

# LECTURE NOTES ON INTRODUCTION TO CAD/CAM

2023 – 2024

**III B. Tech I Semester**  
Mr. K.RAJASEKHAR, Assistant Professor



**ANAMACHARYA INSTITUTE OF TECHNOLOGY  
AND SCIENCES, TIRUPATI**

**(AUTONOMOUS)**

Venkatapuram village, Renigunta mandal, Tirupati – 517 506

**Department of Mechanical Engineering**

**ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND  
SCIENCES, TIRUPATI(AUTONOMOUS)**

**Year: III**

**Semester : I**

**Branch of Study : ME**

| <b>Subject Code</b> | <b>Subject Name</b>     | <b>L</b> | <b>T</b> | <b>P</b> | <b>Credits</b> |
|---------------------|-------------------------|----------|----------|----------|----------------|
| 20APE0302           | Introduction to CAD/CAM | 3        | 0        | 0        | 3              |

**Course Outcomes:**

- CO: 1 Understand the basic concepts components of CAD/CAM concepts of Graphics Techniques.
- CO: 2 Understand the concepts of Geometric representation methods.
- CO: 3 Understand and apply Numerical CNC Part Programming methods.
- CO: 4 Understand the concepts of Group technology and techniques, production flow Analysis.
- CO: 5 Understand the concepts of FMS and its elements.

**UNIT I**

Introduction: Definition and scope of CAD/CAM- Computers in industrial manufacturing, design process Computer Aided Design (CAD)-Computer Aided Manufacturing (CAM)- Computer Integrated Manufacturing (CIM)

Graphics: Data base for graphic modeling-transformation geometry-3D transformations – Clipping-hidden line removal-Colour-shading.

**UNIT II**

Geometric modelling Parametric representation of curves, solids & surfaces. Geometric construction methods-Constraint based modeling- Wireframe, Surface- Bezier , B-Spline Surfaces and Solid- Constructive Solid Geometry, Boundary representation and Cellular Decomposition.

**UNIT III**

NC Control production systems: Introduction to NC, CNC, DNC - Manual part Programming – Computer Assisted Part Programming – Examples using NC codes- Adaptive Control – Canned cycles and subroutines – CAD/ CAM approach to NC part programming – APT language

**UNIT IV**

Role of information systems in manufacturing Discrete part manufacture-information requirements of a production organization-manufacturing strategies Integration requirement – Group Technology-coding-Production flow analysis-computer part programming CAPP implementation techniques.

**UNIT V**

Automated manufacturing systems Flexible Manufacturing systems (FMS) – the FMS concepts – transfer systems – head changing FMS – Introduction to Rapid prototyping, Knowledge Based Engineering, Virtual Reality, Augmented Reality – automated guided vehicle-Robots-automated storage and retrieval systems - computer aided quality controlCMM-Non contact inspection methods.

### **Text Books:**

1. P.N.Rao, CAD/CAM: Principles & Applications-3rd Edition, Tata McGraw Hill.
2. CAD/CAM Concepts & applications/Alavala/PHI

### **Reference Books:**

1. CAD/CAM Theory and Practice / IbrahimZeid / TMH.
2. CAD/CAM/CIM Radha Krishnan & Subramanian / New age
3. Principles of computer Aided Design and Manufacturing / Fanlc / Amirouche / Pearson.
4. Computer Numerical Control Concepts and Programming / Warrens & Seames / Thomson

| <b>List of COs</b> | <b>PO no. and keyword</b>            | <b>Competency</b> | <b>Performance Indicator</b> |
|--------------------|--------------------------------------|-------------------|------------------------------|
| CO: 1              | PO 1: Engineering knowledge          | 1.6               | 1.3.1                        |
| CO: 2              | PO 5: Modern tool usage              | 5.2               | 5.2.2                        |
| CO: 3              | PO 2: Problem analysis               | 2.5               | 2.1.2                        |
| CO: 4              | PO 5: Modern tool usage:             | 5.2               | 5.2.2                        |
| CO: 5              | PO 7: Environment and sustainability | 7.4               | 7.4.1                        |



**ANNAMACHARYA INSTITUTE OF TECHNOLOGY &  
SCIENCES(AUTONOMOUS)**

Karakambadi Road, Venkatapuram (Vi), Tirupati

---

**COURSE OUTCOMES & COURSE OBJECTIVES**

Name of the Faculty : K. Rajasekhar  
Subject : Introduction to CAD/CAM  
Subject Code : **20APE0302**  
Class & Branch / Specialization : III B.TECH I SEM Mechanical Engineering

| <b>S.No</b> | <b>Course Objectives</b>  |
|-------------|---|
| <b>1.</b>   | Understand the basic concepts components of CAD/CAM concepts of Graphics Techniques.  |
| <b>2.</b>   | Understand the concepts of Geometric representation methods.                          |
| <b>3.</b>   | Understand and apply Numerical CNC Part Programming methods.                          |
| <b>4.</b>   | Understand the concepts of Group technology and techniques, production flow Analysis. |
| <b>5.</b>   | Understand the concepts of FMS and its elements.                                      |

**Signature of Faculty**  
**Date :**

**Signature of HOD**  
**Date :**

**Signature of Principal**  
**Date :**

## UNIT I

Introduction: Definition and scope of CAD/CAM- Computers in industrial manufacturing, design process Computer Aided Design (CAD)-Computer Aided Manufacturing (CAM)- Computer Integrated Manufacturing (CIM) Graphics: Data base for graphic modeling-transformation geometry-3D transformations – Clipping-hidden line removal- Colour-shading.

### INTRODUCTION:

In today's global competition, industries cannot survive unless they introduce new products or existing ones with:

- Better quality
- Lower cost
- Shorter lead time

**Computer aided design (CAD):** CAD can be defined as the use of computer systems to assist in creation, modification, analysis and optimization

**Computer aided machining (CAM):** CAM can be defined as the use of computer systems to plan, manage and control a manufacturing plant through either direct or indirect computer interface with the plant's production resources.

- 3D CAD data can be read by CAM software which takes 3D data & CNC machine parameters as inputs and delivers a tool path that cuts metal as per the part designs.
- The tool path so generated can be simulated on the screen to evaluate tool gauging so that the machining run is perfect.

### Computer Integrated manufacturing (CIM):

- Computer-integrated manufacturing (CIM) is manufacturing supported by computers. It is the total integration of Computer Aided Design / Manufacturing and also other business operations and databases.
- Definition: CIM is the integration of total manufacturing enterprise by using integrated systems and data communication coupled with new managerial philosophies that improve organizational and personnel efficiency.
- The CIM concept is that all the operations related to the production function in an integrated computer system to assist, enhance and /or automate the operations.
- The computer system is spread throughout the firm, touching all the activities that support manufacturing.
- In this integrated computer system, the output of one activity serves as the input to the next activity, through the chain of events that starts with the sales order and finishes with shipment of the product.

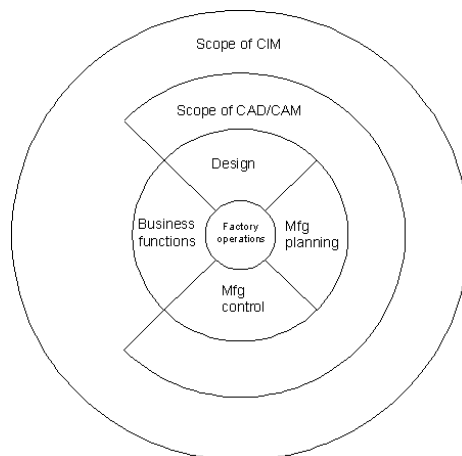
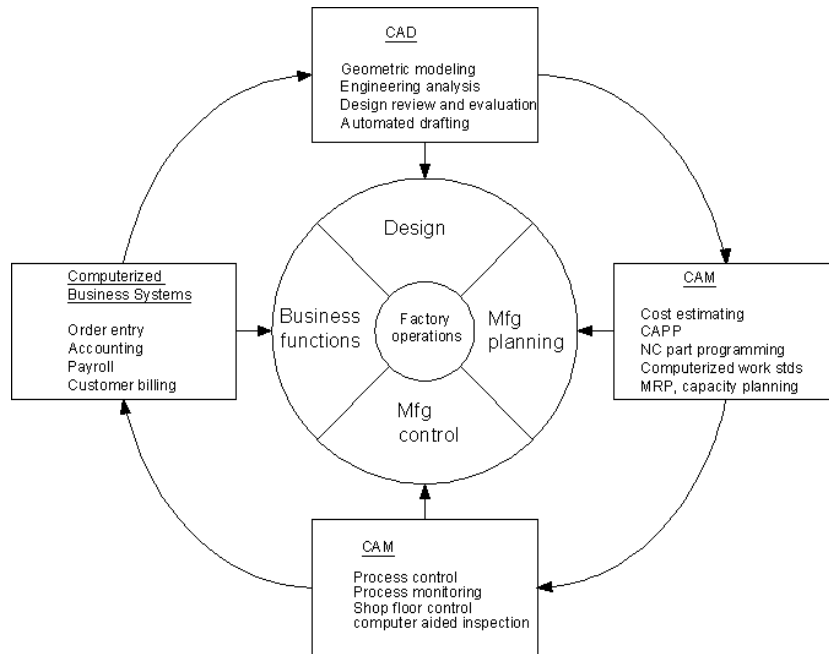


Figure 1 Scope of CAD/CAM and CIM



**Figure 2 Computerized elements of CIM system**

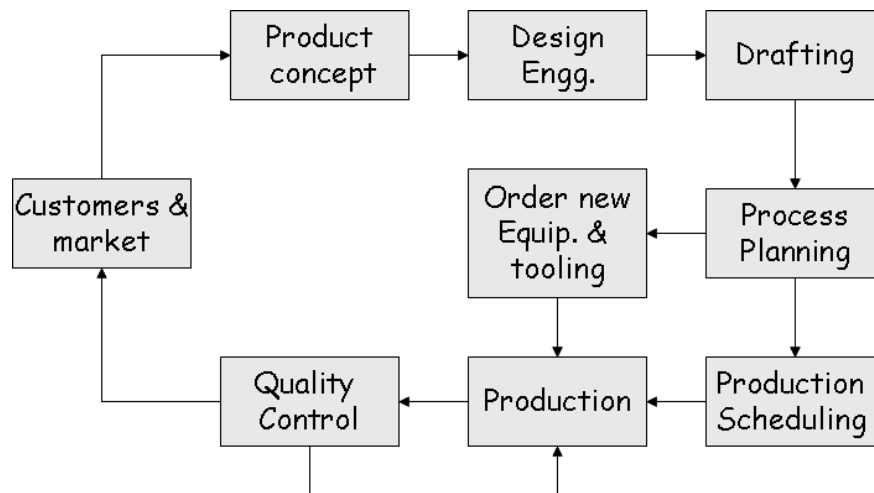
- Customer orders are initially entered by the company's sales force into a computerized order-entry system. The orders contain the specifications describing the product.
- The specifications serve as input to the design.
- New products are designed on a CAD system. The components that comprise the product are designed, the BOM is compiled, and assembly drawings are prepared.
- The output of design serves as input to mfg. engg., where process planning, tool design and similar activities are accomplished to prepare for production.
- The output of mfg. engg. provides input to the PPC, where MRP and scheduling is performed.

### **Computer-Aided Engineering (CAE):**

- Computer-aided Engineering analysis (often referred to as CAE) is the application of computer software in engineering to analyze the robustness and performance of components and assemblies.
- It encompasses simulation, validation and optimization of products and manufacturing tools.
- Parts and assemblies designed in CAD software can be „analyzed“ for their field performances right on the computer screen.
- Most of the olden-day destructive testing methods have found mathematical replacements in the modern-day CAE software.
- CAE software delivers results that help analyzing designs.
- Analysis tools are available for
  - static stress-strain,
  - deflection,
  - thermal,
  - flow,
  - motion,
  - vibration
- This allows designers to “design-right-the-first-time”.
- Use of computer systems to analyze CAD geometry
- Allows designer to simulate and study how the product will behave, allowing for optimization
- Finite-element method (FEM)

- Divides model into interconnected elements
- Solves continuous field problems

### Product Cycle (Design and Manufacturing):



**Figure 3 Product Cycle (Design and Manufacturing)**

- Driven by customers and markets which demand the product
  - In some cases, design functions are performed by the customer and the product is manufactured by a different firm.
  - In other cases, design and manufacturing is accomplished by the same firm
1. The product cycle begins with a concept, an idea of a product.
  2. This concept refined, analyzed, improved and translated into a plan for the product through the design engineering process.
  3. The plan is documented by drafting a set of engineering drawings showing how the product is made and providing a set of specifications indicating how the product should perform
  4. A process plan is formulated which specifies sequence of production operations required to make the product.
  5. New equipment and tools sometimes be acquired to produce the new product.
  6. Scheduling provides a plan that commits the company to manufacture of certain quantities of the product by certain dates.
  7. Once all these plans are formulated product goes into production.
  8. It is followed by quality testing and delivery to the customer.

### Product Cycle revised with CAD/CAM overlaid:

- Computer aided design and computer automated drafting are utilized in conceptualization, design and documentation of the product.
- Computers are used in process planning and scheduling to perform these functions more efficiently.
- Computers are used in production to monitor and control the manufacturing operations.
- In quality control, computer are used to perform inspections and performance tests on the product and its components

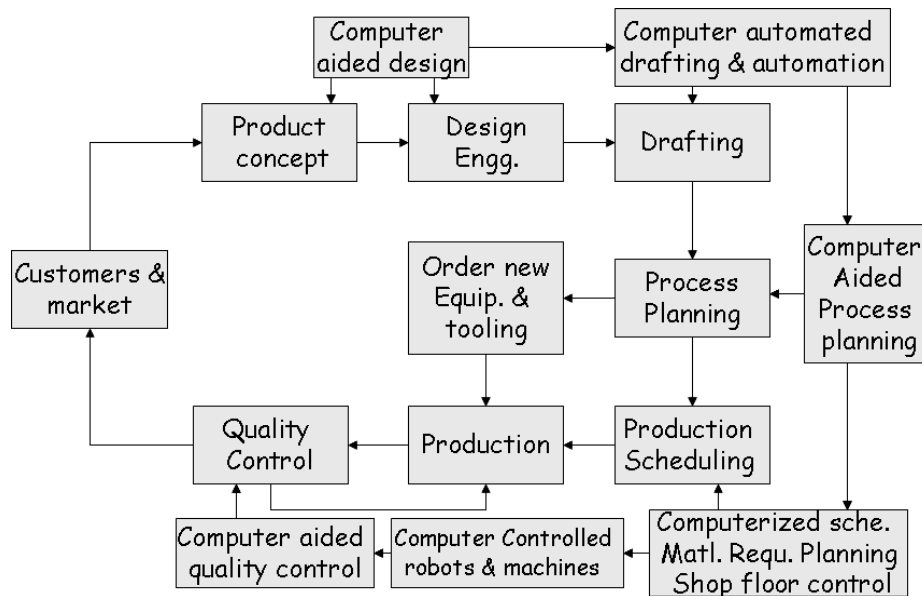


Figure 4 Product Cycle revised with CAD/CAM overlaid

**Mathematical Model for product life cycle:**

- Let T1 be the time required to produce 1 unit of product.
  - This includes sum of individual process times plus time to assemble, inspect and package a single product.
- T2 be time associated with planning and setting for each batch of production
  - This include the ordering of raw materials by purchasing department, time required in production planning to schedule the batch, setup times for each operation etc.
- T3 be the time required for designing the product and for all other activities that are accomplished once for each different product.
  - These include process planning, cost estimating and pricing, building of special tools and fixtures.
  - B be the number of batches produced throughout the product life cycle.
  - Q be the number of units produced in each batch.
  - The aggregate time spent on the product throughout its life cycle can be

$$TT_{LC} = BQT_1 + BT_2 + T_3$$

Where  $TT_{LC}$  is the total time during the product life cycle  
 The average time spent on each unit of product during its life cycle.

$$T_{LC} = T_1 + \frac{T_2}{Q} + \frac{T_3}{BQ}$$

*Where  $T_{LC}$  = Average time*

- In mass and batch production the T2 and T3 terms can be spread over a large number of units.
- Their relative values, therefore becomes less important as the production quantities increase. The T1 term becomes most important term.
- In job shop manufacturing, the T2 and T3 terms can become significant because the quantities are so low.

**Relationship between CAD/CAM & Automation:**

- The goal of CAD/CAM and automation is to reduce the various time elements in product life cycle.
- With this there is increase in productivity and improve in standard of living.
- The automation technology is concerned with reducing the T1 and T2 Elements, with emphasize on the unit production cost (T1).
- The CAD/CAM technology is concerned with all the three terms but is



perhaps focused on the T3 and T2 terms in the life cycle model.

- CAD/CAM has made important contribution towards integrating the functions of design and manufacturing

### The manufacturing database from CAD/CAM:

In the conventional manufacturing cycle practiced for so many years in industry,

- Engineering drawings were Prepared by design draftsman and then used by manufacturing engineer to prepare process plan (“Route Sheet”)
- The activities involved in the designing the product were separated from the activities associated with process planning.
- Essentially a two-step procedure was employed.
- This was both time consuming and involved duplication of efforts by design and manufacturing personnel
- In an integrated CAD/CAM system, a direct link is established between product design and manufacturing.
- It is goal of CAD/CAM not only to automate certain design phases of design and certain phases of mfg., but also to automate the transition from design to manufacturing.
- Computer- based systems have been developed which create much of the data and documentation required to plan and manage the mfg. operations for the product.
- The manufacturing data base is an integrated CAD/CAM data base.
- It includes all the data on the product generated during design (geometry data, BOM , material specifications) as well as additional data required for mfg., much of which based on the product design

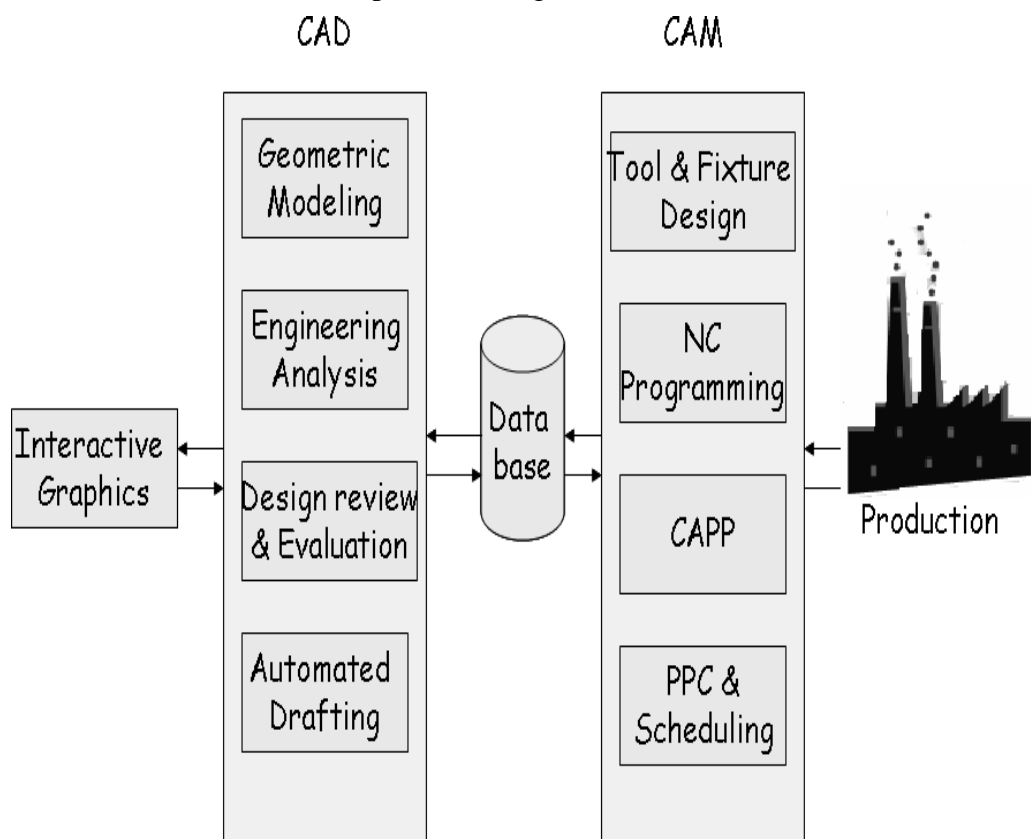


Fig 5. The manufacturing database from CAD/CAM

### **Benefits of CAD:**

- Productivity improvement in Design
- Shorter lead times
- Reduced engineering personnel requirement
- Customer modifications are easier to make
- Rapid response of the design analysis
- Improved accuracy of design
- Avoidance of subcontracting to meet schedules
- Provide better functional analysis to reduce prototype testing
- Avoidance of subcontracting to meet schedules
- Assistance in preparation of documentation
- Minimized transcription errors
- Designs have more standardization
- Fewer errors in NC part programming
- Better communication interfaces and greater understanding among engineers, designers, drafters, management and different project groups.

### **Computer Graphics Software:**

- The graphic software is the collection of programs written to make it convenient for user to operate the computer graphics system.
- It includes programs to generate images on the CRT screen, to manipulate the images, and to accomplish various types interaction between the user and the system.
- Some other programs for implementing specialized functions of CAD/CAM are
  1. Design analysis programs (FEA & kinematic simulation)
  2. Manufacturing planning programs (e.g. automated process planning and numerical control part programming )

### **Ground rules in Designing Software:**

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

### **The software configuration of a graphics system:**

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

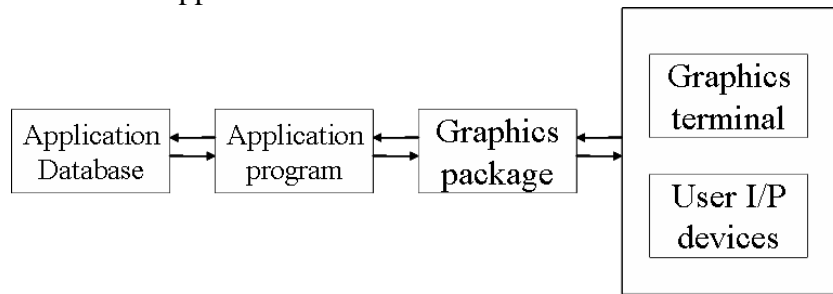
- Interact with the graphics terminal to create and alter images on the screen
- Construct a model of something physical out of the images on the screen. The models are sometimes called as application models
- Enter the model into computer memory and/or secondary storage.

### **Graphics software:**

The graphics software can be divided into three modules

- The graphics package
- The application program

- The application data base



### Application Program

- It controls the storage of data into and retrieves data out of application database
- The application program is driven by the user through the graphics package
- The application program is implemented by the user to construct the model of a physical entity whose image to be viewed on the graphics screen.
- Application programs are written for particular problem areas.
- Problem areas in engineering design would include arch, construction, mech. Components, ext, chem, aerospace.
- Problem areas other than deign include flight simulators, graphical display of data, mathematical analysis, and even artwork

### Graphics Package

- The graphics package is the software support between the user and the graphics terminal.
- It manages the graphical interaction between the user and the system
- It also serves as a s/w support between the user and the application software
- The graphics package consist of input subroutines and output subroutines
- The input subroutine accepts the input commands and data from the user and forward them to the application program
- The output subroutines control the display terminal (or other output device) and converts the application models into 2D or 3D graphics pictures.

### Application data base

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

### **Functions of a Graphics package:**

- Generation of graphics elements
- Transformations
- Display control and windowing function
- Segmenting functions
- User input functions

### Generation of graphics elements

- A graphic element in computer graphics is a basic image entity such as a dot (or point), line segment, circle, and so forth
- The collection of elements in the system could also include alphanumeric characters and special symbols
- There is often a special hardware component in the graphics system associated with the display of many of the elements.
- This speeds up the process of generating the element
- The user can construct the application model out of collection of elements

- The term primitive is used in reference to the graphics element
- E.g. sphere, cube, or cylinder
- In 3D wire frame models and solid modelling, primitives are used as building blocks.

### Transformations

- Transformations are used to change the image on the display screen
- Transformations are applied to graphics elements in order to aid the user in constructing an application model
- It includes enlargement and reduction of the image by a process called scaling, repositioning the image or translation, and rotation.

### Display control and windowing

- This provides the user with the ability to view the image from the desired angle and at the desired magnification
- Another aspect of display control is hidden line removal.

