

Dugan Um

Solid Modeling and Applications

Rapid Prototyping, CAD and CAE Theory

 Springer

Solid Modeling and Applications

Dugan Um

Solid Modeling and Applications

Rapid Prototyping, CAD and CAE Theory



Springer

Dugan Um
Texas A&M University
Corpus Christi, TX, USA

ISBN 978-3-319-21821-2 ISBN 978-3-319-21822-9 (eBook)
DOI 10.1007/978-3-319-21822-9

Library of Congress Control Number: 2015945947

Springer Cham Heidelberg New York Dordrecht London
© Springer International Publishing Switzerland 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

As the rapid introduction of new designs to the market becomes the key success factor in modern industry, demands arise for lessons in solid modeling and applications. Conventional drawing tables are replaced by CAD and CAE technology, while manual machines are upgraded with more flexible and numerically controlled systems on the shop floors. This book addresses scope of lessons primarily for design engineers involved in the disciplines from product design, analysis, and validation. Theoretical backgrounds introduced in this book will help students understand operational knowledge of CAD, CAE, and Rapid Prototyping technology, so that engineers can operate or develop design tools in a more efficient manner. Theoretical outlines as well as mathematical examples introduced in the book will help students understand the concept of each theory up to the level of practical use in real-world applications. The general audiences are mechanical or manufacturing engineers with little or no coding experience for applications of theory, but with limited spreadsheet experiences. The book is designed to enable students to understand and apply theories to practical applications. Therefore, the focus of theories and illustrations in each chapter are prepared to help maximize learning experiences from understanding to practical applications.

Corpus Christi, TX, USA

Dugan Um

Contents

1	Introduction to CAD	1
1.1	Computer Aided Design	2
1.2	Design Process	6
1.2.1	Pahl and Beitz’s Approach	7
1.2.2	Ohsuga’s Approach	9
1.3	Applications of Design Models	11
1.4	Examples by CAD/CAE	12
1.4.1	Disc Rotor	12
1.4.2	Scissor Jack	14
1.4.3	Automotive Rocker Arm	15
2	Graphical Representation for Mechanical Design	17
2.1	Mongian Projection	18
2.2	ANSI Y14	19
2.2.1	Line Style	20
2.2.2	Sectional View	21
2.2.3	Orthographic Projection	21
2.2.4	Pictorial Projection	23
2.3	Tolerance Basics	27
2.3.1	Datum Plane	28
2.3.2	Hole and Shaft Tolerance	32
2.3.3	Geometric Tolerances	40
2.4	Surface Texture	45
	References	49
3	3D Geometric Modeling	51
3.1	Coordinate System	52
3.2	Description of Frame	54
3.3	Mappings	55
3.4	General Transformation Mapping	59
3.5	Transformation Arithmetic	61

3.6	General Form of Rotation	62
3.7	Transformation of a 3D Model	66
3.8	Perspective Projection	69
3.9	3D Modeling Schemes	73
3.9.1	Wireframe Geometry	73
3.9.2	Surface Representation	75
3.9.3	Solid Modeling	79
	References	92
4	Parametric Line and Curve Theory	93
4.1	Data Structure	94
4.2	Parametric Line	95
4.3	Cubic Spline Curve	97
4.4	Bezier Spline Curve	112
4.5	Surface Theory	120
4.5.1	Bilinear Surface	122
4.5.2	Ruled Surface	126
4.5.3	General Curved Surface	130
5	Miscellaneous Issues in Computer Graphics for Modeling	143
5.1	Basic Raster Graphics Algorithms for Drawing in 2D	144
5.1.1	Scan Conversion	144
5.1.2	The Basic Incremental Algorithm	145
5.1.3	Circular Interpolation Using DDA	148
5.2	Hidden Line Removal	152
5.2.1	Polygon Filling Algorithm	153
5.2.2	Visible Surface Testing	155
5.2.3	Z-Buffering Algorithm	156
5.2.4	Polygon Clipping	159
5.2.5	Z-Clipping	162
	References	169
6	Rendering Theory	171
6.1	Color	172
6.1.1	How the Eye Determines Color	174
6.1.2	The Color Matching Experiments	174
6.1.3	A Cousin Color Space	176
6.1.4	The CIE x - y Chromaticity Diagram	177
6.2	Color Display	178
6.3	Dithering	179
6.4	Light Illumination Models	182
6.4.1	Gouraud Shading	183
6.4.2	Phong Shading	184
6.4.3	Other Approaches	188
6.5	Rendering for Shading by Shadow	188
	References	190

7	Rapid Prototyping	191
7.1	Definition	192
7.2	Applications	194
7.2.1	Prototypes for Design Evaluation	196
7.2.2	Prototypes for Function Verification	197
7.2.3	Models for Further Manufacturing Processes	199
7.3	Rapid Prototyping Processes	201
7.3.1	General Principle	201
7.3.2	Specific RP&M Processes	202
7.3.3	RP Machine Trend	209
7.4	Data Structure	212
7.5	Physics Behind SFF	215
7.6	Post-Processing	219
	References	220
8	Finite Element Modeling and Analysis	223
8.1	What Is FEM?	223
8.2	Automatic Mesh Generation	226
8.2.1	Node Generation	227
8.2.2	Mesh Generation	230
8.2.3	Improvement of Mesh Quality	235
8.3	What Is Truss?	236
8.3.1	Matrix Approach in FEM	237
8.3.2	Force–Displacement Relationship	238
8.3.3	Stiffness Matrix for a Single Spring Element	240
8.4	How to Develop Governing Equations?	241
8.5	Example of a Spring Assemblage	243
8.6	Boundary Conditions	245
8.6.1	Homogeneous Boundary Condition	245
8.6.2	Nonhomogeneous Boundary Condition	248
8.7	Assembling the Total Stiffness Matrix by Superposition (Direct Stiffness Method)	249
8.8	Development of Truss Equation	253
8.8.1	Derivation of the Stiffness Matrix for a Bar	254
8.8.2	Transformation of Vectors in Two Dimensional Space	259
8.8.3	Global Stiffness Matrix	260
8.8.4	Computation of Stress for a Bar in x - y Plane	263
8.9	Solution of a Plane Truss	266
	References	286
	Appendix A: Tolerance Classification	287
	Appendix B: Surface Finish Symbols	291
	Index	293

Chapter 1

Introduction to CAD

The Big Picture

Discussion Map

You need to understand terminology and design mechanism with Computer Aided Design.

Discover

Understand basic concept of CAD and benefits of CAD.

Understand terminologies used for modeling technology and CAD.

Understand why designers prefer using CAD compared to manual drawing.

Understand design as a process.

Understand components of CAD system.

CAD is an acronym used for Computer Aided Design, while CAE is for Computer Aided Engineering. CAD is often used for drawing aspects, while CAE is used for analysis aspects. Thanks to advanced analysis software embedded in the most of CAD packages, they are often used together as CAD/CAE. Modern design engineers are likely to use a type of CAD system such as ACAD, Pro-E, Solidworks, or TurboCAD, etc. Although many pros and cons exist in each CAD system, all of the design tools are made with fundamentally similar concept in mind: Help designers facilitate to bring their ideas into reality. It is true to say that it takes time to get used to a CAD system. However, once a designer obtains knowledge and knows how to handle a CAD package, it is more or less the same as a plot understanding aerodynamic principles of a plane. In addition, understanding design as a process is another important aspect to an effective designer. Although design requires trial and error, in general, a good understanding of design as a process will help minimize trial and error cycles and will make a final product more efficient from

manufacturing stand point. In this chapter, our study will be focused on design fundamentals, CAD, CAE, and design processes.

1.1 Computer Aided Design

Design is an art. The most important ingredient in art is creativity. The idea of a new design is in our mind, but it has to be brought out for presentation. Traditionally, it is done by manual drawings using tools such as drafting tables, technical tools, templates, etc. Now in twenty-first century, advanced technologies including computers, man-machine interface devices, and sophisticated software made it all changed to a new paradigm, called Computer Aided Design. Thanks to the advancements in CAD technology, design is expanding its scope to CAE, by which a complex analysis such as stress, thermal, fluid, or dynamic analysis are all possible in a package such as Multi-Sim, ANSYS, or Abacus. In this section, we introduce CAD to discuss about its usefulness, applications, as well as benefits over the manual hand drawings, followed by examples generated by a modern CAD package. Definitions and terminologies introduced in this section will be used further in the later chapters. Therefore, a good understanding of the overall concept of CAD will be beneficial for understanding the rest of the chapters in this book.

Definition: CAD is defined as creation and manipulation of pictures (design prototypes) on a computer to assist engineers in design.

In order to facilitate creation and manipulation of 2D or 3D pictures, CAD provides various geometric models such as patterns, symbols, and diagrams, which are all fundamental elements for CAD. Primarily, geometric models are used for representation of products to realize abstracted ideas in designer's mind and to use for evaluation purposes. In a modern design process, modeling by CAD, in general, is followed by analysis by CAE for design validation. Depending on the purpose of design validation, the same design can be represented in different geometric models. For instance, in Fig. 1.1, the basic design of the connecting rod is represented in two different geometric models. The representation on the left is a simplified kinematic model for statics or dynamics analysis, while the representation on the right is a model for FEM (Finite Element Method) for stress/strain analysis.

By using such models, a designer can not only represent ideas, but also can communicate with other designers to share ideas and exchange details about the product. Since the geometric models allow communication between designers, geometric models are often called language of designer. Even if designers create

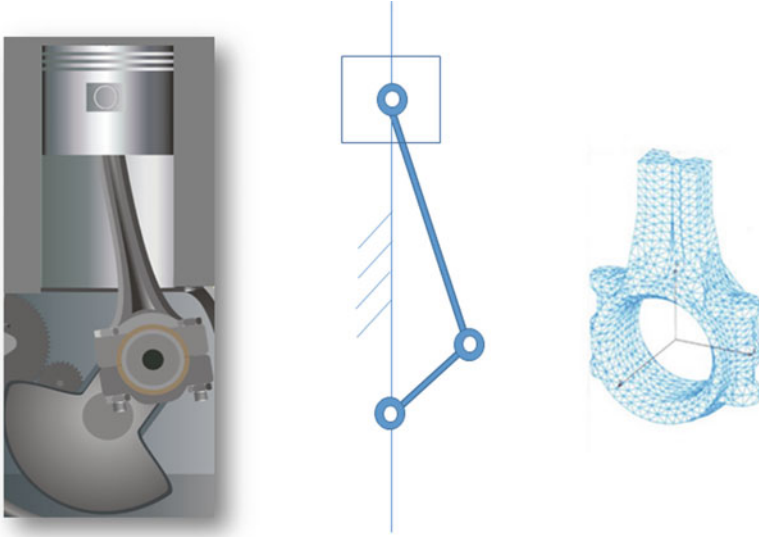


Fig. 1.1 Geometric models change depending on the purpose of analysis

designs with different CAD packages, it can be interpreted easily between designers since the geometric models are all standardized in engineering discipline. Therefore, geometric models are great tools to share design ideas between participants in a design process.

Important Lesson (Two Main Goals of Geometric Models)

- Geometric models of design
 - Patterns, symbols, diagrams
 - Language of designer
- Importance of modeling
 1. Representation
 - Realization of an abstracted ideas
 - Evaluation
 2. Communication
 - Share ideas and designs between participants in a design process

Next question that may come to our mind is that “why CAD?” then. Although designers can achieve two main goals with hand-drawn pictures and diagrams, CAD facilitates many aspects of design process. First of all, CAD can increase the productivity of the designer throughout the entire design process. It helps

conceptualize the ideas and bring it to reality. In addition, thanks to CAE, it will reduce the time for analysis. Second, CAD can improve the quality of the design with more complete analysis ability. Thanks to expedite and elaborate design tools that CAD can provide, designers can test more alternatives to find the optimum solution to help customers' needs. Third, the quality of documentation is improved with better graphics quality, more standardization, and fewer drafting errors. Recent CAD packages mostly provide error checking capability of the final drawing for overlapping, tolerance verification, and grammar and symbolic usages. Forth, the majority of modern CAD packages also provide the ability of creating a manufacturing database directly from the design. Manufacturing data such as BOM (Bill of Material), geometric specifications, dimension of components, and even material specifications can be directly created for futuristic use in manufacturing process. Finally, CAD offers various functions to facilitate the entire drafting process with geometric modeling database, engineering analysis, and design review capabilities, and even automated drafting functions in some advanced CAD packages.

All the aforementioned functions that CAD can provide are the answers to the question of "why CAD?". In Table 1.1, the conventional method of manual drafting is compared to CAD. As is shown in the table, the conventional method still serves engineers for many products. Most of the basic principles of manual drafting have been adopted in modern CAD packages. However, there are numerous benefits that CAD can offer. For instance, in Fig. 1.2, there are six different methods of drawing a circle in CAD as opposed to only one method of drawing a circle in manual drafting with a compass. For instance, a circle can be defined by a center with a

Table 1.1 Conventional method vs. CAD

Conventional method	CAD
Served engineer for many products from screw to building	Provide rich variety of techniques for the definition of geometry
Mongian projection can be used for a drawing as complex as an aircraft	Identical representation is used (compatible with conventional method)
Diagrams may be used to represent virtually any system	Shorten the design process significantly (concurrent engineering)
Skill is required in the construction and interpretation	Minimum skill is required for operation, but analytic skills are required
Possible to have conflicting or erroneous models	Automatic error checking in each model is possible
Hard to deal with complexity of today's products	Suitable to deal with complexity of today's products
Hard to generate further representations for assessment, manufacturing information	Easy to generate further representation
Drawings are easily misread because of ambiguity or error in the drawing or simple human error in the interpretation	High accuracy in representation and less error in interpretation
Size of the representation is constrained by the physical size of the drawing paper	No limitation in representation by size

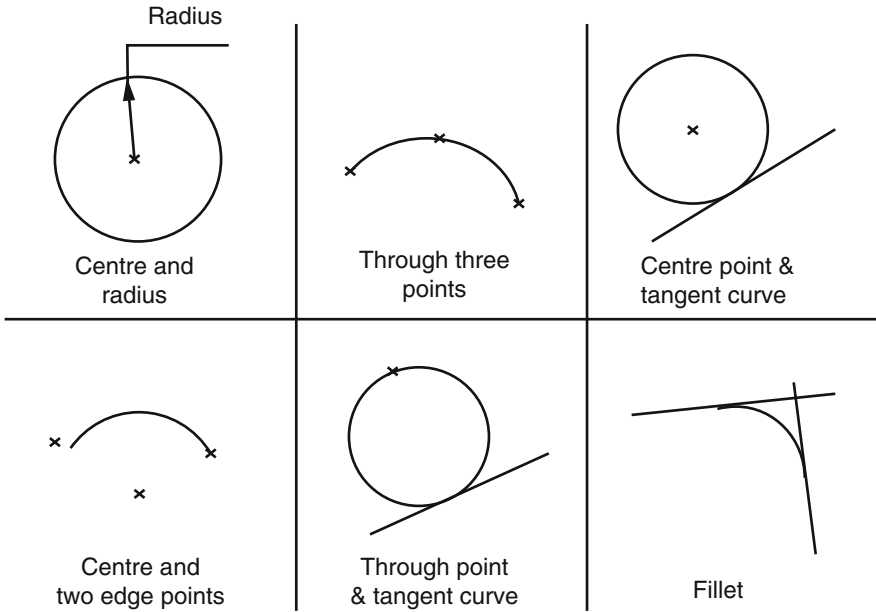


Fig. 1.2 Rich variety of techniques for the definition of geometry

radius, through three points, center with tangential line, center and two edge points, through a point and tangential line, or two fillet lines. CAD also provides means to accomplish concurrent engineering where all relevant disciplines for design and manufacturing are simultaneously involved from the earliest stage of product design so that a new product can be introduced rapidly for a higher percentage share of the market, thus yielding higher profits for the company.

Important Lesson (Why CAD?)

1. To increase the productivity of the designer
2. To improve the quality of the design
3. To improve design documentation
4. To create a manufacturing database
5. Various CAD functions

In summary, the aim of CAD is to apply computers to both modeling and communication of designs. Two levels of usage of CAD have been identified so far. At the basic level, CAD assists drawings, diagrams, and the generation of list of parts in a design. In a more advanced level, CAD provides new techniques that give the designer enhanced facilities in system engineering with the function of CAE. Rapid prototyping is often used for design validation as well as functional verification.

Important Lesson (Two Levels of Usage of CAD)

1. Basic level

- Automate or assist drawings, diagrams, and the generation of lists of parts

2. Advanced level

- Provide enhanced facilities in system engineering

1.2 Design Process

In modern design approach, design is no longer a simple drafting, but it involves other activities such as analysis, documentation, and manufacturing data generation, etc. Therefore, design is a process with multiple stages with iterations for validation of works done in each stage. Proper design process will save time with careful thinking at the early stage. Using a proper model and validation at early stage will not only save time, but also will save material and manufacturing cost. A simple, but an exemplary design process for a small scale project is outlined below.

Define

The problem has to be defined clearly and completely. The objective of the design has to also be stated. Since time, material, and skill sets are all limited for any design team, the most important aspect of the solution for a given project has to be addressed and shared in a design team.

Ideate

Once the problem is defined and the design objective is well understood, all the participants need to be involved in a brainstorming session to come up with a solution. Importance of this state is to discuss as many alternatives as possible. Examine each alternative for possible scenarios to make sure that an optimal solution can be selected.

