COMPUTER-AIDED DESIGN AND MANUFACTURING

4^{TH} SEMESTER
MECHANICAL ENGINEERING

LECTURE NOTE

By
SRI BIPIN KUMAR DASH
Senior Lecturer
Department of Mechanical Engg.
Government polytechnic,Kandhamal
Phulbani-762001

UNIT-I (Introduction)

The **Product Cycle** spans conception to retirement, driven by a **Design Process** (needs, concept, detailed design, production, service) that can be **Sequential** (step-by-step) or **Concurrent** (overlapping, simultaneous), with **Computer-Aided Design (CAD)** tools digitally creating, analyzing (simulations, analysis), and drafting 2D/3D models, integrating seamlessly with Computer-Aided Manufacturing (CAM) for efficiency.

Product Cycle

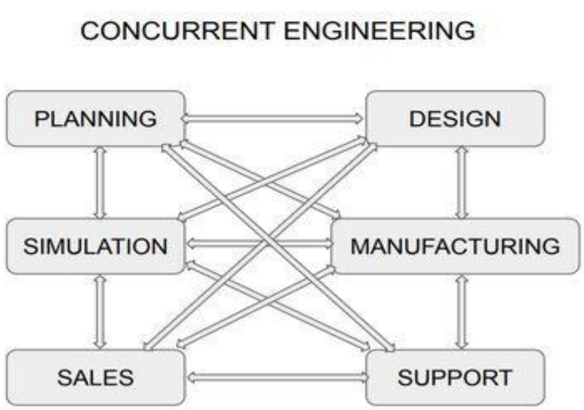
- **Definition:** The entire life of a product, from initial idea to withdrawal from the market.
- **Stages:** Conceive, Design, Realize (Manufacture/Produce), Service/Use, Retire.

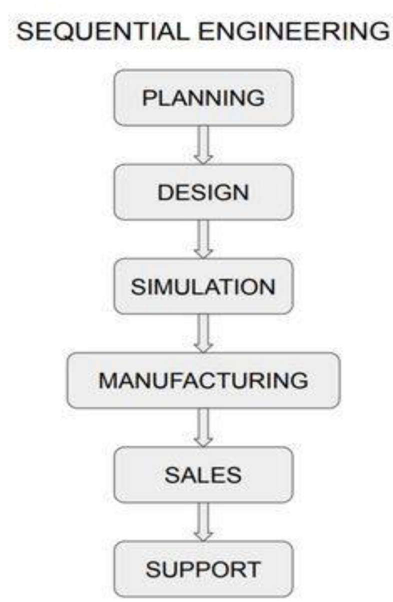
Design Process Models

- **Common Stages:** Problem Identification/Definition, Conceptual Design, Feasibility Study, Preliminary Design, Detailed Design, Production Planning, Manufacturing, Consumption, Retirement.

Sequential vs. Concurrent Engineering

- **Sequential (Traditional):** A linear, "siloed" approach where one stage (e.g., design) must finish before the next (e.g., manufacturing) begins, with communication only at handoffs.
- **Concurrent (Simultaneous/Simultaneous Engineering):** Overlapping development activities across the product lifecycle, using cross-functional teams working in parallel on design, analysis, and manufacturing planning for faster, integrated development.
- Concurrent engineering performs design, manufacturing, and other stages simultaneously using cross-functional teams for faster, integrated product development, while sequential engineering follows a linear, step-by-step process (like the Waterfall model) where one phase finishes before the next begins, leading to potential delays and rework.





-

Computer-Aided Design (CAD)

- **Definition:** Using computer systems to create, modify, analyze, and optimize product designs (2D drawings, 3D models).
- **Key Functions:**
    - **Geometric Modeling:** Building 2D/3D representations.
    - **Engineering Analysis (CAE):** Simulating performance (stress, flow, thermal).
    - **Automated Drafting:** Creating technical drawings (blueprints).
    - **Visualization:** Rendering, animation, shading.
    - **Integration:** Works with CAM for automated production planning.

CAD System Architecture

A CAD system architecture defines the layered structure (User Interface, Application Logic, Data Management) and core components (hardware, OS, software) for creating digital 2D/3D designs, enabling precise modeling, simulation (CAE), and manufacturing (CAM) through integration with other systems for collaboration and workflow efficiency, essentially replacing manual drafting with digital precision and iteration.

Key Components of CAD Architecture

- **Hardware:** The physical workstation, including powerful processing units (CPUs), graphics cards, memory (RAM), storage (HDDs/SSDs), and input/output devices like monitors, keyboards, mice, and specialized controllers (digitizers).
- **Operating System:** Manages the hardware resources and provides a platform for the CAD software, supporting multitasking and networking.
- **Application Software:** The core CAD programs (e.g., AutoCAD, SOLIDWORKS, Revit) that provide tools for geometric modeling, parametric design, assembly, and visualization.
- **Data Management:** Systems for storing, organizing, and retrieving design files, often involving database management for complex projects and version control.

Purpose of CAD System Architecture

CAD architecture provides the framework for digital design, boosting accuracy, speed, and efficiency by replacing manual drawing with powerful software, enabling detailed product development from concept to manufacturing across industries like architecture, engineering, and manufacturing.

Computer Graphics (CG)

Computer Graphics (CG) in CAD (Computer-Aided Design) are fundamental, enabling the creation, visualization, and manipulation of 2D drawings and 3D models digitally, replacing manual drafting with interactive systems that use vector graphics, solid modeling, and rendering for precise design, analysis (like stress tests), simulation, and manufacturing preparation across fields like engineering, architecture, and product design

Core Functions & Concepts of Computer Graphics (CG)

- **Geometric Modeling:** Uses algorithms and vector graphics to define shapes as points, lines, curves, surfaces, and solids.
- **2D Drafting** Creating precise technical drawings (blueprints) for layouts and components
- **3D Modeling:** Building virtual, dimensionally accurate models with spatial relationships and material properties.
- **Visualization:** Displaying models in wireframe, hidden-line, or photorealistic rendered views for better understanding.
- **Analysis & Simulation:** Testing models for physical properties, dynamic behavior, and stress before physical creation.
- **Manufacturing Integration:** Connecting to CAM (Computer-Aided Manufacturing) for CNC machining and other automated processes.