### Segmenting function

- Segmenting provides user with the capability to selectively replace, delete or otherwise modify portions of the image.
- The term segment refers to a particular portion of the image which has been identified for the purpose of modifying it.
- Storage type CRT is unsuited to segmenting function.

### User input output functions

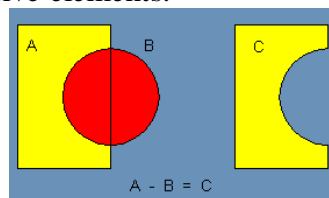
- User input functions constitute a critical set of functions in the graphics package because they permit the operator to enter commands or data to the system.
- The entry is accomplished by means of operator input devices.

### **Constructing the geometry:**

- The use of graphics elements
- Defining the graphics elements
- Editing the geometry

### The use of graphics elements

- The graphics system accomplishes the definition of the model by constructing it out of graphics elements
- These elements are called by the user during the construction process and added one by one, to create the model
- There are several aspects about this construction process.
  1. Each new element is being called but before it is added to the model, the user can specify its size, its position and orientation.  
These specifications are necessary to form the model to proper shape and size. For this purpose various transformations are utilized
  2. Graphics element can be subtracted as well as added  
Another way of saying this is that the model can be formed out of negative elements as well as positive elements.



3. A third feature available during model building is the capability to group several elements together into units which are sometimes called cells  
A cell in this context, refers to a combination of elements which can be called

to use anywhere in the model e.g. a bolt is to be used in several places in mechanical assembly

### Define the graphics elements

- The user has variety of ways to call a particular graphic element and position it on the geometric model
- Points, lines, arcs, circles can be defined in various ways through interaction with the ICG system
- E.g. point can be defined simply by x, y and z coordinates. A polygon would be defined as an ordered set of points representing the corners of the polygon

### Editing the geometry

- A CAD provides system provides editing capabilities to make corrections and adjustments in the geometric model
- When developing the model, the user must be able to delete, move, copy and rotate the components of the model.
- The editing involves selecting the particular portion of the model, and executing the appropriate command

### Editing features in CAD

1. Move an item. This involves the translation of item from one place to another
  2. Duplication an item at another location. The copy function is similar to move function except it preserves a copy of an item at its original location
  3. Rotate an object. This is the rotation transformation, in which the item is rotated through specified angle from its original orientation
  4. Mirror an item. This crates a mirror image of an item about a specified plane.
  5. Delete an item. This causes selected segment of the model to be removed from the screen and from the database.
  6. Trim an line or other component
  7. Scale an item
- The method of selecting the segment of the model varies from system to system
1. With cursor control common method is for a rectangle to be formed on the CRT screen around the model segment. The rectangle is defines by entering the upper left and lower right corners of the rectangle
  2. It involves light pen to be placed over the component t be selected
- The computer must some how show indicate to the user which portion of the model has been selected
  - This includes placing mark on the segment, making segment brighter than the rest of the image, and making the segment blink.

## **Transformations**

- 2-D Transformations
- 3-D Transformations

### **Two-dimensional Transformations**

- To locate a point in a two-axis Cartesian system, the x and y coordinates are specified.
- These coordinate can be treated together as a 1 x 2 matrix : (x, y), e.g. the matrix (1,4) would be interpreted to be point which is 1 unit from the origin in the x-direction and 4 units from the origin in the y-direction.
- This method of representation can be extended further to define a line as a 2 x 2 matrix by giving x and y coordinates of the two end points of the line. The notation would be,

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

### Translation

- Translation involves moving the element from one location to another

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

Where,

$x'', y''$  = coordinates of the translated point

$x, y$  = coordinates of the original point

$m, n$  = movements in the x and y direction

In the matrix notation this can be represented as,  $(x'', y'')$

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

Where,

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

- Scaling

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

The points of an element can be scaled by scaling factor as follows:

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

Where

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

- This would produce an alternation in the size of the element by the factor m in the x-direction and by the factor n in the y direction
- It also repositions the element
- If the scaling factor is  $<1$ , it is moved closer to origin
- If the scaling factor is  $>1$ , it is moved farther from the origin

### Rotation

In this transformation, the points of an object are rotated about the origin by an angle  $\theta$

For a positive angle, this rotation is in the counter clockwise direction

This accomplishes rotation of the object by the same angle, but it also

Moves the object. In matrix notation the procedure will be as follows:

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

Where

$$R = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad \text{the rotation matrix}$$

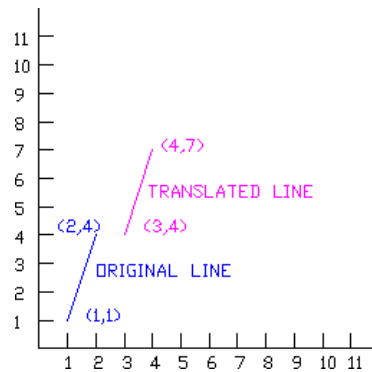
### Example : Translation

Consider the line defined by,

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

Suppose the line to be translate in space by 2 units in x direction and 3 units in the y direction.

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



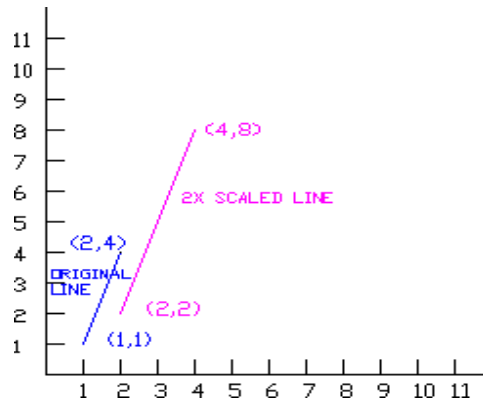
Example : Scaling

- Apply scaling factor of 2 to the line

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

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

The new line will be,



Example: Rotation

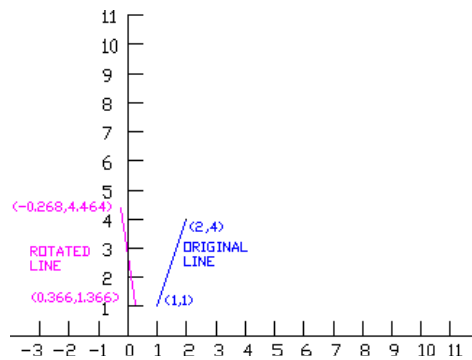
Rotate the line about origin by  $30^\circ$

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

The new line would be defined as:

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

The new line will be,



### 3 Dimensional Transformation

- Transformations by matrix methods can be extended to three dimensional space.

#### Translation

- The translation matrix for a point defined in three dimensional matrix would be,

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

And would be applied by adding increments m, n and p to the respective coordinates of each of the points defining the three-dimensional geometry elements

#### Scaling

The scaling transformation is given by,

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

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

#### Rotation

- Rotation in three dimensions can be defined for each of the axes
- Rotation about z axis by angle  $\theta$  is accomplished by the matrix

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

- Rotation about y axis by angle  $\theta$  is accomplished by the matrix

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

- Rotation about x axis by angle  $\theta$  is accomplished by the matrix

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

### Concatenation



- Single transformations can be combined as a sequence of transformations. This is called as concatenation, and the combined transformations are called concatenated transformations.
- During editing process when a graphics model is being developed, the use of concatenated transformation is quite common
- More than one transformations are usually requires to accomplish the desired transformation

e.g.

1. Rotation of the element about an arbitrary point the element
  2. Magnifying the element but maintaining the location of one of its points in the same location
- In the first case transformations would be: translation to the origin, then rotation about the origin, then translation back to original location
  - In the second case, the element would be scaled (magnified) followed by a translation to locate the desired point as needed
  - The objective of the concatenation is to accomplish a series of image manipulations as a single transformation
  - The concatenation is the product of the two transformation matrix
  - It is important that order of matrix multiplication be the same as the order in which the transformations are to be carried out.

### Example: Concatenation

- A point to be scaled by a factor of 2 and rotated by 45°. Suppose point under consideration was (3, 1). This may be one of the several points defining a geometric element

- First accomplish the transformations sequentially,  
First consider the scaling,

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

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

Next rotation can be

$$\text{performed } (x'', y'') = (x', y')$$

R

$$(x'', y'') = (6, 2) \begin{bmatrix} \cos 45 & \sin 45 \\ -\sin 45 & \cos 45 \end{bmatrix}$$

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

$$= (2.828, 5.657)$$

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

$$SR = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \cos 45 & \sin 45 \\ -\sin 45 & \cos 45 \end{bmatrix}$$

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

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

Now applying this concatenated transformation to the original matrix

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

$$= (2.828, 5.657)$$

Homogeneous representation:

- In order to concatenate the transformations matrices, all transformation matrices should be multiplicative type.
- However, the translation matrix is vector additive while all other are matrix multiplications.
- It is desirable, to express all geometric transformations in form that, they can be concatenated by matrix multiplication only.
- In homogeneous representation, an n-dimensional space is mapped into (n+1) dimensional space.
- Thus a 2 dimensional point [x, y] is represented with the homogeneous coordinate triple (x<sub>h</sub>, y<sub>h</sub>, h)

Where,

$$x = \frac{x_h}{h}, y = \frac{y_h}{h}$$

- Thus, a general homogeneous coordinate can also be written as (h.x, h.y, h).
- For two-dimensional geometric transformations, homogeneous parameter h to be any nonzero value.
- Thus, there is an infinite number of equivalent homogeneous representations for each coordinate point (x, y)
- A convenient choice is simply to set h=1.
- Each two-dimensional position is then represented with coordinates (x, y, 1)
- This facilitates the computer graphics operations where the concatenation of multiple transformations can be easily carried out.
- The translation matrix in multiplication form can be given as,

$$[MT] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ dX & dY & 1 \end{bmatrix}$$

The transformation operation can be written as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ dX & dY & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

The scaling matrix in multiplication form can be given as,

$$[MS] = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The transformation operation can be written as

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

The rotation matrix in multiplication form can be given as,

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

The transformation operation can be written as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

A square ABCD of side 50 units having one of its vertex at A(10,10) is to be scaled by 0.8 along x axis and 1.2 along y-axis. The model is then rotated about origin by 30° in anticlockwise direction. Perform the necessary transformations using homogeneous transformation matrix and show them on graph paper.

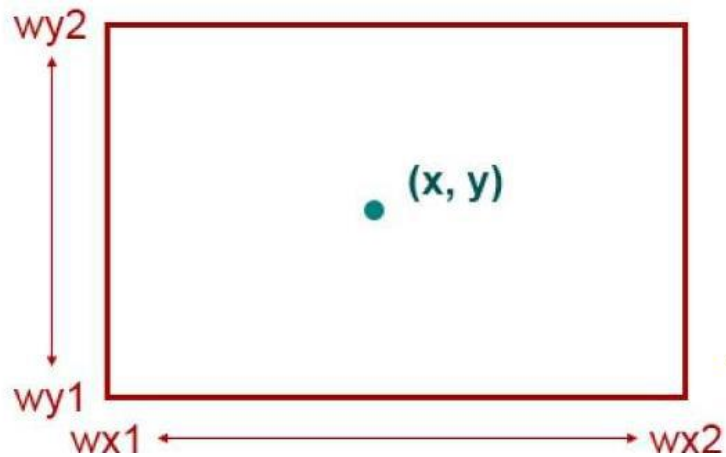
### Viewing & Clipping

The primary use of clipping in computer graphics is to remove objects, lines, or line segments that are outside the viewing pane. The viewing transformation is insensitive to the position of points relative to the viewing volume – especially those points behind the viewer – and it is necessary to remove these points before generating the view.

#### Point Clipping

Clipping a point from a given window is very easy. Consider the following figure, where the rectangle indicates the window. Point clipping tells us whether the given point X,Y is within the given window or not; and decides whether we will use the minimum and maximum coordinates of the window.

The X-coordinate of the given point is inside the window, if X lies in between  $Wx1 \leq X \leq Wx2$ . Same way, Y coordinate of the given point is inside the window, if Y lies in between  $Wy1 \leq Y \leq Wy2$ .

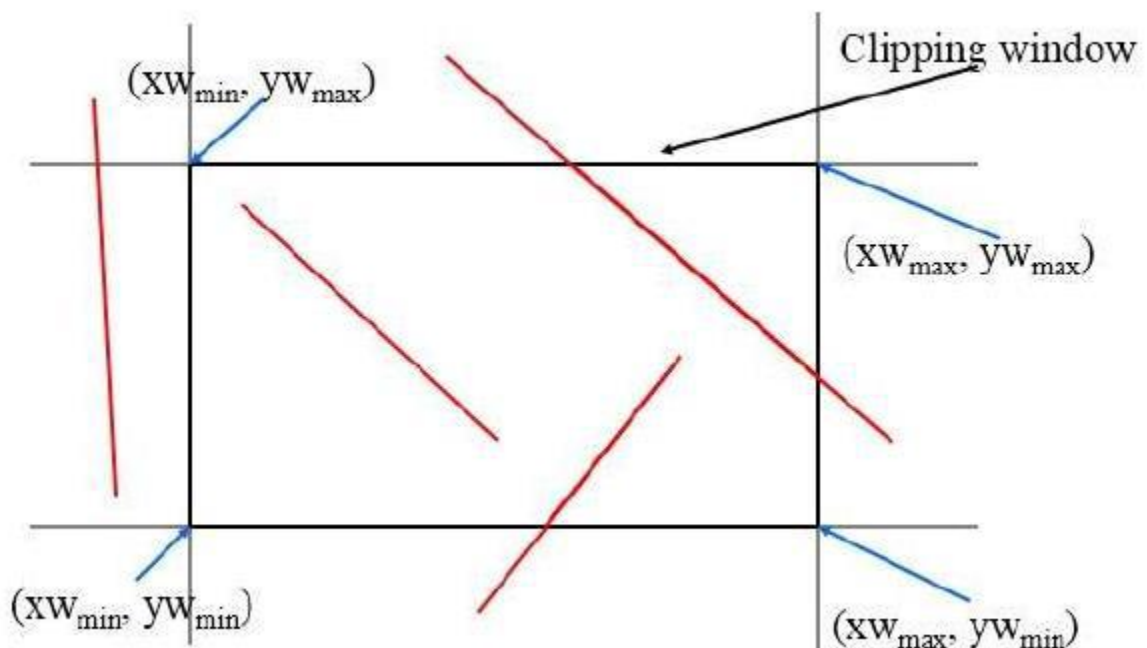


## Line Clipping

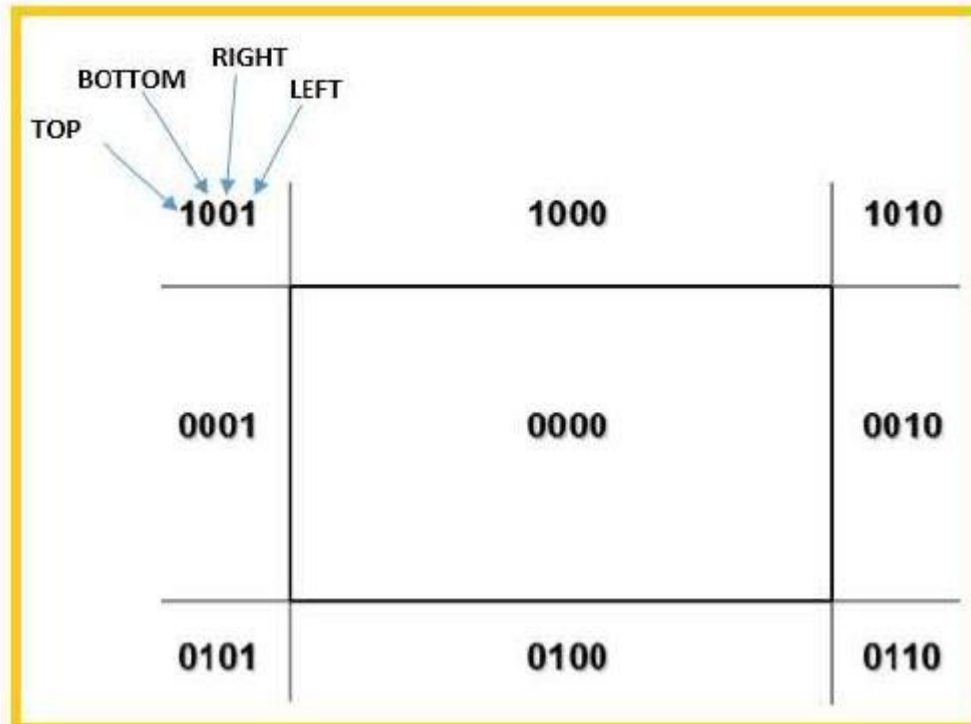
The concept of line clipping is same as point clipping. In line clipping, we will cut the portion of line which is outside of window and keep only the portion that is inside the window.

### Cohen-Sutherland Line Clippings

This algorithm uses the clipping window as shown in the following figure. The minimum coordinate for the clipping region is  $(XW_{min}, YW_{min})$  and the maximum coordinate for the clipping region is  $(XW_{max}, YW_{max})$



We will use 4-bits to divide the entire region. These 4 bits represent the Top, Bottom, Right, and Left of the region as shown in the following figure. Here, the **TOP** and **LEFT** bit is set to 1 because it is the **TOP-LEFT** corner.



There are 3 possibilities for the line –

- Line can be completely inside the window This line should be accepted.
- Line can be completely outside of the window This line will be completely removed from the region.
- Line can be partially inside the window we will find intersection point and draw only that portion of line that is inside region.

## Algorithm

**Step 1** – Assign a region code for each endpoints.

**Step 2** – If both endpoints have a region code **0000** then accept this line.

**Step 3** – Else, perform the logical **AND** operation for both region codes.

**Step 3.1** – If the result is not **0000**, then reject the line.

**Step 3.2** – Else you need clipping.

**Step 3.2.1** – Choose an endpoint of the line that is outside the window.

**Step 3.2.2** – Find the intersection point at the window boundary base on region code.

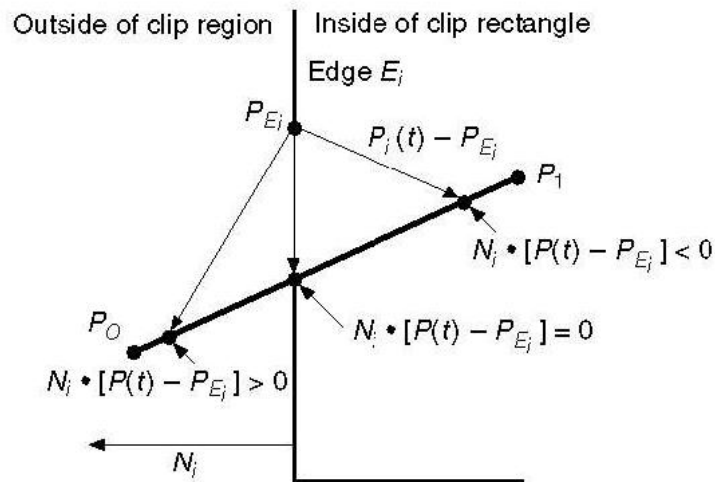
**Step 3.2.3** – Replace endpoint with the intersection point and update the region code.

**Step 3.2.4** – Repeat step 2 until we find a clipped line either trivially accepted or trivially rejected.

**Step 4** – Repeat step 1 for other lines.

# Cyrus-Beck Line Clipping Algorithm

This algorithm is more efficient than Cohen-Sutherland algorithm. It employs parametric line representation and simple dot products.



Parametric equation of line is –

$$P_0P_1: P(t) = P_0 + t(P_1 - P_0)$$

Let  $N_i$  be the outward normal edge  $E_i$ . Now pick any arbitrary point  $P_{E_i}$  on edge  $E_i$  then the dot product  $N_i \cdot [P - P_{E_i}]$  determines whether the point  $P$

is “inside the clip edge” or “outside” the clip edge or “on” the clip edge.

The point  $P$  is inside if  $N_i \cdot [P - P_{E_i}] < 0$

The point  $P$  is outside if  $N_i \cdot [P - P_{E_i}] > 0$

The point  $P$  is on the edge if  $N_i \cdot [P - P_{E_i}] = 0$  *Intersection point*

$$N_i \cdot [P - P_{E_i}] = 0$$

$$N_i \cdot [P_0 + t(P_1 - P_0) - P_{E_i}] = 0 \text{ Replacing } P(t) \text{ with } P_0 + t(P_1 - P_0)$$

$$N_i \cdot [P_0 - P_{E_i}] + N_i \cdot t[P_1 - P_0] = 0$$

$$N_i \cdot [P_0 - P_{E_i}] + N_i \cdot tD = 0 \text{ (substituting } D \text{ for } [P_1 - P_0])$$

$$N_i \cdot [P_0 - P_{E_i}] = -N_i \cdot tD$$

The equation for  $t$  becomes,

$$t = \frac{N_i \cdot [P_o - P_{E_i}]}{-N_i \cdot D}$$

It is valid for the following conditions –

$N_i \neq 0$  error cannot happen

$D \neq 0$  ( $P_1 \neq P_0$ )

$N_i \cdot D \neq 0$  ( $P_0 P_1$  not parallel to  $E_i$ )

## COLOUR -SHADING

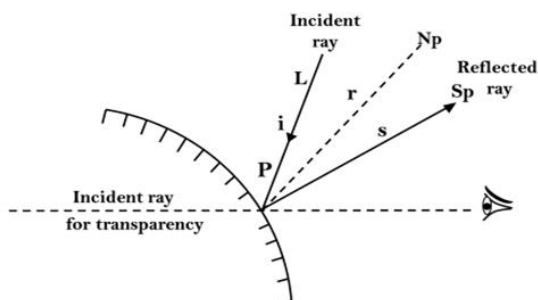
Shading is concerned with the implementation of the illumination model at the pixel points or polygon surfaces of the graphics objects. In particular, in Computer Graphics, Shading is a method used to create or enhance the illusion of depth in an image by varying the amount of darkness in the image.

Shading is referred to as the implementation of the illumination model at the pixel points or polygon surfaces of the graphics objects.

Shading model is used to compute the intensities and colors to display the surface. The shading model has two primary ingredients: properties of the surface and properties of the illumination falling on it. The principal surface property is its reflectance, which determines how much of the incident light is reflected. If a surface has different reflectance for the light of different wavelengths, it will appear to be colored.

An object illumination is also significant in computing intensity. The scene may have to save illumination that is uniform from all direction, called diffuse illumination.

Shading models determine the shade of a point on the surface of an object in terms of a number of attributes. The shading Mode can be decomposed into three parts, a contribution from diffuse illumination, the contribution for one or more specific light sources and a transparency effect. Each of these effects contributes to shading term E which is summed to find the total energy coming from a point on an object. This is the energy a display should generate to present a realistic image of the object. The energy comes not from a point on the surface but a small area around the point.



The simplest form of shading considers only diffuse illumination:

The simplest form of shading considers only diffuse illumination:

$$E_{pd} = R_p I_d$$

where  $E_{pd}$  is the energy coming from point P due to diffuse illumination.  $I_d$  is the diffuse illumination falling on the entire scene, and  $R_p$  is the reflectance coefficient at P which ranges from shading contribution from specific light sources will cause the shade of a surface to vary as to its orientation concerning the light sources changes and will also include specular reflection effects. In the above figure, a point P on a surface, with light arriving at an angle of incidence  $i$ , the angle between the surface normal  $N_p$  and a ray to the light source. If the energy  $I_{ps}$  arriving from the light source is reflected uniformly in all directions, called diffuse reflection, we have

$$E_{ps} = (R_p \cos i) I_{ps}$$

This equation shows the reduction in the intensity of a surface as it's tipped obliquely to the light source. If the angle of incidence  $i$  exceeds  $90^\circ$ , the surface is hidden from the light source and we must set  $E_{ps}$  to zero.

### Constant Intensity Shading

A fast and straightforward method for rendering an object with polygon surfaces is constant intensity shading, also called Flat Shading. In this method, a single intensity is calculated for each polygon. All points over the surface of the polygon are then displayed with the same intensity value. Constant Shading can be useful for quickly displaying the general appearances of the curved surface as shown in fig:



**Fig: A Polygon mesh approximation of an object (a) is rendered with flat shading (b) and With Gouraud shading (c)**

In general, flat shading of polygon facets provides an accurate rendering for an object if all of the following assumptions are valid:-

The object is a polyhedron and is not an approximation of an object with a curved surface.