Design

Before building parts of the design, design validation has to be done. Geometric models or kinematic models can be used to realize the solution proposed at the ideation stage. Multiple alternatives can be examined in this stage as combined activities with the Ideation stage. A CAD package can be an efficient tool for design

validation for static as well as mobile parts. The manufacturing aspect of each part can be discussed as well. Tolerance check can be done by virtual assembly in the design creation.

Manufacture

Manufacturing of the designed parts is the final stage. A proper manufacturing process has to be selected for each part. Careful selection of tools and manufacturing methods is the key to minimize the trial and error cycle in this stage.

In summary, four common steps can be sequentially walked through for a small scale project.

Define

Ideate

Design

Manufacture

The design process introduced above is suitable for a small scale project such as a capstone project at high school or at engineering college, a project for robotics competitions, or a small scale industrial project as well. Industrial design for a large scale manufacturing, in general, requires similar but a more detailed process. Two standard approaches are popular and prevailing in industry: Pahl and Beitz's proposal and Ohsuga's approach.

1.2.1 Pahl and Beitz's Approach

Pahl and Beitz proposed a sequential design process that allows revisiting each stage if needed so that feedback and modification can take place as many times as possible before finalizing the design (see Fig. 1.3). The uniqueness of the Pahl and Beitz's approach is of its sequential manner in design process so that each stage can produce specific outcomes until the final design is produced. The overall process is organized in a way that each stage has an input and an output. Each design stage can be done by an individual, or by a team. Teamwork generally results in better solution since more alternatives can be examined.

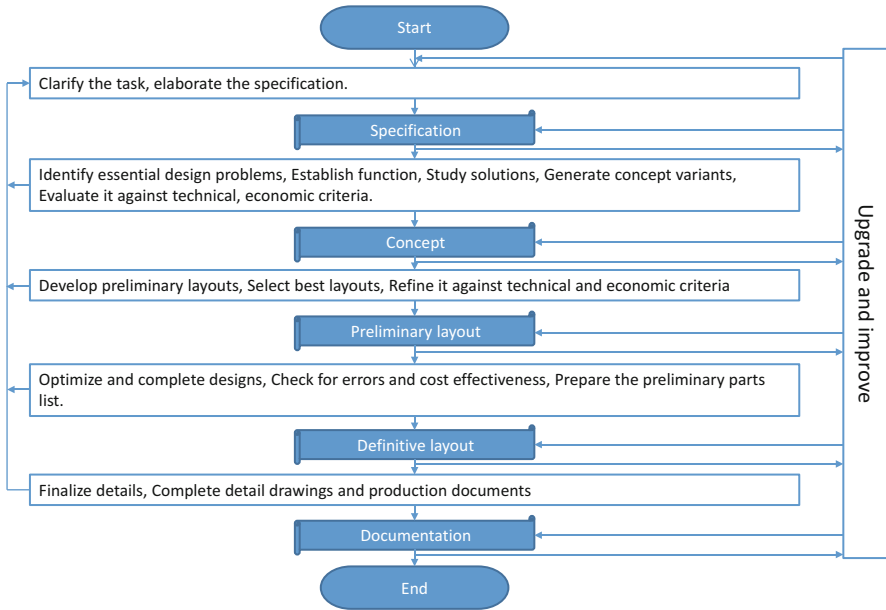


Fig. 1.3 Pahl and Beitz's approach

Specification

The first stage accepts the task as an input and clarifies the task and elaborates the specification. The output of the first stage is the specification. If an upgrade or an improvement is required on the specification, then the process can be reverted to revisit the first stage.

Concept

The second stage, taking the specification as an input, identifies the essential problems, establishes function structures, searches for solution principles, combines and firms up into concept variants, and evaluates against technical and economic criteria. The output from the second stage is the conceptual design or simply the concept. A tangible geometric design is yet to be consolidated at further design stages.

Preliminary Layouts

Once the concept is confirmed, then preliminary layouts and detail geometry have to be developed at the third stage. If several alternative layouts are developed, then the best preliminary layout has to be chosen. Finally, the best alternative has to be

refined and evaluated against technical and economic criteria. Other alternatives can be chosen if the selected layout does not meet the technical or economic criteria. In addition, if no layout can be selected, then the process can be reverted back to the first or second stage to revise the specification or concept.

Definitive layout

The preliminary layout now has to be converted into a definitive layout at the fourth stage. First, the preliminary design has to be optimized and completed with details of precise geometry. It also has to be checked for errors and for cost-effectiveness. Automatic error checking functions, if available, can be utilized. Finally, the preliminary parts list and documents have to be prepared for the definitive layout.

Documentation

At the fifth stage, the final documentation will be revealed by finalizing the details. First, the detail drawings and production documents have to be completed at this stage. Finally, all the documents have to be checked again thoroughly to minimize the probability of failures in manufacturing. Again, if it turns out that the final documents cannot serve the needs in manufacturing or customers' demand, the process can be reverted back to previous stages. However, the number of backtracking should be minimized at all costs at this stage.

In summary, Pahl and Beitz's approach is a sequential process to arrive at the final documentation.

Specification

Concept

Preliminary Layouts

Definitive layout

Documentation

1.2.2 Ohsuga's Approach

While a sequential progress of design process is natural in engineering sense, another view has been evolved by Ohsuga. Instead of viewing the design process as a streamline of sequential progression, he proposed an iteration of several

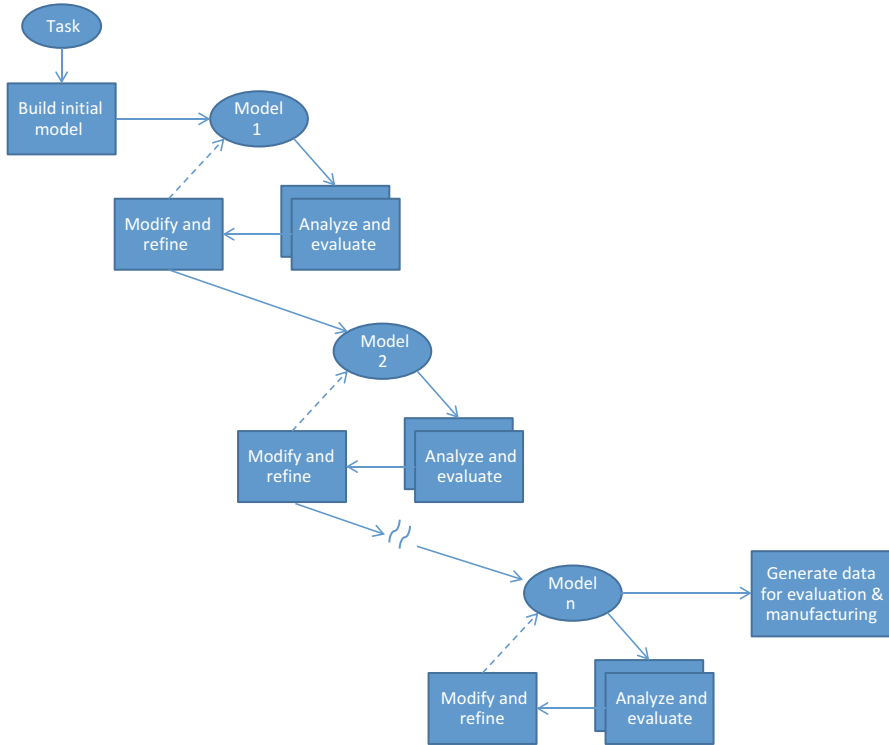


Fig. 1.4 Ohsuga's approach for design process

confined design steps until a satisfactory draft is obtained. As shown in Fig. 1.4, similar design loops that include two steps are represented in series: first, analysis and evaluation, and second, modification and refinement.

Once a preliminary model is built by the requirements, the model will be updated in all of the iterations until the final model is obtained. The uniqueness of Ohsuga's approach is that the design team can take as many iteration loops as possible until the final model is satisfactory. This is a typical trial and error approach to solve a problem if there is no immediate known solution. A solid and reliable inner step will improve the model gradually until no further improvement seems necessary. Once the final model is obtained, then the information for manufacturing data and testing data will be generated. While the Pahl and Beitz's approach is suitable for a sizable project for thorough step-by-step progression, the Ohsuga's approach may suit well for a smaller scale project since the design team can arrive at a final design in a faster pace with minimum number of iterations. Therefore, it is time-efficient, but it is more prone to errors.

We discussed two general design approaches. There are numerous design processes tailored to the needs of each entity. For example, NASA developed a circular design process where it has unique steps defined in each loop, but multiple iterations will be executed until no further improvement is necessary. PBS design squad developed a unique approach for a design process for educational purpose of children. A cyclic design process is embedded in the overall sequential process to optimize students’ thought process in their approach. In short, various combination or alternative design processes can be created to serve for specific needs in a small entity or for more general needs in a large entity with a room for tailoring certain areas for adaptation.

1.3 Applications of Design Models

In the previous section, we discussed about the geometric models, aims of CAD, and design as a process. As a result of the design process, we obtain design documents primarily for manufacturing. The importance of a geometric model is not only to generate design documentation for manufacturing, but also to transform models into other forms for evaluations during the design process (see Fig. 1.5). The design model is transformed to various models for design evaluation with the environment data, load case/duty data, as well as materials data.

As discussed in Sect. 1.1, the design model can be transformed into different models for the evaluation purposes. There are several common models for design evaluations: kinematic models, dynamic models, stress models, and thermal models (see Fig. 1.6). Each evaluation requires a different type of model for careful analysis in each area. A design engineer needs to be able to produce a specific model suitable for each design evaluation process. The evaluation of design is often called CAE,

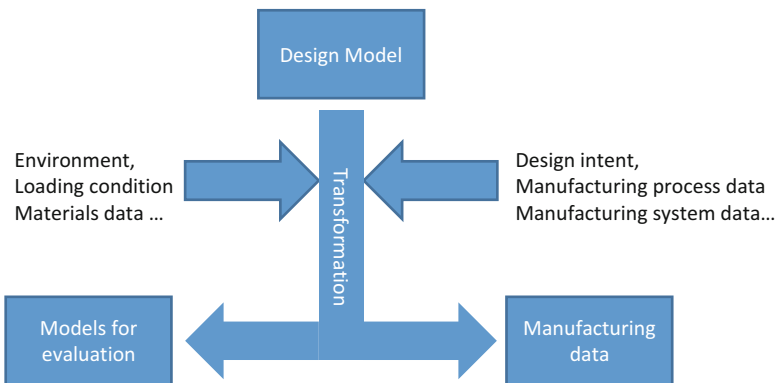
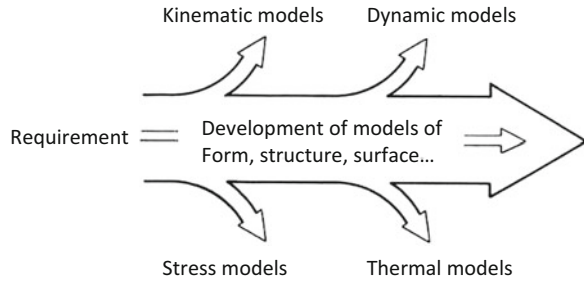


Fig. 1.5 Application of design models

Fig. 1.6 The use of models in design



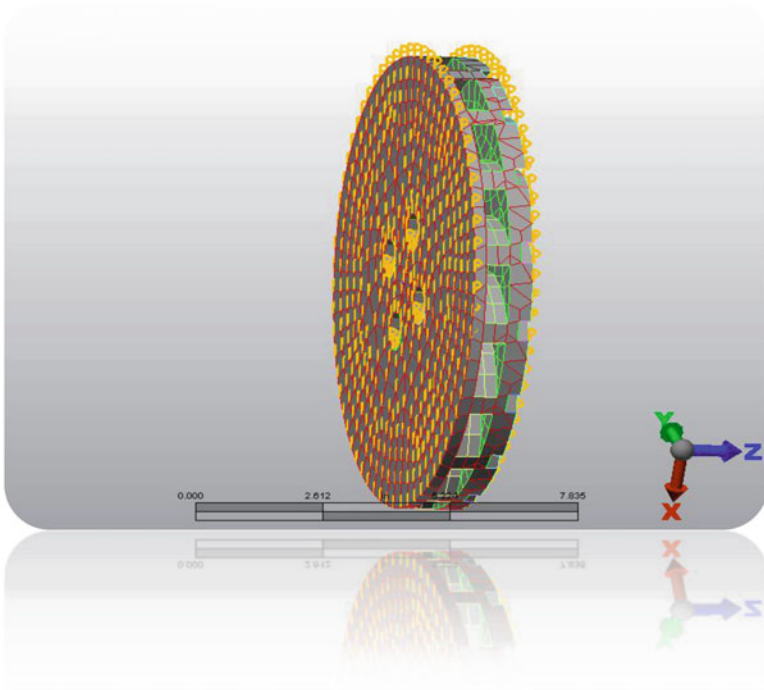
which requires not only knowledge in engineering, but also skill for system operation. Conventionally, the CAE is performed separately from the design stage with the help of a skillful analysis professional. Thanks to the advances made in CAE technology, however, both are now merged into an integrated system for faster design cycle and more alternative evaluation, thus early introduction to the market with higher satisfaction of customers is feasible. In order to transform the design model for manufacturing, the manufacturing data such as design intent, manufacturing process data, and manufacturing system data have to be complemented.

1.4 Examples by CAD/CAE

In this section, several CAD/CAE examples are introduced. The package used for these examples is Auto CAD inventor and Auto CAD Mechanical Engineering Simulator. Students may peruse this section to obtain indirect experiences of CAD/CAE exercises.

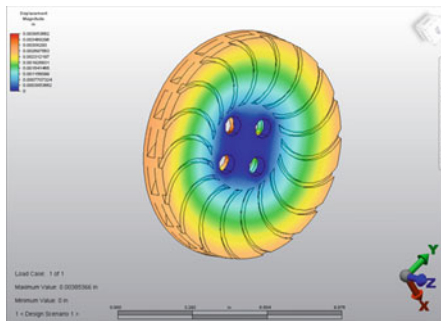
1.4.1 *Disc Rotor*

In this example, two different types of materials are compared in order to determine the best material for the specific type of a rotor disk. Upon observations, the material chosen for this application by comparing the displacement and stress analysis results is Ceramic Grade 447 Cordierite because of its endurance to stress. The load placed on each of the materials during analysis was 800 PSI applied to both sides of the part with the holes being constrained as fully fixed.

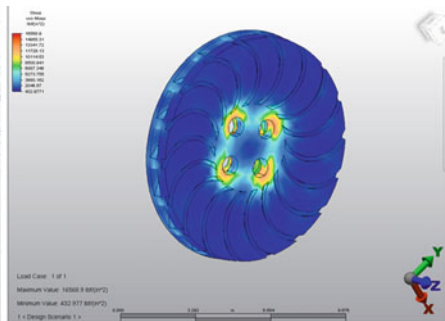


Meshed Ceramic Disc model for CAE

Color-coded analysis results of the chosen material are shown in the figures below. The CAE package can evaluate stress and strain level at each part of the product and visualize the level of stress and strain in color concentration.



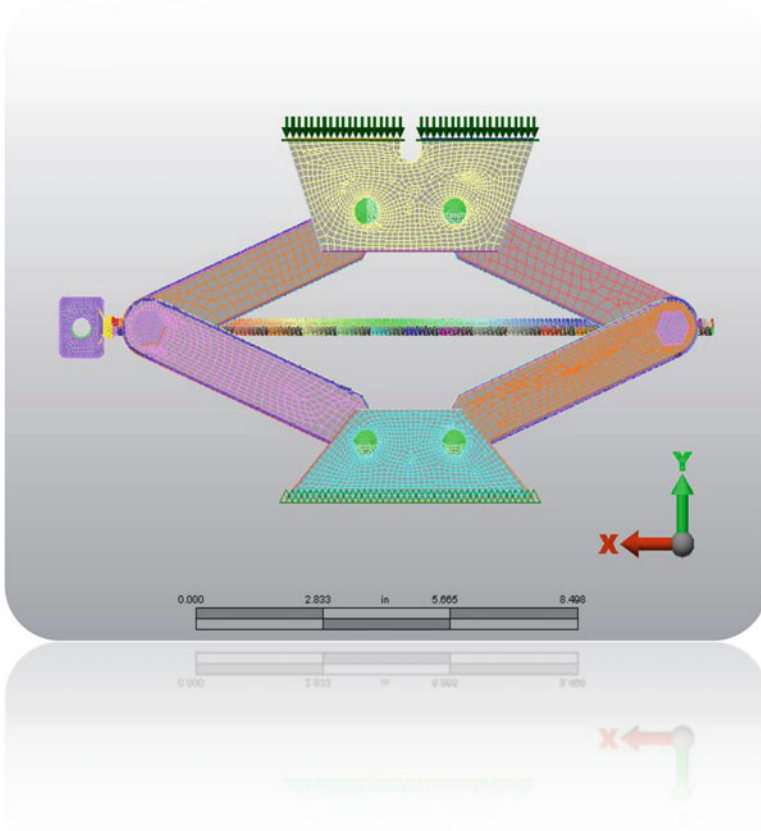
Strain analysis



Stress analysis

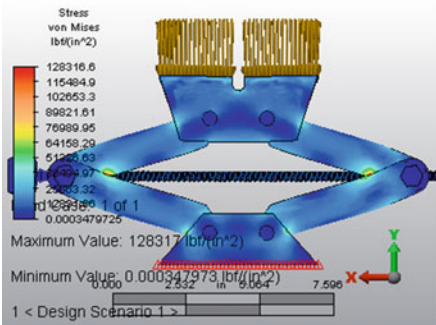
1.4.2 Scissor Jack

The purpose of this work is to compare the strengths of a car jack built using ASTM A36 Steel to one built using Grey Cast ASTM 48-A Grade 50 Iron, in order to compare the values of stress, strain, and displacement. A force of 2000 lbs was used to simulate the weight of an average-sized car. As a result of the analysis, the car jack made by steel demonstrated less deformation compared to the one made by iron.

