ME-430 Introduction to CAD
Lecture Notes- Part 3

Dr. Rajesh N. Dave

Office: 316 MEC (only during office hours); 208 YCEES

Phone: 973 596-5860

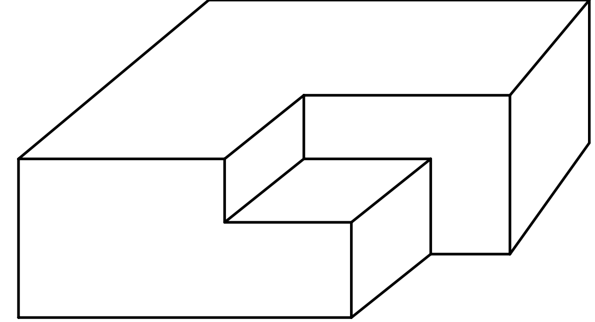
e-mail: dave@adm.njit.edu

Web Page: web.njit.edu/~rdave

Office Hours: Tues/Thurs 9:15 – 9:55 am,
Wed 5:15-5:45 pm, or by appointment

Lecture notes compiled by Profs. Surjunhatha and Dave

Review Questions



1. Briefly describe various methods of solid model representation. Which of these methods are available in the Pro/E package?
2. Based on the material covered in the class as well as the text-book, evaluate the Pro/E as a solid modeling package.
3. How would you define "Computer-Aided Design"? What is the role of computer in the design process?
4. Which new discovery can be considered as the most significant advancement in the field of CAD? (Why?)
5. What is the relationship between CAD and computer graphics?
6. What is the role of computerized drafting in CAD? To what extent computerized drafting can be considered CAD?
7. Give an example of a dimensionally non-homogeneous object created through the sweep technique.
8. Is Wire Frame Geometric Modeling technique a good solid modeling technique? Why?
9. What is the role of Geometric Modeling in CAD?
10. If you were given a choice between a CAD package that only had a CSG technique along with a set of primitives such as block, cylinder, cone, and sphere and a CAD package that only had a sweep technique with extrude and revolve operations, which one will you prefer? Why?
11. How does the technique called *Cell Decomposition* compare with the CSG technique? Does it have any practical applications? Is it available in Pro/E?
12. Sketch a CSG binary tree for the part above.
13. What is the major solid modeling technique that Pro/E is based on?
14. Discuss advantages and disadvantages of the *Spatial Occupancy Enumeration* technique. Is this method available in Pro/E

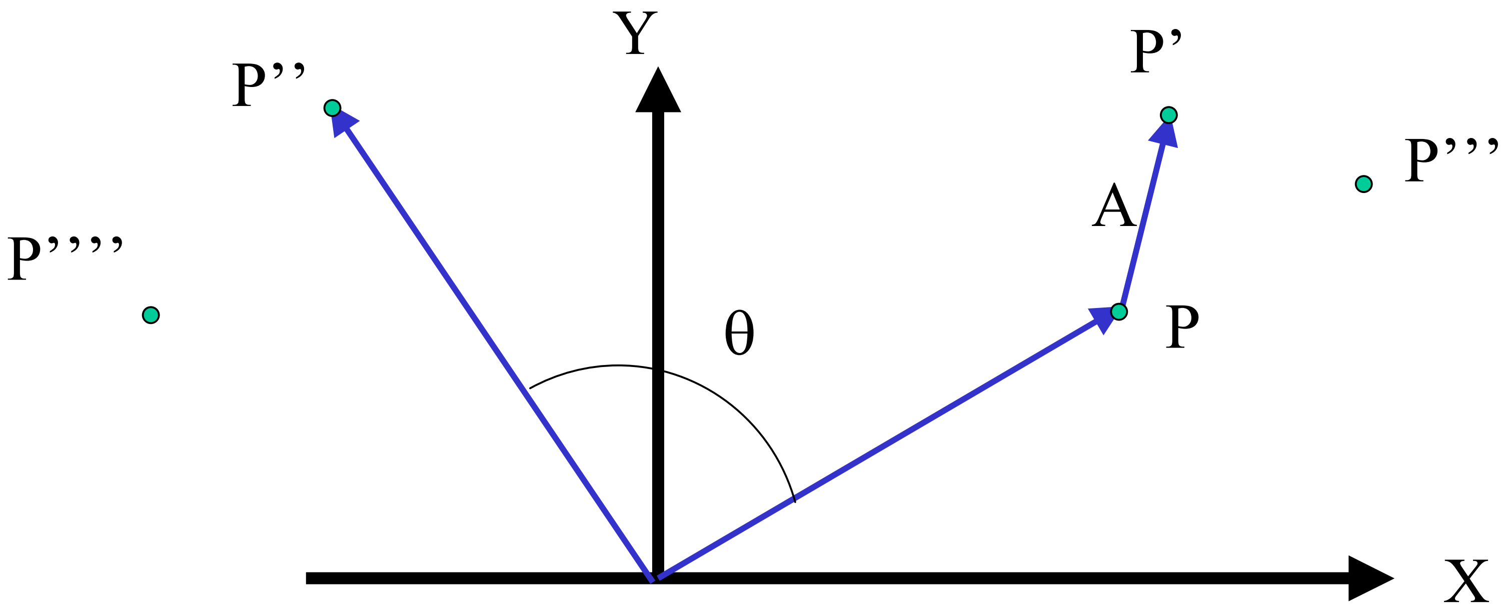
Geometric Transformations/ Computer Graphics

- Computer Graphics/ Visual Realism
  - Low-level Graphics
  - Geometric Transformations - Homogeneous Matrices and Transformations, Projections
  - Graphics Packages - GKS, PHIGS, etc.
  - Wireframe, Surface, or Solid Displays
  - Hidden Line/ Hidden Surface Removal
  - Shaded Images
  - Modeling and Display of
    - Landscapes
    - Trees
    - Clouds
    - Flexible Objects - Ropes, Fabrics, etc.

Geometric transformations

- Primitives (solids, surfaces, curves and points) in a solid modeling package need to be translated and/or rotated, and/or scaled before applying other CAD operations.
- Any rigid body can be considered as a collection of points or particles. Even in a boundary representation, the object has many points/vertices.
- Geometric transformations can be handled by first understanding how a point (or a vector) transforms, followed by transformation of the total Body or Working Coordinate System (WCS) and relate it to general or Model Coordinate System (MCS)

Transformation of Points and Vectors



Translation: $P' = P + A$

Rotation (with respect to origin): $X'' = X \cos\theta - Y \sin\theta$, $Y'' = X \sin\theta + Y \cos\theta$

Scaling (with respect to origin): $X''' = m X$, $Y''' = n Y$

(m is the scale factor in X direction, and n is the scale factor in Y direction)

Reflection (about Y axis – in 2-D) $X'''' = - X$, $Y'''' = Y$

Reflection (about X axis – in 2-D – not shown) $X'''' = X$, $Y'''' = - Y$

Matrix Representation

- All the transformations shown in the previous slide, except translation, can be represented in a matrix form. Examples below are in 2-D, but can be easily extended to 3-D.
- Rotation – about Z axis for a 2-D point in X-Y plane:

$$\begin{vmatrix} X' \\ Y' \end{vmatrix} = \begin{vmatrix} Cos\theta & -Sin\theta \\ Sin\theta & Cos\theta \end{vmatrix} \begin{vmatrix} X \\ Y \end{vmatrix}$$

- Scaling by amount m in X and n in Y

$$\begin{vmatrix} X' \\ Y' \end{vmatrix} = \begin{vmatrix} m & 0 \\ 0 & n \end{vmatrix} \begin{vmatrix} X \\ Y \end{vmatrix}$$

- Reflection about X axis

$$\begin{vmatrix} X' \\ Y' \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix} \begin{vmatrix} X \\ Y \end{vmatrix}$$

Note on Matrix Theory

- If a matrix has m rows and n columns, it is said to be a m x n matrix. It can be represented in an index notation as,
  $A = a_{ij}$, $i = 1$ to m, $j = 1$ to n
- If another matrix B is of size n x p, then the product of A and B is a matrix C of size m x p.

$$C = c_{ij} = \sum_{k=1}^{n} a_{ik}\, b_{kj}$$

- In general, the matrix prodcut AB is not the same as BA. In fact the resulting matrix BA may not even be meaningful. If the products are meaningful, the following rules apply.