All light sources illuminating the objects are sufficiently far from the surface so that  $N$ ,  $L$  and the attenuation function are constant over the surface (where  $N$  is the unit normal to a surface and  $L$  is the unit direction vector to the point light source from a position on the surface).

The viewing position is sufficiently far from the surface so that  $V$ ,  $R$  is constant over the surface (where  $V$  is the unit vector pointer to the viewer from the surface position and  $R$  represent a unit vector in the direction of ideal specular reflection).

## Gouraud shading

This Intensity-Interpolation scheme, developed by Gouraud and usually referred to as Gouraud Shading, renders a polygon surface by linearly interpolating intensity value across the surface. Intensity values for each polygon are coordinate with the value of adjacent polygons along the common edges, thus eliminating the intensity discontinuities that can occur in flat shading.

Each polygon surface is rendered with Gouraud Shading by performing the following calculations:

1. Determining the average unit normal vector at each polygon vertex.
2. Apply an illumination model to each vertex to determine the vertex intensity.
3. Linearly interpolate the vertex intensities over the surface of the polygon.

At each polygon vertex, we obtain a normal vector by averaging the surface normals of all polygons sharing that vertex as shown in fig:

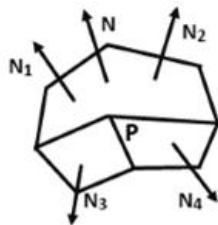


Fig: The normal vertex at vertex  $V$  is calculated as the average of surface normal for each polygon sharing the vertex.

Thus, for any vertex position  $V$ , we acquire the unit vertex normal with the calculation

$$N_V = \frac{\sum_{k=1}^p N_k}{|\sum_{k=1}^p N_k|}$$

Once we have the vertex normals, we can determine the intensity at the vertices from a lighting model.

**Following figures demonstrate the next step:** Interpolating intensities along the polygon edges. For each scan line, the intensities at the intersection of the scan line with a polygon edge are linearly interpolated from the intensities at the edge endpoints. **For example:**

In fig, the polygon edge with endpoint vertices at position 1 and 2 is intersected by the scanline at point 4. A fast method for obtaining the intensities at point 4 is to interpolate between intensities  $I_1$  and  $I_2$  using only the vertical displacement of the scan line.

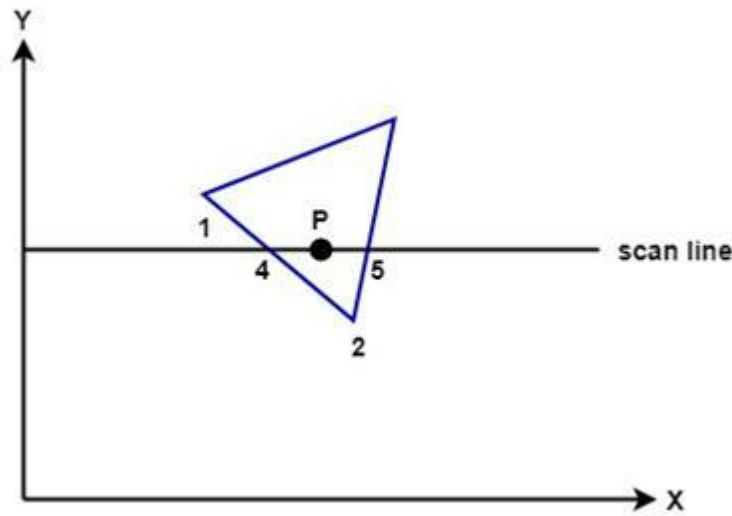


Fig : For Gouraud shading , the intensity at point 4 is linearly interpolated from the intensities at vertices 1 and 2 .The intensity at point is linearly interpolated from intensities at vertices 2 and 3. An interior point P is then assigned an intensity value that is linearly interpolated from intensities at position 4 and 5.

Similarly, the intensity at the right intersection of this scan line (point 5) is interpolated from the intensity values at vertices 2 and 3. Once these bounding intensities are established for a scan line, an interior point (such as point P in the previous fig) is interpolated from the bounding intensities at point

$$I_4 = \frac{y_4 - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_4}{y_1 - y_2} I_2$$

4 and 5 as

$$I_P = \frac{x_5 - x_p}{x_5 - x_4} I_4 + \frac{x_p - x_4}{x_5 - x_4} I_5$$

Incremental calculations are used to obtain successive edge intensity values between scan lines and to obtain successive intensities along a scan line as shown in fig:

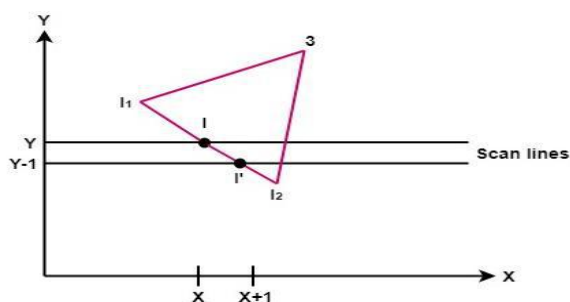


Fig: Incremental interpolation of intensity value along a polygon edge for successive scan lines.

If the intensity at edge position (x, y) is interpolated as

$$I = \frac{y-y_2}{y_1-y_2} I_1 + \frac{y_1-y}{y_1-y_2} I_2$$

Then we can obtain the intensity along this edge for the next scan line, Y-1 as

$$I' = I + \frac{I_2 - I_1}{y_1 - y_2}$$

Similar calculations are used to obtain intensities at successive horizontal pixel positions along each scan line.

When surfaces are to be rendered in color, the intensities of each color component is calculated at the vertices. Gouraud Shading can be connected with a hidden-surface algorithm to fill in the visible polygons along each scan-line. An example of an object-shaded with the Gouraud method appears in the following figure



**Fig: A Polygon mesh approximation of an object  
(a) is rendered with flat shading (b) and With Gouraud shading (c)**

Gouraud Shading discards the intensity discontinuities associated with the constant-shading model, but it has some other deficiencies. Highlights on the surface are sometimes displayed with anomalous shapes, and the linear intensity interpolation can cause bright or dark intensity streaks, called Match bands, to appear on the surface. These effects can be decreased by dividing the surface into a higher number of polygon faces or by using other methods, such as Phong shading, that requires more calculations.

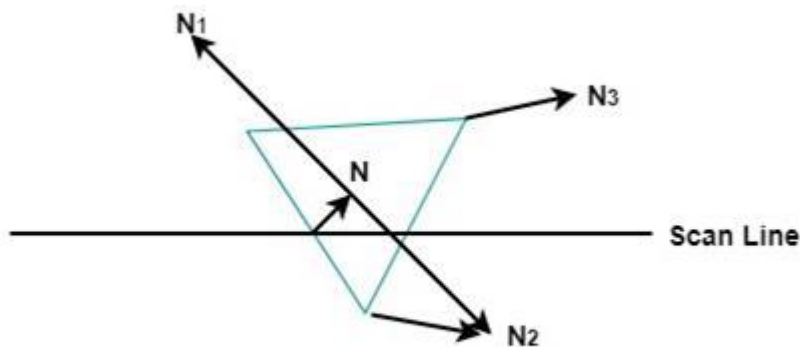
## Phong Shading

A more accurate method for rendering a polygon surface is to interpolate the normal vector and then apply the illumination model to each surface point. This method developed by Phong Bui Tuong is called Phong Shading or normal vector Interpolation Shading. It displays more realistic highlights on a surface and greatly reduces the Match-band effect.

A polygon surface is rendered using Phong shading by carrying out the following steps:

1. Determine the average unit normal vector at each polygon vertex.
2. Linearly & interpolate the vertex normals over the surface of the polygon.
3. Apply an illumination model along each scan line to calculate projected pixel intensities for the surface points.

Interpolation of the surface normal along a polygonal edge between two vertices as shown in fig:



**Fig: Interpolation of surface normals along a polygon edge.**

$$\mathbf{N} = \frac{y - y_2}{y_1 - y_2} \mathbf{N}_1 + \frac{y_1 - y}{y_1 - y_2} \mathbf{N}_2$$

Incremental methods are used to evaluate normals between scan lines and along each scan line. At each pixel position along a scan line, the illumination model is applied to determine the surface intensity at that point.

Intensity calculations using an approximated normal vector at each point along the scan line produce more accurate results than the direct interpolation of intensities, as in Gouraud Shading. The trade-off, however, is that phong shading requires considerably more calculations.

# LECTURE NOTES ON INTRODUCTION TO CAD/CAM

2023 – 2024

**III B. Tech I Semester**  
Mr. K.RAJASEKHAR, Assistant Professor



**ANAMACHARYA INSTITUTE OF TECHNOLOGY  
AND SCIENCES, TIRUPATI**

**(AUTONOMOUS)**

Venkatapuram village, Renigunta mandal, Tirupati – 517 506

**Department of Mechanical Engineering**

**ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND  
SCIENCES, TIRUPATI(AUTONOMOUS)**

**Year: III**

**Semester : I**

**Branch of Study : ME**

| <b>Subject Code</b> | <b>Subject Name</b>     | <b>L</b> | <b>T</b> | <b>P</b> | <b>Credits</b> |
|---------------------|-------------------------|----------|----------|----------|----------------|
| 20APE0302           | Introduction to CAD/CAM | 3        | 0        | 0        | 3              |

**Course Outcomes:**

- CO: 1 Understand the basic concepts components of CAD/CAM concepts of Graphics Techniques.
- CO: 2 Understand the concepts of Geometric representation methods.
- CO: 3 Understand and apply Numerical CNC Part Programming methods.
- CO: 4 Understand the concepts of Group technology and techniques, production flow Analysis.
- CO: 5 Understand the concepts of FMS and its elements.

**UNIT I**

Introduction: Definition and scope of CAD/CAM- Computers in industrial manufacturing, design process Computer Aided Design (CAD)-Computer Aided Manufacturing (CAM)- Computer Integrated Manufacturing (CIM)  
Graphics: Data base for graphic modeling-transformation geometry-3D transformations – Clipping-hidden line removal-Colour-shading.

**UNIT II**

Geometric modelling Parametric representation of curves, solids & surfaces. Geometric construction methods-Constraint based modeling- Wireframe, Surface- Bezier , B-Spline Surfaces and Solid- Constructive Solid Geometry, Boundary representation and Cellular Decomposition.

**UNIT III**

NC Control production systems: Introduction to NC, CNC, DNC - Manual part Programming – Computer Assisted Part Programming – Examples using NC codes- Adaptive Control – Canned cycles and subroutines – CAD/ CAM approach to NC part programming – APT language

**UNIT IV**

Role of information systems in manufacturing Discrete part manufacture-information requirements of a production organization-manufacturing strategies Integration requirement – Group Technology-coding-Production flow analysis-computer part programming CAPP implementation techniques.

**UNIT V**

Automated manufacturing systems Flexible Manufacturing systems (FMS) – the FMS concepts – transfer systems – head changing FMS – Introduction to Rapid prototyping, Knowledge Based Engineering, Virtual Reality, Augmented Reality – automated guided vehicle-Robots-automated storage and retrieval systems - computer aided quality controlCMM-Non contact inspection methods.

### **Text Books:**

1. P.N.Rao, CAD/CAM: Principles & Applications-3rd Edition, Tata McGraw Hill.
2. CAD/CAM Concepts & applications/Alavala/PHI

### **Reference Books:**

1. CAD/CAM Theory and Practice / IbrahimZeid / TMH.
2. CAD/CAM/CIM Radha Krishnan & Subramanian / New age
3. Principles of computer Aided Design and Manufacturing / Fanlc / Amirouche / Pearson.
4. Computer Numerical Control Concepts and Programming / Warrens & Seames / Thomson

| <b>List of COs</b> | <b>PO no. and keyword</b>            | <b>Competency</b> | <b>Performance Indicator</b> |
|--------------------|--------------------------------------|-------------------|------------------------------|
| CO: 1              | PO 1: Engineering knowledge          | 1.6               | 1.3.1                        |
| CO: 2              | PO 5: Modern tool usage              | 5.2               | 5.2.2                        |
| CO: 3              | PO 2: Problem analysis               | 2.5               | 2.1.2                        |
| CO: 4              | PO 5: Modern tool usage:             | 5.2               | 5.2.2                        |
| CO: 5              | PO 7: Environment and sustainability | 7.4               | 7.4.1                        |



**ANNAMACHARYA INSTITUTE OF TECHNOLOGY &  
SCIENCES(AUTONOMOUS)**

Karakambadi Road, Venkatapuram (Vi), Tirupati

---

**COURSE OUTCOMES & COURSE OBJECTIVES**

Name of the Faculty : K. Rajasekhar  
Subject : Introduction to CAD/CAM  
Subject Code : 20APE0302  
Class & Branch / Specialization : III B.TECH I SEM Mechanical Engineering

| S.No | Course Objectives   |
|------|---|
| 1.   | Understand the basic concepts components of CAD/CAM concepts of Graphics Techniques.  |
| 2.   | Understand the concepts of Geometric representation methods.                          |
| 3.   | Understand and apply Numerical CNC Part Programming methods.                          |
| 4.   | Understand the concepts of Group technology and techniques, production flow Analysis. |
| 5.   | Understand the concepts of FMS and its elements.                                      |

**Signature of Faculty**

**Signature of HOD**

**Signature of Principal**

**Date :**

**Date :**

**Date :**

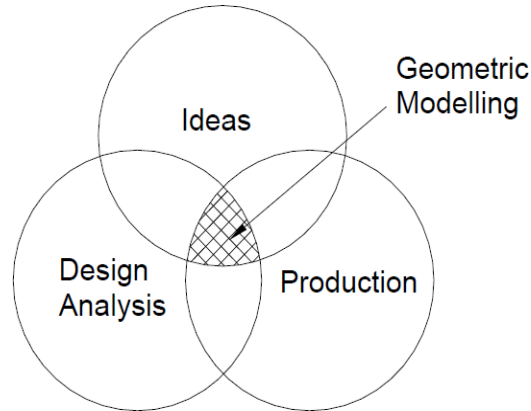


## UNIT II

Geometric modelling parametric representation of curves, solids & surfaces. Geometric construction methods-Constraint based modeling- Wireframe, Surface- Bezier, B-Spline Surfaces and Solid-Constructive Solid Geometry, Boundary representation and Cellular Decomposition.

### GEOMETRIC MODELING

#### INTRODUCTION



Geometric modeling is a branch of applied mathematical and computational geometry that studies methods and algorithms for mathematical description of shapes

The shapes studied in Geometric modeling are mostly 2-Dimensional or 3-Dimensional although many of its tools and principles can be applied to sets of any finite dimension. Most Geometric modeling is done with computer and computer-based applications. 2-Dimensions models are important in computer typography (The art and technique of arranging type in order to make language visible) and technical drawings. 3-Dimensional models are central to CAD and CAD many applied technical fields. Such as civil, mechanical, architecture, geology, medical image processing.

#### NEED OF GEOMETRIC MODELING

For manufacturing automobile engine it require around 10,000 drawings models in various departments with different styles. Once models are prepared using any CAD software, these models can be used in almost all departments of the organization which includes-

1. Designing the parts and assemblies and then performing tolerance analysis of the assemblies.
2. Preparation of production drawings of individual parts, subassemblies, assemblies, tooling and jig-fixtures.
3. Making structural, thermal and kinetic analysis.
4. Considering volume of parts and after extracting material properties from material library, cost estimation can be easily done.
5. Process plans are also prepared for all the components.
6. Using CAM software's, we can model as input for CNC programming.
7. Programming the movement of the components from one station to another is possible using

various material devices such as robots, conveyors, automated guided vehicles.

8. Using coordinate measuring machine (CMM) dimensional and geometric accuracies are checked by extracting various dimensions and parametric from the models only.
9. Other supporting activities such as material requirement and procurement, preparation of bill of materials, manufacturing resource requirement, planning and scheduling, analysis.

With competitions products etc are also possible once geometric models are available thus for all these activities, geometric models become the central that will be manipulate at all these storages. While modeling, geometric models are prepared in such a way that the other modules such as CAM, CAE, Mechanism systems are able to use this information in the most optimal and efficient way.

## **REQUIREMENTS OF GEOMETRIC MODELING**

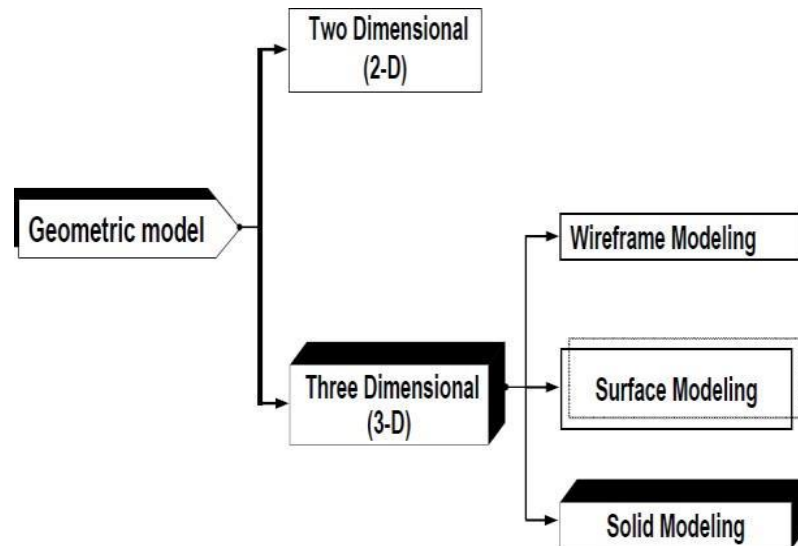
The requirement of geometric modeling is manifold. The conceptual design is the basis of the generation of geometric model. The choice of the geometric model depends on the mechanical functions to be performed by it. A valid geometric model is created by CAD system and its model database is stored. The database of geometric modeling is used for engineering analysis and for design optimization. Design testing and evaluation may necessitate changing the geometric model before finalizing it. When the final design is achieved, it is documented and used for subsequent manufacturing applications, quality and cost analysis.

Good geometric model is designed based in the following important guidelines-

1. Modeling method must be easy to use
2. Completeness of part representation
3. Representation should be able to represent a useful set of geometric objects
4. Model should not create any ambiguity (The ability to express more than one interpretation) to users. A given representation should correspond to one and only one solid
5. To represent the solid normally there should only unique way. With this one can compare two identical solid representations
6. A geometric model must be very accurate. There should not be any approximation
7. Geometric modeling should not create any invalid or impossible models
8. The various transformations such as move, rotation, scale etc and manipulations such as union, intersect, subtract etc should able to be performed on geometric model
9. A good representation should be compact enough for saving space and allow for efficient algorithms to determine desired physical characteristics
10. Geometric model should have shading and rendering (Generation an image for a model by means of a computer program) capability to give realistic effect to the model

## **GEOMETRIC MODELS**

The geometric models are broadly classified on the basis of geometric construction into the



Following categories

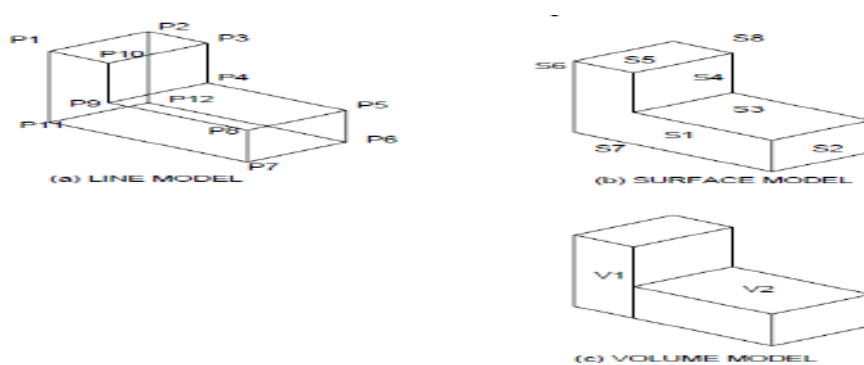
1. Two Dimensional (2D)
2. Three Dimensional (3D)

### Two Dimensional (2D)

The 2D modeling includes the construction of geometrical faces, plane drawings, 2D views (Top, Front, Right, and Light views) of objects. Presently, the application of 2D model is limited to drafting, sheet metal manufacturing, Spot welding, Laser cutting etc.

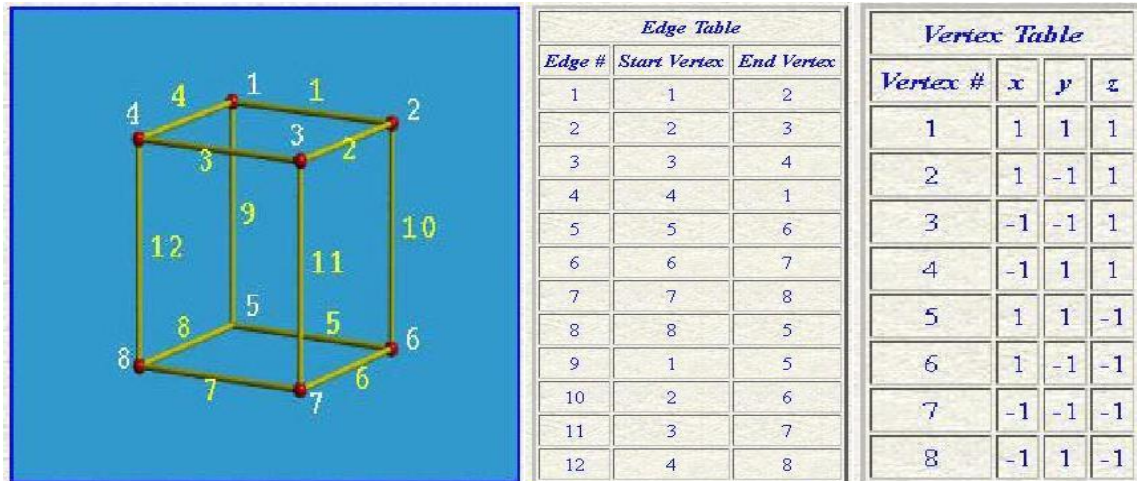
### Three dimensional (3D)

The 3D modeling is widely used for engineering applications. It provides all the information required for animation, design analysis and manufacturing. The 3D objects are shown in below figure.



The 3D models are further sub-divided into **three** groups:

## 1. WIREFRAME MODELING:



Wireframe modeling is the oldest and simplest methods of geometric modeling which can be used to store model mathematically in the computer memory. It contains information about the locations of all the points (vertices) and edges in space coordinates. Various wireframe entities are points, lines, planer arcs, circle, curves etc. Each vertex is defined by x, y, z coordinates. Edges are defined by a pair of vertices and faces are defined as three or more edges. Thus, wireframe is a collection of edges, there is no skin defining the area between the edges. This is the lowest level of modeling and has serious limitations. But it some applications such as tool path simulation it is very convenient to use wireframe models.

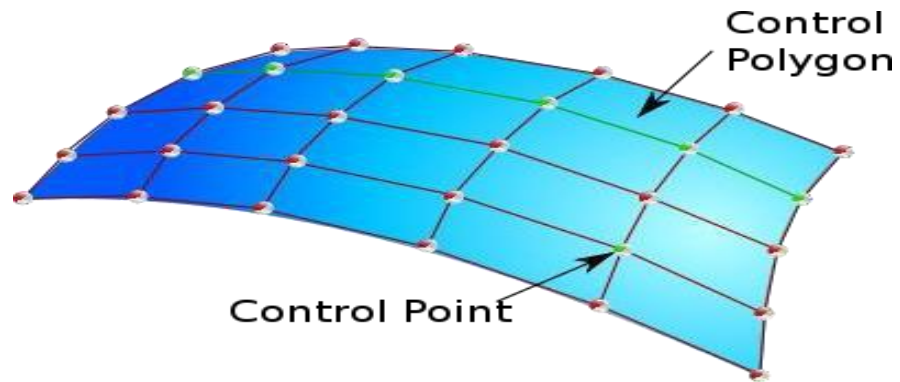
### Advantages of wireframe models:

1. It is simple methods and requires less memory space
2. It forms the basis for surface and solid modeling
3. Manipulations in the model can be done easily and quickly

### Disadvantages:

1. One of the serious limitations in the ambiguity of orientation and viewing plane
2. Cannot model complex curve surfaces
3. Does not represent an actual solid (no surface and volume)
4. Physical properties such as mass, surface area, volume, center of gravity etc. are not possible to calculate
5. Wireframe models has no knowledge of surface faces; therefore, it will not detect interface between two matting components and this is serious drawback especially in component assembly, kinematic analysis, NC tool and robot arm simulation

## 2.SURFACE MODELING:



Surface modeling is the next stage of wireframe modeling. In wireframe modeling models are unable to represent complex surfaces of objects like car, ship, aero plane, wings, castings etc. only a surface profile of these objects. A surface model represents the skin of an object. These skins have no thickness or material type. Surface models define the surface properties, as well as the edges of objects. These are often capable of clearly representing the solid from the manufacturing. However, no information regarding the interior of the solid model would be available which could be relevant for generating the NC cutter data. Further the calculation of properties such as mass and inertia etc would be difficult. Surface modeling facilities would be available as part of the modeling technique and would be used when such surface is present in the product for design. For example, this method is used mode for specific non-analytical surfaces, called sculptures surfaces such as those used for modeling the car bodies and ship-hulls. There are a number of mathematical techniques available for handling these surfaces such as Bezier and B-splines.