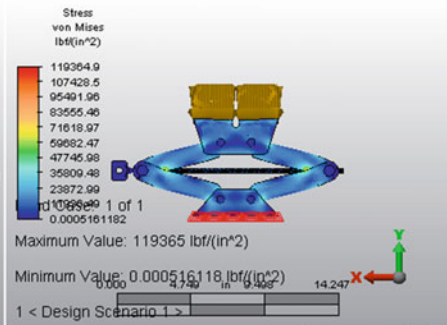


Car Jack FEM model

While iron is cheaper than steel, if we take the safety factor into consideration for design stress, the maximum stress found by simulation may fail the iron car jack. In this study, therefore, steel is chosen to make the car jack for safety and longer life span. Color-coded analysis results are shown in the figures below.



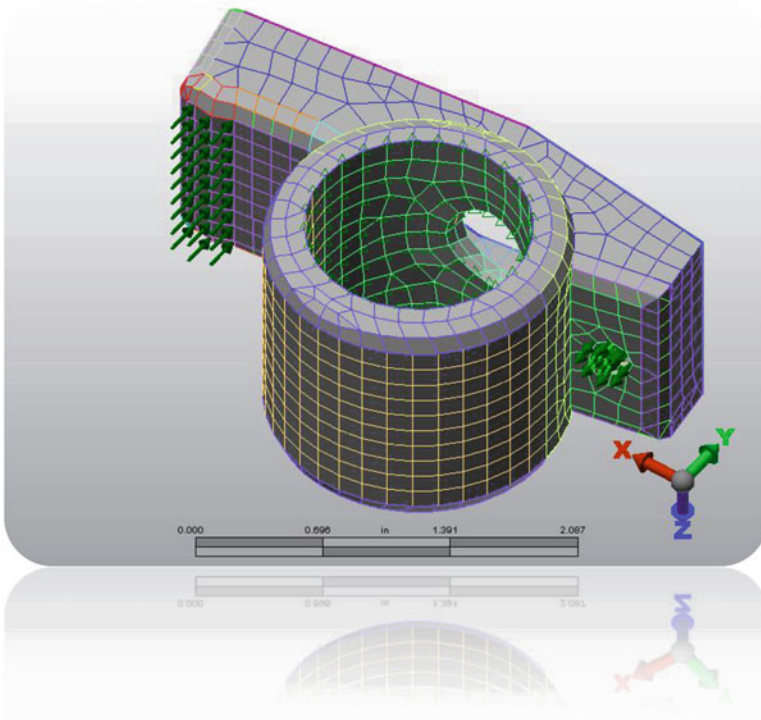
Steel: stress analysis



Iron: stress analysis

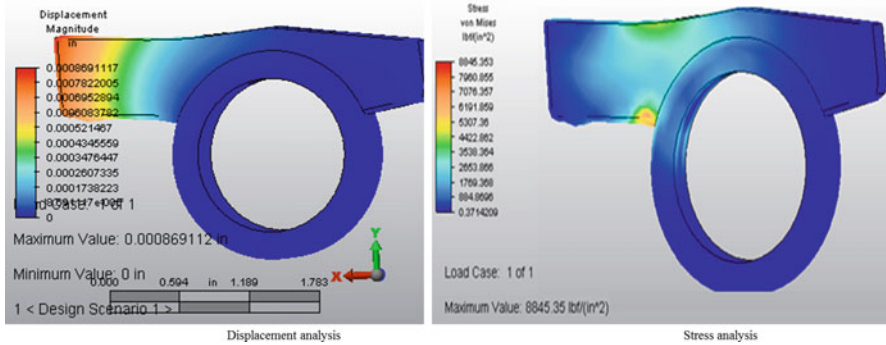
1.4.3 Automotive Rocker Arm

The purpose of this project was to discover the most economic material that would be strong enough for an automotive rocker arm. The objective is to use the strongest and lightest materials capable of withstanding the given loads.



Automotive rocker arm FEM model

The strength of a vehicle's valve train is critical not only for longevity of the motor, but also as a factor in performance. Among various materials tested by CAE simulation, it is found that Toughened Alumina (Al_2O_3 -zro) is chosen among all of alternatives because it is less expensive, easier to machine, stiffer, and lighter than the next best material. Color-coded analysis results are shown in the figures below.



Sample Exercise

Form a discussion group and brainstorm to discuss as to which design process would be more suitable for the following products. Try to come up with your own design process that would better serve the product manufacturing

1. Chopstick design
2. Power transmission shaft design for a vehicle
3. New vehicle design

Chapter 2

Graphical Representation for Mechanical Design

The Big Picture

Discussion Map

To understand geometric modeling techniques and their applications

Discover

How do we represent our product?

What types of models are available for representation?

What are the advantages and disadvantages of each model?

The desire of representing ideas of a novel design is human nature. Humans communicate with each other through various means to share ideas. As engineering becomes an important discipline, difficulty arises as to how to represent an idea of a product and how to share it with other engineers. To that end, engineers invented a drawing table and drawing tools to express their ideas. However, engineers are also challenged to invent geometric models to share ideas in mutually agreeable forms. Geometric models developed in engineering discipline enable engineers to make a significant progress in sharing ideas. The main objective in creating geometric models is to facilitate representation and interpretation of the product ideas. However, representation and interpretation of the drawings made in a two-dimensional paper often result in errors due to the lack of standard formats. Especially, in order to express a three-dimensional shape in a two-dimensional drawing paper, more caution is needed. In order to standardize the drawing procedure, various ideas of descriptive geometry have been proposed. Descriptive geometry is the branch of geometry that allows the representation of three-dimensional objects in two dimensions, by using a specific set of procedures. The resulting techniques are important for engineering, architecture, design, and in art as well [1]. The theoretical basis for descriptive geometry is provided by planar geometric projections. Gaspard Monge is usually considered as the “father of descriptive geometry.” He first developed his

techniques to solve geometric problems in 1765 while working as a draftsman for military fortifications, and later published his findings [2].

2.1 Mongian Projection

Monge's protocols or Mongian projection rules allow an imaginary object to be drawn in such a way that it appears to be a 3-D model. All geometric aspects of the imaginary object are accounted for in true size-to-scale and shape and can be imaged as seen from any position in space. All images are represented on a two-dimensional surface. Descriptive geometry uses the image-creating technique of imaginary, parallel projectors emanating from an imaginary object and intersecting an imaginary plane of projection at right angles. The cumulative points of intersections create the desired image. Below is the summary of the Monge's basic protocols.

- Project two images of an object into mutually perpendicular, arbitrary directions. Each image view accommodates three dimensions of space, two dimensions displayed as full-scale, mutually perpendicular axes and one as an invisible (point view) axis receding into the image space (depth). Each of the two adjacent image views shares a full-scale view of one of the three dimensions of space.
- Either of these images may serve as the beginning point for a third projected view. The third view may begin a fourth projection, and on ad infinitum. These sequential projections each represent a circuitous, 90° turn in space in order to view the object from a different direction.
- Each new projection utilizes a dimension in full scale that appears as a point-view dimension in the previous view. To achieve the full-scale view of this dimension and accommodate it within the new view requires one to ignore the previous view and proceed to the second previous view in which this dimension appears in full-scale.
- Each new view may be created by projecting into any of an infinite number of directions, perpendicular to the previous direction of projection. (Envision the many directions of the spokes of a wagon wheel each perpendicular to the direction of the axle.) The result is one of stepping circuitously about an object in 90° turns and viewing the object from each step. Each new view is added as an additional view to an orthographic projection layout display and appears in an "unfolding of the glass box model."

Mongian Projection

- 3D forms are represented in 2D by mapping points on the object into multiple mutually perpendicular planes of projection
- Parallel projection normal to the planes of projection
- First angle and third angle representations are most popular

Aside from the orthographic, six standard principal views (front, right side, left side, top, bottom, rear), descriptive geometry strives to yield four basic solution views: the true length of a line (i.e., full size, not foreshortened), the point view (end view) of a line, the true shape of a plane (i.e., full size to scale, or not foreshortened), and the edge view of a plane. These often serve to determine the direction of projection for the subsequent view. By the 90° circuitous stepping process, projecting in any direction from the point view of a line yields its true length view, projecting in a direction parallel to a true length line view yields its point view, projecting the point view of any line on a plane yields the plane's edge view, and projecting in a direction perpendicular to the edge view of a plane will yield the true shape (to scale) view. These various views may be called upon to help solve engineering problems posed by solid-geometry principles. Details of Monge's protocol are in ANSI Y14 (American National Standards Institute) or BS 8888 (British standard). Below is the list of ANSI Y14 that contains description of each title associated with it (Table 2.1).

2.2 ANSI Y14

Below is the list of some important aspects of ANSI 14 relevant to engineering drawing principle.

Table 2.1 ANSI Y14 example list

Y14.100	Engineering drawing practices
Y14.24	Types and applications of engineering drawings
Y14.3	Multiview and sectional view drawings
Y14.31	Undimensioned drawings
Y14.36M	Surface texture symbols
Y14.38	Abbreviations and acronyms for use on drawings and related documents
Y14.4M	Pictorial drawing
Y14.41	Digital product definition data practices
Y14.42	Digital approval systems
Y14.5	Dimensioning and tolerancing
Y14.5.1M	Mathematical definition of dimensioning and tolerancing principles
Y14.6	Screw thread representation

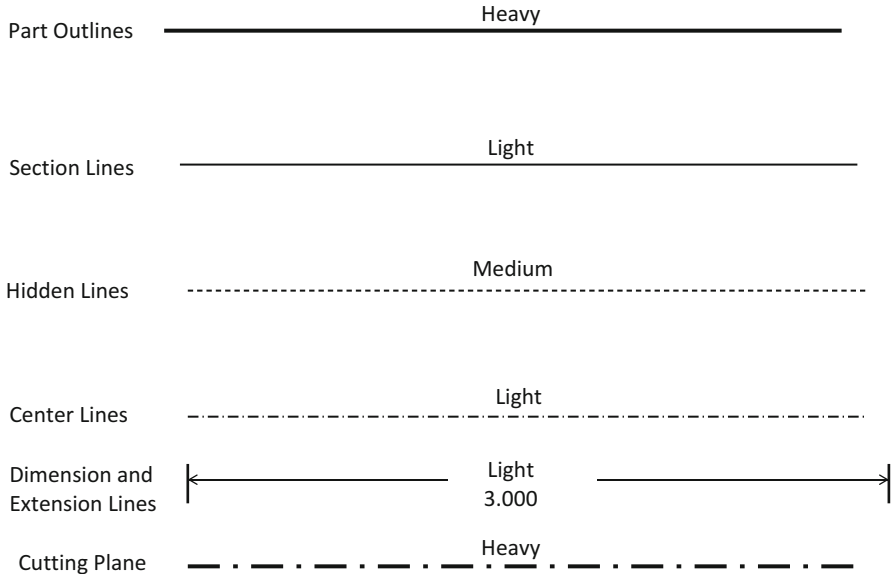


Fig. 2.1 Line style for engineering drawing

- Different line-styles have different meanings on a drawing (see Fig. 2.1).
- The internal form of shapes is described by imagining part of the object removed to show internal details in a sectional view.
- Two principal conventions (first, third angle projection) exist to specify how views should be related to each other on a drawing.
- Projection into a single plane that is not aligned with any of the main faces of an object is known as pictorial projection.
- Dimensions, tolerance, surface conditions are identified using a symbolic representation.
- Repetitive drawing of complex shapes is represented by symbolic notation.

2.2.1 Line Style

Y14 defines meanings of various line styles (see Fig. 2.1). Examples of line styles are in Fig. 2.2.

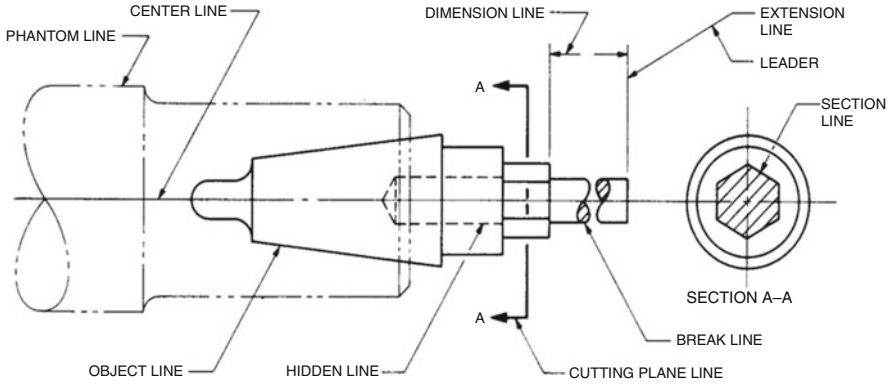


Fig. 2.2 Family of lines

2.2.2 Sectional View

Sectional view is a useful drawing to understand the internal structure of a part. They are used when other view would fail to clearly show the internal details. Sectional views are created by placing an imaginary cutting-plane through the part to expose the interior. There are three different types of sectional view: conventional, half, and full sectional views. The half sectional view is the most popular since it reveals the internal view in conjunction with the exterior view.

2.2.3 Orthographic Projection

Orthographic Projection is a way of drawing a 3D object from different directions. Usually a front, side, and plane views are drawn so that a person looking at the drawing can see all the important sides. Two principal conventions exist to specify how views should be related to each other on a drawing: First Angle and Third Angle. They differ only in the position of the plan, front and side views. In both projections, an object is placed in a box where a parallel project will take place on each side of the box. In first-angle projection, each view is pushed through the object onto the plane furthest from it. For instance, the top view will be pushed through the object and it forms on the bottom plane. In third-angle projection, each view is pulled onto the plane closest to it. Therefore, the top view is on the top of the parallel projection of the orthographic views (Figs. 2.3, 2.4, and 2.5).

First angle orthographic projection is more popular in Europe, while the third angle orthographic projection is popular in USA. Another example is shown in Fig. 2.6.

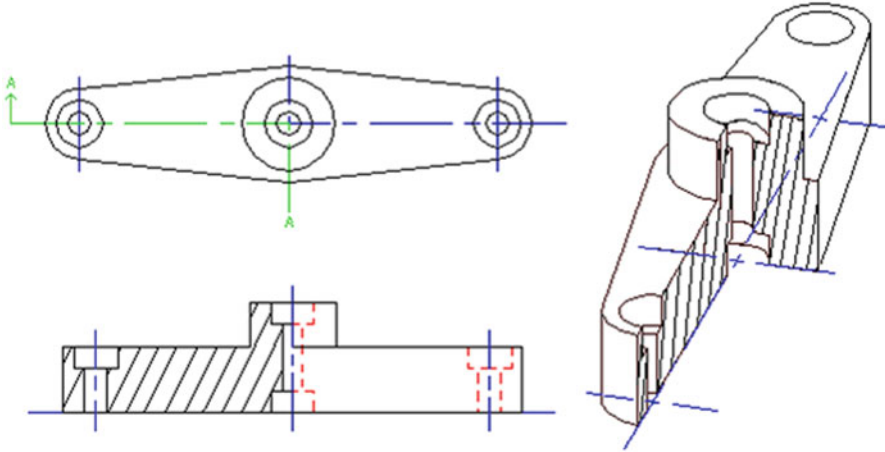


Fig. 2.3 Sectional view (half sectional view)

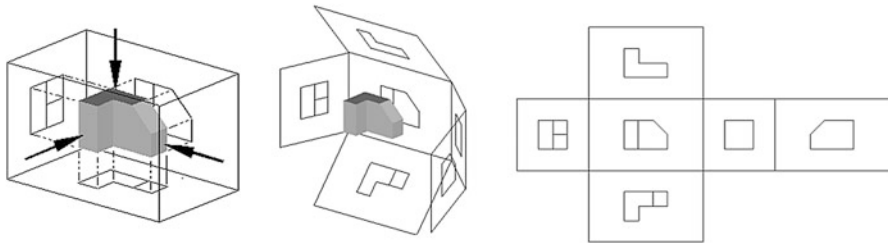


Fig. 2.4 First angle projection

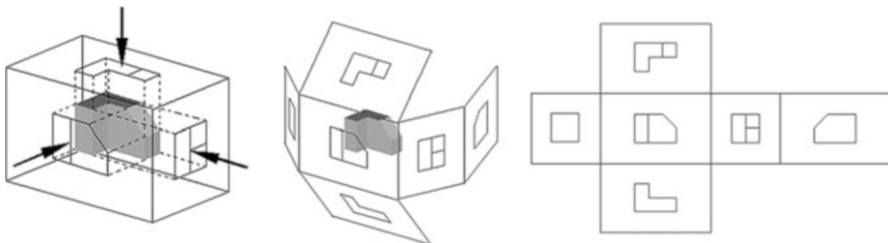
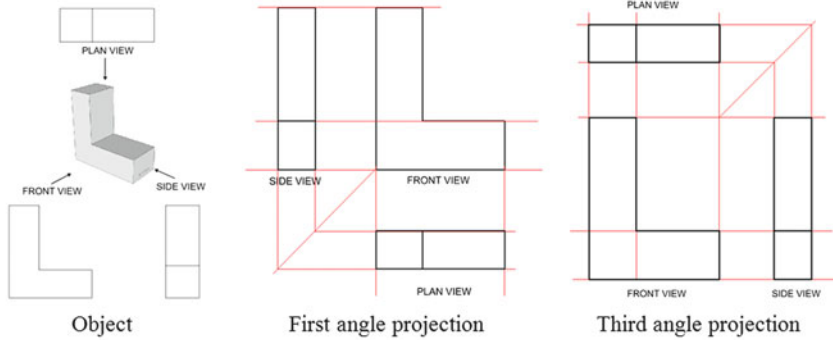


Fig. 2.5 Third angle projection



1st angle and 3rd angle orthographic projections

- 1st angle projection: each side view is pushed through the object onto the plane furthest from it.
- 3rd angle projection: each side view is pulled onto the plane closest to it.