Key Applications Computer Graphics (CG)

- **Mechanical Engineering:** Designing complex parts, assemblies, and systems (e.g., automotive, aerospace).
- **Architecture & Construction:** Creating building plans, exploring layouts, and visualizing structures.
- **Product Design:** Developing everything from jewelry to appliances with detailed visualizations and simulations.
- **Media & Animation:** Creating models for special effects and digital content.

Co-ordinate systems in CAD

In CAD software, coordinate systems provide the mathematical framework for defining the exact location, size, and orientation of 2D and 3D objects. There are two types of co-ordinate system are used in CAD.

- **World Coordinate System (WCS):** The default, fixed reference system. Its origin (0,0,0) and axes (X, Y, Z) are permanent and cannot be moved.
- **User Coordinate System (UCS):** A movable coordinate system that users can reposition and rotate. It is essential for drawing on specific planes in 3D models or aligning geometry with non-orthogonal site layouts.

CAD software typically supports three ways to input these coordinates:

- **Absolute:** Coordinates are measured from the fixed (0,0,0) origin (e.g., 10,10).
- **Relative:** Coordinates are measured from the last point specified, preceded by the @ symbol (e.g., @5,2).
- **Relative Polar:** A distance and angle relative to the last point, entered as @distance<angle (e.g., @10<45).

Homogeneous Coordinate System

In CAD, a **Homogeneous Coordinate System** adds an extra dimension (e.g., 'w' or '1') to standard (x, y, z) coordinates, allowing all geometric transformations (translation, rotation, scaling, shearing) to be handled uniformly with a single matrix multiplication, simplifying complex operations and enabling representation of points at infinity for perspective views, essential for 3D rendering and modeling.

Why CAD Uses Homogeneous Coordinates ?

1. **Unified Transformations:** In standard Cartesian systems, rotation and scaling are matrix multiplications, but translation is an addition. Homogeneous coordinates turn translation into a 3x3 (for 2D) or 4x4 (for 3D) matrix multiplication, allowing all transformations to be combined into a single matrix.
2. **Chaining (Composition):** Multiple operations (e.g., move, then rotate, then scale) can be multiplied together into one final "transformation matrix." This is computationally efficient for high-performance CAD software.
3. **Points at Infinity:** Points with $w=0$ represent "ideal points" or directions at infinity. This is crucial for defining parallel lines that intersect at infinity and for modeling light sources in 3D rendering.
4. **Perspective Projection:** It enables the "division by distance" required to make objects appear smaller as they move further away, which is essential for realistic 3D viewing.

   Common Applications of Homogeneous Coordinates

- **CAD/CAM Software:** Used to position components, handle assembly hierarchies, and generate technical drawings.
- **Graphics Engines:** Modern APIs like OpenGL and Direct3D use 4x4 matrices and 4-element vectors to take advantage of GPU hardware acceleration.

2D and 3D transformations in CAD

2D and 3D transformations in CAD are fundamental operations (translation, rotation, scaling, reflection, shear) that manipulate an object's position, size, or orientation in flat (x, y) or volumetric (x, y, z) space, respectively, using matrix math to allow designers to easily edit and visualize models in software like AutoCAD, SolidWorks, and Creo for precise design and assembly. While 2D transformations stay on a plane, 3D transformations add depth (z-axis) and perspective for realistic modeling, crucial for everything from simple drafting to complex virtual reality applications.

Line Drawing :A CAD line drawing uses Computer-Aided Design software to create precise, scalable technical drawings, forming the foundation for architectural plans, engineering diagrams, and landscape designs, often created using the Line command to specify start/end points, lengths, and angles, with tools like Ortho mode and Object Snaps for accuracy. These digital lines form walls, outlines, or connections, with specialized symbols added for components, and can be printed or shared.

Clipping: Clipping in CAD is the process of defining a boundary to show only a specific portion of an object, like an Xref, image, block, or viewport, hiding the rest to clean up drawings and focus on relevant details. Commands like XCLIP, IMAGECLIP, and VPCLIP (or the unified CLIP in newer versions) let you create polygonal or rectangular clipping paths, often using existing lines or drawing new ones, to crop complex elements like linked drawings or aerial photos.

CAD/CAM

CAD/CAM (Computer-Aided Design/Computer-Aided Manufacturing) integrates computer systems for designing (CAD) and manufacturing (CAM) products, allowing seamless transition from digital creation to automated production, improving precision, reducing costs, and speeding up development by using software to create 3D models and generate instructions for CNC machines. This combined technology streamlines workflows, automates processes like milling and laser cutting, and helps create complex, high-quality parts efficiently.

CAD (Computer-Aided Design)

- **Function:** Uses software to create, modify, analyze, and optimize 2D and 3D product designs and engineering drawings.
- **Process:** Designers build digital models with precise geometry, dimensions, and material specifications
- **Benefits:** Easier editing, storage, and retrieval of designs, replacing traditional manual drafting.

CAM (Computer-Aided Manufacturing)

- **Function:** Uses software to plan, manage, and control manufacturing processes, directly interacting with production machines.
- **Process:** Converts CAD design data into machine instructions (like G-code) for automated machinery (CNC, robots).
- **Benefits:** Automates production, reduces labor, improves quality, and allows simulation for error checking before machining.

Types of Production

- **Job Shop Production:** Single, highly customized products (prototypes, aircraft parts) using flexible CAD/CAM tools for complex designs.
- **Batch Production:** Producing groups (batches) of similar items with variations (e.g., furniture, clothing).
- **Mass Production:** High-volume, standardized products (automobiles, appliances) leveraging automation from CAD models to CNC machines.
- **Continuous Flow:** 24/7 production of liquids or powders (chemicals, oil, food).