### **Advantage:**

1. Eliminates much ambiguity and non-uniqueness present in wireframe models by hiding lines not seen
2. Renders the model for better visualization and presentation, objects appear more realistic
3. Provides the surface geometry for CAM, NC machine
4. Provides the geometry needed by the manufacturing engineer for mould and die design
5. This can be used to design and analysis complex free-formed surfaces of ship hulls, aero plane fuselages and bodies
6. Surface properties such as roughness, color and reflectivity can be assigned and demonstrated

### **Disadvantages:**

1. Provides no information about the inside of an object
2. Curved surfaces need a fine mesh to be accurate
3. Provides wrong results if mesh is too coarse

4. Complicated computation, depending on the number of surfaces

### 3. SOLID MODELING:

Solid modeling is the most powerful of 3D modeling technique. This includes vertices (nodes), edges, surfaces, weight and volume. This model consisting of the complete description of the solid in a certain form is the most ideal representation, as all the information requires at every stage of product cycle can be obtained with technique. Defining an object with a solid model is the easiest of the available three modeling techniques. The model is a complete and unambiguous representation of a precisely enclosed and filled volume. Solid model contain bath geometric and available to represent the solid, but geometry two techniques are very famous, these are constructive solid geometry (CSG) and boundary representation (B-rep)

#### Advantages:

1. Mass properties such as area, volume, weight, centre of gravity and moment of inertia can be determined quickly
2. It allows the design engineer to develop and evaluate alternative concepts for parts and assemblies while the design is still a theoretical model.
3. Solid models are non-ambiguous
4. Easily exported to different FEM programs for analysis
5. It can be used in newly manufacturing techniques; CIM, CAM, design for manufacturing (DFM)
6. 2D standard drawings, assembly drawings and exploded drawings are generated from the model.

#### Disadvantage:

1. More intensive computation than wireframe and surface modeling
2. Requires more powerful computers (faster with the more memory)

#### DIFFERENCE AMONG WIREFRAME, SURFACE AND SOLID MODELING

| Parameter              | Wireframe model  | Surface model  | Solid model   |
|------------------------|--|--|---|
| 1. Computer memory     | Less   | Moderate   | Large   |
| 2. Entities used       | Points, line, circle, arc, ellipse, synthetic curves, such as Bezier, hermite, B-spline etc. | Plane, revolve, ruled, tabulated, free form surfaces | Solid primitives as cone, cube, wedge, cylinder, sphere etc |
| 3. Input data required | More   | Moderate   | Less  |

|   |                            |                                       |  |
|---|----------------------------|---------------------------------------|--|
| 4. Automatic orthographic, perspective, isometric view generation                               | Impossible                 | Impossible                            | Easily possible  |
| 5. NC code generation   | Not possible               | Automatic possible                    | Automatic possible   |
| 6. Interference between mating parts  | Not possible to detect     | Can detect                            | Can detect   |
| 7. Rendering and shadow effect  | Not possible               | possible                              | possible   |
| 8. Cross sectioning   | Not possible done manually | Not possible                          | Possible done automatically  |
| 9. Elimination of hidden lines  | Done manually              | May be possible                       | Possible   |
| 10. Calculation of physical properties such as volume, surface area, center of gravity, M.I etc | Not possible               | Possible to calculate some properties | Possible to calculate all properties required for analysis                   |
| 11. Design parameters optimization  | Not possible               | Not possible                          | Possible   |
| 12. Generation of assembly and detail drawings from model and vice versa                        | Not possible               | Not possible                          | Possible   |
| 13. CAD/CAM/CAE modules   | Drafting                   | Drafting, design                      | Drafting, design, manufacturing, analysis, assembly, mechanism, optimization |

## CURVE REPRESENTATION

- Curve is defined as the locus of point moving with one-degree freedom.
- A curve can be represented by following two methods either by storing its analytical equation or by storing an array of co-ordinates of various points

Curves can be described mathematically by following methods:

- I. Non-parametric form
  - a) Explicit form
  - b) Implicit form
- II. Parametric form

**I. Non-parametric form:** In this, the object is described by its co-ordinates with respect to current reference frame in use.

**a) Explicit form:** In this, the co-ordinates of y and z of a point on curve are expressed as two separate functions of x as independent variable.

$$\begin{aligned}
 P &= [x \quad y \quad z] \\
 &= [x \quad f(x) \quad g(x)]
 \end{aligned}$$

**b) Implicit form:** In this, the co-ordinates of x, y and z of a point on curve are related together by two functions.

$$F(x, y, z) = 0$$

$$G(x, y, z) = 0$$

### Limitation of nonparametric representation of curves are:

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

### Parametric form:

In this, a parameter is introduced and the co-ordinates of x, y and z are expressed as functions of this parameters. This parameter acts as a local co-ordinate for points on curve.

$$P(u) = [x \quad y \quad z] \\ = [x(u) \quad y(u) \quad z(u)]$$

### Classification of Curves

– Curves can be classified as shown below mainly in two categories:

#### 1. Analytical Curves:

- The curves which are defined by the analytical equations are known as analytical curves.
- Examples: line, circle, ellipse, parabola, hyperbola etc.

#### 2. Synthetic Curves:

- The curves which are defined by set of data points are known as synthetic curves.
- Combination of polynomial segments to represent a desired curve is called a synthetic curve.
- These curves can be conveniently twisted, shaped or bent by changing the control points.
- Examples: Cubic spline curve, B-spline curve, Bezier curve etc.
- Synthetic curves take up where the analytic curves leave – the latter are not that efficient at geometric design of mechanical parts.
- Need for synthetic curves arise in two occasions:
  - When a curve is represented by a collection of measured data points,
  - When an existing curve must change to meet new design requirements the designer needs a curve representation that is directly related to the data points and is flexible enough to bend, twist or change the shape by changing one or more data points.
- Some examples of complex geometric design are: Car bodies, Ship hulls, Airplane fuselage and wings, Propeller blades, Shoe insoles, aesthetically designed bottles etc.
- When the curve pass through all the data points, then the curve is known as Interpolant curve. It sounds more logical but can be more unstable and ringing.
- When a smooth curve is approximated through the data points, then the curve is



known as approximation curve. It turns out to be convenient.

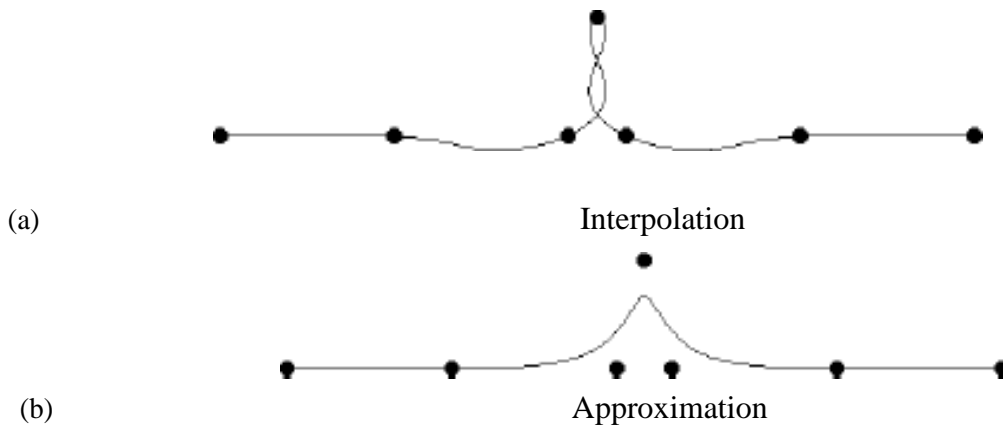


Fig.2.1 Interpolant and Approximated Curves

- Synthetic curve pass through defined data points and can thus represented by polynomial equations.
- The parametric form for any synthetic curve is:

$$P(u) = C_3 u^3 + C_2 u^2 + C_1 u + C_0$$

where,  $u$  = Parameter and  $C_i$  = Polynomial coefficients

### Parametric representation of analytical curve

#### ➤ Lines

- A line can be represented in a parametric form by an input of two endpoints or by one point and a length and direction. Besides these, modifiers and filters can be used to represent these lines. The two important methods of establishing equations for lines are considered.

**A line connecting two endpoints in parametric form, having the values of parameter  $u$  as 0 and 1 at the endpoints:**

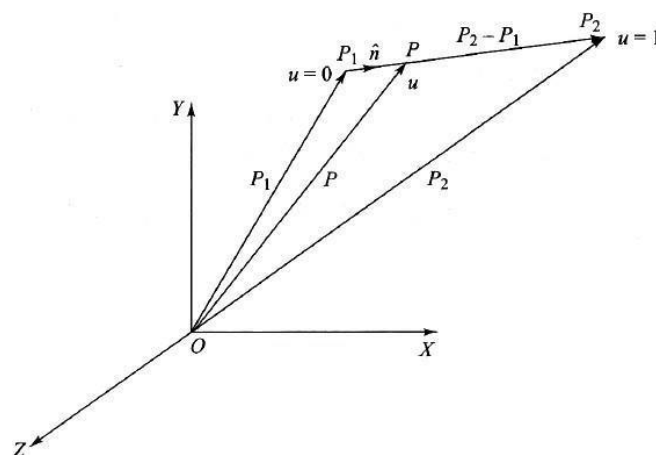


Fig 2.2 Line between Two Endpoints

- Consider a line between two given endpoints  $P_1(x_1, y_1, z_1)$  and  $P_2(x_2, y_2, z_2)$  as shown in Fig. 2.2. The position vectors for these two points would be  $P_1$  and  $P_2$ . Consider any point  $P(x, y, z)$  on the line  $P_1 P_2$ .

- Let a parametric system be introduced such that the point P is represented by the parameter  $u$  such that its values are 0 and 1 at endpoints  $P_1$  and  $P_2$  respectively.
- From  $\Delta OPP_1$ , we can express the vector between endpoints  $P_1$  and P as  $P - P_1$ . This vector represents the parameter  $u$ .

The mapping between the Cartesian and parametric space can be done by the following relation:

$$\frac{\text{Distance of P from } P_1 \text{ in Cartesian space}}{\text{Distance of P from } P_1 \text{ in parametric space}} = \frac{\text{Endpoint Length in Cartesian space}}{\text{Endpoint Length in parametric space}}$$

$$\frac{P - P_1}{u} = \frac{P_2 - P_1}{1}$$

$$P - P_1 = u(P_2 - P_1)$$

$$P = P_1 + u(P_2 - P_1) \quad \text{where, } 0 \leq u \leq 1$$

- This equation represents the vector equation of the line in parametric form. In scalar form, it can be written as:

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

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

$$z = z_1 + u(z_2 - z_1) \quad \text{where, } 0 \leq u \leq 1$$

- Any point on the above line or its extension can be found by substituting the value of Parameter  $u$ . The tangent vector of the line would be represented by:

$$P' = P_2 - P_1$$

- In scalar form these would have the coordinates expressed by the following equations:

$$x' = x_2 - x_1$$

$$y' = y_2 - y_1$$

$$z' = z_2 - z_1$$

- It can be seen that the tangent vector is independent of the parameter  $u$ , thus representing the constant slope of the line. Above equations are sufficient to represent all properties of the line for CAD/CAM representation. The length  $L$  and direction vector  $n$  can be calculated using the following equations :

$$L = |P_2 - P_1|$$

$$= \sqrt{\left(\frac{x_2 - x_1}{1}\right)^2 + \left(\frac{y_2 - y_1}{1}\right)^2 + \left(\frac{z_2 - z_1}{1}\right)^2}$$

$$\hat{n} = \frac{P_2 - P_1}{L}$$

**A line passing through a point and defined by a length and direction:**

Consider a case, where a line is to pass through a point  $P_1$  and has its length  $L$  and direction vector  $n$  defined as input as shown in Fig. 2.3. The computation of the endpoint of this line needs to be done based on the above inputs. As seen earlier,

$$\hat{n} = \frac{P - P_1}{L}$$

$$P = P_1 + L(\hat{n})$$

where,  $-\infty \leq L \leq \infty$

Once the user inputs  $P_1$ ,  $L$  and  $n$ , the endpoint can be calculated as per the above equation. The two endpoints can then be expressed in parametric form with values of parameter  $u$  as 0 and 1.

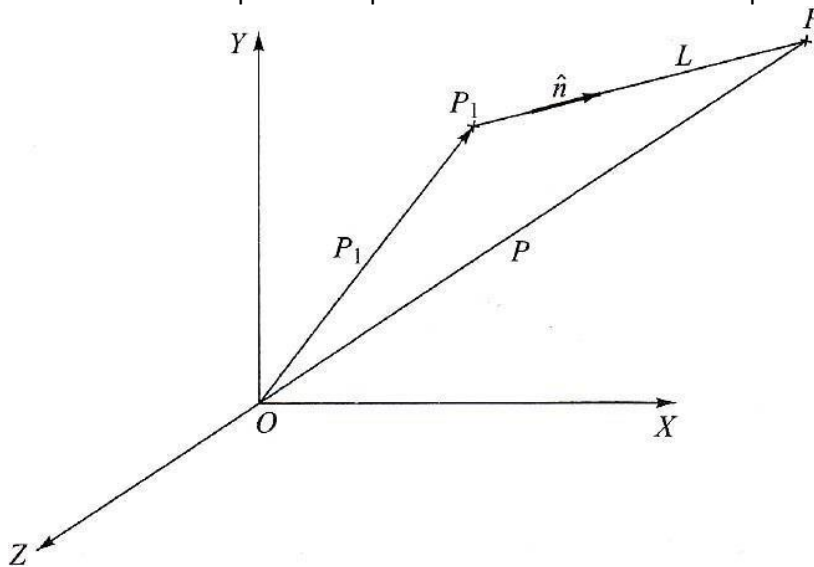


Fig. 2.3 Line passing through a point and defined by a Length and Direction

### Parallel Lines

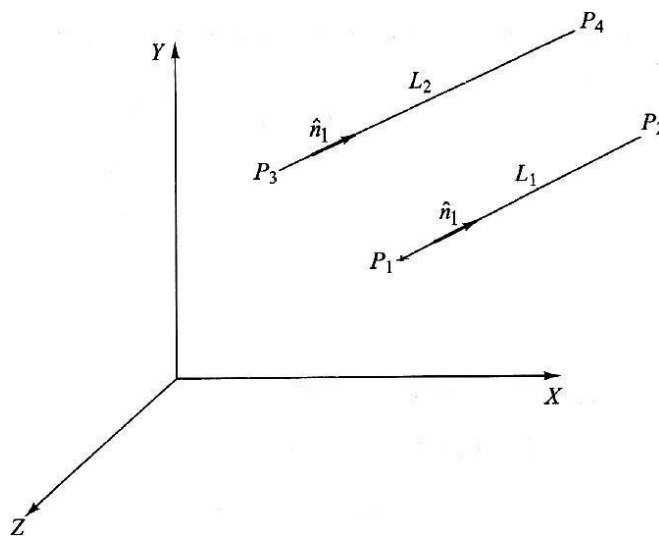


Fig. 2.4 Parallel Lines

- Assume that the existing line has the two endpoints  $P_1$  and  $P_2$ , a length  $L_1$ , and a

direction defined by the unit vector length  $L_2$  and endpoints  $P_3$  and  $P_4$ .

The unit vector  $n_1$  is given by

- The unit vector for the parallel line will be same as unit vector for the existing line.

$$\hat{n} = \frac{P_4 - P_3}{L_2}$$

$$P_4 = P_3 + L_2 \hat{n}$$

- The equation of parallel line becomes,

$$P = P_3 + u(P_4 - P_3)$$

➤ **Angle between two lines**

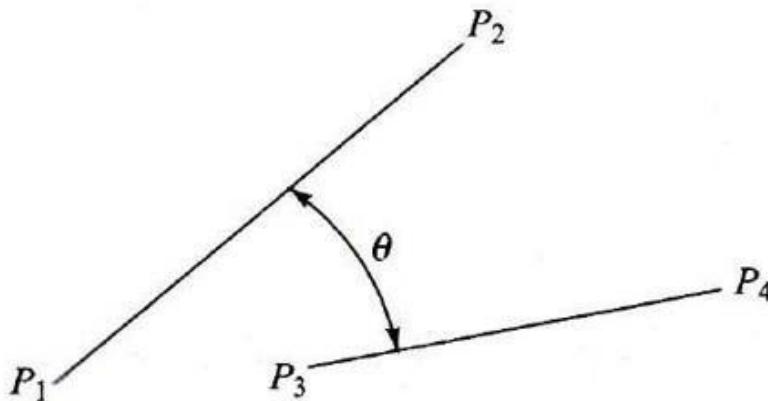


Fig. 2.5

- Fig. 2.5 represents two lines  $L_1$  and  $L_2$  having endpoints are  $P_1, P_2$  and  $P_3, P_4$ , the angle measurement command uses the equation

$$\cos\theta = \frac{(P_2 - P_1) \cdot (P_4 - P_3)}{|P_2 - P_1| |P_4 - P_3|}$$

$$\cos\theta = L_1 \cdot L_2$$

**Distance of a point from line**

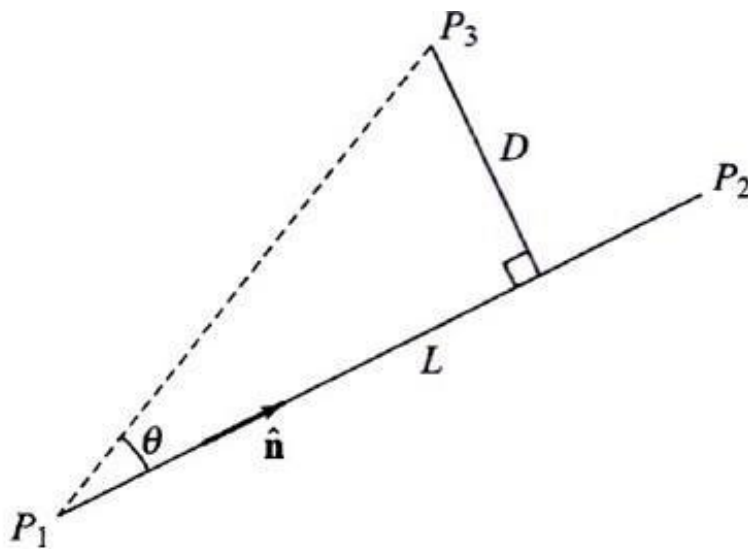


Fig. 2.6

Fig. 2.6 shows the distance  $D$  from point  $P_3$  to the line whose endpoints are  $P_1$  and  $P_2$  and direction is  $\hat{n}$ . Its length is  $L$ .  $D$  is given by

$$D = \frac{|(P_3 - P_1) \times \hat{n}|}{|\hat{n}|} = \frac{|(P_3 - P_1) \times \frac{(P_2 - P_1)}{|P_2 - P_1|}|}{1}$$

**Example 2.1:** For the position vectors  $P_1[1 \ 2]$  and  $P_2[4 \ 3]$ , determine the parametric representation of line segment between them. Also determine the slope and tangent vector of line segment.

A parametric representation of line is

$$\begin{aligned} P &= P_1 + u(P_2 - P_1) \\ &= [1 \ 2] + u([4 \ 3] - [1 \ 2]) \\ &= [1 \ 2] + u[3 \ 1] \end{aligned}$$

Parametric representation of x and y components are

$$\begin{aligned} x(u) &= x_1 + u(x_2 - x_1) \\ &= 1 + 3u \\ y(u) &= y_1 + u(y_2 - y_1) \\ &= 2 + u \end{aligned}$$

The tangent vector is obtained by differentiating  $P(u)$

$$\begin{aligned} P'(u) &= \frac{dx}{du} \hat{i} + \frac{dy}{du} \hat{j} \\ &= [3 \ 1] \end{aligned}$$

The slope of line segment is

$$\begin{aligned} \frac{dy}{dx} &= \frac{dy/du}{dx/du} \\ &= \frac{1}{3} \end{aligned}$$

## Circles

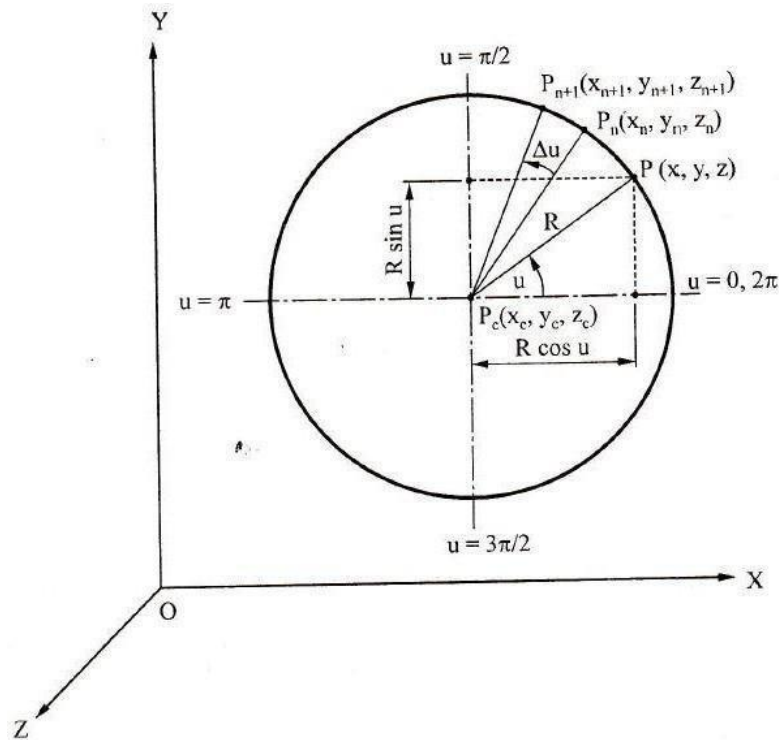


Fig. 2.7 Circle in Parametric Form

- A circle is represented in the CAD/CAM database by storing the values of its center and its radius. In a parametric form, a circle shown in Fig. 2.7, can be represented by the following set of equations:

$$x = x_c + R \cos u$$

$$y = y_c + R \sin u$$

$$z = z_c$$

$$\text{where, } 0 \leq u \leq 2\pi$$

- In this case the parameter  $u$  represents the angle measured from the  $X$  axis to any point  $P$  on the circumference of the circle. By incrementing the values of  $u$  from  $0$  to  $2\pi$ , the intermediate points on the circle can be worked out.
- These points can then be connected by straight lines for the purpose of display. However, this method of computation is inefficient as it requires trigonometric evaluation of functions.
- A convenient method to determine the different points on the circle is to use the incremental equations. Consider any point  $P_n(x_n, y_n, z_n)$  on the circle.

Let the subsequent incremental point be  $P_{n+1}(x_{n+1}, y_{n+1}, z_{n+1})$ . From the above equation, we get:

$$\begin{aligned} x_n &= x_c + R \cos u \\ \therefore R \cos u &= x_n - x_c \end{aligned} \quad (2.1)$$

$$\begin{aligned} y_n &= y_c + R \sin u \\ \therefore R \sin u &= y_n - y_c \end{aligned} \quad (2.2)$$

- Thus, the circle can start from an arbitrary point and successive points with equal spacing

can be calculated.  $\cos \Delta u$  &  $\sin \Delta u$  have to be cal

Similarly, 
$$x_{n+1} = x_c + R \cos (u + \Delta u) \quad (2.3)$$

$$y_{n+1} = y_c + R \sin (u + \Delta u) \quad (2.4)$$

$$z_{n+1} = z_n \quad (2.5)$$

From (2.3), (2.1) & (2.2)

$$x_{n+1} = x_c + R \cos u \cos \Delta u - R \sin u \sin \Delta u$$

$$\therefore x_{n+1} = x_c + (x_n - x_c) \cos \Delta u - (y_n - y_c) \sin \Delta u \quad (2.6)$$

From (2.4), (2.1) & (2.2)

$$y_{n+1} = y_c + R \sin u \cos \Delta u + R \cos u \sin \Delta u$$

$$\therefore y_{n+1} = y_c + (y_n - y_c) \cos \Delta u + (x_n - x_c) \sin \Delta u \quad (2.7)$$

And  $z_{n+1} = z_c$

culated only once, this eliminates the computations of trigonometric functions for each point.