Fig. 2.6 First and third angle orthographic projection

First Angle and Third Angle Orthographic Projections

- First angle projection: each side view is pushed through the object onto the plane furthest from it.
- Third angle projection: each side view is pulled onto the plane closest to it.

2.2.4 Pictorial Projection

Projection into a single plane that is not aligned with any of the main faces of an object is known as pictorial projection. Pictorial sketches are a type of technical illustration that shows several faces of an object at once. Axonometric and oblique pictorial sketches use a parallel projection technique and are frequently used in technical documents, sales literature, maintenance manuals, and documentation supplements in engineering drawings.

Perspective sketches are the most realistic types of drawings used in engineering and technology. A perspective drawing creates a pictorial view of an object that resembles the way we see an object. It is the best method for representing an object in three dimensions. Axonometric projection is often used for perspective sketch. It is a parallel projection technique used to create pictorial drawings of objects by rotating the object on an axis relative to a projection plane to create a pictorial view. One of the following three principle projection techniques is used in technical drawing:

1. Axonometric
2. Oblique
3. Perspective

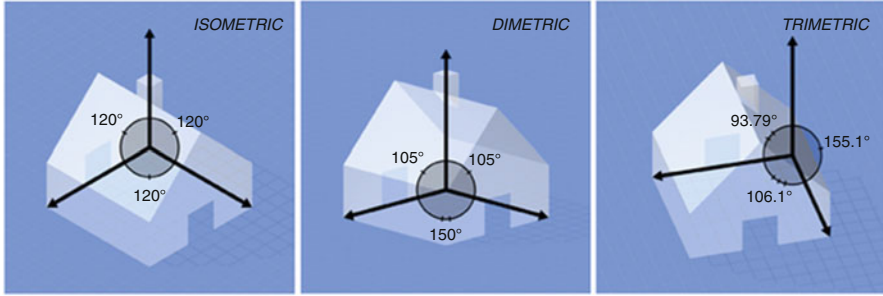


Fig. 2.7 Axonometric projection

2.2.4.1 Axonometric Projections

In axonometric and oblique projection, the observer is theoretically infinitely far away from the projection plane. Only in perspective projection is the viewer at some finite distance to the object. The differences between a multiview drawing and an axonometric drawing are that, in a multiview, only two dimensions of an object are visible in each view and more than one view is required to define the object; whereas, in an axonometric drawing, the object is rotated about an axis to display all three dimensions, and only one view is required. There are three main types of axonometric projection: *isometric*, *dimetric*, and *trimetric* projection (Fig. 2.7).

Axonometric drawings are classified by the angles between the lines comprising the axonometric axes. When all three angles are unequal, the drawing is classified as trimetric. When two of the three angles are equal, the drawing is classified as dimetric. When all three angles are equal, the drawing is classified as an isometric. The forward tilt of the cube causes the edges and planes of the cube to become foreshortened as it is projected onto the picture plane. Thus, the projected lengths are approximately 80 % of the true lengths and an isometric projection ruler must be used. If the drawing is drawn at full scale, it is called an isometric drawing. Isometric drawings are almost always preferred over other to axonometric projections for engineering drawings because they are easier to understand and to produce.

2.2.4.2 Oblique Projections

Oblique projection is a unique form of parallel projection. As the name indicates, oblique projection results when the projectors are parallel to each other at some angle other than perpendicular to the projection plane. If the principal view of the object is placed such that its surfaces are parallel to the projection plane, the resulting projection is an oblique pictorial projection. Historically, the most descriptive face of an object in oblique projection is placed parallel to the frontal plane (Fig. 2.8).

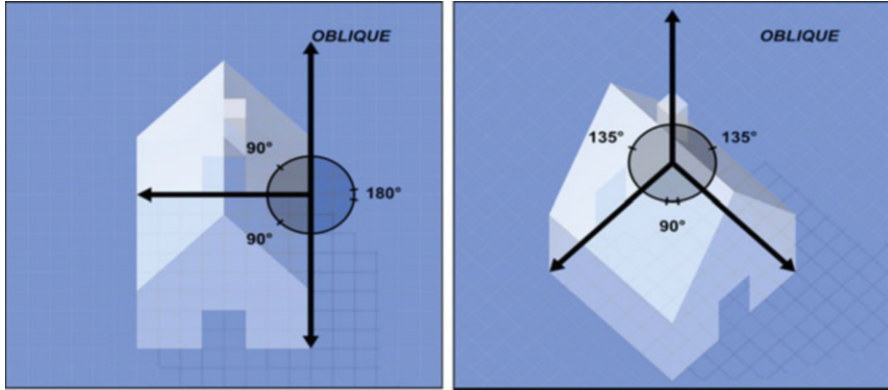


Fig. 2.8 Oblique projection

The actual angles that the projectors make with the plane of projection in oblique projection are not fixed, thus different angles can be used. However, angles for receding edges of between 30° and 60° are preferable because they offer minimum distortion of the object. Several types of oblique drawings include:

1. The cavalier oblique is drawn in true length along the receding axis.
2. The cabinet oblique is drawn in half the true length along the receding axis.
3. The general oblique can be drawn in anywhere from full to half-length along the receding axis.

The half-size receding axis on the cabinet oblique reduces the amount of distortion. Any face of an object that is placed parallel to the frontal plane in oblique projection will be drawn true size and shape. Thus, the first rule in creating an oblique drawing is to develop the drawing so that cylinders or irregular surfaces are placed parallel to the frontal plane. This allows these features to be drawn quicker and without distortion. A second rule in developing oblique drawings is that the longest dimension of an object should be located parallel to the frontal plane. If there is conflict between these two rules, always draw the cylindrical or irregular surfaces parallel to the frontal plane because representing this geometry without distortion is more advantageous.

2.2.4.3 Perspective Projection

Perspective drawing techniques are used primarily because they are the closest to representing objects and scenes as we would view them in the real world. When the human eye views a scene, objects in the distance appear smaller than objects close by—this is known as perspective effect. While orthographic projection ignores this effect to allow accurate measurements, perspective definition shows distant objects as smaller to provide additional realism. The perspective projection requires a more

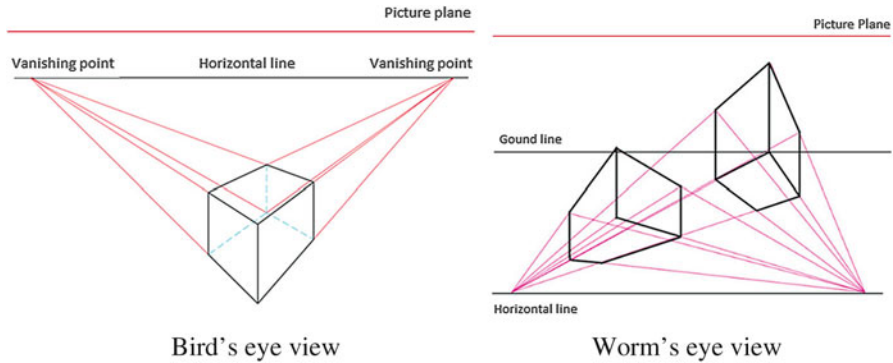


Fig. 2.9 Perspective projections

involved definition as compared to orthographic projections. A conceptual aid to understanding the mechanics of this projection is to imagine the 2D projection as though the object(s) are being viewed through a camera viewfinder. The camera's position, orientation, and field of view control the behavior of the projection transformation. One of the most important features of perspective drawings is the convergence of parallel edges as they recede from the viewer. Therefore, the same object can be seen larger in the close vicinity, while the one at the far distance will be seen smaller. Terminology will be important for both explaining perspective drawing techniques and laying them out. Important terms include:

- The horizon line is the position that represents the eye level of the observer.
- The station point in the perspective drawing is the eye of the observer.
- The picture plane is the plane upon which the object is projected.
- A vanishing point is the position on the horizon where lines of projection converge.
- The ground line represents the plane on which the object rests (Fig. 2.9).

Perspective views are classified according to the number of vanishing points. Increasing the number of vanishing points increases the realism of the drawing, but also increases the drawing difficulty. The vanishing points for one- and two-point perspective drawings both go to the horizon line. The third vanishing point in a three-point perspective drawing is located perpendicular to the horizon line. Four different types of perspective views include:

1. Bird's eye view: ground line below the horizon line.
2. Human's eye view: ground line 6 ft below the horizon line.
3. Ground's eye view: ground line at the same level as the horizon line.
4. Worm's eye view: ground line above the horizon line.

There are several important variables in perspective projection. Selection of variables emphasizes the importance of planning ahead for a drawing by deciding on a number of key variables. Since the primary purpose of a perspective drawing is

to convey a sense of realism, it is important to define the relationship of the observer to the object. The important variables are:

1. Distance of the object from the picture plane
2. Position for the station point
3. Position of the ground line relative to the horizon line
4. Number of vanishing points

For example, the depiction of a toaster would probably have the object fairly close to the picture plane with the observer looking either straight ahead or slightly down at it (i.e., human's eye or bird's eye view). On the other hand, a large building would be farther away using either a worm's eye or ground's eye view, depending on how far the building is from the observer.

Pictorial Projection

Projection into a single plane that is not aligned with any of the main faces of an object is known as pictorial projection.

1. Axonometric projection: the object is rotated about an axis to display all three dimensions.
2. Oblique projection: projectors are parallel to each other, but at some angle other than perpendicular to the projection plane.
3. Perspective projection: perspective projection shows distant objects as smaller to provide additional realism.

2.3 Tolerance Basics

Tolerance is defined as an allowable variation in part dimension. Designers often specify the tolerance and surface finish of a designed part to enable interchangeability. As different companies produce identical or similar parts worldwide, parts have been standardized to facilitate part exchange. However, different accuracy of the part dimension often causes problems in substitution from one part to another. Therefore, when a new part is designed, tolerance has to be specified so that a part from different brands must be interchangeable. In addition, tolerance makes direct impact on manufacturing cost of parts. Although a part has to be made to meet the required dimensional accuracy, it does not need to be overdesigned for tolerance accuracy, which results in higher manufacturing cost. Understanding of customers' demand for dimensional accuracy is important in part design, not only to meet the need, but also to minimize the manufacturing cost. Therefore, tolerance must be as large as possible without interfering with the function of the part. There are three methods in tolerance specification.

1. Unilateral: Permits variation in only one direction from the basic dimension
2. Bilateral: Permits variation in both directions from the basic dimension
3. Limit forms: Represents largest and smallest sizes

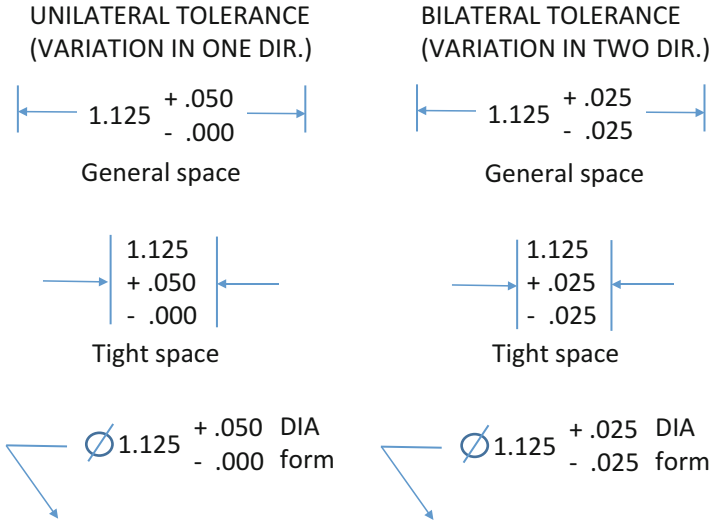


Fig. 2.10 Tolerance methods

Fig. 2.11 Feature control frames (FCF)

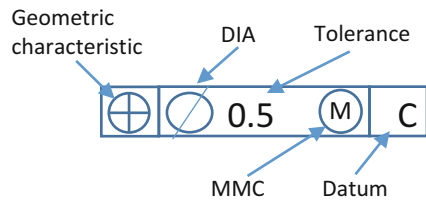


Figure 2.10 below includes examples of different tolerance methods.

Feature Control Frames (FCF) are often used to specify the geometric tolerances such as location, orientation, form, or profile accuracies (see Fig. 2.11). As shown in the figure below, the first column indicates the type of tolerance with a unique symbol (see Fig. 2.11). In the second column, the level of tolerance is specified followed by the datum plane. The symbol (M) stands for MMC (Maximum Material Condition), which is discussed in Sect. 2.3.1.

2.3.1 Datum Plane

Datum planes or datum lines are used to specify the reference or baseline for specifying tolerances. Once a datum plane is specified, all of the tolerances are specified from the datum plane so that the dimensional changes can be minimized by eliminating the tolerance accumulation. The tolerancing exercise without the use of datum plane is called chain tolerancing. For instance, in Fig. 2.12, while the

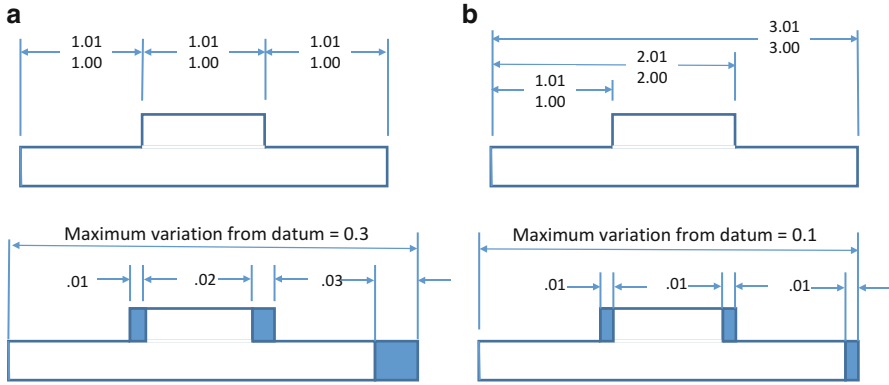


Fig. 2.12 Datum plane. (a) Chain dimensions. (b) Datum plane (BASELINE) dimensions

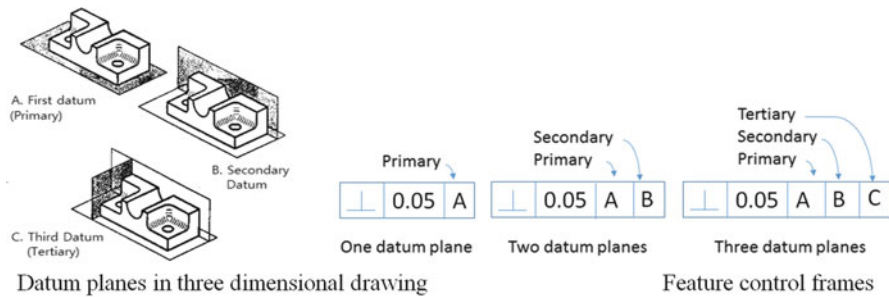


Fig. 2.13 Datum planes and their priority in FCF

accumulation of the dimensional error by chain tolerancing is 0.06 on the right hand side, the dimensional error by using the datum plane is only 0.02, which is the same for all of the dimensioning in the figure. Engineers are always encouraged to use the concept of datum plane to minimize the accumulation of dimensioning errors.

In a three-dimensional drawing, more than one datum plane may be necessary. In general, three mutually perpendicular datum planes are required to dimension a part accurately (see Fig. 2.13). In order to be consistent in dimensioning, we need to list the order of priority of datum planes in the FCF since the priority is important for manufacturing as well as inspection. For a cylindrical part, two theoretical planes are normally used for datum plane specification. More details of the geometric symbols and tolerance specifications are discussed in the later session (Sect. 2.3.3).

The symbol (M) in FCF represents MMC, while the symbol (L) is used for Least Material Condition (LMC). In the example shown in Fig. 2.14, one can modify the tolerance specification by using (M) or (L) symbols.

