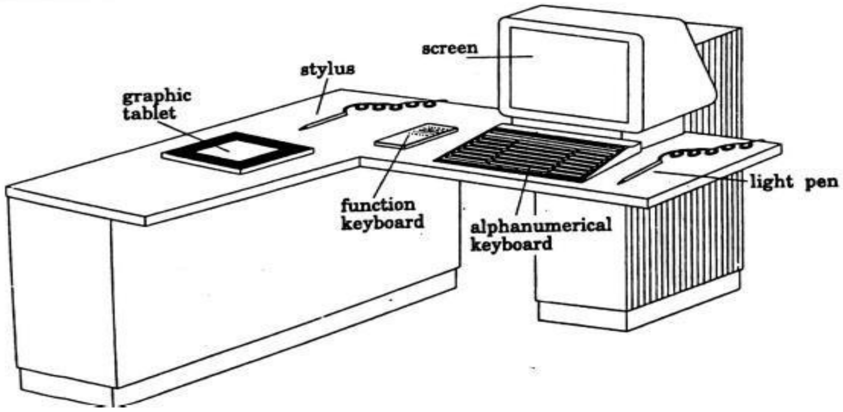
- ➢ In Interactive Computer Graphics (ICG) the user interacts with the compute and comprises the following important functions:
- ➢ **Modeling:** which is concerned with the description of an object in terms of its spatial Co ordinates, lines, areas, edges, surfaces, and volume
- ➢ **Storage:** which is concerned with the storage of the model in the memory of the computer
- ➢ **Manipulation:** which is used in the construction of the model from basic primitives in Combination with Boolean algebra
- ➢ **Viewing:** in the case the computer is used to look at the model from a specific angle and presents on its screen what it sees.



Typical Hardware setup of a Graphic System

WORKSTATION

- A workstation comprises of the devices that allow the user to create and design objects, using both graphic and non-graphic instructions and data. A Stand alone workstation refers to CAD workstations that can process data and output information independent of other computer systems or workstations. It includes its own software, hardware, and peripherals.



A typical work station

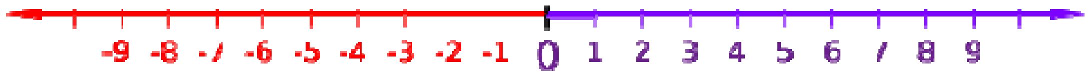
10. Briefly explain the Coordinate Systems and its types

- A coordinate system is one which uses one or more numbers, or coordinates, to uniquely determine the position of a point or other geometric element on a manifold such as Euclidean space.

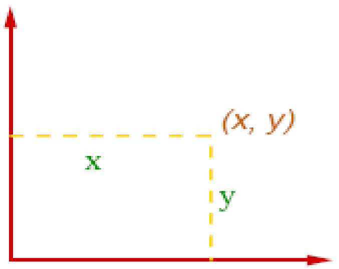
Common coordinate systems are:

Number line

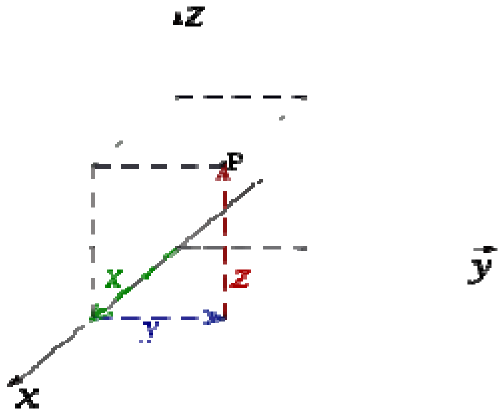
- The simplest example of a coordinate system is the identification of points on a linewith real numbers using the number line In this system, an arbitrary point O (the origin) is chosen on a given line.
- The coordinate of a point P is defined as the signed distance from O to P, where the signed distance is the distance taken as positive or negative depending on which side of the line P lies. Each point is given a unique coordinate and each real number is the coordinate of a unique point.



Cartesian coordinate system [(x,y) and (x,y,z)]

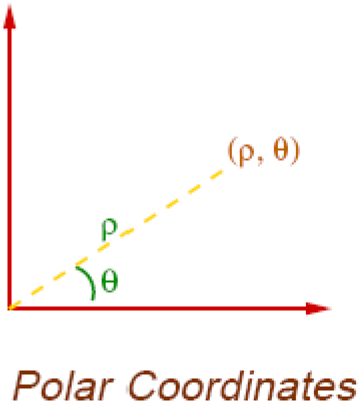


Cartesian Coordinates



Polar coordinate system (ρ,θ)

- Another common coordinate system for the plane is the polar coordinate system. A point is chosen as the pole and a ray from this point is taken as the polar axis.



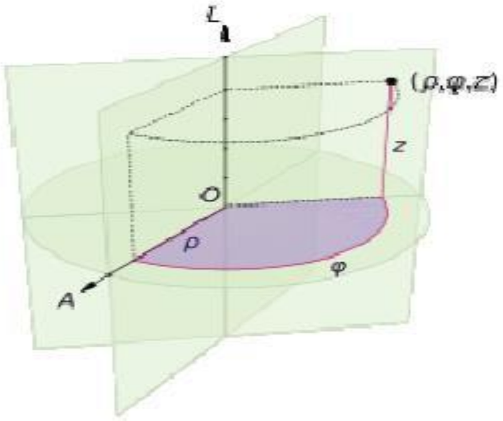
Polar Coordinates

$\rho = r$

- For a given angle θ, there is a single line through the pole whose angle with the polar axis is θ (measured counter clockwise from the axis to the line). Then there is a unique point on this line whose signed distance from the origin is r for given number r.
- For a given pair of coordinates (r, θ) there is a single point, but any point is represented by many pairs of coordinates. For example (r, θ), (r, θ+2π) and (−r, θ+π) are all polar coordinates for the same point. The pole is represented by (0,θ) for any value of θ.

Cylindrical Coordinate systems

- A cylindrical coordinate system is a three-dimensional coordinate system that specifies point positions by the distance from a chosen reference axis, the direction from the axis relative to a chosen reference direction, and the distance from a chosen reference plane perpendicular to the axis. The latter distance is given as a positive or negative number depending on which side of the reference plane faces the point.
- The origin of the system is the point where all three coordinates can be given as zero. This is the intersection between the reference plane and the axis.
- The axis is variously called the cylindrical or longitudinal axis, to differentiate it from the polar axis, which is the ray that lies in the reference plane, starting at the origin and pointing in the reference direction.



- The distance from the axis may be called the radial distance or radius, while the angular coordinate is sometimes referred to as the angular position or as azimuthu.

θ is elevation:

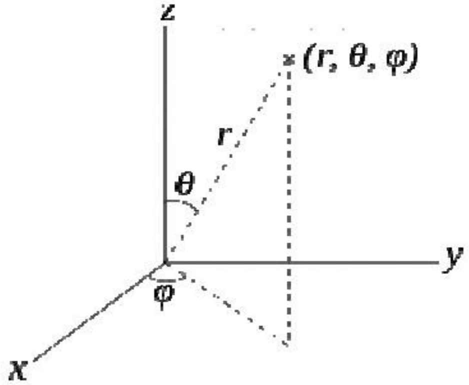
$$\rho = r\cos\theta$$
$$\varphi = \varphi$$
$$z = r\sin\theta$$

θ is inclination:

$$\rho = r\sin\theta$$
$$\varphi = \varphi$$
$$z = r\cos\theta$$

Spherical Coordinate systems

- A spherical coordinate system is a coordinate system for three-dimensional space where the position of a point is specified by three numbers: the radial distance of that point from a fixed origin, its polar angle measured from a fixed zenith direction, and the azimuth angle of its orthogonal projection on a reference plane that passes through the origin and is orthogonal to the zenith, measured from a fixed reference direction on that plane.
- The radial distance is also called the radius or radial coordinate. The polar angle may be called co-latitude zenith angle, normal angle, or inclination angle



θ is elevation:

$$r = \sqrt{\rho^2 + z^2}$$
$$\theta = \arctan(z/\rho)$$
$$\varphi = \varphi$$

θ is inclination:

$$r = \sqrt{\rho^2 + z^2}$$
$$\theta = \arctan(\rho/z)$$
$$\varphi = \varphi$$

Homogeneous coordinate system

- Three dimensional representation of a two dimensional plane is called Homogeneous Co -ordinates. The respective system is called Homogeneous coordinate system.

11. Explain the concepts of translation, scaling and rotators in 2-D transformation. (JAN 2014)

- 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. When we sweep a curve, it 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.

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.

Two-Dimensional (2D) Geometric Transformation:

- ➢ A **geometric transformation** is an operation that modifies its shape, size, position, orientation etc. with respect to its current configuration operating on the vertices (position vectors).

Some of the important 2D transformations include:

a. Translation
b. Scaling
c. Rotation
d. Reflection
e. Shear

2D translation

Translation is nothing but moving an object across the screen from one position to another.

The translation transformation positions the object to a new location.

- ➢ Translation is the process of moving an object from one position to another.
- ➢ The translation transformation positions the object to a new location.
- ➢ The translation is accomplished by adding the coordinates of each corner point the distance through which the object is to moved.

Translation of a point:

- ➢ Consider a 2D point P, having coordinates x,y. To translate point P by a distance Δ� in x-direction and Δ�in y-direction, a translation matrix T is added to the original matrix. Now the point has new coordinates $P'(�', �')$

The translation distance (Δ�,Δ�) is called translation vector.

$P'(�', �')$

Δy

(�, �)

Δx

(0,0
)

Translation of a point

Let

P= Original position of the point,

P’= New position of the point, and

T= Translation matrix

tx(Δ�), ty(Δ�)are the Translation distance along the x and yaxis respectively.

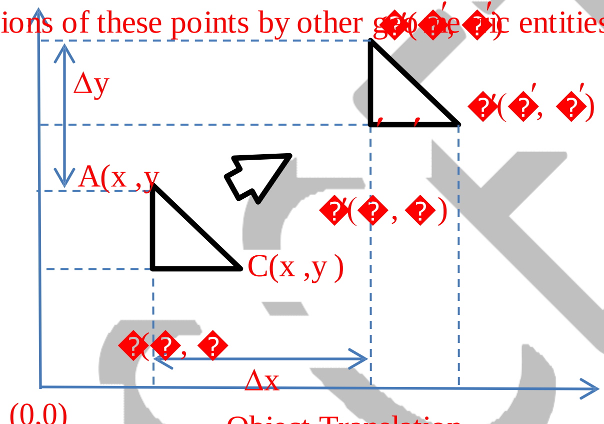
$$x' = x + t_x, \quad y' = y + t_y$$

$$\mathbf{P} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \mathbf{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

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

Translation of an object:

When the object is to moved, all the points of it are to be translated. The translation of keypoints and connections of these points by other geometric entities like lines, arcs, etc.



Object Translation

)

�� �� ��
�� �� ��
= ��
� + [Δ� Δ�]
�
�� ��
��
�
�

Example: Translate a triangle ABC with co-ordinates A (1,1), B(5,2), C(3,3), about the origin by 3 units in the x-direction and 2 units in the y-direction.

Solution:

Given that Δ�= 3 a�dΔ� = 2

�� �
��
And ��
�

VPSCET

1 1

= 5 2

�

3 3

� ��

��′ ��′ ��

��′ ��′ ��

= ��

� + [Δ� Δ�]

�

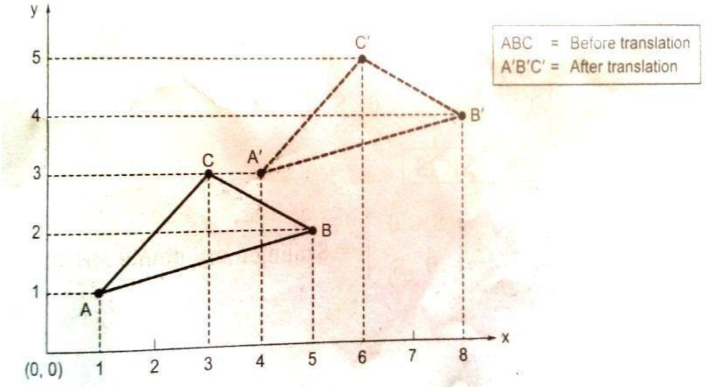
��′ ��′

��

�

�

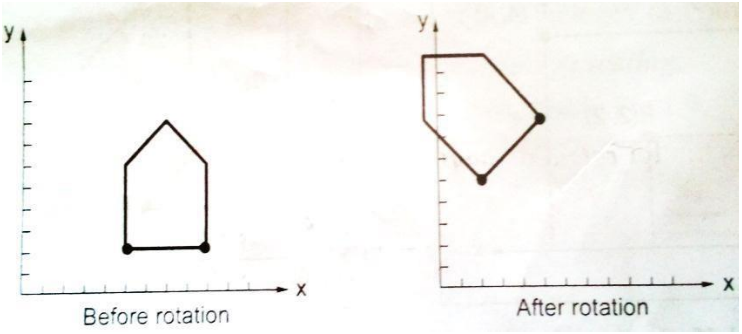
$$\begin{bmatrix} \text{��}' & \text{��}' \\ \text{��}' & \text{��}' \\ \text{��}' & \text{��}' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 5 & 2 \\ 3 & 3 \end{bmatrix} + [3 \;\; 2] = \begin{bmatrix} 4 & 3 \\ 8 & 4 \\ 6 & 5 \end{bmatrix}$$



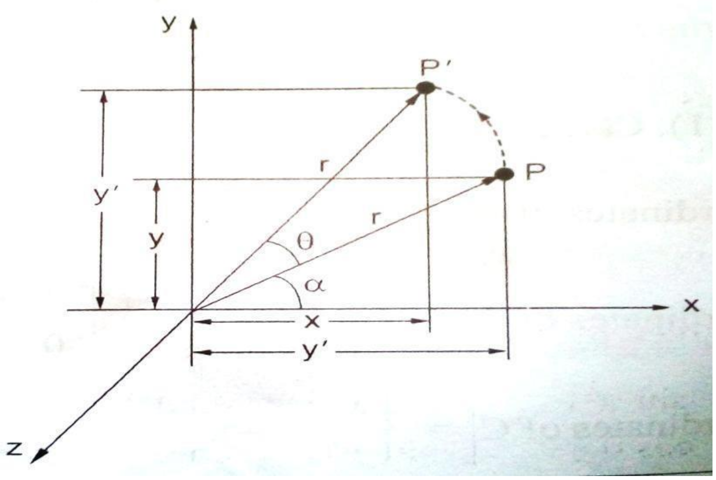
The new coordinates of the translated triangle are A’(4,3), B’(8,4) and C’(6,5).

2D Rotation

- ➢ Rotation refers to the movement an object in such a way that the distance between a certain fixed point and any given point of that body remains constant.
- ➢ Rotation transformation techniques is commonly used in rendering and animation tehniques.



- ➢ Consider the rotation of a point from initial position a P(x,y) to a new position P’(x’,y’) by an angle θ about the origin
- ➢ Here counterclockwise rotation is consider as positive and clockwise rotation as negative.



Mathematically, the coordinates of P’ are given by

X’ = r cos(α+θ) =r cosαcosθ –r sinαsin θ

Y’ = r sin(α+θ) = r sin αcosθ + r cosα sinθ

Substituting x = r cosα and y = r sinα into the equations gives

X’ = x cosθ – y sin θ

Y’ = x cosθ + y sinθ

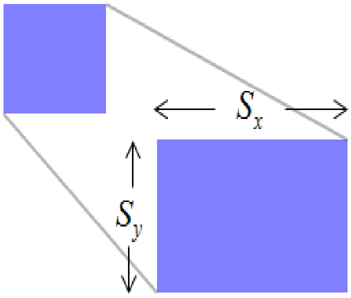
Rotation in angle □□ about a

pivot (rotation) point □□x_r , y_r

□.

2D Scaling

- Scaling is the transformation used to change, increase or decrease, the size of an object.
- Scaling can be achieved by multiplying the original coordinates of an object by the scaling factor Sxalong x-direction and Sy along y-direction.
- Scaling factor is always positive, if scaling factor is less than 1, the object is compressed; if more than 1, the object is stretched.
- If scale factors are equal i.e., Sx=Sy=S, the object changes in size only and not in shape. This scaling is known as **uniform scaling**.
- If scale factors are different i.e., Sx≠Sy, the object changes in size only and not in shape. This scaling is known as **non-uniform scaling**.



x□□□□x y□□□□y□□s_y

□□s_x ,

□□ □s 0 □□□□x□

x□ x

□

□ □□□□□ □□□□□ □

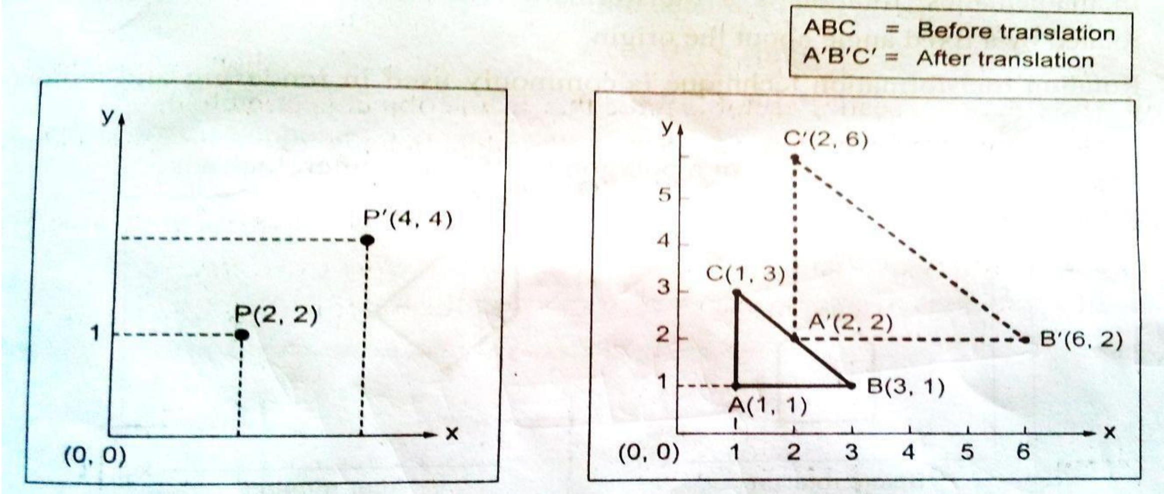
□□y□□ s_y □□□□y□

□

□0

P□□□□S□□P

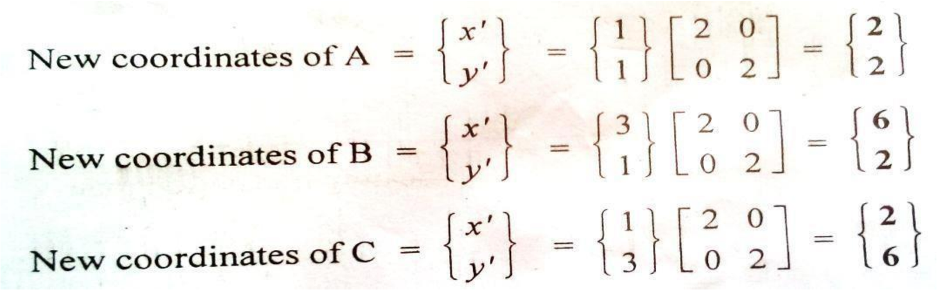
Scaling about a fixed point □x_f , y_f



Example:

If a triangle A(1,1), B(3,1), C(1,3) is scaled by a factor 2, find the new coordinates of the triangle.

Given: A(1,1), B(3,1), C(1,3); uniform scale factor =2, the effect of scaling is represented graphically in Fig.



$$\text{New coordinates of A} = \begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{Bmatrix} 2 \\ 2 \end{Bmatrix}$$

$$\text{New coordinates of B} = \begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{Bmatrix} 3 \\ 1 \end{Bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{Bmatrix} 6 \\ 2 \end{Bmatrix}$$

$$\text{New coordinates of C} = \begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{Bmatrix} 1 \\ 3 \end{Bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{Bmatrix} 2 \\ 6 \end{Bmatrix}$$

2D reflection

- Reflection is a transformation in which the direction of one axis is reversed.
- Reflection transformation produces a mirror image of an object.
- The reflection transformation is useful in the construction of symmetric objects. If the object is symmetric with respect to plane, only half of the geometry is created and then the half model is copied by reflection to develop the full model.

2D Translation
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \mathbf{P}' = \mathbf{T}(t_x, t_y) \cdot \mathbf{P}$$

2D Rotation
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \mathbf{P}' = \mathbf{R}(\theta) \cdot \mathbf{P}$$

Composite translations:
$$\begin{bmatrix} 1 & 0 & t_{2x} \\ 0 & 1 & t_{2y} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_{1x} \\ 0 & 1 & t_{1y} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_{1x} + t_{2x} \\ 0 & 1 & t_{1y} + t_{2y} \\ 0 & 0 & 1 \end{bmatrix}$$

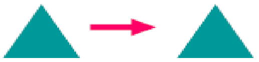
$$\mathbf{T}(t_{2x}, t_{2y}) \cdot \mathbf{T}(t_{1x}, t_{1y}) = \mathbf{T}(t_{1x} + t_{2x}, t_{1y} + t_{2y})$$

In scaling transformation, the original coordinates of an object are multiplied by the given scale factor. 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.

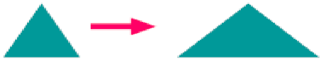
Let us consider some typical cases

- Case 1: a=d=1 and b=c=0 – No Change (identity)
- Case 2: d=1, b=c=0 – Scaling in x coordinate
- Case 3: b=c=0 – Scaling in both x and y coordinates

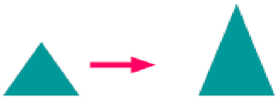
$$[x^* \quad y^*] = [x \quad y]\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [x \quad y]$$



$$[x^* \quad y^*] = [x \quad y]\begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix} = [ax \quad y]$$



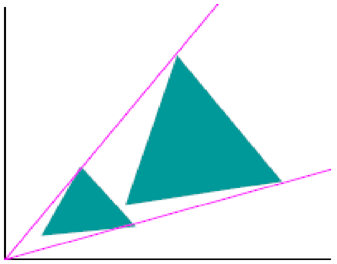
$$[x^* \quad y^*] = [x \quad y]\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} = [ax \quad by]$$



- Case 4; a=d =|s|>1 – Enlargement of the original entity

- Case 5: 0<a=d=|s|<1 – Compression of the entity

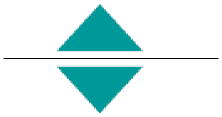
$$[x^* \quad y^*] = [x \quad y]\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} = [ax \quad by]$$



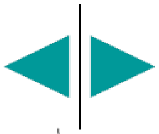
- Case 6: b=c=0, a=1,d=-1 – Reflection about x- axis

- Case 7: b=c=0, a=-1,d=1 – Reflection about y- axis
- Case 8: b=c=0, a=d<0 – Reflection about the origin

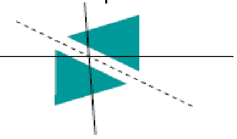
$$[x* \quad y*] = [x \quad y]\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = [x \quad -y]$$



$$[x* \quad y*] = [x \quad y]\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = [-x \quad y]$$



$$[x* \quad y*] = [x \quad y]\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = [-x \quad -y]$$



Homogeneous Coordinates

Each of the above transformations with the exception of translation can be represented as a row vector X, Y and a 2 X 2 matrix. However, 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].

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

$$[X1 \; Y1 \; 1] = [X \; Y \; 1]\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

X-shear

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

Y-shear

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

After multiplying this vectorby 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 three dimensional representation of a two dimensional plane is called homogeneous co- ordinates and the transformation using the homogeneous co-ordinates is called homogeneous transformation.

12. Explain the concept of 3D transformation. (JAN 2014)

- ➢ A three-dimensional object has a three-dimensional geometry, and therefore, it requires a three-dimensional coordinate transformation.
- ➢ A right handed coordinate system is used to carry out a 3-D transformation.
- ➢ The scaling and translation transformations are essentially the same as two-dimensional transformations.
- ➢ However, the points matrix will have a non-zero 3rd column. Additionally, the transformation matrices contain some non-zero values in the third row and third column.

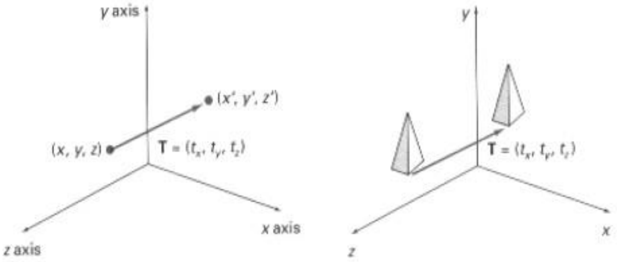
Translation

In three-dimensional homogeneous coordinate representation, a point is transformed from position P(x,y,z) to P'= (x',y',z') this can be written as:

$x' = x+ t_X$ y'

$= y+ ty$

$z' = z+ t_Z$



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotation **P’=T.P**

- *Z-axis Rotation*

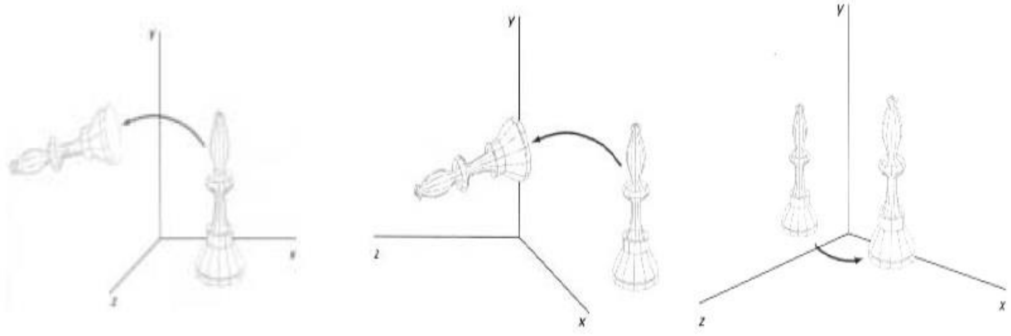
$$\boxed{\begin{aligned} x' &= xCos\theta - ySin\theta \\ y' &= xSin\theta + yCos\theta \\ z' &= z \end{aligned}} \Longleftrightarrow \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} Cos\theta & -Sin\theta & 0 & 0 \\ Sin\theta & Cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- *X-axis Rotation*

$$\boxed{\begin{aligned} y' &= yCos\theta - zSin\theta \\ z' &= xSos\theta + zCin\theta \\ x' &= x \end{aligned}} \Longleftrightarrow \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & Cos\theta & -Sin\theta & 0 \\ 0 & Sin\theta & Cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- *Y-axis Rotation*

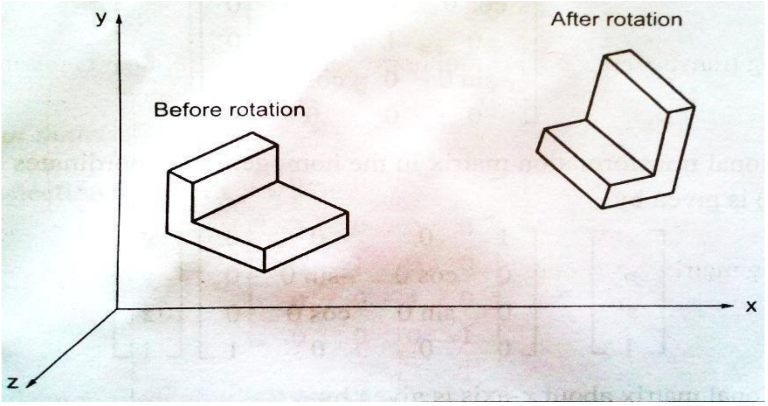
$$\boxed{\begin{aligned} z' &= zCos\theta - xSin\theta \\ x' &= zSin\theta + xCos\theta \\ y' &= y \end{aligned}} \Longleftrightarrow \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} Cos\theta & 0 & Sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -Sin\theta & 0 & Cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



Parallel to one of the Coordinate Axis

➢ In special cases where an object is to be rotated about an axis that is parallel to one of the coordinate axis, we can obtain the desired rotation with the following transformation sequence.

1. Translate the object so that the rotation axis coincides with the parallel coordinate axis (for simplicity, let us take x-axis).
2. Perform the specified rotation about that axis.
3. Translate the object so that the rotation axis is moved back to its original position.

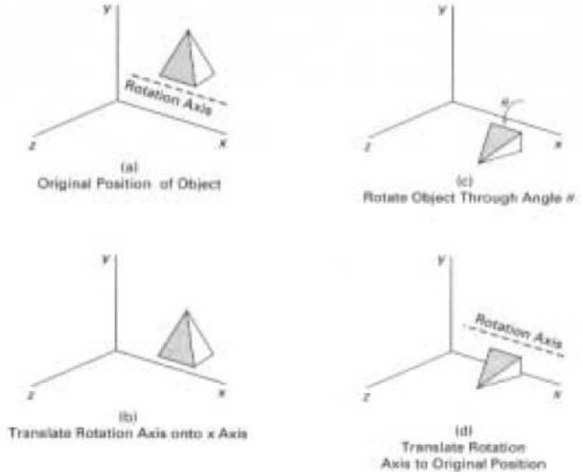


Thus any coordinate position P on the object is transformed sequence as follows:

$$\boxed{P' = T^{-1}.R_x(\theta).T.P}$$

where the composite matrix for the transformation is:

$$\boxed{R(\theta) = T^{-1}.R_x(\theta).T}$$



(a) Original Position of Object

(c) Rotate Object Through Angle θ

(b) Translate Rotation Axis onto x Axis

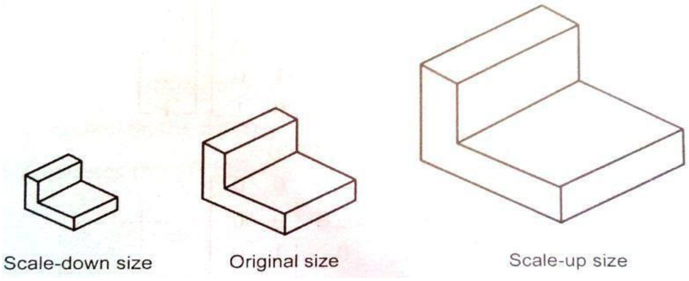
(d) Translate Rotation Axis to Original Position

Scaling

The matrix expression for the scaling transformation of a position P = (x, y, z) relative to coordinate origin can be written

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

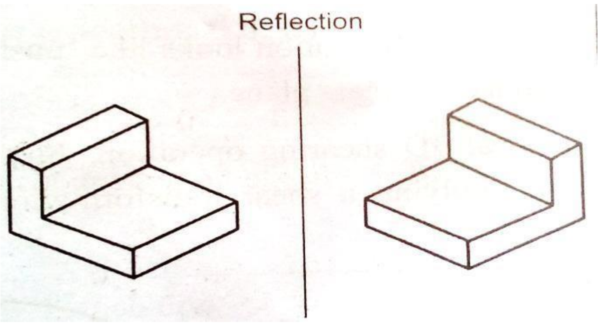
OR $$P' = S.P$$



The matrix representation for an arbitrary fixed-point (xf, yf, zf) can be expressed as:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & (1-s_x)x_f \\ 0 & s_y & 0 & (1-s_y)y_f \\ 0 & 0 & s_z & (1-s_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Reflection



- ➢ The matrix expression for the reflection transformation of a position P = (x, y, z) relative to x-y plane can be written as:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- ➢ Transformation matrices for inverting x and y values are defined similarly, as reflections relative to yzplane and xzplane, respectively.

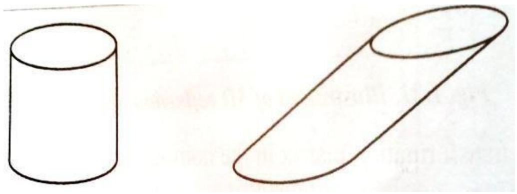
Shears:

The matrix expression for the shearing transformation of a position P = (x, y, z), t produce z-axis shear, can be written as:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Parameters a and b can be assigned any real values. The effect of this transformation

  is to alter x- and y- coordinate values by an amount that is proportional to the z value, while leaving the z coordinate unchanged.



- Shearing transformations for the x axis and y axis are defined similarly.

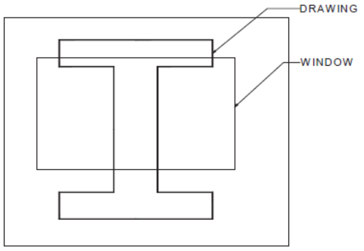
13. Write short notes on clipping, view ports, line drawing. (JAN 2014 & NOV 2016)

(Or)

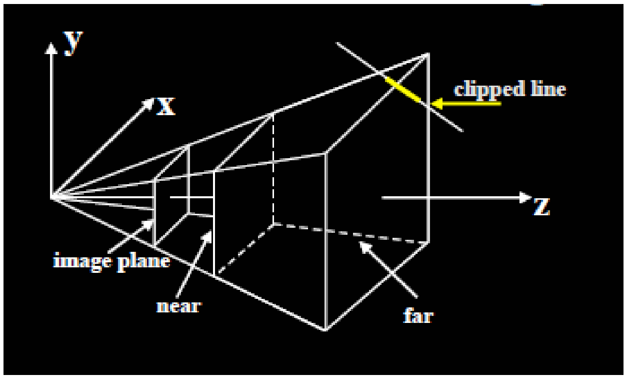
Explain the Cohen - Sutherland line-clipping approach with proper sketches. (M.E. JAN2010 & NOV 2016)

Clipping

- ➢ Clipping is the process of determining the visible portions of a drawing lying within a window. 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. If the element of a drawing crosses the boundary the point of inter-section is determined and only portions which lie inside are drawn. Readers are advised to refer to books on computer graphics for typical clipping algorithms like Cohen-Sutherland clipping algorithm. Fig shows an example of clipping.



- ➢ There is something missing between projections and viewing.
- ➢ Before projecting, we need to eliminate the portion of scene that is outside the viewing frustum

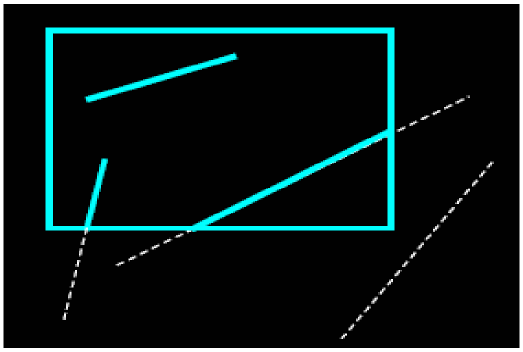


Modify end points of lines to lie in rectangle

Method: Is end-point inside the clip region - half-plane tests

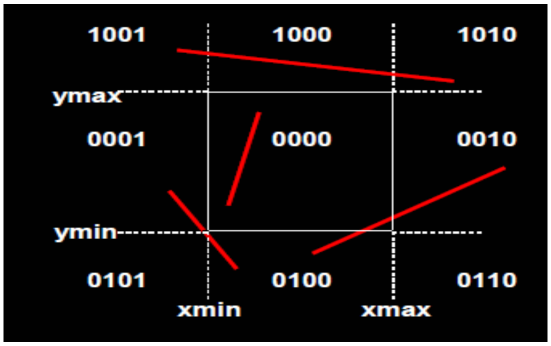
–If outside, calculate intersection between the line and the clipping rectangle and make this the new end point

- Both endpoints inside: trivial accept
- One inside: find intersection and clip
- Both outside: either clip or reject (tricky case)



Cohen-Sutherland Algorithm

• Uses out codes to encode the half-plane tests results



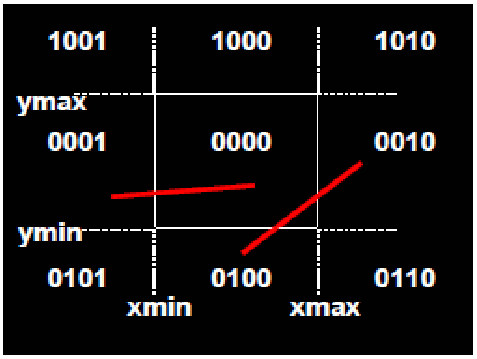
Rules:

– Trivial accept: out code (end1) and out code (end2) both zero

– Trivial reject: out code (end1) & (bitwise and) out code (end2)

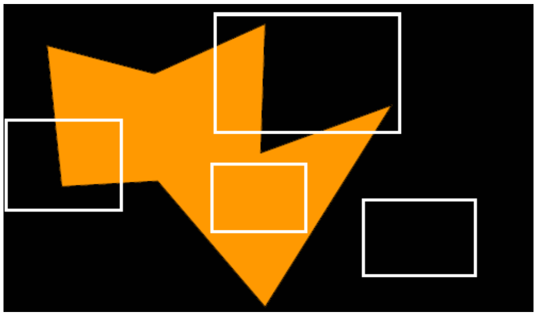
Nonzero Else subdivide

If neither trivial accept nor reject:
–Pick an outside endpoint (with nonzero out code)
–Pick an edge that is crossed (nonzero bit of out code)
–Find line's intersection with that edge
–Replace outside endpoint with intersection point
–Repeat until trivial accept or reject



Polygon clipping

Convert a polygon into one or more polygons that form the intersection of the original with the clip window.



Sutherland-Hodgman

Polygon Clipping Algorithm

10. Sub problem:

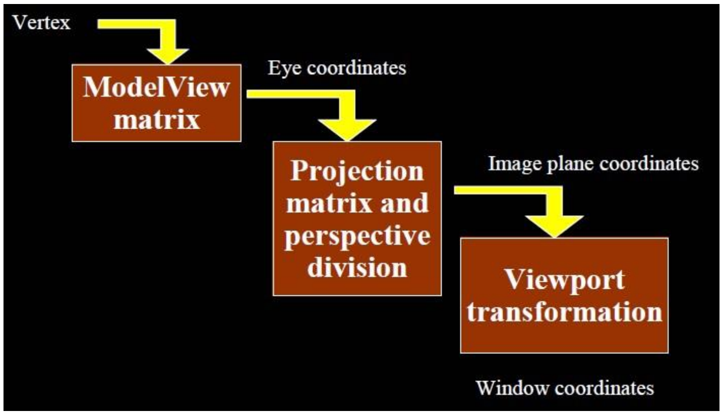
Clip a polygon (vertex list) against a single clip plane

Output the vertex list(s) for the resulting clipped polygon(s)

• Clip against all four planes
–generalizes to 3D (6 planes)
–generalizes to any convex clip polygon/polyhedron
To clip vertex list against one half-plane:
• if first vertex is inside - output it
• loop through list testing inside/outside transition - output depends on transition:
in-to-in: output vertex

out-to-in: output intersection and vertex out-to-
out: no output
in-to-out: output intersection

View port



Drawing of Lines

BRESENHAM'S LINE ALGORITHM

- Bresenham's algorithm enables the selection of optimum raster locations to represent a straight line. In this algorithm either pixels along X or Y directions are incremented by one unit depending upon the slope of the line.
- The increment in the other direction is determined by examining the error or distance between actual line location and the nearest grid locations.
- The principle of Bresenham's algorithm can be explained with the aid of Fig a. If the slope of the line (in the first octant) is more than 1/2, the pixel point in the Y direction is shifted by one. Thus lines L1 and L2 passes through pixel (0, 0). For line L2 slope is greater than 1/2; hence the pixel point is (1, 1) whereas for L1 the slope is less than 1/2 and hence (1, 0) is the pixel point.
- Bresenham's algorithm selects optimum raster locations with minimum computation. To accomplish this, the 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. Only the sign of this error needs be examined.
- Consider the line of slope m = 0.4 and passing through (0, 0) in Fig (a). The error team e is initialized to –1/2. The next raster point can be determined by adding the slope (m) to the error term.

i.e. $e = e + m$

$e = -0.5 + 0.4 = -0.1$

- Since e is negative, the line will pass below the middle of the pixel. Hence the pixel is at the same horizontal level i.e., (0, 1). For the next location X is incremented to 2. Error e = -0.1 + 0.4 = 0.3. Since e is positive the line passes above the middle of the raster line. Hence the location of pixel is (2, 1).
- Before examining the next pixel location the error term has to be re-initialized as its value is positive. Re-initialization is done by subtracting one from the current e value. Hence e = 0.3 - 1 = -0.7. Adding the slope 0.4 we get e = -0.3. Table 3.2 shows the computed values and the location of pixels. A plot of the pixel location is shown in Fig. (b).

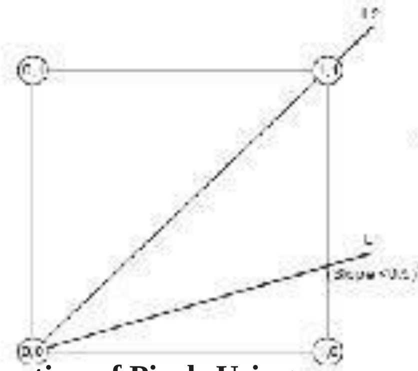


Fig. a Location of Pixels Using Bresenham Algorithm

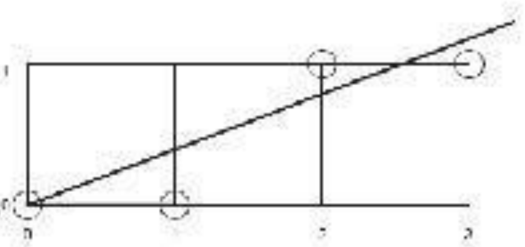


Fig. b Pixels for Line of Slope, m = 0.4

- The speed of the Bresenham's algorithm can be increased by using integer arithmetic and eliminating division to determine slope.The pseudo code and a C-program for implementing Bresenham'

salgorithm are given below:

Pseudo code for Bresenham's line-drawing algorithm

Given a line from x1, y1 to x2, y2...

```
dx is the difference between the x components of end
points dy is the difference between the y components of
end points ix is the absolute value of
dx iy is the absolute value
of dyinc is the larger of dx,
dyplotx is x1
ploty is y1 (the beginning of line)
x starts at 0
y starts at 0
```

plot a pixel at plotx, ploty increment x using ix increment y using iy plot is false if x is greater than inc plot is true decrement x using inc increment plotx if dx is positive decrement plotx id dx is negative if y is greater than inc plot is true decrement y using inc incrementploty if dy is positive decrement ploty if dy is negative if plot is true, plot a pixel at plotx, ploty increment

Program in Turbo-C to draw a line

```
# include<stdio. h>
# include<graphics. h>
# include<stdlb. h>
void draw line (int x1, int y1, int x2, int y2), void
main (void)

{
draw line (100, 100, 50, 50) ;
}
void draw line (int x1, int y1, int x2 m int y2)
{
        int dx, dy, inc, ix, iy, x, y, plot, plotx, ploty, i ;
        intgd,  gm ;
        gd = DETECT ;
        initgraph (&gd, &gm, " ") ; dx
        =  x1 – x2 ;
        dy = y1 – y2 ;
        ix =  abs  (dx)  ;  iy
        = abs (dy) ;
        inc = max (ix, iy) ;
        x = y = 0 ;
        plot x = x1 ; plot y
        = y1 ;
        for (i = 0 ; i <inc ; i ++)
{
        x + = ix ; y + iy
        ; plot = 0
        if (x
{       >inc)

        plot = 1 ;
        x – = inc
        ; if (dx
        < 0)
else    plot x – = 1 ;

}       plotx + = 1 ;

{               if (y

                >inc)

                plot = 1
```

```
                  ; y – =
                  inc ;
                  if
                  (dy)
            else  ploty – = 1 ;

      }           ploty + = 1

                  ; if (plot)
            else  putpixel (plotx, ploty, 1)
            }
            getch ( ) ;
            closegraph ( )
            ;
```

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. One method of generating a line segment is a symmetrical digital differential analyzer (DDA).