  $A\,(BC) = (AB)\,C$
  $(A + B)\,C = AC + BC$ or $C\,(A + B) = CA + CB$

Homogeneous Transformations

- The geometric transformations such as scaling, rotation, reflection can be applied successively to any point or object, and the final effect can be represented as a single, concatenated matrix. For example, is scaling (matrix A) is followed by reflection (matrix B) followed by a rotation (matrix C), then the complete effect is a single matrix D = ABC.
- In real situations, the transformation also involves translations. Since translation cannot be carried out in matrix form, the concept of using a single transformation matrix does not work.
- Can you think of a way to do translation by applying a matrix transformation, i.e. carrying out a product of the point-vector with a transforming matrix?

Homogeneous Transformations (contd.)

- One could use the concept of homogeneous coordinates to represent all the operations conveniently in a matrix form.
- Ordinary Cartesian coordinates can be converted to homogeneous coordinates by adding an extra dimension, representing the homogeneous coordinate. Thus a point P in Cartesian coordinate is transformed to the homogeneous coordinate as follows:

  $P = [x\ y\ ]^T \rightarrow P' = [x\ y\ 1]^T$ or $p' = [xH\ yH\ H]^T$, where H is the homogeneous scale factor

  In 3-D,
  $P = [x\ y\ z\ ]^T \rightarrow P' = [x\ y\ z\ 1]^T$ or $p' = [xH\ yH\ zH\ H]^T$, where H is the homogeneous scale factor

  (note that transpose is taken, as normally the position is a column vector of size 3 x 1)

Homogeneous Transformations (contd.)

- After carrying out the transformations in the higher dimension, one can revert back to the original space as follows:

  $P' = [x\ y\ H]^T \rightarrow P = [x/H\ y/H]^T$ where H is the homogeneous scale factor

  In 3-D,
  $P' = [x\ y\ z\ H]^T \rightarrow P = [x/H\ y/H\ z/H]^T$, where H is the homogeneous scale factor

  (note that transpose is taken, as normally the position is a column vector of size 3 x 1)

Homogeneous Transformations (contd.)

- Can you guess how one can carry out a translation through a matrix product using the concept of homogeneous transformation?

  Translation by amount l in x, m in y and n in z:

$$T_{trans} = \begin{vmatrix} 1 & 0 & 0 & l \\ 0 & 1 & 0 & m \\ 0 & 0 & 1 & n \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_{rot-X-\theta} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & Cos\theta & -Sin\theta & 0 \\ 0 & Sin\theta & Cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad T_{scaling} = \begin{vmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_{reflect-XY} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad T_{reflect-X} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_{rot-Y-\theta} = \begin{vmatrix} Cos\theta & 0 & Sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -Sin\theta & 0 & Cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$T_{rot-Z-\theta} = \begin{vmatrix} Cos\theta & -Sin\theta & 0 & 0 \\ Sin\theta & Cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

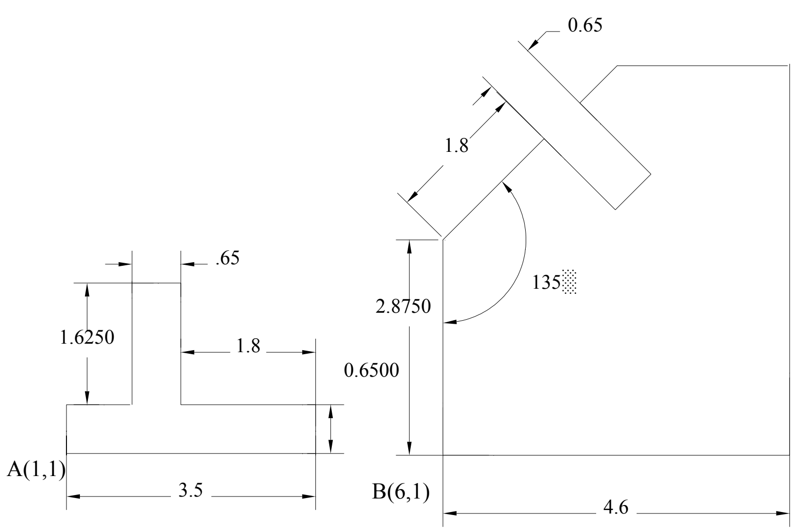
Projections

- In computer graphics, 3-D objects must be displayed on 2-D surfaces such as computer screens or paper
- This requires projection of a 3-D object onto a 2-D surface
- There are two major types of projections, parallel and perspective
- When you sketch an object, which projection do you normally use?

Projections (contd.)

- When a projection is done using parallel rays of projection, then the result is a parallel projection, and if the rays are all originating (or arriving) at a point, it is called a perspective projection.
- For a parallel projection, if the projection rays are normal to the projection plane, then the result is called axonometric or orthographic projection -- otherwise it is called an oblique projection.