Manufacturing models and Metrics

In CAD/CAM (Computer-Aided Design and Computer-Aided Manufacturing), manufacturing models and metrics provide the mathematical and digital framework used to design parts and evaluate production performance.

Manufacturing Models in CAD/CAM :
Manufacturing models are digital or mathematical representations of a product or its production process. These models allow for virtual testing and optimization before physical manufacturing begins.

- **Geometric Models:** Represent the physical shape of a part using mathematical methods like **NURBS** (Non-Uniform Rational B-Splines) or **B-Rep** (Boundary Representation).
- **Mathematical Production Models:** These models calculate how a factory operates. For example, the **Operation Cycle Time (T_C)**

  $Tc = To + Th + Tth$

  where, T_O = processing time, Th =handling time, and Tth is tool handling time.

- **Simulation Models:** These represent the machining process itself, such as toolpath simulations to detect collisions or finite element analysis (FEA) to predict mechanical stresses.
- **Data Models:** Shared databases in integrated CAD/CAM environments ensure that design changes are automatically reflected in manufacturing instructions (G-code).

Key Manufacturing Metrics
Metrics are quantitative measures used to track the efficiency, cost, and quality of a production facility.

- **Production Rate (Rp):** The number of units produced per unit of time.
- **Plant Capacity:** The maximum rate of output that a plant can sustain under normal working conditions.
- **Utilization:** The proportion of time that a machine or system is actually being used for production compared to its total available time.
- **Availability:** Measured as the ratio of actual operating time to scheduled production time, often impacted by breakdowns (MTBF - Mean Time Between Failures).
- **Lead Time:** The total time required to process a given part through the plant, from initial order to finished product.
- **Quality Metrics:** Includes the reduction in **scrap and rework** (typically 20–30% lower with CAD/CAM) and maintaining tolerances (e.g., ISO 2768-1 standards).

Benefits of Integrated Metrics
Using these metrics within an integrated CAD/CAM environment (like Autodesk Fusion 360 or Siemens NX) leads to:

- **40–60% faster** programming time through automation.
- **15–25% increase** in equipment utilization.
- **Reduced time-to-market** due to shorter development cycles.

Mathematical models in CAD/CAM

Mathematical models in CAD/CAM use equations for geometric representation, process simulation, and performance prediction, enabling automated design, analysis (like FEM), toolpath optimization, and efficiency measurement (speed, yield) in manufacturing, by defining features, transformations (rotation, translation), and relationships between parts, tools, and machines for better productivity and quality.

Key Areas of Mathematical Modeling in CAD/CAM

- **Geometric Modeling:** Describes objects using mathematical equations for curves, surfaces, and solids (e.g., splines, NURBS, constructive solid geometry).
    - **Transformations:** Uses matrices for rotation, scaling, translation to manipulate designs.
    - **Feature Representation:** Models physical features (holes, slots) for automated manufacturing instructions.
- **Process Simulation & Analysis:** Predicts manufacturing outcomes before production.
    - **Toolpath Generation:** Models tool-workpiece engagement (e.g., helical grooves) using tangency conditions to optimize tool movement.
    - **Performance Metrics:** Calculates productivity, efficiency, and accuracy (e.g., time to machine, material waste).
    - **Interference Detection:** Models grinding wheel paths to avoid collisions.
- **System Optimization:** Creates algorithms for independent decision-making in automation.
    - **Mapping Functions:** Links design features to manufacturing features (e.g., design intent to machining operations).
    - **Algorithms:** Develops high-order algorithms for automated production planning and control, improving automation levels.

Examples of Mathematical Models

- **Feature-Based Models:** Mapping functions ($g_{\text{d-m}}$) to translate design features to manufacturing features.
- **Simulation Models:** Using software like VERICUT to test toolpaths and improve yield.
- **Biogeneric Models:** Mathematical models (e.g., Biog. CAD) for generating complex dental crown morphology faster and more naturally than conventional methods.

Benefits in CAD/CAM

- **Increased Productivity:** Reduces design, analysis, and manufacturing time.
- **Improved Quality:** Minimizes errors through detailed analysis and simulation.
- **Enhanced Automation:** Enables automated decision-making and process control.

UNIT-II(Geometric Modelling)

What is Geometric modeling?

Geometric modelling is the mathematical description and computer-based representation of 2D or 3D object shapes, widely used in CAD/CAM to create, manipulate, and analyze digital prototypes. It uses curves, surfaces, and solids to create precise shapes for manufacturing, engineering, and digital design.

Representation of curves:

Curve representation in computer graphics involves defining smooth shapes using mathematical formulas, primarily through parametric curves, Bezier curves, and B-splines. These methods use control points to define shapes, allowing for easy editing and rendering, with cubic splines being common for smooth, complex curves. Curves are used to create smooth, complex shapes in computer-aided design (CAD) and font design, transforming from control points into simple line segments (polylines) or triangles for display.

The need of representation of curve:
Representing curves mathematically provides a **precise and scalable** way to describe shapes that remains smooth at any resolution, which is essential for high-quality graphic design and typography. It allows computers to store complex visual information using **minimal data**, replacing thousands of individual pixels with a few simple coordinates and equations. In engineering and manufacturing, these representations ensure **aerodynamic smoothness and structural integrity** by allowing designers to control the continuity between surfaces. Furthermore, mathematical curves enable **dynamic animation**, letting artists manipulate shapes easily by moving a few "control points" rather than redrawing every frame. Finally, these representations are vital for **scientific analysis**, providing the formulas needed to calculate velocity, force, and distance along a specific path.

Hermite Curve

- **Definition:** A Cubic Hermite spline is a curve defined by two endpoints ($P0$,$P1$) and the tangent vectors (slopes) at these endpoints ($P'0$,$P'1$)
- **Characteristics:** It is an interpolating curve, meaning it passes through the endpoints.
- **Usage:** Used for interpolation and ensuring smooth connectivity between segments because of the control over tangent direction.