UNIT II

GEOMETRIC MODELING

TWO MARKS

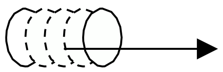
1. **What is meant by Wire frame model? (MAY 2016)**

Wire frame model:

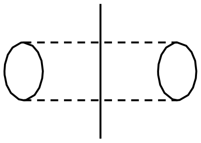
- A wireframe model consists of points and curves only, and looks as if it’s made up with a bunch of wires. Joining points and curves creates wire frame models. These models can be ambiguous and unable to provide mass property calculations, hidden surface removal, or generation of shaded images. Wireframe models are mainly used for a quick verification of design ideas. This is the simplest CAD model of an object.
- Advantages of this type of model include ease of creation and low level hardware and software requirements. Additionally, the data storage requirement is low.
- The main disadvantage of a wireframe model is that it can be very confusing to visualize. For example, a blind hole in a box may look like a solid cylinder, as shown in the figure.

2. **Discuss about surface model**

Sweeping a curve around or along an axis can create a surface model. The figures below show two instances of generating a surface model.



Generating a cylinder by sweeping a circle
circle in the direction of an axis



generating a donut by sweeping a
around an axis

3. Describe the advantages and limitations of Surface Model

Surface Model is useful for representing surfaces such as a soft-drink bottle, automobile fender, aircraft wing, and in general, any complicated curved surface. One of the limitations of a surface model is that there is no geometric definition of points that lie inside or outside the surface.

4. Discuss about Solid models.

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

Solid models are the most preferred form of CAD models and represent unambiguous image of a component. A solid model can be used to analyze the moment of inertia, mass, volume, sections of the model, etc.

5. List out some of the Application of Solid Models

Application of Solid Models

In mechanical engineering, a solid model is used for the following applications:

1. **Graphics**: generating drawings, surface and solid models
2. **Design**: Mass property calculation, interference analysis, finite element modeling, kinematics and mechanism analysis, animation, etc.
3. **Manufacturing**: Tool path generation and verification, process planning, dimension inspection, tolerance and surface finish.
4. **Component Assembly**: Application to robotics and flexible manufacturing: Assembly planning, vision algorithm, kinematics and dynamics driven by solid models.

6. What are the different forms of Solid Model Representation?

There are three different forms in which a solid model can be represented in CAD:

- Wire frame Model
- Surface Model
- Solid Model

7. What are the uses of Parametric Equations?

Parametric equations describe the dependent and independent variables in terms of a parameter. The equation can be converted to a non-parametric form, by eliminating the dependent and independent variables from the equation. Parametric equations allow great versatility in constructing space curves that are multi-valued and easily manipulated.

8. **Define Hermite cubic curve.**

Hermite cubic curve is also known as parametric cubic curve, and cubic spline. This curve is used to interpolate given data points that result in a synthetic curve, but not a free form, unlike the Bezier and B-spline curves. The most commonly used cubic spline is a three-dimensional planar curve (not twisted). The curve is defined by two data points that lie at the beginning and at the end of the curve, along with the slopes at these points. When two end points and their slopes define a curve, the curve is called a Hermite cubic curve.

9. **Cubic splines are not as popular as the Bezier and B-spline curves, why?**

In design applications, cubic splines are not as popular as the Bezier and B-spline curves. There are two reasons for this:

- The curve cannot be modified locally, i.e., when a data point is moved, the entire curve is affected, resulting ina global control, as shown in the figure.
- The order of the curve is always constant (cubic), regardless of the number of data points. Increase in the number of data points increases shapeflexibility, however, this requires more data points, creating more splines that are joined together (only two data points and slopes are utilized for each spline).

Effect of Moving the Data Point

Effect of Change in slope

10. **Write on Blending Two or More Bezier Curves.**

Two or more Bezier curves can be blended to provide a desired curve of a complex nature. When joining curves, slope continuity is maintained by having three collinear points, the middle one being common to the adjoining curves, as shown.

Point V3is the middle point of the common points V2,, V3,,and V4 of curves A and B.

11. Describes about curve entities with examples

All existing CAD/CAM systems provide users with curve entities, which can be divided into analytical and synthetic entities. Analytical entities are points, lines, arcs and circles, fillets and chambers, and conics (ellipses, parabolas, and hyperbolas). Synthetic entities include various types of spline (cubic spline and B-spline) and Bezier curves.

12. List the methods of defining lines

- Points
- Horizontal or vertical
- Parallel or perpendicular to an existing line
- Tangent to existing entities

13. List the methods of defining circles

- Radius
- Three points
- Center and a point on the circle
- Tangent to a line, pass through a given, and with given radius.

14. List the methods of defining synthetic curves

- A cubic spline
- Bezier curve
- B-spline curves: curve interpolates data points
- B-spline curves: curve extrapolates data points

15. Defines analytical curves. (NOV 2015)

Analytical curves are defines as those that can described by analytical equations such as lines, circles, and conics.

16. Defines Synthetic curves

Synthetic curves are the ones that are described by sets of data points such as splines and Bezier curves.

17. Discuss on the importance of order of continuity.

The order of continuity becomes important when a complex curves is modeled by several curve segments pieced together end-to-end. Zero-order continuity (C0) yields a position-continuous curve. First (C1) and second (C2) order continuity imply slope- and curvature-continuous curves, respectively.

18. State the Conditions satisfies by solid models

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.
3. **Finite**: The solid must be finite in size.

19. List out the Solid Model Creation Scheme

A solid model can be generated by the following schemes.

1. Constructive Solid Geometry (CSG)
2. Boundary Representation (B-Rep)
3. Sweeping

20. Define CSG.

Constructive Solid Geometry Scheme

This scheme is based on the principle that two primitives can be combined to produce a new solid model. 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.

21. What are steps involved in generating a solid model?

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.

22. Define Boundary Representation (B-Rep) Scheme.

This scheme is based on the concept that a physical object is bounded by a set of faces. A solid model is created by combining faces and contains vertices, edges, loops, and bodies. 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.

23. What is meant by 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.

24. Write down the Euler's equation for valid polyhedral

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

25. Define interpolated curve? (NOV 2015)

Cubic Spline curves pass through all the data points and therefore they can be called as interpolated curves. Bezier and B-Spline curves do not pass through all the data points, instead, they pass through the vicinity of these data points. Both the cubic spline and Bezier curve have first-order continuity, whereas, B-Spline curves have a second-order continuity.

26. Define Plane Surface

This is the simplest surface, requires 3 non-coincidental points to define an infinite plane. The plane surface can be used to generate cross sectional views by intersecting a surface or solid model with it.

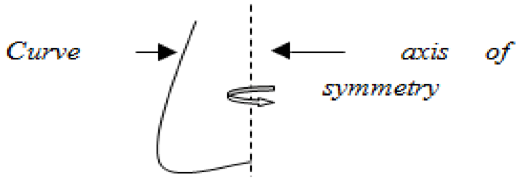
27. What is ruled (lofted) Surface?

This is a linear surface. It interpolates linearly between two boundary curves that define the surface. Boundary curves can be any wire frame entity. The surface is ideal to represent surfaces that do not have any twists or kinks.

Boundary Curve

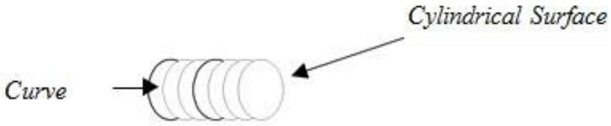
28. What is meant by Surface of Revolution?

This is an axisymmetric surface that can model axisymmetric objects. It is generated by rotating a planar wire frame entity in space about the axis of symmetry of a given angle.



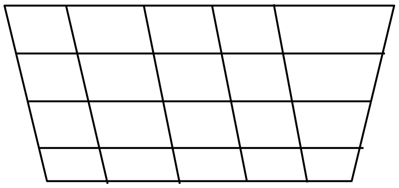
29. What is meant by Tabulated Surface?

This is a surface generated by translating a planar curve a given distance along a specified direction. The plane of the curve is perpendicular to the axis of the generated cylinder.



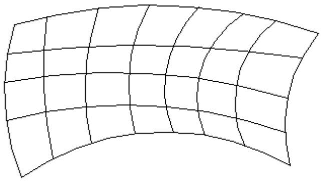
30. How the Bi-linear Surface is generated?

This 3-D surface is generated by interpolation of 4 endpoints. Bi-linear surfaces are very useful in finite element analysis. A mechanical structure is discretized into elements, which are generated by interpolating 4 node points to forma 2-D solid element.



31. What is Coons Patch?

Coons patch or surface is generated by the interpolation of 4 edge curves as shown



32. What is meant by Bezier Surface?

This is a synthetic surface similar to the Bezier curve and is obtained by transformation of a Bezier curve. It permits twists and a kink in the surface.The surface does not pass through all the data points. A tensor product Bezier surface is an extension of the Bezier curve in two parametric directions *u* and *v*.

33. What is meant by B-Spline Surface?

Thisisasyntheticsurfaceanddoesnotpassthroughalldatapoints.Thesurfaceis capable of giving very smooth contours, and can be reshaped with local controls.

34. Mention some of the application of Bilinear Surfaces

Bilinear patches are extensively used in 2-D finite element analysis (FEA). In FEA, an engineering structure is defined by several bilinear surfaces (elements), which are created by joining points on the structure’s geometry, called nodes. The nodes are connected to other nodes to create quadrilateral surfaces. Points not lying on the nodes are calculated by interpolation. Thus, the entire structure is completely defined by the nodes and the bilinear surfaces.

35. List out some of the drawbacks of Bilinear Surfaces

Bilinear surfaces have a very limited use, mainly for FEA. Since only 4points can be used in the interpolation, the smoothness of the generated surface is limited. Additionally, there is no flexibility to control shapes of the surface, unlike the sweeped surfaces.

36. Define complete solid.

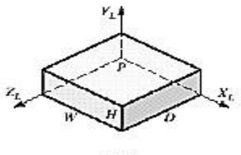
A complete solid is one which enables a point in space to be classified relative to the object, if it is inside, outside, or on the object. This is sometimes called spatial addressability.

37. Define Geometry and Tolerance.

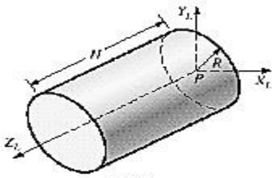
Geometry sometimes called metric information is the actual dimensions that define the entities of the object.

Topology sometimes called combinational structure is the connectivity and associatively of the object entities.

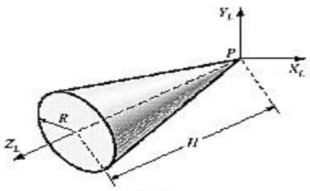
38. Sketch some common primitives



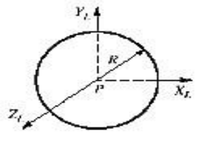
Block



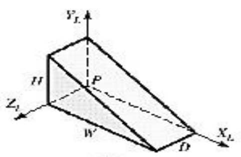
Cylinder



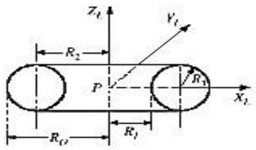
Cone



Sphere



Wedge



Torus

39. Define closed surface.

A closed surface is one that iis continous without breaks.An orientable surface is one where it is possible to distinguish two sides by using the direction of the surface normalto point inside or outside the solid model. Each face is bounded by edges, and each edge is bounded by vertices.

40. Define Feature.

A feature is a basic building block that describes the design, like a keyway on a shaft. Each feature indicates how to add material (like a rib) or remove a portion of material (like a cut or a hole). Features adjust automatically to changes in the design thereby allowing the capture of design intent. This also saves time when design changes are made. Because features have the ability to intelligently reference other features, the changes made will navigate through design, updating the 3-D model in all affected areas.

41. What are the types of features available in solid modeling software?

In typical solid modeling software the designer can create a feature in two basic ways. One is to sketch a section of the shape to be added and then extrude, revolve, or sweep it to create the shape. These are called sketched features. Another type of feature is the pick-and-place feature. Here the designer simply performs an engineering operation such as placing a hole, chamfering or rounding a set of edges, or shelling out the model.

42. What is generic form representing curves and surface?

The generic form in which any generic point (x, y, z) satisfies a relationship in implicit form in x, y, and z i.e. f(x, y, z) = 0. A single such constraint generally describes a surface while two constraints considered together can be thought of as a curve which is the intersection of two surfaces. This may be expressed in an explicit form in the following manner: x = g1(y, z)**,** y = g2(x, z)**,** z = g3(x, y)

43. What is parametric form representing curves and surface?

A parametric curve is one whose defining equations are in terms of a simple common independent variable known as parametric variable. In the parametric form, the representation is done by a set of functions. A curve may be represented by

x = x (u)
y = y (u)
z = z (u)

where x, y, z are co-ordinates of the points on the curve which are functions of some parameter u and the parametric variable is constrained in the interval. For example, a point (x, y) is located at an angle θ from +X axis on a circle with centre at (0, 0) and radius= 1 can be described in parametric form as:

x = Cos θ,

y = Sin θ
where θ is the parameter. Surfaces are described similarly for which x, y and z are functions two independent parameters u and v.

44. Why parametric form more popular than generic form?

- More degrees of freedom/control
- Parametric equations can be transformed directly
- Infinite slopes can be handled without computational breakdown
- Easy to express as vectors
- Amenable to plotting and digitizing
- Inherently bounded
- Separation of variables
- Each variable is treated alike
-

45. Modeling of curves must satisfy?

The modeling of curves is done to satisfy the following:

i. Axis independence,
ii. Global and local control,
iii. Smoothness of curves
iv. Versatility and
iv. Continuity between adjoining segments of curve.

46. Mention the advantages of Bezier curve over cubic spline

The advantages of Bezier curve over cubic spline is that the direction of the curve at the joints can be defined and changed simply by specifying the position of the second and third data points. Changing a control point not only affects the shape of the curve near the control point but has an influence throughout the curve. This lack of local control is a major weakness of Bezier curve.

47. Define Curved Solids.

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

48. Discuss the primitives used in creating a solid.

•**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.

49. Discuss about 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.

50. Define NURBS.

A non-uniform a -spline curve is defined on a knot vector where the interior knot spans are not equal. A rational a-spline is defined by a set of four dimensional control points.

51. What is B-splines

B- splines are obtained from β-splines by introducing two new degrees of freedom: bias and tension. These can be applied uniformly or non-uniformly.

52. Discuss about Representation of Surfaces

A surface can be defined as the locus of points which satisfy a constraint equation in the form of F(X, Y, Z) = 0. In parametric form a surface may be represented as

$x = X(u, v)$
$y = Y(u, v)$
$z = Z(u, v)$

where X, Y and Z are suitable functions of two parameters u and v. For example, the parametric representation of the surface of a sphere whose center is at the origin of coordinates and of radius R is

$x = X(\theta, \varphi) = R \sin \varphi \cos \varphi$
$y = Y(\theta, \varphi) = R \sin \varphi \sin \varphi$
$z = Z(\theta, \varphi) = R \cos \varphi$

53. Explain about the design of surfaces

The design of surfaces may be based on quadrics like ellipsoid, hyperboloid, cone, hyperbolic cylinder, parabolic cylinder, elliptic cylinder and elliptic parabolic. A surface may be generated by sweeping a pattern curve along a spline curve. The swept surface may also be linear, conical linear or circular swept surface.

54. What are the characteristics of B-spline surface?

A rectangular set of data or control points creates the surface. This set forms the vertices of the characteristic polyhedron that approximates and controls the shape of the resulting surface. The degree of the surface is independent of the number of control points and continuity is automatically maintained throughout the surface by virtue of the form of blending functions.

55. What is surface trimming?

Trimming of surface is segmentation or intersection problem. It is the process of reduction in area of an entity with respect to the other entity. If the surface is to be trimmed between two point trimming boundaries the problem is a segmentation problem.

56. What do mean by intersection in surface?

The intersection problem involving surfaces is complex and non-linear in nature. It depends on whether it is a surface/curve or surface/surface interaction as well as on the representation, parametric or implicit of the surface or curves.

57. What do you mean by projection of surface?

Projecting an entity onto a plane or a surface is useful in applications such as determining shadows or finding the position of the entity relative to the plane or the surface.

58. Differentiate Geometry and topology.

GEOMETRY	TOPOLOGY
1.It is the actual dimensions that defines the entities of the object	1.It is the connectivity and associatively of the object entities
2.It is the visible geometry and graphical information at user point of view	2.It is the non-graphical relational information stored in the solid model database in user point of view
3.It is also called as metric information	3.It is also called as combinatorial structure

59. List a few commonly used solid primitives.

The most commonly used solid primitives are

a) Block b) Cylinder c) Cone d) Sphere e) Wedge f) Torus

60. What are Boolean operations? List the Boolean operations

Boolean operation is the process of getting a required part from two or more primary parts. The process are 1) Addition (+) 2) Subtraction (-) 3) Union (u) 4) Intersection (n).

61. What is Rigidity?

Rigidity implies that the shape of a solid model is invariant and does not depend on the model location or orientation in space.

62. Define domain in representation schemes.

The domain of a representation scheme is the class of objects that the scheme can represent or it is the geometric coverage of the scheme.

63. How Validity checks are achieved in representation schemes?

The validity checks are achieved in three ways: a) Test the resulting database via given algorithm b) Build checks into the scheme generator or design scheme elements c) Manipulate via given syntax.

64. List out the formal properties of representation scheme.

The formal properties are 1) Domain 2) Validity 3) Completeness or unambiguousness 4) Uniqueness

65. Enumerate the various polyhedral objects.

a) Simple polyhedral b) polyhedral with faces of inner loops c) polyhedral with not through holes d) polyhedral with handles (through holes)

66. What are the open polyhedral objects?

a) Wire polyhedral b) shell polyhedral c) Lamina polyhedral d) open three-dimensional polyhedral

67. What are rational curves importances?

Rational curves are defined by algebraic ratio of two polynomials. They draw their theories from projective geometry. Importance is of their invariance under projective transformation that is the perspective image of a rational curve is a rational curve.

68. Distinguish Analytic and Synthetic curve

Analytic Curves	Synthetic Curves
1.The Analytic curves are defines by analytic equations.	1.The Synthetic curves are defines by the set of data points or control points
2.Analytic curves are easy in computation but not flexible for designers	2.Synthetic curves are hard in computation but very flexible to designers in making new models
3.Lines, conics are analytic curves	3.Splines, Bezier curves are synthetic curves

69. What do you mean by Sculptured surface?

Any of the surface patches introduced thus far is seldom enough by itself to model complex surfaces typically encountered in design and manufacturing applications. These complex surfaces are known as sculptured or free-form surfaces.

70. List the surfaces representations

The surfaces are a) Bi cubic surface b) Bezier surface c) B-spline surface d) Coons surface c) Blending offset surface d) Triangular surface e) Sculptured surface f) Rational surface.

71. Enumerate the Surface entities provided in CAD/CAM system.

The major surface entities provided by the CAD/CAM system are

1) Plane surface 2) Ruled (lofted) surface 3) Surface of revolution 4) Tabulated cylinder 5) Bezier surface 6) B-Spline surface 7) Coons patch 8) Fillet surface 9) Offset Surface

72. What is surface segmentation?

Surface segmentation is a reparametrization or parameter transformation of the surface while keeping the degree of its polynomial in *u* and *v* unchanged.

73. Write down the eccentricity value for ellipse, parabola and hyperbola.

The value of eccentricity less than one is ellipses; those with eccentricity equal to one are parabolas, and those with eccentricity greater than one is hyperbolas.

74. Define Conic section.

A conic section is a curve created as the intersection of a cone with a plane. In analytic geometry, a conic may be described as a plane algebraic curve of degree two, and as a quadric of dimension two.

75. Define 'focus' of a curve.

In geometry, the focus is pair of special points with reference to which any of a variety of curves is constructed.

76. Write a short note 'Hermite curve'. (MAY 2016)

A Hermite curve is a spline where every piece is a third degree polynomial defined in Hermite form: that is, by its values and initial derivatives at the end points of the equivalent domain interval.

77. Define Quadratic Bezier curve.(NOV 2016)

As shown in the figure, a quadratic Bezier curve is the path defined by the function B(*t*), given points P0, P1, and P2, .

78. List out the various Bezier curves based on control points.

Linear Bezier curve, Cubic Bezier curve and Quadratic Bezier curve

79. Describe Rational Bezier curve.

The rational Bezier curve includes variable weights (w) to provide closer approximations to arbitrary shapes. For Rational Bezier Curve, the numerator is a weighted Bernstein form Bezier and the denominator is a weighted sum of Bernstein polynomials.

80. Define NURBS.

In computer graphics, a powerful extension of B-splines is non-uniform rational B-splines is NURBS. NURBS are basically B-splines in uniform coordinates. Like B-splines, they are described by their order, and a knot vector, and a set of control points, but unlike B-splines, the control points have a weight.

81. Write down 'Free form surface'.

Freeform surface is used in CAD and other computer graphics software to describe the skin of a 3D geometric element. Freeform surfaces do not have rigid radial dimensions, unlike regular surfaces such as planes, cylinders and conic surfaces.

82. Write down the advantages and limitations of surface modelling.

Advantages of Surface modelling:

1. Surface modelling can be used to check the real look of the product with coloring and shading.
2. Surface modelling can be used to perform interference checking.
3. As the surface models precisely define the part geometry such as surface and boundaries, they can help to produce machine instructions automatically.
4. Complex surface features can be created very easily.
5. Un ambiguities in the interpretation of object is less than wire frame models by using the provision of hidden line removal.

83. Describe the 'Surface patch'.

Surface patch which is used to create a surface using curves that form closed boundaries.

84. List out properties of B-Spline.

i. The maximum order in every parametric direction is limited to the number of describing polygon vertices in that path.

ii. The surface is changeable to an affine transformation.

iii. The deviation diminishing property of B-spline surface is not well recognized.

UNIT – II 16 MARKS

GEOMETRIC MODELING

Syllabus: 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.

1. Describe about the Role of curves in Geometric Modeling

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, curves can be classified as follows:

Curves:

1. Analytical Curves a) Non-Parametric curves –Implicit, Explicit curves b) Parametric curves
2. Synthetic curves a) Hermite curves b) B-Spline c) Bezier curves

- **Analytical Curves**: This type of curve can be represented by a simple mathematicalequation, 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.
- **Interpolated curves**: An interpolated curve is drawn by interpolating the given data pointsand has a fixed form, dictated by the given data points. These curves have some limited flexibility in shape creation, dictated by the data points.
- **Approximated Curves**: These curves provide the most flexibility in drawing curves of verycomplex shapes. The model of a curved automobile fender can be easily created with the help of approximated curves and surfaces.
- 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.

2. Explain how the curves are represented in parametric form.(Nov05, May04)

(Or)

Parametric and Non-parametric Equations of a Curve

- The mathematical representation of a curve can be classified as either parametric or non- parametric (natural). A non-parametric equation has the form,

  $y = c1+ c2x + c3x^2+ c4x^3$ *Explicit non-parametric equation*

- This is an example of an explicit non-parametric curve form. Inthis equation, there is a unique single value of the dependent variable for each value of the independent variable. The implicit non-parametric form of an equation is,

  $(x – x_c)^2+ (y – y_c)^2= r^2$ *Implicit non-parametric equation*

In this equation, no distinction is made between the dependent and the independent variables.

Parametric Equations:

- Parametric equations describe the dependent and independent variables in terms of a parameter. The equation can be converted to a non-parametric form, by eliminating the dependent and independent variables from the equation.
- Parametric equations allow great versatility in constructing space curves that are multi-valued and easily manipulated.
- Parametric curves can be defined in a constrained period ($0\leq t \leq 1$); since curves are usually bounded incomputer graphics, this characteristic is of considerable importance. Therefore, parametric form is the most common form of curve representation in geometric modeling. Examples of parametric and non-parametric equations follow.

Non-Parametric		Parametric
***Circle*:**	$x^2 + y^2 = r^2$	$x = r \cos\theta, y = r \sin\theta$

Where, θ is the parameter.

- CAD programs prefer a parametric equation for generating a curve. Parametric equations are converted into matrix equations – to facilitate a computer solution, and then varying a parameter from0 to 1 creates the points or curves. In this course, we will use the following parameters, with the range indicated,

$$0 \le t \le 1 \qquad 0 \le s \le 1 \qquad 0 \le \theta \le 2\pi s \qquad 0 \le \phi \le 2\pi s$$

Fixed-Form or Analytical Curves

Equation of a Straight Line:

- The simplest fixed-form curve is a straight line. Parametric equation of a straight line is given as,

$$P(t) = A + (B-A)\, t \qquad (2.1)$$

The parametric equation of line AB can be derived as, $x = x_1 + (x_2 - x_1)t$, $y = y_1 + (y_2 - y_1)t$, the point P on the line is sweeped from A to B, as the value of t is varied from0 to 1.

Circular Curve

The non-parametric equation of a circle is,

$$(x - X_c)^2 + (y - y_c)^2 = r^2 \qquad (2.2)$$

Where, x_c,and $y_c$are coordinates of the center, and r is radius of the circle.

- If we were to use this form of the equation for plotting a circle or a circular curve, we will first calculate several values of x and y along the circumference of the circle, and then plot them.
- The curve thus generated will be of a poor quality, unless we plot a very large number of data points, which will result in a significant demand for storage of these data points. Therefore, as stated earlier, in CAD programs, we use a parametric equation, which avoids the need for storage of the data points, and provides a smooth curve.

Parametric equation of the above circle can be written as,

$$x_i = x_c + r \cos\theta$$

$y_i = y_c + r \sin\theta$ (2.3)

This equation is converted into a matrix form so that a computer can solve it.

Simplifying the equation we get,

$$x_{i+1} = x_c + (x_i - x_c)\cos\Delta\theta - (y_i - y_c)\sin\Delta\theta$$

$$y_{i+1} = y_c + (x_i - x_c)\sin\Delta\theta + (y_i - y_c)\cos\Delta\theta \quad (2.4)$$

Even though, equations (2.4) can be used as an iterative formula to plot a circle or a circular curve, using the EXCEL or MATLAB, or any other plot routines. The original CAD programs used iterative formulas to generate curves. The BASIC and FORTRAN languages were used to write the CAD codes.

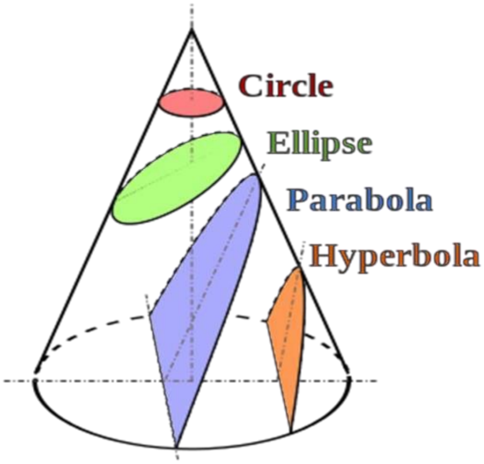
3. Write short notes on Representation of curve.

Representation of curves

- A curve is an entity related to a line but which is not required to be straight.
- A curve is a topological space which is internally homeomorphism to a line; this shows that a curve is a set of points which close to each of its points looks like a line, up to a deformation.
- A conic section is a curve created as the intersection of a cone with a plane. In analytic geometry, a conic may be described as a plane algebraic curve of degree two, and as a quadric of dimension two.
- There are several of added geometric definitions possible.
- One of the most practical, in that it involves only the plane, is that a non-circular conic has those points whose distances to various point, called a 'focus', and several line, called a 'directrix', are in a fixed ratio, called the 'eccentricity'.

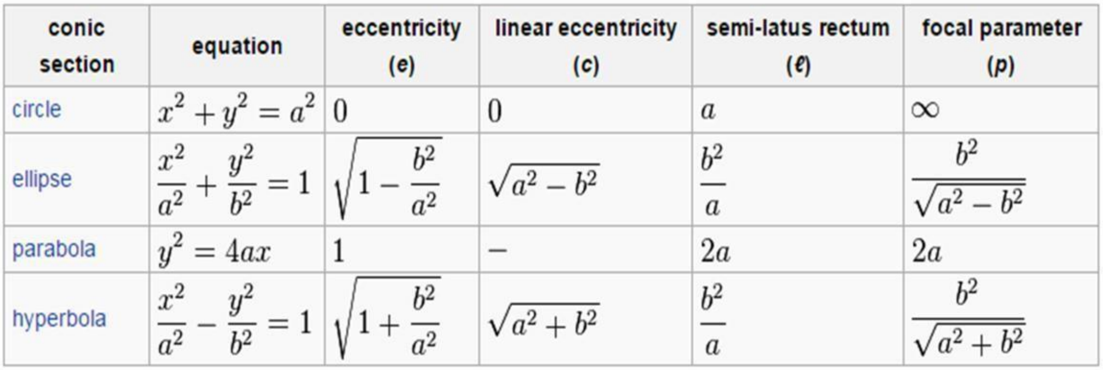
Conic Section

- Conventionally, the three kinds of conic section are the hyperbola, the ellipse and the parabola.



- The circle is a unique case of the ellipse, and is of adequate interest in its own right that it is sometimes described the fourth kind of conic section.
- The method of a conic relates to its ‘eccentricity’, those with eccentricity less than one is ellipses, those with eccentricity equal to one is parabolas, and those with eccentricity greater than one is hyperbolas.
- In the focus, directrix describes a conic the circle is a limiting with eccentricity zero. In modern geometry some degenerate methods, such as the combination of two lines, are integrated as conics as well.
- The three kinds of conic sections are the ellipse, parabola, and hyperbola. The circle can be taken as a fourth kind of ellipse. The circle and the ellipse occur when the intersection of plane and cone is a closed curve.
- The circle is generated when the cutting plane is parallel to the generating of the cone. If the cutting plane is parallel to accurately one generating line of the cone, then the conic is unbounded and is mentioned a parabola. In the other case, the figure is a hyperbola.
- Different factors are connected with a conic section, as shown in the Table 2.1. For the ellipse, the table shows the case of ‘a’ > ‘b’, for which the major axis is horizontal; for the other case, interchange the symbols ‘a’ and ‘b’. For the hyperbola the east-west opening case is specified. In all cases, ‘a’ and ‘b’ are positive.

Table: Conic Sections



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

- The non-circular conic sections are accurately those curves that, for a point 'F', a line 'L' not having 'F' and a number 'e' which is non-negative, are the locus of points whose distance to 'F' equals 'e' multiplies their distance to 'L'. 'F' is called the focus, 'L' the directrix, and 'e' the eccentricity.

  i. Linear eccentricity (c) is the space between the center and the focus.

  ii. Latus rectum (l) is parallel to the directrix and passing via the focus.

  iii. Semi-latus rectum (l) is half the latus rectum.

  iv. Focal parameter (p) is the distance from the focus to the directrix.

  The relationship for the above : p*e = l and a*e=c.

4. Describe the Hermite and Bezier curves. Write short notes on Analytical curve and synthetic curve in details. (JAN 2014, NOV 2015 & NOV 2016).

(Or)

Explain how a Bezier curve is defined. (NOV 2016, MAY 2016)

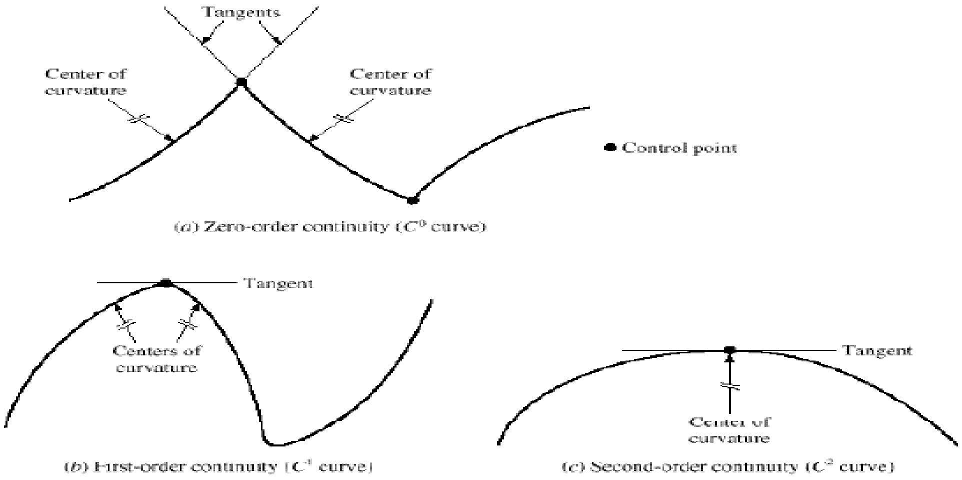
(Or)

What are the advantages of Bezier curves over cubic spline.(Mar06)

- The analytical and interpolated curves are insufficient to meet the requirements of mechanical parts that have complex curved shapes, such as, propeller blades, aircraft fuselage, automobile body, etc.
- These components contain non- analytical, synthetic curves. Design of curved boundaries and surfaces require curve representations that can be manipulated by changing data points, which will create bends and sharp turns in the shape of the curve.
- The curves are called synthetic curves, and the data points are called vertices or control

points. If the curve passes through all the data points, it is called an interpolant (interpolated). Smoothness of the curve is the most important requirement of a synthetic curve. Various continuity requirements at the data points can be specified to impose various degrees of smoothness of the curve. A complex curve may consist of several curve segments joined together. Smoothness of the resulting curve is assured by imposing one of the continuity requirements.

- A cubic polynomial is the lowest degree polynomial that can guarantee a C2 curve.
- Higher order polynomials are not used in CAD, because they tend to oscillate about the control points and require large data storage.
- Major CAD/CAM systems provide three types of synthetic curves: Hermite Cubic Spline, Bezier Curves, and B-Spline Curves.
- Zero order continuity (C0) assures a continuous curve, first order continuity (C1) assures a continuous slope, and a second order continuity (C2) assures a continuous curvature, as shown below.
- Cubic Spline curves pass through all the data points and therefore they can be called as interpolated curves.
- Bezier and B-Spline curves do not pass through all the data points, instead, they pass through the vicinity of these data points.
- Both the cubic spline and Bezier curve have first-order continuity, whereas, B-Spline curves have a second-order continuity.



(a) Zero-order continuity (C^0 curve)

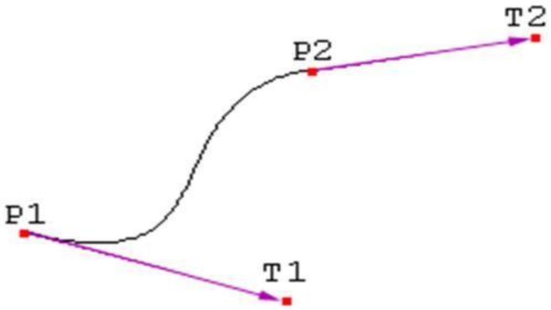
(b) First-order continuity (C^1 curve)

(c) Second-order continuity (C^2 curve)

Order of continuity of curve

Hermite curve

- A Hermite curve is a spline where every piece is a third degree polynomial defined in Hermite form: that is, by its values and initial derivatives at the end points of the equivalent domain interval.
- Cubic Hermite splines are normally used for interpolation of numeric values defined at certain dispute values x1,x2,x3, ….., xn, to achieve a smooth continuous function. The data should have the preferred function value and derivative at each Xk.
- The Hermite formula is used to every interval (Xk, Xk+1) individually. The resulting spline become continuous and will have first derivative.
- Cubic polynomial splines are specially used in computer geometric modeling to attain curves that pass via defined points of the plane in 3D space. In these purposes, each coordinate of the plane is individually interpolated by a cubic spline function of a divided parameter‘t’.
- Cubic splines can be completed to functions of different parameters, in several ways. Bicubic splines are frequently used to interpolate data on a common rectangular grid, such as pixel values in a digital picture.
- Bicubic surface patches, described by three bicubic splines, are an necessary tool in computer graphics. Hermite curves are simple to calculate but also more powerful. They are used to well interpolate between key points.



Hermite curve

The following vectors needs to compute a Hermite curve:

P1: the start point of the Hermite curve
T1: the tangent to the start point
P2: the endpoint of the Hermite curve
T2: the tangent to the endpoint

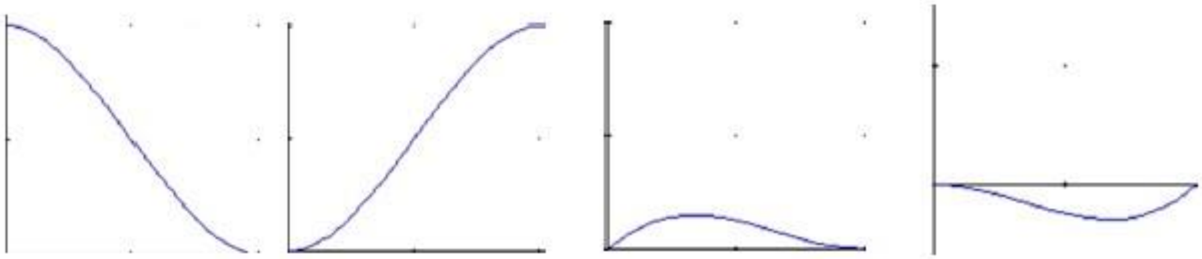
These four vectors are basically multiplied with four Hermite basis functions h1(s), h2(s), h3(s) and,h4(s) and added together.

h1(s) = 2s3 - 3s2 + 1

h2(s) = -2s3 + 3s2

h3(s) = s3 - 2s2 +s h4(s) = s3 - s2

Figure shows the functions of Hermite Curve of the 4 functions (from left to right: h1, h2, h3, h4).



Functions of Hermite curve

- A closer view at functions ‘h1’ and ‘h2’, the result shows that function ‘h1’ starts at one and goes slowly to zero and function ‘h2’ starts at zero and goes slowly to one.
- At the moment, multiply the start point with function ‘h1’ and the endpoint with function ‘h2’. Let s varies from zero to one to interpolate between start and endpoint of Hermite Curve. Function ‘h3’ and function ‘h4’ are used to the tangents in the similar way. They make confident that the Hermite curve bends in the desired direction at the start and endpoint.

- In matrix form, for hermite curve the equation can be written as,

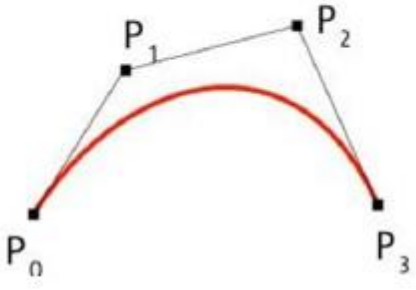
$$P(t) = [t^3 \quad t^2 \quad t \quad 1] \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} P(0) \\ P(1) \\ P'(0) \\ P'(1) \end{pmatrix}$$

The equation in short form can be written as: $P(t) = [t]\,[M]_H[G]$

Where, the terms $[M]_H$ is called Hermite matrix of a cubic spline, and represents the constant matrix. The term [G] is called geometric coefficient matrix.

Bezier curve

- Bezier curves are extensively applied in CAD to model smooth curves. As the curve is totally limited in the convex hull of its control points P0, P1,P2& P3, the points can be graphically represented and applied to manipulate the curve logically.
- The control points P0 and P3 of the polygon lie on the curve (Fig). The other two vertices described the order, derivatives and curve shape. The Bezier curve is commonly tangent to first and last vertices.
- Cubic Bezier curves and Quadratic Bezier curves are very common.
- Higher degree Bezier curves are highly computational to evaluate. When more complex shapes are required, Bezier curves in low order are patched together to produce a composite Bezier curve.
- A composite Bezier curve is usually described to as a ‘path’ in vector graphics standards and programs. For smoothness assurance, the control point at which two curves meet should be on the line between the two control points on both sides.

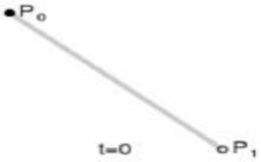


Bezier curve

- A general adaptive method is recursive subdivision, in which a curve's control points are verified to view if the curve approximates a line segment to within a low tolerance.
- If not, the curve is further divided parametrically into two segments, $0 \leq t \leq 0.5$ and $0.5 \leq t \leq 1$, and the same process is used recursively to each half. There are future promote differencing techniques, but more care must be taken to analyze error transmission.

- Analytical methods where a Bezier is intersected with every scan line engage finding roots of cubic polynomials and having with multiple roots, so they are not often applied in practice.
- A Bezier curve is described by a set of control points P0 through Pn, where ‘n’ is order of curve. The initial and end control points are commonly the end points of the curve; but, the intermediate control points normally do not lie on the curve.

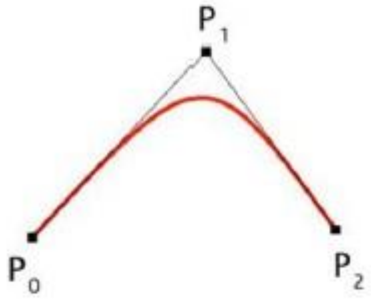
(i) **Linear Bezier curves**



Linear Bezier curve

As shown in the figure, the given points P0 and P1, a linear Bezier curve is merely a straight line between those two points. The Bezier curve is represented by
And it is similar to linear interpolation.

(ii) **Quadratic Bezier curves**



Quadratic Bezier curve

- As shown in the figure, a quadratic Bezier curve is the path defined by the function B(t), given points P0, P1, and P2,
- This can be interpreted as the linear interpolate of respective points on the linear Bezier curves from P0 to P1 and from P1 to P2 respectively. Reshuffle the preceding equation gives:

$$\mathrm{B}(t) = (1-t)^2\mathbf{P}_0 + 2(1-t)t\mathbf{P}_1 + t^2\mathbf{P}_2 \text{ , } t \in [0,1].$$

The derivative of the Bezier curve with respect to the value‘t’ is

$$\mathrm{B}'(t) = 2(1-t)(\mathbf{P}_1 - \mathbf{P}_0) + 2t(\mathbf{P}_2 - \mathbf{P}_1).$$

- From which it can be finished that the tangents to the curve at P0 and P2 intersect at P1. While‘t’ increases from zero to one, the curve departs from P0 in the direction of P1, then turns to land at P2 from the direction of P1.

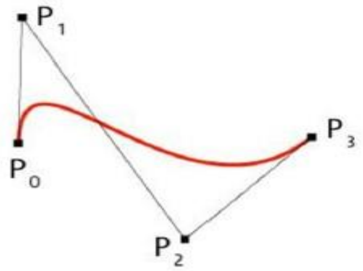
The following equation is a second derivative of the Bezier curve with respect to‘t’:

$$\mathrm{B}''(t) = 2(\mathbf{P}_2 - 2\mathbf{P}_1 + \mathbf{P}_0).$$

- A quadratic Bezier curve is represent a parabolic segment. Since a parabola curve is a conic section, a few sources refer to quadratic Beziers as ‘conic arcs’.

(iii) **Cubic Bezier curves**

- As shown in figure, four control points P0, P1, P2 and P3 in the higher-dimensional space describe as a Cubic Bezier curve. The curve begins at P0 going on the way to P1 and reaches at P3 coming from the direction of P2.
- Typically, it will not pass through control points P1 / P2; these points are only there to give directional data. The distance between P0 and P1 determines ‘how fast’ and ‘how far’ the curve travels towards P1 before turning towards P2.



Cubic Bezier curve

- The function B Pi, Pj, Pk (t) for the quadratic Bezier curve written by points Pi, Pj, and Pk, the cubic Bezier curve can be described as a linear blending of two quadratic Bezier curves:

The open form of the curve is:

$$B(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3 \ , t \in [0,1].$$

For several choices of P1 and P2 the Bezier curve may meet itself.