In the case of MMC, the block (or the outside of the part) is at its Maximum Size or its largest size allowed within the dimensions tolerance and the hole (or the

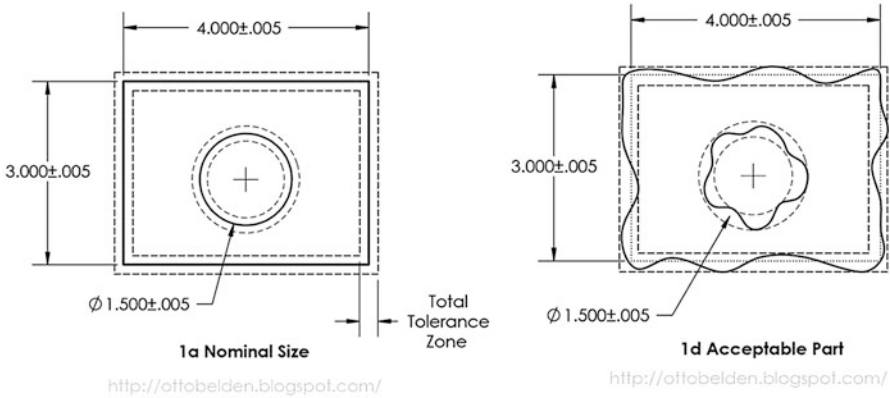


Fig. 2.14 Tolerance example and acceptable part based on the tolerance specification [3]

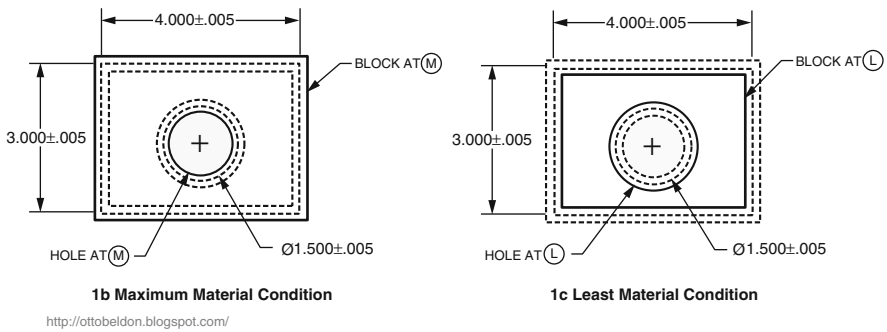


Fig. 2.15 MMC and LMC [4]

inside of the part) is at its smallest Size. At MMC all the internal features are at their smallest sizes and all the external features are at their largest sizes allowed by tolerances. Least Material Condition or LMC is the same concept as MMC in that when something has the least amount of material, the hole (or all internal features) is at their largest sizes and the external features are at their smallest sizes allowed by the dimensions tolerance. MMC and LMC become very important when the part is used for mating with other parts. For example, you might have a hole where a pin has to fit through and you have to be sure that the hole is big enough for the pin. In this case you will want to consider how big the hole is going to be at MMC because that would be the smallest the hole could ever be. While the hole is at MMC (or smallest size), you would have to also consider the pin at MMC (biggest size) and be sure they will fit together. The MMC sizes in this case would be the “worst case” fit for the pin and the hole (Fig. 2.15).

Figure 2.16 demonstrates the importance of the datum plane priority. Two datum planes are specified for a part in the figure, showing several different consequences

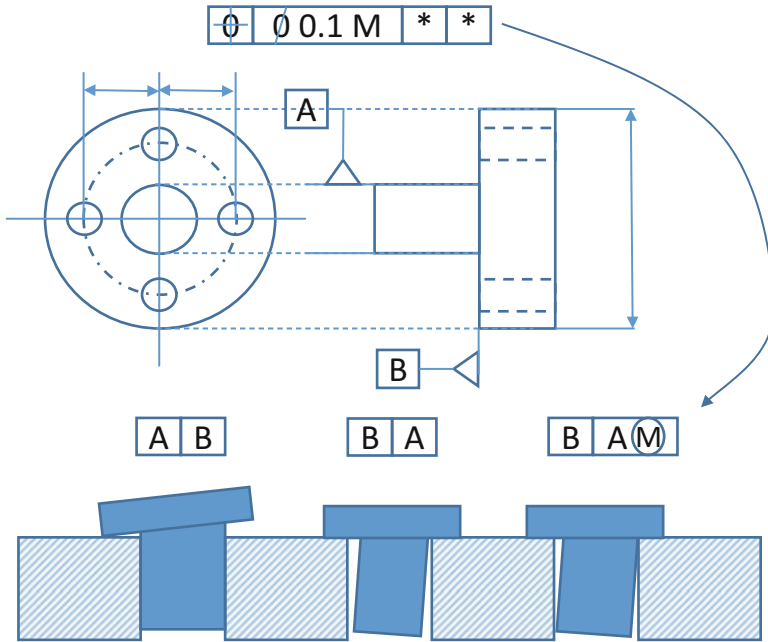
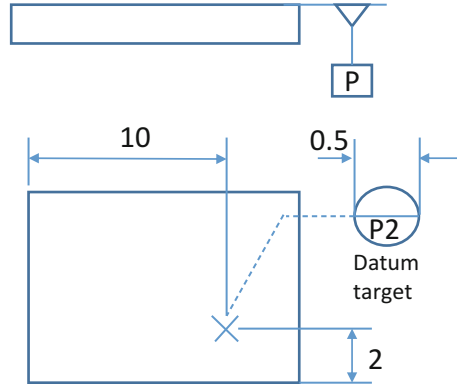


Fig. 2.16 The importance of the datum plane priority

resulting from the different datum plane priority. Let's assume the tolerance specification of this pin and hole case is the transition fit (see Fig. 2.19), meaning the pin has to be put to the hole tightly with little or no tolerance. Notice that the 2nd case may cause trouble since the datum axis has been sloped since the datum plane "B" is the first priority. If a manufacturer made the pin part with respect to the original datum axis, then the pin may not fit to the hole. However, in the 3rd case since the datum plane "A" is specified with MMC, the pin should be made to fit into the hole in the worst case. This will require extra inspection of the pin, but it guarantees the pin and the hole will mate each other. Therefore, MMC or LMC symbols have to be carefully used since they may cause extra cost for manufacturing.

Datum planes, in general, have to provide superior accuracy compared to other dimensions, since the tolerances have to be within the specified limit if measured from any part of the datum plane. Therefore, the surface used for the datum plane has to satisfy higher level of accuracy to guarantee the total dimensioning error by the specified tolerance limit. Less intensive datum specifications can be used for general cases by using datum targets. In Fig. 2.17, the symbol, "x," is used to specify the datum target. As shown in the figure, the tolerance will be specified with respect to the datum target instead of using plane surface. As shown in Fig. 2.17, a datum target line can be also used when a part is supported on a contact line. Datum targets, in general, reduce the cost of manufacturing significantly since it relaxes the required accuracy of datum planes.

Fig. 2.17 Datum targets



MMC and LMC

At MMC, all the internal features are at their smallest sizes and all the external features are at their biggest sizes allowed by the dimensions tolerances.

LMC is the same concept as MMC in that when something has the least amount of material, the hole (or all internal features) is at their largest sizes and the external features are at their smallest sizes allowed by the dimensions tolerance.

2.3.2 Hole and Shaft Tolerance

Calculating tolerances between holes and shafts that fit together is so common in engineering design that a group of standard values and notations has been established. Terminologies in tolerancing for Metric and English unit system are different. The list below includes important terminologies in hole and shaft tolerancing exercise for English unit system.

- Limits of tolerance: the maximum and minimum sizes of features
- Allowance: the tightest fit between two mating parts
- Nominal size: a general size of a shaft or hole
- Basic size: The size to which plus-and-minus tolerance is applied
- Fit: the degree of tightness or looseness between two assembled parts (Clearance, Interference, Transition, Line Fits)

Meanwhile, the list below includes important terminologies common in metric unit system.

- Deviation: the difference between the hole or shaft size and the basic size
- Upper deviation: the difference between the maximum permissible size of a part and its basic size

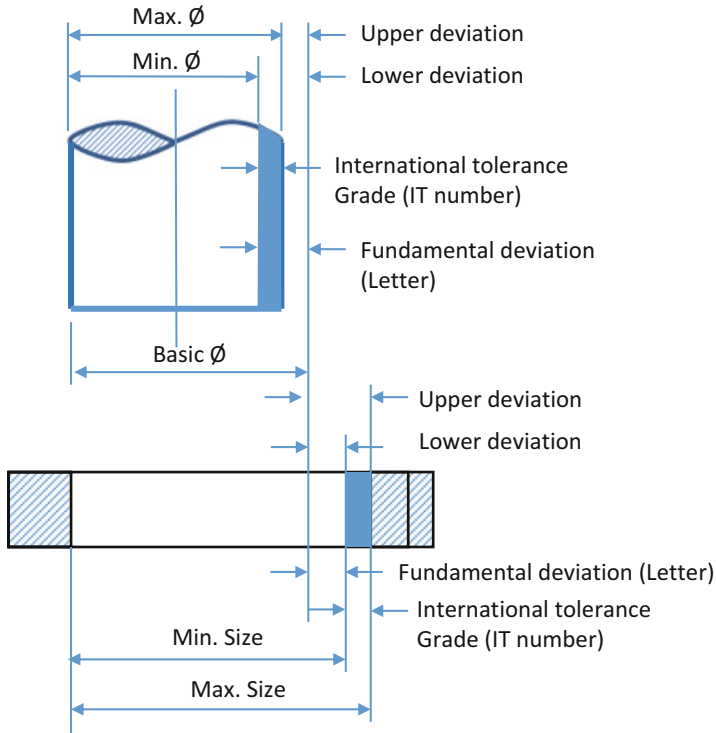


Fig. 2.18 Terminologies in tolerance

- Lower deviation: the difference between the minimum permissible size of a part and its basic size
- Fundamental deviation: the deviation closest to the basic size
- International tolerance (IT) grade: a series of tolerances that vary with basic size to provide a uniform level of accuracy within a given grade
- Tolerance zone: a combination of the fundamental deviation and the tolerance grade (Fig. 2.18)

2.3.2.1 Type of Fits

One of the most important aspects of tolerance specification is about the type of fits for two different parts. There are four general fitting methods used in industry: clearance fit, interference fit, transition fit, and line fit. One of the typical examples in tolerance decision making is the fittings between a hole and a shaft (see Fig. 2.19). Clearance fit provides a clearance between two assembled mating parts. A clearance fit always defines the maximum shaft diameter as smaller than the minimum hole diameter. Therefore, two parts do not meet at all with positive

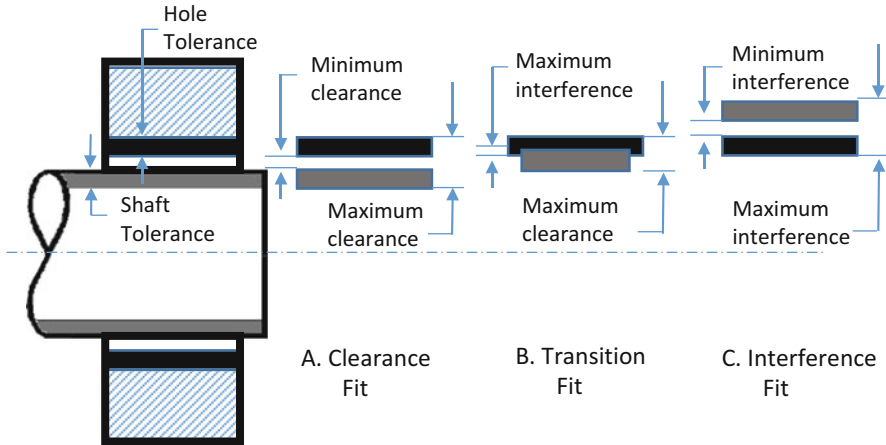


Fig. 2.19 Different type of fittings for tolerance specification

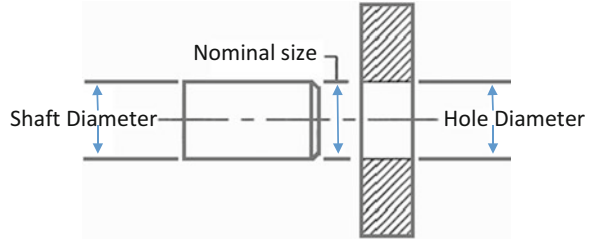
clearance between each other. This type of fit is used for sliding mechanisms between two parts. Interference fit is the tolerance exercise where two parts are in contact with each other. The dimension of the smaller part has to be larger than the larger part in interference fit so that no relative motion is allowed between two parts. That is, an interference fit always defines the minimum shaft diameter as larger than the maximum hole diameter, or more simply said, the shaft is always bigger than the hole. Therefore, in interference fit, the shaft will be assembled by a forced fit to cause the shaft not to move relatively through the hole. The transition fit is the fitting exercise that requires the level tolerance somewhere in between clearance fit and interference fit.

Various Types of Fits

- **Clearance fit** provides a clearance between two assembled mating parts.
- **Interference fit** results in a binding fit (require parts to be forced together).
- **Transition fit** ranges from an interference to a clearance fit.
- **Line fit** results in surface contact or clearance when the limits are reached.

2.3.2.2 Tolerance Calculation for Hole and Shaft System (Metric Values)

The term nominal refers to the approximate size of an object that matches a common fraction or whole number (Fig. 2.20). A shaft with a dimension of 1.500 ± 0.003 is said to have a nominal size of “one and a half inches.” A dimension of 1.500 ± 0.005 is still said to have a nominal size of one and a half inches. In both examples, 1.5 is the closest common fraction.

Fig. 2.20 Nominal size

When it comes to tolerance calculation for the hole and shaft tolerancing, there are two different systems available: basic hole system and basic shaft system. Basic hole system uses the smallest hole size as the basic diameter. The smallest diameter is used since a hole can be enlarged later if needed. In basic shaft systems, the largest diameter size of the shaft will be used as the basic diameter since the shaft can be machined to a smaller size later if needed. This system is used when shafts are available in uniform standard sizes.

How do we specify tolerance for the hole and shaft design then? First of all, we need to determine which system will be used for the tolerancing. Between hole and shaft system, the primary part is the one thought to be fixed in its diameter. Then the mating part has to be naturally selected for tolerancing. For instance, if a designer needs to make a shaft for a hole on a wheel axle, then the basic shaft system has to be selected (Fig. 2.21).

Once the nominal size is specified, standard sizes have to be considered for the first choice. It is important that designers always consider preferred and standard sizes when selecting sizes for shaft/hole design. Since most tooling is set up to match these sizes, manufacturing is greatly simplified when preferred and standard sizes are specified. First choice increases by 25 %, whereas second choice increases by 12 %. The best practice is to choose a basic diameter from the first column (standard stock size) to minimize cost. Second, a type of fit has to be specified. There are several subclassifications for each type of fit (see Tables 2.2, 2.3, and 2.4). A specific description for each category of fit follows. Finally, use ANSI B4.2 to find tolerance for the hole and the shaft.

Hole and Shaft Tolerance (Metric)

- Determine which system to use (hole or shaft)
- Use preferred basic sizes to compute tolerances
- Choose basic diameter from the first column (standard stock size) to minimize cost
- Determine the type of fit (Table 2.2–2.4)
- Use ANSI B4.2 to find tolerance for hole and shaft

CLASS RC9 FIT
(1.97-3.15 DIA)

LIMITS OF CLEARANCE	HOLE	SHAFT
9.0	7.0	-9.0
20.5	0	-13.5

HOLE: 2.5000 BASIC DIA

Upper Limit	Lower Limit
2.5000	2.5000
.0070	0

2.5070 ----- 2.5000

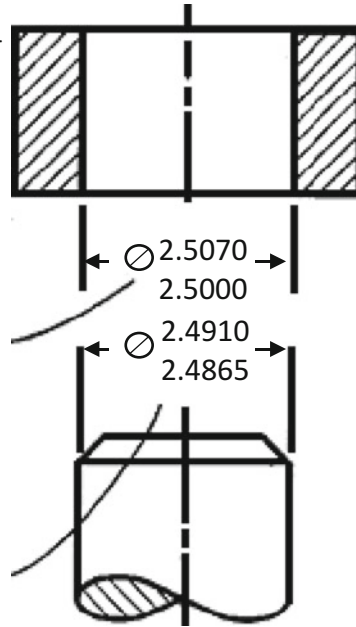
SHAFT: 2.5000 BASIC DIA

Upper Limit	Lower Limit
2.5000	2.5000
-.0090	-.0135

2.4910 ----- 2.4865

Limits of Clearance

2.5000	2.5070
2.4910	2.4865
+.0090	+.0205



Since basic DIA
Appears on hole,
This is a basic
Hole system

Fig. 2.21 Basic hole and shaft system

Table 2.2 Clearance fit (Metric)

Clearance fits	
H11/c11 or C11/h11	Loose running fit
H9/d9 or D9/h9	Free running fit
H8/f7 or F8/h7	Close running fit
H7/g6 or G7/h6	Sliding fit
H7/h6	Locational clearance fit

Table 2.3 Transition fit

Transition fits	
H7/k6 or K7/h6	Locational transition fit
H7/n6 or N7/h6	Locational transition fit

Table 2.4 Interference fit

Interference fits	
H7/p6 or P7/h6	Locational transition fit
H7/s6 or S7/h6	Medium drive fit
H7/u6 or U7/h6	Force fit

PREFERRED SIZES (mm)			
First Choice	Second Choice	First Choice	Second Choice
1	1.1	12	14
1.2	1.4	16	18
1.6	1.8	20	22
2	2.2	25	28
2.5	2.8	30	35
3	3.5	40	45
4	4.5	50	55
5	5.5	60	70
6	7	80	90
8	9	100	110
10	11	120	140

Fig. 2.22 Preferred sizes for hole and shaft design (Metric)

Sample Problem 2.1

Determine hole and shaft dimensions for the following design specification.
 Hole basis, close running fit, basic size: 7

Solution

First, from the preferred size in Fig. 2.22, since 7 is not listed in the figure, we can choose the one closest. We can choose either 6 or 8 unless it has to be manufactured at the correct size of 7. Let's set up the basic size to be 8. From Table 2.2, the subclassification of the type of fit becomes H8/f7 or F8/h7. Since it is a hole basis system, the right symbol has to be H8/f7 (Fig. 2.23). Now from the ANSI B4.2, the tolerance for the basic size 8 turned out to be following (see Appendix A)

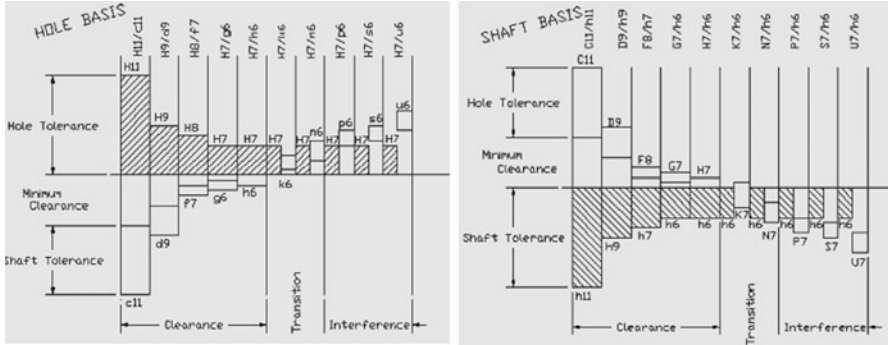


Fig. 2.23 Subclassifications of fitting types (Metric)

Basic Size	...	Close Running		
⋮		Hole H8	Shaft f7	Fit
⋮		⋮		
8 Max	...	8.022	7.987	0.050
Min		8.000	7.972	0.013

The size of the hole and shaft becomes,

Hole: 8.022, 8.000

Shaft: 7.987, 7.972

Sample Problem 2.2

Determine hole and shaft dimensions for the following design specification. Hole basis, close running fit, basic size: 12.