Bézier Curve

- **Definition:** A Bézier curve is defined by a set of control points ($P0$ through Pn). A cubic Bézier curve uses four points, where the curve starts at $P0$ and ends at $P3$,with $P1$and $P2$ controlling the shape.
- **Characteristics:** It is an approximating curve, generally not passing through intermediate control points, but staying within their convex hull.
- **Usage:** Widely used in vector graphics (e.g., SVG, PostScript), font design, and animations due to ease of manipulation.

B-spline (Basis spline) curve

A B-spline (Basis spline) curve is a piecewise polynomial curve defined by a set of control points, a degree, and a knot vector, offering high flexibility and local control in geometric modeling. Unlike Bezier curves, B-splines allow local changes to the curve's shape without affecting the entire structure and decouple the curve's degree from the number of control points.

Advantages of B-spline curve over Bézier Curves

- Independent Degree: The degree of the polynomial (k) can be chosen independently of the number of control points (n+1), allowing for lower-degree, easier-to-compute curves even with many points.
- Local Control: Changing a single control point does not affect the whole curve, unlike Bezier curves.
- Flexibility: Easily supports, adding control points without increasing the degree, and sharp corners can be created if needed.

RATIONAL CURVE

Rational curves are a fundamental concept in Computer-Aided Design (CAD) and Manufacturing (CAM), acting as a bridge between polynomial splines (like Bezier curves) and the ability to accurately represent conic sections such as circles, ellipses, and hyperbolas. They are a ratio of sums of polynomials, offering greater flexibility and control over the shape of curves compared to non-rational curves.

A rational curve is expressed as the ratio of two polynomial functions. While non-rational curves (polynomials) can only approximate conic sections, rational curves can represent them exactly.

Role in CAD/CAM

- **Shape Control:** Rational curves, particularly through NURBS, are used to model complex, smooth, and free-form surfaces in industrial design (e.g., automobile bodywork, aircraft).
- **Precision:** Because they can represent conic sections (circles, ellipses) exactly, they are crucial for manufacturing parts requiring high precision.
- **Data Exchange:** Rational curves (like NURBS) are a standard format for transferring geometric data between different CAD packages.
- **Computational Efficiency:** Although more complex than non-rational curves, their ability to accurately represent shapes reduces the need for complex, piecewise approximations

Applications

- **Conic Sections:** Rational quadratic Bezier curves are specifically used to represent ellipses, parabolas, and hyperbolas by adjusting weights.

- **Surface Modeling:** Used for generating smooth surfaces (e.g., in FEM applications, airfoil cross-sections).
- **CAD/CAM Software:** Widely implemented in CAD software to create smooth, editable curves for design and manufacturing, often replacing traditional physical splines.

Surface modeling

- Surface modeling techniques, essential for creating complex, aesthetic, and aerodynamic 3D shapes (car bodies, consumer products), define objects by their boundary faces rather than internal volume. Key methods include wireframe-based lofting, sweeping, and revolving, alongside synthetic, curvature-continuous patches like Bezier, B-Spline, and Coons patches to achieve smooth, complex surface transitions.
- Use surface diagnostic tools to detect gaps or overlaps in the mesh.
- Ensure all surfaces are joined (knitted) properly.
- Utilize curvature analysis to ensure smooth, aesthetic surfaces

Key Surface Modeling Techniques

- **Extrude Surface:** Creates a surface by extending a 2D profile along a straight path.
- **Revolve Surface:** Generates an axi-symmetric surface by revolving a profile around a central axis.
- **Loft Surface:** Creates a smooth, complex surface by transitioning between multiple 2D cross-section curves.
- **Sweep Surface:** Generates a surface by moving a profile curve along a defined path curve.
- **Boundary/Patch Surface:** Fills a complex, multi-sided hole by connecting boundary curves, allowing for high continuity for smooth transitions.
- **Ruled Surface:** Creates a surface linearly interpolated between two boundary curves.
- **Fillet/Blend Surfaces:** Blends sharp edges to ensure smooth tangential transitions between adjacent surfaces.
- **Offset Surface:** Creates a new surface at a designated distance from an existing one.

surface patch

A surface patch in CAD (typically command `SURFPATCH` in AutoCAD) creates a new surface to cap or fill an open gap formed by a closed loop of edges or curves. It is used to turn open, hollow models into closed solids, offering options for continuity (Position, Tangent, Curvature) and bulge magnitude to blend smoothly with existing geometry.

purpose: Closes openings, creates caps, and fixes surface modeling gaps, often preparing a surface for thickening into a solid.

Common Applications:

- Closing the top of an extruded cylinder or complex surface shape.
- Repairing broken or unfinished surface models in CAD.
- Creating complex organic shapes that are not easily defined by basic extrusion or revolution.

Coons Patch

A Coons patch in CAD is a surface modeling technique that creates a smooth, interpolated surface bounded by four existing boundary curves. Used to close gaps or patch holes in geometry, this method uses bilinear or bicubic blending to define a surface between curves (lines, arcs, or splines), often applied in reverse engineering and free-form design.

Bicubic patch

A bicubic patch in CAD is a parametric surface defined by 16 control points (a 4x4 grid) in a bicubic equation, often used to create complex, smooth, and flexible freeform shapes. These patches, including Bézier, Coons, or Hermite forms, are essential for sculpting surfaces in CAD/CAM, providing high-degree curvature (cubic) for modeling organic, automotive, or aerospace shapes.

A bicubic patch is mathematically defined as a surface where both parametric directions (u and v) are cubic functions. It uses 16 control points, where only the four corner points typically lie directly on the surface.

BEZIER SURFACE

Bezier surfaces in CAD are parametric, free-form surfaces defined by a grid of control points, widely used for creating smooth, organic shapes like automobile bodies and consumer products. Shaped by Bernstein polynomials, the surface is generally contained within the convex hull of its control points, ensuring intuitive manipulation.