- Any sequence of any four dissimilar points can be changed to a cubic Bezier curve that goes via all four points in order. Given the beginning and ending point of a few cubic Bezier curve, and the points beside the curve equivalent to t = 1/3 and t = 2/3, the control points for the original Bezier curve can be improved.

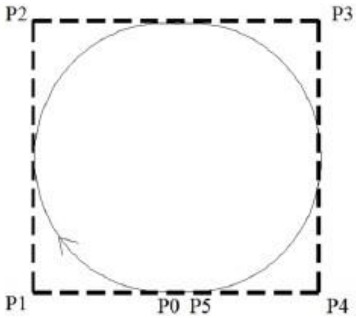
The following equation represent first derivative of the cubic Bezier curve with respect to t:

$$B'(t) = 3(1-t)^2(P_1 - P_0) + 6(1-t)t(P_2 - P_1) + 3t^2(P_3 - P_2) .$$

The following equation represent second derivative of the Bezier curve with respect to t:

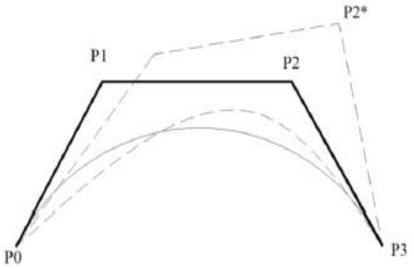
Properties Bezier curve

- The Bezier curve starts at P0 and ends at Pn; this is known as ‘endpoint interpolation’ property.
- The Bezier curve is a straight line when all the control points of a cure are collinear.
- The beginning of the Bezier curve is tangent to the first portion of the Bezier polygon.
- A Bezier curve can be divided at any point into two sub curves, each of which is also a Bezier curve.
- A few curves that look like simple, such as the circle, cannot be expressed accurately by a Bezier; via four piece cubic Bezier curve can similar a circle, with a maximum radial error of less than one part in a thousand.



Circular Bezier curve

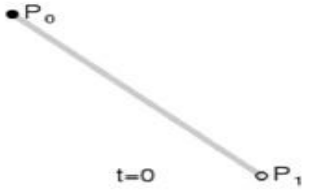
- Each quadratic Bezier curve is become a cubic Bezier curve, and more commonly, each degree ‘n’ Bezier curve is also a degree ‘m’ curve for any m > n.
- Bezier curves have the different diminishing property. A Bezier curves does not ‘ripple’ more than the polygon of its control points, and may actually ‘ripple’ less than that.
- Bezier curve is similar with respect to t and (1-t). This represents that the sequence of control points defining the curve can be changes without modify of the curve shape.
- Bezier curve shape can be edited by either modifying one or more vertices of its polygon or by keeping the polygon unchanged or simplifying multiple coincident points at a vertex.



Bezier curve shape

Construction of Bezier curves

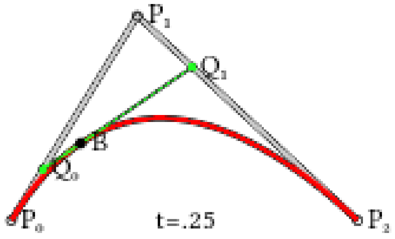
(i) Linear curves:



Construction of linear Bezier curve

- The figure shows the function for a linear Bezier curve can be via of as describing how far B(t) is from P0 to P1 with respect to ‘t’. When t equals to 0.25, B(t) is one quarter of the way from point P0 to P1. As ‘t’ varies from 0 to 1, B(t) shows a straight line from P0 to P1.

(ii) Quadratic curves



Construction of linear Quadratic curve

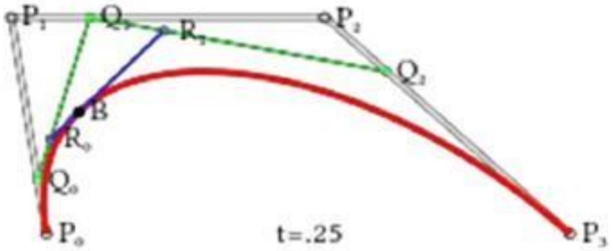
As shown in figure a quadratic Bezier curves one can develop by intermediate points Q0 and Q1 such that as ‘t’ varies from 0 to 1:

Point Q0 (t) modifying from P0 to P1 and expresses a linear Bezier curve.

Point Q1 (t) modifying from P1 to P2 and expresses a linear Bezier curve.

Point B (t) is interpolated linearly between Q0 (t) to Q1 (t) and expresses a quadratic Bezier curve.

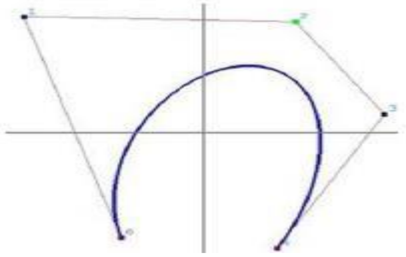
(iii) Higher-order curves



Construction of Higher-order curve

As shown in figure, a higher-order curves one requires correspondingly higher intermediate points. For create cubic curves, intermediate points Q0, Q1, and Q2 that express as linear Bezier curves, and points R0 and R1 that express as quadratic Bezier curves.

Rational Bezier curve



Rational Bezier Curve

- The rational Bezier curve includes variable weights (w) to provide closer approximations to arbitrary shapes. For Rational Bezier Curve, the numerator is a weighted Bernstein form Bezier and the denominator is a weighted sum of Bernstein polynomials. Rational Bezier curves can be used to signify segments of conic sections accurately, including circular arcs.

line Curve

- B-spline curves use a blending function, which generates a smooth, single parametric polynomial curve through any number of points.
- To generate a Bezier curve of the same quality of smoothness, we will have to use several pieces of Bezier curves.
- Unlike the Bezier curve, the degree of the polynomial can be selected independently of the number of control points.
- The degree of the blending function controls the degree of the resulting B-spline curve. The curve has good local control, i.e., if one vertex is moved, only some curve segments are affected, and the rest of the curve remains unchanged.

The mathematical derivation of the B-spline curve is complex and beyond the scope of this course. The equation is of the form:

$P\ (t) = \Sigma N_{i,k}(t)\ V_i$

Where, P(t) is a point on the curve.

i indicates the position of control

point i k is order of curve

$N_{i,k}(t)$ are blending functions

$V_i$are control points

The matrix form of the uniform cubic B-spline curve is:

$$P_i\,(t) = 1/6[t^3 \quad t^2 \quad t \quad 1] \begin{array}{cccc} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{array} \quad \begin{array}{c} V_{i-1} \\ V_i \\ V_{i+1} \\ V_{i+2} \end{array}$$

5. Explain briefly about the surfaces and its types?

(Or)

Describe the effect of characteristic polyhedron over the resulting Bezier surface.

(Or)

Techniques for surface modeling

(Or)

Write short notes on surface patch, Coons and bicubic patches (NOV 2016)

(Or)

Explain the Bezier and B-spline surfaces

- ➢ Wire frame models are unable to represent complex surfaces of objects like car, ship, airplane wing, castings etc.
- ➢ A surface model can be used to represent the surface profile of these objects. Also, surface model can be used for calculating mass properties, interference between parts, generating cross-sectioned views, generating finite element mesh, and generating NC tool paths for continuous path machining.
- ➢ Additionally, surface model can be used to fit experimental data, discretized solutions of differential equations, construction of pressure surface, construction of stress distribution etc.
- ➢ Surface creation on a CAD system usually requires wire frame entities: lines, curves, points, etc. All analytical and synthetic curves can be used to generate surfaces.
- ➢ In order to visualize surfaces on a graphic display, a mesh, say mxn in size is usually

displayed; the mesh size is controlled by the user. Most CAD systems provide options to set the mesh size.

- A surface of an object is more complete and less ambiguous representation than its wire frame model; it is an extension of a wireframe 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).
- Databases of surface models are centralized and associative; manipulation of surface entities in one view is automatically reflected in the other views. Surface model scan be shaded and represented with hidden lines.

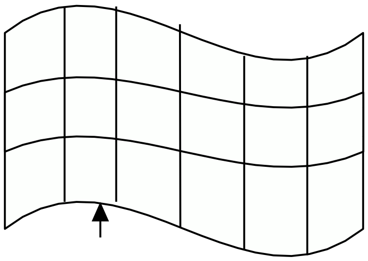
Types of Surfaces

Plane Surface

- This is the simplest surface, requires 3 non-coincidental points to define an infinite plane. The plane surface can be used to generate cross sectional views by intersecting a surface or solid model with it.

Ruled(lofted)Surface

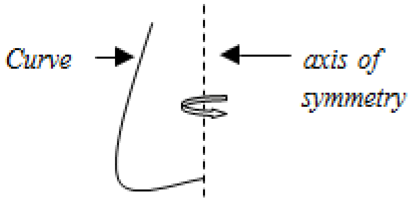
- This is a linear surface. It interpolates linearly between two boundary curves that define the surface. Boundary curves can be any wire frame entity. The surface is ideal to represent surfaces that do not have any twists or kinks.

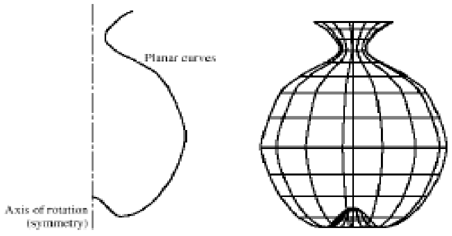


Boundary Curve

Surface of Revolution

- This is an axisymmetric surface that can model axisymmetric objects. It is generated by rotating a planar wire frame entity in space about the axis of symmetry of a given angle.

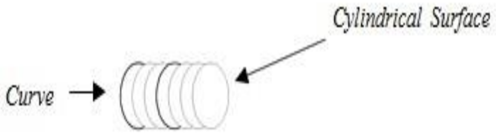


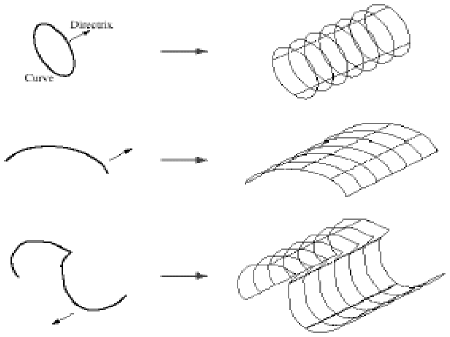


Surface of revolution.

Tabulated Surface

- This is a surface generated by translating a planar curve a given distance along a specified direction. The plane of the curve is perpendicular to the axis of the generated cylinder.

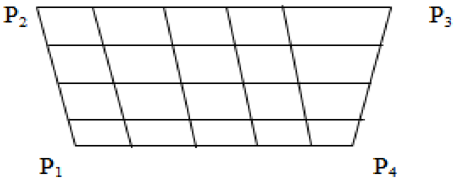




Tabulated surface.

Bi-linear Surface

- This 3-D surface is generated by interpolation of 4 endpoints. Bi-linear surfaces are very useful in finite element analysis. A mechanical structure is discretized into elements, which are generated by interpolating 4 node points to forma 2-D solid element.

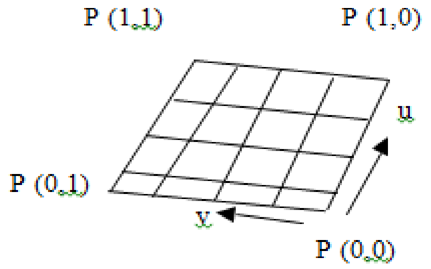


Interpolated Surfaces – Bilinear Surface

A bilinear surface is obtained by linear interpolation between four points, which may or may not lie in the same plane. The four points appear as vertices or corner points and the parameter values u and v create lines at various intervals to provide the surface visibility, shown in the figure. The parameters u and v are defined as

$0 \leq u \leq 1$, and $0 \leq v \leq 1$

Bilinear Patch



The interpolated parametric equation of a bilinear surface is given as:

$$P(u,v) = (1-u)(1-v)P(0,0) + u(1-v)P(1,0) + (1-u)P(0,1) + u\,v\,P(1,1)$$

In matrix form, it can be written as

$$P(u,v) = [(1-u)(1-v) \;\; u(1-v) \;\; (1-u)vuv]$$

$$\begin{bmatrix} P(0,0) \\ P(1,0) \\ P(0,1) \\ P(1,1) \end{bmatrix}$$

Node points in FEA

Application of Bilinear Surfaces

- Bilinear patches are extensively used in 2-D finite element analysis (FEA).
- In FEA, an engineering structureisdefined by several bilinear surfaces(elements), which are created by joining points on the structure's geometry, called nodes.
- The nodes are connected to other nodes to create quadrilateral surfaces.Points not lying onthe nodes are calculated by interpolation. Thus, the entire structure is completely defined by the nodes and the bilinear surfaces.

Drawbacks of Bilinear Surfaces

- Bilinear surfaces have a very limited use mainly for FEA.Since only 4points can be used in the interpolation, the smoothness of the generated surface is limited.

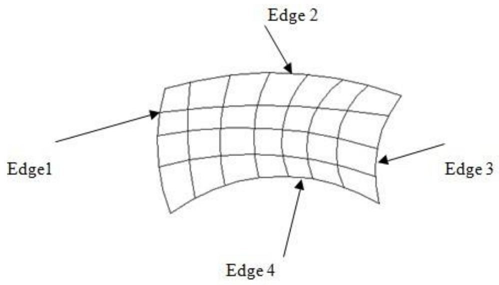
- Additionally, there is no flexibility to control shapes of the surface, unlike the sweeped surfaces.

Surface Patch

- The patch is the fundamental building block for surfaces. The two variables u and v vary across the patch; the patch may be termed bi parametric. The parametric variables often lie in the range 0 to 1. Fixing the value of one of the parametric variables results in a curve on the patch in terms of the other variable (Isoperimetric curve). Figure shows a surface with curves at intervals of u and v of 0 : 1.

Coons Patch

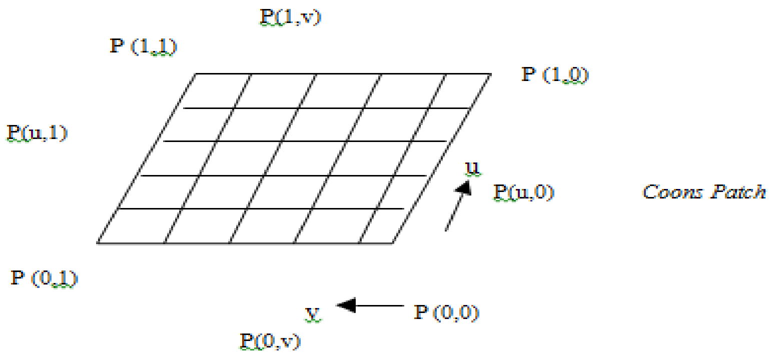
Coons patch or surface is generated by the interpolation of 4 edge curves as shown.



Interpolated Surfaces – Coons Patch

- A linear interpolation between four bounded curves is used to generate a Coons surface, also called as Coons patch. The method is credited to S. Coons who developed this concept for generating a surface.
- Linear interpolation between the boundary curves P(0,v), P(u,0), P(1,v) , and P(u,1) gives the equation

$$Q(u,v) = (1-v)\ P(u,0) + u\ P(1,v) + v\ P(u,1) + (1-u)\ P(0,v)$$



- Theaboveequationgiveswrongvaluesatthecorners(u,v=0and1).Forexample, substituting the values of u and v we get,

$$Q(0,0) = P(0,0) + P(0,0) = 2P(0,0)$$

$$Q(1,0) = 2P(1,0), \text{ etc.}$$

- Which are obviously wrong values. Therefore, The coons patch is created by modification of the interpolation equation, where the corners are subtracted. The modified interpolation equation is given as,

$$P(u,v) = (1-v)\, P(u,0) + u\, P(1,v) - v\, P(u,1) + (1-u)\, P(0,v) -$$
$$(1-u)(1-v)\, P(0,0) - u\,(1-v) \times P(1,0) - (1-u)\, v\, P(0,1) - u\, v\, P(1,1).$$

For computational purposes, it is more convenient to write this equation as,

$$P(u,v) = [(1-v) \quad u \quad v \quad (1-u)\,] \begin{pmatrix} P(u,0) \\ P(1,v) \\ P(u,1) \\ P(0,v) \end{pmatrix} \qquad \text{Eqns. of the boundary curves}$$

$$-\,[(1-u)(1-v) \quad u(1-v) \quad (1-u)v \quad uv] \begin{pmatrix} P(0,0) \\ P(1,0) \\ P(0,1) \\ P(1,1) \end{pmatrix} \qquad \text{End-points (coordinates)}$$

Which gives,

$$Q(u,v) = [(1-u) \quad u \quad 1] \begin{pmatrix} -P(0,0) & -P(0,1) & P(0,v) \\ -P(1,0) & -P(1,1) & P(1,v) \\ P(u,0) & P(u,1) & 0 \end{pmatrix} \begin{pmatrix} (1-v) \\ v \\ 1 \end{pmatrix}$$

Other interpolated surfaces include the Parametric Cubic patches.

Applications

- Coons surface is easy to create, and therefore, many 2-D CAD packages utilize it for generating models. However, it has only a limited application since the surface is inflexible and cannot create very smooth surfaces.
- It would be very difficult to produce a smooth automobile fender using the Coons surface. Several CAD software, including AutoCAD, use this surface for generating surfaces between 4-bounded edges.

Bicubic Patch

- The bi-cubic patch is used for surface descriptions defined in terms of point and tangent vector information. The general form of the expressions for a bi-cubic patch is given by:

$$\overrightarrow{p}(u, v) = \sum_{i=0}^{3} \sum_{j=0}^{3} \overrightarrow{k_{ij}}\, u^i\, v^j$$

- This is a vector equation with 16 unknown parameters k_{ij} which can be found by Lagrange interpolation through 4 x 4 grid.

Bezier Surface

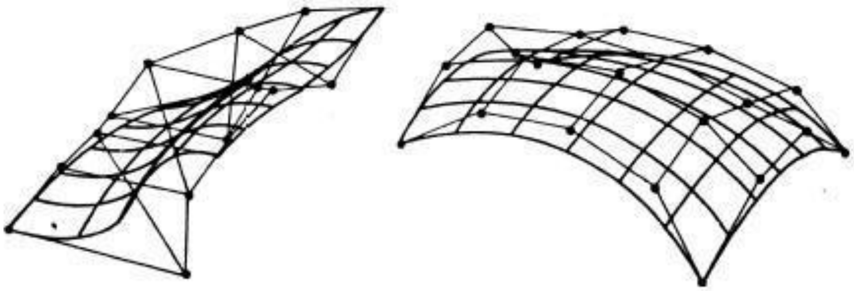
- ThisisasyntheticsurfacesimilartotheBeziercurveandisobtainedbytransformation ofaBeziercurve.Itpermitstwistsandkinksinthesurface.Thesurfacedoesnotpass through all the data points.
- The Bezier surface formulation use a characteristic polygon
- Points the Bezier surface are given by

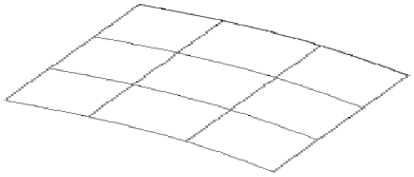
$$\overrightarrow{p}(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \overrightarrow{B}_{i,m}(u)\, \overrightarrow{B}_{j,n}\, \overrightarrow{p_{ij}};\ u, v, \epsilon\ [0, 1]$$

Where,

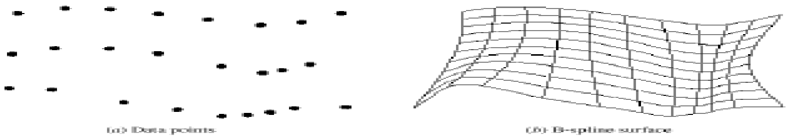
$\overrightarrow{p_{ij}};$ - Vertices of the characteristic polygon

$\overrightarrow{B}_{j,n}$ $\overrightarrow{B}_{i,n}$, - Blending functions





Bezier surface.



(a) Data points

(b) B-spline surface

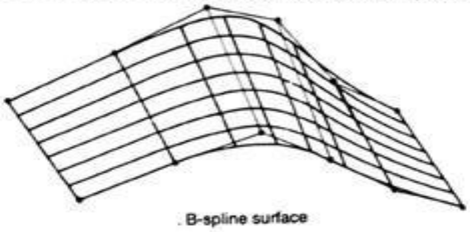
B-spline surface.

line Surface

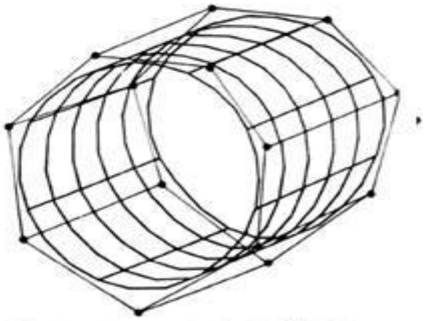
- Thisisasyntheticsurfaceanddoesnotpassthroughalldatapoints.Thesurfaceis capable of giving very smooth contours, and can be reshaped with local controls.
- MathematicalderivationoftheB-splinesurfaceisbeyondthescopeofthiscourse.Only limited mathematical consideration will be given here.
- Computer generated surfaces play a very important part in manufacturing of engineering products. A surface generated by a CAD program provides a very accurate and smooth surface, which can be generated by NC machines without any room for misinterpretation.
- Therefore, in manufacturing, computer generated surfaces are preferred. Since surfaces are mathematical models, we can quickly find the centroid, surface area, etc. Another advantage of CAD surfaces is that they can be easily modified.
- The expression for the B-spline surfaces is given by

$$\vec{p}(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,k}(u)\, N_{j,l}(v)\, \vec{p_{ij}}$$

p_{ij} are the vertices of the defining polygon and $N_{i,k}$ and $N_{j,l}$ are blending functions.



. B-spline surface

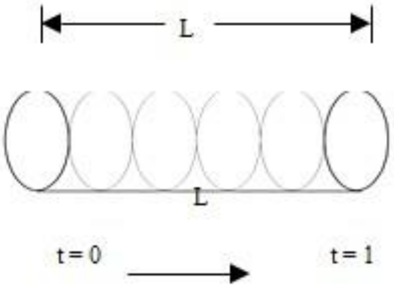


B-spline surface for a closed control polygon

6. With respect to surface modeling explain the concept of surface revolution and tabulated cylinder in detail. (JAN 2015)

Linearly Swept Surfaces

- A swept surface is generated when a curve is parametrically translated or rotated.
- In CAD, a surface is represented by a series of curves, which are parametrically generated at various instances.
- For example, a cylindrical surface is generated when a circular arc is translated up to the given dimension using a parameter t, where t varies as, 0≤ t ≤1.
- In the figure shown, the cylindrical surface is generated when a circular arc is translated a distance L, with the interim instances at t = 0.1, 0.2, 0.3, … 1.
- Here, the parameter t is given 10 values, and therefore, the surface of the cylinder is represented by 10 circular curves.
- The appearance of the surface improves as the parameter t varies at smaller intervals. Thus, if it is varied with ∆t = 0.01, there will be 100 circular curves representing the surface.



- A surface is an extension of a curve. The parametric representation of a curve is given by a single-vector equation of the form:

  P(t) = [x(t) y(t) z(t)]

- Here, only one parametric variable or one degree of freedom is needed. Whereas, a surface representation requires two parametric variables, and the equation is given as:

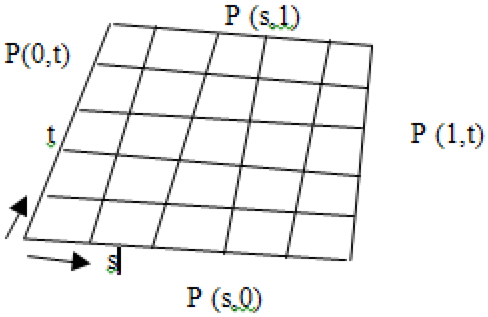
  Q(s,t) = [x(s,t) y(s,t) z(s,t)]

- Tracing a point in the s and t directions, as shown in the figure on the next page, generates a surface. One parameter variable is kept constant while varying the other one. A series of curves is created along the s and t directions. For example, constraining the parameters s and t between zero and 1, the set of curves generated along the s direction is,

  P(0,t), P(0.1,t), P(0.2, t)………….P(1, t)

  and the other set of curves along the t direction is,

  P(s,0), P(s,0.1),…P(S, 0.9), P(s,1).



- Thus, creation of a surface requires creation of the multiple curves that constitute it.

This concept can be applied to both, the surface that has an analytical formulation (conic sections) and to a free-form surface (Bezier, B-spline).

Revolved Surfaces (Circular Sweep)

- Surface of revolution is obtained by rotating a plane-curve around an axis. In the figure shown, line AB is rotated about the z-axis through an angle of 2πradians, generating a cylinder. A line or curve when revolved can generate all kinds of surfaces, based on the condition of rotation. Any point on the surface is a function of two parameters t and θ.
- Thus, the generated surface is a rotational transformation of a line (or curve), except θ is not constant, but has values, $0 \leq \theta \leq 2\pi$.

Circular Sweep of a Synthetic Curve

- Equation of a synthetic curve (free-form curve), as derived earlier, is given as,

P (t) = [t] [M] [V]

- The surface of revolution is then give by,

  $Q(t, \theta) = [t] [M] [V] [Tr]^{\theta} = [Q(t)][Tr]^{\theta}$

Where, $Q(t,\theta)$ is the equation of the curve, and $[Tr]^{\theta}$is the rotation matrix about the z- axis.

Note: To rotate the curve about the axis, we will have to use the translation and rotation matrices.

Creating a Surface by Parametric Sweeping

- In the examples given above, sweeping a curve parametrically generated the surfaces. In parametric sweeping procedure, a surface is generated through the movement of a line or a curve along or around a defined path. The curve is sweeped as the sweep parameter is varied from the values of 0 to 1, creating several instances of the curve along the sweep path. In general, the equation of the surface can be given as,

$$Q(t, s) = P(t)\, T(s)$$

- Where, P (t) is the parametric equation of a curve and T(s) is the sweep transformation based on the shape of the path. The sweep transformation can consist of translation, scaling, rotation or a combined transformation. If the path is a straight line, the points along the path on the line can be represented by,

$$x(s) = as \quad y(s = bs \quad z(s) = cs$$

and T (s) is given as,

$$T(s) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ as & bs & cs & 1 \end{pmatrix}$$

- ➢ Where, a, b, c are coordinate values, and $0 \leq s \leq 1$
- ➢ This is equivalent to a three-dimensional translation of a curve with several traces generated along the path, controlled by how the parameter s is varied.

Creating a Surface by Sweeping a polygon

Any polygon can be sweeped around a given path to generate a surface. The equation of the surface is given as,

$$Q(s, t) = [P]\{T(s)]$$

Where, [P] is the point matrix, and T(s) is the transformation matrix.

Creating a Parametric Cubic Patch

Parametric cubic patch or surface is generated by four boundary curves; the curves are parametric cubic polynomials. The equation of a parametric cubic curve was defined earlieras:

$$= [t^3 \quad t^2 \quad t \quad 1] \underbrace{\begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}}_{\textit{Constant matrix for } n = 3} \underbrace{\begin{pmatrix} P(0) \\ P(1) \\ P'(0) \\ P'(1) \end{pmatrix}}_{\textit{geometry matrix}}$$

Where P(0) = Coordinates of the first point at t = 0

P(1) = coordinates of the last point at t = 1

P'(0) = values of the slope sin x, y, z directions at t = 0

P'(1) = values of the slope sin x, y, z directions at t = 1

Analogous to a cubic curve, a parametric cubic surface can be defined by16 points:

- 4 points for coordinates of the corner points
- 8 points for slopes in the s & t directions
- 4 points for twist vectors (second derivatives)

Using a procedure similar to the one carried out in the derivation of the cubic curve, we can derive the geometric coefficient matrix for the surface, which is given as,

$$[G]_H = \left(\begin{array}{cc|cc} P(0,0) & P(0,1) & P_t(0,0) & P_t(0,1) \\ P(1,0) & P(1,1) & P_t(1,0) & P_t(1,1) \\ \hline P_s(0,0) & P_s(0,1) & P_{st}(0,0) & P_{st}(0,1) \\ P_s(1,0) & P_s(1,1) & P_{st}(1,0) & P_{st}(1,1) \end{array}\right)$$

This can be broken into 4 groups as,

$$\left(\begin{array}{c|c} \text{Position of corner points,} & \text{Derivatives w.r.t. t of corner points} \\ \hline \text{Derivatives w.r.t s at corner points,} & \text{Cross derivatives at corner points} \end{array}\right)$$

Twist vectors, not shown here, are the partial derivatives: $dP_s/dt\&dP_t/ds$. These vectors control the internal shape of the surface.

With the geometric coefficient matrix defined, the equation of the surface can be written as,

$$P(s.t) = [s]\,[M]_H\,[G]_H\,[M_H]^T[t]^T$$

Where: $[s] = [s^3 \;\; s^2 \;\; s \;\;\; 1]$

$[M]_H$= [Constant matrix for n = 3

$][M_H]^T$= Transpose of $[M]_H$

$[G]_H$= Geometry matrix as defined by the 16 points, and

$$[t]^T = \begin{matrix} t^3 \\ t^2 \\ t \end{matrix}$$

7. Write short notes on Bezier Surface

Bezier Surface

- Just as parametric cubic curves are extended to parametric cubic patches, Bezier curves may be extended to Bezier surface patch.
- While the surface passes through the four corner points, the control points control all

other points on the surface.

- Using the placement of these points to specify edge slope is more intuitive than determining the parametric slopes and twist vectors for the parametric cubic curve surface. Bezier surface, as a result, is easier to use because the control points themselves approximate the location of the desired surface.
- Bezier surfaces can be generated with any order of the Bezier curve. Two surface patches can be joined and the two surfaces do not have to be of the same order, one can be cubic and the other a quadratic.
- Blending Bezier patches with slope continuity requires that(1) control points on the common edges be shared and (2) three control points – one on the edge and ones on the either sides of the edge – forma straight line, as shown in the figure below.

In Bezier surface:

- The surface takes the general shape of the control points.
- The surface is contained within the convex hull of the control points.
- The corner of the surface and the corner control points are coincident.

General Equation of the Bezier surface is given as,

$$Q(s,t) = \Sigma\Sigma V_{i,j}B_{i,n}(s)B_{j,m}(t)$$

$$0 \leq s,t \leq 1$$

$V_{i,j}$ defines the control points

$B_{i,n}(s)$&$B_{j,m}(t)$are the Bernstein blending functions in the s and t directions.

In matrix form, the Bezier surface canbe represented by,

$$Q(s,t) = [S]\,[M]_B\,[V]_B[([M)^T]_B[t]^T$$

For a cubic surface this equation reduces to:

$$Q(s,t) = [s^3 \quad s^2 \quad s \quad 1] \begin{pmatrix} -1 & 3 & -3 & 1 \\ -3 & -6 & 3 & 0 \\ 3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_{0,0} & v_{0,1} & v_{0,2} & v_{0,3} \\ v_{1,0} & v_{1,1} & v_{1,2} & v_{1,3} \\ v_{2,0} & v_{2,1} & v_{2,2} & v_{2,3} \\ v_{3,0} & v_{3,1} & v_{3,2} & v_{3,3} \end{pmatrix} x$$

$$\begin{pmatrix} 1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}$$

Note that, to represent a cubic Bezier surface, 16 control points must be specified, and several Bezier surfaces can be combined to create a complex surface.

8. Explain geometric modeling and types of model representation.

Classification of Geometric Modeling

Computer representation of the geometry of a component using software is called a geometric model. Geometric modeling is done in three principal ways. They are:

i. Wire frame modeling

ii. urface modeling

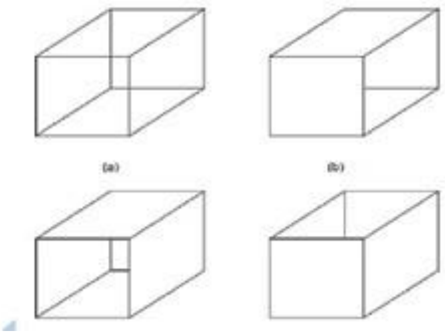
iii. Solid modeling

These modeling methods have distinct features and applications.

Wire Frame Modeling

- In wire frame modeling the object is represented by its edges. In the initial stages of CAD, wire frame models were in 2-D. Subsequently 3-D wire frame modeling software was introduced.
- The wire frame model of a box is shown in Fig. (a). The object appears as if it is made out of thin wires. Fig. (b), (c) and 6.2(d) show three objects which can have the same wire frame model of the box.
- Thus in the case of complex parts wire frame models can be confusing. Some clarity can be obtained through hidden line elimination.

- Though this type of modeling may not provide unambiguous understanding of the object, this has been the method traditionally used in the 2-D representation of the object, where orthographic views like plan, elevation, end view etc are used to describe the object graphically



A comparison between 2-D and 3-D models is given below:

2 - D Models	3-D Wire Frame Models
1.Ends (vertices) of lines are represented by their X and Y coordinates 2.Curved edges are represented by circles, ellipses, splines etc. Additional views and sectional views are necessary to represent a complex object with clarity. 3.3-D image reconstruction is tedious. Uses only one global coordinate system	1.Ends of lines are represented by their X, Y and Z coordinates. 2.Curved surfaces are represented by suitably spaced generators. Hidden line or hidden surface elimination is a must to interpret complex components correctly. 3.2-D views as well as various pictorial views can be generated easily. May require the use of several user coordinate systems to create features on different faces of the component.

SURFACE MODELING

- In this approach, a component is represented by its surfaces which in turn are represented by their vertices and edges. For example, eight surfaces are put together to create a box, as shown in Fig.
- Surface modeling has been very popular in aerospace product design and automotive design. Surface modeling has been particularly useful in the development of manufacturing codes for automobile panels and the complex doubly curved shapes of aerospace structures and dies and moulds.

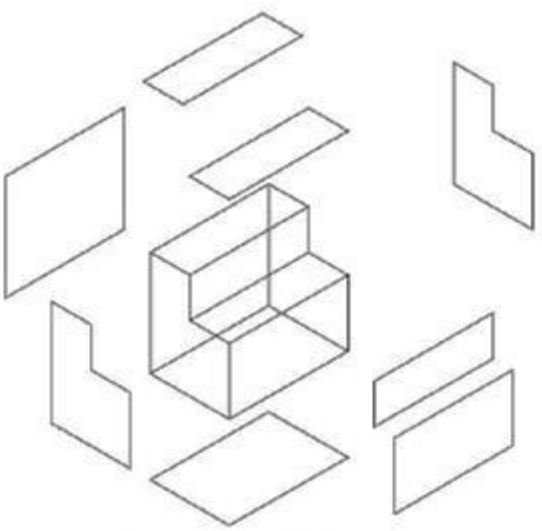


Fig. Surface Representation

- Apart from standard surface types available for surface modeling (box, pyramid, wedge, dome, sphere, cone, torus, dish and mesh) techniques are available for interactive modeling and editing of curved surface geometry.
- Surfaces can be created through an assembly of polygonal meshes or using advanced curve and surface modeling techniques like B-splines or NURBS (Non-Uniform Rational B-splines).
- Standard primitives used in a typical surface modeling software are shown in Fig.. Tabulated surfaces, ruled surfaces and edge surfaces and revolved are simple ways in which curved geometry could be created and edited.

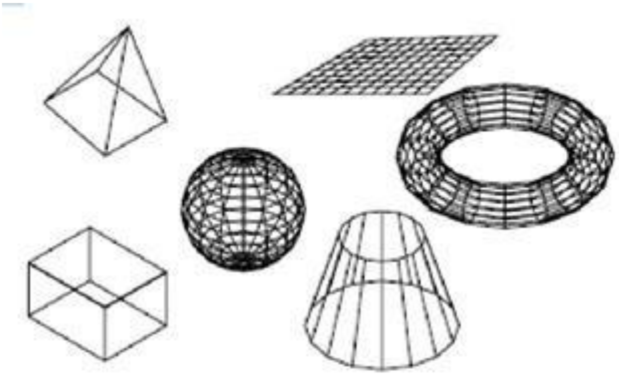


Fig. Typical Approaches in Surface Modeling

SOLID MODELING (NOV 2015)

The representation of solid models uses the fundamental idea that a physical object divides the 3-D Euclidean space into two regions, one exterior and one interior, separated by the boundary of the solid. Solid models are:

- bounded
- Homogeneously three dimensional
- Finite

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)

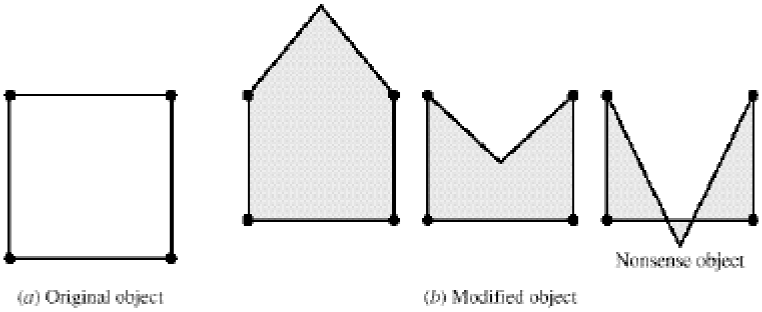
ii. Boundary representation (B-Rep)

iii. Hybrid method which is a combination of B-Rep and CSG.

9. Explain brief notes on Boundary representation (basic elements, faces, Euler equation, Boundary models). (NOV 2016, MAY 2016)

Boundary Representation (B-rep)

- A B-rep (boundary) model of an object consists of faces, edges, vertices, loops and handles.
- A **face** is closed, orient able, and bounded surface An edge is a bounded curve.
- A **vertex** is a point in E3. A loop is a hole in a face.
- A **handle** is a through hole in a solid (body).
- The database of a boundary model contains both its topology and its geometry.
- Topology is created by performing Euler operations and geometry is created by performing Euclidean calculations.
- Euler operations are used to create, manipulate and edit the faces, edges, and vertices of a boundary model. Euler operators, as Boolean operators, ensure the validity (closeness. no dangling faces or edges, and so forth) of boundary models.
- Geometry includes coordinate. of vertices and rigid motion and transformation.
- Topology and geometry are interrelated and cannot be separated entirely from each other.
- Figure shows a square which, after dividing its top edges by introducing a new vertex, is still valid topologically but produces a nonsense object depending on the geometry of the new vertex.



(a) Original object

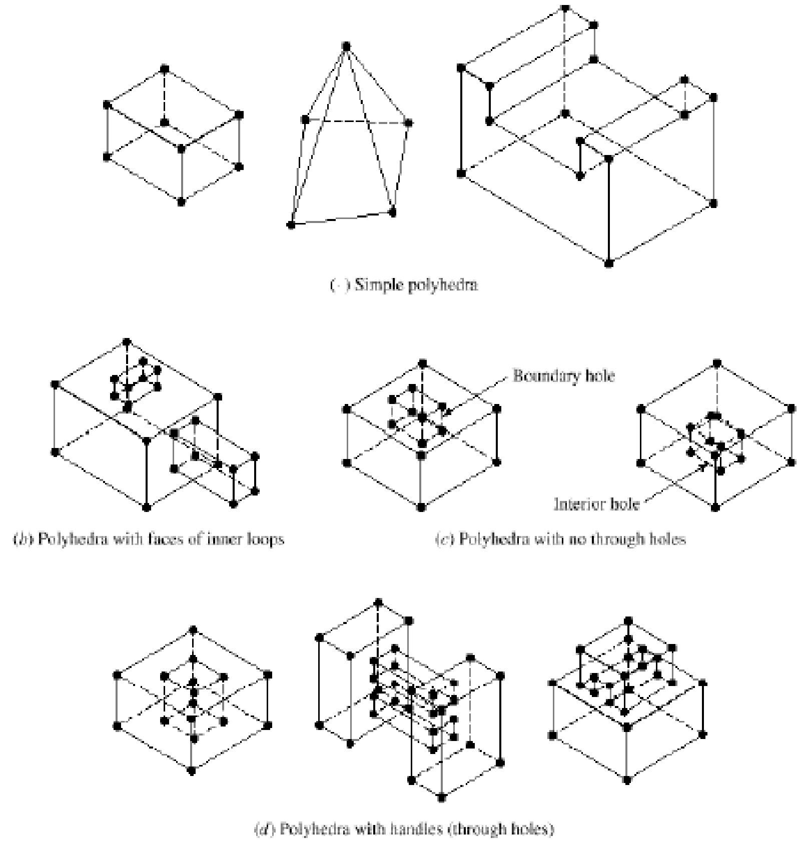
(b) Modified object

Effect of topology and geometry on boundary models.

Basic Elements

- Objects that are often encountered in engineering applications can be classified as either polyhedral or curved objects.
- A polyhedral object (plane-faced polyhedron) consists of planar faces (or sides) connected at straight (linear) edges which, in tum, are connected at vertices.
- A cube or a tetrahedron is an obvious example. A curved object (curved polyhedron) is similar to a polyhedral object but with curved faces and edges instead.
- The identification of faces, edges, and vertices for curved closed objects such as a sphere or a cylinder needs careful attention as will be seen later in this section. Polyhedral objects are simpler to deal with and arc covered first.
- The basic elements of a B-rep are faces, edge, vertices, loops, handles, and bodies (solids themselves). B-rep models can be simple or complex, as Figure shows. Polyhedral objects can be classified into four classes.
- **First class:** The first class, Figure a) is the simple polyhedra. These do not have holes, and each face is bounded by a single set of connected edges.
- **Second class**: The second class, Figure b), is similar to the first with the exception that a face may have loops.
- **Third class**, Figure c, includes objects with holes that do not go through the entire object. For this class, a hole may have a face coincident with the object boundary; in this case we call it a boundary hole. Or if it is an interior hole (as a void or crack inside the object). it has no faces on the boundary.
- The **fourth class**, Figure 9.2ld, includes objects that have holes that go through the entire object. Topologically, these through holes are called handles.

- The topological name for the number of handles in an object is genus. There exist topological constraints on the B-rep elements to ensure model validity. A vertex is always a unique point (an ordered triplet) in space. An edge is a nonself-intresecting, directed space curve bounded by two vertices that are not necessarily distinct (as in the case of a closed edge).



( ) Simple polyhedra

(b) Polyhedra with faces of inner loops

(c) Polyhedra with no through holes

(d) Polyhedra with handles (through holes)

Types of polyhedral objects

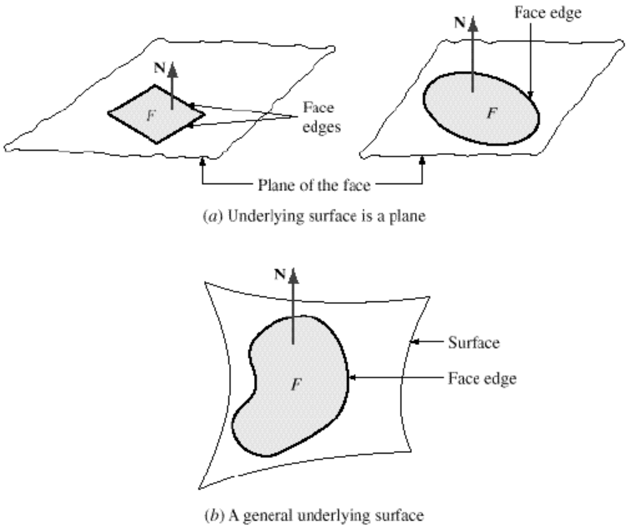
A **face** is a finite connected, non self-intersecting region of a closed oriented surface bounded by edges.