- This algorithm is useful for hardware implementation to speed up the circle generation & display.
- If two endpoints of diameter of circle are given, then the center & radius of circle are calculated as below.

## Ellipse

- Mathematically the ellipse is a curve generated by a point moving in space such that at any position the sum of its distances from two fixed points (foci) is constant and equal to the major diameter. Each focus is located on the major axis of the ellipse at a distance from its center equal to  $\sqrt{A^2 - B^2}$  (A and B are the major and minor radii). Circular holes and forms become ellipses when they are viewed obliquely relative to their planes

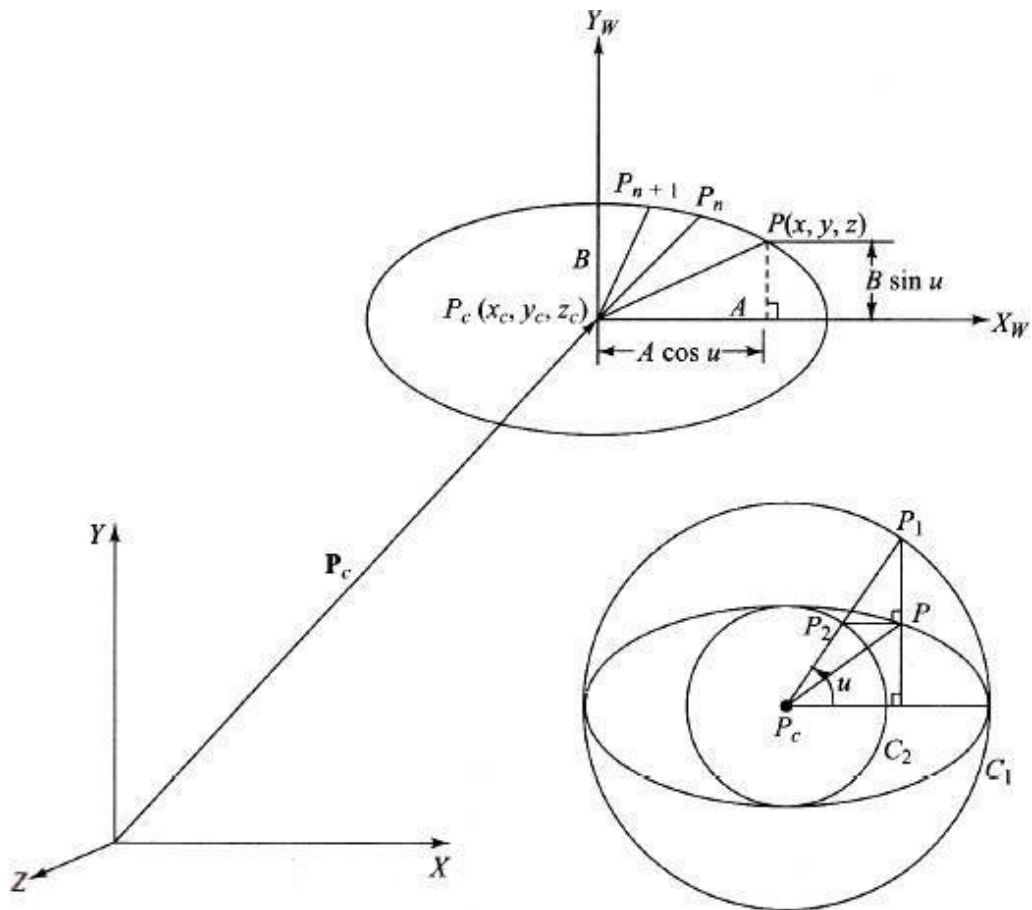


Fig. 2.8 Ellipse Defined by a Center, Major and Minor Axes

- However, four conditions (points and/or tangent vectors) are required to define the geometric shape of an ellipse.
- Fig. 2.8 shows an ellipse with point  $P_c$  as the center and the lengths of half of the major and minor axes are  $A$  and  $B$  respectively. The parametric equation of an ellipse can be written as assuming the plane of the ellipse is the  $XY$  plane.
- The parameter  $u$  is the angle as in the case of a circle. However, for a point  $P$  shown in the figure, it is not the angle between the line  $PP_c$  and the major axis of the ellipse. Instead, it is defined as shown.
- To find point  $P$  on the ellipse that corresponds to an angle  $u$ , the two concentric circles  $C_1$  and  $C_2$  are constructed with centers at  $P_c$  and radii of  $A$  and  $B$  respectively. A radial line is constructed at the angle  $u$  to intersect both circles at points  $P_1$  and  $P_2$  respectively.

If a line parallel to the minor axis is drawn from  $P_1$  and a line parallel to the major axis is drawn from  $P_2$ , the intersection of these two lines defines the point  $P$ .

$$\begin{aligned}
 x &= x_c + A \cos u \\
 y &= y_c + B \sin u \\
 z &= z_c
 \end{aligned}
 \quad \left| \quad \begin{aligned}
 & \\
 & \\
 & 0 \leq u \leq 2\pi
 \end{aligned}
 \right.$$

Similar development as in the case of a circle results in the following recursive relationships which are useful for generating points on the ellipse for display purposes without excessive evaluations of trigonometric functions:



$$\begin{aligned}
x_{n+1} &= x_c + (x_n - x_c) \cos \Delta u - \frac{A}{R} (y_n - y_c) \sin \Delta u \\
y_{n+1} &= y_c + (y_n - y_c) \cos \Delta u + \frac{B}{R} (x_n - x_c) \sin \Delta u \\
z_{n+1} &= z_n
\end{aligned} \tag{2.8}$$

If the ellipse major axis is inclined with an angle  $\alpha$  relative to the x axis as shown in Fig. 2.8, the ellipse equation becomes

$$x = x_c + A \cos u \cos \alpha - B \sin u \sin \alpha$$

$$y = y_c + A \cos u \sin \alpha + B \sin u \cos \alpha$$

$$z = z_c$$

### Parabola

- The parabola is defined mathematically as a curve generated by a point that moves such that its distance from a fixed point (the focus  $P_F$ ) is always equal to its distance to a fixed line (the directrix) as shown in Fig. 2.9.
- The vertex  $P_V$  is the intersection point of the parabola with its axis of symmetry. It is located midway between directrix and focus. Focus lies on the axis of symmetry.
- Useful applications of the parabolic curve in engineering design include its use in parabolic sound and light reflectors, radar antennas and in bridge arches.
- Three conditions are required to define a parabola, a parabolic curve, or a parabolic arc. The default plane of a parabola is the XY plane of the current WCS at the time of construction.
- The database of a parabola usually stores the coordinates of its vertex, distances  $y_{HW}$  and  $y_{LW}$ , that define its endpoints as shown in Fig. 2.9, the distance A between the focus and the vertex (the focal distance) and the orientation angle  $\alpha$ .
- Unlike the ellipse, the parabola is not a closed curve. Thus, the two endpoints determine the amount of the parabola to be displayed.

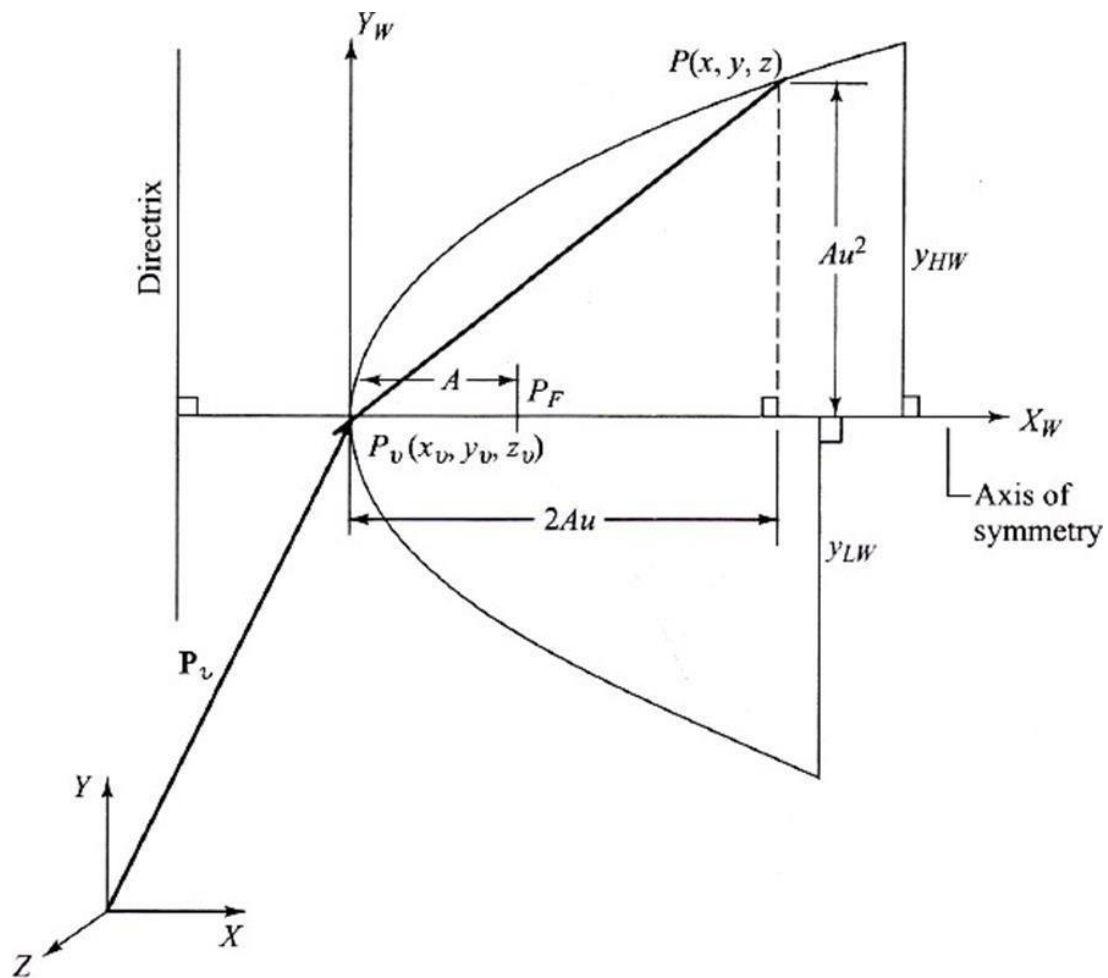


Fig. 2.9 Basic geometry of Parabola

- Assuming the local coordinate system of the parabola as shown in Fig. 2.9, its parametric equation can be written as

$$\left. \begin{aligned} x &= x_v + Au^2 \\ y &= y_v + 2Au \\ z &= z_v \end{aligned} \right\} 0 \leq u \leq \infty$$

- If the range of the y coordinate is limited to  $y_{HW}$  and  $y_{LW}$  for positive and negative values respectively, the corresponding u values become

$$u_H = \frac{y_{HW}}{2A}$$

$$u_L = \frac{y_{LW}}{2A}$$

- The recursive relationships to generate points on the parabola are obtained by substituting  $u_n + \Delta u$  for points  $n + 1$ . This gives

$$x_{n+1} = x_n + (y_n - y_v)\Delta u + A(\Delta u)^2$$

$$y_{n+1} = y_n + 2A \Delta u$$

$$z_{n+1} = z_n$$

## Hyperbolas

A hyperbola is described mathematically as a curve generated by a point moving such that at any position the difference of its distances from the fixed points (foci)  $F$  and  $F'$  is a constant and equal to the transverse axis of the hyperbola

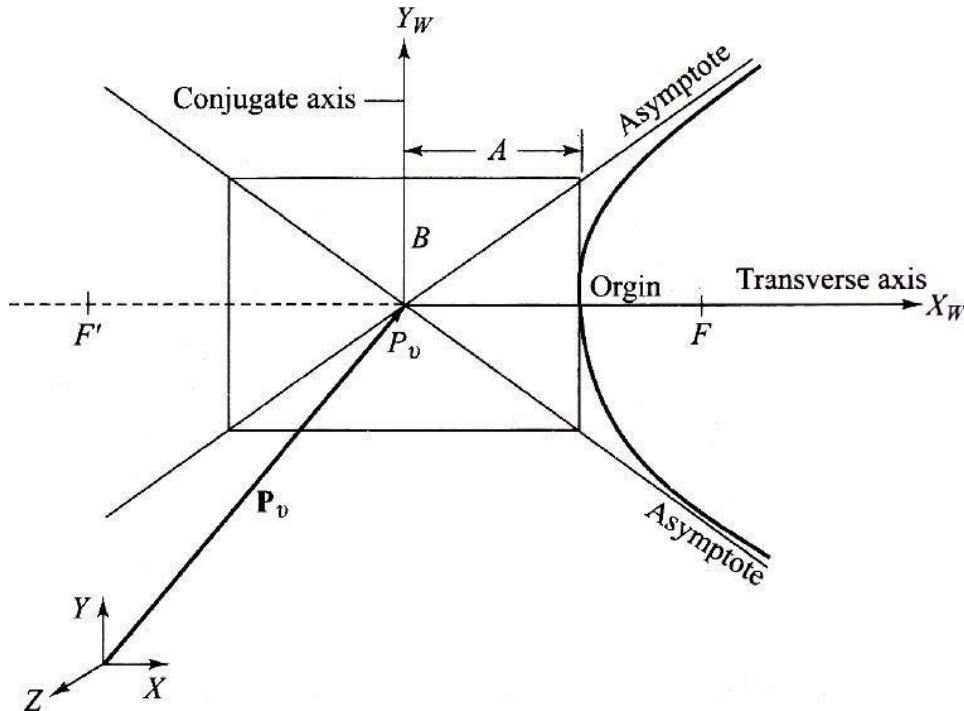


Fig. 2.10 Hyperbola Geometry

- Fig. 2.10 shows the geometry of a hyperbola.
- The parametric equation of a hyperbola is given by

$$x = x_v + A \cosh u$$

$$y = y_v + B \sinh u$$

$$z = z_v$$

- This equation is based on the nonparametric implicit equation of the hyperbola which can be written as

$$\frac{(x - x_v)^2}{A^2} - \frac{(y - y_v)^2}{B^2} = 1$$

by utilizing the identity  $\cosh^2 u - \sinh^2 u = 1$ .

## Conics

- Conic curves or conic sections form the most general form of quadratic curves. Lines, circles, ellipses, parabolas and hyperbolas are all special forms of conic curves. They all can be generated when a right circular cone of revolution is cut by planes at different angles relative to the cone axis—thus the derivation of the name conics.
- Straight lines result from intersecting a cone with a plane parallel to its axis and passing through its vertex. Circles result when the cone is sectioned by a plane

perpendicular to its axis while ellipses, parabolas and hyperbolas are generated when the plane is inclined to the axis by various angles.

### Parametric representation of synthetic curve

- Analytic curves, are usually not sufficient to meet geometric design requirements of mechanical parts. Products such as car bodies, ship hulls, airplane fuselage and wings, propeller blades, shoe insoles and bottles are a few examples that require free-form, or synthetic, curves and surfaces.
- The need for synthetic curves in design arises on two occasions: when a curve is represented by a collection of measured data points and when an existing curve must change to meet new design requirements.
- In the latter occasion, the designer would need a curve representation that is directly related to the data points and is flexible enough to bend, twist, or change the curve shape by changing one or more data points.
- Data points are usually called control points and the curve itself is called an interpolant if it passes through all the data points.

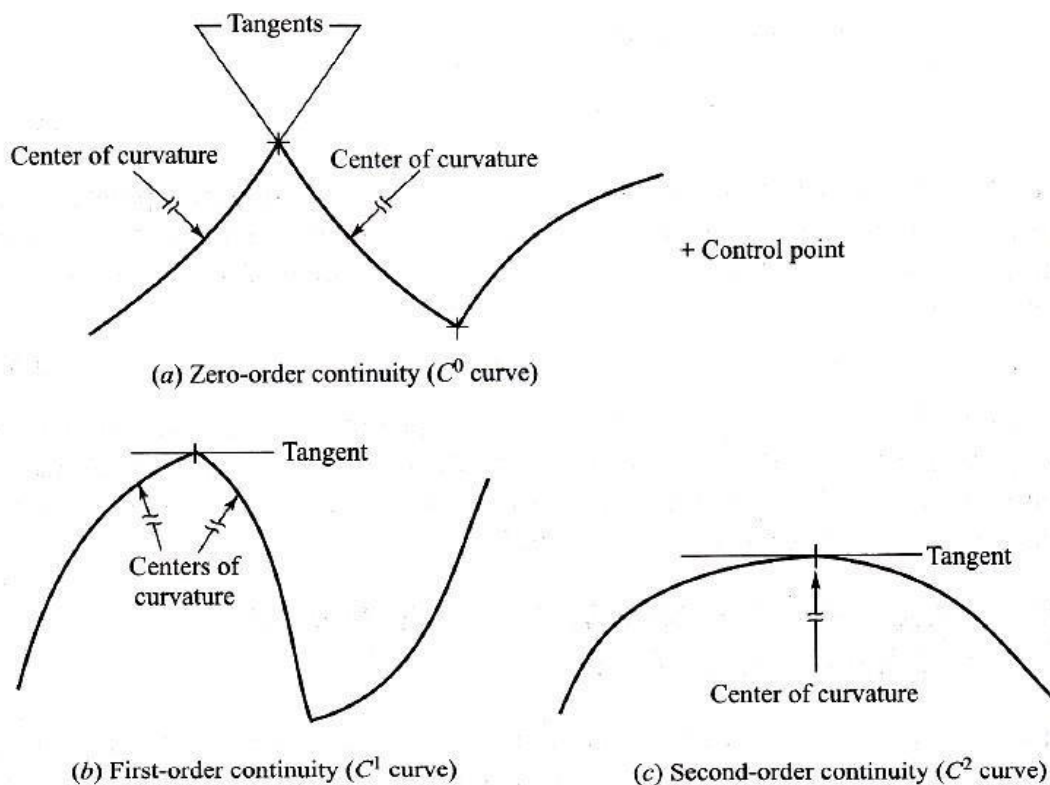


Fig. 2.11 Various Orders of Continuity of Curves

- Mathematically, synthetic curves represent a curve-fitting problem to construct a smooth curve that passes through given data points. Therefore, polynomials are the typical form of these curves.
- Various continuity requirements can be specified at the data points to impose various degrees of smoothness of the resulting curve. The order of continuity becomes important when a complex curve is modeled by several curve segments pieced together end to end.
- Zero-order continuity ( $C^0$ ) yields a position continuous curve. First ( $C^1$ )- and second ( $C^2$ )-order continuities imply slope and curvature continuous curves respectively.

- A  $C^1$  curve is the minimum acceptable curve for engineering design. Fig. 2.11 shows geometrical interpretation of these orders of continuity.
- A cubic polynomial is the minimum order polynomial that can guarantee the generation of  $C^0$ ,  $C^1$ , or  $C^2$  curves. In addition, the cubic polynomial is the lowest-degree polynomial that permits inflection within a curve segment and that allows representation of nonplanar (twisted) three dimensional curves in space.
- Higher-order polynomials are not commonly used in CAD/CAM because they tend to oscillate about control points, are computationally inconvenient and are uneconomical of storing curve and surface representations in the computer.
- The type of input data and its influence on the control of the resulting synthetic curve determine the use and effectiveness of the curve in design.
- For example, curve segments that require positions of control points and/or tangent vectors at these points are easier to deal with and gather data for than those that might require curvature information.
- Also, the designer may prefer to control the shape of the curve locally instead of globally by changing the control point(s). If changing a control point results in changing the curve locally in the vicinity of that point, local control of the curve is achieved; otherwise global control results.
- Major CAD/CAM systems provide three types of synthetic curves: Hermite cubic spline, Bezier and B-spline curves. The cubic spline curve passes through the data points and therefore is an interpolant.
- Bezier and B-spline curves in general approximate the data points, that is, they do not pass through them. Under certain conditions, the B-spline curve can be an interpolant. Both the cubic spline and Bezier curves have a first-order continuity and the B-spline curve has a second-order continuity.

#### **Hermite Cubic Spline Curve:**

- They are used to interpolate the given data but not to design free-form curves. Splines derive their name from “French curves or splines”
- It connects two data (end) points and utilizes a cubic equation.
- Four conditions are required to determine the coefficients of the equation – two end points and the two tangent vectors at these two points.
- It passes through the control points and therefore it is an Interpolant. It has only up to  $C_1$  continuity.
- The curve cannot be modified locally, i.e., when a data point is moved, the entire curve is affected, resulting in a global control.
- The order of the curve is always constant (cubic), regardless of the number of data points. Increase in the number of data points increases shape flexibility, however, this requires more data points, creating more splines that are joined together (only two data points and slopes are utilized for each spline).

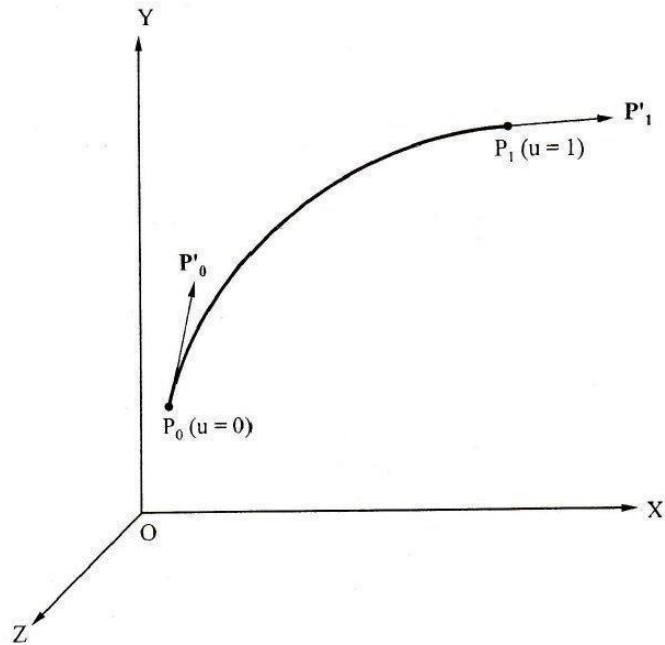


Fig. 2.12 Hermite Cubic Spline Curve

- A cubic spline curve utilizes a cubic equation of the following form:

$$P(u) = C_3 u^3 + C_2 u^2 + C_1 u + C_0 \quad (2.10)$$

where,  $0 \leq u \leq 1$

- The cubic spline is defined by the two endpoints  $P_0$ ,  $P_1$  and the tangent vectors at these points. The tangent vector can be found by differentiating equ.(2.10).

$$P'(u) = 3C_3 u^2 + 2C_2 u + C_1 \quad (2.11)$$

- To determine the coefficients of the curve, consider the two endpoints  $P_0$ ,  $P_1$  and the tangent vectors as shown in Fig. 2.12.