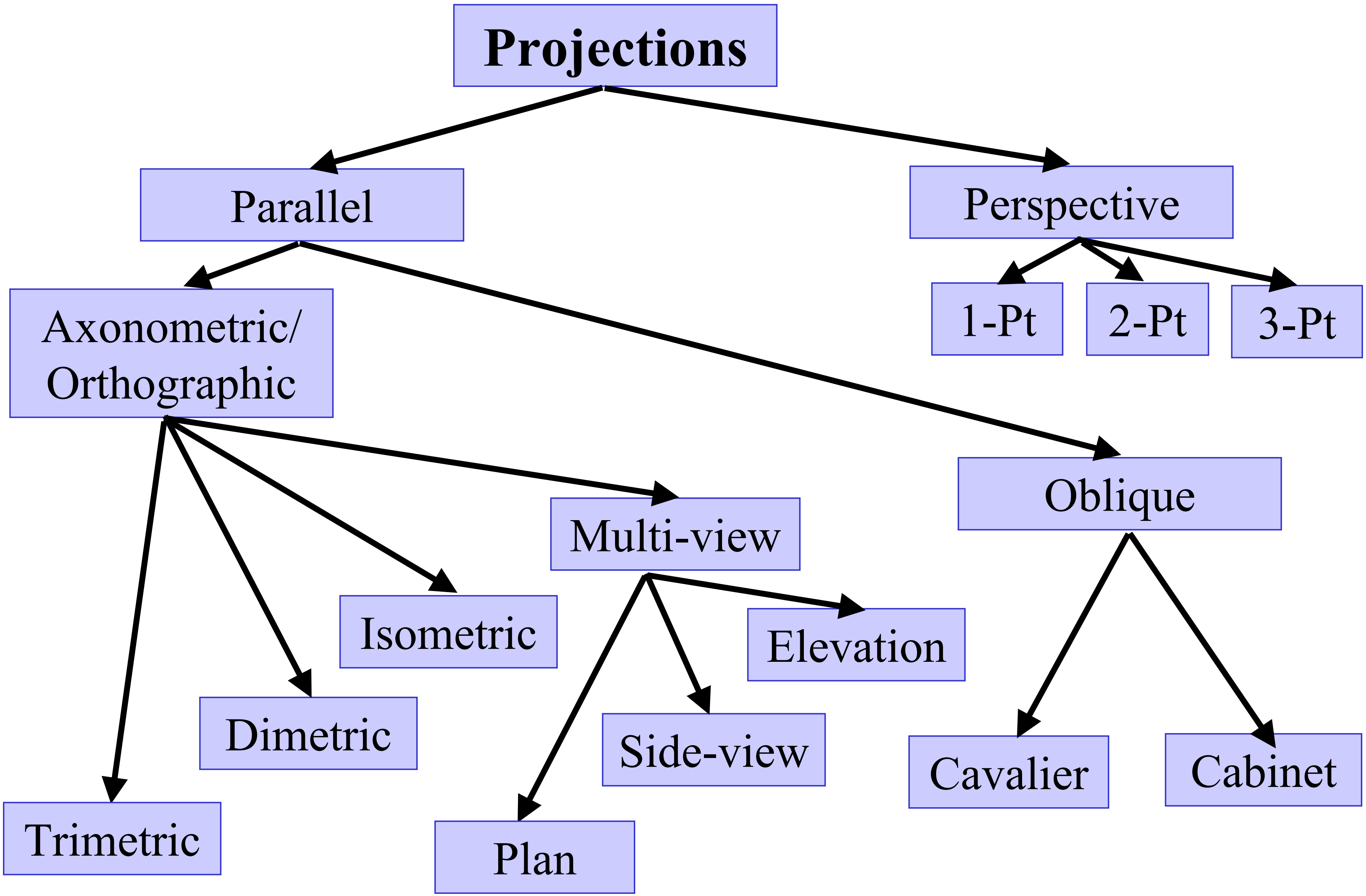
Review Questions



1.
Consider a plane in 3-D space such that the *z* axes and the point (1,1,1) lie on it. Derive a homogeneous transformation matrix which can not only perform a 3-D reflection about this plane but will also magnify the objects to twice their original size.

2.
Reflection of a line joining points $P_1(1,1)$, $P_2(2,4)$ is to be carried out about line joining (2,2) and (5,5). Derive a 4×4 homogeneous transformation matrix which can perform this operation, and compute the new coordinates of P_1 and P_2.

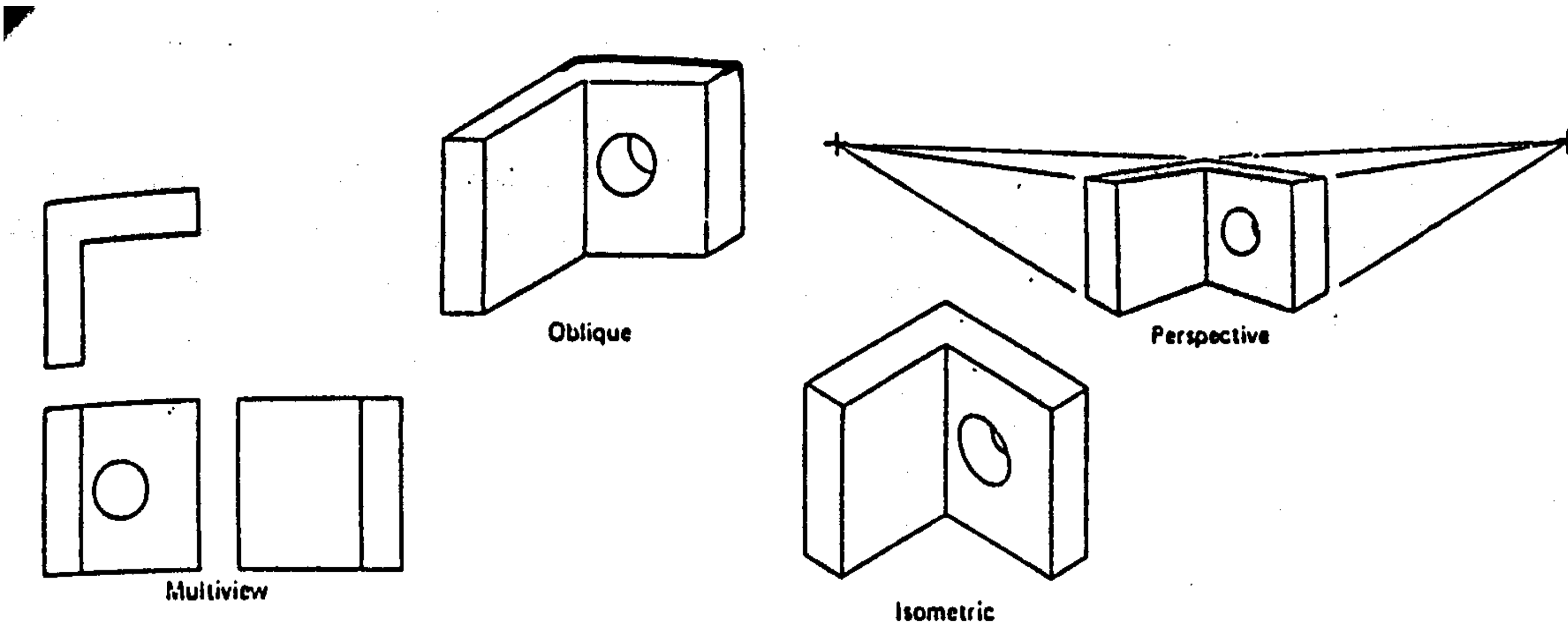
3.
Consider the following two-dimensional objects where the part on the left needs to be moved so as to fit the opening in the mating part situated as shown. Note that the lower left corner of the left part is at A(1,1), and of the mating part on right is at B(6,1). Derive a 3×3 homogeneous transformation matrix which can perform this operation. **Provide the final matrix.**


Projections
Parallel
Perspective
Axonometric/
Orthographic
1-Pt
2-Pt
3-Pt
Oblique
Multi-view
Isometric
Elevation
Dimetric
Side-view
Cavalier
Cabinet
Trimetric
Plan

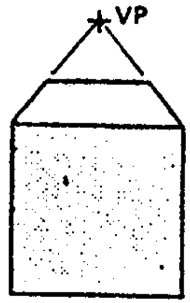
Axonometric or Orthographic Projections

- Multi-view: These projections are along the three principal axes, i.e., the direction of the projection is parallel to one of the three principal axis.
  - Projection in the direction of the vertical axis (i.e. plane of projection is the horizontal plane) is the *Plan-View*
  - Projection in the direction of the axis coming out of the front plane (i.e. plane of projection is the front plane) is the *Elevation,* sometimes called *Front-View*
  - Projection in the direction of the horizontal axis in the front-plane (i.e. plane of projection is the vertical plane, aligned to one side) is the *Side-View*
- Isometric: The direction of projection makes equal angles with all three principal axes
- Dimetric: The direction of projection makes equal angles with two principal axes
- Trimetric: The direction of projection makes unequal angles with all three principal axes