A **loop** defines a non self-intersecting, piecewise, closed space curve. The body (sometimes called a shell) is the solid itself. A minimum body is a point.

The **object** on the right of Figure c) has two bodies (the exterior and interior cubes), and any other object in Figure has only one body. A solid that has more than one body is known as a disjoint solid.

Some CAD/CAM systems may not allow the creation of disjoint solids. An example is cutting a block with another block to produce two disconnected blocks.

- Faces of boundary models possess certain essential properties and characteristics that ensure the regularity of the model, that is, the model has an interior and a boundary.
- The face of a solid is a subset of the solid boundary, and the union of all faces of a solid defines such boundary. Face is 2D homogeneous regions, so they have areas and no dangling edges.
- In addition, a face is a subset of some underlying closed oriented surface. Figure shows the relationship between a face and its surface.
- At each point on the face, there is a surface normal N that has a sign associated with it to indicate if it points into or away from the solid interior.
- One convention is to assume that N is positive if it points away from the solid. It is desirable, but not required, that a face has a constant surface normal.



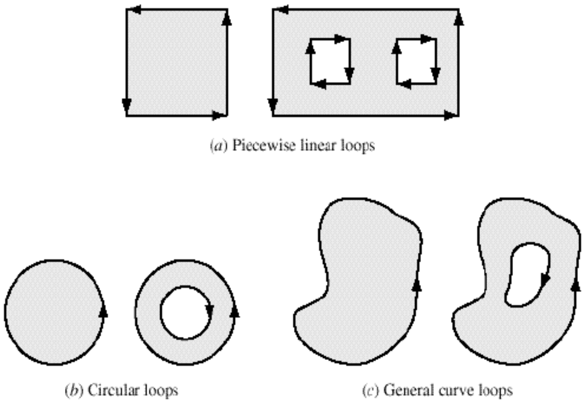
(*a*) Underlying surface is a plane

(*b*) A general underlying surface

Definition of a face

- The representation of a face must ensure that both the face and the solid interiors can be deduced from the representation.
- The direction of the face's surface normal is used to indicate the inside or outside of the model. The face interior can be determined by traversing the face loops in a certain direction or assigning flags to them.
- In traversing loops, the boundary edges of the face (known as the **face outer loop**) are traversed.

For example in a counterclockwise direction and the edges of the inner loops are traversed in the opposite direction, clockwise. Figure shows some traversal examples.



(*a*) Piecewise linear loops

(*b*) Circular loops

(*c*) General curve loops

Traversal of face's loops.

- Euler Equations Having defined the boundary model element proved that polyhedra that are homomorphic to a sphere, that is, their faces are non self-intersecting and belong to closed orient able surfaces. are topologically valid if they satisfy the following equation:

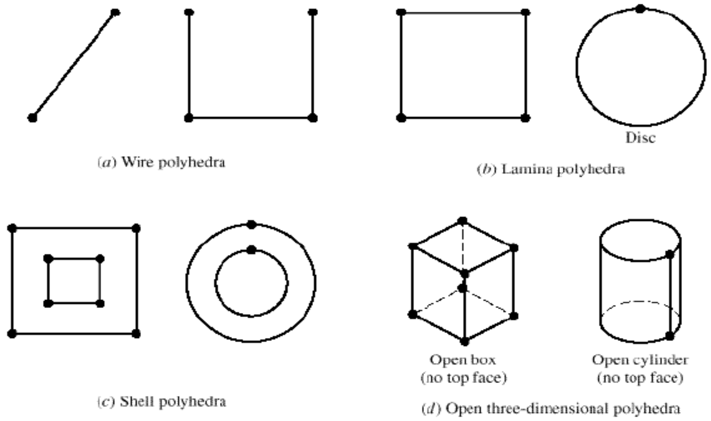
$$F-E+ V-L = 2 B-G$$

Where F. E, V, L, B, and G are the number of faces, edges. Vertices faces inner loops, bodies, and genus. Eq. is known as the **Euler or Euler-Poincare Law**.

- The simplest version of this equation is F - E + V = 2, which applies to polyhedra shown in Figure a. shows the counts of the various variables of Eq. for polyhedra shown in Figure.
- The numbering of these polyhedra in the table is taken from left to right. top to bottom with the top left cube being polyhedron number 1 and the bottom right object being number 9.
- Euler's Jaw given by Eq. applies to closed polyhedral objects only. However, open polyhedral objects do not satisfy Eq. This class of objects includes open polyhedra that may result from the construction of boundary models of closed objects as well as all 20 polygonal objects. Open objects satisfy the Following Euler's law:

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

- Shows some examples of open objects. The reader can easily verify that they satisfy Eq. In the equation, B refers to an open body which can be a wire, an area, or a volume.
- All the objects in Figure have one body, and only the two bodies of Figure c have one genus each.



(*a*) Wire polyhedra

(*b*) Lamina polyhedra

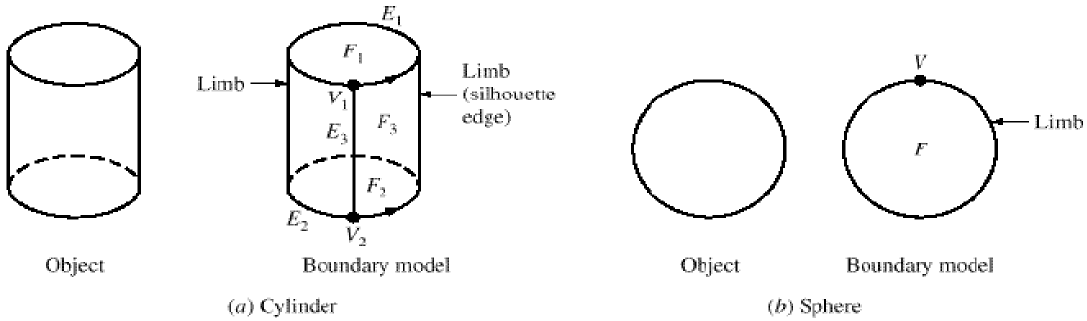
(*c*) Shell polyhedra

(*d*) Open three-dimensional polyhedra

Open polyhedral objects.

Curved and Faceted B-rep Models

- We now turn from polyhedral objects to curved objects such as cylinders and spheres. The same rules and guidelines for boundary modeling discussed thus far apply.
- The major difference between the two types of objects results if closed curved edges or faces exist. Consider, for example, the closed cylinder and sphere shown in Figure.
- A closed cylindrical face has one edge and two vertices and a spherical face has one vertex and no edges. The boundary model of a closed cylinder has three faces (top, bottom, and cylindrical face itself), two vertices, and one edge connecting the two vertices.
- The other "edges" are for visualization purposes. They are called limbs, virtual edges, or silhouette edges. The boundary model of a sphere consists of one face, one vertex, and no edges.

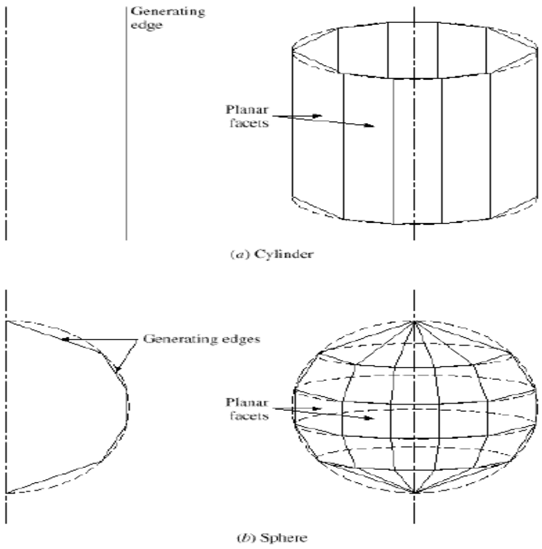


(*a*) Cylinder

(*b*) Sphere

B-rep of a cylinder and a sphere

- ➢ If the curved objects are represented by storing the equations of curves and surfaces of edges and faces, respectively, the resulting boundary scheme is known as an exact **B-rep scheme**.
- ➢ Another alternative is the approximate or faceted B-rep (sometimes called tessellation rep). In this scheme, any curved face is divided into planar facets Figure shows a faceted B-rep of a cylinder and sphere.
- ➢ The faceted cylinder is generated by rotating a line incrementally about the cylinder axis the desired total number of facets.
- ➢ A faceted sphere is formed in a similar way by rotating 111 connected-line segments about the sphere axis for a total of 11 sides.
- ➢ Faceted objects, although continuous, are no longer as smooth as the original curved objects they represent. Increasing the number of facets results in more accurate representation.



(*a*) Cylinder

(*b*) Sphere

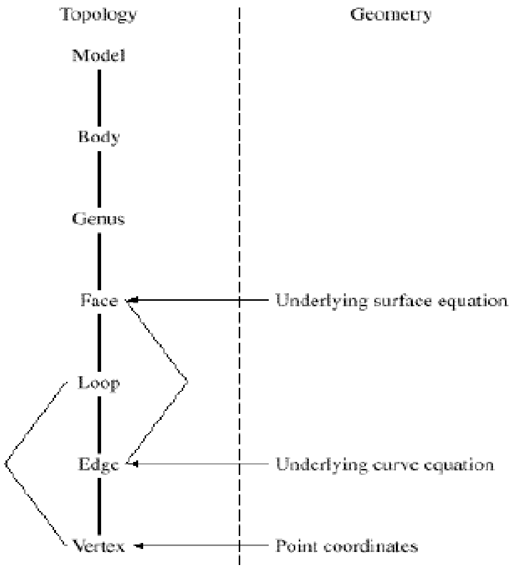
Faceted B-rep of a cylinder and a sphere.

Faced B-rep of a cylinder and a sphere

- A data structure for a boundary model stores both topology and geometry. The structure shown in Figure is based on Eq.
- A relational database model is very effective for implementing such a data structure. Lists for bodies, faces, loops. Edges and vertices are generated and stored in tables. Each line in Figure represents a pointer in the database.

Building Operations

- Equation forms the basis upon which to develop building operations to create B-rep models of complex objects. Euler operators which are based on this equation, are 1101 used in practice.
- CAD systems use B-rep only as an internal representation scheme for solid models. CAD designers create models using either the primitive or the features approach. The CAD software converts users' input to B-rep format before storing it in the model database.



Data structure for B-rep models.

Advantages of b-rep

- Appropriate to construct solid models of unusual shapes
- Relatively simple to convert a b-rep model to wireframe model

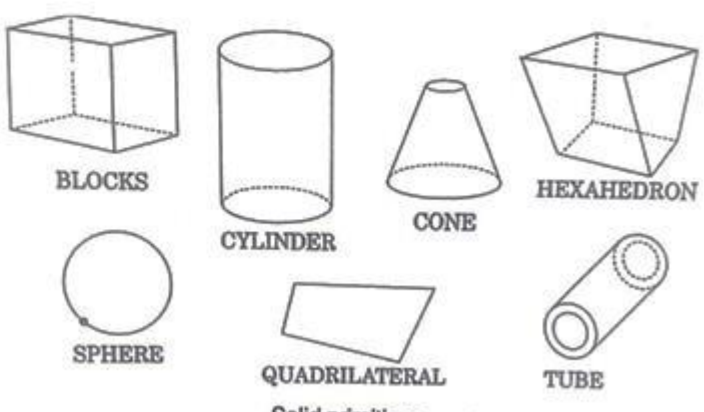
Disadvantages of b-rep

- Requires more storage
- Not suitable for applications like tool path generation
- Slow manipulation

10. Explain brief notes on constructive solid geometry (CSG and C-rep)

- Constructive solid geometry (CSG) (formerly called computational binary solid geometry) is a technique used in solid modeling.
- Constructive solid geometry allows a modeler to create a complex surface or object by using Boolean operators to combine objects.
- Often CSG presents a model or surface that appears visually complex, but is actually little more than cleverly combined or de-combined objects

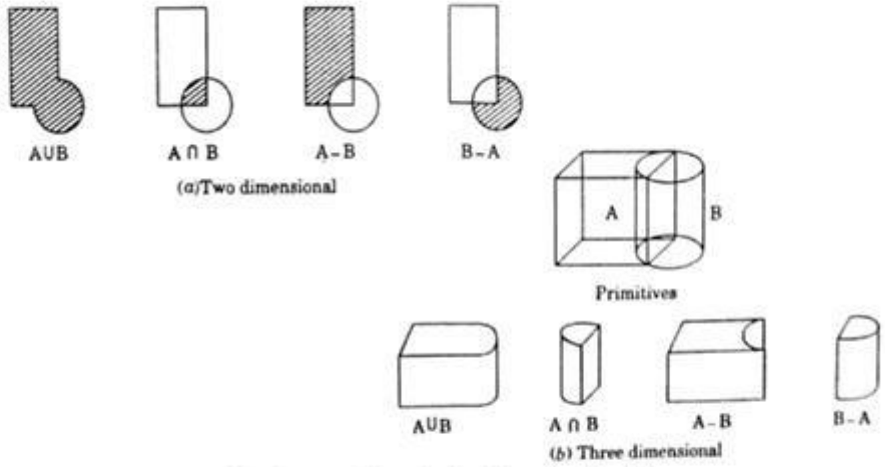
- The simplest solid objects used for the representation are called **primitives**. Typically they are the objects of simple shape:
  - cuboids
  - cylinders
  - prisms
  - pyramids
  - spheres
  - cones



Solid primitives.

- The set of allowable primitives is limited by each software package. Some software packages allow CSG on curved objects while other packages do not
- It is said that on object is constructed from primitives by means of allowable operations which are typically Boolean operations on sets: union, intersection, and difference as well as geometric transformation of those sets.

Boolean operations

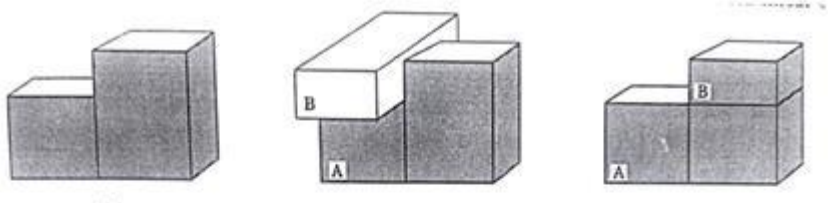


(a) Two dimensional

Primitives

(b) Three dimensional

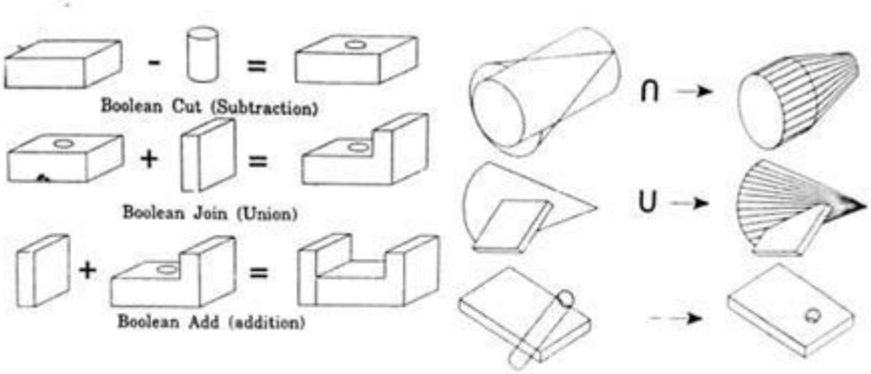
Boolean operations of a block *A* and cylinder *B*.



(a)

(b) A -* B

(c) A U* B

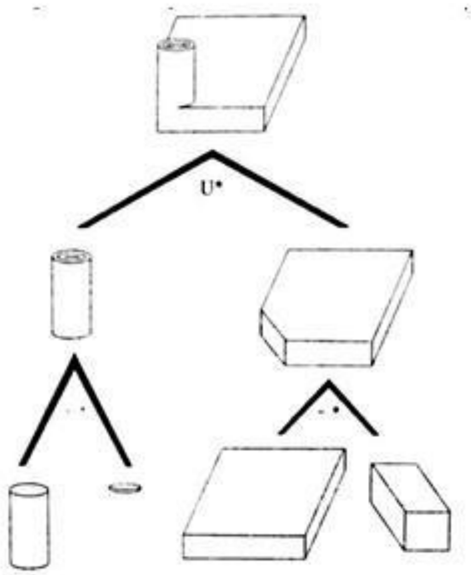


Boolean Cut (Subtraction)

Boolean Join (Union)

Boolean Add (addition)

CSG Tree



An object defined by CSG and its tree.

UNIT- IV

ASSEMBLY OF PARTS

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

1. **Explain about the Assembly modeling of parts in detail. (Nov 2015)**

It is a technology and method used by CAD systems to handle multiple files which represent the components within a product.

A realistic assembly procedure can increase efficiency, cost reduction and improve the recycling of product.

To overcome these problems, various simulations based on digital mock-ups of products are required.

Even though modeling and analysis software presently applied at various stages of the product development process.

As the contact between objects is at the basis of the assembly simulations need 3D objects shapes, the contact detection is address adhere as the first step in the Assembly simulation process.

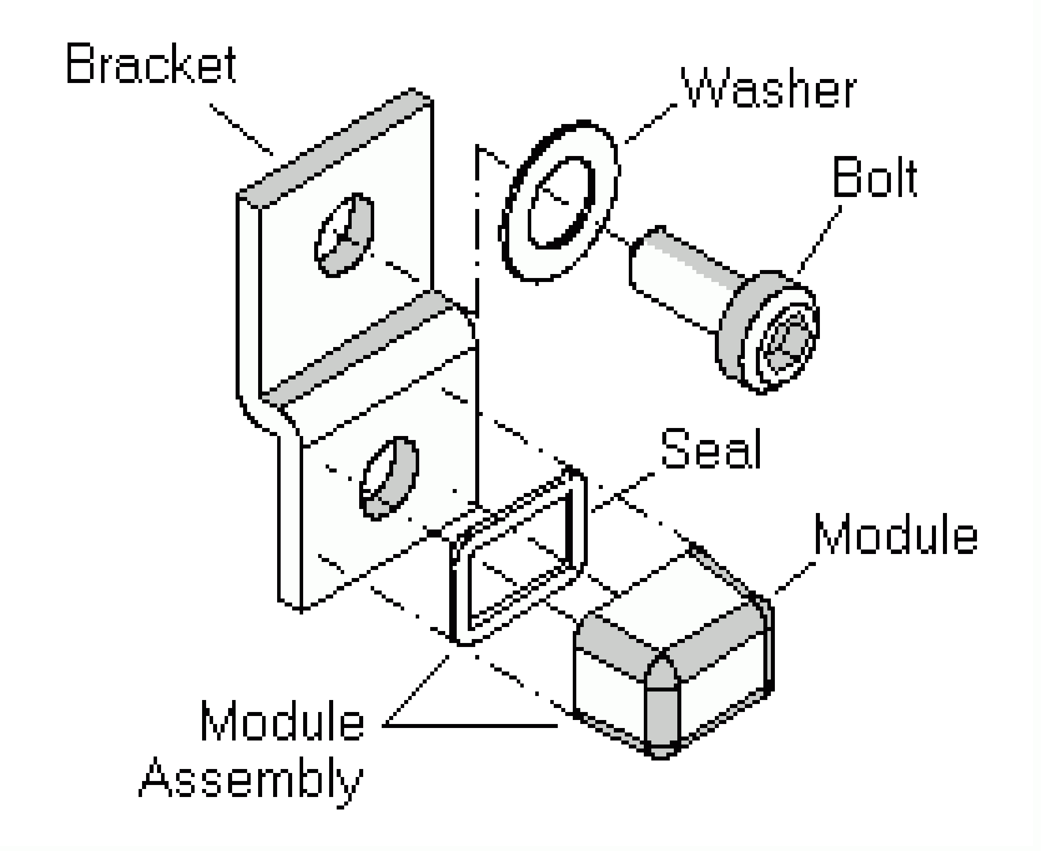
The equivalent procedure establishes links between shapes, contact mock-ups and component kinematics, which gives a basic set of meaningful data

All mechanical parts are applying one of the common CAD modelers.

Thus, the existing assembly modules of 3D CAD software and their definite method to modeling assemblies have a tough influence on how products are calculated.

Also, for the realistic simulation, the data exchange CAD to Virtual Reality is one of the significant problems presently faced by the virtual prototyping community.

Assembly models have additional data than simply the sum of their components with assembly modeling interference verifies between parts and assembly specific data such as mass properties.



Assembly of parts

In most engineering designs, the product of interest is a composition of parts, formed into an assembly.

Parts and/or subassemblies of a given product can be modeled separately, most often by different members of the design team, on a CAD/CAM system.

Instances (copies) of the part can then be merged into a base part or a host to generate the assembly model.

1. These systems provide users with an assembly mode to create assemblies. (Other modes are part and drawing modes.)
2. After a designer creates all the individual parts of an assembly, the designer uses them to create the assembly.
3. Assembly modeling raises two modeling issues that do not exist at the part modeling level Hierarchy and Mating.

Hierarchy: Individual parts and subassemblies must be assembled in the right hierarchy (sequence).

Which is captured (stored) in an assembly tree for each assembly or product. The assembly tree may not be unique, as there may be more than one sequence to create the same assembly.

Mating: Mating conditions are used to determine the mating (Spatial relationships and orientations) between the assembly parts.

For example, the axes of a shaft and a hole may have to be lined up, in which case a concentric mating condition is required.

Two faces may have to be planar, in which case a planar (coplanar) condition is used.

Assembly:

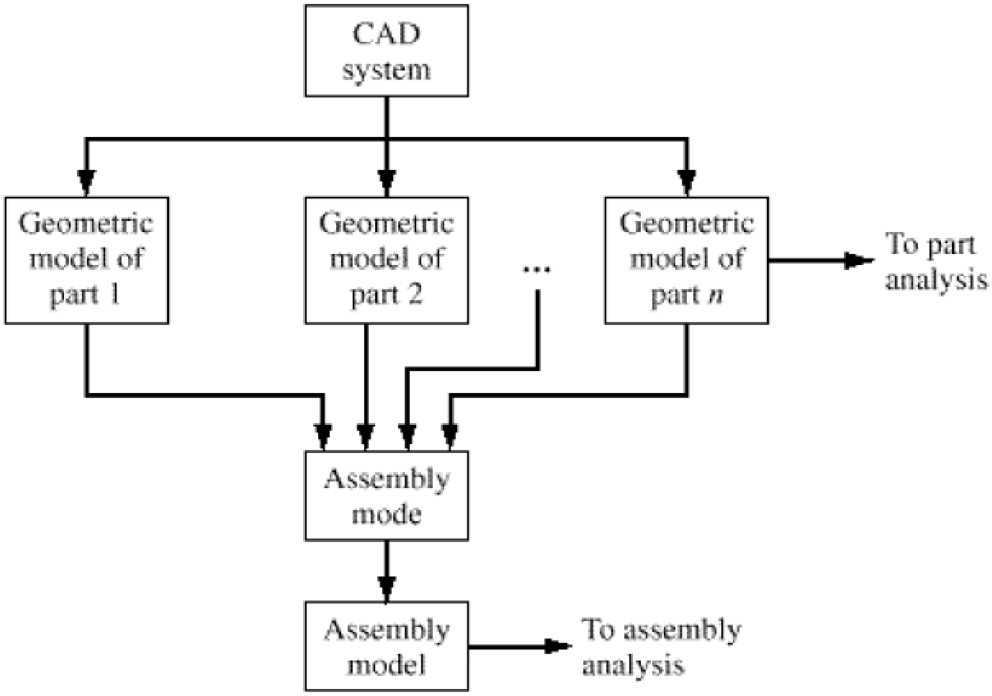
An assembly is a collection of independent parts, it is important to know the nature and the structure of dependencies between parts in an assembly to be able to model the assembly properly.

For example, whether a part can be moved and which other parts will move with it.

The assembly model must include the spatial positions and hierarchical relationships among the parts, and the assembly (attachment) relationships (mating conditions) between parts.

Figure shows how an assembly model can be created using a CAD system.

- Designers first create the individual parts.
- They can also analyze the parts separately.


CAD system
Geometric model of part 1
Geometric model of part 2
...
Geometric model of part *n*
To part analysis
Assembly mode
Assembly model
To assembly analysis

Creating an assembly model.

Once the parts design is complete, designers can proceed to create the assembly and analyze it.

Creating the assembly from its parts requires specifying the spatial and mating relationships between the parts.

Assembly analysis may include interference checking, mass properties, kinematic and dynamic analysis, and finite element analysis.

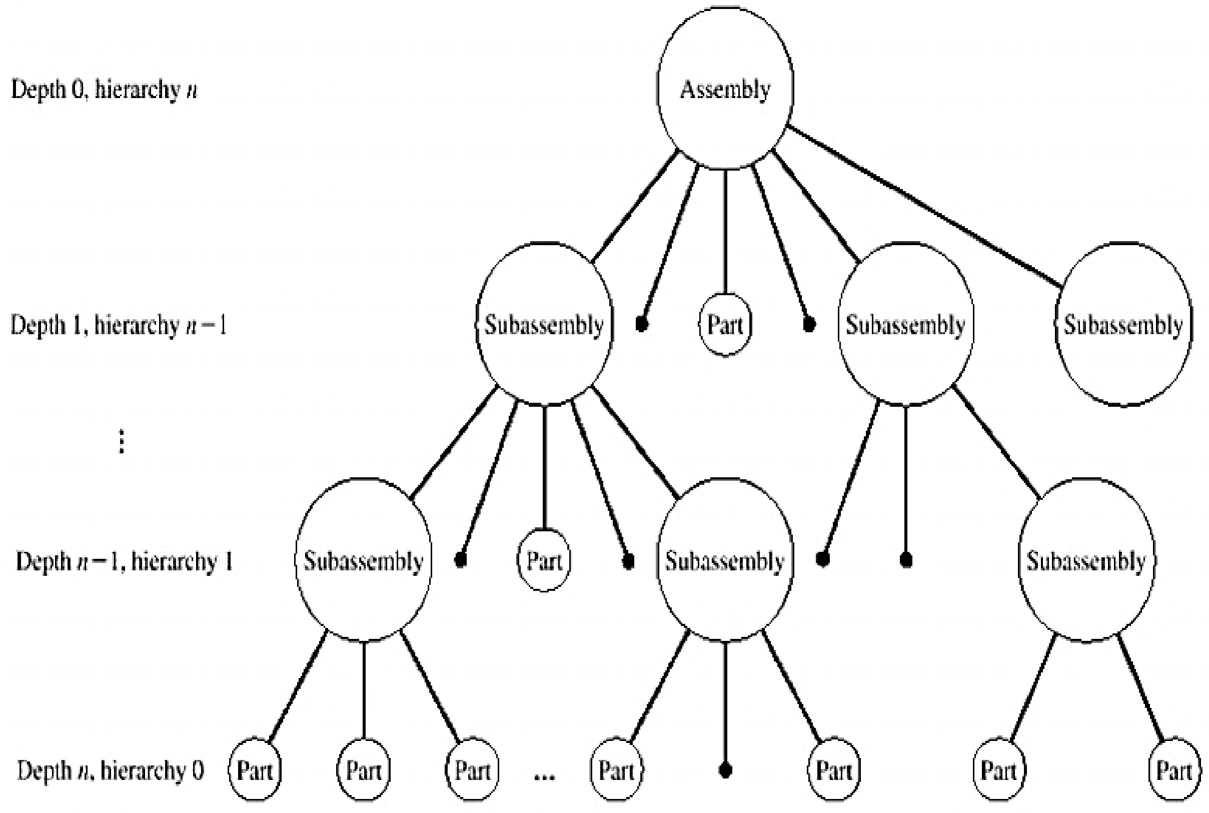
CAD systems establish a link between an assembly and its individual parts such that designers need only change individual parts for design modification, and the system updates the assembly model automatically.

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.

Assembly tree

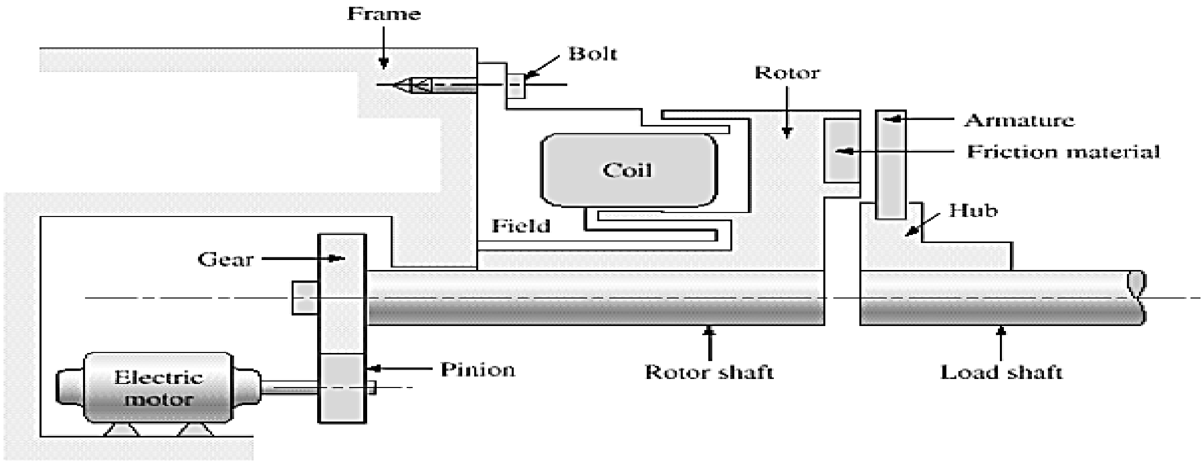
Example: The clutch consists of three main elements: the field, the rotor and the armature.

The field and the coil are held stationary, and the rotor is driven by the electric motor through the gear, pinion, and rotor shaft.

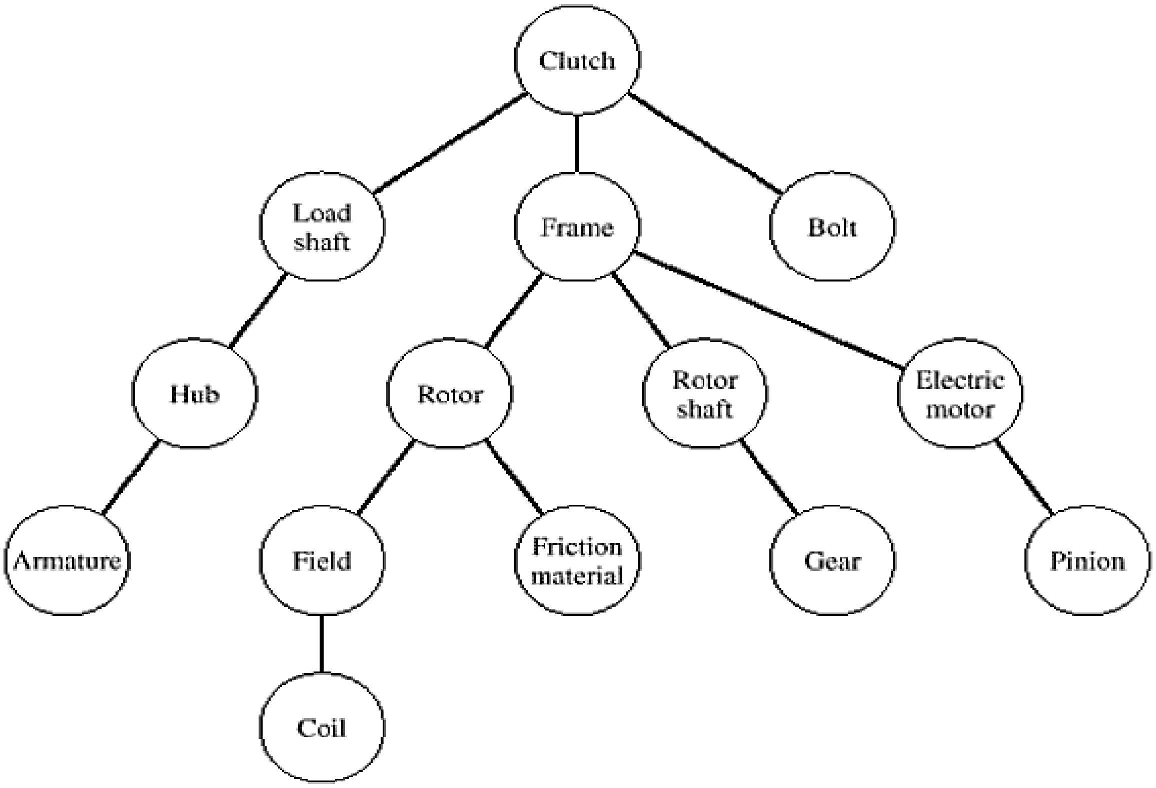
The armature is attached to the load shaft through the hub. The load shaft carries the load (not shown) to be overcome by the clutch.

The assembly tree for this clutch is shown in Figure. The tree represents an assembly sequence by which the clutch assembly can be produced.

The assembly tree is not unique; other valid assembly sequences can be generated.



Electric clutch assembly.



Electric clutch assembly.

Assembly Planning

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**.

This means that the relations between the assembly parts should be easy to change and update.

When a designer changes some of the assembly parameters, the others should update accordingly.

Before we begin to create an assembly, we should consider the following issues:

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

Assembly components are individual parts or entire subassemblies that, in turn, consist of other subassemblies and/or parts, as in assembly tree.

Identifying these dependencies helps us to determine the best approach to create the assembly: bottom-up, top-down, or both.

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

These dependencies include symmetry and geometric arrays (patterns).

Understanding these dependencies helps us to optimize the creation of assemblies.

3. Analyze the order of assembling the parts.

This order determines the ease of the assembly process on the shop floor. It also determines the cost of creating the assembly.

The order also affects the manufacturing processes of the parts.

For example, if an assembly requires a part with a flat, blind hole, the hole must be milled with a flat-end mill-cutting tool and cannot be drilled.

2. **Explain the Mating Conditions in details.**

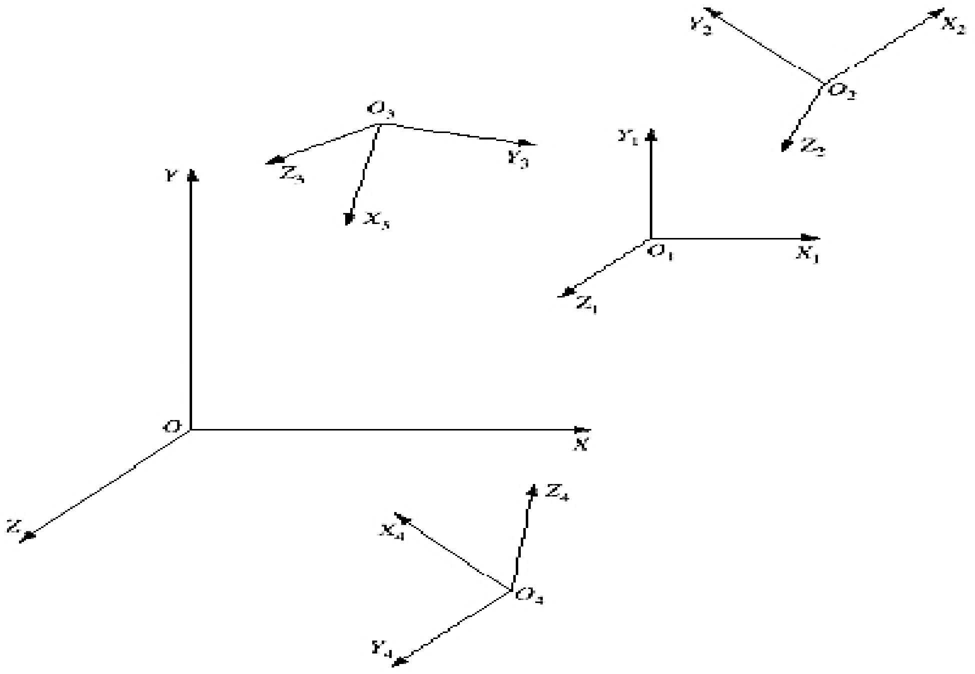
Mating Conditions

Individual parts of an assembly are usually created separately using a CAD/CAM system and then merged (assembled) together, using a *merge* or *insert* command, to form the assembly.

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 become 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.



Assembling parts.

Figure shows an example; The XYZ is the global coordinate system of the database of the assembly model.

Its origin 0 is the (0, 0, 0) point. The X_1 Y_1 Z_1, X_2 Y_2 Z_2, X_3 Y_3 Z_3 and $X_4Y_4Z_4$ are local coordinate systems of four parts that make the assembly.

Their origins O_1, O_2, O_3, and O_4 are located properly relative to the assembly origin O, and their orientations relative to the XYZ coordinate system reflect the proper orientations of the parts in their assembly.

Mating conditions can be provided interactively with ease because they use simple geometric entities such as faces and centerlines.

Example:

a mating condition can consist of planar faces butting up against one another, or requiring centerline of individual parts to be collinear.

The most common mating conditions are coincident, concentric, tangent, coplanar, parallel faces, and perpendicular faces.

When using mating conditions to assemble two parts, keep in mind that there are six degrees of freedom in E^3.

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.

Mating conditions between two parts may require, for example, coincident condition of two faces and two points on the faces.

If we mate the two faces using the coincident condition only, they can still rotate relative to each other.

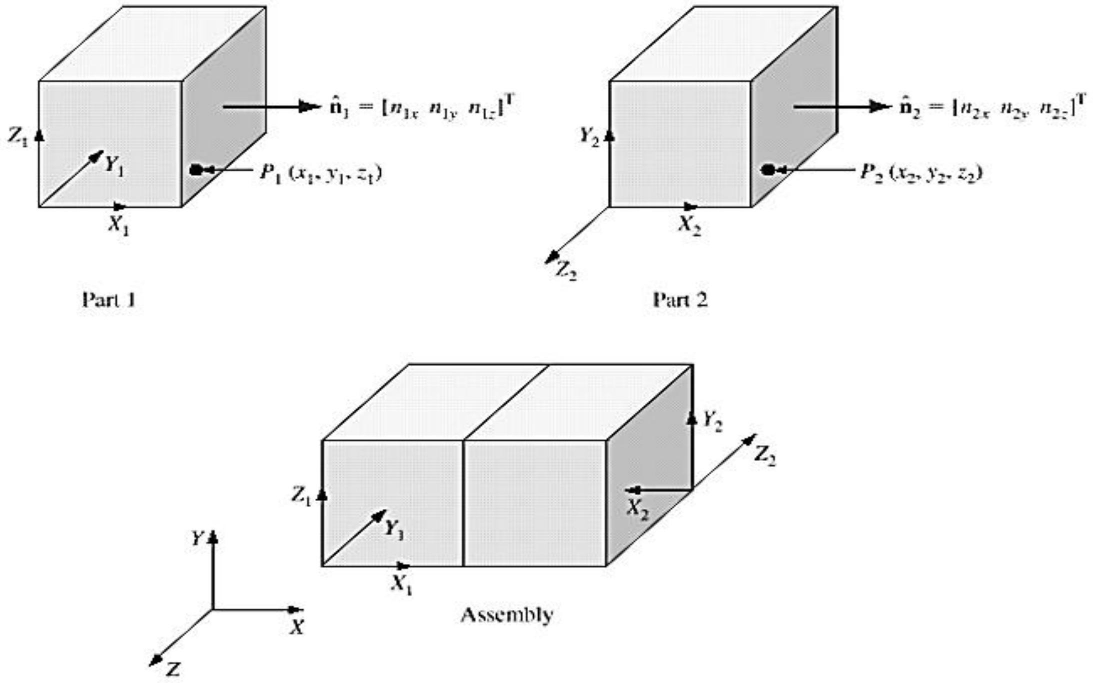
Coincident:

The coincident mating condition is applied between two planar faces, or between a planar face and a cylindrical face (shaft).

Part 1 and Part 2 have the MCSs $X_1Y_1Z_1$ and $X_2Y_2Z_2$, respectively. The planar face of Part1 is specified by the unit normal $n_1^{\wedge}$ and by the point P_1 with respect to the $X_1Y_1Z_1$ coordinate system.

Similarly the planar face of Part 2 is specified by n2$^{\wedge}$ and P_2 with respect to the $X_2Y_2Z_2$ coordinate system.

The coincident condition is satisfied by forcing n1$^{\wedge}$ and $n_2^{\wedge}$ to be opposite each other, and the two faces touch each other such that $P_1$and P_2 are coincident.



Coincident mating condition

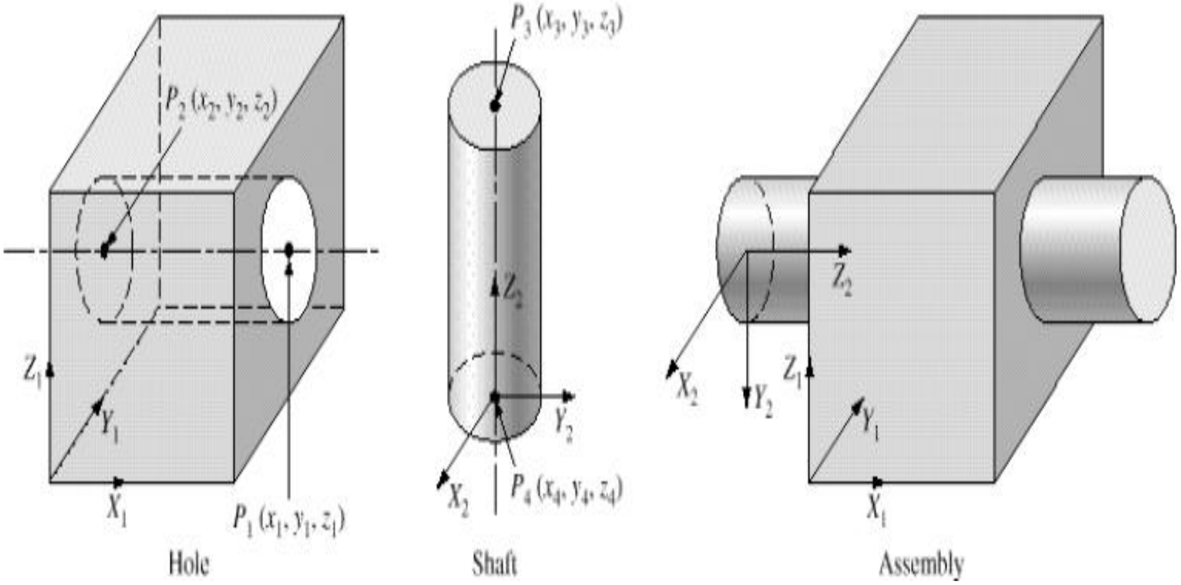
Concentric:

The concentric mating condition holds between two cylindrical faces: a shaft cylindrical face and a hole cylindrical face, as shown in Figure.

The concentric 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 $P_1$and P_2 defined with respect to the $X_1Y_1Z_1$ MCS.

Similarly, the shaft axis is specified by the two points $P_3$and $P_4$with respect to the $X_2Y_2Z_2$ MCS.