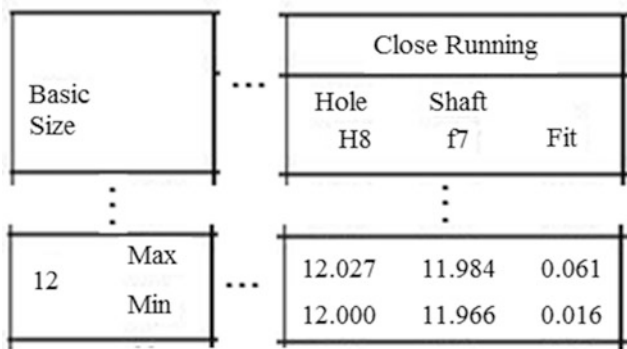
Solution

First, from the preferred size, 12 is a preference first choice as shown in Fig. 2.22.

From Fig. 2.23, the subclassification of the type of fit becomes H8/f7 or F8/h7.

Since it is a hole basis system, the right symbol has to be H8/f7 (from Table 2.2).

Now from the ANSI B4.2, the tolerance for the basic size 12 turned out to be following (see Appendix A).



Therefore, the size of the hole and shaft becomes,

Hole: 12.027, 12.000

Shaft: 11.984, 11.966

2.3.2.3 Tolerance Calculation for Hole and Shaft System (Inch Values)

American standard uses the inch values where the nominal values are not specific, but rather in certain range. Below is the Fits defined by inch value classification system.

- RC = Running and sliding fits
- LC = Clearance locational fits
- LT = Transitional locational fits
- LN = Interference fits
- FN = Force fits

Each of these categories has several subclassifications within it, defined by a number. For instance, Class RC1, Class RC2 through Class RC9 belongs to the RC classification. The values in the table (Appendix A) are thousandths of an inch. Therefore, the value of 0.5 means 0.0005 in.

Sample Problem 2.3

Determine hole and shaft dimensions for a Class LN1 Interference fit based on a nominal diameter of 0.25 in. Use hole basis values.

Solution

From the chart for inch values in Appendix A, the tolerance data for the nominal size range will be from 0.24 to 0.4. For a Class LN1, the tolerance for the hole is 0–0.4, while the tolerance for the shaft is 0.4–0.65. Remember tolerance values are one thousandth of an inch.

Nominal Size Range, Inches	Class LN 1		
	Limits of Interference	Standard Limits	
		Hole H6	Shaft n5
0.24– 0.40	0	+0.4	+0.65
	0.65	0	+0.4

Therefore, the size of the hole and shaft becomes,

Hole: 0.2500, 0.2504

Shaft: 0.2540, 0.2565

2.3.3 Geometric Tolerances

In addition to the hole and shaft tolerancing, geometric tolerancing is important in Geometric Dimensioning and Tolerancing (GD&T) exercise. Geometric tolerances are the specific tolerances that control locations, forms, profiles, orientations, and runout. Several different types of geometric tolerance symbols are shown in Fig. 2.24.

2.3.3.1 Location Tolerance

Location tolerance is the tolerance exercise for the accuracy of the position, concentricity, and symmetry of certain features. The level of tolerance for the position accuracy is specified by position circle, square, or rectangular tolerance zone for locating the center of a hole. As shown in Fig. 2.25, FCF specifies the diameter of the circular tolerance zone. In the example of the circular tolerance zone in Fig. 2.25, FCF indicates that the tolerance level is 0.5 mm. This means that the center of the circle can be placed within the circle of 0.5 mm diameter. Circular tolerance zone will result in better accuracy than square tolerance zone. Concentricity is the feature of location that specifies the location of two cylinders that share the same axis.

INDIVIDUAL FEATURES	Form	Straightness	
		Flatness	
		Circularity	
		Cylindricity	
BOTH	Profile	Profile: Line	
		Profile: Surface	
RELATED FEATURES	Orientation	Angularity	
		Perpendicularity	
		Parallelism	
	Location	Position	
		Concentricity	
		Symmetry	
	Runout	Runout: Total	
		Runout: Total	

Fig. 2.24 Geometric tolerance symbols

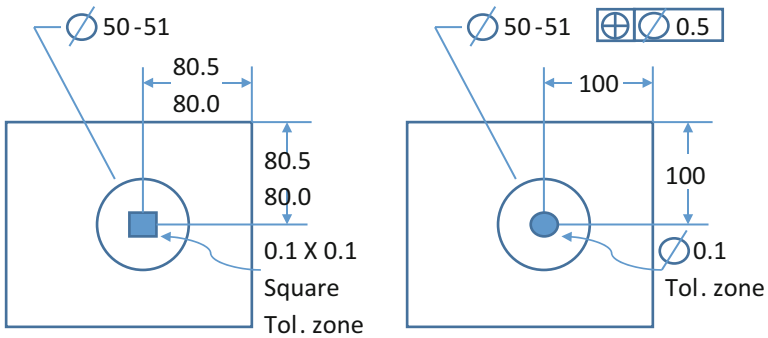


Fig. 2.25 Location tolerance zone

Another example of the location tolerancing is the symmetry tolerance. Symmetry is the tolerance measure of location in which a feature is symmetrical with the same contour and size on opposite sides of a central plane. Figure 2.26 depicts an example of the symmetry tolerance.

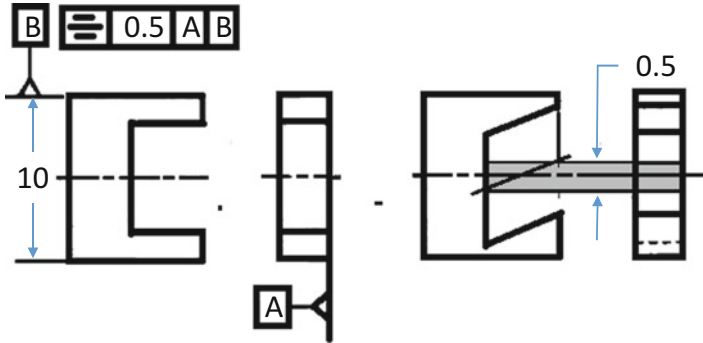


Fig. 2.26 Symmetry tolerance

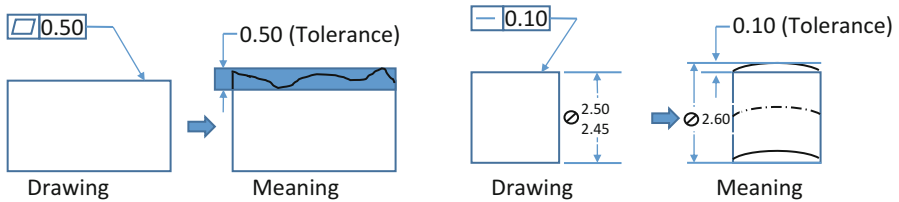


Fig. 2.27 Flatness and straightness

2.3.3.2 Form Tolerance

Form tolerance is to indicate the level of accuracy of forms such as flatness, straightness, circularity, and cylindricity. For example, a surface is said to be flat when all its elements are in one plane. Likewise, a surface is straight if all its elements are on a straight line within a specified tolerance zone. The flatness is relevant to the surface roughness, while the straightness indicates slope or angle constraints of the plane. Therefore, a surface could be flat satisfying the flatness tolerance, but it could be sloped without satisfying the straightness (Fig. 2.27).

Circularity and cylindricity are both tolerance measures of circular features. A surface of revolution is circular when all points on the surface intersected by a plane perpendicular to its axis are equidistant from the axis (see Fig. 2.28). Likewise, a surface of revolution is cylindrical when all its elements lie within a cylindrical tolerance zone. Cylindricity tolerance has to guarantee the combination of tolerances of roundness and straightness at the same time.

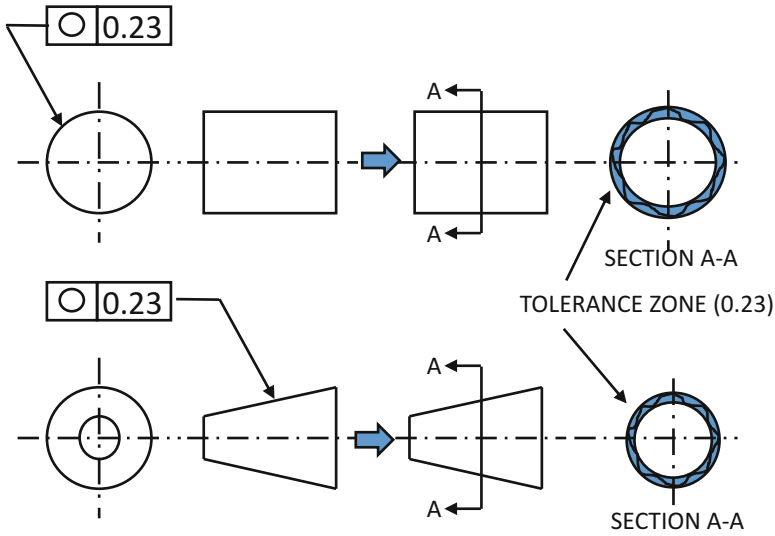


Fig. 2.28 Circularity tolerance

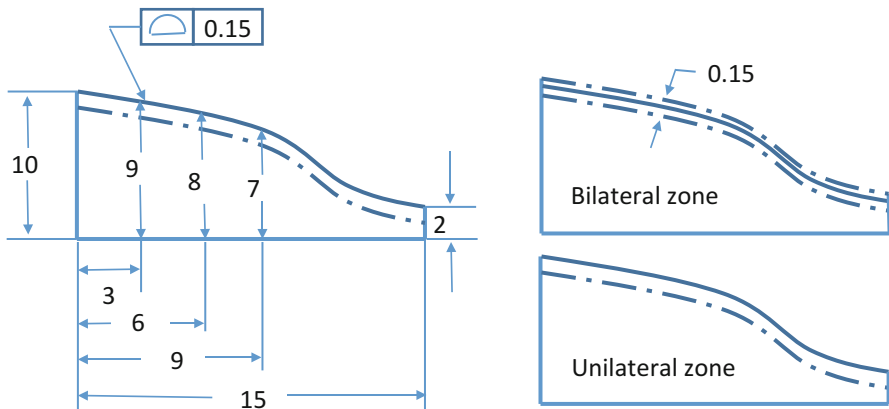


Fig. 2.29 Profile tolerance

2.3.3.3 Profile Tolerance

Profile tolerance specifies tolerances for a contoured shape formed by arcs or irregular curves. Profiles are, in general, defined by $x-y$ coordinates in 2D drawings as shown in Fig. 2.29. Please note that the tolerance value in FCF changes its meaning depending on the tolerancing method.

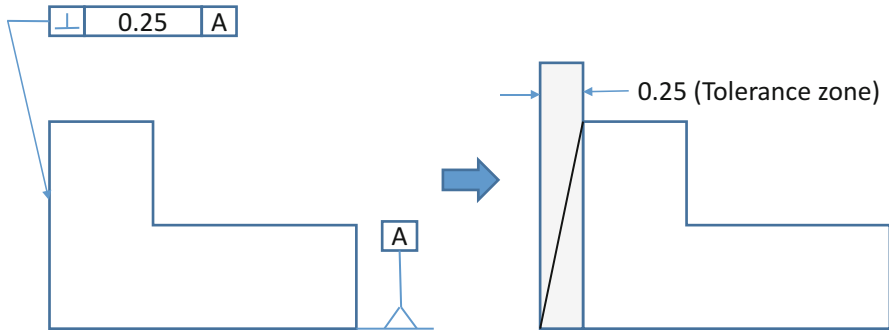


Fig. 2.30 Orientation tolerance

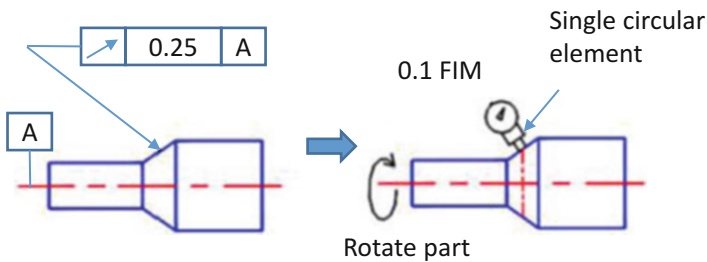


Fig. 2.31 Runout tolerance

2.3.3.4 Orientation Tolerance

There are two main orientation tolerance exercises: perpendicularity and angularity. Perpendicularity is the tolerance of orientation that gives a tolerance zone for a plane perpendicular to a specified datum plane (Fig. 2.30), while angularity is for a surface or line that is at an angle from a datum or an axis.

2.3.3.5 Runout Tolerance

Runout tolerance is a way of controlling multiple features by relating them to a common datum axis. The features of rotation must fall within the prescribed tolerance at full indicator movement (FIM). This is different than circularity, which controls overall roundness. Runout is usually applied to parts with circular cross-sections that must be assembled like drill bits, segmented shafts, or machine tool components. Runout helps to limit the axis offset of two parts to ensure they can spin and wear evenly.

In the example shown in Fig. 2.31, the FCF can be read “each circular element of this surface must have FIM of less than 0.1 relative to datum A.” Note that the indicator is applied perpendicular to the measured surface, and that this tolerance

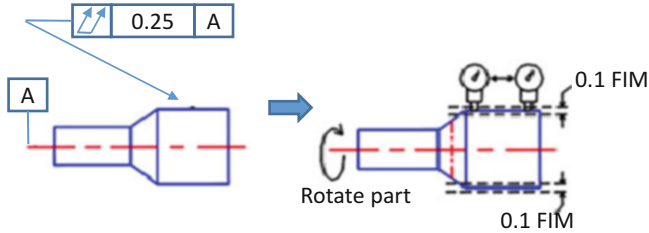


Fig. 2.32 Total runout tolerance

controls only individual circular elements and not the whole surface simultaneously.

Total runout is a complex tolerance that controls a feature's straightness, profile, angularity, and other geometric variations. Total runout is different from runout because it applies to entire surface simultaneously instead of individual circular element. An example of total runout tolerance is shown in Fig. 2.32. The top figure shows a total runout tolerance applied to a horizontal surface. The FCF can be read "this entire surface must have FIM of less than 0.1 relative to datum A." The figure on the right shows how total runout is verified. Note that the indicator is applied all along and is perpendicular to the surface to which the tolerance is applied.

2.4 Surface Texture

Surface texture is the variation in a surface, including *roughness*, *waviness*, *lay*, and *flaws*. The indication of the surface texture is important in that it changes the manufacturing cost dramatically. Below is the list of important terminologies for the surface texture evaluation, which are also defined pictorially in Fig. 2.33.