B-Spline Surface

B-spline surfaces in CAD are flexible, smooth, and locally controlled 3D modeling tools defined by a grid of control points, a degree, and knot vectors. Unlike Bezier surfaces, B-splines allow local modification—moving one control point only affects the immediate surface area, making them ideal for complex, freeform surfaces in car bodies or aesthetic designs

Solid modelling techniques

Solid modelling techniques in CAD enable the creation of accurate, unambiguous 3D models containing volume and topological data for engineering analysis, manufacturing, and visualization. Key methods include Constructive Solid Geometry (CSG) using boolean operations, Boundary Representation (B-rep) focusing on faces and edges, and Parametric/Feature-based modeling for rapid design iteration.

Advantages of Solid Modelling
Solid modelling supports precise mass property calculations (weight, volume, centroid), interference detection in assemblies, Finite Element Analysis (FEA), and CAM operations like CNC toolpath generation.

Common Solid Operations

- **Protrusion/Extrusion:** Creating a 3D volume from a 2D sketch.
- **Boolean Operations:** Joining, cutting, or intersecting two or more solid bodies.

- **Pick-and-Place Features:** Applying surface-level changes like filleting, chamfering, or shelling (creating hollow, thin-walled objects).

Key Solid Modelling Techniques

- **Constructive Solid Geometry (CSG):** This technique builds complex models by combining basic 3D shapes (primitives) such as cubes, cylinders, and spheres using Boolean operations—Union (joining), Difference (subtracting), and Intersection.

  Constructive Solid Geometry (CSG) is a CAD modeling technique that creates complex 3D shapes by combining simple geometric primitives (blocks, cylinders, spheres) using Boolean operations—Union, Difference, and Intersection. It uses a procedural "tree" structure to store how parts are combined, ensuring solid, watertight models.

- **Boundary Representation (B-rep):** Defines a solid by its outer boundaries: vertices, edges, and faces. B-rep represents the "shell" of an object, requiring the geometry to be fully enclosed to ensure a valid solid, often used in finite element analysis.

  Boundary Representation (B-rep) is a widely used solid modeling method in CAD that defines a 3D object by its enclosing surfaces—faces, edges, and vertices. It combines topological data (connectivity) and geometric data (surfaces/curves) to precisely define a closed volume, distinguishing inside from outside, essential for engineering design, analysis, and CAM.

- **Parametric Feature-Based Modelling:** Widely used in modern CAD software (e.g., SolidWorks, Creo), this approach creates models using features (extrudes, revolves, holes, fillets) defined by parameters (dimensions, constraints). It allows designers to edit a dimension and automatically update the entire part.
- **Sweeping:** Generates solid volume by moving a 2D profile along a specified path or revolving it around an axis (Revolve).
- **Hybrid Modelling:** Combines CSG and B-rep techniques, allowing users to create complex geometries that are easy to manipulate and visually accurate

UNIT-III (CAD STANDARDS)

What is CAD standards ?

CAD standards are a set of uniform rules, guidelines, and best practices governing how 2D drawings and 3D models are created, organized, and delivered in CAD software. They ensure consistency, accuracy, and efficiency across projects by standardizing layers, line types, text styles, and file naming, often enforced through .DWS files

Graphics standards in computer graphics are established, agreed-upon specifications for software interfaces and data formats that ensure program portability, device independence, and interoperability between different hardware and software systems. Key standards like GKS and PHIGS enable graphics applications to run on various devices without rewriting code, primarily managed by organizations like ISO and ANSI.

Key Graphics Standards and Software

- **Graphical Kernel System (GKS):** Recognized as one of the first ISO/ANSI standards, GKS handles 2D graphics, providing a low-level, device-independent interface.
- **Programmer's Hierarchical Interactive Graphics Standard (PHIGS):** A sophisticated standard for 3D modeling, rendering, and complex, hierarchical interactive graphics.
- **PHIGS+:** An extension of PHIGS that provides advanced 3D surface-shading and lighting capabilities.
- **Computer Graphics Interface (CGI):** Formerly known as VDI (Virtual Device Interface), it defines the interaction between graphics software (like GKS) and the device driver code.
- **Computer Graphics Metafile (CGM):** Formerly VDM (Virtual Device Metafile), it is a standard for storing and transferring picture description information between different platforms.
- **OpenGL:** A widely used open standard Application Programmer Interface (API) for rendering 2D and 3D graphics, designed to work directly with hardware accelerators.

Graphical Kernel System (GKS)

GKS is an ISO standard used for 2D computer graphics programming that provides a device-independent graphics interface. , it provides a device-independent, portable framework for vector graphics. It enables programmers to create, manipulate, and store 2D pictures across different platforms using primitives like lines, text, and polygons.

Objective of GKS

Answer: The main objective of the **Graphical Kernel System (GKS)** in Computer-Aided Design (CAD) is to provide a **standardized, device-independent, 2D graphics programming framework. It also** provides a standard graphics interface that allows programs to run on different graphics devices without change.

Graphics primitives are basic drawing elements such as points, lines, polygons, and text used to create images. Graphics primitives in GKS (Graphical Kernel System) are the fundamental, standardized, two-dimensional building blocks used to create, display, and manipulate computer graphics images, ensuring portability across different platforms and hardware devices. These core elements include polylines, polymarkers, text, fill areas, cell arrays, and generalized drawing primitives

The main features of GKS are:

- Support for 2D graphics only
- Device-independent graphics programming
- Use of graphics primitives like lines, polygons, and text
- Logical coordinate system for drawing
- Ability to support multiple input and output devices
- These features make GKS simple and portable for basic graphics applications.

UNIT-IV

FUNDAMENTAL OF CNC AND PART PROGRAMING

Introduction to NC and CNC Systems

Numerical Control (NC) vs. Computer Numerical Control (CNC)

- **NC Systems**: Use hardwired electronic circuits to read instructions from punched tape. They lack internal memory and cannot store or edit programs on the machine.
- **CNC Systems**: Use an internal microprocessor or computer controller. They store programs in memory, support on-screen editing, and offer real-time diagnostics.