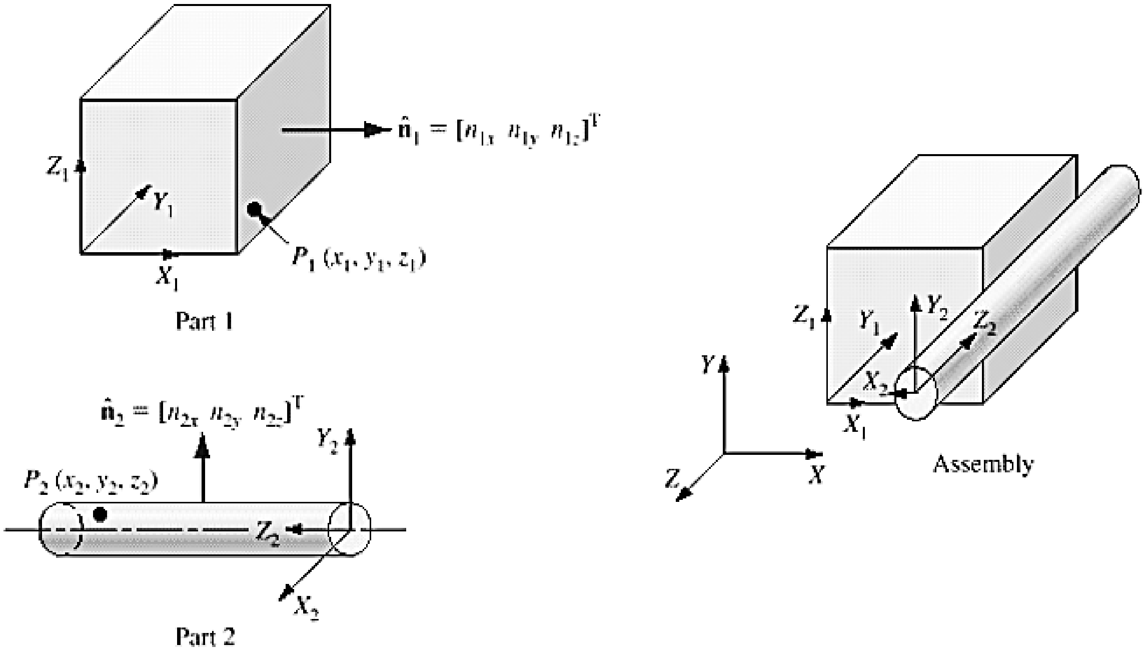
Concentric mating condition

Tangent:

The tangent mating condition is applicable between two planar/cylindrical or cylindrical/cylindrical faces, as shown in Figure.

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 use at least one cylindrical face. While the latter uses two planar (flat) faces.



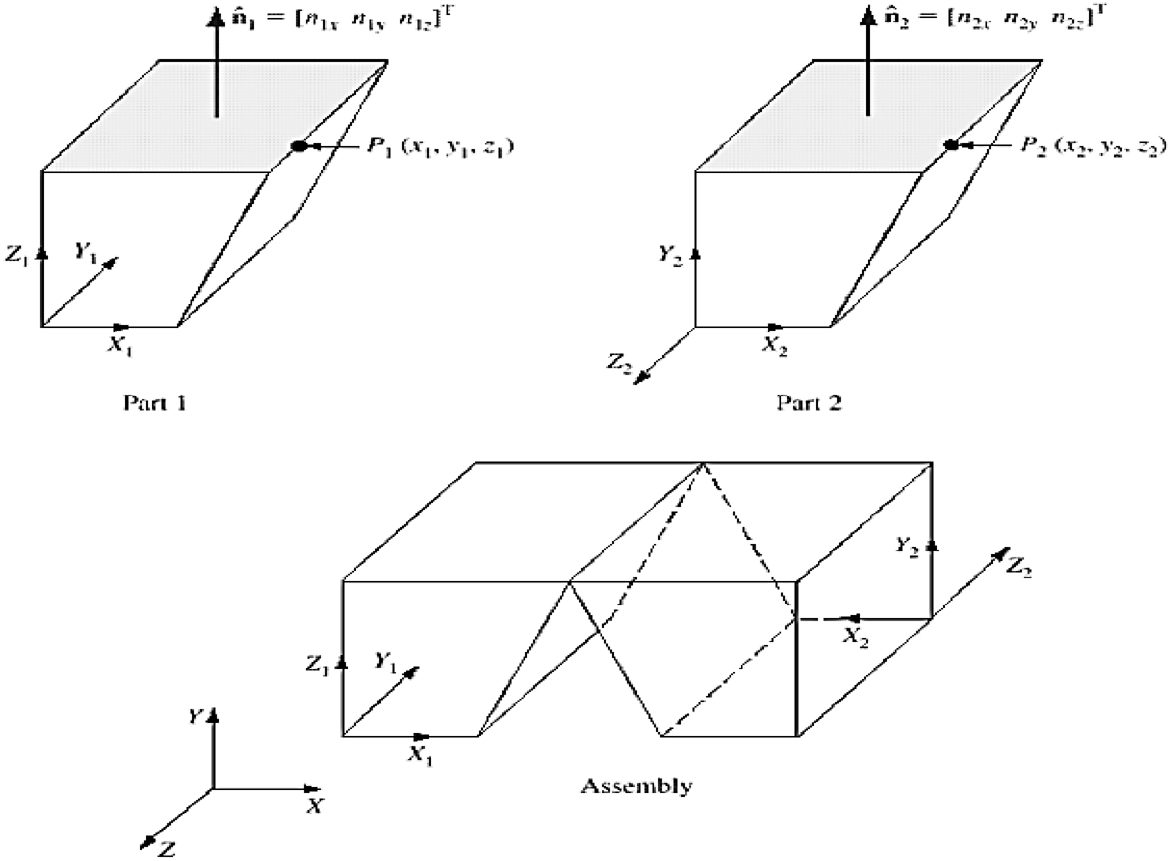
Coplanar:

The coplanar mating condition holds between two planar faces when they lie in the same plane. This condition is illustrated in Figure.

It is similar to the coincident condition except that the points $P_1$and $P_2$are chosen to lie on the two edges to mate.

The coplanar condition is the complement (opposite) of the coincident condition and is satisfied by forcing the two normal's $n_1^{\wedge}$ and $n_2^{\wedge}$to be in the same direction.

The coincident, concentric, tangent, and coplanar mating conditions as described allow both rotational and translational freedom of movement between the mating parts



Part 1

Part 2

Assembly

Coplanar mating condition

The parallel faces mating condition is similar to the coincident mating condition except that the two mating faces are not in contact with one another.

This condition is achieved by forcing the normal of the two faces to be parallel and opposite.

If we separate the two mating faces shown in coincident Fig, we have a parallel faces mating condition.

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

The two shaded faces of Part 1 and Part 2 shown in tangent mating condition fig are made perpendicular to each other by forcing their normal's n_1^and n_2^ to be perpendicular.

3. Describe Bottom Up and Top Down assembly design with example.(Nov 2016,May 2017)

Bottom-Up Assembly Approach

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

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 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.

When we insert parts into the assembly, we insert copies of the parts. These copies are known as instances.

We can use multiple instances of any part if the assembly requires it.

The CAD software maintains a link (known as a pointer in database and programming jargon between each instance and its original part.

If we change the original part, we can change all its instances in an assembly by simply updating the assembly.

These links (assembly links) between an assembly and its individual parts is the most important fundamental concept behind assembly modeling.

As a matter of fact, an assembly link is bidirectional; we change the part and update the assembly, or we change the instance in the assembly and update the part.

In a ‘bottom up’ assembly design, complex assemblies are divided into minor subassemblies and parts.

Every part is considered as individual part by one or more designers. The parts can be archived in a **library** in one or more 3D Files.

This is the high effective way to generate and manage complex assemblies. Every part is included into the active part making a component request and thus an assembly.

The component will be the child of the active part and the nit will be the active part.

Hence an instance of the actual part is applied; it revises automatically if the archived part is edited by activating.

Bottom up Hierarchy:

The bottom up ‘assembly design hierarchy of the basic assembly is shown in figure. All the parts exist prior to **Part1**.

When **Part1**is generated, it becomes the active. It would utilize the menu sequence to add **Bracket** and it becomes the active part.

Insert> Component

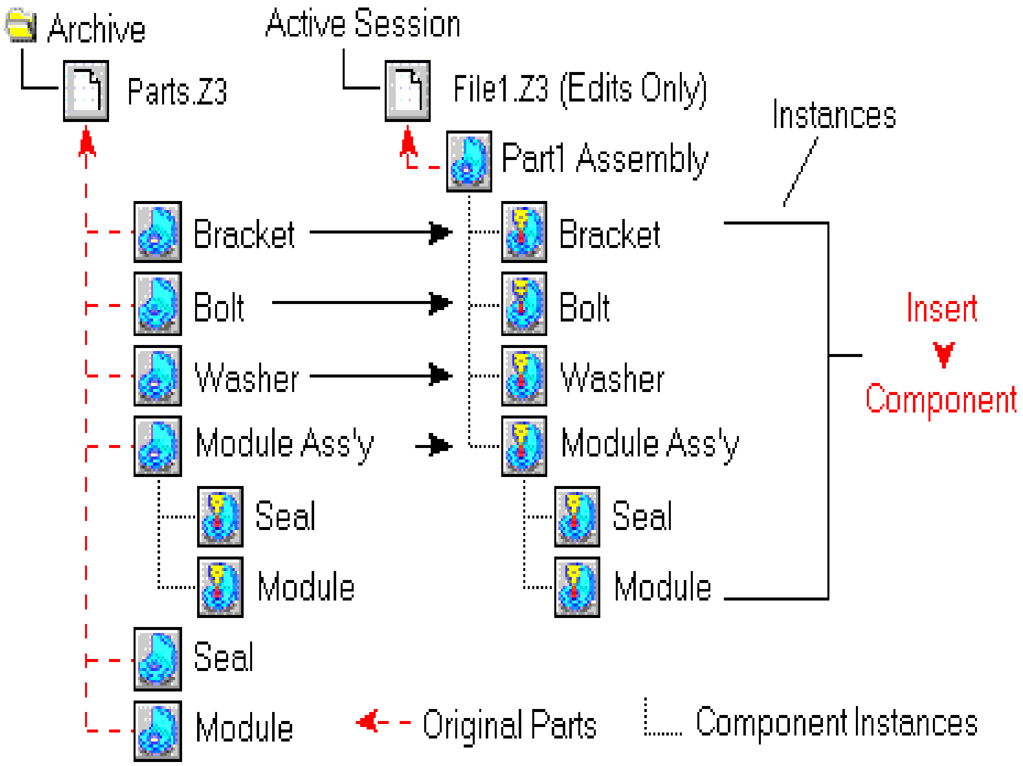
Or

Assembly Design Tool Bar

As per example shown in figure, 'Bracket' is a child of **Part1.** The dashed line represents that 'Bracket' exists in the 3D file **Parts Z3**.

The dotted line represents that '**Bracket'** is inserted into **Part-1**. After **Bracket** is added, **Part1** is **redefined**.

Bolt and **Washer** are then added the same process and **Part-1** is **reactivated** again.



Bottom up Design–Part1

Module of subassembly is added similar as '**Bracket', 'Bolt',** and '**Washer'** again becoming a child of Part-**1**.

But because **Module Subassembly** already has the two items **Seal** and **Module**, they are added and continue as its children.

Sequence of operations (Fig.):

- **File-1** has 1part.
- **Part-1** has4 components.
- **Module Subassembly** has2 components.
- All of the items are illustrations of the original parts that reside in the ZW 3D file **Parts Z3.**
- If**File-1**is eliminated from the active assembly before it is saved and **Part1** are removed. The original parts placed in the file **Parts Z3** are not changed.
- If**File-1**is **saved** and **Part1** is also saved.
- If**File-1**is erased and **Part1** is also erased.

The advantages of bottom-up approach:

It is the preferred technique if the parts have already been constructed, as in the case of off-the-shelf parts. It also allows designers to focus on the individual parts.

It also makes it easier and simpler to maintain the relationships and regeneration behavior of parts than in the top-down approach.

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.

It allows a project leader to break up product specifications, assign work teams, and enforce downstream design changes at a high level.

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 supplier.

This tight control allows distributed design teams to work concurrently within a common product framework.

It also allows detailed design to begin while the assembly layout is being finalized.

The top-down assembly approach begins with an assembly layout sketch (also known as assembly sketch or skeleton model).

The layout serves as the behind-the-scenes backbone of the assembly. The layout defines components in the context of an assembly.

The process of creating an assembly using the top-down approach is as follows:

1. Log in and start the CAD/CAM system.

2. Select the assembly mode.

3. Open a new assembly file.

4. Create a sketch in which various entities represent components (parts or subassemblies) in the assembly.

Indicate a tentative location for each component. Capturing the overall design intent of the assembly.

5. Create more sketches if needed. A complex assembly may require multiple sketches, similar to creating complex features of individual parts.

For example, we may use the Front, Top and Right sketch planes to create the assembly layout.

6. Use the sketch to define the component size. Skeletal shape, and location within the assembly.

7. Edit the assembly sketch as needed until the assembly layout is finalized. For example, change the locations or the sizes of some components.

8. Create the individual parts and save them in their respective files. Make sure that each part references the assembly sketch.

9. Evaluate the assembly after the parts are fully constructed, if needed. Modify the assembly sketch. And update the assembly and parts. Such an update is an automatic process.

10. Save the assembly file and exit the CAD system.

In a ‘top down’ assembly design all parts are classically designed by the similar person within a single part.

3D assembly handles ‘top down’ method by allowing to **design and creation of** a component while working the active part.

Hence, the active part will be an assembly part. The part becomes a child of the active part and then it will be the active part.

The part, when generated, is an instance of a base part which will be a root object located in the active file.

Every part is activated and modified as needed. The ‘top down’ assembly design has its benefits.

If the project is terminated or to go in a different new direction, removing the file will remove the part and all of its components.

Top down Hierarchy

The 'top down' assembly method is shown in a figure and one of the components exist prior to**Part-1**.

When **Part-1**isgenerated, it will be the active part. The following command sequence to generate Bracket and create it the active part.

Assembly Design Toolbar:

Bracket is a child of **Part-1.** The dashed line illustrates that by default when **Bracket** is generated; it is attached to**File-1**.

The dotted line illustrates that **Bracket** is attachedinto**Part-1**. When **Bracket** is executed **Part1** is **reactivated**.

Bolt and **Washer** are then generated using the similar process and **Part-1** is **reactivated** again.

Subassembly Module is generated like the **Bracket, Bolt,** and **Washer** again will be a child of **Part1.** But, **Module Subassembly** remains active when **seal** is developed.

Seal will be the active part and by defaulted so exists in **File-1** but is inserted into **Module Subassembly** hence it was active at the time of seal was created.

Subassembly Module is then **reactivated** and **Module** is generated like a **Seal.**

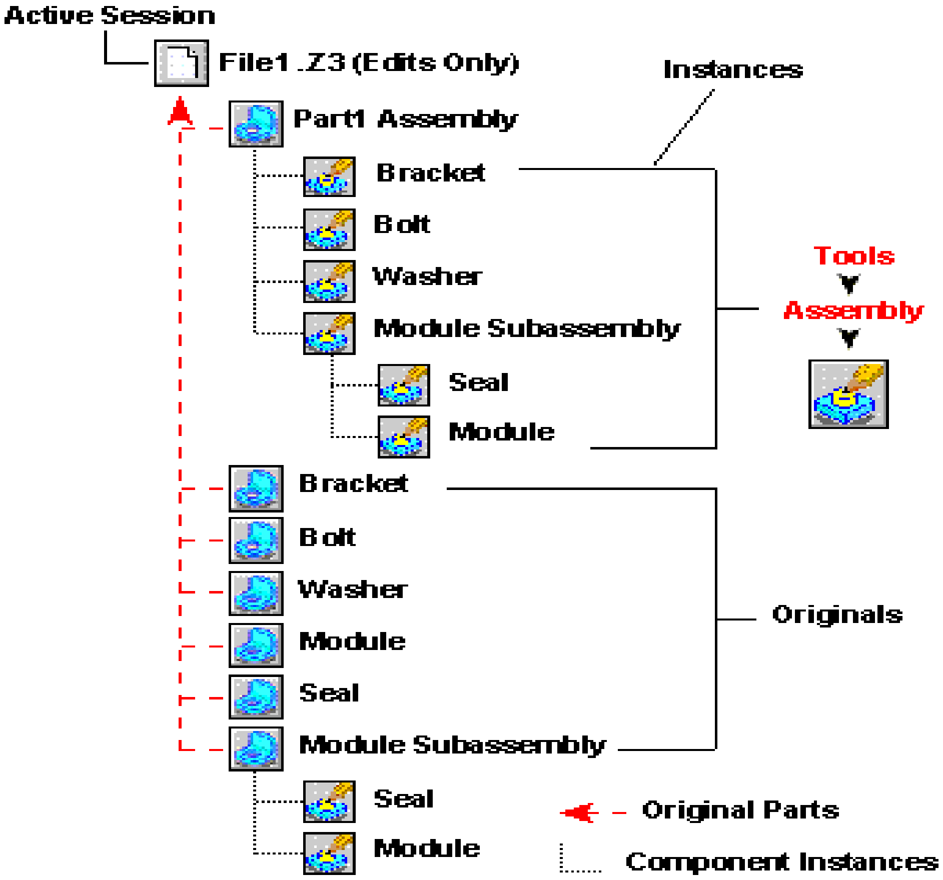
Sequence of operations:

Part-1 contains 4 components, which are illustrations of the basic parts located in File-1.

Subassembly Module contains 2 components which are also illustrations of the basic parts located in File-1.

If File-1 is saved it has all of its original parts.

If File-1 is erased, it and all of its basic parts are erased.



Top down Design– Part1

The advantages of top-down approach:

The major advantage is that if we change the layout sketch, the assembly and its parts are automatically updated upon exiting the sketch.

We make all the changes quickly in one place, the assembly layout sketch.

Testing Mating Conditions

Mating conditions are the key to the successful creation of assemblies. Components could over- or under-mate.

An under-mated component becomes a floating component in the assembly modeling pace.

The component can only move along its current free degrees of freedom. For example, if the translation or rotation about the X axis is not constrained, the component moves or rotates along this axis.

Both the animation files and exploded views require moving the parts in space along their free degrees of freedom.

Finally, testing the mating conditions provides the opportunity to ensure that an assembly functions as it supposed to according to its design.

Managing Assemblies

Afterward insert components into an assembly; we can perform many operations to manage them.

We can hide, freeze (suppress), copy or delete them.

When hiding a component, we make it invisible from the display screen; however, it still gets updated when we update the assembly model.

Freezing a component does the opposite to hiding it: when we freeze it, it is still visible on the screen, but it is not updated.

Copying components is useful for building symmetric assemblies.

A manager allows a designer to assign mating conditions. Group them, Copy them, or delete them.

For example, if a designer drops a shaft on a hole. The CAD system can display a concentric mate symbol to guide the designer to use the correct mate.

Working with Subassemblies

When an assembly is a component of another assembly, it is known as a subassembly.

We can nest subassemblies in multiple levels in the assembly tree to reflect the product hierarchy.

There are three alternatives for creating subassemblies using CAD systems:

Create an assembly document (file) as a separate operation. Then insert it as a component into a higher-level assembly model.

This makes it a subassembly of the higher-level assembly.

Insert a new empty subassembly at any hierarchical level of an open assembly model (file), and then add components to it.

Form a subassembly by selecting a group of existing components from an assembly.

As we develop assemblies, we may have a need to edit their subassemblies. We can work with subassemblies in a variety of ways

- Dissolve a subassembly into individual components.
- Edit the subassembly structure by moving its components up or down the assembly tree or to a different tree branch.

- Simply drag the components from the assembly tree manager and drop where appropriate in other tree locations (levels).
- Move components in and out of subassemblies to the parent assembly.

4. Explain the Inference of Position and Orientation in details.

The inference of the position (location) and orientation of a pan in an assembly from mating conditions requires computing its 4x4 homogeneous transformation matrix from these conditions.

This matrix relates the part's local coordinate system (pan MCS) to the assembly's global coordinate system (assembly MCS).

The orientation is represented by the rotation matrix between the two systems. The transformation matrix can be written as

$$[T] = \left(\begin{array}{ccc:c} m_x & q_y & r_z & x \\ m_y & q_y & r_y & y \\ m_z & q_y & r_z & z \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

This matrix has 12 variables (nine rotational and three translation elements) that must be determined from the mating conditions.

For an assembly of N pans, and choosing one of them a host, N-1 transformation matrices have to be computed.

Therefore, the variables to solve for simultaneously are the 12(N – 1) elements of these matrices.

WCS Method

The simplest method for specifying the location and orientation of each part in an assembly is to provide the 4x4 homogeneous transformation matrix [T] directly, instead of inferring it from mating conditions.

The matrix transforms the coordinates of the geometric entities of the part from its MCS to the assembly MCS.

One way for the user to provide the transformation matrix interactively is by defining a WCS in the assembly model such that its origin and orientation match the final location and orientation of the MCS of the part that we need to insert into the assembly.

We then write the [T] matrix, that converts the coordinate from the part MCS (now the WCS) to the assembly MCS.

When we apply [T] to all the part geometry, we effectively insert (merge) the part into the assembly.

Designers and manufacturers should check jointly that a provided product can be assembled, without interference between parts, before the product to be manufactured. Similarly all the CAD tools presently have the potential to directly analyze the possibility of a specified assembly plan for a product.

An assessment of previous assembly sequence and optimization research explains that most previous assembly planners apply either feature-mating or inter ference-free techniques to find assembly part interference interaction.

In both feature-mating and interference-free techniques focused upon the basic geometrical data and restrictions for the designed product, which are generally contained in connected CAD files.

When completely automate the procedure of creating a professional assembly plan, geometrical information for CAD models should be automatically taken from CAD files, analyzed for interference relationships between components in the assembly and then designed for utilized the assembly analysis tools.

Most of the previous assembly sequence planners do not have the potential to complete the three tasks; they need users to manually input part attributes or interference data, which is so time-consuming.

Determining Interference Relationships between Parts

In automated assembly schemes, most parts are assembled along with the principal axis.

Hence, to fine interference between parts while assembly, the projected technique referred six assembly directions along with the principal assembly axis: +*x*, -*x*, +*y*, -*y*, +*z*, and –*z* .But the method could be improved, to think other assembly directions, as required.

The projected system uses projection of part coordinates onto planes in three principal axis (x,y,z) to find the obstruction between parts sliding along some of the six principal assembly axis.

The projections overlap between any two parts in a specified axis direction shows a potential interference between the two parts, when one of the two parts slides along the specified direction, with respect to the other.

Vertex coordinates for overlapped projections are then evaluated to find if real collisions would happen between parts with overlapped projections.

The planned process stores the determined interference data for allocated assembly direction in a group of interference free matrices, for compatibility with previous planners of assembly.

The swept volume interference and the multiple interference detection systems are appropriate for three-dimensional interference determination between B-REP entities.

But, both techniques were developed for real-time interference detection between two moving parts in a simulation environment. As a result, these two techniques are expensive in computationally.

Forth reassembly planning issue, actual collision in finding capacity along subjective relative motion vectors is not require.

Instead an efficient computational technique is required for finding if two parts will collide when they are assembled in a specified order along any one of the six principle assembly axis.

Mate Method

In a typical assembly, the mating conditions between two components are not enough by themselves to completely constraint the two components.

The mating conditions along with the properties of [T] provide the necessary equations to solve for the 12 (N – 1) variables.

The number of equations is always equal to or greater than the number of variables.

Therefore, the method of solution must account for the number of redundant equations, and it must eliminate these equations from the system of equations to be solved.

The three basic mating conditions: coincident, concentric, and coplanar.

For the coincident condition Figure, each face where the two parts mate is specified by a unit normal and a point described in the MCS of its corresponding part. Let [T_1] and [T_2] be the transformation matrices from the $X_1Y_1Z_1$and $X_2Y_2Z_2$ coordinate systems, respectively, to the MCS of the assembly.

The unit normal and the two points specifying the mating conditions can be expressed in terms of the MCS (XYZ) system as follows.

$$\begin{pmatrix} n_{1x}^{a} \\ n_{1y}^{a} \\ n_{1z}^{a} \\ 0 \end{pmatrix} = [T_1] \begin{pmatrix} n_{1x} \\ n_{1x} \\ n_{1x} \\ 0 \end{pmatrix}$$

In the preceding equations, the superscript a indicate assembly.

The coincident condition requires that the directions of the two unit normal's use be equal and opposite, and the two points muse lie in the same plane al which the two faces mate.

The coincident condition requires four equations that can be expressed as follows:

$$n_{1x}^{a} = -n_{2x}^{a}$$

$$n_{1y}^{a} = -n_{2y}^{a}$$

$$n_{1Z}^{a} = -n_{2z}^{a}$$

The concentric condition requires the center lines of the shaft and the hole to be collinear mating figure. The equation of the centerline of, say, the hole can be written as:

If the shaft axis is collinear with the hole centerline, points P_3 and P_4defining the axis should satisfy above equation.

These points must first be transformed using [T_2] to the MCS coordinates. The constraint equations required for each concentric condition can be written as:

Above two Equations, yield three combinations of equations resulting in six equations for each concentric condition.

A free rotating part is defined here as a part which rotates freely about a centerline axis.

The rotation of these parts usually does not alter the appearance of the assembly, thus the name free rotation.

There are infinite possible orientations of a free rotating part.

Assembly Analysis

CAD systems provide various tools to analyze assemblies once they are created.

List of assembly analysis activities are:

1. **Generate assembly drawings**.

An assembly drawing is no different from a part drawing. This drawing may include the standard four views or just an isometric view.

The procedure for creating an assembly drawing is the same as a part drawing.

2. **Generate a parts list.**

The parts list is also known as bill of materials (BOM). This list is a table that shows the part names and how many instances of each part are used in the assembly.

The above screenshot shows an example. The BOM is usually inserted in the assembly drawing via a command sequence such as Insert=>Bill Of Materials=> select a template =>Open=> select the default template=> input information. Solid Works uses an Excel sheet as the BOM file.

3. **Generate an exploded view.**

This is an isometric (ISO) view that shows the parts of an assembly displayed apart from each other.

The normal ISO view is known as a collapsed (assembly) view. We may create an exploded view by specifying the direction and distance to which each part will move while exploding.

Parts can be exploded in the directions of their permissible degrees of freedom after applying the mating conditions.

An exploded view may consist of one or more explode steps. Each step shows the assembly component(s) displaced into a new location.

All the steps of the exploded view are stored in the view itself.

Upon displaying the view, the steps are displayed sequentially to give the illusion of animation or assembly sequence.

The explode steps may be edited after they are created, or new steps may be added.

Finally, an exploded view may be collapsed to return it to its assembled state.

4. **Generate sectional views.**

This is similar to generating sectional or other types of views of individual parts.

5. **Perform interference checking.**

Allow you to check if any parts of an assembly pierce each other or not.

If interference is detected between two parts, the CAD system displays the interference volume to allow users to examine and rectify/eliminate it.

The user may also request the CAD system to report overlapping or tangent edges/faces as interference if needed.

Here is an example of interference detection using Solid Works: Click this sequence: Tools=> Interference Detection=>select the components to be checked for interference =>Check.

The system will indicate the interference volume, if any, between the components.

Click the Treat coincidence as interference checkbox if you want coincident entities (faces, edges, or vertices that touch or overlap) to be reported as interferences. Otherwise, touching or overlapping entities are ignored.

6. **Perform collision detection.**

We detect the collision of a component with other components of an assembly, by moving or rotating it along its degrees of freedom.

Collision implies that the component cannot be assembled correctly into the assembly, or it cannot move or rotate freely in its final position in the assembly.

7. **Perform mass property calculations.**

Calculate the mass properties of an entire assembly as calculating for a single part.

The centroid or the inertia properties of an entire assembly or subassembly may be important in some applications.

5. Explain the Mass Property in assembly modelling in details.

Mass property calculation was one of the first engineering applications to be implemented in CAD/CAM systems.

Mass property calculations usually involve evaluating various integrals. Exact evaluation of these integrals is only possible for simple shapes.

For complex shapes, approximate methods are usually used to evaluate these integrals.

These methods have the important property that. They monotonically converge to the exact solution which is, of course, not known.

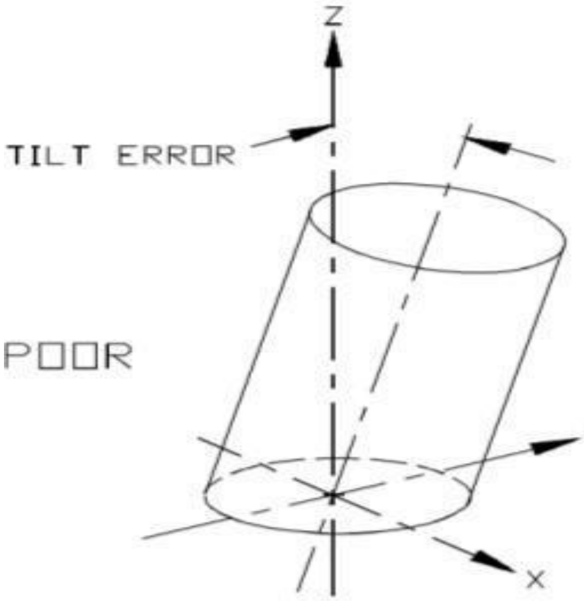
Mass property algorithms that utilize these methods are fully automatic and require no additional input except mass attributes, such as the density of the model.

The first step in finding mass properties is to setup the location of the X,Y and Z axis.

The correctness of the calculations will depend completely on the knowledge used in choosing the axis.

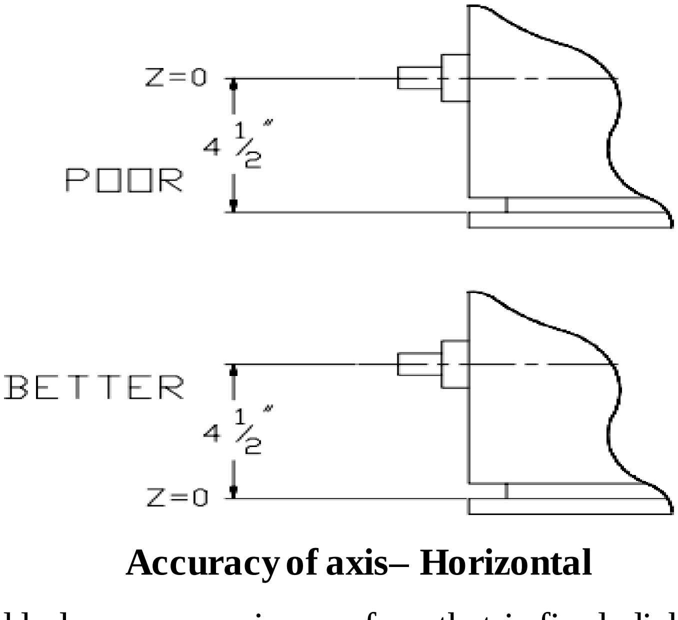
Hypothetically, these axes can be at any position relative to the object being considered, offered the axes are equally perpendicular.

But, in reality, except the axes are chosen to be at a position that can be precisely measured and identified, the calculations are meaningless.



Accuracy of axis– Vertical

As shown in the figure, the axes do not create a best reference hence a small error in squareness of the base of the cylinder origins the object to tilt away from the vertical axis



Accuracy of axis– Horizontal

An axis should always pass via a surface that is firmly linked with the bulk of the component.

As shown in the figure, it would be best to position the origin (Z=0) at the end of the component rather than the fitting that is freely dimensioned virtual to the end.

Center of gravity

The center of gravity of an object is:

- Described the 'centers of mass' of the object.
- The location where the object would balance.
- The single point where the static balance moments are all zero about three mutually perpendicular axis.
- The centroid of object the volume when the object is homogeneous.
- Thepointwherethetotalmassofthecomponentcouldbemeasuredtobeconcentratedwhile static calculations.
- The point about where the component rotates in free space
- The point via the gravity force can be considered to perform
- The point at which an exterior force must be used to create translation of an object in space

Center of gravity location is stated in unit so if length along the three axes (X,Y and Z).

These three components of the vector distance from the base of the coordinate system to the Center of gravity location.

CG of composite masses is computed from moments considered about the origin .The essential dimensions of moment are Force and Distance.

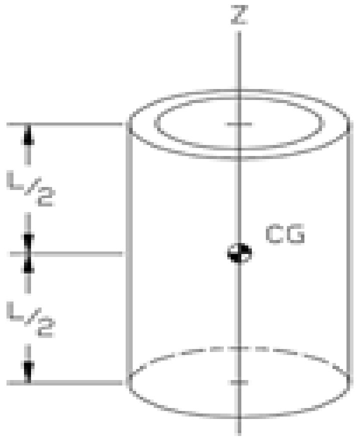
On the other hand, Mass moment may be utilized any units of Mass times Distance.

For homogeneous components, volume moments may also be considered. C are should be taken to be confident that moments for all parts are defined incompatible units.

Component distances for CG position may be either positive or negative, and in reality their polarity based on the reference axis position.

The CG of a homogeneous component is determined by determining the Centroid of its volume.

In practical, the majority of components are not homogeneous, so that the CG must be calculated by adding the offset moments along all of the three axes.



Center of Gravity

Geometric Properties

Geometric properties, specifically length, area. Surface area, and volume; form the basis for mass property calculations.

Curve Length

Calculating the length of a given curve between two end points is useful in many applications.

For example, in mechanism analysis, we might be interested in calculating the length of a given locus in space.

To calculate the length of a spatial curve between two points $P_1$and P_2, consider the curve shown in Figure.

Given an incremental length L of the curve. the curve's total length P_1 and P_2 can be given by the following integral:

$$L=\int_{P1}^{P2} dL$$

Curves are usually represented in a parametric form. If the length element ΔL is bounded by the points P_i and P_{i+1} as shown in Figure.

ΔL can be approximated by the length of the vector connecting the two points, that is

$$\Delta L = |P_{i+1}-P_i| \qquad \text{Or}$$

$$\Delta L = \sqrt{\ } (\Delta x)^2+ (\Delta y)^2 +(\Delta z)^2$$

Dividing both sides by Δu and taking the limit when u approaches zero, we get

$$\lim_{u \quad 0} \frac{L}{u} = \lim_{u \quad 0} \sqrt{\frac{x}{u}^2 + \frac{y}{u}^2 + \frac{z}{u}^2}$$

$$\frac{dL}{du} = \sqrt{\frac{dx}{du}^2 + \frac{dy}{du}^2 + \frac{dz}{du}^2}$$

$$dL = \sqrt{x^2 + y^2 + z^2}\, du$$

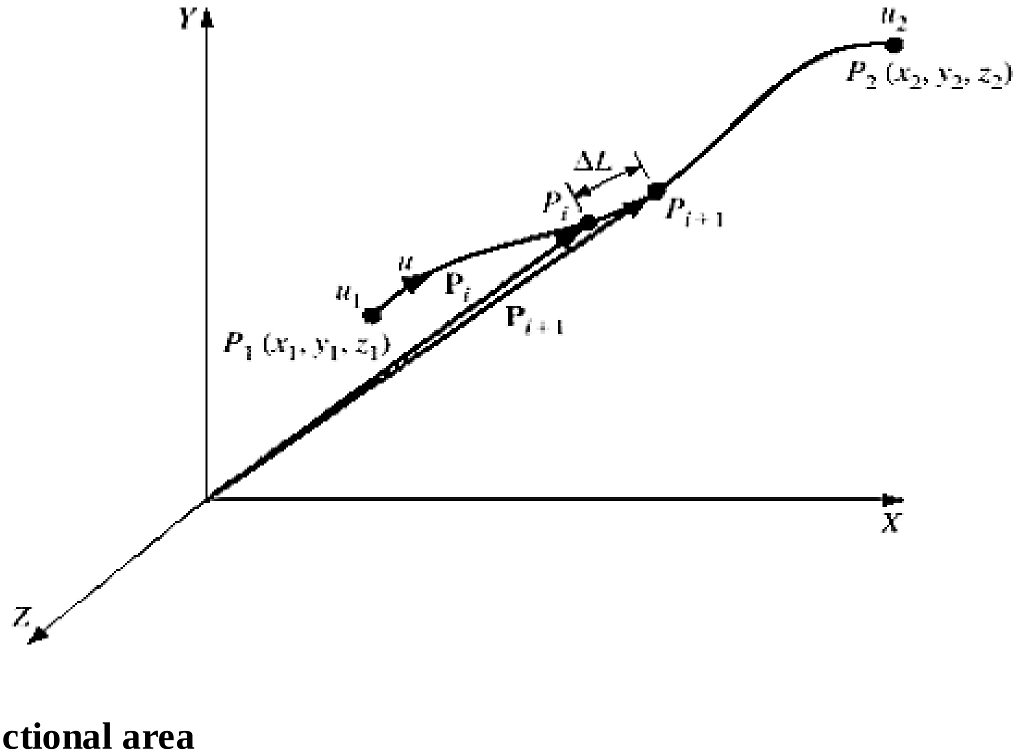
Substituting dL value in initial equation gives,

$$L = {}_{u_1}^{u_2}\sqrt{x^{2}+y^{2}+z^{2}}\,du$$

$$L = {}_{u_1}^{u_2}\sqrt{\mathbf{P}\quad\mathbf{P}}\;du$$

The equation gives the exact length of a curve segment bounded by the parametric values u_1 andu_2 as the integral, with respect to u, of the square root of the dot product of the tangent vector of the curve.

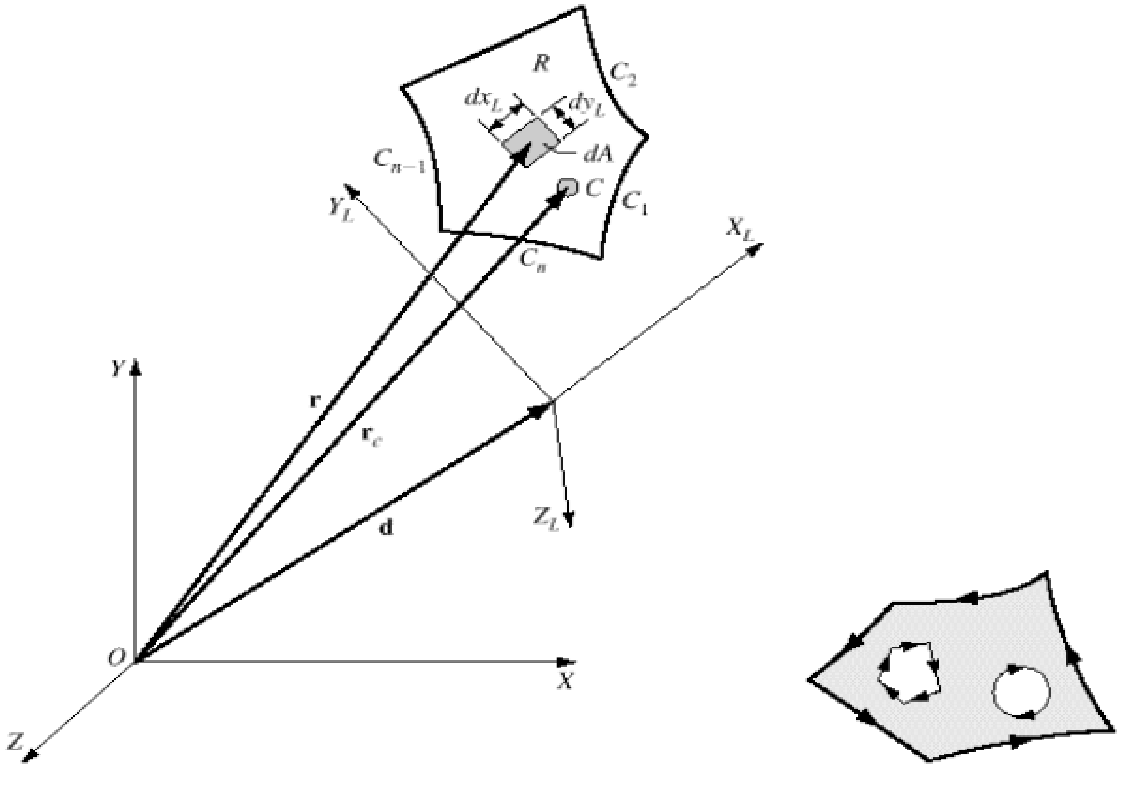
The equation requires that the C^1 continuous. It applies to both open and closed curves.



Cross sectional area

A cross-sectional area is a planar region bounded by a closed boundary. The boundary consists of a set of C^1 continuous curves connected together. Thus, the boundary is piecewise continuous.

There are three properties associated with a planar region: the length of its contour (boundary), its area, and its centroid.



(a) Area orientation

(b) Boundary traversal

Cross-sectional area.

Figure shows a region R that is oriented generally in space. The region's plane coincident with the X_LY_L plane of the $X_L\ Y_L\ Z_L$ local coordinate system.

The region is bounded by the curves C_1, C_2, ….., and C_n, that is,

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

Where L_i is the length of curve C_i

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

$$A=\iint dA= \iint dx_L\, dy_L$$

The evaluation of the integral in above Equation requires mapping this equation from the local coordinate y terms of the region R to the global system (MCS).

The parametric equations describing the boundary curves are stored in the region's geometric database with respect to the latter. The net cross-sectional area of a connected region is given by:

$$A_m=A-\sum_{j=1}^{m} A_h$$

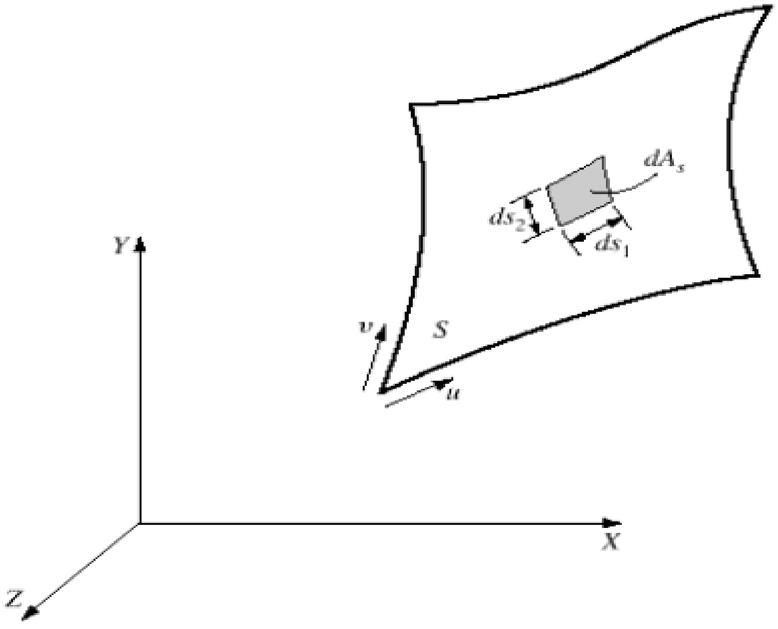
Where A_m is the net area, A is the area of the singly connected region bounded by the outside boundary by the outside boundary, and A_{hj} is the area of the jth hole. Each of the areas A and any A_{hj}.

6. **List out and describe various mass computed properties for a cross section.**

Surface Area

The surface area A_s of a bounded surface, shown in Figure, can be formulated in a similar way to the cross-sectional area.

The major difference is that As is not planar in general as in the case of a cylindrical, spherical, B-spline, or a Bezier surface.



Surface area.