- Roughness: the finest of the irregularities in the surface caused by the manufacturing process
- Waviness: a widely spaced variation that exceeds the roughness width cutoff
- Roughness height: The average deviation from the mean plane of the surface measured
- Roughness width: the width between successive peaks and valleys forming the roughness
- Roughness width cutoff: the largest spacing of repetitive irregularities
- Waviness height: the peak-to-valley distance between waves
- Waviness width: the spacing between wave peaks or wave valleys
- Lay: the direction of the surface pattern caused by the production method used
- Flaws: irregularities or defects occurring infrequently or at widely varying intervals of a surface including cracks, blow holes, checks, ridges, scratches. . .
- Contact area: the surface that will make contact with a mating surface

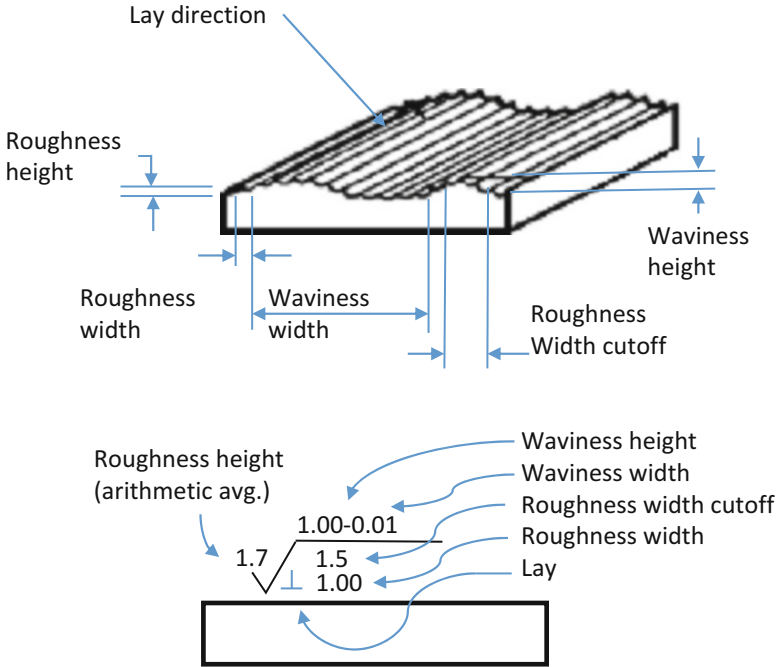


Fig. 2.33 Surface texture

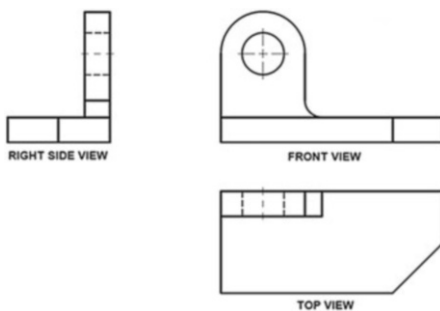
Exercise Problems

- 2.1. Determine hole and shaft dimensions for the following design specification.
 Hole basis, Locational clearance fit, basic size: 5
 (answer)
 Hole: 5.012, 5.00.
 Shaft: 5.000, 4.99.
- 2.2. Determine hole and shaft dimensions for a class LN2 Interference fit based on a nominal diameter of 0.55 in. Use hole basis values.
 (answer)
 Hole: 0.5500, 0.550.
 Shaft: 0.5507, 0.551.
- 2.3. Using class LN3 interference fit based on a nominal diameter of 1 in, determine hole and shaft dimensions.
 (answer)
 Hole: 1.0000, 1.000.
 Shaft: 1.0120, 1.017.

2.4. Draw the first angle project of the given part below.



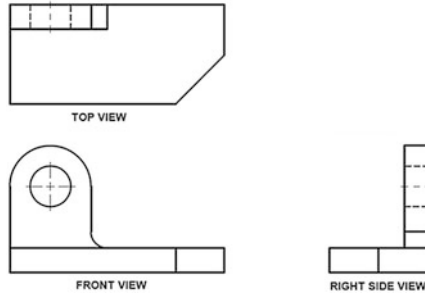
(answer)



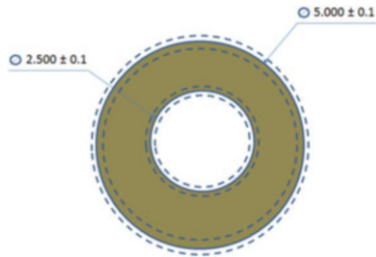
2.5. Draw the third angle project of the given part below.



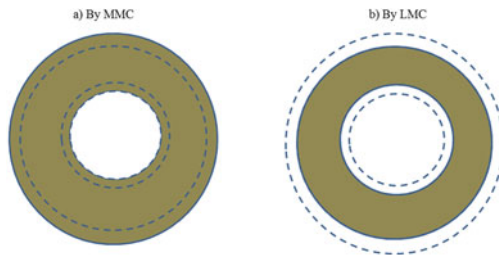
(answer)



- 2.6. For the given tolerance condition in the figure below, draw most likely results by
- (a) MMC.
 - (b) LMC.



(answer)



References

1. Malkevitch J (April 2003) Mathematics and art. Feature Column Archive (American Mathematical Society)
2. Carlbom I, Paciorek J (1978) Planar geometric projections and viewing transformations. *ACM Computing Surveys* 10(4):465–502. doi:10.1145/356744.356750
3. <http://ottobelden.blogspot.com/2011/06/gd-maximum-and-least-material-condition.html>
4. <http://ottobelden.blogspot.com/2011/06/gd-maximum-and-least-material-condition.html>

Chapter 3

3D Geometric Modeling

The Big Picture

Discussion Map

You need to understand terminology and basic principles of 3D geometric modeling.

Discover

- Understand coordinate system.
- Understand the description of frame.
- Understand mapping between two frames.
- Understand general transformation.
- Understand transformation arithmetic.
- Understand general form of rotation.
- Understand 3D modeling schemes

3D modeling becomes an important and critical technology for design engineers and CAD users. Most of the modern CAD packages support 3D modeling for design and visualization. As the computer graphics and rendering technologies are advanced, 3D modeling has been also developed up to the level at which a realistic representation of a virtually any product is made possible. Parametric design capability expedites the representation of ideas, and coloring tools help manifest ideas realistically. Although many 3D modeling technologies have been developed recently, the scope of this book covers the most important and conventional 3D modeling technologies: wireframe, surface modeling, and solid modeling. This will lead us to understand the basic principles of 3D modeling technology and help understand various branches made thereafter.

3.1 Coordinate System

In order to understand the representation and manipulation of 3D parts on the computer screen, we need to learn about coordinates as well as coordinate transformation. The most common coordinate system used for 3D modeling is Cartesian coordinate system, where each axis is perpendicular to each other (see Fig. 3.1). Cartesian coordinate system is a 3D space where models are constructed (right-handed). There are two coordinate systems to represent designed parts efficiently: global coordinate system (GCS) and local coordinate system (LCS). The GCS is a fixed Cartesian coordinate system used for the overall definition of models, while the LCS is a Cartesian coordinate system used to assist in the construction and operation of the model. The coordinate transformation defines the mathematical relationship between the GCS and the LCS. A local coordinate usually is assumed to be attached on each model, thus exists as many as the number of represented models.

Since the LCS is attached on a part, we need a set of parameters to identify the current location of it as the part is manipulated by the designer. If we regard each axis of the LCS as a vector defined in the 3D space of a GCS, the simplest set of parameters is the location of their origin and angles of each axis of the LCS with respect to the GCS. If P_x, P_y, P_z represent the vector, A_p , to the origin of the LCS,

$$A_p = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}. \quad (3.1)$$

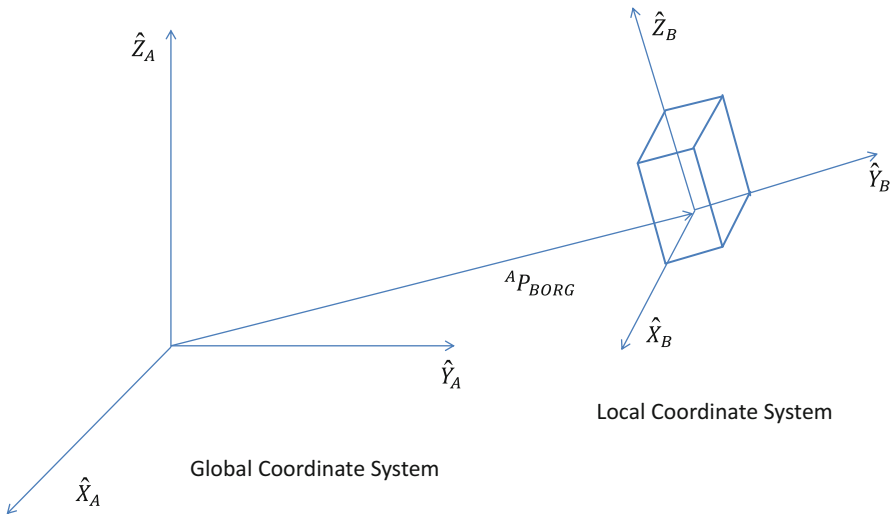


Fig. 3.1 Global and local coordinate system

Suppose a designer wants to observe a part such as the box in Fig. 3.1 from different angle. In order to specify the orientation of a part, we specify the orientation of each axis of the LCS in GCS. It is a mathematical problem as to how to represent the coordinate system {B} relative to the coordinate system {A}. First we define each axis of {B} as a unit vector and all three vectors represent the three principal axes of {B} in terms of the coordinate system {A}, such as ${}^A\hat{X}_B$, ${}^A\hat{Y}_B$ and ${}^A\hat{Z}_B$. Unit vector is a vector whose magnitude is 1 unit. The first element of the set of vectors represents the x -axis of the LCS. It is represented as a unit vector of X_B in the coordinate system {A}. Now we can describe each element of the unit vector of the system {B} with respect to the coordinate system {A} as below.

$${}^A\hat{X}_B = \begin{bmatrix} r_{11} \cdot \hat{X}_A \\ r_{21} \cdot \hat{Y}_A \\ r_{31} \cdot \hat{Z}_A \end{bmatrix}, \quad {}^A\hat{Y}_B = \begin{bmatrix} r_{12} \cdot \hat{X}_A \\ r_{22} \cdot \hat{Y}_A \\ r_{32} \cdot \hat{Z}_A \end{bmatrix}, \quad {}^A\hat{Z}_B = \begin{bmatrix} r_{13} \cdot \hat{X}_A \\ r_{23} \cdot \hat{Y}_A \\ r_{33} \cdot \hat{Z}_A \end{bmatrix} \quad (3.2)$$

The parameter r_{11} represents direction cosine or dot product between \hat{X}_B and \hat{X}_A . That is, the projection of \hat{X}_B as a vector at the x -axis of {A}. The parameter r_{21} represents the projection of \hat{X}_B as a vector at the y -axis of {A}. Likewise, the parameter r_{31} represents the projection of \hat{X}_B as a vector at the z -axis of {A}. In the same token, r_{12} – r_{33} represent projection of the axis \hat{Z}_B as a vector at each axis of {A}. In order to represent the above three equations in a simpler and compact form, we use the following expression for convenience.

$$\begin{bmatrix} {}^A\hat{X}_B & {}^A\hat{Y}_B & {}^A\hat{Z}_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} {}^A\hat{X}_B & {}^A\hat{Y}_B & {}^A\hat{Z}_B \end{bmatrix} = {}^A_B\mathbf{R} \quad (3.3)$$

We define the symbol ${}^A_B\mathbf{R}$ as a rotational matrix between the system {A} and {B}. Since r_{ij} is simply the projection of each axis of {B} on the axes of {A}, using the direction cosine or dot product of two vectors, (3.3) becomes the following equation.

$${}^A_B\mathbf{R} = \begin{bmatrix} {}^A\hat{X}_B & {}^A\hat{Y}_B & {}^A\hat{Z}_B \end{bmatrix} = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix} \quad (3.4)$$

Sample Problem 3.1

Prove the following relationship.

$${}^A_B\mathbf{R} = {}^B_A\mathbf{R}^{-1}$$

Solution

Based on the definition of the ${}^A_B\mathbf{R}$ in (3.4), ${}^B_A\mathbf{R}$ has to be configured as below.

$${}^B_A\mathbf{R} = \begin{bmatrix} \widehat{X}_A \cdot \widehat{X}_B & \widehat{Y}_A \cdot \widehat{X}_B & \widehat{Z}_A \cdot \widehat{X}_B \\ \widehat{X}_A \cdot \widehat{Y}_B & \widehat{Y}_A \cdot \widehat{Y}_B & \widehat{Z}_A \cdot \widehat{Y}_B \\ \widehat{X}_A \cdot \widehat{Z}_B & \widehat{Y}_A \cdot \widehat{Z}_B & \widehat{Z}_A \cdot \widehat{Z}_B \end{bmatrix} \quad (3.5)$$

Now, if we take transpose on the matrix above, we will obtain;

$${}^B_A\mathbf{R}^T = \begin{bmatrix} \widehat{X}_A \cdot \widehat{X}_B & \widehat{X}_A \cdot \widehat{Y}_B & \widehat{X}_A \cdot \widehat{Z}_B \\ \widehat{Y}_A \cdot \widehat{X}_B & \widehat{Y}_A \cdot \widehat{Y}_B & \widehat{Y}_A \cdot \widehat{Z}_B \\ \widehat{Z}_A \cdot \widehat{X}_B & \widehat{Z}_A \cdot \widehat{Y}_B & \widehat{Z}_A \cdot \widehat{Z}_B \end{bmatrix} \quad (3.6)$$

Since dot product has the commutative property, the first row can be rewritten as below.

$$[\widehat{X}_B \cdot \widehat{X}_A \quad \widehat{Y}_B \cdot \widehat{X}_A \quad \widehat{Z}_B \cdot \widehat{X}_A]$$

This is the same as the first row of ${}^A_B\mathbf{R}$ in (3.4). Likewise the second and the third rows will become the same as the second and the third rows in the matrix, ${}^A_B\mathbf{R}$. Therefore, it has been proven that ${}^A_B\mathbf{R} = {}^B_A\mathbf{R}^T$.

Since the rotational matrix is an orthogonal matrix, the transpose and inverse of the rotational matrix is identical. Therefore,

$${}^A_B\mathbf{R} = {}^B_A\mathbf{R}^{-1}$$

3.2 Description of Frame

In order to describe the relationship between local and global coordinates, the translational relation has to be specified. In other words, the location and orientation of every LCS has to be defined with respect to a GCS. To that end, we use a frame, which is a 4×4 matrix to define the orientation and the position of the origin of a local coordinate with respect to the GCS. Therefore, a frame is composed of a position vector and a 3×3 orientation matrix as below.

$$\text{Frame}\{B\} = \{{}^A_B\mathbf{R}, {}^A\mathbf{P}_{\text{BORG}}\}, \quad (3.7)$$

where ${}^A\mathbf{P}_{\text{BORG}}$ is the vector that locates the origin of the coordinate system $\{B\}$ relative to a GCS, $\{A\}$. Therefore, we use a frame as a manifestation of the relationship between a LCS to a GCS. For example, a local frame $\{B\}$ can be expressed as below.

$$\text{Frame } \{B\} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \end{bmatrix} \tag{3.8}$$

In summary, a frame is a description of one coordinate system relative to another.

3.3 Mappings

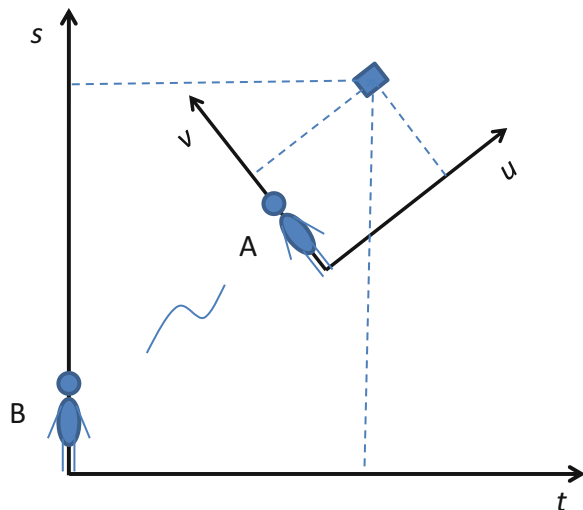
In order to represent a 3D geometric object on the screen, a mapping between a global coordinate and a local coordinate has to be identified. The mapping between two coordinate systems will represent the changing description by using a frame.

This is the same as the situation where two observers are at each coordinate system and they can only talk by phone (see Fig. 3.2). The observer (A) at the coordinate, $v-u$, can watch the object in his coordinate system, while the observer (B) at the coordinate, $s-t$, can't see the object since it is too far away. However, what if the observer, B, can talk to the observer, A, by phone? Then the observer, B, once informed about the location of the object in $v-u$ coordinate can somehow figure out the location of the object in $s-t$ coordinate if the observer A can state its own location and orientation in $s-t$ coordinate. This entire process is known as coordinate transformation, or simply mapping. Let's assume that we try to represent a simple dot in 3D space with two coordinate systems whose corresponding axes are aligned to the same directions (see Fig. 3.3).

If we represent the same dot both from the GCS and a LCS, then the vector that points to the point from {A} can be represented as a sum of the vector representation from the frame {B} plus the mapping between {A} and {B} such that;

$${}^A P = {}^B P + {}^A P_{\text{BORG}}. \tag{3.9}$$

Fig. 3.2 Coordinate transformation



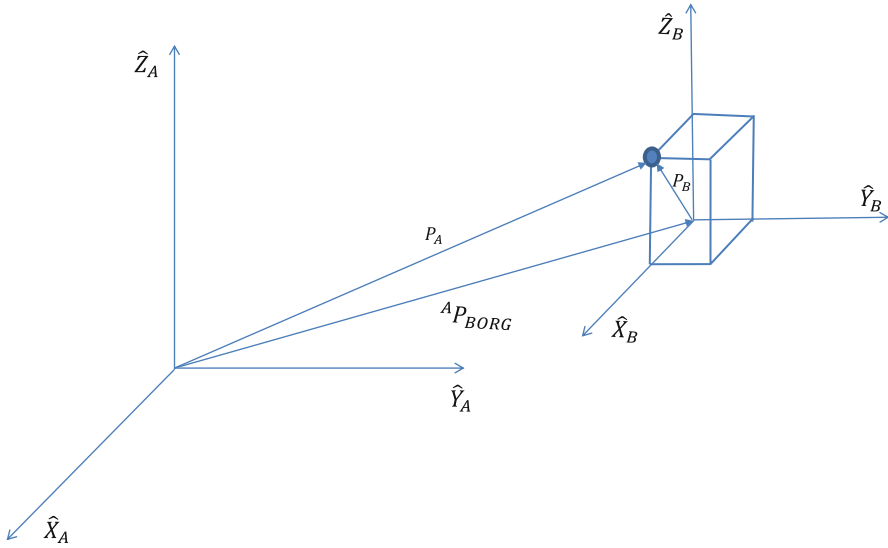


Fig. 3.3 Translational mapping between coordinate systems

The mapping between two vectors from two different coordinate systems can be represented by a translational vector, ${}^A P_{BORG}$, when two coordinates are parallel along the corresponding axes.