Machine Axes and Coordinate Systems

CNC machines use a standard **Right-Hand Cartesian Coordinate System** to establish tool movement vectors.

- **Z-Axis**: Always aligned parallel to the principal machine spindle. For a milling machine, it controls tool depth; for a lathe, it controls the longitudinal movement along the work piece
- **X-Axis**: Always horizontal and perpendicular to the Z-axis. On a lathe, it controls the diameter of the part. On a milling machine, it controls the longest horizontal travel table axis.
- **Y-Axis**: Perpendicular to both X and Z axes to complete the 3D coordinate system (primarily used in milling).
- **Rotational Axes**: Designated as **A** (rotating around X), **B** (rotating around Y), and **C** (rotating around Z).

Machine Origin vs. Work Origin

- **Machine Home** : The fixed, permanent reference point defined by the machine manufacturer.
- **Work Home or Part Zero**: A floating reference point chosen by the programmer on the work piece to make blueprint dimensions match program coordinates.

2. CNC Machine Tools and Operation

Principle of Operation

1. **CAD/CAM Design**: A 2D or 3D model is drawn and converted into tool paths.
2. **NC Program Input**: G-code data is loaded into the CNC Controller via USB, network, or DNC.
3. **Signal Processing**: The controller decodes the code lines and calculates precise trajectory coordinates.
4. **Servo Execution**: Feedback loop systems send voltage pulses to axis drive motors.
5. **Physical Machining**: The cutting tool interacts with the work piece while sensors feed positional data back to the controller.

```
[CAD Model] ➔ [CAM / G-Code] ➔ [CNC Controller] ➔ [Servo Drives] ➔ [Machine Axis
Motion]
                                      ▲                                        |
                                      └─────────── [Feedback Sensors] ─────────┘
```

Construction Features and Structures

Modern CNC machines require immense rigidity and thermal stability to handle high acceleration forces.

- **Machine Bed**: Cast iron or resin-bonded concrete structures designed to dampen structural vibrations.
- **Guideways**: Linear motion (LM) guides or hardened box ways minimize friction and eliminate stick-slip.
- **Recirculating Ball Screws**: Replaces standard lead screws to eliminate mechanical backlash and provide precise positioning down to micron levels.

Drives and Controllers

- **Spindle Drives**: High-torque AC or DC motors providing infinitely variable speeds.
- **Axis Drives**: **Servo motors** featuring closed-loop optical encoder feedback are preferred over open-loop stepper motors for precision operations.
- **CNC Controller**: The computational unit (e.g., Fanuc, Siemens, Haas) handling geometric interpolation, tool offsets, and safety interlocks.

3. 2D vs. 3D Machining

- **2D Machining**: Tool paths where the cutting depth is fixed, and cutting occurs only along the and axes simultaneously (e.g., flat profiling, drilling).
- **2.5D Machining**: The tool changes depth along the axis, but only executes cutting motion while is stationary on a fixed plane.
- **3D Machining**: Simultaneous multi-axis motion across X, Y, and Z vectors to carve complex organic surfaces, injection molds, and sculpted aerospace components.

4. Manual Part Programming (Lathe & Milling)

Types of Programming

- **Manual Programming**: The programmer manually calculates coordinate intersections and writes text-based G-code lines.
- **Conversational Programming**: The operator inputs parameters (e.g., pocket depth, diameter) into visual menu prompts on the shop floor controller.
- **Computer-Aided Manufacturing (CAM)**: Software automatically computes complex tool paths from 3D CAD files.

Preparatory (G) Codes & Miscellaneous (M) Codes

Common G-Codes	Function	Common M-Codes	Function
G00	Rapid positioning	**M00**	Program stop
G01	Linear interpolation (Feed)	**M03**	Spindle ON (Clockwise)
G02	Circular interpolation (CW)	**M04**	Spindle ON (Counter-Clockwise)
G03	Circular interpolation (CCW)	**M05**	Spindle Stop
G21	Metric units (mm)	**M06**	Tool change
G90	Absolute programming	**M08**	Coolant ON
G91	Incremental programming	**M09**	Coolant OFF
G94	Feed per minute (Milling)	**M30**	Program end & Reset

5. Sample Manual Programs

Example A: CNC Milling Program (2D Profile Cavity)

This program cuts a square outer contour from absolute point

```
O1001 (SAMPLE MILLING PROGRAM)
N10 G21 G90 G94 G40 (Metric, Absolute, Feed/min, Cutter comp cancel)
N20 T01 M06 (Select Tool 1, Execute Tool Change)
N30 S1500 M03 (Spindle speed 1500 RPM, CW)
N40 G00 X-10.0 Y-10.0 Z5.0 M08 (Rapid to safe start clearance, Coolant ON)
N50 G01 Z-5.0 F100 (Feed down to cutting depth)
N60 G01 X100.0 F250 (Cut along X-axis to X100)
N70 G01 Y100.0 (Cut along Y-axis to Y100)
N80 G01 X0.0 (Cut back along X-axis to X0)
N90 G01 Y0.0 (Cut back along Y-axis to Y0)
N100 G00 Z10.0 M09 (Rapid retract, Coolant OFF)
N110 G28 G91 Z0 M05 (Return Z home, Spindle stop)
N120 M30 (End of program)
```

Example B: CNC Lathe Program (Turning Profile)

This program performs a simple outer diameter reduction turning step on a cylindrical bar.

```
O2002 (SAMPLE LATHE PROGRAM)
N10 G21 G18 G99 G40 (Metric, XZ Plane, Feed/rev, Tool comp cancel)
N20 T0101 (Select Tool 1, Apply Offset 1)
N30 G96 S200 M03 (Constant surface speed 200 m/min, Spindle CW)
N40 G00 X55.0 Z2.0 M08 (Rapid positioning near workpiece blank)
N50 G01 X50.0 Z0.0 F0.2 (Face the front edge down)
N60 G00 X50.0 Z2.0 (Retract)
N70 G01 Z-40.0 F0.25 (Turn diameter down to Z-40mm depth)
N80 G01 X55.0 (Retract clean away from part wall)
N90 G00 X150.0 Z50.0 M09 (Move to safe area, Coolant OFF)
N100 M30 (End of program)
```

6. Advanced Programming Features

Cutting Cycles (Canned Cycles)

Canned cycles condense multiple lines of repeated tool motions into single blocks.