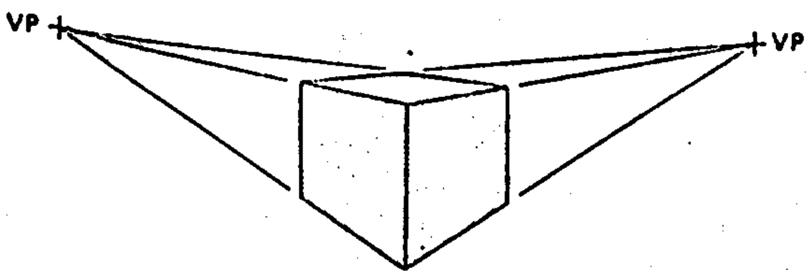
Examples of projections



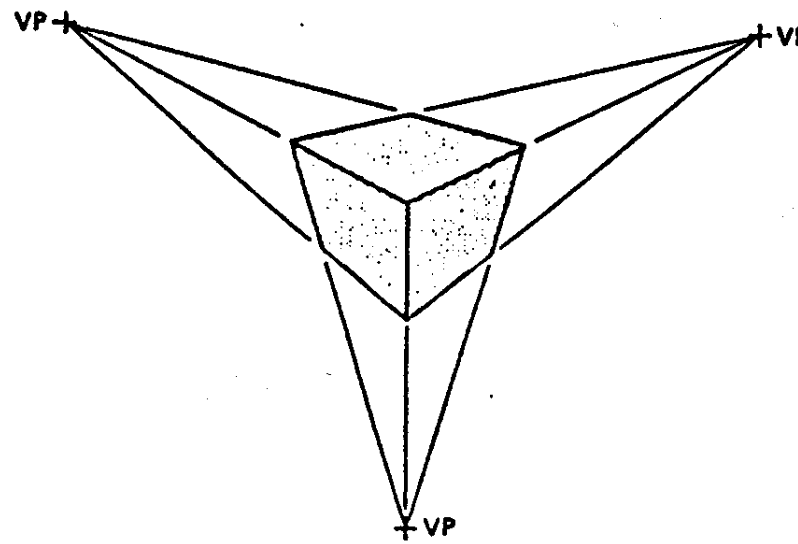
Examples of perspective projections



One point

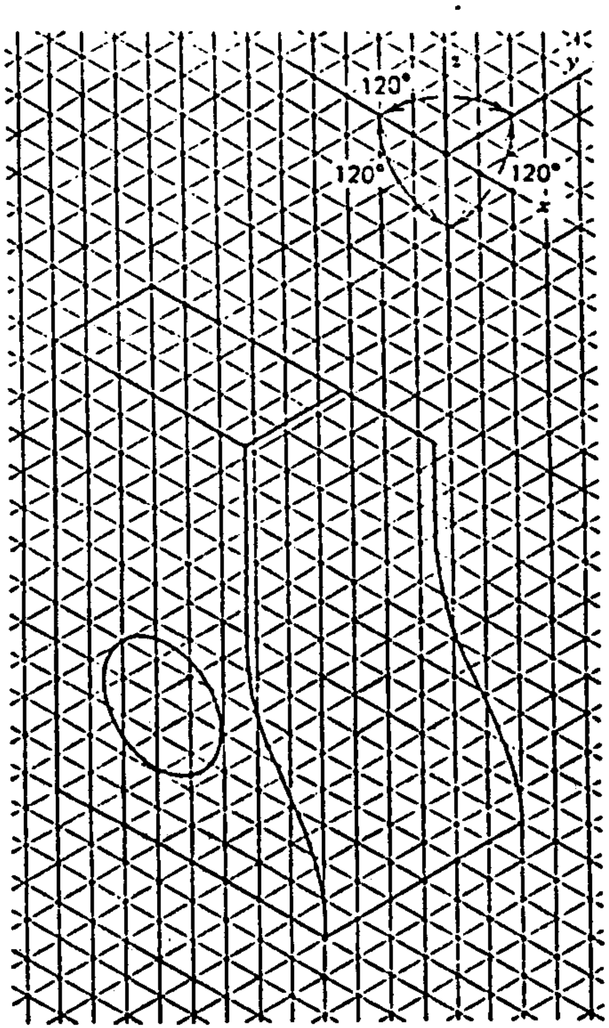


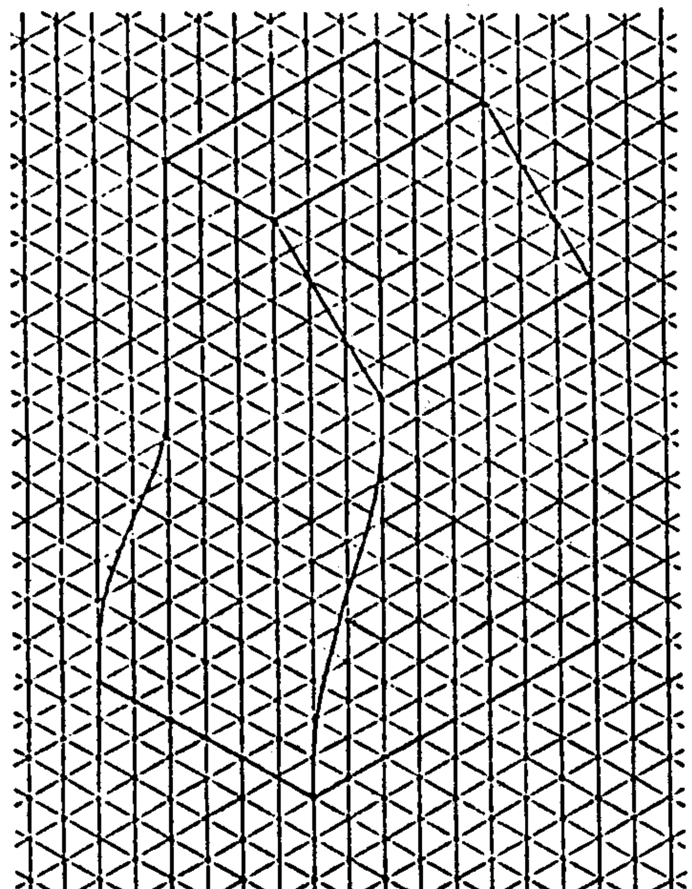
Two point



Three point

Grid to sketch an isometric projections





Lecture notes compiled by Profs. Surjunhatha and Dave

Deriving matrices for the Projections

- All the projections can be handled mathematically by the use of homogeneous transformation matrices.
- The transformation is then applied to every point of the solid to obtain a corresponding two-dimensional point of projection.
- The simplest transformation matrix is that of multi-view projection. For example, projection onto X-Y plane will be obtained by the following matrix (also called the Plan-View):

$$T_{plan} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Elevation and Side-View

- While plan-view is easy to obtain, the elevation and side-view are slightly more involved. For the elevation, first the projection of the model is taken onto the as after projecting onto the YZ plane, the projected image must be somehow transformed to the XY plane, as normally, that is the plane that the computer screen corresponds to.
  - Another way to look at this is that the model should be rotated so that its front is now a "top" and then the Plan-View of that may be taken.
- The analysis for the Side-View is similar, and the combined effect for these cases can be captured for example by the following matrices.

$$T_{elevation} = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad T_{side-view} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

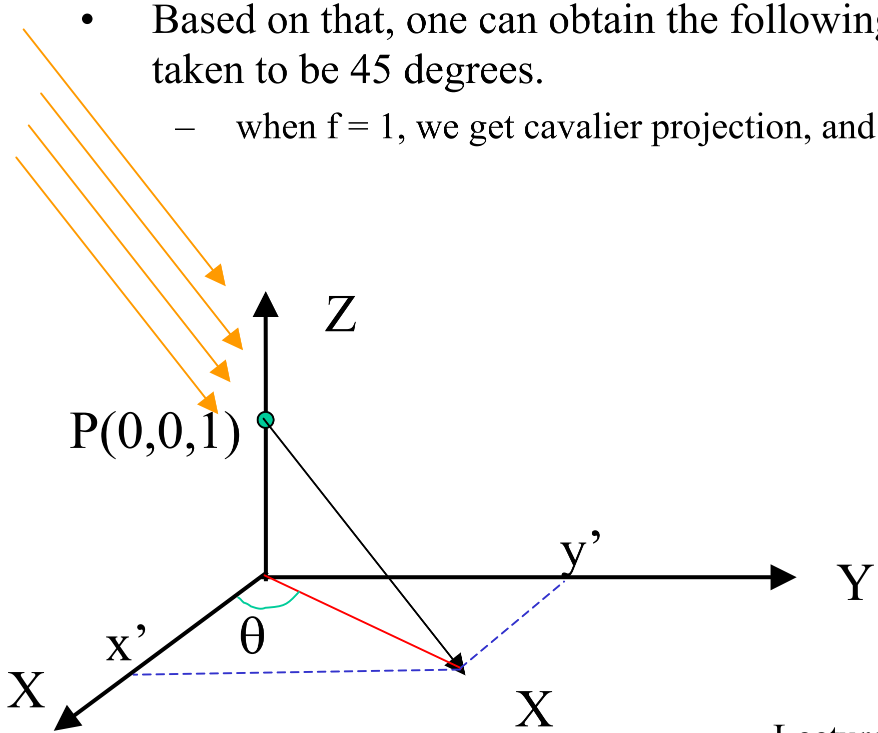
Isometric, trimetric & dimetric projections