The surface area A_s of the surface is given by:

$$A_s = \int dA_s$$

The sides are taken along the u and v directed as d_{s1} and d_{s2}

Then equation can be written as: $A_s = \iint ds1\ ds2$

$$d_{s1} = \sqrt{}P_{u\,.}P_u du$$

$$d_{s2} = \sqrt{}P_{v\,.}P_v dv$$

After substituting the values the equation becomes,

$$A_s = \int_{v1}^{v2}\int_{u1}^{u2}\sqrt{}(P_u.P_u)\ (P_v.P_v)\ du\ dv$$

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

$$A_s = \sum_{i=1}^{n} A_{si}$$

Volume

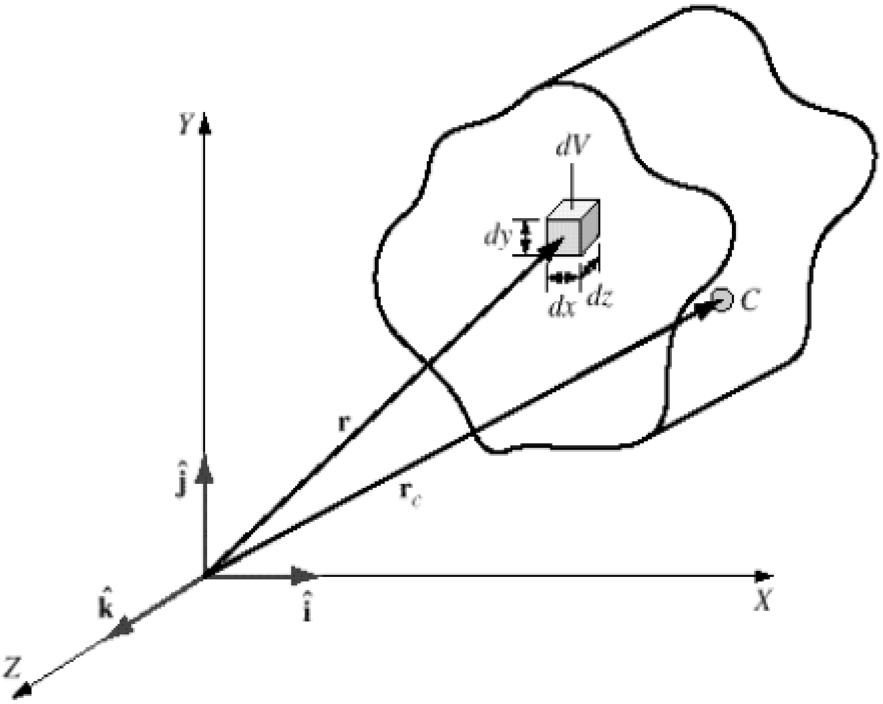
Figure shows an object whose volume is V. The volume V can be expressed as a triple integral by integrating the volume element dV, that is,

$$\mathrm{V}=\iiint dV=\iiint dxdydz$$

The volume integral of above Eq. can be changed into a surface integral using the Gauss divergence theorem, which can be expressed as follows:

$$\iiint F.n\,dS=\iiint \Delta.FdV$$

$$\mathrm{V}=\iiint dV=\iint z(k.n)dAs$$



Volume.

7. Explain in detail about mass property calculation in CAD

Mass Properties

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 g and/or the density of the object material is not constant.

We assume that g and ρ are constants. Therefore the three centers (of volume, of Mass, and of weight) coincide and equal the centroid (of the volume) of the object.

This assumption implies that objects of interest are homogeneous and are always close to the surface of the earth.

Mass

Mass is the amount of matter contained in an object.it depends on its volume and density of the materials.

Here, the mass of the object is calculated initially by considering a small element and it is then expanded to whole object.

The mass of an object can be formulated in a way similar to formulating its volume. If we replace the volume element dV, and density(ρ), the small mass (dm) we can write:

$$dm \ =\rho dV$$

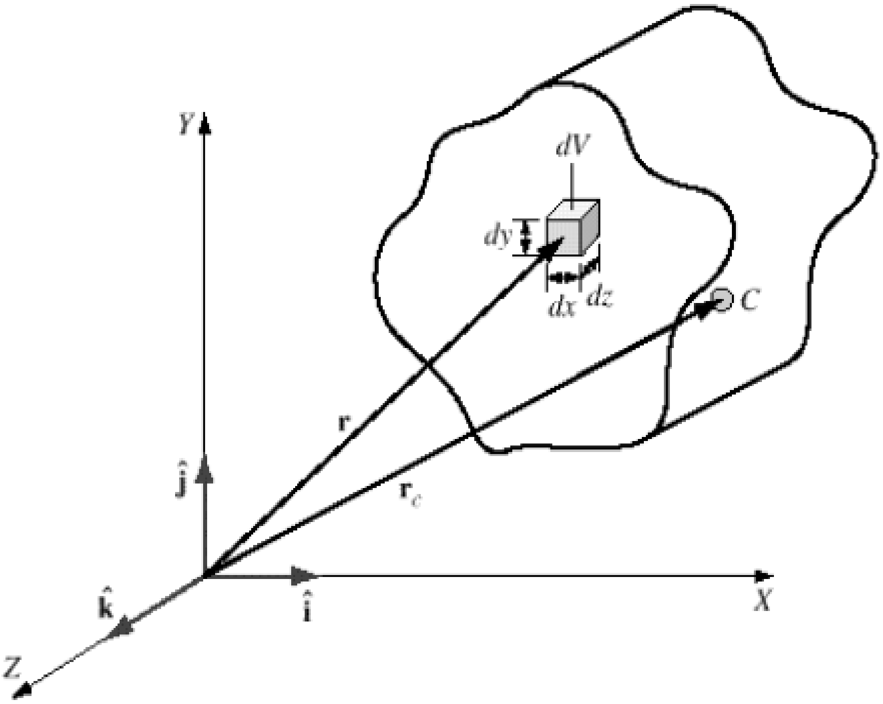
Integrating Eq. over the distributed mass of the object gives:

$$M=\iiint \rho dV$$

Assuming the density ρ to be uniform, then Eqn becomes,

$$M=\rho\iiint dV=\rho V$$

Once the volume of an object is calculated, it is multiplied by its density to obtain its mass.



Volume.

Centroid

The centroid can be thought of as the location on an object which experiences only translation and no rotation when a force is applied to the object.

The centroid of area for a symmetrical object such as square or circle is located at the center of the object.

The center of mass of an object can be calculated using the Eqn,

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

If we substitute ρV form m in Eq. and the center of mass is coincident with the center of volume. The equation leads to the center of mass (centroid).

First Moments of Inertia

The first moment of an area, volume, or mass is a mathematical property that appears in various calculations.

It is defined as the moment of an object property (area, volume, or mass) with respect to a given plane.

For a lumped mass, the first moment of the mass about a given plane is equal to the product of the mass and its perpendicular distance from the plane.

Using this definition, the first moments of a distributed mass of an object with respect to the XY, XZ, and YZ planes are given by, respectively:

$$M_{xy} = \iiint_m z\,dm = \iiint_V z\,dV$$

$$M_{xz} = \iiint_m y\,dm = \iiint_V y\,dV$$

$$M_{yz} = \iiint_m x\,dm = \iiint_V x\,dV$$

We can easily recognize that these expressions for first moments appear in the centroid gives:

$$Mxy=\rho Vz,=mz_c$$

$$Mxz =\rho Vy_c = my_c$$

$$Myz =\rho Vx_c = mx_c$$

As Equation show, the first moments are by-products of the volume and centroid calculations.

Second Moments and Products of Inertia

The second moment of inertia of a lumped mass about a given axis is the product of the mass and the square of the perpendicular distance between the mass and the axis.

The second moments of inertia of a distributed mass about the X, Y, and Z axes can be written as:

$$I_{xx} = \iiint\limits_{m} y^2 + z^2 \;dm = \rho \iiint\limits_{V} y^2 + z^2 \;dV$$

$$I_{yy} = \iiint\limits_{m} x^2 + z^2 \;dm = \rho \iiint\limits_{V} x^2 + z^2 \;dV$$

$$I_{zz} = \iiint\limits_{m} x^2 + y^2 \;dm = \rho \iiint\limits_{V} x^2 + y^2 \;dV$$

The second moments of inertia can be written as:

$$I_{xy} = \rho \int_{v1}^{v2} \int_{u1}^{u2} xyzK \;\hat{\mathbf{k}} \;\; \hat{\mathbf{n}} \;\; du\,dv$$

$$I_{xz} = \rho \int_{v1}^{v2} \int_{u1}^{u2} \frac{xz^2}{2} K \;\hat{\mathbf{k}} \;\; \hat{\mathbf{n}} \;\; du\,dv$$

$$I_{yz} = \rho \int_{v1}^{v2} \int_{u1}^{u2} \frac{yz^2}{2} K \;\hat{\mathbf{k}} \;\; \hat{\mathbf{n}} \;\; du\,dv$$

And for multiple surface objects;

$$I_{xy} = \rho \sum_{i=1}^{n} \int_{vi1}^{vi2} \int_{u_{i1}}^{u_{i2}} xyzK_i \;\hat{\mathbf{k}} \;\; \hat{\mathbf{n}}_i \;\; du\,dv$$

$$I_{xz} = \rho \sum_{i=1}^{n} \int_{vi1}^{vi2} \int_{u_{i1}}^{u_{i2}} \frac{xz^2}{2} K_i \;\hat{\mathbf{k}} \;\; \hat{\mathbf{n}}_i \;\; du\,dv$$

$$I_{yz} = \rho \sum_{i=1}^{n} \int_{vi1}^{vi2} \int_{u_{i1}}^{u_{i2}} \frac{yz^2}{2} K_i \;\hat{\mathbf{k}} \;\; \hat{\mathbf{n}}_i \;\; du\,dv$$

Property Mapping

The volume and mass properties already presented are formulated with respect to a given xyz coordinate system.

For a geometric model, this system is the MCS of the model database. The MCS is a convenient system because all the curve, surface. or solid equations and other geometric information are stored with respect to this system.

If mass properties are to be calculated with respect to other systems, these properties must be mapped from the MCS to the other systems.

In the following discussions, let us assume that mass properties are already available in the MCS (XYZ) and we need to map them to a given WCS (X_wY_w,Z_w).

The coordinates of the centroid (x_c, y_c, z_c) can be mapped from the MCS to the WCS using the following equation:

$$r_{cW}=[T]r_c$$

$$\begin{bmatrix} x_{cW} \\ y_{cW} \\ z_{cW} \\ 1 \end{bmatrix} = T \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

Where [T] is the general mapping matrix given by Eqn.

Once the centroid is mapped, the first moments M_{xy}, M_{xy} , and $M_{xy}$are automatically mapped.

$$\begin{bmatrix} M_{xyW} \\ M_{xzW} \\ M_{yzW} \\ 1 \end{bmatrix} = m \begin{bmatrix} x_{cW} \\ y_{cW} \\ z_{cW} \\ 1 \end{bmatrix} = m \ T \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

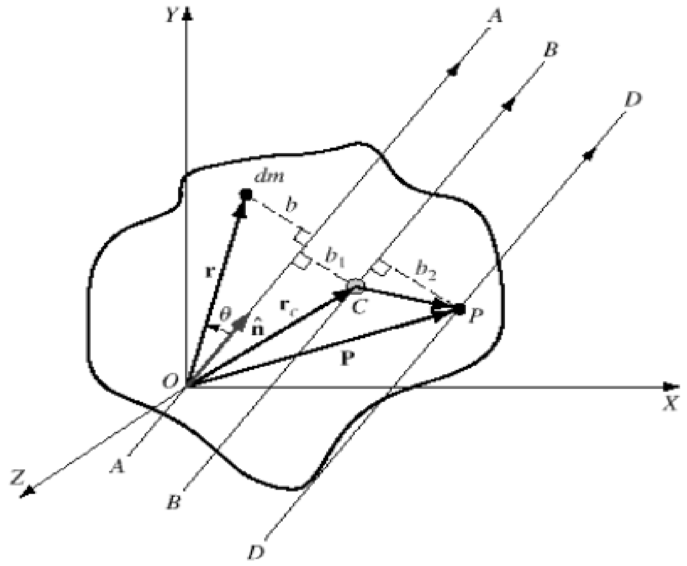
The second moments and products of inertia of an object with respect to its MCS can be mapped to compute the moments of inertia of the objects about any arbitrary axis.

The axis may or may not pass through the origin of the MCS. The axis AA passes through the origin O, and its direction in space is defined by the unit vector n by definition,

$$I_{aa} = \iiint_m b^2 dm$$

Where b is the perpendicular distance from dm to AA. if the position of dm is located using r, then b=sinθ=|nxr|. Hence, Iaa can be expressed as:

$$I_{aa} = \iiint |nXr|^2 dm = \iiint (nXr).(nXr) dm$$

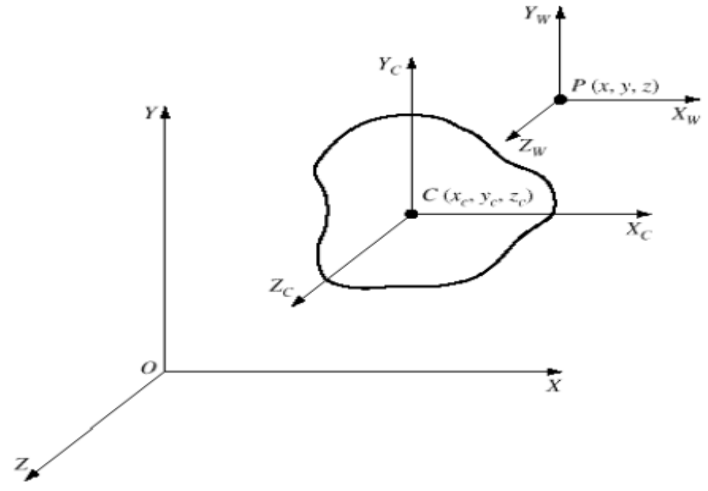


Moments of inertia about an arbitrary axis.

Applying the parallel axis theorem twice, we can write:

$$I_{xx\ C} = I_{xx} - m\ \ y_c^2 + z_c^2$$
$$I_{yy\ C} = I_{yy} - m\ \ x_c^2 + z_c^2$$
$$I_{zz\ C} = I_{zz} - \ \ x_c^2 + y_c^2$$
$$I_{xy\ C} = I_{xy} - m x_c y_c$$
$$I_{xz\ C} = I_{xz} - m x_c z_c$$
$$I_{yz\ C} = I_{yz} - m y_c z_c$$

$$I_{xx\ W} = I_{xx\ C} + m\ \ y - y_c\ ^2 + \ z - z_c\ ^2$$
$$I_{yy\ W} = I_{yy\ C} + m\ \ x - x_c\ ^2 + \ z - z_c\ ^2$$
$$I_{zz\ W} = I_{zz\ C} + m\ \ x - x_c\ ^2 + \ y - y_c\ ^2$$
$$I_{xy\ W} = I_{xy\ C} + m\ x - x_c\ \ y - y_c$$
$$I_{xz\ W} = I_{xz\ C} + m\ x - x_c\ \ z - z_c$$
$$I_{yz\ W} = I_{yz\ C} + m\ y - y_c\ \ z - z_c$$



Moments of inertia about axes parallel to the MCS.

Principal moments of inertia are useful properties to determine for an object. Their corresponding axes are the principal axes of inertia.

These axes are characterized such that products of inertia with respect to them are zeros.

Thus, the principal inertia tensor is a diagonal matrix whose diagonal elements are the principal moments of inertia I_x, $I_{y,}$ and I_z .These moments are the root of the following cubic equation:

$$I^3 - I_{xx} + I_{yy} + I_{zz} \; I^2 + \; I_{xx}I_{yy} + I_{yy}I_{zz} + I_{zz}I_{xx} - I_{xy}^2 - I_{yz}^2 - I_{xz}^2 \; I$$
$$- \; I_{xx}I_{yy}I_{zz} - 2I_{xy}I_{yz}I_{xz} - I_{xx}I_{yz}^2 - I_{yy}I_{xz}^2 - I_{zz}I_{xy}^2 \;\; = \; 0$$

The radius of gyration R_g of an object with respect to an axis is one of the inertial properties of the object. It is defined by the following equation.

$$R_g \; = \; \sqrt{\frac{I}{m}}$$

Properties Evaluation

All the equations are integral equations in the parametric space. Some are line integrals to be evaluated over curves or edges of the object.

In this case, the integrand is a function of the parameter 11. Others are surface integrals to be evaluated over surfaces or faces of the object, in which case the integrand is a function of the parameters u and v.

These surface integrals can be further reduced to line integrals for analytic surfaces.

These two types of integrals take the following general form:

$$\text{I}=\int_{u1}^{u2} f(u)du$$

$$\text{I}=\int_{v1}^{v2} \int_{u1}^{u2} f(u,v)\, dvdu$$

Let us apply the two methods to first Eqn. To evaluate the integral in Eq. numerically, the integral is approximated by a polynomial and a remainder, that is,

$$I=I_a+ R$$

Where Ia is the approximate value of I, and R is the remainder. Ia usually takes the form of a polynomial, and R is the source of error or approximation in the numerical evaluation.

In **Newton-Cotes integration**, it is assumed that the sampling points off (u) are spaced at equal distances in the interval [u_1, u_2]. If we use (n+ l) sampling points, we can define:

$$u_0 = u_1 \qquad u_n = u_2 \qquad h = \frac{u_2 - u_1}{n}$$

and the integration formula for the Newton-Cotes method can be written as:

$$I = \int_{u_1}^{u_2} f\ u\ du = u_2 - u_1 \sum_{i=0}^{n} C_i^n f_i + R$$

Where C^n_i: are the Newton-Cotes constants and $f_i = f(u_i) = f(u_0 + ih)$. The cases n=1 and n=2 are the well-known trapezoidal and Simpson rules shown in Figure. The constants for these two rules are given by:

$n = 1;$

$$C_0^1 = C_1^1 = \tfrac{1}{2}$$

$n = 2;$

$$C_0^2 = \tfrac{1}{6} \qquad C_1^2 = \tfrac{4}{6} \qquad C_2^2 = \tfrac{1}{6}$$

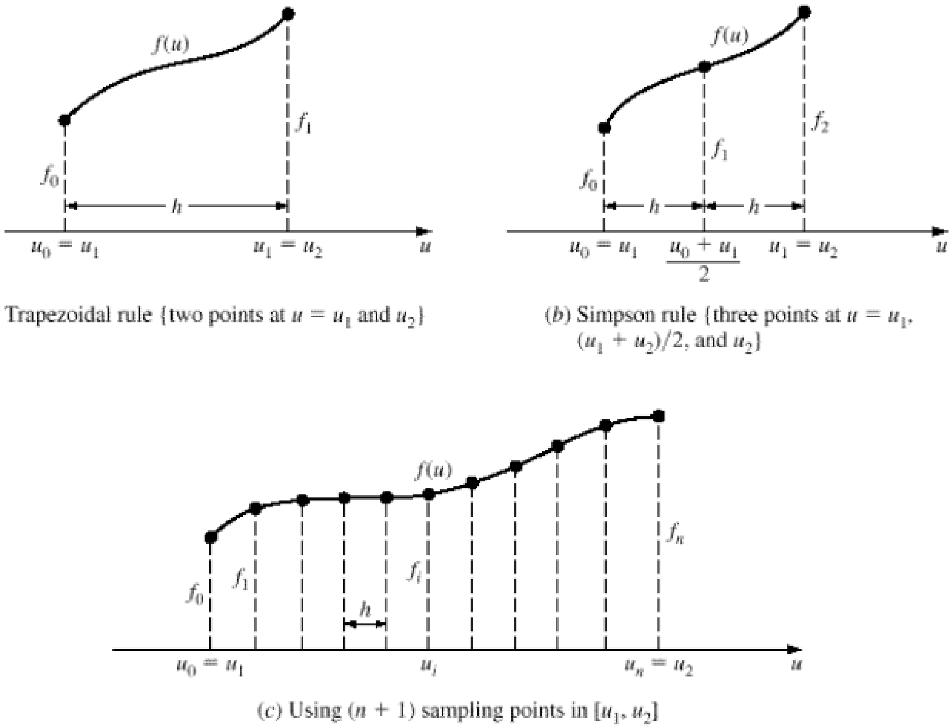
Therefore Eqn for trapezoidal rule:

$$I = \int_{u_1}^{u_2} f\ u\ \ du \quad \frac{u_2 - u_1}{2}\ f_0 + f_1 \quad = \frac{h}{2}\ f_0 + f_1$$

And for the simpson rule:

$$I = \int_{u_1}^{u_2} f\ u\ \ du \quad \frac{u_2 - u_1}{6}\ f_0 + 4f_1 + f_2$$

$$= \frac{h}{3}\ f_0 + 4f_1 + f_2$$



Trapezoidal rule {two points at $u = u_1$ and u_2}

(b) Simpson rule {three points at $u = u_1$, $(u_1 + u_2)/2$, and u_2}

(c) Using (n + 1) sampling points in $[u_1, u_2]$

Sampling points for trapezoidal and Simpson rules.

The minimum number of sampling points in the interval $[u_1,u_2]$are two and three for the trapezoidal and Simpson rules, respectively.

The two methods can still be applied if (n + 1) sampling points are used in the interval $[u_1, u_2]$ as shown in Figure c.

In this case, apply each method repetitively using two sampling points for the former and three for the latter, and then add to obtain:

For trapezoidal rule:

$$I = \int_{u_1}^{u_2} f \;\, u \;\; du \quad h \;\tfrac{1}{2}f_0 + f_1 + f_2 + \qquad + f_{n-1} + \tfrac{1}{2}f_n$$

Simpson rule:

$$I = \int_{u_1}^{u_2} f \;\, u \;\; du \quad \frac{h}{3}\; f_0 + 4f_1 + 2f_2 + 4f_3 + \qquad + 2f_{n-2} + 4f_{n-1} + f_n$$

For the Simpson rule, n must be an even number. The error E introduced by Newton-Cotes integration is equal to the remainder R. An upper bound on the error E is given by: trapezoidal rule:

$$E < \frac{u_2 - u_1}{10}^{3} \left[\frac{d^2 f}{du^2}\right]_{u_2}$$

Simpson rule:

$$E < \frac{u_2 - u_1}{1000}^{5} \left[\frac{d^4 f}{du^4}\right]_{u_2}$$

In the integration of geometric-related equations such as mass property calculations, a function is called to evaluate the function f(u) at given points. These points may be chosen anywhere in the interval [u_1,u_2].

Therefore, it seems natural to optimize the positions of the sampling points to improve the accuracy of numerical integration.

Gauss quadrature is a numerical integration method in which both the positions of the sampling points and the associated weights have been optimized.

In general, Gauss quadrature is usually more accurate than Newton-Cotes methods.

Mass Properties on CAD/CAM System

Several mass properties commands and/or modifiers exist on CAD/CAM systems to support calculations for 2D planar areas, $2_{1/2}$D objects, and 3Dobjects.

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/in^3 or g/mm^3). The screenshot on the right shows an example of these options.

8. List out various fundamental rules for Geometric tolerance (May 2017)

Engineering Tolerances

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.

Needs for Tolerances

The assignment of actual values to the tolerance limits has a major influence on the overall cost and quality of an assembly or a product.

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.

In order for mating features (faces) of mating parts to fit together and operate properly, each part must be manufactured within these tolerance limits.

Tolerancing is an important and essential element of mass production and interchangeable manufacturing by which parts can be made in widely separated locations and then brought together for assembly.

Tolerancing also makes it possible for spare parts to replace broken or worn ones in existing products successfully without interchangeable manufacturing.

Modern industry could not exist, and without effective size control by the engineer, interchangeable manufacturing could not be achieved.

Tolerancing information is essential for part design and manufacturing. Design engineers need tolerance analysis to distribute allowances among related design dimensions to check design results, or to design assemblies.

Production engineers need tolerances for process planning, assembly operations, part inspection, and other production activities.

Conventional Tolerances

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

Nominal size is the designation used for the purpose of general identification. 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.

It is the decimal equivalent of the nominal size. The number of decimal places determines the precision or accuracy required.

For example, if the nominal size of a pipe is $3_{1/2}$ in and we require accuracy to three decimal places, its basic size becomes 3.500 in.

We should point out that 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. It is a combination of the basic size and its assigned tolerance.

It is the size that results from producing the part, including all the manufacturing imperfections.

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. Tolerance can be expressed in either of two ways.

A bilateral tolerance is specified as plus or minus deviation from the basic size, for example, 1.750 ± 0.002 in.

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

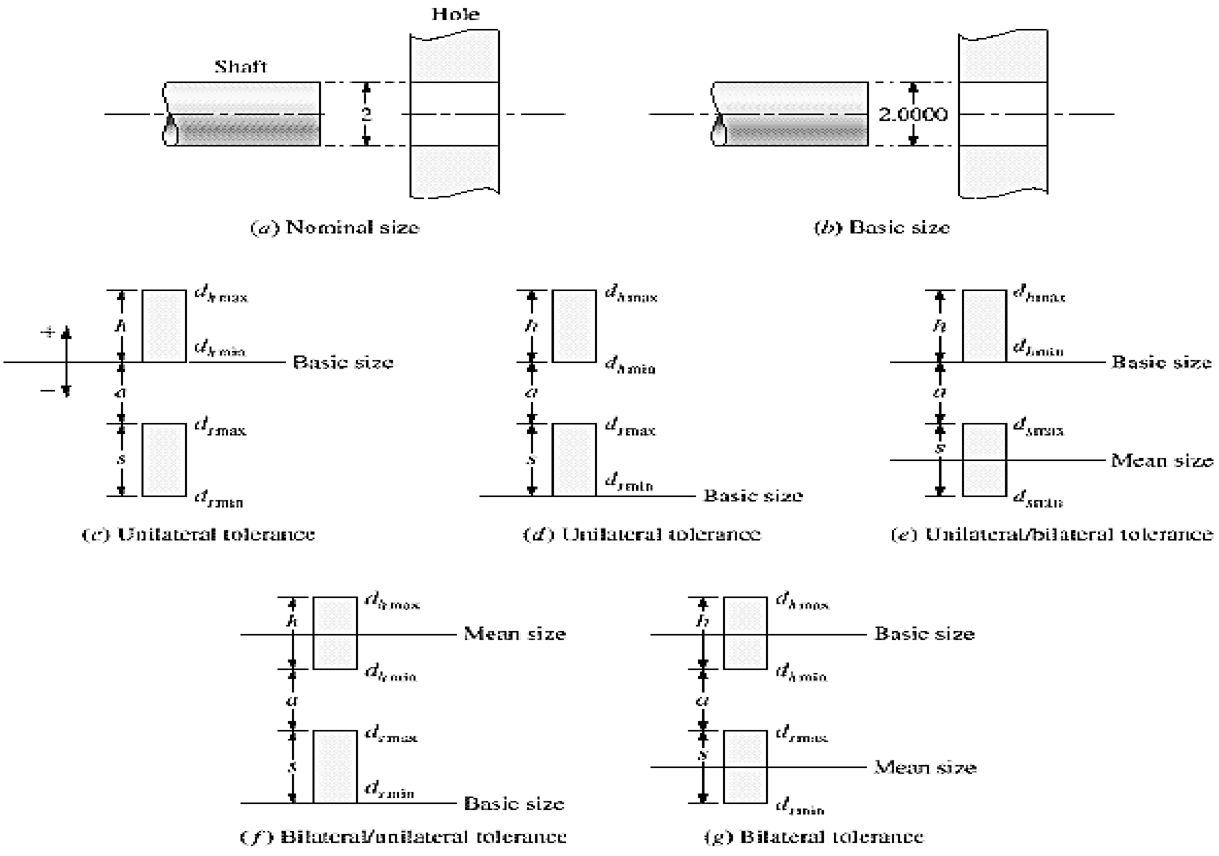
Allowance is the difference between the maximum material limits of mating parts. It is the minimum clearance (positive allowance) or maximum interference (negative allowance) between mating parts.

It is also known as the tightest fit between mating parts-the smallest clearance or the largest interference. Allowance is an algebraic value it could be a positive or negative value.

Allowance is used to determine the type of lit (clearance, transition, or interference)

Clearance Fit Tolerance Calculation

Assuming that allowance and tolerances are specified to four decimal places, the precision required is then the same, and the basic size is expressed as 2.0000 in as shown in Figure.



(*a*) Nominal size

(*b*) Basic size

(*c*) Unilateral tolerance

(*d*) Unilateral tolerance

(*e*) Unilateral/bilateral tolerance

(*f*) Bilateral/unilateral tolerance

(*g*) Bilateral tolerance

From the functional and assembly requirements between the gear and the shaft (power is transmitted between the two via a key), a clearance fit between the two parts is adequate.

Let us assume a tolerance II for the hole in the gear of value 0.0030 in, shaft tolerances of 0.0030 in, and an allowance a of 0.0030 in. The variables h, s, and a each define a tolerance zone.

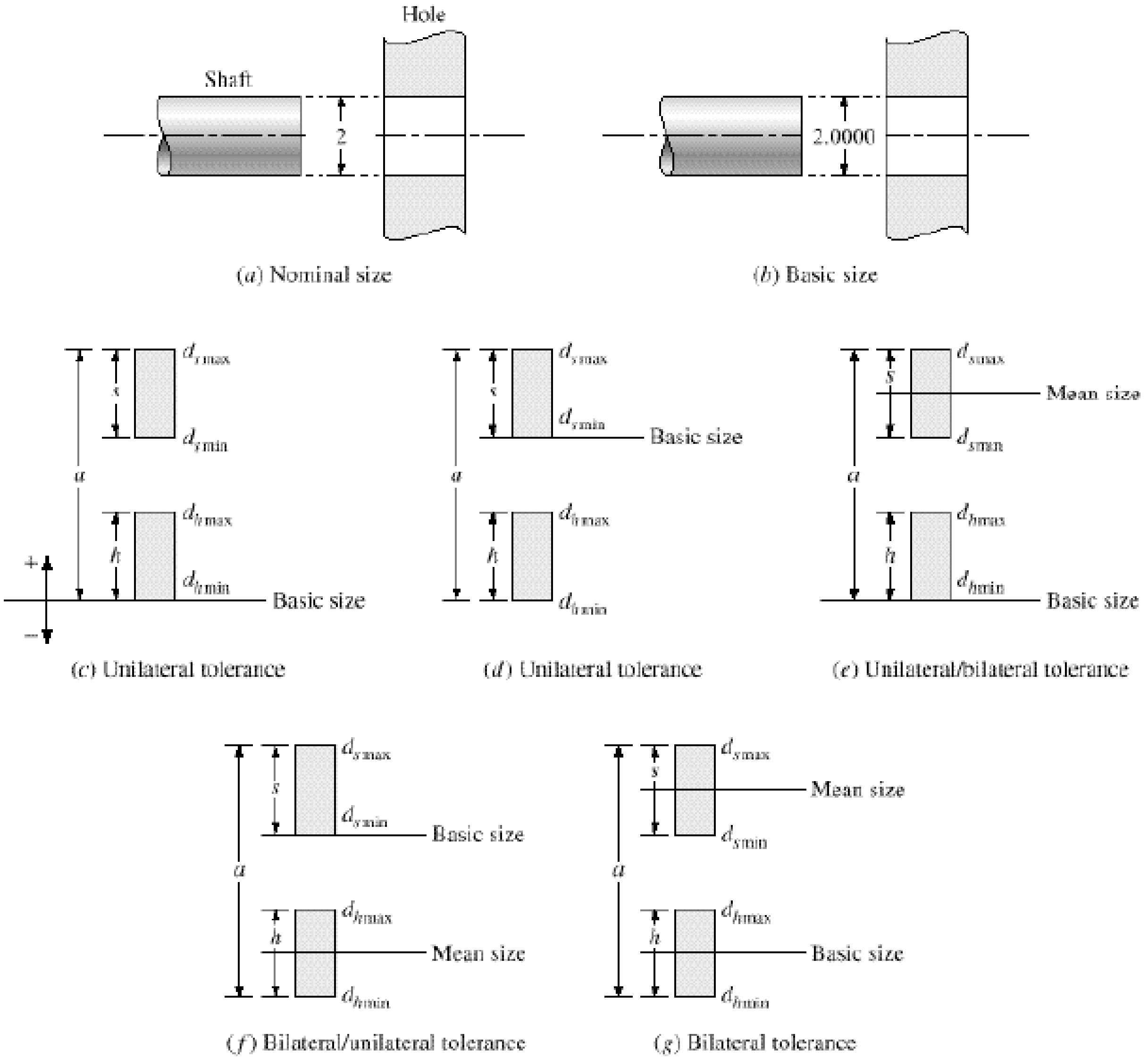
In order to determine the diameters of the gear hole and the shaft given the values for h, s, and a, let us consider all the possible variations in the values of these dimensions relative to the basic size Ten possibilities exist.

Five of them are shown as bar diagrams in Figure c to g. The other five possibilities are complements (reverse the location of the bar adjacent to the basic-size datum) of those shown in the figure.

The shaded bars are the tolerance zones. The relative locations of these zones with respect to the basic size and the mean size determine the type of tolerance, and consequently the hole and shaft diameters.

To determine the maximum and minimum diameters for the hole (d_{max} and d_{min}) and the shaft (dmax and d_{min}), a tolerance zone (hors) is allocated relative to the basic-size datum, followed by the allowance a and the other tolerance zone.

Interference Fit Tolerance Calculations



(*a*) Nominal size

(*b*) Basic size

(*c*) Unilateral tolerance

(*d*) Unilateral tolerance

(*e*) Unilateral/bilateral tolerance

(*f*) Bilateral/unilateral tolerance

(*g*) Bilateral tolerance

Fits and Limits

Fits between mating parts specify the range of tightness or looseness that may result from the application of specific tolerances to mating parts.

The differences between the hole and the shaft limits determine the range of tightness or looseness.

Fits between mating parts 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 10 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 tits.

In clearance fit one part is always loose relative 10 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 tits which can transmit torques or force between mating parts (e.g., a shaft and a pulley) are usually referred to as force fits.

Location interference tits are used when accuracy of location is important and for parts requiring alignment.

A **transition fit** is a fit which may result in either clearance or interference condition.

There exist two systems: the basic hole system and the basic shaft system. Both systems assume unilateral tolerances h and for the hole and the shaft, respectively.

In **the basic hole system**, the minimum hole is taken as the basic size, and the allowance and tolerances are applied accordingly. Figure lc shows an example of using the basic hole system to apply tolerances and calculate limits.

In **the basic shaft system** the maximum shaft is taken as the basic size, and the allowance and tolerances are applied accordingly. The complement of Figured would show an example of using the basic shaft system to apply tolerance and calculate limits.

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 to 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.

Datum's

A datum is a "theoretical plane" which acts as a master reference for locating features (surfaces) of a part during manufacturing and inspection.

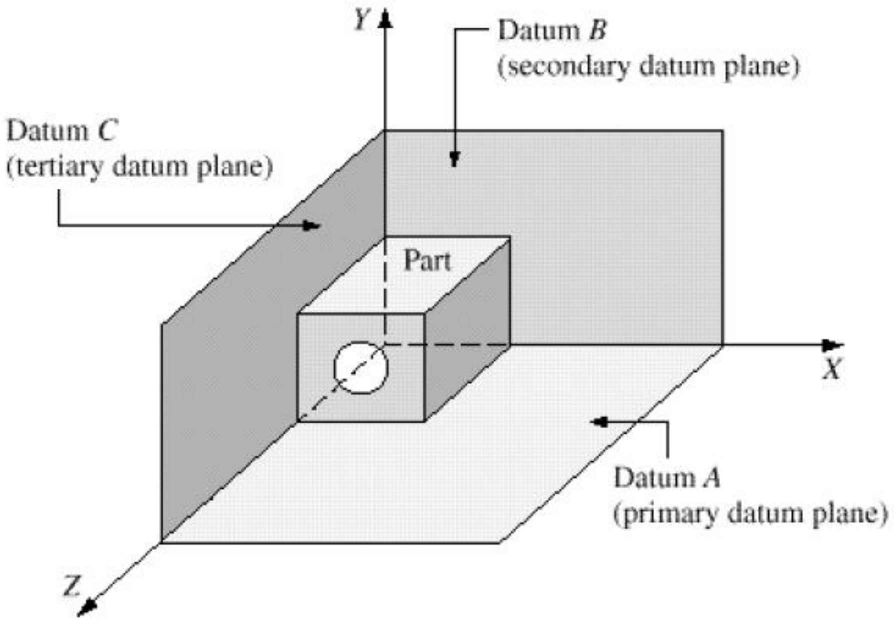
A datum may be a point, a line, or a plane. A datum plane may be created along each of the three (X, Y, and Z) axes as shown in Figure.

The three mutually orthogonal datum planes form what is sometimes called the datum reference system (also known as the master datum system),

The most important plane to part measurements is termed datum A (the horizontal plane in Figure) and is known as the primary datum.

The second most important is datum B (known as the secondary datum), and the least important is datum C (tertiary datum).

The designer usually decides the priority of importance of the three datum planes based on the functional and assembly requirements of the part.



Datum reference system.

9. **Describe in detail about Tolerance analysis and explain the types of analysis. (Nov 2015, May 2016, Nov 2016, May 2016)**

Tolerance Analysis

The designer must make sure that the combined effect or accumulation of all these tolerances (tolerance stack up) does not cause an inoperable or malfunctioning assembly.

Analysis of tolerance and their stack up is important because tolerance assignment is usually done on a part-by-part basis.

Tolerance analysis is defined as the process of checking the tolerances to verify that all the design constraints are met Tolerance analysis 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.

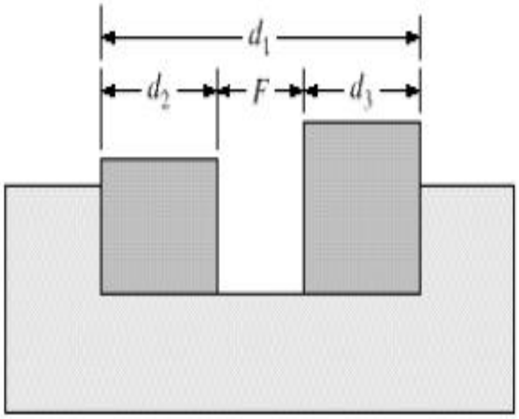
Product dimensions and variables that control the behavior of a design function are called design function variables.

The variability of design functions is used to assess the suitability of a particular tolerance specification.

Figure shows an example of a design function for the case of two blocks assembled into a slot.

The design function F is the clearance between the two blocks, and is a function of the dimensions of the slot and the two blocks.

A tolerance specification for these dimensions is satisfactory if it prevents F from being less than zero.



d_1, d_2, d_3 = product dimensions (design function variables)
F = design function
$F = f(d_1, d_2, d_3) = d_1 - d_2 - d_3$
$F \geq 0$, accept assembly
$F < 0$, reject assembly

Tolerance representation.

The formulation of tolerance analysis can be stated as follows. Given a set of tolerance

{T} = {T_1,T_2.....T_n) on a set of dimensions{d) = {d_1, d_2... , d_n), and a set of design constraints{C}={ C_1,C_2.....,C_n}, The dimensions in the set { d} include both nominal dimensions {d,v] and their tolerances {T}, that is.

$$d = d_N + T$$

To assess tolerance suitability, we formulate a design function in terms of {d}, as follows:

$$F = f \ \{d\} \ = f \ d_1 \ d_2 \qquad d_n$$

The variability of F due to variability in {d} is determined (using methods described in the discussion that follows).

If F satisfies {C} all the time, {T) is satisfactory and the assembly is accepted. If not, {T} is unsatisfactory and the assembly is rejected.

Design functions are often complex, and their formulation forms the hardest part of tolerance analysis and can be time consuming.

Tolerance analysis methods can be divided into two types. In the simpler type, dimensions have conventional tolerances, and the result of tolerance analysis is the nominal value of the design function (F_N) and its upper (F_{max}) and lower (F_{min}) 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 constraint.

The upper and lower limits of the design function represent the worst possible combination of the tolerances of the design function variables.

However, the like hood of worst-case combination of these tolerances in any particular product is very low. Therefore, **worst-case tolerance analysis** is very conservative.

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. Manufacturing costs are reduced by loosening up the tolerances and accepting a calculated risk that the design constraints {C} may not be satisfied 100% of the time.

By assuming a probability distribution for each tolerance dimension, it is possible to determine the likelihood that the specified design limits will be exceeded.

Effectively, a reject rate is determined for the assembly. A non-zero reject rates may be preferable to an increase in individual part manufacturing costs due to tight tolerances. Both the worst-case and statistical approaches are used in practice.

Worst-Case Arithmetic Method

The arithmetic tolerance method is the worst-case analysis 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.

All manufactured parts are interchangeable since the maximum values are used. 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.

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.

Typically, the probabilistic distribution curve is assumed to be a normal distribution curve. This method is used in both batch and mass production.

It allows for variability in manufacturing conditions such as tool wear machine conditions, and random errors.

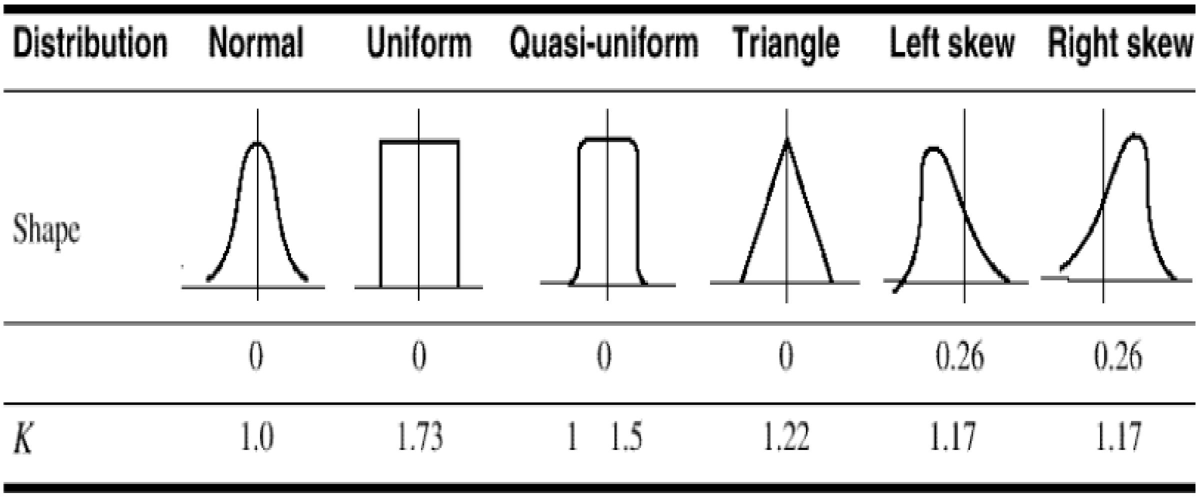
It increases the manufacturing efficiency by increasing tolerance limits and, therefore, reducing the required accuracy of manufacturing.

This method is applied to a closed-loop dimension set {d} with each element d, of the set having a probability distribution curve.

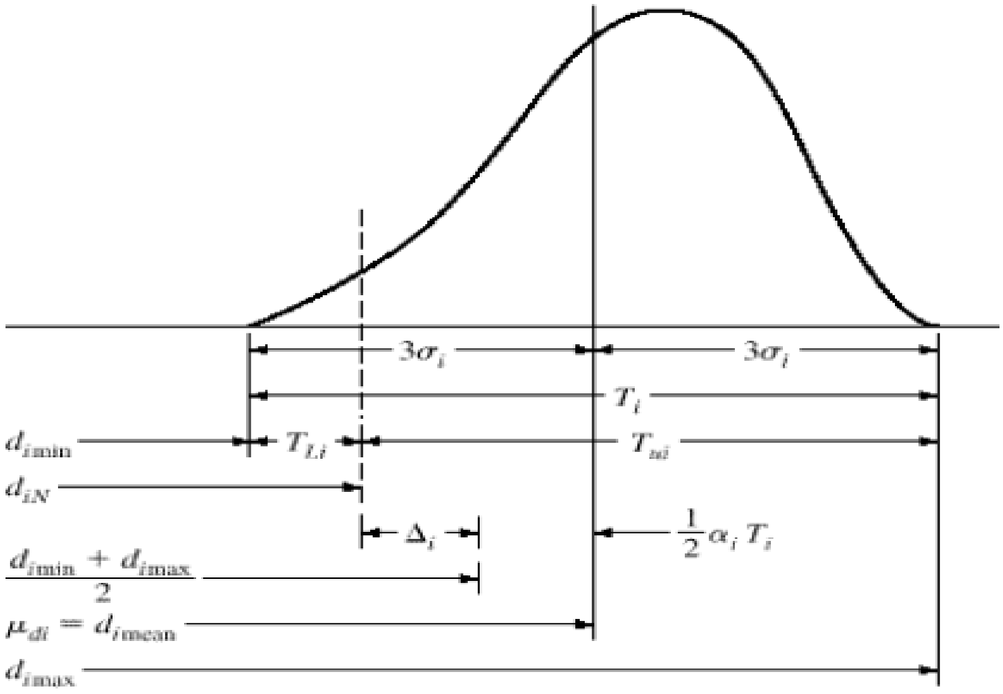
The design function F is obtained in the same way as in the arithmetic method. The tolerance information about F [similar to Equ.] can be obtained statistically as follows.