- **Milling (G81/G83)**: Fixed drilling cycles. **G83** executes peck drilling to clear chips from deep holes.
- **Turning (G71/G72)**: Stock removal cycles. **G71** handles rough turning across a profile definition block without manually writing out every single pass.

Loops and Subprograms

When identical geometries must be machined at different locations, programmers use subprograms to avoid duplicating lines of code.

- **M98**: Calls a subprogram specified by an address code.
- **M99**: Ends a subprogram and returns operation back to the main parent program.

Macros (Parametric Programming)

Macros use variables (e.g., `#101 = 50.0`) and logic switches (`IF`, `THEN`, `GOTO`) inside G-code. This allows a single program to dynamically adjust its dimensions to cut families of similar parts with different scaling rules.

7. Introduction to CAM Packages

Computer-Aided Manufacturing (CAM) software automates the programming pipeline for intricate geometric geometries. Common industry packages include Mastercam, Autodesk Fusion 360, and SolidCAM.

The CAM Workflow

1. **Geometry Import**: A solid model file (.STEP, .IGS, or native CAD data) is brought into the workspace environment.
2. **Job Setup**: The raw stock bounding box boundaries and coordinate origins are assigned.
3. **Toolpath Strategy Selection**: The programmer chooses specific strategies (e.g., adaptive pocket roughing, 3D contour finishing, parallel clearing).
4. **Tool Parameter Assignment**: Feed rates, rotational speeds, and tool geometric sizing are configured.
5. **Simulation**: Virtual verification checks for machine gouges, tool holder interference, or collisions.
6. **Post-Processing**: The CAM engine translates software tool paths into machine-specific G-code syntax corresponding to the targeted controller.

UNIT-V

CELLULAR MANUFACTURING AND FLEXIBLE MANUFACTURING SYSTEM (FMS)

Group Technology (GT) & Part Families

Group Technology (GT)

Group Technology is a manufacturing philosophy where similar parts are identified and grouped together to exploit their design and manufacturing similarities. Instead of treating every part as unique, GT rationalizes production by standardizing processes for a collection of similar parts.

Part Families

A **Part Family** is a collection of parts that possess similarities in geometric shape and size (design attributes) or in the processing steps required for their manufacture (manufacturing attributes).

- **Design Attributes:** External/internal shape, dimensions, aspect ratios, tolerances, and material type.
- **Manufacturing Attributes:** Operations (milling, turning), tools required, cutting speed, batch size, and machine routing.

```
                 [ All Factory Parts ]
                           |
              ┌────────────┴────────────┐
      [Part Family A]            [Part Family B]
     (Similar Shapes)         (Similar Processes)
```

Methods for Forming Part Families

1. **Visual Inspection:** Grouping parts visually by looking at physical components or drawings. It is the simplest and least expensive method but highly subjective and unreliable for large volumes.
2. **Parts Classification and Coding:** Categorizing parts systematically using a alphanumeric code based on features.
3. **Production Flow Analysis (PFA):** Grouping parts by examining the actual manufacturing route sheets rather than part drawings.

Parts Classification and Coding Systems

Classification structures assign code numbers to parts to reveal design and manufacturing features. There are three basic structures:

- **Monocode (Hierarchical Code):** Each digit's meaning depends on the value of the previous digit. Captures deep information in fewer digits but is complex to design.
- **Polycode (Attribute Code / Chain Code):** Every digit has a fixed meaning independent of the others. Easy to learn and parse but results in longer codes.
- **Hybrid Code:** A mixture of mono code and poly code features. Most commercial coding systems (like Opitz) use this layout.

Opitz Parts Classification & Coding System

The Opitz system is a widely used **hybrid coding system** consisting of a core 9-digit alphanumeric string, optionally followed by 4 secondary digits.

```
[ 1 ][ 2 ][ 3 ][ 4 ][ 5 ]   -   [ 6 ][ 7 ][ 8 ][ 9 ]   -   [ A ][ B ][ C ][ D ]
└─────── Form Code ──────┘      └ Supplementary Code ┘      └─ Secondary Code ─┘
```

1. **Form Code (Digits 1–5):** Focuses on part design attributes (primary shape, envelope dimensions, internal/external surfaces).
2. **Supplementary Code (Digits 6–9):** Focuses on manufacturing variables (dimensions, raw material type, accuracy).
3. **Secondary Code (Digits A–D):** Completely optional; tailored by individual companies to track specific operation sequences.

General Breakdown of Form Code (First 5 Digits)

- **Digit 1 (Part Class):** Identifies if the part is rotational or non-rotational based on its Length-to-Diameter (L/D) ratio.
  - `0`: Rotational (L/D ≤ 0.5)
  - `1`: Rotational (0.5 < L/D < 3)
  - `2`: Rotational (L/D ≥ 3)
  - `3`, `4`: Special rotational parts
  - `6`, `7`, `8`, `9`: Non-rotational (prismatic) parts
- **Digit 2:** External shape and external surface features.
- **Digit 3:** Internal shape and internal features (e.g., holes, steps, threads).
- **Digit 4:** Machining requirements for plane surfaces (e.g., flats, slots).
- **Digit 5:** Auxiliary holes, teeth, or gear-like features.