- Applying conditions at  $u = 0$ , in equations (2.10) and (2.11)

$$P_0 = C_0 \quad (2.12)$$

$$P'_0 = C_1 \quad (2.13)$$

- Applying conditions at  $u = 1$ , in equations (2.10) and (2.11)

$$P_1 = C_3 + C_2 + C_1 + C_0 \quad (2.14)$$

$$P'_1 = 3C_3 + 2C_2 + C_1 \quad (2.15)$$

- Substituting (2.12) and (2.13) in (2.14) and (2.15), we get:

$$P_1 = C_3 + C_2 + P'_0 + P_0 \quad (2.16)$$

$$P'_1 = 3C_3 + 2C_2 + P'_0 \quad (2.17)$$

- By solving simultaneous equations (2.16) and (2.17), we get

$$\begin{aligned} 3P_1 - P'_1 &= C_2 + 2P'_0 + 3P_0 \\ \therefore C_2 &= 3P_1 - P'_1 - 2P'_0 - 3P_0 \end{aligned} \quad (2.18)$$

- Similarly again by solving simultaneous equations (2.16) and (2.17), we get

$$\begin{aligned} P'_1 - 2P_1 &= C_3 - P'_0 - 2P_0 \\ \therefore C_3 &= P'_1 - 2P_1 + P'_0 + 2P_0 \end{aligned} \quad (2.19)$$

- Substituting (2.12), (2.13), (2.18) and (2.19) in (2.10), we get:

$$\begin{aligned}
P(u) &= (P'_1 - 2P'_0 + P'_1 + 2P'_0)u^3 + (3P_1 - P'_1 - 2P'_0 - 3P_0)u^2 + P'_1 u + P_0 \\
\therefore P(u) &= P'_1 u^3 - 2P'_0 u^3 + P'_1 u^3 + 2P_1 u^3 + 3P_0 u^2 - P'_1 u^2 - 2P'_0 u^2 - 3P_0 u^2 + P'_1 u + P_0 \\
\therefore P(u) &= 2P_1 u^3 - 3P_0 u^2 + P_0 - 2P_1 u^3 + 3P_0 u^2 + P'_1 u^3 - 2P'_0 u^2 + P'_1 u + P_0 u^3 - P'_1 u^2 \\
\therefore P(u) &= \underline{P_0}(2u^3 - 3u^2 + 1) + P_1(-2u^3 + 3u^2) + P'_1(u^3 - 2u^2 + u) + P'_0(u^3 - u^2) \quad (2.20)
\end{aligned}$$

- The expression can be written in matrix form as under:

$$\begin{aligned}
P(u) &= \begin{bmatrix} P_0 & P_1 & P'_0 & P'_1 \end{bmatrix} \begin{bmatrix} 2u^3 - 3u^2 + 1 \\ -2u^3 + 3u^2 \\ u^3 - 2u^2 + u \\ u^3 - u^2 \end{bmatrix} \\
\therefore P(u) &= \begin{bmatrix} P_0 & P_1 & P'_0 & P'_1 \end{bmatrix} \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}
\end{aligned}$$

- On similar lines, the tangent vector equation can be determined by differentiating equation (2.20)

$$P'(u) = \underline{P'_0}(6u^2 - 6u) + P_1(-6u^2 + 6u) + P'_1(3u^2 - 4u + 1) + P'_0(3u^2 - 2u)$$

$$\begin{aligned}
\therefore P'(u) &= \begin{bmatrix} P_0 & P_1 & P'_0 & P'_1 \end{bmatrix} \begin{bmatrix} 6u^2 - 6u \\ -6u^2 + 6u \\ 3u^2 - 4u + 1 \\ 3u^2 - 2u \end{bmatrix} \\
\therefore P'(u) &= \begin{bmatrix} P_0 & P_1 & P'_0 & P'_1 \end{bmatrix} \begin{bmatrix} 0 & 6 & -6 & 0 \\ 0 & -6 & 6 & 0 \\ 0 & 3 & -4 & 1 \\ 0 & 3 & -2 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}
\end{aligned}$$

Changing the values of endpoints or tangent vectors would modify the shape of the curve.

## Bezier Curves

- Cubic splines are based on interpolation techniques. Curves resulting from these techniques pass through the given points.
- Another alternative to create curves is to use approximation techniques which produce curves that do not pass through the given data points. Instead, these points are used to control the shape of the resulting curves.
- Most often, approximation techniques are preferred over interpolation techniques in curve design due to the added flexibility and the additional intuitive feel provided by the former. Bezier and B-spline curves are examples based on approximation techniques.
- As its mathematics show shortly, the major differences between the Bezier curve and the cubic spline curve are:
  1. The shape of Bezier curve is controlled by its defining points only. First derivatives are not used in the curve development as in the case of the cubic spline. This allows the designer a much better feel for the relationship between input (points) and output (curve).
  2. The order or the degree of Bezier curve is variable and is related to the number of points defining it;  $n + 1$  points define an  $n$ th degree curve which permits higher- order

continuity. This is not the case for cubic splines where the degree is always cubic for a spline segment.

- The Bezier curve is smoother than the cubic spline because it has higher order derivatives.

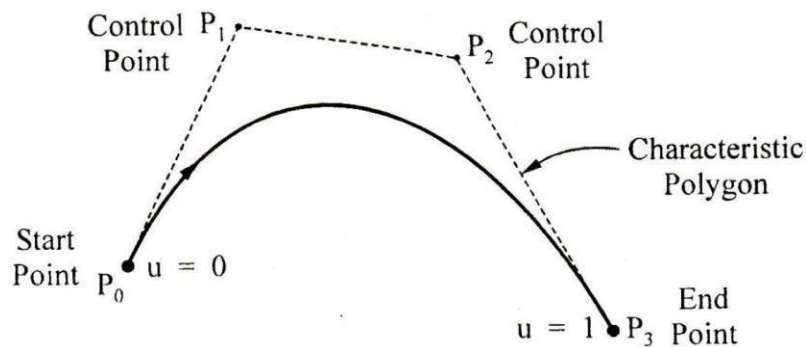


Fig. 2.15 Beizer Curve

- The Bezier curve is defined in terms of  $(n+1)$  points. This points are called as control points, where  $n$  is the degree of the curve.
- If  $n=3$  control points are 4, as shown in the Fig 2.15.
- These control points form the vertices of the control polygon or Bezier characteristic polygon.
- The control or Bezier characteristic polygon uniquely defines the curve.

Properties of Beizer Curve:

- The degree of polynomial defining the curve segment is one less than the no. of defining polygon points  $(n-1)$ .
- The curve follows the shape of the defining polygon.
- Only the first and last control points or vertices of polygon actually lie on curve.
- The other vertices define order & shape of the curve. (see Fig. 3.8)

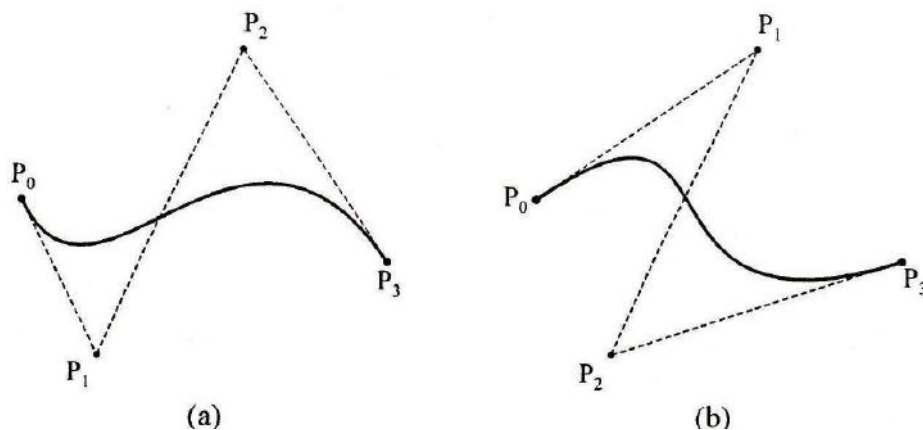


Fig. 2.16 Same data points but different orders for Beizer Curve

- The curve is also always tangent to first and last segments of polygon.
- The same Beizer curve would be generated, if the sequence of the points is changed from  $P_0 - P_1 - P_2 - P_3$  to  $P_3 - P_2 - P_1 - P_0$ .
- The flexibility of the shape would increase with increase in number of vertices of the polygon.

It is having global control on the shape of the curve.



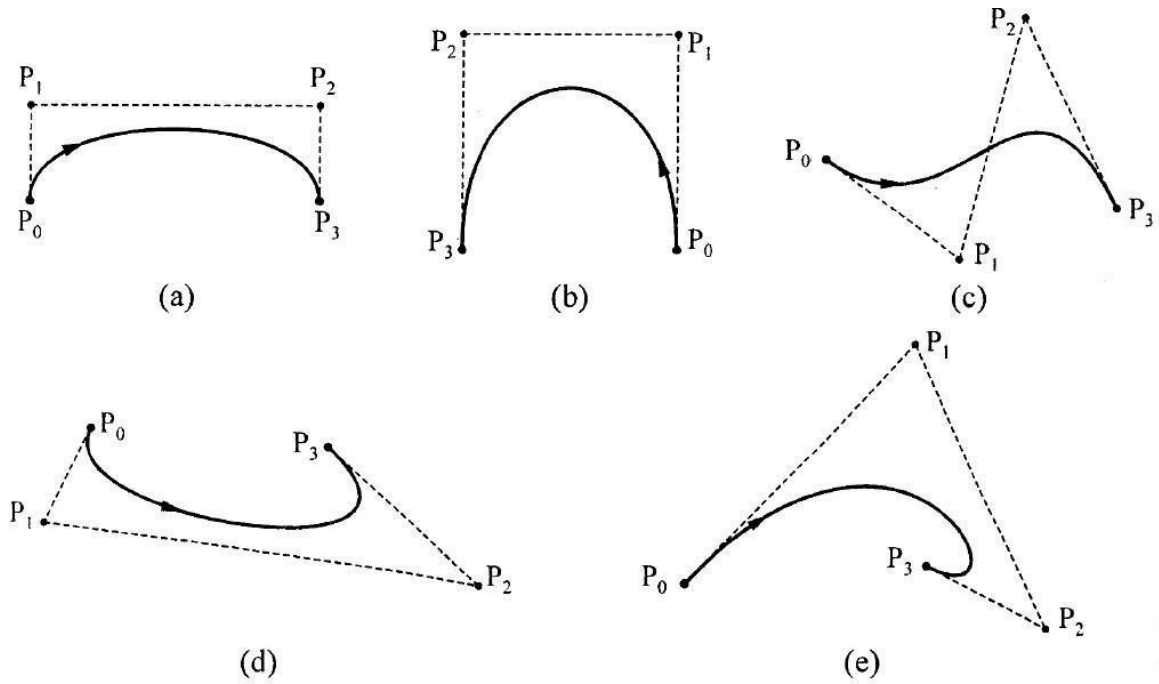


Fig. 2.17 Some Typical Examples of Beizer Curve

- Mathematically, for  $n + 1$  control points, the Bezier curve is defined by the following polynomial of degree  $n$ :

$$P(u) = \sum_{i=0}^n P_i B_{i,n}(u), \quad 0 \leq u \leq 1 \quad (2.23)$$

- where  $P(u)$  is any point on the curve and  $P_i$  is a control point.  $B_{i,n}$  are the Bernstein polynomials. Thus, the Bezier curve has a Bernstein basis. The Bernstein polynomial serves as the blending or basis function for the Bezier curve and is given by

$$B_{i,n}(u) = \frac{C(n,i) u^i (1-u)^{n-i}}{n!} \quad (2.24)$$

- where  $C(n,i)$  is the binomial coefficient

$$C(n,i) = \frac{n!}{i!(n-i)!} \quad (2.25)$$

- Utilizing Eqs. (2.24) and (2.25) and observing that  $C(n,0) = C(n,n) = 1$ ,

- Eq. (2.23) can be expanded to give

$$P(u) = P_0(1-u)^n + P_1 C(n,1) u(1-u)^{n-1} + P_2 C(n,2) u^2(1-u)^{n-2} + \dots + P_{n-1} C(n,n-1) u^{n-1}(1-u) + P_n u^n, \quad 0 \leq u \leq 1$$

- Bezier curve with 3 control points

$$P(u) = P_0(1-u)^2 + P_1 C(2,1) u(1-u)^{2-1} + P_2 C(2,2) u^2(1-u)^{2-2}$$

$$P(u) = (1-u)^2 P_0 + 2u(1-u) P_1 + u^2 P_2$$

- Bezier curve with 4 vertices

$$P(u) = P_0(1-u)^3 + P_1 C(3,1) u(1-u)^{3-1} + P_2 C(3,2) u^2(1-u)^{3-2} + P_3 C(3,3) u^3(1-u)^{3-3}$$

$$P(u) = (1-u)^3 P_0 + 3u(1-u)^2 P_1 + 3u^2(1-u) P_2 + u^3 P_3$$

- Bezier curve with 5 vertices

$$P(u) = P_0(1-u)^4 + 4P_1(1-u)^3u + 6P_2(1-u)^2u^2 + 4P_3(1-u)u^3 + P_4u^4$$

**Spline Curve:**

- B-Spline curves are proper and powerful generalization of Bezier curve.
- They are very similar to Bezier curve and are also defined with the help of characteristic polygon.
- The major advantage of B-Spline curve is due to its ability to control the curve shapelocally, as opposed to global control (see Fig. 2.20).

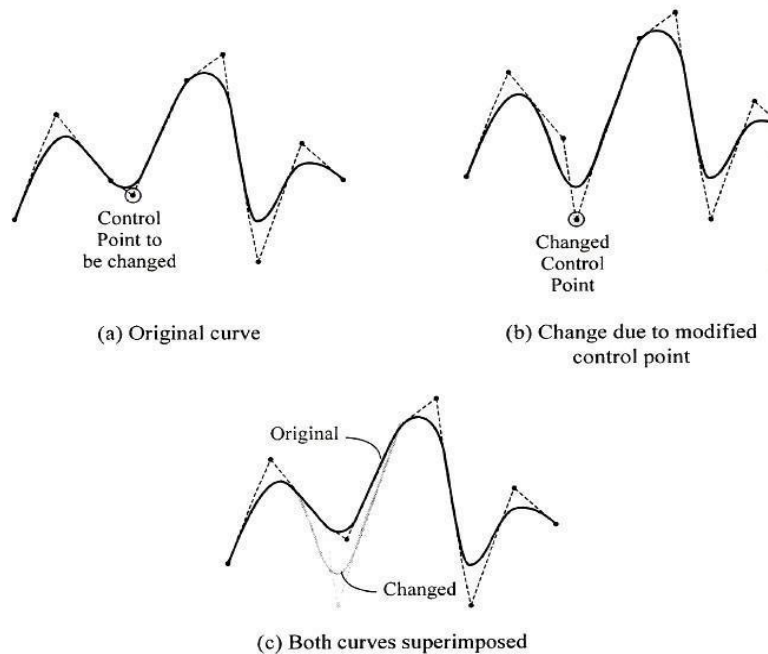


Fig. 2.20 Local Control of a B-Spline Curve

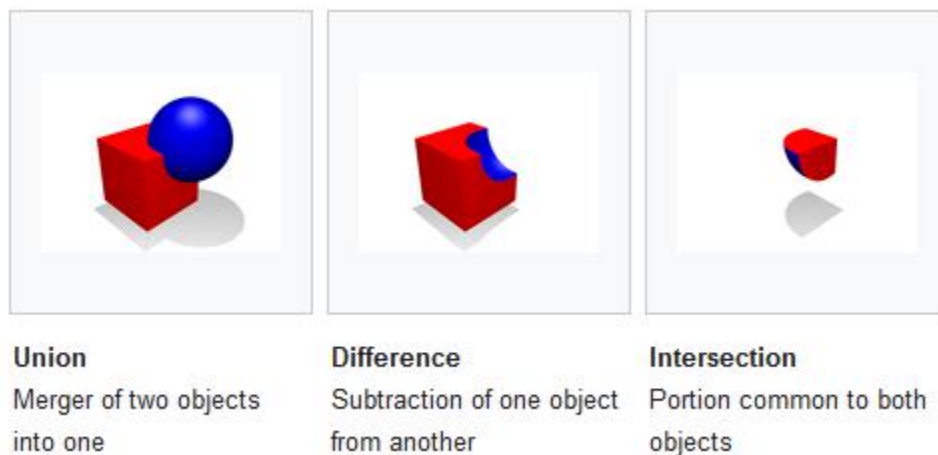
- B-Spline curves allow a local control over the shape of the spline curve.
- B-Spline curves allow us to vary the control points without changing the degree of the polynomial (i.e. number of control points can be added or subtracted).
- The degree of the curve is independent of the number of control points (i.e. four control points can generate a linear, quadratic or cubic curve).
- It gives better control over the shape of curve.
- They are more complex compared to Bezier curves.
- B-spline curve defined by  $n + 1$  control points  $P_i$  is given by

$$P(u) = \sum_{i=0}^n P_i N_{i,k}(u), \quad 0 \leq u \leq u_{\max} \tag{2.28}$$

## Surfaces and Solid- Constructive Solid Geometry, Boundary representation and Cellular Decomposition

### Constructive Solid Geometry

Constructive solid geometry has a number of practical uses. It is used in cases where simple geometric objects are desired, or where mathematical accuracy is important. Nearly all engineering CAD packages use CSG (where it may be useful for representing tool cuts, and features where parts must fit together). **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 simpler objects, potentially generating visually complex objects by combining a few primitive ones. In 3D computer graphics and CAD, CSG is often used in procedural modeling. CSG can also be performed on polygonal meshes, and may or may not be procedural and/or parametric. Contrast CSG with polygon mesh modeling and box modeling. The simplest solid objects used for the representation are called geometric primitives. Typically they are the objects of simple shape: cuboids, cylinders, prisms, pyramids, spheres, cones. The set of allowable primitives is limited by each software package. Some software packages allow CSG on curved objects while other packages do not. An 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 transformations of those sets. A primitive can typically be described by a procedure which accepts some number of parameters; for example, a sphere may be described by the coordinates of its center point, along with a radius value. These primitives can be combined into compound objects using operations like these:



Combining these elementary operations, it is possible to build up objects with high complexity starting from simple ones.

### Ray tracing

Rendering of constructive solid geometry is particularly simple when ray tracing. Ray tracers intersect a ray with both primitives that are being operated on, apply the operator to the intersection intervals along the 1D ray, and then take the point closest to the camera along the ray as being the result.

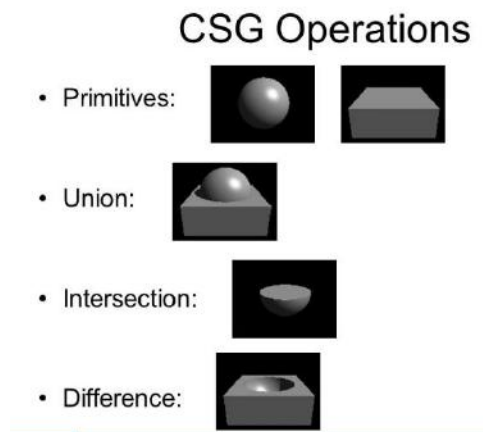
### Applications:

Constructive solid geometry has a number of practical uses. It is used in cases where simple geometric objects are desired, or where mathematical accuracy is important. Nearly all engineering CAD packages use CSG (where it may be useful for representing tool cuts, and features where parts must fit together).

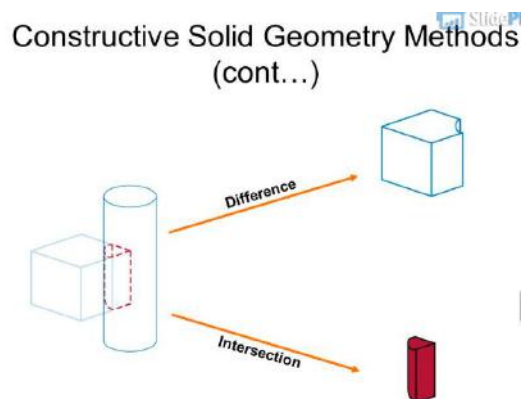
The Quake engine and Unreal Engine both use this system, as does Hammer (the native Source engine level editor), and Torque Game Engine/Torque Game Engine Advanced. CSG is popular because a modeler can use a set of relatively simple objects to create very complicated geometry. When CSG is procedural or parametric, the user can revise their complex geometry by changing the position of objects or by changing the Boolean operation used to combine those objects.

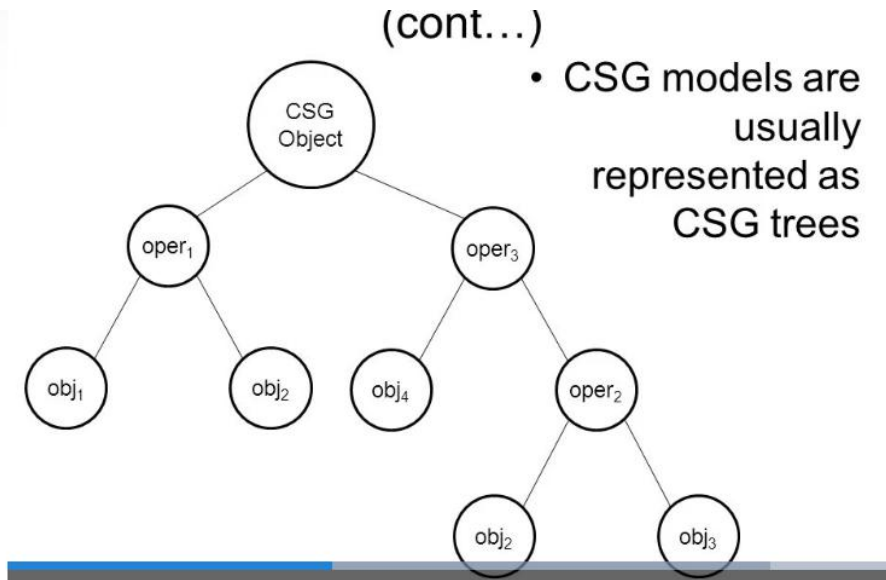
One of the advantages of CSG is that it can easily assure that objects are "solid" or water-tight if all of the primitive shapes are water-tight. This can be important for some manufacturing or engineering computation applications. By comparison, when creating geometry based upon boundary representations, additional topological data is required, or consistency checks must be performed to assure that the given boundary description specifies a valid solid object.

A convenient property of CSG shapes is that it is easy to classify arbitrary points as being either inside or outside the shape created by CSG. The point is simply classified against all the underlying primitives and the resulting boolean expression is evaluated. This is a desirable quality for some applications such as ray tracing.



Constructive solid geometry usually starts with a small set of primitives such as blocks, pyramids, spheres and cones. Two objects are initially created and combined using some Boolean operation to create a new object. This object can then be combined with another primitive to make another new object. This process is continuous until modeling is complete.

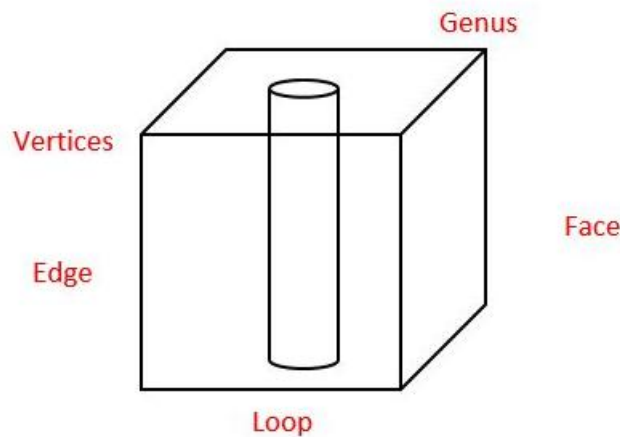




### Boundary representation

In solid modeling and computer-aided design, **boundary representation** (often abbreviated **B-rep** or **BREP**) is a method for representing a 3D shape by defining the limits of its volume. A solid is represented as a collection of connected surface elements, which define the boundary between interior and exterior points.

A boundary representation of a model comprises topological components (faces, edges and vertices) and the connections between them, along with geometric definitions for those components (surfaces, curves and points, respectively). A face is a bounded portion of a surface; an edge is a bounded piece of a curve and a vertex lies at a point. Other elements are the shell (a set of connected faces), the loop (a circuit of edges bounding a face) and loop-edge links (also known as winged edge links or half-edges) which are used to create the edge circuits. In solid modeling and computer-aided design, boundary representation (often abbreviated B-rep or BREP) is **a method for representing a 3D shape by defining the limits of its volume**. A solid is represented as a collection of connected surface elements, which define the boundary between interior and exterior points.



B-Rep or boundary representation is a rendering technique for solid modeling .It is popular approach to create a model of a physical object .Brep is that a three

dimensional object model is enclosed by surfaces or faces and has its own interior and exterior .It describes the shape as a collection of surfaces which separate its interior from exterior environment .it is suitable for complex designs ,polygon facets are one of the examples of boundary representation.