A normal distribution is considered the basis of the analysis Parameters relating other distributions to the normal distribution are shown in Table. Figure shows the parameters of a distribution curve for one of the elements of the dimension set.

Table: Probabilistic distribution curve



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


$3\sigma_i$
$3\sigma_i$
T_i
$d_{i\min}$
T_{Li}
T_{ui}
d_{iN}
Δ_i
$\frac{1}{2}\alpha_i T_i$
$\frac{d_{i\min} + d_{i\max}}{2}$
$\mu_{di} = d_{i\text{mean}}$
$d_{i\max}$

Probabilistic distribution curve of a dimension.

Monte Carlo Simulation Method

The previous two methods are only applicable to conventional tolerances- mainly for closed-loop dimensional sets with linear design functions.

When 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 $0.01\sqrt{a^2 + b^2 + c^2}$. To use the worst-case statistical method is less obvious and may require linearizing Eq.

These 2D and 3D functions can be written first in terms of nominal dimensions and then perturbed using the geometric tolerances.

While this approach enables including geometric tolerances in the tolerance analysis, it still requires a designed function which it may not be possible to find.

One of the methods that seem useful in performing tolerance analysis using geometric tolerance without a need for an explicit design function is the Monte Carlo simulation method.

The idea of the Monie Carlo method stems from the manufacturing practice wherein a prototype of a given assembly is built and assembled to test its tolerances and functionality.

The Monte Carlo method achieves the same result (within the reliability limits of the method) without the time and cost of part manufacturing.

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.

Either a worst-case statistical analysis (using tolerance limits which give d_{max} and d_{min} or just a statistical analysis can be performed.

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 chat 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.

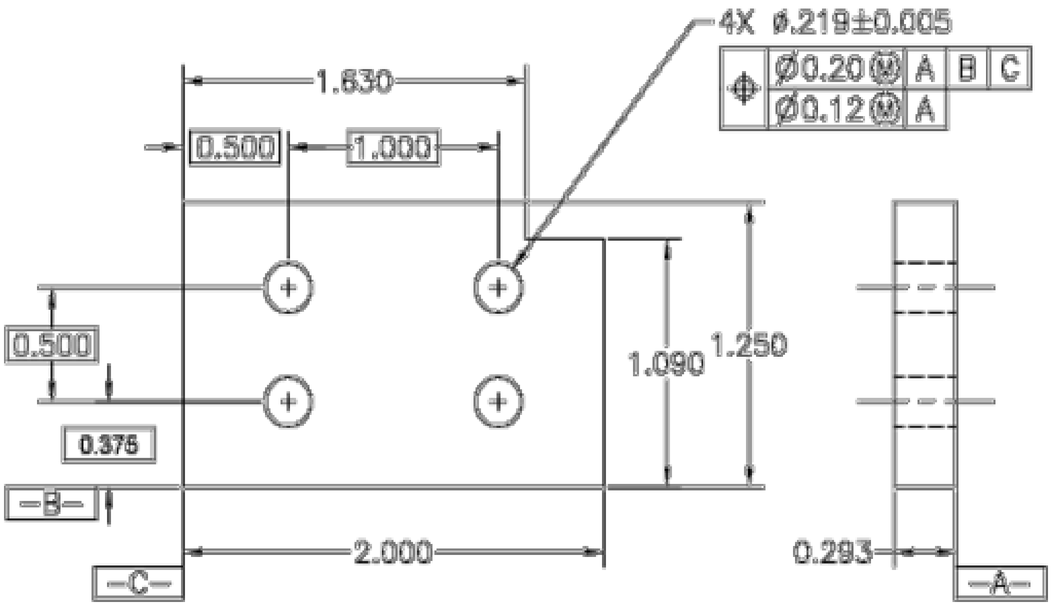
Thus a statistical Distribution can be generated for each design constraint. Probabilities of accepting or rejecting assemblies can, therefore, be estimated.

Geometric Tolerance

The function of geometric tolerance is to explain the engineering objective of components and assemblies.

The datum reference frame can explain how the part. Tolerance can accurately define the dimensional needs for a part, permitting over 50% more tolerance than coordinate dimensioning n a few cases.

Suitable purpose of tolerance will confirm that the part described on the drawing has the preferred form, fit and purpose with the highest possible tolerances.



Geometric Tolerance

Fundamental rules for Geometric Tolerance

1. All dimensions should have a tolerance. Each attribute one very manufactured component is subject to change; hence, the limits of acceptable difference must be defined.

   Plus and minus tolerances may be used to dimensions from a common tolerance block.

2. Dimensions describe the geometry and allowable change. Measurement and scaling of the drawing is not permitted excluding in certain cases.

3. Engineering drawings describe the necessities of completed parts. Each dimension and tolerance needed to define the completed part shall be shown on the drawing.

   If extra dimensions would be useful, but are not necessary, they may be noted as reference.

4. Dimensions should be used to attributes and arranged in such away as to show the purpose of the features. In addition, dimensions should not be subject to

more than one explanation.

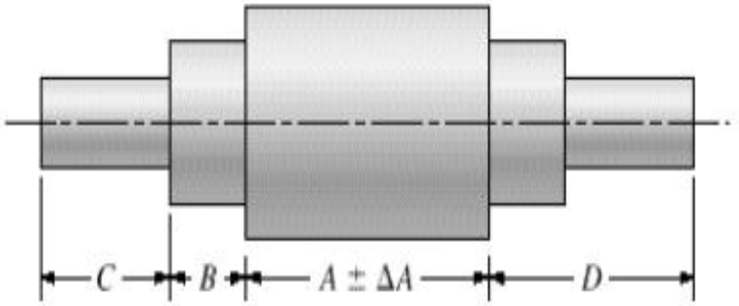
5. Descriptions of manufacturing systems should be avoided. The geometry should be explained without defining the technique of manufacture.
6. If some sizes are needed during manufacturing but are not wanted in the final geometry they should be noticeable as non-mandatory.
7. All dimensioning and tolerance should be placed foremost readability and should be used to visible lines in true profiles.
8. When geometry is usually restricted by code, the dimension(s)shall be integrated with code number in comments below the dimension.
9. If not openly declared, all dimensions and tolerances are only suitable when the item is in free.
10. Dimensions and tolerances indicate to the full length, width, and depth.

10.Explain in detail about Tolerance Synthesis

Tolerance Synthesis

In tolerance analysis, tolerances on dimensions are known and the tolerance on a resultant dimension is to be determined.

In terms of a design function, the variability of the function is to be obtained **Tolerance synthesis** (also called tolerance distribution) is the inverse problem; the allowable tolerance, or allowable variations in the design function.



Tolerance synthesis.

There are two criteria to distribute a tolerance: the equal tolerance criterion and the equal precision criterion.

The former simply distributes the tolerance equally among dimensions that affect the given tolerance.

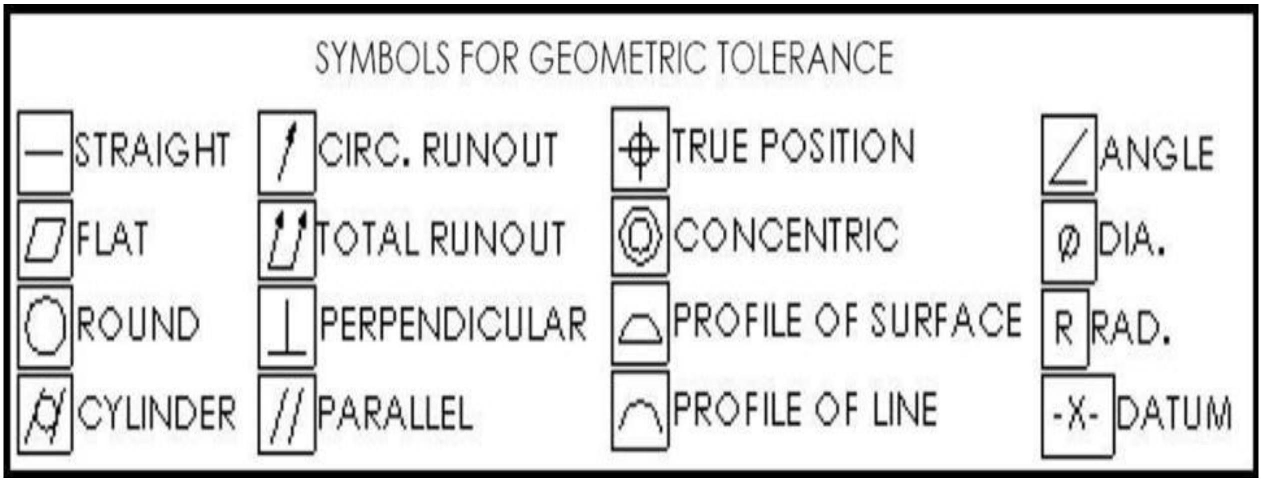
While it is easy to compute manually, it is not accurate based on design and manufacturing experience.

The latter is more suitable to real designs, but it involves more complicated calculations.

Tolerance Symbols

Symbols for tolerances are bilateral unless otherwise defined. For example, the location of a hole has a tolerance of 0.020mm.

This indicates that the hole can move+/- .010mm, which is an equal bilateral tolerance. It does not consider that the hole can move+0.015/-0.005mm,which is an unequal bilateral tolerance.



Symbols for Geometric Tolerance

Advantages of Tolerance Analysis

1. Accurate part assembly.
2. Elimination of assembly rework
3. Improvement in assembly quality.
4. Reduction of assembly cost.
5. High customer satisfaction.
6. Effectiveness of out-sourcing.

Limitations of Tolerance Analysis

1. Time consuming process.
2. Skill require for complex assemblies.

UNIT-V

1. **Explain in detail about the product data exchange with types of translator. (AU APR 2017)**

Or

Explain Data Exchange standards and translator used in data exchange.(AU MAY 2016)

Or

Explain the two types of translators with its advantages and disadvantages

Product Data Exchange

- ➢ 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. These are shape, non-shape, design, and manufacturing data.

1. **Shape data** consists of both geometrical and topological information.
2. **Non shape data** includes shaded images and measuring unit of the database.
3. **Design data** includes FEM/FEA.
4. **Manufacturing data** includes tolerancing and bill of materials.

- ➢ Where 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 here data communication problems arise.
- ➢ Many organizations have been collaborating to set exchange standards to make it manageable for users of CAD/CAM systems to communicate their product information effectively.

Types of Translators

- The direct solution produces **direct translators**. It 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 native formats.

- The indirect solution produces **indirect translators**. It is more general and 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.

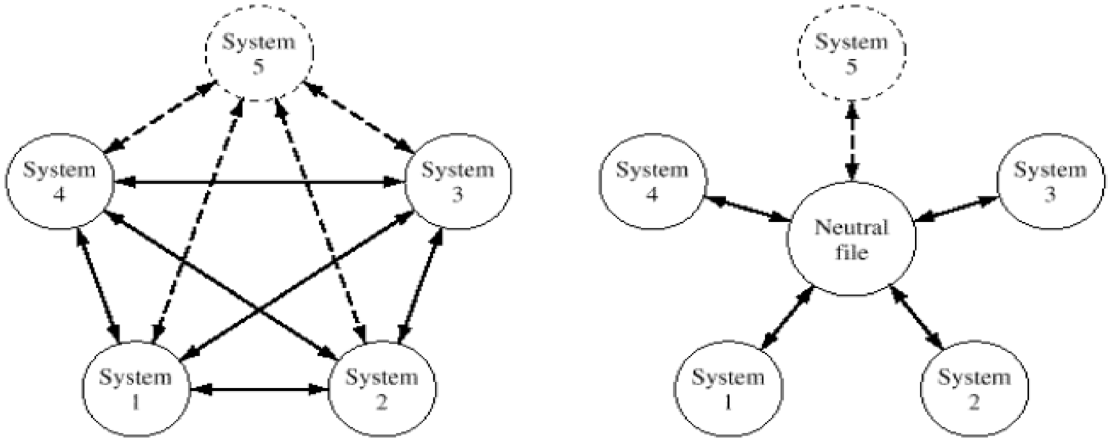
Direct translators

- Direct translators convert data directly in one step. They are typically written by computer service companies that specialize in CAD/CAM database conversion.
- Direct translators are considered dedicated translation programs, two of which link a system pair as indicated by the dual direction arrows.

For example: Two translators are needed to transfer data between System1 and System 2; one from System 1to System 2 and the other from System 2 to System 1.

Indirect Translators:

- Indirect translators utilize some neutral file format, which reflects the neutral database structure.
- Each translation system has its own pair of translators to translate data to and from the neutral format.
- The translator that converts data from the native format of a given CAD/CAM system to the neutral format is called a preprocessor, while the translator that does the opposite translation is known as a postprocessor,



(*a*) Direct translators

(*b*) Indirect translators

Types of CAD/CAM data translators

- ➢ 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.
- ➢ In general, if modeling data is to be transferred between all possible pairs of n CAD/CAM systems, then the total number of translators, N, that must be written is given by:

$$N = 2 \begin{matrix} n \\ 2 \end{matrix} = n \;\; n-1$$

$$\begin{matrix} n \\ 2 \end{matrix} = \frac{n!}{2 \;\; n-2 \;\; !}$$

- ➢ The symbol "!" indicates factorial. The coefficient 2 of the left-hand side of the same equation reflects the fact that two translators must be written for each pair of systems.
- ➢ Adding one system to the existing 11 systems would require writing 2n additional translators.
- ➢ 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.
- ➢ Indirect translators, they do not suffer from the increasing number of programs that have to be written as in the case of direct ones.

For the case of n systems, the total number of indirect translators that must be written is given by:

N=2n

- Adding one system (System5) on existing systems would only require writing two additional translators (shown dashed in the figure) regardless of the value of n.
- 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.

2. Describe the structure of IGES file (AU NOV 2016, AU APR 2017,AU MAY 2016, NOV 2015)

Or

Describe the IGES methodology

Or

Compare various testing methods of IGES processors

Or

Explain any three product data exchange standards in detail.

IGES

- 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.
- Currently, it supports solid modeling including both B-rep and CSG schemes.
- LGES defines a neutral database, in the form of a neutral file format which describes an "IGES model” of modeling data of a given product.
- The IGES model can be read and interpreted by dissimilar CAD/CAM systems.
- Therefore, the corresponding product data can be exchanged among these systems. IGES describes the possible entities that can be used to build an IGES model.
- The necessary parameters (data) for the definition of model entities, and the possible relationships and associatively between model entities.

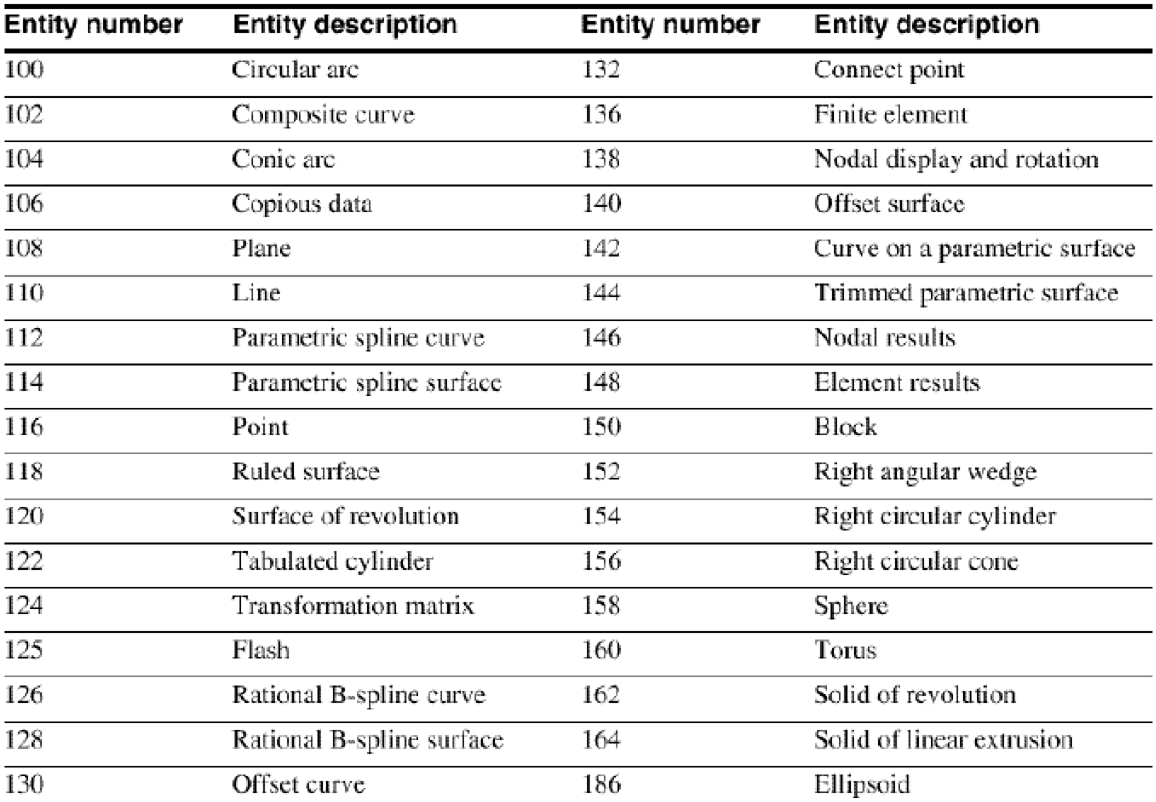
- An IGES model is based on the concept of entities. The fundamental unit of information in the model, and consequently in the IGES file, is the entity; all product data is expressed as a list of predefined entities.
- Each entity defined by IGES is assigned a specific entity type number to refer to it in the IGES file.
- Entity numbers 1 through599 and 700 through 5000 are allocated for specific assignments. Entity type numbers 600 through 699 and 10000 through 99999 are for implementer-defined entities (via macro definitions). Entity type numbers 5001 through 9999 are reserved for macro entities.
- 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, co-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), subfigures and external cross reference entities (for surfaces and assemblies), Symbols (e.g., mechanical and electrical symbols), and macros (to define parametric parts).

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 typesin IGES arc 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.
- Each curve has a starting point, an ending point, and a parameter u. The information may not be enough to uniquely define the curve.
- As in the case of a circular or conic arc. Thus, some entity types refer to a "counterclockwise direction” with respect to a WCS. In IGES, the definition of this direction is based on an observer positioned along the positive Z axis of the WCS and looking down upon the XY plane of the WCS.
- IGES reserves entity numbers 100 to 199 inclusive for its geometric entities. Sample entity type numbers used by IGES are shown in Table.
- 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 to uniquely and completely define the entity.

IGES geometric entities

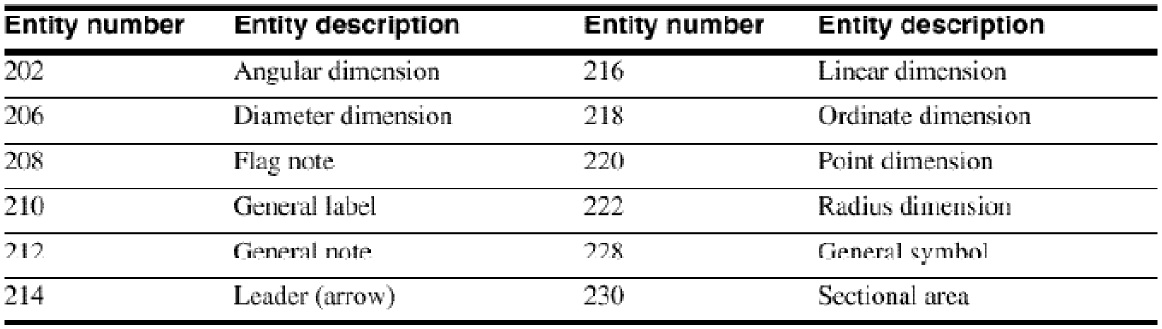


Entity number	Entity description	Entity number	Entity description
100	Circular arc	132	Connect point
102	Composite curve	136	Finite element
104	Conic arc	138	Nodal display and rotation
106	Copious data	140	Offset surface
108	Plane	142	Curve on a parametric surface
110	Line	144	Trimmed parametric surface
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	Flash	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

Annotation Entities

- Drafting data are represented in IGES via its annotation entities. Many TGES 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.

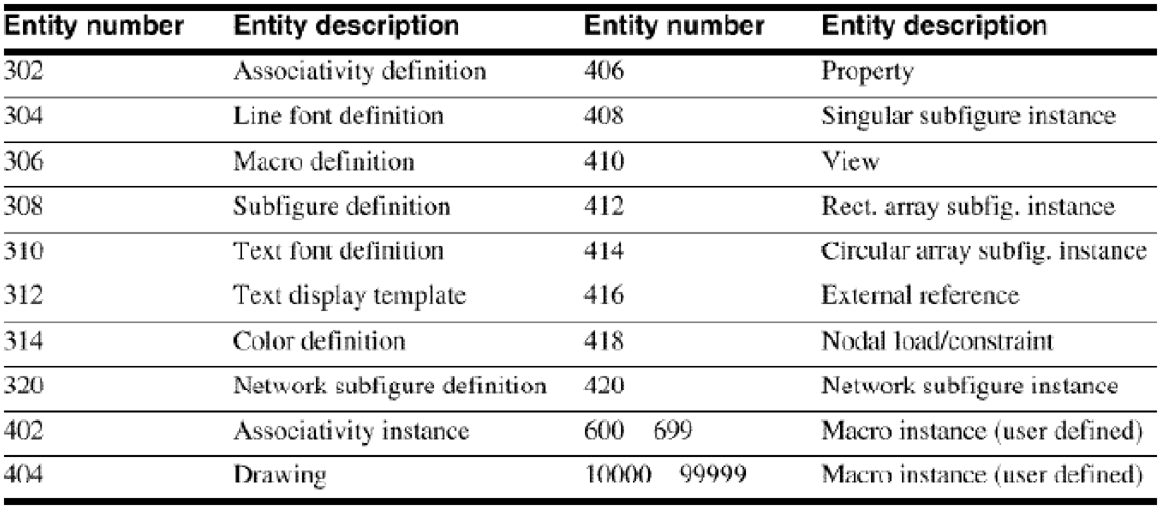
Table IGES annotation entities



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 permits a valuable set of product data to be represented via its structure entities.
- ➢ These entities include associatively, drawing, view, external reference, property, subfigure, macro and attribute entities.
- ➢ Attributes include line fonts, text fonts and color definition.

Table IGES structure entities



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 load/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)

- ➢ The associativity definition entity (type number 302) allows IGES to define a special relationship (called **associativity schema**) between various entities of a given model.
- ➢ The collection of entities that are related to each other via the associativity schema is called a **class**.

- Two kinds of associativity's are permitted within IGES. Predefined associativity's have (form) number 1 to 5000, and the second kind is implementer defined and has numbers 5001 to 9999.
- Each time an associativity relation is needed in the IGES file an associativity instance entity (type number 402) is used.
- The external reference entity (type number 416) enables IGES files to relate to each other.
- This entity provides a link between an entity in one file and the definition or a logically related entity in another file. Three forms of external reference entity are defined.

Form 0 is used when a single definition from the referenced file, which may contain a collection of definitions, is desired.

Form 1 is used when the entire file is to be instanced, which is the case where the referenced file contains a complete subassembly.

Form 3 is used when an entity in one file refers to another entity in a separate file. This is the case when each sheet of a drawing is a separate tile and, for example, a flange on one sheet mates with a flange on another sheet.

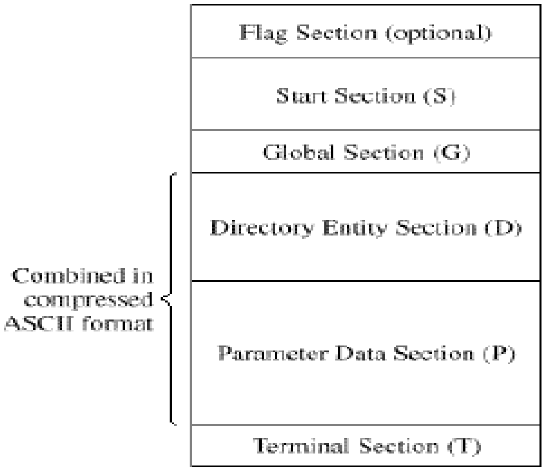
- The property entity (type number 406) in IGES contains numerical and textural data. Due to the wide range of properties, each one is assigned a form number and each form number may contain different property types (p types).

For example, form number 1 l contains tabular data that is organized under 11 p types. P types 1, 2, 3, 4, as an example refer to Young's modulus, Poisson's ratio, shear modulus, and material matrix, respectively. There are 17 form numbers that can be specified with the property entity.

- The macro capability in IGES (type number 306) allows the family of parts and/or entities grouped by the user for special purposes to be exchanged.
- Macros can only define a "new" entity in terms of the entities supported by IGES. This capability allows the extension of IGES beyond its common entity subset by utilizing a formal mechanism which is a part of IGES itself.

File Structure and Format

- A typical CAD/CAM system which supports IGES usually provides its users with two IGES commands.
- An IGES file consists of a sequence of records. Depending on the chosen file format, the record length can be fixed or variable. There are two different formats 10 represent IGES data in a file: ASCLI and binary.
- An IGES file has a specific structure. It consists of six sections which must appear in the following order: Flag section (optional), Start section (S), Global section (G), Directory Entry (D) section, Parameter Data (P) section, and Terminal section (T), as shown in Figure.
- The section code, also called the identification character (Column 73 of each record) for the IGES file sections. These codes are S. G, D. P. and T. The Flag section does not have a code.
- The **Flag** section is used only with compressed ASCII and binary format. It is a single record (line) that precedes the Start section in the IGES file with the character "C" in Column 73 to identify the file as compressed ASCII.
- The compressed ASCLI form is intended to be simply converted to and from the regular ASCII form.
- In the Binary file format, the Flag section is called the Binary information section, and the first byte (eight bits) of this section has the ASCII letter "B" as the file identifier.


Flag Section (optional)
Start Section (S)
Global Section (G)
Directory Entity Section (D)
Parameter Data Section (P)
Terminal Section (T)
Combined in compressed ASCII format

IGES file structure

- The **Start** section is a human-readable introduction to the file.
- It is commonly described as a "prologue" to the IGES file.
- This section includes user-relevant information, such as the name of the CAD/CAM system generating the lGES file, and a brief description of the product being converted. IGES does not specify how this section could be used.
- The **Global** section includes information describing the preprocessor and information needed by the post processorto interpret the file.
- The **Directory Entity** section is a list of all the entities defined in the IGES file together with certain attributes associated with them.
- The entry for each entity occupies two 80-character records, which are divided into a total of 20 8-character fields. The first and the eleventh (beginning of the second record of any given entity) fields contain the entity type number.
- The second field contains a pointer to the parameter data entry for the entity in the Parameter Data section.
- The pointer of an entity is simply its sequence number in the Directory Entry section. Some of the entity attributes specified in this section are line font, layer number. Transformation matrix, line weight and color.
- The **Parameter Data** section contains the actual data defining each entity listed in the Directory Entry section.
- For example: a straight line entity is defined by the six coordinates of its two end points. While each entity always has two records in the Directory Entry section, the number of records needed for each entity in the Parameter Data section varies from one entity to another (minimum is one record) and depends on the amount of data.
- Parameter data are placed in free format in columns 1 through 64. The parameter delimiter (usually a comma) is used to separate parameters, and the record delimiter (usually a semicolon) is used to terminate the list of parameters.
- Both delimiters are specified in the Global section of the IGES file. Column 65 is left blank.
- Columns 66 to 72 on all Parameter Data records contain the entity pointer specified in the first record of the entity in the Data Entry section.

- The **Terminate** section contains a single record which specifies the number of records in each of the four preceding sections for checking purposes.

3. Explain STEP architecture with neat sketch. (AU NOV 2016, AU MAY 2016)

STEP

- STEP (Standard for Exchange of Product data) is an exchange for product data in support of industrial automation.
- "Product data" is interpreted to be more general than the "product definition data" which forms the core philosophy of IGES.
- "Product data" encompasses data relevant to the entire life cycle of a product such as design, manufacturing, quality assurance, testing, and support.
- STEP is an ISO standard.
- It has absorbed PDES (Product Data Exchange Standard) which was an ANSI standard.
- In order to support industrial automation, STEP files are fully interpretable by computer.
- For example, tolerance information would be carried in a form directly interpretable by a computer rather than a computerized text form which requires human intervention to interpret.
- In addition this information would be associated with those entities in the model affected by the tolerance.
- Thus the general emphasis of STEP is to eliminate the human presence from the "product data" exchange; that is, to obviate the u e of engineering drawings and other paper documents as necessary means of passing information between different product phases that may be performed on similar or dissimilar CAD/CAM systems.
- There is a fundamental difference in philosophy between exchanging data in lGES and in STEP.
- The central unit of data exchange in the lGES model is the entity, while the central unit of data exchange in the STEP model is the application, which contains various types of entities.

- Therefore, when data is exchanged between systems, it is done in terms of "application" units.
- This approach maintains all the meaningful associativity's and relationships between the application entities which make industrial automation possible.
- The **goal of STEP** is tore present all product information in a common data format, throughout a product's entire life cycle.
- The data being transferred is geometry (e.g., curves, surfaces, solids) but also analysis, manufacturing, implementation, and testing procedures.
- Essentially, STEP is a common structure, operating as a template, for sharing data among multiple users, across all functional areas.

4. Write short notes on STEP in details. (AU MAY 2016, NOV 2016)

STEP (Standard for the Exchange of Product model Data)

- Standard for Exchange of Product Model Data
- Uses a formal model for data exchange
- Information is modeled using the EXPRESS language
- EXPRESS has elements of Pascal, C, and other languages
- It contains constructs for defining data types and structures, but not for processing data
- EXPRESS describes geometry and other information in a standard, unambiguous way

Classes of STEP Parts

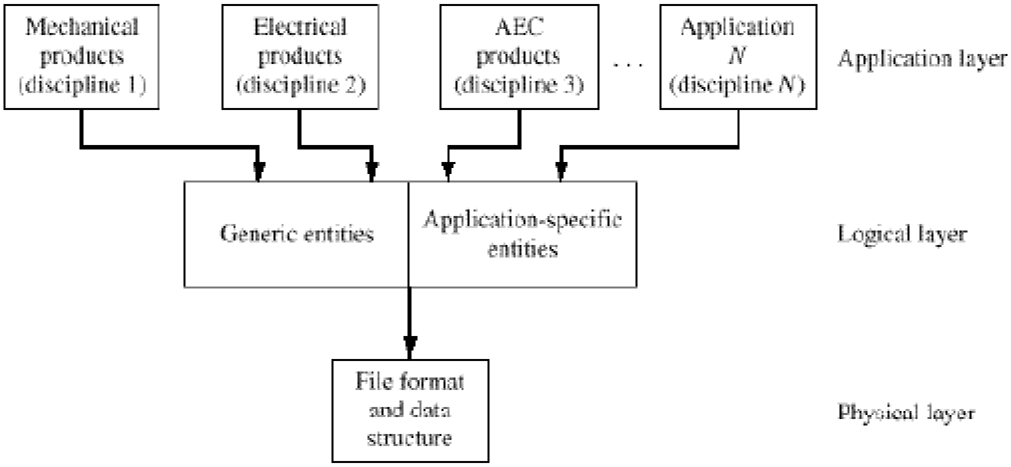
- Introductory
- Description methods
- Implementation methods
- Conformance testing methodology and framework
- Integrated resources
- Application protocols
- Abstract test suites
- Application interpreted constructs

Architecture

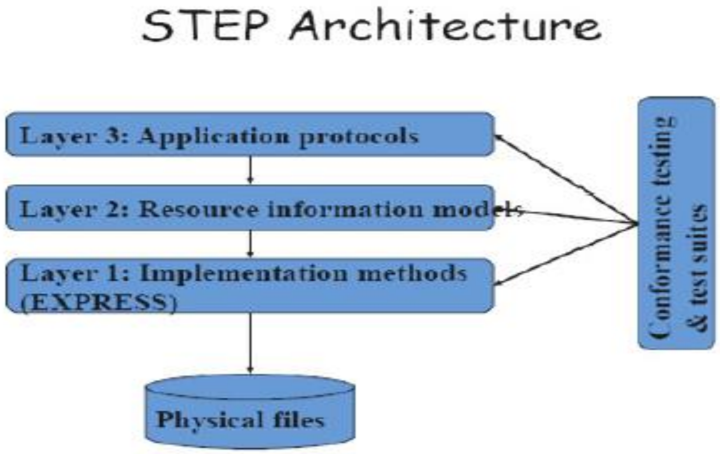
- To achieve the STEP philosophy, product data is exchanged by STEP according to "discipline models" or "schemas."
- Both the sender who originates the discipline model and the receiver of the model must be aware of the meaning of the discipline model being exchanged in order to recover the correct meaning of data in the exchange.
- Discipline models are standardized and defined by STEP to be interpreted and used by another computer. Examples are the Mechanical Products discipline model, Electrical Produces discipline model, and ABC discipline model.
- Thus the concept of discipline models makes STEP flexible to accommodate any future models and application areas when they become available.

STEP uses three-layer architecture, as shown in Figure that forms the core of STEP structure.

- The three layers are the application layer, the logical layer and the physical layer.
- The application layer is the interface between the user and STEP.
- It contains all the descriptions and information of various application areas. The purpose of the logical layer is to provide a consistent, computer-independent description of the data constructs that contain the information co be exchanged.
- The physical layer deals with the data structures and data format for the exchange file itself. The main goal here is to establish and maintain efficiency in the file size and processing time.


Mechanical products (discipline 1)
Electrical products (discipline 2)
AEC products (discipline 3)
. . .
Application N (discipline N)
Application layer
Generic entities
Application-specific entities
Logical layer
File format and data structure
Physical layer

STEP three-layer architecture.


STEP Architecture
Layer 3: Application protocols
Layer 2: Resource information models
Layer 1: Implementation methods (EXPRESS)
Physical files
Conformance testing & test suites

Implementation

- STEP is built on a data exchange language, called EXPRESS, to formally describe a model and the file formal that stores it. EXPRESS store both the model data and semantics.
- The basic unit in EXPRESS is the entity.
- An **entity** is a collection of data, constraints, and operations. The operations work on data. A set of entities make up a model.
- The relationships (semantics) between model entities are carried over, and maintained by STEP, from the native CAD database of the model.
- EXPRESS follows the object oriented (00) paradigm. It first defines an object schema and then defines instances of such object.
- Consider the example of a circle. EXPRESS define the circle (the object) schema as its center and radius.
- The center is a point, and the radius is a real number. EXPRESS defines a point schema as three numbers for the .r, y, and z coordinates. The circle center is an instance of the point object.
- There are several sections within STEP, called Application Protocols (APs), which are built into a common data model.
- These APs include definitions not only of typical geometry and drafting elements, but also of data types and processes for specific industries such

as automotive, aerospace, shipbuilding, electronics, plant construction and maintenance.

- Sample APs are AP203 (STEP format to save solid models) and AP210 (STEP format to save electronics).
- AP203 is further divided into classes for defining wireframe geometry, surfaces and solid modeling.
- One of the latest significant developments in STEP is the recent agreement to provide mapping to XML (extensible markup language).
- This new technology is rapidly becoming the preferred method for complex data access on the Web.
- The flexibility and growing availability of commercial Web/XML tools with STEP greatly increases the sharing of information across disciplines, with universal access.
- XML lends itself well to STEP. XML is a standard format for data representation. It complements XHTML (extensible hypertext markup language) which is a standard for data presentation. XML defines data schema in a DTD (document type definition) file.

ACIS and DXF

- ACIS is used as a kernel in a number of commercial CAD/CAM systems. Spatial, the maker of ACIS provides a translator for these systems use.
- Spatial translator allows the exchange of solid. surface, and wireframe data via a variety of neutral and native formats, including IGES, STEP, Pro/E, SolidWorks, CATIA, Para solid (PS), Unigraphics (UG), and Inventor. The major systems therefore, offer theseformat in their translation menus.
- DXF (Data exchange Format) is standard due to its popularity.
- DXF is an AutoCAD format. Autodesk 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.
- A DXF file consists of four sections: Header, Tables. Blocks, and Entities.
- The Header section includes the AutoCAD system settings such as dimension style and layers.
- The Tables section includes line styles and user-defined coordinate systems. The Blocks section includes drawing blocks (instances). The Entities section includes entity definition and data.

5. Write short note on CALS in details.

CONTINUOUSACQUISITION AND LIFE-CYCLE SUPPORT (CALS)

- Developed by US Department of Defense
- Prescribes formats for storage and exchange of technical data
- Technical publications an important focus

Important CALS Standards

- Standard Generalized Markup Language (SGML)-developed in 1960sIBM

i. document description language

ii. Separates content from structure (formatting)

iii. Uses “tags” to define headings, sections, chapters etc.

iv. HTML is based on SGML

• Computer Graphics Metafile (CGM)

i. Developed in 1986

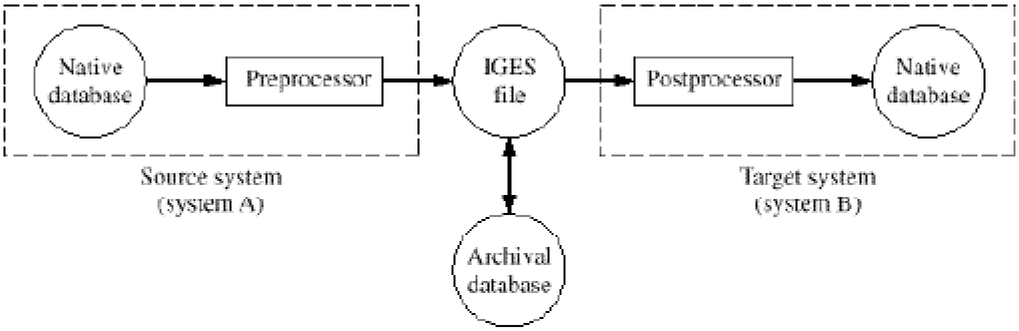
ii. Vector file format for illustrations and drawings

iii. All graphical elements can be specified in a textual source file that can be compiled into a binary file or one of two text representations

6. **Explain in detail on Processors and its types.**

Processors

- A standard, such as IGES or STEP, in itself is just a document describing what should go into a data file.
- Interested developers (CAD/CAM vendors or companies specialized in database transfer) must interpret Understand, and implement the standard into programs, often called processors or translators.
- The processors translate from their systems to the standard format and vice versa. The software that translates from the native file format of a given CAD/CAM system to a standard format is called a preprocessor.
- The software that translates in the opposite way (from a standard to a CAD/CAM system) is called a postprocessor.
- The user interface to access these processors usually takes the form of simple commands, accompanied by proper dialogues.
- Figure shows file exchange using a translator.
- The source system is the originating or sending CAD/CAM system, and the target system is the receiving one.
- The archival database is a side benefit of using standards. Such archived databases could be kept for as long as needed.
- If system B in Figure becomes the source and system A becomes the target, the processors reverse positions.



Pre-and Postprocessors of a translator

Design and Implementation

- Designing and writing processors is a significant challenge. A typical database might contain many instances of many entity types.
- Many of these entity types involve complex mathematics and complex data structures.
- Problems in writing a processor relate to the definition and format of the standard itself. Some of these problems are:

1. **Entity set**. Any standard (IGES or STEP) does not and cannot contain a real superset of entities which are found in all of today's CAD/CAM systems.

The standard may contain an entity which has no equivalence on a specific CAD/CAM system. Or, the system may contain an entity for which no standard entity exists.

A processor could either ignore translating the entity, or translate it into a similar one.

2. **Format.** While a standard allows exchanging complex structures and relationships, its format must be processible by a wide range of different computer systems and therefore can only use simple data formats and management methods known to these systems and, in the meantime independent of any system specifics.

3. **Limitations of individual CAD/CAM systems.** These limitations are based on specific systems and are related to things such as model size, model space and data precision.

The designing of processors with all the preceding problems in mind divides into the following steps:

1. **Analyze and tabulate entity characteristics.** This step involves the study of the entity mathematical representations utilized by both the standard and the CAD/CAM system. In many cases, an entity can be represented by a number of nearly, but not completely, equivalent methods.

2. **Define conversion algorithms.** Step I clearly provides the information required to design the proper conversion algorithm to convert an entity to and from the standard.

3. **Develop a complete specification of the processors.** Steps 1 and 2 form the core of the design process of processors. Once completed, other specifications of the processors

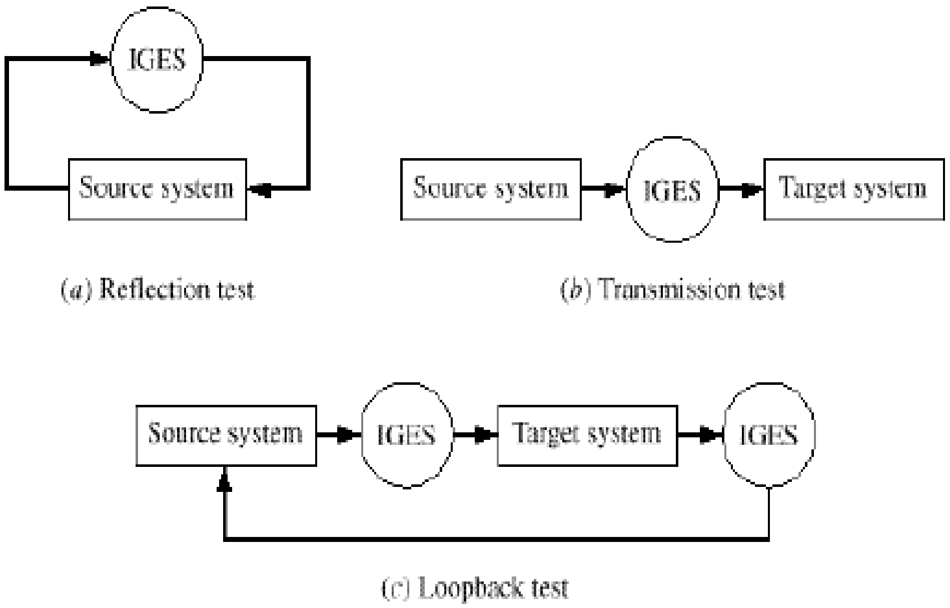
must be developed. These include the standard revision that the processor ought to support, the subset of the standard entities it can support, and the user interface of the processor.

4. Design verification procedures. Careful verification of processors is very important because processors operate at the interface between different organizations and vendors.

- Processors must be verified by constructing test data running it through the processors, and comparing the actual results with those expected.
- Ideally two sets of test data would be required: a set for implementations to use during processor development, and a more comprehensive set for final processor verification.
- In addition more customized tests for specific user requirements can also be developed in collaboration between users and implementors of the standard.