- Other types of orthographic projections are obtained by carrying out by performing two rotations and then projecting onto the X-Y plane. For example, the isometric, dimetric and trimetric projections can be obtained by performing the following three steps.
  - 1) Rotate about X axis by angle, θ,
  - 2) Then rotate about Y axis by angle, ϕ, and
  - 3) Project onto X-Y plane.
- The composite matrix is shown below. The values of θ and ϕ are chosen so that one can obtain isometric, dimetric or trimetric projections. For instance, in an isometric projection, unit vectors in x,y and z directions will have equal projected lengths. That condition gives two equations for two unknown angles θ and ϕ. Similarly for a dimetric projection, two of the three unit vectors will have equal projected lengths, and thus that condition gives us one equation only. Therefore, only one of the two angles can be solved for and the other angle may be chosen arbitrarily.

$$T_{composite} = \begin{vmatrix} \cos\phi & \sin\theta\sin\phi & \sin\phi\cos\theta & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Oblique Projections

- In an oblique projection, the rays are parallel, but at an angle other than 90 degrees to the plane of projection. Hence unlike the case of Plan-View where the matrix was very simple, one needs to derive the transformation matrix for the case of oblique projection.
- Let us consider a point P(0, 0, 1) and see how its oblique projection falls on the X-T plane.
  - First, let us take oblique parallel rays (shown in orange color), and see where the "shadow" of point P will fall – call it P'. Let this shadow (shown as a red line, assuming it is a shadow of a stick 1 unit high at the origin) have a length f, and make an angle θ as shown with the X axis. In that case, the points x' and y' for the shadow will be: x' = f cosθ; and y' = f sinθ
- Since the point (0,0,1) has a projection (x', y', o) as shown above, any other point (x, y, z), will have a projection: x' = z f cosθ + x; and y' = z f sinθ + y.
- Based on that, one can obtain the following matrix; where θ can be arbitrary, but normally it is taken to be 45 degrees.
  - when f = 1, we get cavalier projection, and for f = ½, we get cabinet projection



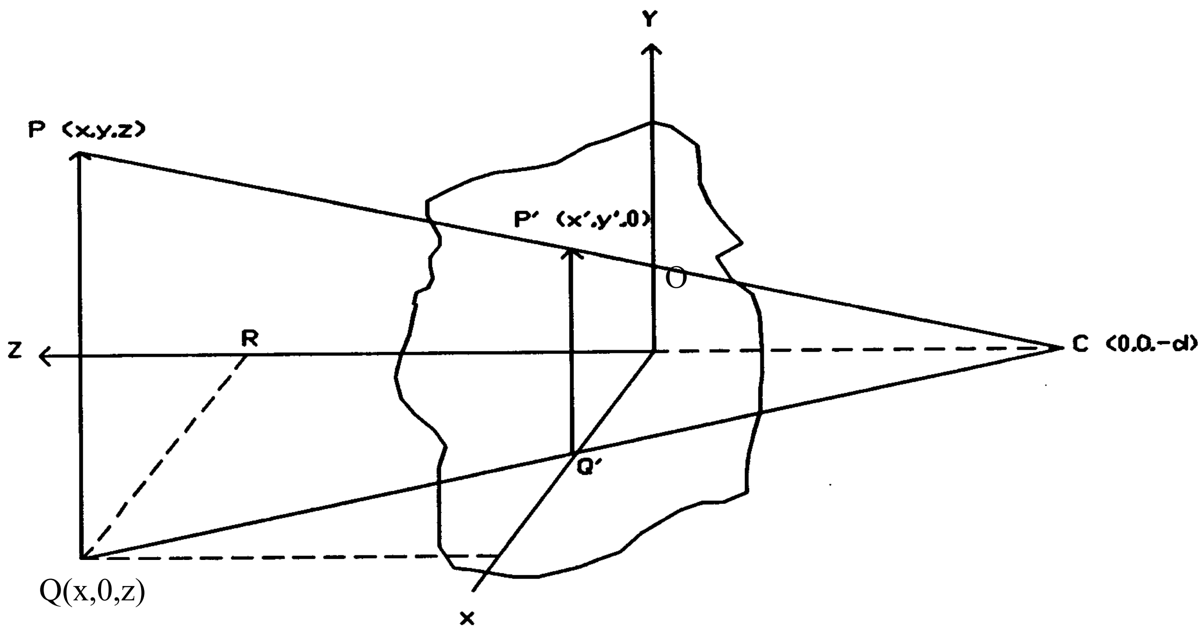
$$T_{oblique} = \begin{vmatrix} 1 & 0 & f\cos\theta & 0 \\ 0 & 1 & f\sin\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Concatenated Transformations: Review Questions

1. Consider a task of converting from a cylinder to a truncated cone. Let the center of the mass of the cylinder be positioned at point (a,b,c) and its center axis be parallel to the z axis. The nominal dimensions are; height h, and radius r. The nominal dimensions of the truncated cone are; height h1, radius of the top surface, r1 and radius of the bottom surface, r2. For this transformation, derive equations for obtaining new (x,y,z) coordinates from the old ones such that the transformed objects center of mass is at (p,q,r), and the cone axis is aligned with the x axis. Is it possible to develop a 4X4 transformation matrix for this operation? Explain your answer.
2. Derive a 4X4 homogeneous transformation matrix to perform an isometric projection. Show all the steps clearly.

Perspective Projections

- In a perspective projection, the rays are not parallel, and they originate from a single point of source, which is usually the location of the viewer. To derive the transformation matrix for the case of perspective projection, the following diagram and analysis are required.
- The center of projection (point C(0,0,-d)) is taken at a distance d from the origin in the negative Z direction as shown below, and the object is a "stick" of height y, its upper tip is located at point P(x,y,z) and, its base is at point Q(x,o,z) as shown.
- By connecting point P to C and point Q to C, one can find the projections on the XY plane as P' and Q' respectively.
- It is clear that the coordinate of P', x', y' can be computed using similar triangles PQC and P'Q'C (for y') and RQC and OQC (for x') and provide:
- x' = x(d/(z+d)), and y' = y(d/(z+d)). Note that z' = 0 in XY plane.
- How do we convert this to a homogeneous transformation matrix?

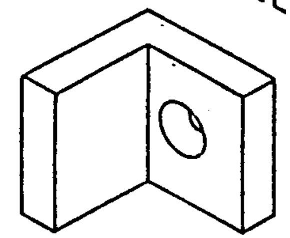


Perspective Projections (contd.)