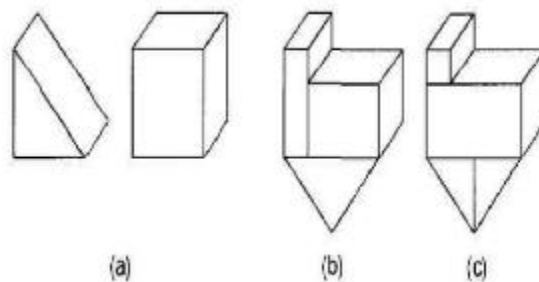
These are the following primitives of B-Rep

- 1.vertices : it is a point where two or more edges meet with another
- 2.edges : it is a line or curve enclosed between two vertices
3. faces : it is a surface or plane of a solid
- 4.loop:It is a hole in a face
- 5.genus: it is through a hole in a solid

#### **CELLULAR DECOMPOSITION:**

Cell decomposition makes use of bottom up construction in which complex shapes are created from simple primitives by sticking them together but the cells must not intersect while sharing single point, edge or face.

In cellular decomposition, an object may be represented as the addition of cells into which it may be decomposed. Each cell decomposition system described a set of primitive cells that are typically parameterized and are frequently curved. Cell decomposition differs from primitive instancing in that we may compose more complicated objects from simple, primitive ones as a bottom up fashion by gluing them together. The glue operation may be thought of as a limited form union in which the objects should not intersect. Further limitation on gluing cells frequently require that two cells share a single point, edge or face. Although cell –decomposition representation of an object is unambiguous .Cell decompositions are also hard to validate, as each pair of cells should potentially be tested for intersection. cell decomposition is significant for use in finite element analysis.



# LECTURE NOTES ON INTRODUCTION TO CAD/CAM 2022 – 2023

**III B. Tech I Semester**  
Mr. K.RAJASEKHAR, Assistant Professor



**ANAMACHARYA INSTITUTE OF TECHNOLOGY AND  
SCIENCES, TIRUPATI**  
(AUTONOMOUS)

Venkatapuram village, Renigunta mandal, Tirupati – 517 506

**Department of Mechanical Engineering**



**ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES, TIRUPATI  
(AUTONOMOUS)**

Year: III

Semester : I

Branch of Study : ME

| Subject Code | Subject Name            | L | T | P | Credits |
|--------------|-------------------------|---|---|---|---------|
| 20APE0302    | Introduction to CAD/CAM | 3 | 0 | 0 | 3       |

**Course Outcomes:**

- CO: 1 Understand the basic concepts components of CAD/CAM concepts of Graphics Techniques.
- CO: 2 Understand the concepts of Geometric representation methods.
- CO: 3 Understand and apply Numerical CNC Part Programming methods.
- CO: 4 Understand the concepts of Group technology and techniques, production flow Analysis.
- CO: 5 Understand the concepts of FMS and its elements.

**UNIT I**

Introduction: Definition and scope of CAD/CAM- Computers in industrial manufacturing, design process Computer Aided Design (CAD)-Computer Aided Manufacturing (CAM)- Computer Integrated Manufacturing (CIM)  
Graphics: Data base for graphic modeling-transformation geometry-3D transformations – Clipping-hidden line removal-Colour-shading.

**UNIT II**

Geometric modelling Parametric representation of curves, solids & surfaces. Geometric construction methods-Constraint based modeling- Wireframe, Surface- Bezier , B-Spline Surfaces and Solid- Constructive Solid Geometry, Boundary representation and Cellular Decomposition.

**UNIT III**

NC Control production systems: Introduction to NC, CNC, DNC - Manual part Programming – Computer Assisted Part Programming – Examples using NC codes- Adaptive Control – Canned cycles and subroutines – CAD/ CAM approach to NC part programming – APT language

**UNIT IV**

Role of information systems in manufacturing Discrete part manufacture-information requirements of a production organization-manufacturing strategies Integration requirement – Group Technology-coding-Production flow analysis-computer part programming CAPP implementation techniques.

**UNIT V**

Automated manufacturing systems Flexible Manufacturing systems (FMS) – the FMS concepts – transfer systems – head changing FMS – Introduction to Rapid prototyping, Knowledge Based Engineering, Virtual Reality, Augmented Reality – automated guided vehicle-Robots-automated storage and retrieval systems - computer aided quality controlCMM-Non contact inspection methods.



**Text Books:**

1. P.N.Rao, CAD/CAM: Principles & Applications-3rd Edition, Tata McGraw Hill.
2. CAD/CAM Concepts & applications/Alavala/PHI

**Reference Books:**

1. CAD/CAM Theory and Practice / IbrahimZeid / TMH.
2. CAD/CAM/CIM Radha Krishnan & Subramanian / New age
3. Principles of computer Aided Design and Manufacturing / Fanlc / Amirouche / Pearson.
4. Computer Numerical Control Concepts and Programming / Warrens & Seames / Thomson

| <b>List of COs</b> | <b>PO no. and keyword</b>            | <b>Competency</b> | <b>Performance Indicator</b> |
|--------------------|--------------------------------------|-------------------|------------------------------|
| CO: 1              | PO 1: Engineering knowledge          | 1.6               | 1.3.1                        |
| CO: 2              | PO 5: Modern tool usage              | 5.2               | 5.2.2                        |
| CO: 3              | PO 2: Problem analysis               | 2.5               | 2.1.2                        |
| CO: 4              | PO 5: Modern tool usage:             | 5.2               | 5.2.2                        |
| CO: 5              | PO 7: Environment and sustainability | 7.4               | 7.4.1                        |



## ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES(AUTONOMOUS)

Karakambadi Road, Venkatapuram (Vi), Tirupati

### COURSE OUTCOMES & COURSE OBJECTIVES

|                                 |   |
|---------------------------------|---|
| Name of the Faculty             | : K. Rajasekhar                           |
| Subject                         | : Introduction to CAD/CAM                 |
| Subject Code                    | : <b>20APE0302</b>                        |
| Class & Branch / Specialization | : III B.TECH I SEM Mechanical Engineering |

| S.No | Course Objectives   |
|------|---|
| 1.   | Understand the basic concepts components of CAD/CAM concepts of Graphics Techniques.  |
| 2.   | Understand the concepts of Geometric representation methods.                          |
| 3.   | Understand and apply Numerical CNC Part Programming methods.                          |
| 4.   | Understand the concepts of Group technology and techniques, production flow Analysis. |
| 5.   | Understand the concepts of FMS and its elements.                                      |

Signature of Faculty

Date :

Signature of HOD

Date :

Signature of Principal

Date :

NC Control production systems: Introduction to NC, CNC, DNC - Manual part Programming – Computer Assisted Part Programming – Examples using NC codes- Adaptive Control – Canned cycles and subroutines – CAD/CAM approach to NC part programming – APT language

### **UNIT -3**

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

#### **NC MODES:**

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

#### **NC ELEMENTS:**

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

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

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

#### **Program of instructions:**

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

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

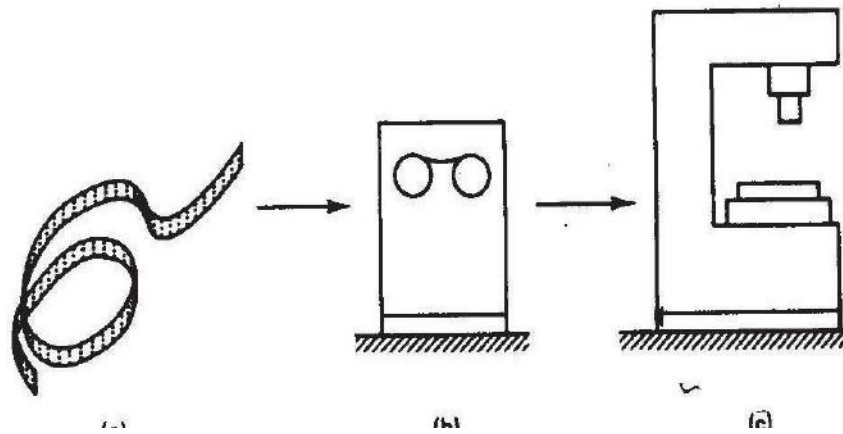


Figure three basic components of a numerical control system:

- a) Program of instructions
- b) Controller unit

c) Machine tool of a direct link with a computer. This is called direct numerical control or DNC,. The program of instructions is prepared by some one called a part programmer. The programmer's job is to provide a set of detailed instructions by which the sequence of processing steps is to be performed. For a machining operation, the processing steps involve the relative movement between the cutting tool and the workpiece.

### **Controller unit:**

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

The tape reader is an electromechanical device for winding and reading the punched tape containing the program of instructions. The data contained on the tape are read in to the data buffer. The purpose of this device is to store the input instructions in logical blocks of information. A block of information usually represents one complete step in the sequence of processing elements.

For example, one block may be the data required to move the machine table to a certain position and drill a hole at that location.

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

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

Another element NC system, which may be physically part of the controller unit or part of the machine tool, is the control panel. The control panel or control console contains the dials and switches by which the machine operator runs the NC system. It may also contain data displays to provide information the operator.

Although NC system is an automatic system, the human operator is still needed to turn the machine on an off, to change tools (some NC systems automatic tool changers), to load and unload the machine and to perform various other duties. To be able to discharge these duties, the operator must be able to control the system, and this is done through the control panel.

#### **NUMERICAL CONTROL ELEMENTS:**

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

NC machines range in complexity from simple tape- controlled drill presses to highly sophisticated and versatile machining centers. The NC machining center was first

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

### **NC MACHINE TOOL AND THEIR STRUCTURE:**

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

**1. process planning:** The engineering drawing of the work part must be interpreted in terms of the manufacturing processes to be used. This step is Referred to as process planning and it is concerned with the preparation of a route sheet. The route sheet is a listing of the sequence of operations which must be performed on the work part. It is called a route sheet

Because it also lists the machines through which the part must be routed in order to accomplish the sequence of operations. We assume that some of the operations will be performed on one or more NC machines.

**2. part programming:** A part programmer plans the process for the portions of the job to be accomplished by NC. Part programmers are knowledgeable about the

machining process and they have been trained to program for numerical control. They are responsible for planning the sequence of machining steps to be performed by NC and to document these in a special format. There are two ways to program for NC:

### **MANUAL PART PROGRAMMING, COMPUTER ASSISTED PART PROGRAMMING**

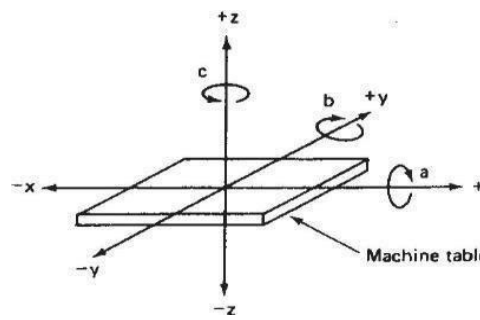
In manual part programming, the machining instructions are prepared on a form called a part program manuscript. The manuscript is a listing of the relative cutter / workpiece positions which must be followed to machine the part.

In computer assisted part programming much of the tedious computational work required in manual part programming is transferred to the computer. This is especially appropriate for complex workpiece geometries and jobs with many machining steps. Use of the computer in these situations results in significant savings in part programming time.

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

**4. Tape verification:** After the punched tape has been prepared, a method is usually provided for checking the accuracy of the tape. Sometimes the tape is checked by running it through a computer program which plots the various tool movements (or table movements) on paper. In this way, major errors in the tape can be discovered. The "acid test" of the tape involves trying it out on the machine tool to make the part. A foam or plastic material is sometimes used for this tryout. Programming errors

are not uncommon, and it may require about three attempts before the tape is correct and ready to use in production. The final step in the NC procedure to use the NC tape in production. This involves ordering the raw



work parts specifying

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

### NC COORDINATE SYSTEMS:

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

Two axes, x and y, are defined in the plane of the table, as shown in figure. The z axis is perpendicular to this plane and movement in the z direction is controlled by the vertical motion of the spindle. The positive and negative directions of motion of tool relative to table along these axes as shown in figure. A NC drill presses are classified as either two-axis or three-axis machines, depending on whether or not they have the capability to control the z-axis. A numerical control milling machine and similar machine tools (boring mill for example) use an axis system similar to that of the drill press. However, in addition to the three linear axes, these machines may possess



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

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

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

### **Manual Part Programming**

In manual programming, the part programmer specifies the machining instructions on a form called a manuscript. Manuscripts come in various forms, depending on the machine tool and tape format to be used. For example, the manuscript form for a two-axis point-to-point drilling machine would be different than one for a three-axis contouring machine. As mentioned, the manuscript is a listing of the relative tool and workpiece locations. It also includes other data, such as preparatory commands, miscellaneous instructions, and speed/feed specifications, all of which are needed to operate the machine under tape control. The manuscript

is designed so that the NC tape can be typed directly from it on a Flexo writer or similar tape- punch device. We shall divide manual programming jobs into two categories : point-to-point jobs and contouring jobs. Except for

complex work parts with many holes to be drilled, manual programming is ideally suited for point-to-point applications. On the other hand, except for the simplest milling and turning jobs, manual programming can become quite time-consuming for applications requiring continuous-path control of the tool. Accordingly, we shall only concern ourselves with manual part programming for point-to-point operations in this chapter. Manual contour programming requires such tedious and detailed calculations that the space needed for the topic would be more than is warranted by the basic purpose of this book, which is to survey the field of automated manufacturing systems.

## **COMPUTER ASSISTED PART PROGRAMMING**

Computer Assisted Part Programming was relatively simple, it was a suitable application for manual programming. Most parts machined on NC systems are considerably more complex. In the more complicated point-to-point jobs and in contouring applications, manual part programming becomes an extremely tedious task and subject to error. In these instances it is much more appropriate to employ the high-speed digital computer to assist in the part programming process. Many part programming language systems have been developed to automatically perform most of the calculations which the programmer would otherwise be forced to do. This saves time and results in a more accurate and more efficient part program.

### **The part programmer's job**

The difference in the part programmer's job between manual programming and computer-assisted programming is this. With manual programming, a manuscript is used which is formatted so that the NC tape can be typed directly from it. With computer-assisted part programming, the machining instructions are written in English-like statements of the NC programming language, which are then processed by the computer to prepare the tape. The computer automatically punches the tape in the proper tape format for the particular NC machine.

When utilizing one of the NC programming languages, part programming can be summarized as consisting basically of two tasks :

1. Defining the geometry of the work part
2. Specifying the tool path and/or operation sequence

Let us now consider these two tasks in computer-assisted part programming. Our frame of reference will be for a contouring application, but the concepts apply for a positioning application as well.

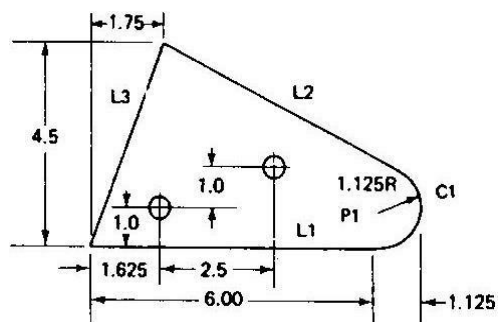


Fig: 1 Basic Geomentry Element

### WORK PART GEOMETRY DEFINITION:

No matter how complicated the work part may appear, it is composed of basic geometric elements. Using a relatively simple work part to illustrate, consider the component shown in Figure 1. Although somewhat irregular in overall appearance, the outline of the part consists of intersecting straight lines and a partial circle. The holes in the part can be expressed in terms of the center location and radius of the hole. Nearly any component that can be conceived by a designer can be described by points, straight lines, planes, circles, cylinders, and other mathematically defined surfaces. It is the part programmer's task to enumerate the component elements out of which the work part is formed. Each geometric element must be identified and the dimensions and location of the element explicitly defined. Using the APT programming language as an example, the following statement might be used to define a point :

P1 – POINT/6.0, 1.125,0

The point is identified by the symbol P1 and is located at  $x = 6.0$ ,  $y = 1.125$ , and  $z = 0$ . Similarly, a circle in the  $x$ - $y$  plane might be defined by the APT statement

C1 = CIRCLE/CENTER, P1 , RADIUS, 1.12

The center of circle C1 is Pi (previously defined) and the radius is 1.125.

The various geometric elements in the drawing of Figure 9.7 would be identified in a similar fashion by the part programmer.

## **TOOL PATH CONSTRUCTION**

After defining the work part geometry, the programmer must next construct the path that the cutter will follow to machine the part. This tool path specification involves a detailed step-by-step sequence of cutter moves. The moves are made along the geometry elements which have previously been defined. To illustrate, using Figure 1 and the APT language, the following statement could be used to command the tool to make a left turn from line L2 onto line L3 : GOLFT/L3, PAST, L1

This assumes the tool was previously located at the intersection of lines L2 and L3 and had just finished a cut along L2. The statement directs the tool to cut along L3 until it just passes line L1. By using statements similar to the above, the tool can be directed to machine along the work part surfaces, to go to point locations, to drill holes at those point locations, and so on. In addition to geometry definition and tool path specification, the part programmer also provides other commands to the NC system. However, let us await Section 9.5, where we will consider a wide range of possible APT statements.

### **The computer's job**

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

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

The sequence of these steps and their relationships to the part programmer and the machine tool are illustrated in Figure 2

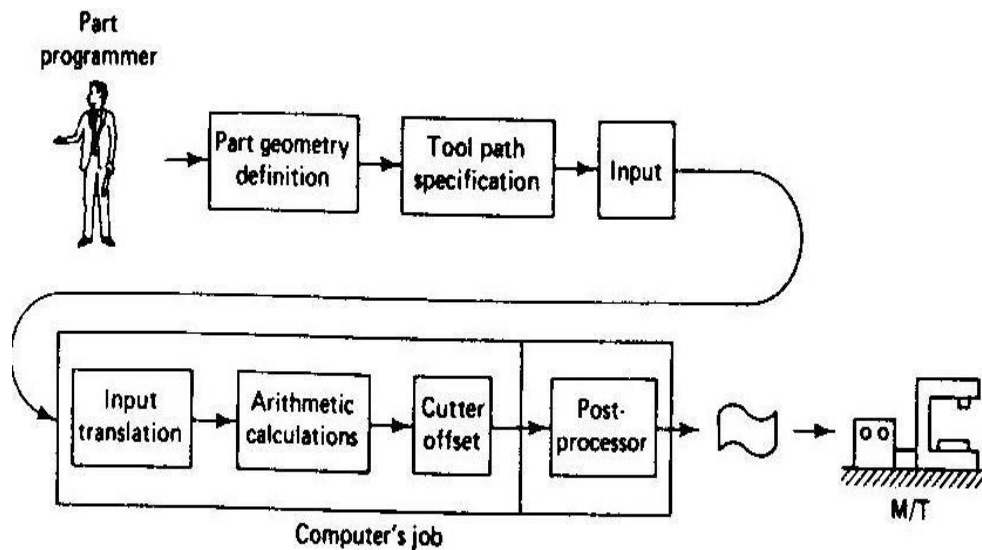


Fig.2 Steps in Computer Assisted Part Programming

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

**ARITHMETIC CALCULATIONS:** The arithmetic calculations unit of the system consists of a comprehensive set of subroutines for solving the mathematics required to generate the part surface. These subroutines are called by the various part programming language statements. The arithmetic unit is really the fundamental element in the part programming package. This unit frees the programmer from the time-consuming geometry and trigonometry calculations to concentrate on the work part processing.

**CUTTER OFFSET COMPUTATION:** When we described the second task of the part programmer as that of constructing the tool path, we ignored one basic factor the size of the cutting tool. The actual tool path is different from the part outline. It is because the tool path is the path taken by the center of the cutter. It is at the periphery of the cutter that machining takes place.

The purpose of the cutter offset computation is to offset the tool path from the desired part surface by the radius of the cutter. This means that the part programmer can define the exact part outline in his geometry statements. Thanks to the cutter offset calculation provided by the programming system, he need not concern himself with this task. The cutter offset problem is illustrated in Figure 3

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

The post processor is really a separate computer program that has been written to prepare the punched tape for a specific machine tool. The input to the post processor is the output from the other three components: a series of cutter locations and other instructions. This is referred to as the CLFILE or CLDATA (CL stands for cutter location). The output of the post processor is the NC tape written in the correct format for the machine on which it is to be used.

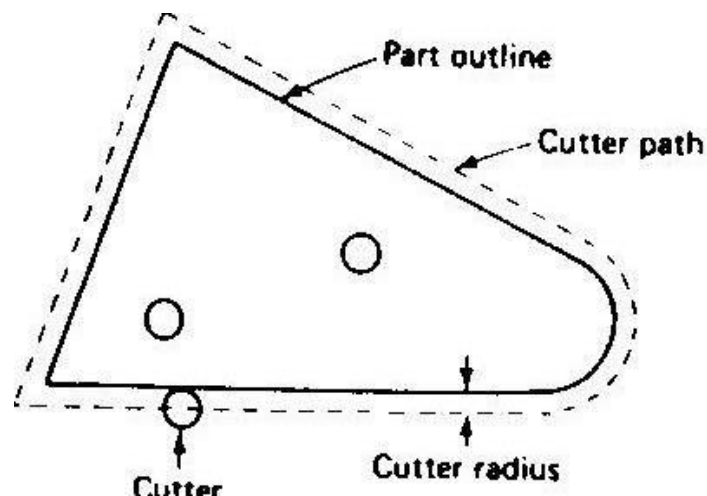


Fig.3 Cutter Offset Problem in computer-assisted part programming

## Structure of Part Program

A *part program* is simply an NC program used to manufacture a part. *Part programming* for NC maybe performed manually (*manual part programming*) or by the aid of a computer (*Computer-aided partprogramming*).

Many *programming languages* have been developed for part programming. The first that used Englishlike statements and one of the most popular languages is called APT (for *Automatically*

*Programmed Tools*). Many variations of APT have been developed, including ADAPT (ADaptation of APT), EXAPT (a European flavor of APT), UNIAPT (APT controller for smaller computer systems), etc.

NC programming for complex parts are generated using advanced computer programs (CAD/CAM programs), which create automatically the machine code (so called *G-code*) in a graphic environment. Machine code is also largely used for manual part programming of simple shapes and is covered in the present section.

## **Machine code**

The structure of a NC program written in machine code is standardized and for a two-axis NC system has the following format:

*NC program block* consists of a number of program words. The NC program is executed block by block: each next block is entered in the system and executed only after entirely completing the current block.

Each *program word* is an ordered set of characteristics, letters and numbers, to specify a single action of the machine tool. Program words fall into two categories,

• *modal*, which are active in the block in which they are specified and remain active in the subsequent blocks until another program word overrides them;

- *non-modal*, which are only active in the block in which they are specified.

Some of the most important program words are as follows

- *sequence numbers* (N\*\*\*\*)

Sequence numbers are a means of identifying program blocks. In some systems they are not required although sequence numbers are needed in most *canned cycles* (covered later in this section);

- *preparatory functions (also G-codes) (G\*\*)*

Preparatory functions are used to set up the mode in which the rest of the operation is to be executed.