Simple Numerical Problems in Opitz Coding

Problem 1: Rotational Step-Shaft

Statement: Determine the Form Code (first 5 digits) for a smooth rotational stepped shaft with no threads, a total length of 150 mm, and a maximum diameter of 60 mm. It features a keyway on the outer flat surface but has no internal bores or auxiliary features.
Reference Table Data (Simplified for Exam Use):

- **Digit 1:** Rotational part with L/D = 150/60 = 2.5. From standard Opitz charts, 0.5 < L/D < 3 → **Code 1**.
- **Digit 2:** Stepped on one end with smooth outer features → **Code 1**.
- **Digit 3:** Solid cylinder, no internal holing/bore → **Code 0**.
- **Digit 4:** Flat surface machining present (keyway slot) → **Code 1**.
- **Digit 5:** No auxiliary holes or gear teeth → **Code 0**.

Answer: The Opitz Form Code is **11010**.

Production Flow Analysis (PFA)

Production Flow Analysis (PFA) is a systematic method that groups parts into families and machines into cells by analyzing information given on **route sheets** (process plans) rather than geometric profiles.

Steps in PFA

1. **Data Collection:** Gather part numbers and their routing information (the sequence of machines they must visit).
2. **Sorting of Route Sheets:** Categorize parts into groups that share identical or highly similar machine routings.

3. **PFA Matrix Generation (Part-Machine Incidence Matrix):** Construct a binary chart where rows represent machines and columns represent parts. An entry of `1` indicates that a part requires that specific machine; `0` indicates it does not.
4. **Cluster Analysis:** Rearrange the matrix rows and columns using algorithms (like **Rank Order Clustering**) to group the `1` entries into blocks along the main diagonal. These blocks represent the final machine cells.

```
  Initial Matrix                 Clustered Matrix
    P1  P2  P3                      P2  P1  P3
M1   1   0   1                 M2    1   0   0  <-- Cell 1
M2   0   1   0      ──→        M1    0   1   1  <-- Cell 2
M3   0   0   1                 M3    0   1   1  <-- Cell 2
```

Cellular Manufacturing & Composite Part Concept

Cellular Manufacturing

Cellular Manufacturing is the physical application of Group Technology. Dissimilar machines are physically organized into a processing cluster (called a **machine cell**) dedicated to fabricating a specific part family. This completely replaces old functional layouts (where all lathes were in one room and all mills in another).

Composite Part Concept

A **Composite Part** is a hypothetical, ideal part that encompasses **all** geometric and manufacturing attributes present across an entire part family.

- Instead of designing a cell to produce a single part, the production cell is engineered to accommodate this master "composite part".
- Any actual part within that family will only require a subset of the operations built into the composite cell, ensuring no structural setup changes are ever needed.

```
Part 1: [Cylinder + Hole] ──┐
                            ├──→ Composite Part: [Cylinder + Hole + Keyway]
Part 2: [Cylinder + Keyway] ┘                   │
                                                ▼
                         Designed Cell Layout (Lathe + Drill + Mill)
```

Flexible Manufacturing Systems (FMS)

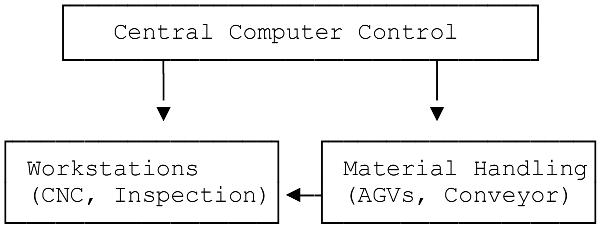
A **Flexible Manufacturing System (FMS)** is a highly automated, computer-controlled production system. It integrates automated material handling equipment with CNC workstations, enabling the plant to produce a variety of different parts concurrently with zero downtime for changeovers.

Types of Flexibility in FMS

- **Machine Flexibility:** Ability of a single machine to perform multiple types of operations on a part.
- **Routing Flexibility:** Ability to route a part through alternative machine paths if a bottleneck or breakdown occurs.
- **Product Flexibility:** Ease of switching to a completely new product family.
- **Volume Flexibility:** Ability to operate efficiently and profitably at varying production volumes.
- **Process Flexibility:** Ability to switch between different step sequences for the same part profile.
- **Expansion Flexibility:** Ease of adding new machines or expanding physical system physical boundaries.

FMS Components & Operations

An FMS is composed of three foundational elements connected by a centralized computer processor:



1. **Workstations:** Typically CNC machine tools, automated assembly equipment, coordinate measuring machines (CMMs), and part-washing stations.
2. **Automated Material Handling and Storage System:** Equipment used to transport parts and assemblies between workstations. This includes:
   - **AGVs (Automated Guided Vehicles)** and Automated Conveyors.
   - Robotic arms for part loading/unloading.
   - AS/RS (Automated Storage and Retrieval Systems) for raw stocks.
3. **Computer Control System:** The central brain coordinating the NC part programs, tracking inventory, directing material carriers, and generating schedule diagnostics.

FMS Planning, Control, Applications & Benefits

FMS Applications & Benefits

- **Applications:** Ideal for mid-variety, mid-volume production (e.g., aerospace frames, automotive engine blocks, complex hydraulic bodies). **Benefits:**
   - Drastic reduction in manufacturing lead times and setup durations.
   - Reduced Work-in-Process (WIP) inventories.
   - High machine tool utilization efficiency.
   - Improved component quality due to minimized human error.

FMS Planning and Control Issues

Managing an FMS is split into two primary phases:

Phase	Core Objective / Decision Areas
Planning Phase	System design decisions: choosing the mix of part families, selecting proper machinery counts, sizing handling systems, and budgeting tools.
Control Phase	Real-time scheduling decisions: dynamic part routing, tool wear monitoring, tracking machine health, and managing instantaneous system traffic bottlenecks.

Quantitative Analysis in FMS

Quantitative modeling uses mathematical calculations to evaluate system operational variables like production capacity, machine usage, and optimal transport speeds.