- Note that the derivation in the previous slide results only in a single vanishing-point perspective. If 2-pt or 3-pt perspective are required, then the perspective transformation matrix need to be multiplied by appropriate one or two rotation matrices.
- As a reminder: x' = x(d/(z+d)), and y' = y(d/(z+d)). Note that z' = 0 in XY plane.
- How do we convert the information above to a homogeneous transformation matrix?
- Can you verify that this matrix accomplishes what we need?

$$T = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \dfrac{1}{d} & 1 \end{vmatrix}$$

Homework

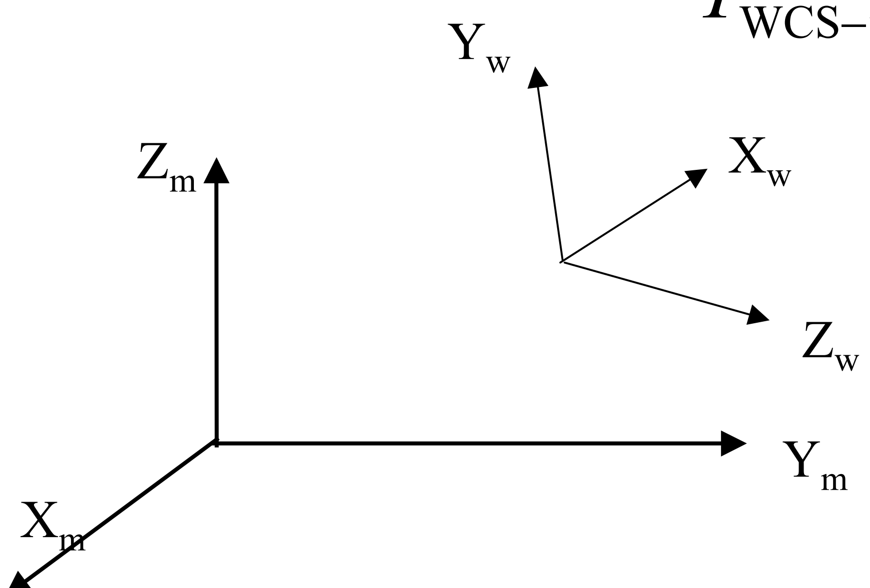


- If you were given a choice between a CAD package that only had a CSG technique along with a set of primitives such as block, cylinder, cone, and sphere and a CAD package that only had a sweep technique with extrude and revolve operations, which one will you prefer? Why?
- How does the technique called *Cell Decomposition* compare with the CSG technique? Does it have any practical applications? Is it available in Pro/E?
- Sketch a CSG binary tree for the part above. Asume you only have blocks and cylinders as primitives.
- What is the major solid modeling technique that Pro/E is based on?
- Discuss advantages and disadvantages of the *Spatial Occupancy Enumeration* technique. Is this method available in Pro/E.
- Reflection of a line joining points $P_1(1,1)$, $P_2(2,4)$ is to be carried out about line joining (2,2) and (5,5). Derive a 4X4 homogeneous transformation matrix which can perform this operation, and compute the new coordinates of P_1 and P_2.
- Derive a 4X4 homogeneous transformation matrix that will transform a unit cylinder (r=1, H=1) with CG at location (x1,y1,z1), and its axes parallel to z axis to a cylinder with radius R, and height H with its CG at location (x2,y2,z2) with its axis parallel to x axis.

Transforming from one coordinate system to another

- Model Coordinate system vs Working coordinate system
- The transformation matrix between a point (x, y, z) in the working coordinate system to the model coordinate system is compose of the following, where $\mathbf{i}_w$, $\mathbf{j}_w$ and $\mathbf{k}_w$ are unit directional vectors of the WCS system in terms of the MCS system, and the vector $\mathbf{p}_w$ represents the position of the origin of the WCS in terms of the MCS:

$$T_{\text{WCS-to-MCS}} = \begin{vmatrix} i_{wx} & j_{wx} & k_{wx} & p_{wx} \\ i_{wy} & j_{wy} & k_{wy} & p_{wy} \\ i_{wz} & j_{wz} & k_{wz} & p_{wz} \\ 0 & 0 & 0 & 1 \end{vmatrix}$$



Review Question

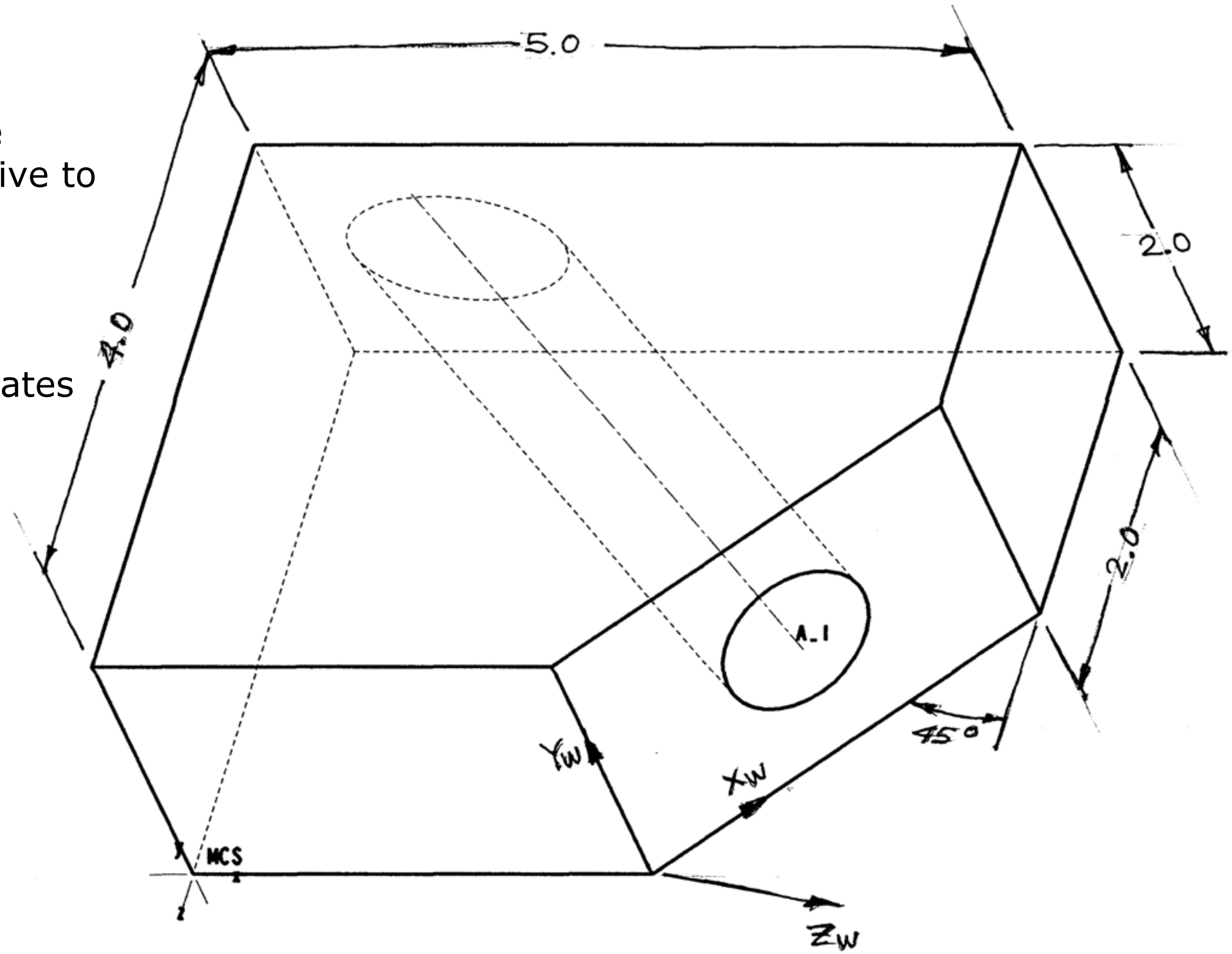
For the object as shown in the figure below, the Model Coordinate System (MCS – X,Y, and Z axes) is located on the lower left corner of the front surface. The Working Coordinate System (WCS – X_w, Y_w, and Z_w axes) is on the lower left corner of the inclined surface. The center of the hole (shown as ellipse) is the center of the inclined plane. The transformation equation can be written as

$$P = [T]\, P_w$$

a) Find the coordinates of the center of the circle (hole) relative to WCS.

b) Find the transformation matrix [T].

c) Find the coordinates of the center of circle (hole) measured in the MCS.