Some of examples of G-codes are given in the table:

G00 Positioning (not cutting)  
 G01 Linear interpolation  
 G02 Clockwise circular interpolation  
 G03 Counterclockwise circular interpolation  
 G20 Inch data input  
 G21 Metric data input  
 G54 Workpart coordinate  
 preset G80 Canned cycle cancel  
 G81-89 Canned cycles  
 G90 Absolute programming  
 G91 Incremental programming

- *Dimension words (D\*\*\*\*.\*\*\*),* where D stands for X, Z, U, or W. Dimension words specify the coordinate positions of the programmed path. X and Z specify the *absolute coordinates*, and U and W specify the *incremental coordinates* (absolute and incremental programming are explained later in this section);
- *Arc center coordinates (D\*\*\*\*.\*\*\*),* where D stands for I, or K. Arc center coordinates specify the incremental coordinate position of the arc center (I in the direction of X-axis, and K in the direction of Z-axis), measured from the arc starting point;

- *Feed function (F\*\*.\*\*) (F\*\*.\*\*)*

Specifies the velocity of feed motion;

- *Spindle control function (S\*\*\*\*)*

Specifies spindle rotational speed in revolutions per minute, or cutting velocity in meter per minute depending on the type of NC system and machine tool;

- *Tool calls (T\*\*.\*\*) (T\*\*.\*\*)*

The tool call word is used to access the required tool. It also gives the information for the radial compensation of tool corner wear for each new run of the program (and each new part);

- *Miscellaneous functions (M\*\*)*

The M-function performs miscellaneous machine actions such as these listed in the table:

M00 Program stop  
 M02 Program end  
 M03 Start spindle CW  
 M04 Start spindle CCW



M05 Stop spindle  
M06 Execute tool change  
M07 Turn coolant on  
M25 Open chuck  
M26 Close chuck

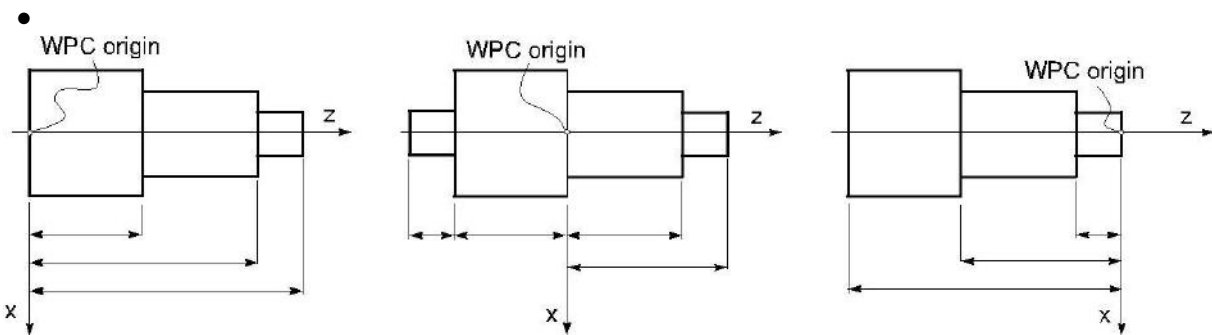
### ***Program points***

The NC system must know where the part is positioned in the work space. The procedure for defining the *work coordinate system (WPC)* is called *work piece coordinate setting*. Two important factors deal with work piece coordinate setting,

☒ where the *part datum* (the origin of the WPC) is situated with respect to the workpiece;

- Where the part datum is situated with respect to the machine tool.
- 

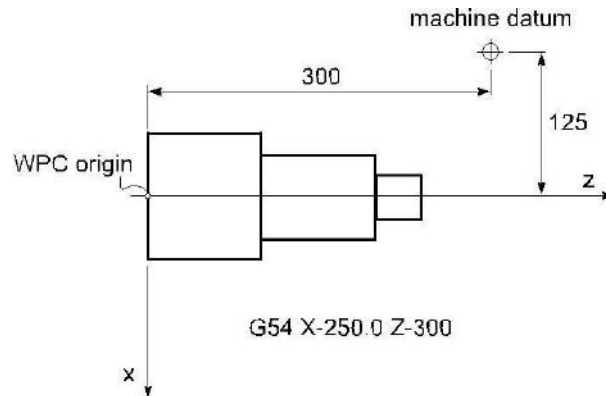
The WPC origin may be located at any part of the work piece, but to avoid dimensional recalculations and respectively errors, the good programmers will chose the WPC origin at the point, from where the part features are dimensioned:



**Fig:4 Selection of Work Piece Coordinate origin**

The methods for locating the positions of the WPC origin with respect to the machine tool varies for each machine tool. Some systems use a *zero-set button* to set the WPC origin. On other types of NC systems, the WPC is set with a G54 or a G92 code followed by X, and Z dimensions.

The G54 code tells the machine where the position of the WPC measured from the machine zero point is. *Machine zero point (machine datum)* is a fixed point on the machine tool and cannot be programmed or altered.



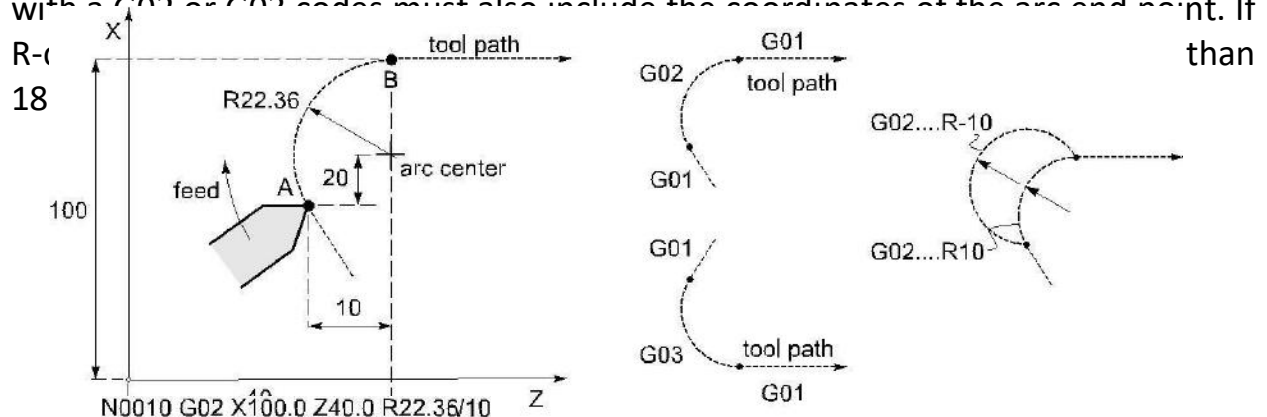
**Fig:5 Setting of WPC**

Another important point is the *program start point* (also *tool home position*). This point is selected by the programmer at some distance from the work piece, not too far to save some time when the tool returns home, and not too close to allow for safe indexing of the tool turret when the cutting tool is changed. The program, therefore the new part machining, starts and ends with the tool at home position, but the tool needs also to be returned to home whenever a tool change take place during the program execution.

**Linear and circular interpolation**

A G01 *linear interpolation* code moves the tool to a position with coordinates defined with program words in a straight, including angular line at the specified with F-code feed rate. The command is modal and is active until either a G00, or G02, or G03 overrides it.

NC system are capable of commanding a circular motion. Arc movement is known as *circular interpolation* and is carried out with a G02 (*clockwise circular interpolation*) or G03 (*counter clockwise circular interpolation*) codes. The arc radius is specified either by the incremental dimensional words I and K, which defines the position of arc centerpoint with respect to the arc start point, or directly by the radius R-code. In both methods, the program block, which starts with a G02 or G03 codes must also include the coordinates of the arc end point. If

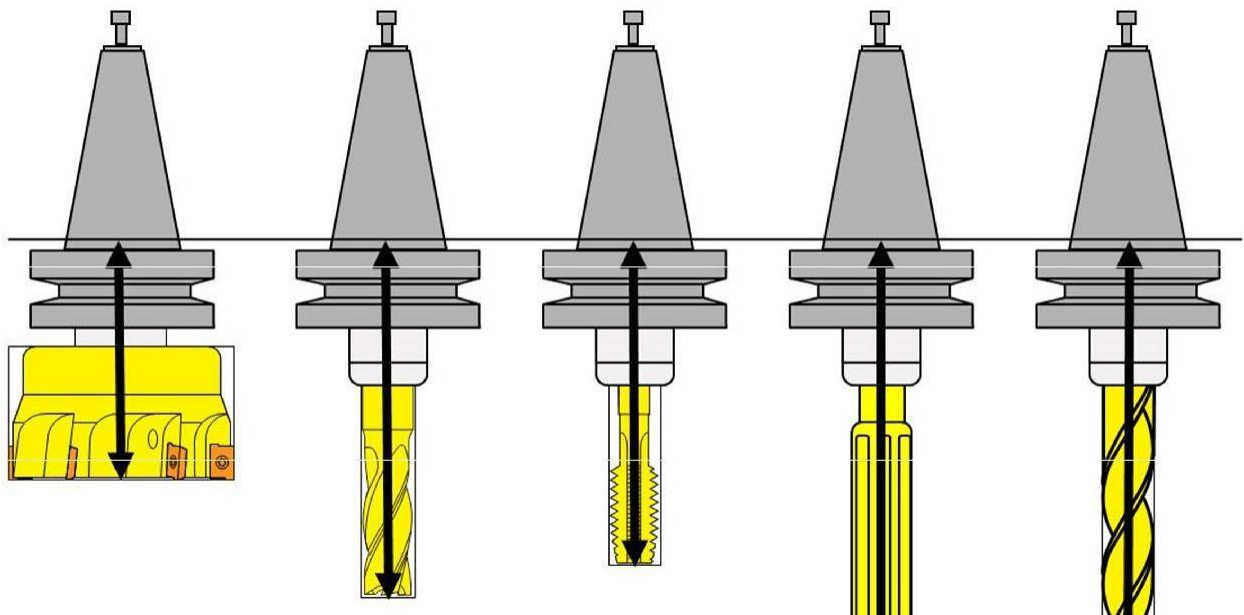


**Fig:6 Linear and circular interpolation.**

### **TOOL LENGTH COMPENSATION**

**Tool length compensation** is instated with a G43 word. Included within the G43 command is an H-word that specifies the **offset** number in which the **tool length compensation** value is stored. You must also include a Z-word in the G43 command, telling the machine where you want the **tool tip** to be positioned.

Cutting tools used on machining centers differ from one another. For one thing, there are a variety of cutting tool types that are used on machining centers, including center drills, spotdrills, drills, taps, reamers, boring bars, end mills, and face mills (among many others). Each type of tool requires a different way of gripping the actual cutting tool in its holder. Some tools (like some straight shank tools) use a collet system. Others (like end mills) use a set- screw to hold the cutting tool in place. Yet others (like face mills and taps) require a very special style of tool Holder—designed especially for the cutting tool.



**Fig:7 Tool Length compensation**

Tool length compensation will allow you to write programs even though you don't know how long the cutting tools will be at production time.

### **Tool's Length Will Vary from Each Time it is Assembled**

When a cutting tool is assembled more than once (even with the same components), its length will usually vary. Consider, for examples, straight shank

tools that are placed in collet holders. Each time you assemble the tool, it will be of a different length. Tool length compensation will allow you to use the same program over and over again, even though each tool's length changes from one time the job is run to the next.

### **Tool Data is Entered Separately from the Program**

The same program will work regardless of how long each cutting tool is. The program tells the control where to look for the length of each tool. During setup, the setup person (or someone) assembles and measures each cutting tool. The length of each tool is then placed in the appropriate location (a tool offset register).

### **Sizing and Trial Machining Must Often be Done**

The importance of being able to trial machine in order to machine the first work piece correctly. And during a given tool's life a tool will wear and cause the surface being machined to change. Tool length compensation allows the setup person and operator to easily hold size for Z-axis related dimensions (pocket depths, hole-depths, etc.). The program need not be changed when work piece dimensions must be adjusted.

### **CUTTER RADIUS COMPENSATION**

**Cutter Radius Compensation** also known as CRC, is a function of the CNC controller to automatically shift the tool from the **cutter** center line to the **cutter** edge along the programmed **cutter** path. If it were not for CRC the **cutter** path locations would have to be calculated to the **cutter** center line. Cutter Radius Compensation (CRC) can be used to program continuous path milling operations.

There are a few special rules that must be applied to the programming method when using CRC.

- 1) The point before the CRC approach and exit from the contour cut must be equal to or greater than the cutter radius.
- 2) CRC can only be started and stopped along linear motion.
- 3) CRC can only be applied to 2 axis of motion at a time.
- 4) The use of more than one non motion block in a row is not allowed.

### **Three main advantages to using Cutter Radius Compensation (CRC) programming**

- 1) The programmer can use point values that are directly along the cutter path rather than having to calculate the tool center.

2) During production runs, cutters of different sizes can be used such as undersize tools when the original size tool is no longer available. It is a simple radius offset value change in the machine control to achieve this. The programming will remain the same.

3) A program section or operation can be used over and over with different tools to create roughing passes and finish passes. This is done by changing the radius offset value and using multiple offsets.

## **TOOL NOSE RADIUS COMPENSATION**

During processing of the work piece contour, due to the tool radius, tip and a half Diameter of the existence of the tool tip and work piece center or imaginary contours do not coincide. If the digital system does not have automatic tool radius compensation function can only be converted into contour programming tool center path, and then processed. When the tool wear, heavy wear, tool change, to re-calculate the tool center path, modify the program. However, when the numerical control system with automatic tool radius compensation function, you can just press the work piece contour programming,

CNC system will automatically calculate the tool center path, deviate from the contour of the tool with a radius value, even if the tool wear, heavy wear, tool change, also need to modify the radius of the tool deviate from the contour value that offsets, tool radius compensation automatically, without modifying the program.

CNC lathe programming is based on imaginary tip movement; the actual tip location is a small circular, conical surface and in the turning arc will produce processing errors. If the use of nose radius compensation, pre-tip to tip radius and arc parameters such as input to the tool position within the database, according to work piece contour programming, numerical control system automatically calculates the tool center trajectory, the control center path for cutting knives processing, eliminating the tip of the arc caused by processing errors. You can also modify the value to eliminate abrasion tool wear or tool failure caused by processing errors.

Similarly, in CNC milling, the tool radius offset pre-register in the specified register, the use of the tool radius compensation commands, by adjusting the tool radius compensation value to compensate for tool wear, in order to eliminate the tool wear caused by processing errors. At the same time even if a replacement tool or tool re-grinding, as long as the contour is the same,

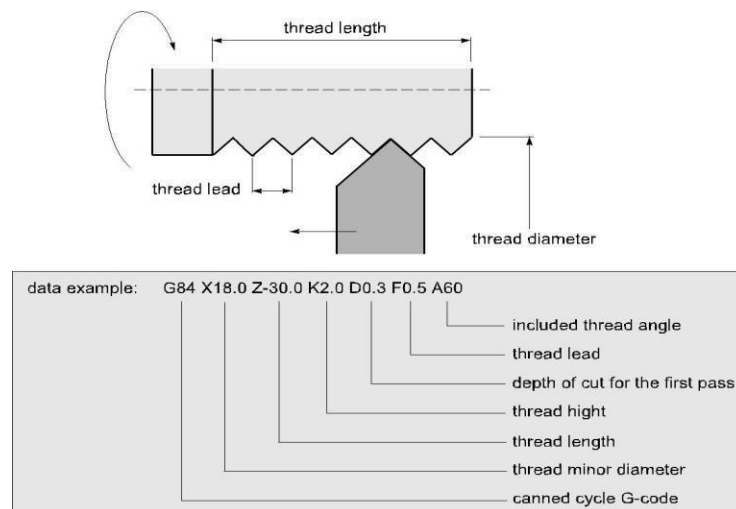
processing the same coordinate system, you can use the original program. At the same time as the application can also adjust the tool radius compensation amount to the same profile using the same procedures under the same conditions, rough and finish.

## Cycles

The repetitive program (and machining) sequence is called a cycle. Cycles are classified into two principle groups,

☒ canned cycles (also fixed cycles), and user-defined cycles (sub-routines).

Canned cycles are an inbuilt feature of the NC system. The usage of canned cycles makes easier programming for threading, drilling holes and other repetitive machining tasks. The next figure illustrates a thread cutting canned cycle:



**Fig:7 Threaded canned cycle**

*User sub-routines* are useful, when the necessary canned cycle is not available.

The user sub-routine is a NC program, which describes a sequence of operations, which is often repeated when machining particular part. The sub-routine is called from the main NC program with a M98 command.

A special type of user-defined cycles is so-called *macros*, which are generic cycles with parametric variables. The macro is called from the main program with a set of numerical values for these variables. This allow to use one and the same macro to machine different in size, but similar in shape components. Programming with macros is often referred to as a parametric programming.

## Mirror Image

The mirroring command is used when features of components shares symmetry about one or more axes and are also dimensionally identical. By using this code components can be machined using a single set of data and length of programs can be reduced.

- G10 cancellation of mirroring
- G11 Mirror image on X axis
- G12 Mirror image on Y axis
- G13 Mirror image on Z axis

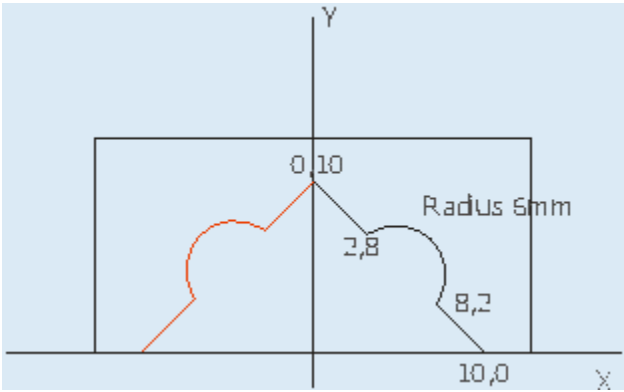


Fig. 8 Illustrative Example for mirroring

|                        |   |
|------------------------|---|
| 00002                  | Program number                                      |
| N02 G21                | Metric programming                                  |
| N03 M03 S1000          | Spindle start clockwise with 1000rpm                |
| N04 G00 X0.0 Y0.0      | Rapid motion towards (0,0)                          |
| N05 G00 Z10.0          | Rapid motion towards Z=10 plane                     |
| N06 G00 X10.0          | Machining positive X side                           |
| N07 G00 Z0.0           |   |
| N08 G01 X8.0 Y2.0      |   |
| N09 G03 X2.0 Y8.0 R6.0 |   |
| N10 G01 X0.0 Y10.0     |   |
| N11 G00 Z10.0          |   |
| N12 G00 X0.0 Y0.0      |   |
| N13 G11                | Mirror image on X axis                              |
| N14 G25 N06.12         | Repeat lines 6 to 12 (machining of negative X side) |
| N15 G10                | Cancellation of mirror image                        |
| N16 M05 M09            | Spindle stop and program end                        |

## Parametric Programming

Parametric programming can be compared to any computer programming language like BASIC, C Language, and PASCAL. However, this programming

language resides right in the CNC control and can be accessed at G code level, meaning you can combine manual programming techniques with parametric programming techniques. Computer-related features like variables, arithmetic, logic statements, and looping are available. Like computer programming languages, parametric programming comes in several versions. The most popular is Custom Macro B (used by Fanuc and Fanuc-compatible controls). Others include User Task (from Okuma), Q Routine (from Sodick), and Advanced Programming Language [APL] (from G& L)

In addition to having many computer-related features, most versions of parametric programming have extensive CNC-related features. Custom macro, for example, allows the CNC user to access many things about the CNC control (tool offsets, axis position, alarms, generate G codes, and program protection) right from within a CNC program. These things are impossible with only normal G code programming techniques.

### **Applications:**

Many companies have excellent applications for custom macro and don't even know it. Of course, if you don't even know you have an application for something, it's impossible to even consider using it. While these applications are covered in much greater detail during our video course and CD-rom course, applications for custom macro fall into five basic categories.

### **Example:**

To stress what can be done with parametric programming, we show a simple example written in custom macro B for a machining center application. It will machine a mill a hole of any size at any location. Notice how similar this program is to a program written in BASIC.

#### Program

```
O0001 (Program number)
#100=1. (Diameter of end
mill)#101=3.0 (X position
of hole) #102=1.5 (Y
position of hole)
#103=.5 (Depth of counter
bored hole)#104=400 (Speed in
RPM)
#105=3.5 (Feed rate in IPM)
#106=3. (Tool length offset
number)
#107=2.0 (Diameter of counter bored hole)
```



```
G90 G54 S#104 M03 (Select abs mode, coordinate system, start
spindle)G00 X#101 Y#102 (Rapid to hole center)
G43 H#106 Z.1 (Instate tool length compensation, rapid to
approach Z position)G01 Z-#103 F[#105 / 2]
Y[#102 + #107 / 2 - #100 / 2] F#105
G02 J-[#107 / 2 - #100
/ 2]G01 Y#102
G00
Z.1
M30
```

### **Automatically Programmed Tool (APT)**

The API language was the product of the MIT developmental work on NC programmingsystems. Its development began in June 1956, and it was first used in production around 1959. Today it is the most widely used language in the United States. Although first intended as a contouring language, modern versions of APT can be used for both positioning and continuous- path programming and continuous-path programming in up to five axes.

AUTOSPOT (AUTOMATIC SYSTEM FOR POSITIONNING TOOLS). This was developed by IBM and first introduced in 1962 for PTP programming. Today's version of AUTOSPOT canbe used for contouring as well.

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

COMPACT II. This is a package available from Manufacturing Data Systems, Inc. (MDSI), a firm based in Ann Arbor, Michigan. The NC language is similar to SPLIT in many of its features. MDSI leases the COMPACT II system to its users on a time-sharing basis. The part programmer uses a remote terminal to feed the program into one of the MDSI computers, which in tum produces the NC tape.

ADAPT (ADAPTATION OF APT). Several part programming languages are based directly on the APT program. One of these is ADAPT, which was developed by IBM under Air Force contract. It was intended to provide many of the features of APT but to utilize a significantly smaller computer. ADAPT is not as powerful as APT, but can be used to program for both positioning and contouring jobs.

EXAPT (EXTENDED SUBSET OF APT). This was developed in Germany starting around 1964 and is based on the APT language. There are three versions: EXAPT I designed for positioning (drilling and also straight-cut milling), EXAPT II designed for turning, and EXAPT III designed for limited contouring operations. One of the important features of EXAPT is that it attempts to compute optimum feeds and speeds automatically.

## **PART PROGRAMMING USING CAD/CAM**

A CAD/CAM system is a computer interactive graphics system equipped with software to accomplish certain tasks in design and manufacturing functions. One of the important tasks performed on a CAD/CAM system is NC part programming. In this method of part programming, portions of the procedure usually done by the part programmer are instead done by the computer. The two main tasks of a part programmer in a computer assisted programming are (a) defining the part geometry and (b) specifying the tool path. The proposed methodology is used to automate both of these tasks.

### **Part geometry definition:**

The fundamental objective of CAD/CAM system is to integrate the design engineering and manufacturing engineering functions. Certainly one of the important design functions is to design the individual components of the product. If a CAD/CAM system is used, a computer graphics model of each part is developed by the designer and stored in the CAD/CAM database. That model contains all of the geometric, dimensional and material specifications for the part.

When the same CAD/CAM system, or a CAM system that has access to the same CAD database in which the part model resides, is used to perform NC part programming it makes little sense to re-create the geometry of the part during the programming procedure. Instead, the programmer has the capability to retrieve the part geometry model from the storage and to use that model to construct the appropriate cutter path. The significant advantage of using CAD/CAM in this way is that it eliminates one of the time consuming steps in computer assisted part programming geometry definition.

After the part geometry has been retrieved, the usual procedure is to label the geometric elements that will be used during part programming. These labels are the variable names (symbols) given to the lines, circles and surfaces that comprise the part. Most systems have the capacity to automatically label the geometry elements of the part and to display the labels on the display.

If the NC programmer does not have access to the data base, then the NC Programming

must be defined. This is done by using similar interactive graphics techniques that the product designer would use to design the part. Points are defined in a coordinate system using the computer graphics system, lines and circles are defined from the points, surfaces are defined, and so forth,

to construct a geometric model of the part. The advantage of using the interactive graphics system over conventional computer assisted part programming is that the programmer receives immediate visual verification of the definitions being created. This tends to improve the speed and accuracy of the geometry definition process.

### **Tool path generation using CAD/CAM:**

The second task of the NC programmer in computer assisted part programming is tool path Specification. The first step in specifying the tool path is to select the cutting tool for the operation. Most CAD/CAM systems have tool libraries that can be called by the programmer to identify what tools are available in the tool crib. The programmer must decide which of the available tools is most appropriate for the operation under consideration and specify it for the tool path. This permits the tool diameter and other dimensions to be entered automatically for tool offset calculations. If the desired cutting tool is not available in the library, an appropriate tool can be specified by the programmer. It then becomes part of the library for future use.

The next step is tool path definition. There are differences in capabilities of the various CAD/CAM systems, which result in different approaches for generating the tool path. The most basic approach involves the use of the interactive graphics system to enter the motion

Commands one by one, similar to computer assisted part programming. Individual statements in APT or other part programming language are entered and the CAD/CAM system provides an immediate graphic display of the action resulting from the command, thereby validating the statement. A more advanced approach for generating tool path commands is to use one of the automatic software modules available on the CAD/CAM system. These modules have been developed to accomplish a number of common machining cycles for milling, drilling and turning. They are subroutines in the NC programming package that can be called and the Required parameters given to execute the machining cycle.

### **Computer Automated part programming:**

In the CAD/CAM approach to NC part programming, several aspects of the procedure are automated. In the future, it should be possible to automate the complete NC part programming procedure. The proposed system is an automated system where the input is a geometric model of a part that has been defined during product design and the output is a NC part program. The system possesses sufficient logic and decision making capability to accomplish NC part programming for the entire part without human assistance.

This can most readily be done for certain NC processes that involve well defined, relatively medium complex part geometries. Special algorithms have been developed to process the design data and generate the NC program.