Testing and Verification

- There is an IGES Test, Evaluate and Support Committee whose function is to provide test data.
- There is an IGES test library prepared by the committee which allows testing of the basic implementation of an entity.
- However the library does not allow the checking of the variations that occur in production data due to numerical and computational errors.
- These variations must be tested by implementors and users themselves **Verification** of the results of a processor is a time-intensive task.
- In most cases, it is not sufficient to check converted models visually; more comprehensive tests are needed. The common methods of testing are shown in Figure.



(*a*) Reflection test

(*b*) Transmission test

(*c*) Loopback test

Methods of verification of processors.

They are:

1. **Reflection test.** In this test, a neutral file created by a translator preprocessor is read by its own postprocessor to create a native file of the translated model.

This test is used to establish that a translator’s processors could read and write common entities making them symmetric.

2. **Transmission test.** Here a neutral file of a model created by the preprocessor of a source system is transferred to a target system whose post processor is used to re-create the model on the target system.

This test essentially determines the capabilities of the preprocessor and the postprocessor of the source and the target systems, respectively.

3. **Loopback test.** In this test, a neutral file created by the source system is read by the target system which in turn creates another neutral file and then transfers this file back to the source system to read it.

This test checks the pre- and postprocessors of both the source and the target systems.

Error Handling

Error handling and reporting when processing a neutral file is important. There are two major error sources when processing IGES files: programming errors in the processor, and misinterpretation of the standard itself.

These sources apply to both pre- and postprocessors. The way a processor reports these errors and the information given with these reports, determine whether the correction of an error becomes a laborious task or not.

The preprocessor should report the entity type, number of unprocessed entries, reasons for un processing, and other relevant database information of these unprocessed entities.

On the other hand, the postprocessor should report the number of unprocessed entities, their types, their forms, their record numbers in the Directory Entry and Parameter Data sections, and the reasons for un processing.

It should also report any invalid or missing data encountered in reading neutral files, especially those that were edited.

7. **Discuss what is Collaborative design (AU NOV 2016, AU APR 2017)**

Or

Explain in detail the communication standards

Or

Explain the principles and protocols for communication standards Collaborative design

- Collaborative design is an interactive process of real-time communication between members of a design team who are physically in different locations.
- During this process, several engineers or a team of designers are simultaneously involved to agree or disagree on design issues and address them during the early phase of product design.
- Concurrent engineering is considered the grander philosophy that motivates collaborative design as a key ingredient.

- Collaborative design fosters the ability to conceptually design a product in a distributed manner by involving supply-chain and other members who have the expertise to accomplish the design task at hand.
- It has the advantages over traditional design, of increasing productivity, shortening product development lifecycle, and improving competitiveness.
- The **key advantage of collaborative** over traditional design is real-time communication among team members regardless of where they are Collaborative design is the closest thing to face-to-face meetings between team members in the same room.
- Collaborative design is more than just exchanging documents between designers on time.
- When there is real collaboration in design, there is a common design goal to achieve in real time.
- All design partners make effective use of each other's insight and talent. Teams must be stimulated in creativity, and collaboration must be organized both synchronously and asynchronously.

Distributed Computing

- Distributed computing is the pervasive computing model today.
- It is based on the client/server model that is central to the Internet's distributed network hardware/communication and services.
- The client/server model uses two computer programs installed on two separate, but networked, computers at different locations.
- The **client** program (or client, for short), installed on one computer (known as the client computer) communicates with the **server** program (or server, for short) installed on the other computer, as shown in Figure.
- The client requests services from the server. While the server is running, it acts as a daemon. A **daemon** is a process that awaits(listens to) client requests.
- When it receives a request from a client, it grants the request and sends a response back to the client.

Instant Messaging

- Communication via the Web has two modes: synchronous or asynchronous.
- **Synchronous communication** is an activity in which the two communicating persons send messages to (or receive them from) each other simultaneously, in real time.

Examples include instant messaging (IM) and chatting.

- **Asynchronous communication** is an activity in which the two communicating persons send messages to (or receive them from) each other independently, at different times. Examples include e-mail and newsgroups.

8. Explain the widely using communication tool for 3D model

Virtual Reality Modeling Language

- Virtual reality modeling language (VRML) lends itself well to collaborative design and CAD/CAM as a useful communication tool for 3D models.
- VRML is a graphics system that intermediates communication between output devices (graphics displays) and the application model (CAD part file).
- Collaborative design tools convert a CAD file into a VRML file that can be rendered in a Web browser to serve as the center of discussion between team members during a real-time design session.
- The resulting VRML file does not require a CAD system. It can be displayed in a Web browser that has a VRML plug-in installed.
- Team members can rotate the VRML model and mark it up in real time.
- VRML uses orientations viewpoints, and transformations to move objects in space. It uses polygons and surfaces to describe the objects.
- It uses different types of light sources (such as ambient light, directional lights, spotlights. and point lights) to render objects.
- It also uses shading techniques (flat Gourmand and Phong) and ray-tracing algorithms to add realism to models. It also uses texture mapping.
- VRML files use the extension as the file type.

- A VRML file stores a scene graph that describes a 3D scene. VRML resembles to a great extent Java 30 and uses the object concepts.
- VRML uses the instance concept when a scene uses the same basic shape (primitive) multiple times.
- Different types of VRML nodes exist. **Appearance nodes** describe the colors, shading and lighting in the scene.
- Geometry nodes describe the shapes used in the scene. The shapes that VRML uses are the solids primitives. **Grouping nodes** describe geometric transformations.

Traditional Design

- Traditional design is a serialized. It begins with the design need and ends with the documentation of the design.
- In between come activities such as developing design requirements and specifications generating model geometry.
- Performing modeling and simulation, carrying detailed analysis and design optimization, and evaluation.
- While concepts such as DFA (design for assembly), DFM (design for manufacturing), DFX (design for anything), and concurrent engineering have helped to improve and shorten the design process. it still largely remains a serialized activity.
- The main problem with the traditional design approach is the lack of collaborative and continuous communication between design teams and members on a daily, or even hourly basis leading up to crucial deadlines

Collaborative Design

- A collaborative design environment provides a decision-support and resource management system that interconnects computational resources (CAD tools, design tools, resource planning, project management tools, etc.) and users.
- It also provides a communication infrastructure to enable distinct resources in different geographical locations and users to seamlessly collaborate and solve problems.

- By linking these tools within such an environment, they become available to users, regardless of where the tools and/or users are located.
- The main difference between collaborative and traditional design is that the former make the communication feasible and seamless anytime.
- We expect that it will take time for the philosophy of collaborative design to catch on and become a mainstream daily design tool and activity.

Collaboration Principles

A list of some principles:

l. **Ownership control:** Each group that generates design information should own andcontrol it.

This implies that the group is responsible for the updates and maintenance of the information.

After a collaborative design is over the group should organize and store the information and stand by ready for any other change that are deemed necessary by future collaboration.

2. **Distribution control:** Only the required information should be provided by group owners, and only when needed.

This help prevents "choking" other designers by overwhelming them with too much design knowledge when they do not need it.

3. **Damage control**. This principle relates to information “defects." What happens if wrong information (known as information defects) is generated inadvertently during a session of collaboration?

These defects should not be propagated through future sessions. These defects should be brought to the attention of its owners to correct as soon as possible to prevent extensive future damage.

4. **Access control.** Information should be provided only to those who need it. Information that does not concern a design group should not be given to it to avoid confusing it.

This is different from distribution control. Distribution and access control complement each other. After we identify a design group that should get design information (access control here), we apply the principle of distribution control to this design information.

5. **Guidelines control.** Designers should not feel constrained during collaborative design sessions.

It is imperative to the success of collaboration that designers feel free, within reasonable limits, to share and exchange design information with other designers, especially those from other organizations.

In this regard, "soft" design guidelines should benefit most cases and should avoid difficult exception cases.

On the other hand, organizations should avoid imposing “hard" design constraints such as "it must be clone this way, and only this way."

Collaboration Approaches

The three approaches are:

l. Collaboration **based on design methodologies.** This approach focuses on generating formalized design methodologies by investigating how the technical design decisions are made.

The design process models are often implied in these design methodologies and theories.

These models include the systematic design model, axiomatic design model, Quality function deployment (QFD), general design theory, and so forth. Basically, these design methodologies provide the guidelines for making technical decisions more consciously and systematically.

Collaborative design tools that belong to this approach would implement some of these design models in tools and software systems.

Designers can focus on this aspect of design during a collaboration session by using these tools and systems.

2. **Collaboration based on design workflow**. This approach views the design process as a workflow with task dependencies and information exchange.

The methodologies and techniques of this approach come mainly from research in business operation and project management.

From this aspect, engineering design is viewed as an information-driven process among design activities.

The design organization is viewed as a stochastic processing network in which engineering resources are workstations that perform different tasks.

The design tasks themselves are jobs that flow among the workstations. With workstations and tasks in mind, a set of techniques to manipulate the design activities are developed to aid during collaborative design sessions.

Such techniques include signal flow diagrams. Design structure matrices, and design process networks.

3. **Collaboration based on CAD/CAM/CAE.** Several methodologies from CAD, CAM, and CAE (computer-aided engineering) areas view collaborative design as individuals and groups accessing data and sharing design information.

The design process is specified accordingly as the management of the product data at different abstraction levels.

During this management, the technological, scientific, and interdisciplinary dependencies of the design information could be established, handled and maintained by a product model the CAD model.

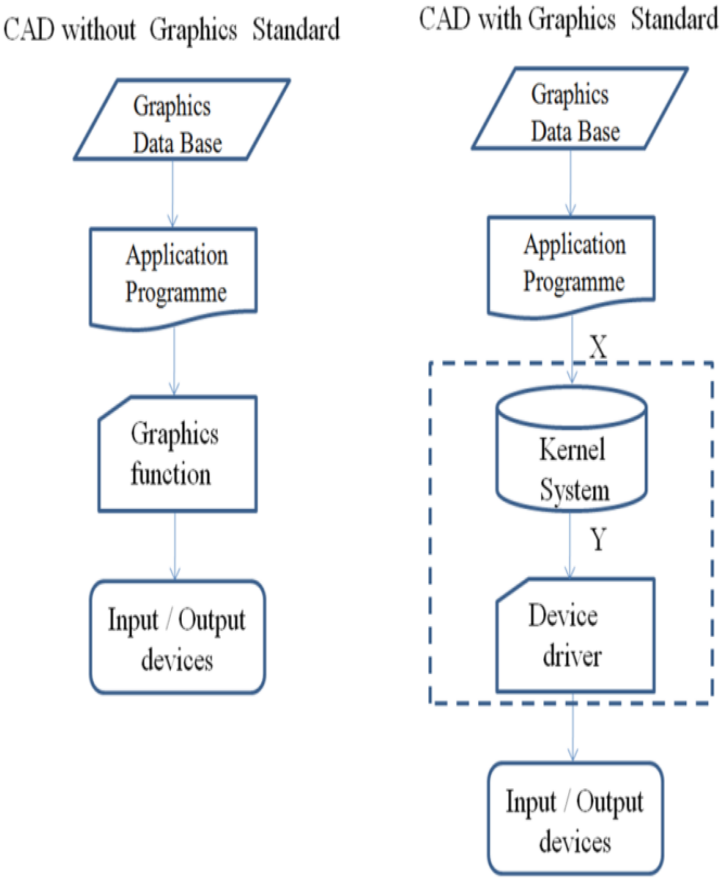
Information systems are built around these models to support the storage and processing of various types of interest to designers.

These systems use IT (information technologies) and CAD/CAM/CAE techniques. PLM (product lifecycle management) systems are built based on this TT approach.

CAD Standards

The purpose of CAD standard is that the CAD software should not be device-independent and should connect to any input device via a device driver and to any graphics display via a device drive.

The graphics system is divided into two parts: the kernel system, which is hardware independent and the device driver, which is hardware dependent.



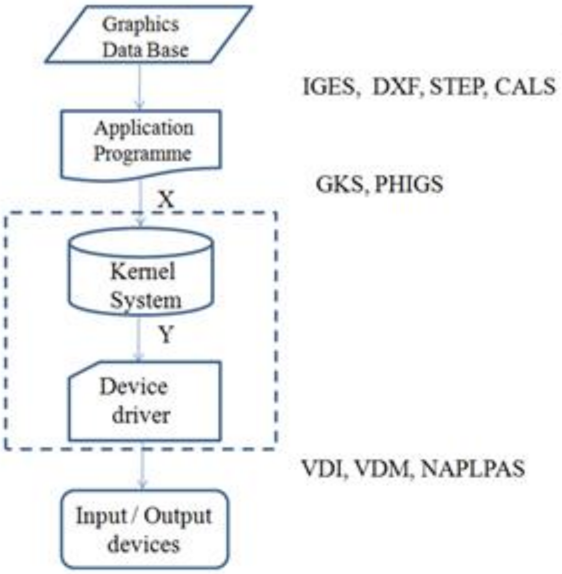
9. **Explain the graphics standard in detail**

GRAPHICS STANDARDS

Various standards in graphics programming

The following international organizations involved to develop the graphics standards:

- ACM(Association for Computer Machinery)
- ANSI(American National Standards Institute)
- ISO(International Standards Organization)
- GIN(German Standards Institute)



Graphics Standards in Graphics Programming

As a result of these international organization efforts, various standard functions at various levels of the graphics system developed. These are:

1. IGES (Initial Graphics Exchange Specification) enables an exchange of model data basis among CAD system.
2. DXF (Drawing/Data Exchange Format) file format was meant to provide an exact representation of the data in the standard CAD file format.
3. STEP (Standard for the Exchange of Product model data) can be used to exchange data between CAD, Computer Aided Manufacturing (CAM) , Computer Aided Engineering(CAE), product data management/enterprise data modeling(PDES) and other CAx systems.
4. CALS (Computer Aided Acquisition and Logistic Support) is an US Department of Defense initiative with the aim of applying computer technology in Logistic support.
5. GKS (Graphics Kernel System) provides a set of drawing features for two-dimensional vector graphics suitable for charting and similar duties.
6. PHIGS (Programmer’s Hierarchical Interactive Graphic System) The PHIGS standard defines a set of functions and data structures to be used by a programmer to manipulate and display 3-D graphical objects.
7. VDI (Virtual Device Interface) lies between GKS or PHIGS and the device driver code. VDI is now called CGI (Computer Graphics Interface).

8. VDM (Virtual Device Metafile) can be stored or transmitted from graphics device to another. VDM is now called CGM (Computer Graphics Metafile).

9. NAPLPS (North American Presentation-Level Protocol Syntax) describes text and graphics in the form of sequences of bytes in ASCII code.

10. Explain various layers of GKS. (AU NOV 2016, AU MAY 2016)

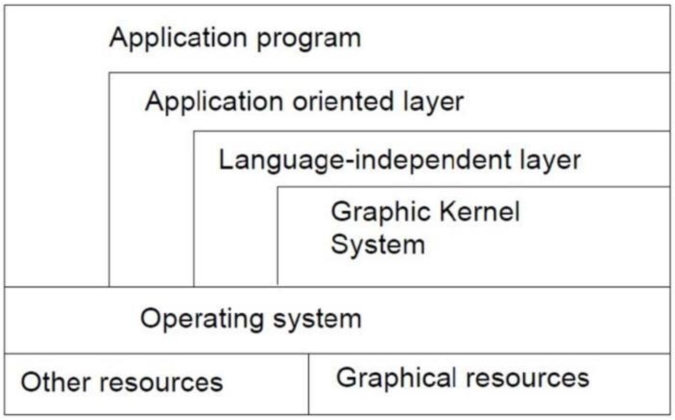
Or

What is Graphics kernel system (GKS)? Explain with a suitable example.

Graphics Kernel System (GKS)

The Graphical Kernel System (GKS) was the first ISO standard for computer graphics in low- level. GKS offers a group of drawing aspects for 2D vector graphics appropriate for mapping and related duties.

The calls are defined to be moveable across various programming languages, graphics hardware, so that applications noted to use GKS will be willingly portable to different devices and platforms.



LayersofGKS

- The following documents are representing GKS standards:
- The language bindings are called in ISO 8651 standard.
- ANSIX3.124 (1985) is part of ANSI standard.
- ISO/IEC7942 noted in ISO standard, first part of 1985 and two to four parts of 1997-99.

- ISO8805 and ISO 8806.

The main uses of the GKS standard are**:**

- To give for portability of application graphics programs.
- To assist in the learning of graphics systems by application programmers.
- To offer strategy for manufacturers in relating practical graphics capabilities.

The GKS consists of three basic parts:

i) A casual exhibition of the substances of the standard which contains such things a show text is placed, how polygonal zones are to be filled, and so onward.

ii) An official of the **descriptive** material in (i), by way of conceptual the ideas in to separate functional explanations.

iii) These functional descriptions have such data as description so input and output parameters, specific descriptions of the result of every function should have references into the **descriptive** material in (i), and a description of fault situation. The functional descriptions in this division are language autonomous.

iv) Language bindings are an execution of the abstract functions explained in (ii) in a explicit computer language such as C.

GKS arrange its functionality in to twelve functional stages, based on the complexity of the graphical input and output.

There are four stages of output (m,0,1,2) and three stages of input (A,B, C). NCARGKS has a complete execution of the GKSC bindings at level 0 A.

GKS Output Primitives

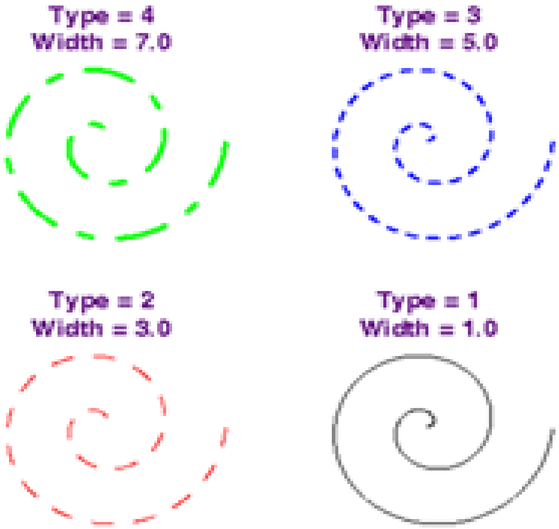
GKS is based on a number of elements that may be drawn in an object know as graphical primitives.

The fundamental set of primitives has the word names POLYLINE, POLYMARKER, FILLAREA, TEXT and CELLARRAY, even though a few implementations widen this basic set.

i) POLYLINES

The GKS function for drawing line segments is called 'POLYLINE'. The 'POLYLINE' command takes an array of X-Y coordinates and creates line segments joining them. The elements that organize the look of a 'POLYLINE' are (Fig):

- Line type : solid, dashed or dotted.
- Line width scale factor: thickness of the line.
- Polyline color index : color of the line.

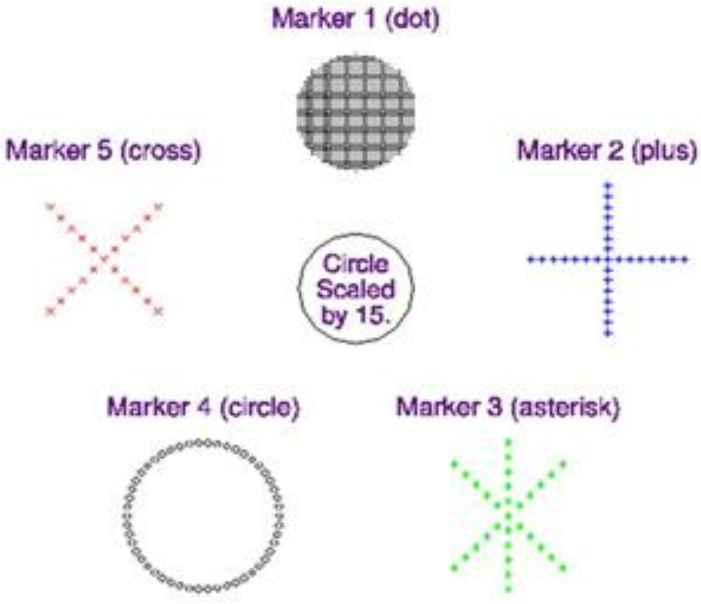


GKSPOLYLINES

ii) POLYMARKERS

The GKS 'POLYMARKER' function permits to draw symbols of marker centered at coordinate points. The features that control the look of 'POLYMARKERS' are (Fig.):

Marker characters	:	dot, plus, asterisk, circle, or
Marker size scale factor	:	size of marker
Polymakers color index	:	color off the marker



GKS POLYMARKERS

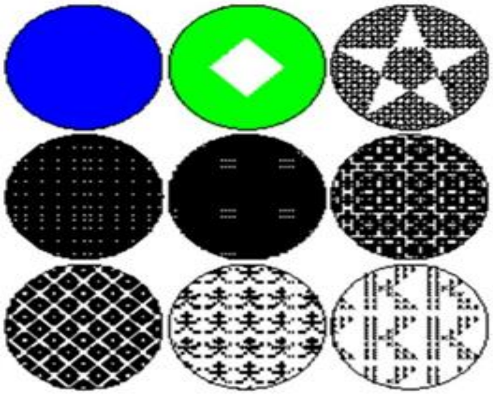
iii) FILLAREA

The GKS 'FILL AREA' function permits to denote a polygonal shape of a zone to be filled with various interior shapes. The features that control the look of fill areas are (Fig.):

FILL AREA interior style : solid colors, hatch patterns.

FILL AREA style index : horizontal lines; vertical lines; left slant lines; right slant lines; horizontal and vertical lines; or left slant and right slant lines.

Fill area color index : color of the fill patterns/solid areas.

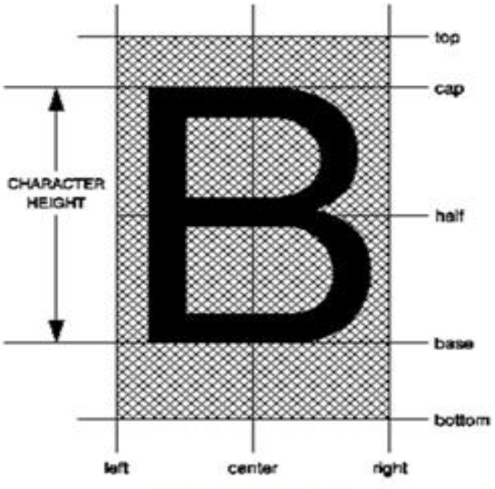


GKS FILLAREA

iv) TEXT

The GKSTEXT function permits to sketch a text string at a specified coordinate place.The features that control the look of text are:

- Text font and precision : text font should be used for the characters
- Character expansion factor : height-to-width ratio of each character.
- Character spacing : additional white space should be inserted between characters
- Text color index : color the text string
- Character height : size of the characters
- Character up vector : angle the text
- Text path: direction the text should be written (right, left, up, or down).
- Text alignment : vertical and horizontal centering options for the text string.

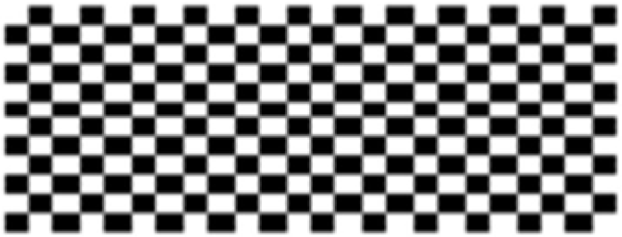


GKS TEXT

v) CELL ARRAY

The GKS CELL ARRAY function shows raster like pictures in a device autonomous manner.

The CELL ARRAY function takes the two corner points of a rectangle that indicate, a number of partitions (M) in the X direction and a number of partitions (N) in the Y direction. It then partitions the rectangle into MxN sub rectangles noted as cells.



GKS CELL ARRAY

Standard for exchange images

A graphics standard proposed for interactive Three Dimensional applications should assure different criteria.

It should be introduced on platforms with changing graphics abilities without sacrificing the graphics quality of the primary hardware and without compromising control over the hardware's function.

It must offer a normal interface that permits a programmer to explain rendering processes quickly.

To end with, the interface should be flexible adequate to contain additions, hence that as new graphics operations become important, these operations can be given without sacrificing the original interface.

Open GL meets these measures by giving a simple interface to the basic operations of 3D graphics rendering.

It supports basic graphics primitives, basic rendering operations and lighting calculations. It also helps advanced rendering attributes such as texture mapping.

11. Explain OpenGL with schematic diagram. (AU NOV 2015)

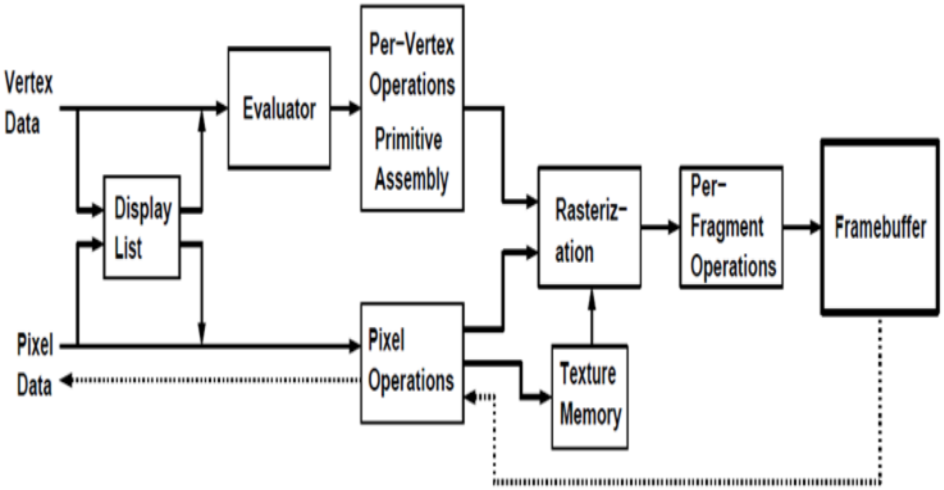
Or

Explain briefly about Open graphics Library Open Graphics Library

Open GL draws primitives into a structured buffer focus to a various electable modes. Every Point, line, polygon, or bitmap are called as a primitive.

Each mode can be modified separately; the parameters of one do not affect the parameters of others.

Modes defined, primitives detailed, and other Open GL operations explained by giving commands in the form of procedure calls.



Schematic diagram of Open GL

Figure shows a schematic diagram of Open GL.Commands go into Open GL on the left. The majority commands may be collected in a 'displaylist' for executing at a later time. If not, commands are successfully sent through a pipe line for processing.

The first stage gives an effective means for resembling curve and surface geometry by estimating polynomial functions of input data.

The next stage works on geometric primitives explained by vertices. In this stage vertices are converted, and primitives are clipped to a seeing volume in creation for the next stage.

All ‘fragment’ created is supplied to the next stage that executes processes on personal fragments before they lastly change the structural buffer.

These operations contain restricted updates into the structural buffer based on incoming and formerly saved depth values, combination of incoming colors with stored colors, as well as covering and other logical operations on fragment values.

To end with, rectangle pixels and bitmaps by pass the vertex processing part of the pipeline to move a group of fragments in a straight line to the individual fragment actions, finally rooting a block of pixels to be written to the frame buffer.

Values can also be read back from the frame buffer or duplicated from one part of the frame buffer to another.

Features of Open GL

i)Based on IRIS GL

Open GL is supported on Silicon Graphics’ Integrated Rater Imaging System Graphics Library (IRIS GL).

Though it would have been potential to have designed a totally new Application Programmer’s Interface (API), practice with IRIS GL offered insight into what programmers need and don’t need in a Three Dimensional graphics API.

Additional, creation of OpenGL similar to Integrated Rater Imaging System Graphics Library where feasible builds OpenGL most likely to be admitted; there are various successful IRISGL applications, and programmers of IRISGL will have a simple time switching to Open GL.

ii) Low-Level

A critical target of Open GL is to offer device independence while still permitting total contact to hardware.

Therefore the API gives permission to graphics operations at the lowest level that still gives device independence.

iii) Fine-Grained Control

Due to minimize the needs on how an application utilizing the Application Programmer's Interface must save and present its information, the API must give a suggestion to state entity parts of geometric entities and operations on them.

This fine-grained control is necessary so that these mechanism and operations may be defined in any order and so that control of rendering operations is comfortable to contain the needs of various applications.

iv) Modal

A modal Application Programmer's Interface arises in executions in which processes function in parallel on different primitives.

In that cases, a mode modify must be transmit to all processors so that all collects the new parameters before it processes its next primitive.

A mode change is thus developed serially, stopping primitive processing until all processors have collected the modifications, and decreasing performance accordingly.

v) Frame buffer

Most of Open GL needs that the graphics hardware has a frame buffer. This is a realistic conditions in ceil most all interactive graphics run on systems with frame buffers.

vi) Not Programmable

Open GL does not give a programming language. Its function may be organized by turning actions on or off or specifying factors to operations, but the rendering algorithms are basically fixed.

One basis for this decision is that, for performance basis, graphics hardware is generally designed to apply particular operations in a defined order; changing these operations with random algorithms is generally infeasible.

Programmability would variance with maintenance of the API close to the hardware and thus with the objective of maximum performance.

vii) Geometry and Images

Open GLgivessupportformanagingboth3Dand2D geometry.

An Application Programmer's Interface for utilize with geometry should also give guidance for reading, writing, and copying images, because geometry and images are regularly joint, as when a Three Dimensional view is laid over a background image. Various per-fragment processes that are applied to fragments beginning from geometric primitives apply uniformly well to fragments corresponding to pixels in making it simple to mix images with geometry.

UNIT-V
CAD STANDARDS

PART-A

1. **What are the basic elements associated with a CAD system?**

   i) Operator (user) ii) Graphics support system iii) other user interface support system iv).Application functions v) Database

2. **What are the reasons for evolving a graphic standard? (AU NOV 2016)**

   i) Need for exchanging graphic data between different computer systems.

   ii) Need for a clear distinction between modeling and review aspects.

3. **What are the primary purposes of the Graphical Kernel System (GKS).**

   (i) To provide for portability of graphics application programs.

   (ii) To aid in the understanding of graphics method by application programs.

   (iii)To provide the guidelines for manufacturers in describing useful graphics capabilities.

4. **Define Open Graphics Library.(AU APR 2017)**

   - OpenGL (Open Graphics Library)is a cross-language, multi-platform application programming interface (application program interface-API) for rendering2D and 3D vector graphics.
   - The API is typically used to interact with a graphics processing unit (GPU), to achieve hardware-accelerated rendering.

5. **Define Data exchange standards in CAD.(AU NOV 2015)**

   - CAD data exchange involves a number of software technologies and methods to translate data from one Computer-aided design system to another CAD file format.
   - This PLMtechnology is required to facilitate collaborative work (CPD) between OEMs and their suppliers.

6. **Name the basic methods of transferring data. (AU MAY 2015)**

There are basically three methods of transferring data from one CAD system to another.

1. Direct CAD system export/import
2. Direct 3rd party translators.
3. Intermediate data exchange formats.

7. **What is the use of IGES system? (AU NOV 2016)**

- ➢ The Initial Graphics Exchange Specification (IGES) is a file format which defines a vendor Neutral data format that allows the digital exchange of information among design (CAD) systems.

8.What are the three types of data types in IGES?

i) Geometric

ii) Annotation

iii) Structure

9. **Mention the different file section in IGES**

i) Flag section(optional) ii) Start section(S) iii) Global section(G) iv) Directory entry(D)section v)Parameter Data section vi) Terminal section

10. **What is the use of STEP data exchange format?(AU NOV 2015)**

- ➢ STEP (Standard for the Exchange of Product model data) can be used to exchange data between CAD, computer aided manufacturing,computer-aided engineering, product data management/enterprise data modelingand other CAx systems.
- ➢ STEP is addressing product data from mechanical and electrical design, geometric dimensioning and toleranceanalysis and manufacturing, with additional information specific to various industries such as automotive, aerospace, building construction, ship, oil and gas, process plants and others.

11. **What are the different parts of STEP?**

i) Environment ii) Integrated data models iii) Top parts

12. Mention the requirements of a conformant STEP application

i) Implementation of either a preprocessor or a postprocessor or both.

ii) Using one of the STEP implementation methods STEP-File, STEP-XML or SDAI for the AIM/MIM data model andsupporting one or several conformance classes of an AP.

13. What are the three layers of architecture?

i) Application layer ii) Logical layer iii) Physical layer

14. What is the need for CALS (Continuous Acquisition and Lifecycle Support)? (AU APR 2016)

- To establish an integrated system in which any information obtained at each stage (survey, designing, construction and maintenance) of public works should be digitalized so that every person of each section both in public and private sectors could share and re-use the information.

15. What is Graphic Communication? (AU MAY 2017)

- Graphic communication as the name suggests is about communicating through graphics.
- It is the process of creating and producing drawings or images that get a message across.
- For example a poster is advertising an event, a drawing showing the plans for a building or a 3D computer model showing what a new product will look like.

16. Define communication protocol?

- Communicating systems use well-defined formats (protocol) for exchanging messages.
- Each message has an exact meaning intended to elicit a response from a range of possible responses pre-determined for that particular situation.
- Thus, a protocol must define the syntax, semantics, and synchronization of communication; the specified behavior is typically independent of how it is to be implemented.

17. What are the model communication policies?

1. Openness accessibility: availability and willingness to respond.
2. Truthfulness: unconditional honesty is the only policy.
3. Responsiveness: recognition that any constituent concern is by definition legitimate and must be addressed.
4. No secrets: our behavior, our attitudes, our plans, even our strategic discussions must be unchallengeable, unassailable, and positive

18. List out the international organizations involved to develop the graphics standards:

- ACM (Association for Computer Machinery)
- ANSI (American National Standards Institute)
- ISO (International Standards Organization)
- GIN (German Standards Institute)
- CALS, GKS , PHIGS , VDI , VDM , NAPLPS

19. List out the various standards in graphics programming

- IGES (Initial Graphics Exchange Specification)
- DXF (Drawing / Data Exchange Format)
- STEP (Standard for the Exchange of Product model data)

20. Define Graphics Kernel System (GKS) (AU NOV 2016)

- The Graphical Kernel System (GKS) was the first ISO standard for computer graphics in low- level, established in 1977.
- GKS offers a group of drawing aspects for 2D vector graphics appropriate for mapping and related duties.

21. Enumerate Open Graphics Library.

- OpenGL draws primitives into a structured buffer focus to a various selectable modes. Every Point, line, polygon, or bitmap are called as a primitive.
- Each mode can be modified separately; the parameters of one do not affect the parameters of others.

-

22. Narrate IRIS GL.

- OpenGL is supported on Silicon Graphics' Integrated Rater Imaging System Graphics Library (IRIS GL). Though it would have been potential to have designed a totally new Application Programmer's Interface (API), practice with IRIS GL offered insight into what programmers need and don't need in a Three Dimensional graphics API

23. Define NAPLPS

- NAPLPS (North American Presentation- Level Protocol Syntax) describes text and graphics in the form of sequences of bytes in ASCII code.

24. Define IGES

- IGES (Initial Graphics Exchange Specification) enables an exchange of model data basis amongCAD system.

25. Define DXF

- DXF (Drawing / Data Exchange Format) file format was meant to provide an exact representation of the data in the standard CAD file format.

26. Define STEP (AU APR 2016)

- STEP (Standard for the Exchange of Product model data) can be used to exchange data between CAD, Computer Aided Manufacturing (CAM) , Computer Aided Engineering (CAE) , product data management/enterprise data modeling (PDES) and other CAx systems.

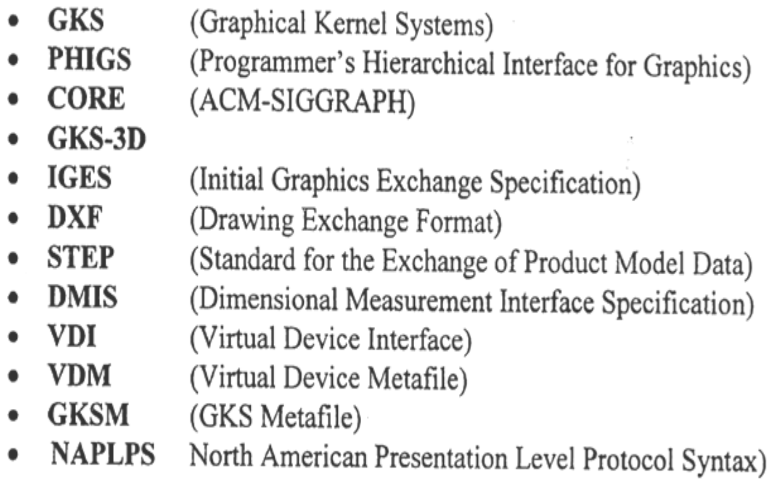
27. Define GKS (AU APR 2017)

- GKS (Graphics Kernel System) provides a set of drawing features for two-dimensional vector graphics suitable for charting and similar duties.

28. Define PHIGS

- PHIGS (Programmer's Hierarchical Interactive Graphic System) the PHIGS standard defines a set of functions and data structures to be used by a programmer to manipulate and display 3-D graphical objects.

29. List some Standards used in computer graphics.



• **GKS**	(Graphical Kernel Systems)
• **PHIGS**	(Programmer's Hierarchical Interface for Graphics)
• **CORE**	(ACM-SIGGRAPH)
• **GKS-3D**	
• **IGES**	(Initial Graphics Exchange Specification)
• **DXF**	(Drawing Exchange Format)
• **STEP**	(Standard for the Exchange of Product Model Data)
• **DMIS**	(Dimensional Measurement Interface Specification)
• **VDI**	(Virtual Device Interface)
• **VDM**	(Virtual Device Metafile)
• **GKSM**	(GKS Metafile)
• **NAPLPS**	North American Presentation Level Protocol Syntax)

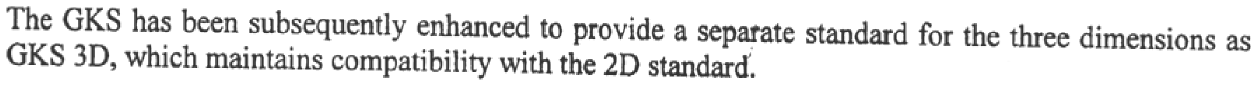
30. What are the Types of Standards used in CAD?

- Graphics Standards
- Data Exchange Standards
- Communication Standards

31. Write aim of Graphics Standardization. (AU NOV 2016)

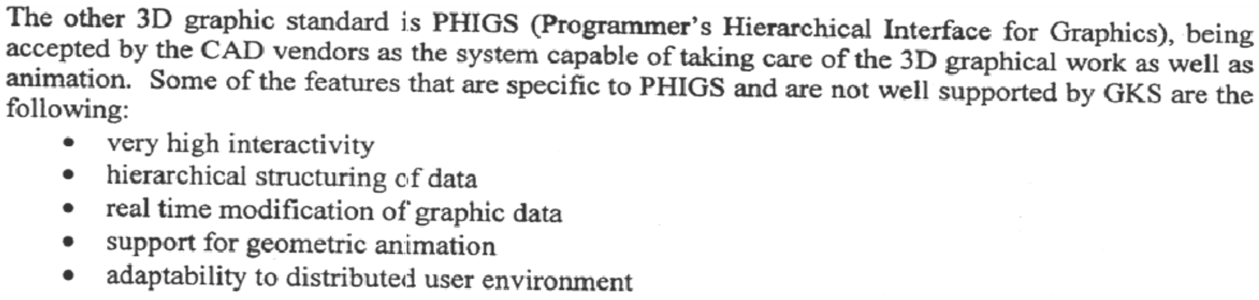
- To provide versatility in the combination of Software and Hardware items of turnkey systems
- To allow the creation of portable application software package applicable for wide range of hardware makes enumerate configurations.

32. Enumerate GKS 3D. (AU MAY 2015)



The GKS has been subsequently enhanced to provide a separate standard for the three dimensions as GKS 3D, which maintains compatibility with the 2D standard.

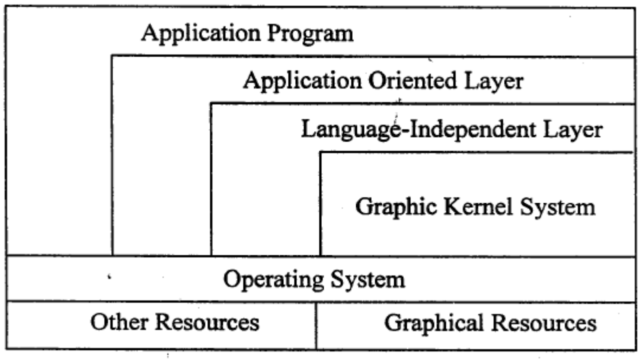
33. Write short note on PHIGS.



The other 3D graphic standard is PHIGS (Programmer's Hierarchical Interface for Graphics), being accepted by the CAD vendors as the system capable of taking care of the 3D graphical work as well as animation. Some of the features that are specific to PHIGS and are not well supported by GKS are the following:

- very high interactivity
- hierarchical structuring of data
- real time modification of graphic data
- support for geometric animation
- adaptability to distributed user environment

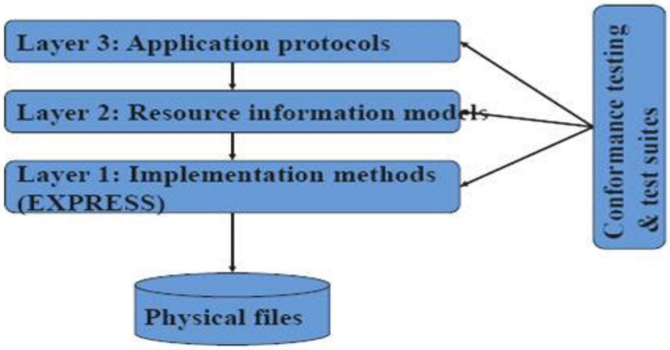
34. Sketchthelayer modelof GKS.



35. Write features of Continuous Acquisition and Life-cycle Support (CALS).

- ➢ Developed by US Department of Defense
- ➢ Prescribes formats for storage and exchange of technical data
- ➢ Technical publications an important focus

36. Sketch STEP Architecture.


Layer 3: Application protocols
Layer 2: Resource information models
Layer 1: Implementation methods (EXPRESS)
Physical files
Conformance testing & test suites

37. List the Classes of STEP Parts.

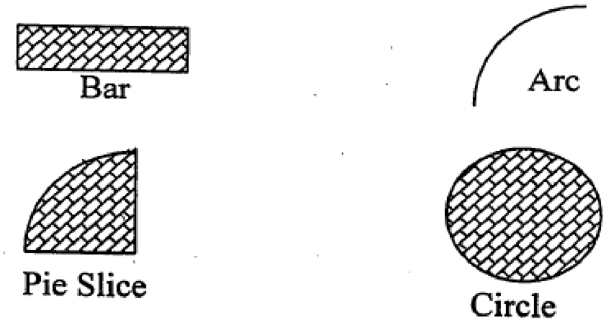
- ➢ Introductory
- ➢ Description methods

- Implementation methods
- Conformance testing methodology and frame work
- Integrated resources
- Application protocols
- Abstracttest suites

38. What is meant by Open GL (Open Graphics Library)? (AU APR 2015)

- **OpenGL** is a cross-language, multi-plat form application programming interface (API) for rendering 2D and 3D vector graphics. The AP List Typically used to interact with a graphics processing unit (GPU), to achieve hardware-accelerated rendering.

39. List Graphics primitive's in GKS with sketch.



40. Write short note on Computer Graphics Metafile (CGM).

- Developed in 1986 vector file format for illustrations and drawings all graphical element scan be specified in a textual source file that can be compiled in to a binary file or one of two text representations.

41. What are Important CALS Standards?

- Standard Generalized Markup Language (SGML)
- Computer Graphics Metafile (CGM)