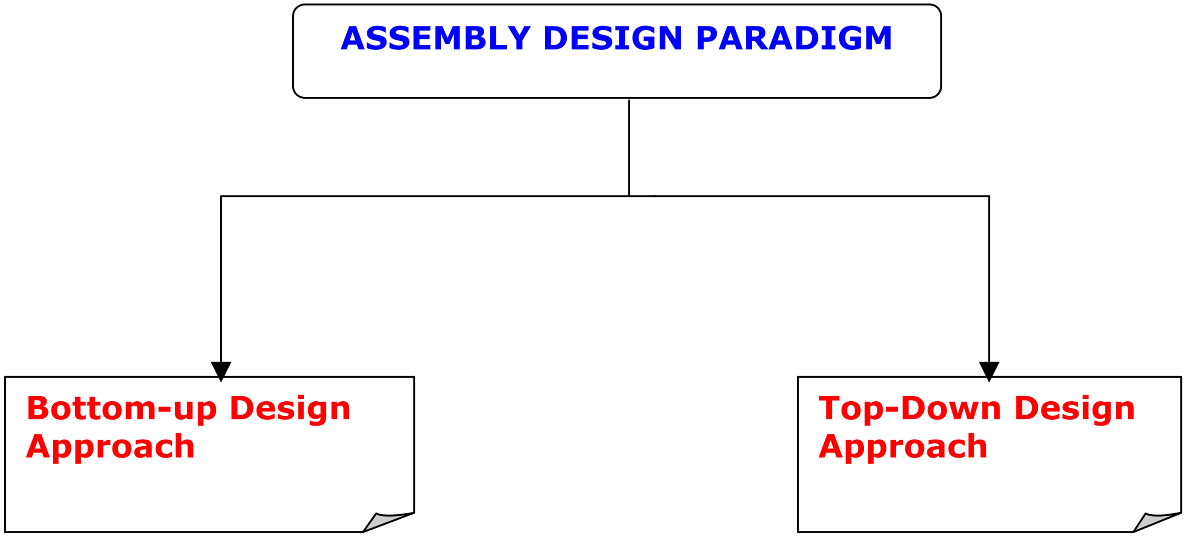


Assembly Modeling

- The assembly model is needed for
  - Interference detection between parts
  - Motion simulation
  - Constraint satisfaction
  - Evaluation of the ability to be assembled
  - Assembly manufacturing planning
- Assembly design is based on the following considerations
  - Kinematics
  - Ability to interchange of parts
  - Geometric arrangement of components to produce compact package
  - Ability to assemble and disassemble
  - Collisions and interference
  - Tolerance allocation to produce the proper quality function

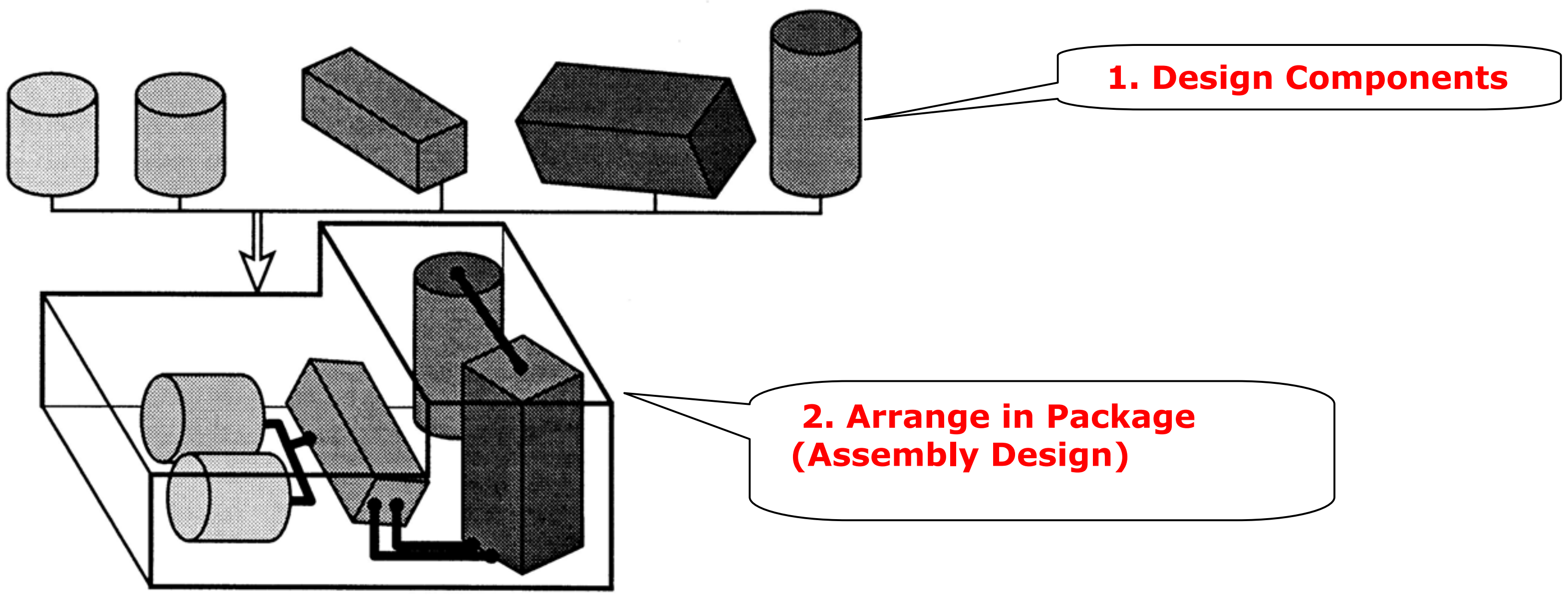
Design Strategy for Assembly

- Design is an iterative process, hence many modifications and alterations may be required before the design is finalized
- For example, after modeling a piston-cylinder assembly in internal combustion engine, if the designer changes the cylinder size, the modeler (i.e. the software) should be able to validate, determine, and propagate the rest of changes needed to make the assembly still valid.



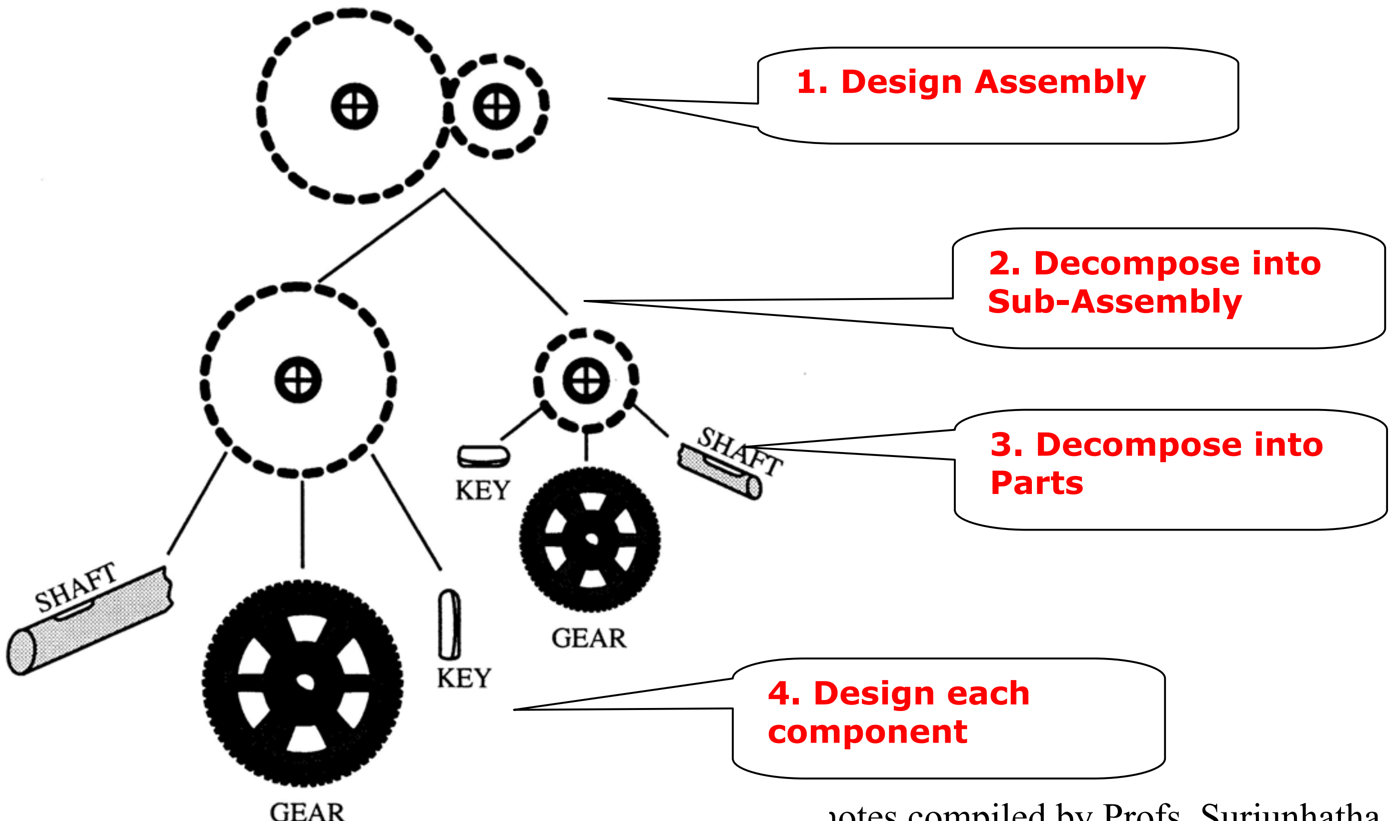
Bottom-up Design

- This is a widely used approach in commercial modelers
- The assembly components are modeled as separate parts without referencing other assembly components. This means if the other assembly components change, the changes are not propagated to the assembly component. The changes need to be made in a number of places
- It requires detailed design of all the constituent parts and sub-assemblies before laying out the design for the assembly



Top-Down Assembly Design

- Top Down Design is a method of designing something that starts with the complete item then breaks it down into smaller and smaller components.
  - More natural way to design assemblies.
  - The designer begins with an abstract concept and recursively divides it into logical sub-assemblies until the level of parts is reached. Conceptual design of assemblies.

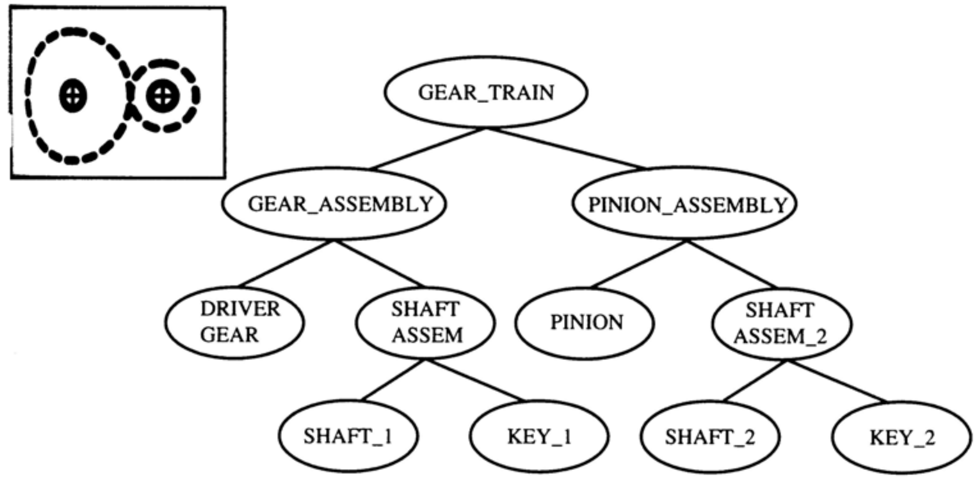


Top-Down Assembly Design

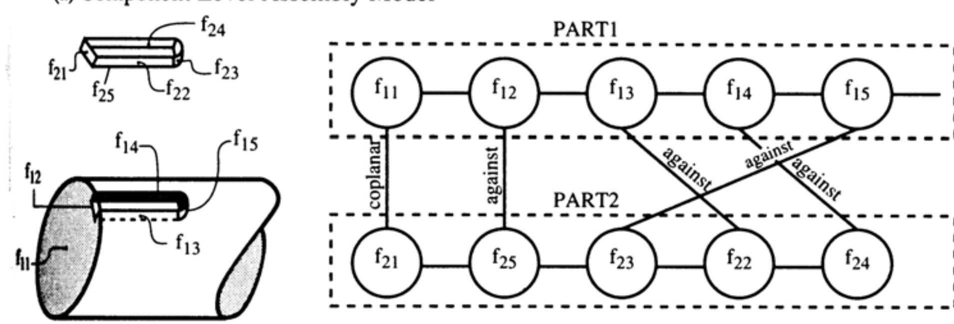
- A conceptual assembly of the gear pair is designed first, determining pitch diameters, number of teeth, and rpms.
- Then, it is decomposed into subassemblies and parts. So steps are:
  1. **Plan** – Think carefully and plan your assembly.
  2. **Define what is known** – Capture certain requirements that are known before you start.
  3. **Create a hand sketch of the assembly skeleton**. Before starting the assembly model, create a simple hand sketch. Outline during an initial design review ideas and functions that should be included within the design. Document how the system should move.
  4. **Create the structure** – Define the major subassemblies and skeleton parts.
  5. **Review and capture the design intent** – Complete and test the assembly skeleton. Insure the features are named and the assembly motion works as intended.
  6. **Create and mate the subassemblies and assembly components** – Create the subassembly component using the skeleton to define and mate the components.

Assembly models at three abstraction levels

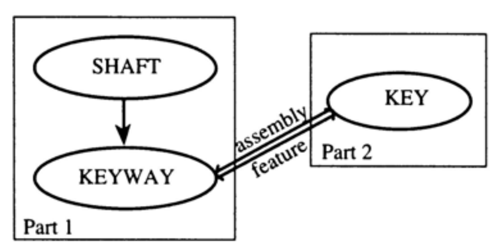
- Component Level
- Geometry Level
- Feature Level



(a) Component Level Assembly Model



(b) Geometry Level Assembly Model



(c) Feature Level Assembly Model

Geometry Level Assembly

- Mating conditions should be assigned as the relationship between geometric entities in assembly
- Transformation matrices are required for this task

Mating conditions

Against:	Two faces against each other, normal pointing in opposite directions.
Align coplanar	Two faces aligned to lie in the same surface
Co-axial:	Two axes aligned to lie in the same straight line
Coincident:	Two points constrained to be coincident

Feature Level Assembly

- Related to above example, the five geometric constraints in geometric level assembly (b) are replaced by a single assembly feature in (c).