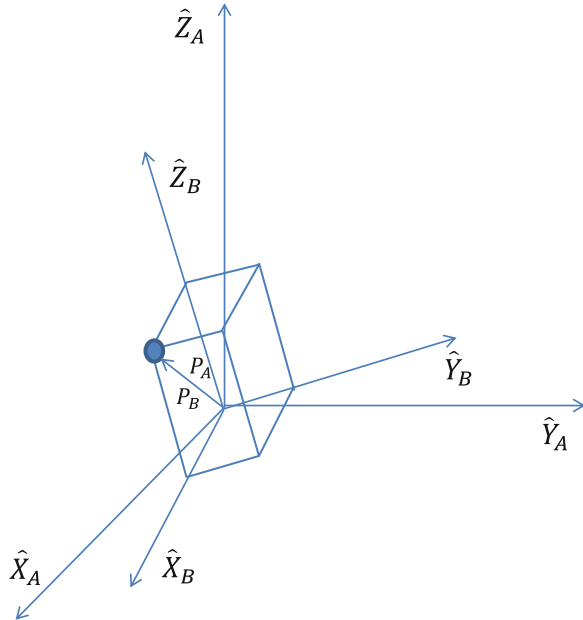
Let us consider a case where only the rotation took place between two coordinate systems (see Fig. 3.4). There is no translation between the origins of two systems. Then the question will be again, “how do we represent an already defined vector ${}^B P$ in {B} with respect to {A}?” The answer is to use a rotational matrix introduced in Sect. 3.1. Once the rotation matrix, ${}^B A R$, whose columns are the unit vectors of {B} defined in {A} is known, then the description of the orientation of {B} is known relative to {A}. Therefore, the vector P_A is simply a vector P_B whose components are the projections of vector components along the axes in {B} onto the axial directions of the frame {A}.

Now, let us define a projection of each axis of {B} on x -axis of {A} by cosine of angles (r_{11} to r_{31} in the equation below) between each axis of {B} relative to the x -axis of {A}.

$${}^A \hat{X}_B = \begin{bmatrix} r_{11} \cdot \hat{X}_B \\ r_{21} \cdot \hat{Y}_B \\ r_{31} \cdot \hat{Z}_B \end{bmatrix} \quad (3.10)$$

Again, \hat{X}_B in (3.10) is a unit vector of the x -axis of the frame {B}, and r_{11} , for example, is the cosine or dot product of the x -axis of {B} and the x -axis of {A}. Therefore, $r_{11} \cdot \hat{X}_B$ is simply a projection of the x -axis of {B} projected on the x -axis of {A}. If we add all of the projections of unit vectors along x , y , z axes of {B} on

Fig. 3.4 Rotational mapping between frames



x -axis of $\{A\}$, that becomes magnitude of the vector ${}^B\mathbf{P}$ along the x -axis of $\{A\}$, which is the x component of ${}^A\mathbf{P}$. Therefore, the projection of the vector, ${}^B\mathbf{P}$, on $\{A\}$ can be obtained by the equation below.

$${}^A\mathbf{P}_x = {}^A\hat{\mathbf{X}}_B \cdot {}^B\mathbf{P}$$

Since ${}^A\hat{\mathbf{X}}_B$ and ${}^B\mathbf{P}$ are both column vectors, we replace ${}^A\hat{\mathbf{X}}_B$ with ${}^B\hat{\mathbf{X}}_A^T$ because they are equal as shown in the Sample Problem 3.1, thus,

$${}^A\mathbf{P}_x = {}^B\hat{\mathbf{X}}_A^T \cdot {}^B\mathbf{P} \tag{3.11}$$

If we apply the same sequence to the projections along the y and z axes, then we obtain two more vector equations as below.

$${}^A\mathbf{P}_y = {}^B\hat{\mathbf{Y}}_A^T \cdot {}^B\mathbf{P} \tag{3.12}$$

$${}^A\mathbf{P}_z = {}^B\hat{\mathbf{Z}}_A^T \cdot {}^B\mathbf{P} \tag{3.13}$$

Notice from (3.3) that the vectors ${}^B\hat{\mathbf{X}}_A$, ${}^B\hat{\mathbf{Y}}_A$, and ${}^B\hat{\mathbf{Z}}_A$ together form a rotational matrix of ${}^B\mathbf{R}_A^T$. Therefore, all three equations from (3.11) to (3.13) become a simple equation such as;

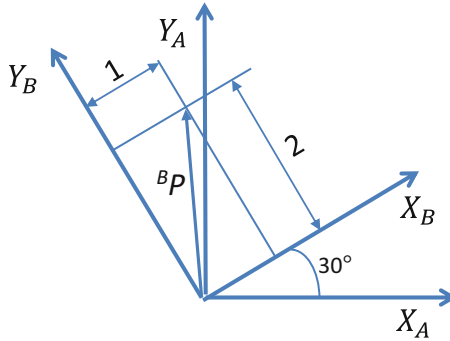
$${}^A\mathbf{P} = {}^B\mathbf{R}_A^T \cdot {}^B\mathbf{P} \tag{3.14}$$

Now, since ${}^B_A R^T = {}^A_B R^B$ from the Sample Problem 3.1, the equation can be rewritten as;

$${}^A P = {}^A_B R^B \cdot {}^B P \tag{3.15}$$

Sample Problem 3.2

The figure shown below has a frame {B} that is rotated relative to frame {A} around Z-axis by 30°.



Z-axis is pointing out of the page following the right-hand rule. For a given ${}^B P$ vector such that;

$${}^B P = \begin{bmatrix} 1.0 \\ 2.0 \\ 0.0 \end{bmatrix},$$

Find the vector ${}^A P$.

Solution

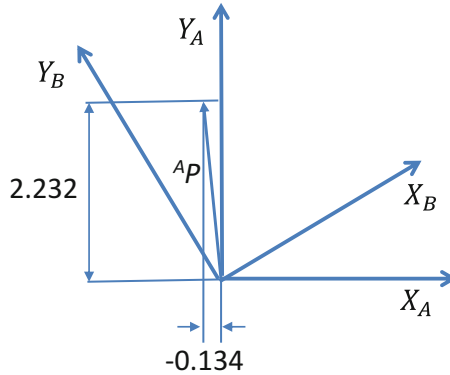
Since, ${}^A P = {}^A_B R^B \cdot {}^B P$, if we calculate ${}^A_B R$,

$$\begin{aligned} {}^A_B R &= \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix} \\ &= \begin{bmatrix} \cos 30 & \cos (90 + 30) & \cos 90 \\ \cos (90 - 30) & \cos 30 & \cos 90 \\ \cos 90 & \cos 90 & \cos 0 \end{bmatrix} = \begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Therefore, the vector ${}^A\mathbf{P}$ becomes;

$${}^A\mathbf{P} = \begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1.0 \\ 2.0 \\ 0.0 \end{bmatrix} = \begin{bmatrix} -0.134 \\ 2.232 \\ 0.000 \end{bmatrix}$$

The resultant vector ${}^A\mathbf{P}$ is depicted in the figure below for verification.



3.4 General Transformation Mapping

When a mapping involves both translation and rotation, then the general transformation mapping has to be used. The general transformation mapping is obtained by adding (3.9) and (3.15) as below.

$${}^A\mathbf{P} = {}^A_B\mathbf{R} \cdot {}^B\mathbf{P} + {}^A\mathbf{P}_{\text{BORG}} \tag{3.16}$$

In order to simplify the general transformation in (3.16), we introduce a transformation matrix, T such that,

$${}^A\mathbf{P} = {}^A_B\mathbf{R} \cdot {}^B\mathbf{P} + {}^A\mathbf{P}_{\text{BORG}} = {}^A_B\mathbf{T} \cdot {}^B\mathbf{P} \tag{3.17}$$

In order to make the matrix T a homogeneous transformation matrix, the following 4×4 configuration is used.

$$\begin{bmatrix} {}^A\mathbf{P} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} {}^A_B\mathbf{R} & {}^A\mathbf{P}_{\text{BORG}} \\ \mathbf{000} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} {}^B\mathbf{P} \\ \mathbf{1} \end{bmatrix} \tag{3.18}$$

In (3.18), a “1” is added as the last element of the 4×1 matrix. In addition, a row of “[0 0 0 1]” is added as the last row of the 4×4 matrix to make it a homogeneous transformation matrix. As a result, the matrix T allows us to combine rotation and translation into one single matrix.

General Transformation Matrix

A general transformation matrix, ${}^A_B T$, transforms a vector in the frame {B} to a vector in the frame {A}. Likewise, a general transformation matrix, ${}^B_A T$, transforms a vector in the frame {A} to a vector in the frame {B}.

Although ${}^A_B T$ transforms a vector in the frame of {B} to a vector in the frame of {A}, it expresses the general coordinate transformation of the frame {A} to the frame {B}.

Sample Problem 3.3

For a given vector, ${}^B P$, below, if a frame {B} is rotated relative to {A} about Z-axis by 30° , and translated by 10 units in X_A and 5 units in Y_A , describe the matrix T for the general transformation between {A} and {B}. In addition, obtain the resultant vector ${}^A P$ by the general transformation.

$${}^B P = \begin{bmatrix} 3.0 \\ 7.0 \\ 0.0 \\ 1 \end{bmatrix}$$

Solution

Since ${}^A_B T^B = \begin{bmatrix} {}^A_B R & {}^A P_{\text{BORG}} \\ \mathbf{000} & \mathbf{1} \end{bmatrix}$, we need to define ${}^A_B R$, and ${}^A P_{\text{BORG}}$.

Using the rotational matrix ${}^A_B R$ from the Sample Problem 3.2, ${}^A_B R$, and ${}^A P_{\text{BORG}}$ are defined as below.

$${}^A_B R = \begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad {}^A P_{\text{BORG}} = \begin{bmatrix} 10 \\ 5.0 \\ 0.0 \end{bmatrix}$$

Therefore the general transformation matrix T becomes;

$${}^A_B T^B = \begin{bmatrix} 0.866 & -0.5 & 0 & 10 \\ 0.5 & 0.866 & 0 & 5.0 \\ 0 & 0 & 1 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using the matrix ${}^A_B T$, the new vector, ${}^A P$, in frame $\{A\}$ becomes,

$${}^A P = \begin{bmatrix} 9.098 \\ 12.562 \\ 0 \\ 1 \end{bmatrix}$$

3.5 Transformation Arithmetic

Since the general transformation matrix T is homogeneous and linear, the following three relationships stand.

1. Compound transformation

For two transformations which are defined as below,

$${}^A P = {}^A_B T \cdot {}^B P, \quad {}^B P = {}^B_C T \cdot {}^C P,$$

the vector, ${}^A P$ can be rewritten as a chain transformation starting with ${}^C P$ such that,

$${}^A P = {}^A_B T \cdot ({}^B_C T \cdot {}^C P) = {}^A_C T \cdot {}^C P,$$

where

$${}^A_C T = {}^A_B T \cdot {}^B_C T. \quad (3.19)$$

2. Inverse transformation

For a general transformation T such that,

$${}^A P = {}^A_B T \cdot {}^B P,$$

Multiplication of the inverse of ${}^A_B T$ on both sides will make the above equation as below.

$${}^A_B T^{-1} \cdot {}^A P = {}^A_B T^{-1} \cdot {}^A_B T \cdot {}^B P$$

Since ${}^A_B T^{-1} \cdot {}^A_B T$ become I matrix,

$${}^A_B T^{-1} \cdot {}^A P = {}^B P$$

or

$${}^B\mathbf{P} = {}^A\mathbf{T}^{-1} \cdot {}^A\mathbf{P}$$

Since ${}^B\mathbf{P} = {}^B\mathbf{T} \cdot {}^A\mathbf{P}$ by definition, the following relationship is true.

$${}^B\mathbf{T} = {}^A\mathbf{T}^{-1} \tag{3.20}$$

3. Transform equation

For multiple transformations, the following relationships are all true.

$${}^U\mathbf{T} = {}^U\mathbf{T} \cdot {}^A\mathbf{T} = {}^U\mathbf{T} \cdot {}^A\mathbf{T} \cdot {}^B\mathbf{T}$$

Therefore the following relationship is also true.

$${}^U\mathbf{T} \cdot {}^A\mathbf{T} = {}^U\mathbf{T} \cdot {}^A\mathbf{T} \cdot {}^B\mathbf{T} \tag{3.21}$$

3.6 General Form of Rotation

As the name implies, a 3D modeling in general involves translation and rotation around all three axes in 3D space. The general approach to describe rotations around all three axes is to perform rotation around each axis in sequence. For instance, one can specify a rotation around x -axis, followed by y -axis and then by z -axis. We call this sequence as x - y - z fixed angle rotation (Fig. 3.5).

For instance, an angle by γ around x -axis, and an angle by β around y -axis, and an angle by α around z -axis result in a sequence of x - y - z fixed angle rotation described as below.

$${}^A\mathbf{R}_{xyz}(\gamma, \beta, \alpha) = \mathbf{R}_z(\alpha) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_x(\gamma) \tag{3.22}$$

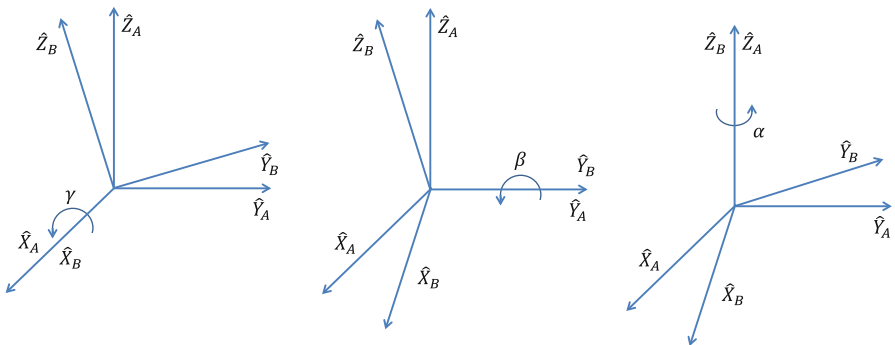


Fig. 3.5 X-Y-Z fixed angle rotation

By using the definition in (3.5), (3.22) becomes,

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & \cos(90 + \alpha) & \cos 90 \\ \cos(90 - \alpha) & \cos \alpha & \cos 90 \\ \cos 90 & \cos 90 & \cos 0 \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Likewise, $R_y(\beta)$ and $R_x(\gamma)$ are defined as below.

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}, \quad R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix}$$

Therefore, the resulting rotation around all three axes become,

$${}^A_B R_{xyz}(\gamma, \beta, \alpha) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}, \quad (3.23)$$

where “c” stands for cosine and “s” stands for sin. Notice that the sequence of the x - y - z fixed angle rotation starts from the right to the left as shown in (3.22) so that the order of rotational angle and the order of rotational matrices are in opposite order. This sometimes causes confusion, and as a remedy, Z - Y - X Euler angle has been proposed. In the Euler angle notation, each rotation is performed about an axis of the moving system {B} rather than a fixed system {A}. In other words, each angle of rotation is defined by the change from the previously moved system, not from the originally fixed system. Therefore, the rotational matrix around all three axes becomes as follows (Fig. 3.6).

$${}^A_B R_{z'y'x'}(\alpha, \beta, \gamma) = {}^A_{B'} R(\alpha) \cdot {}^{B'}_{B''} R(\beta) \cdot {}^{B''}_{B'''} R(\gamma) \quad (3.24)$$

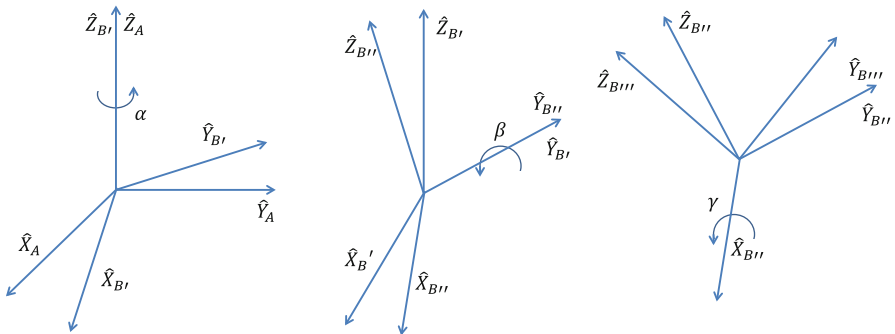


Fig. 3.6 Z-Y-X Euler angle rotation

The equation of each rotational matrix with the $z-y-x$ Euler angle rotation becomes as below.

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}$$

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix}$$

Therefore, the resulting rotation around all three axes becomes,

$$\begin{aligned} {}^A_B R_z y' x'(\alpha, \beta, \gamma) &= R_z(\alpha) R_y(\beta) R_x(\gamma) \\ &= \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}, \end{aligned} \quad (3.25)$$

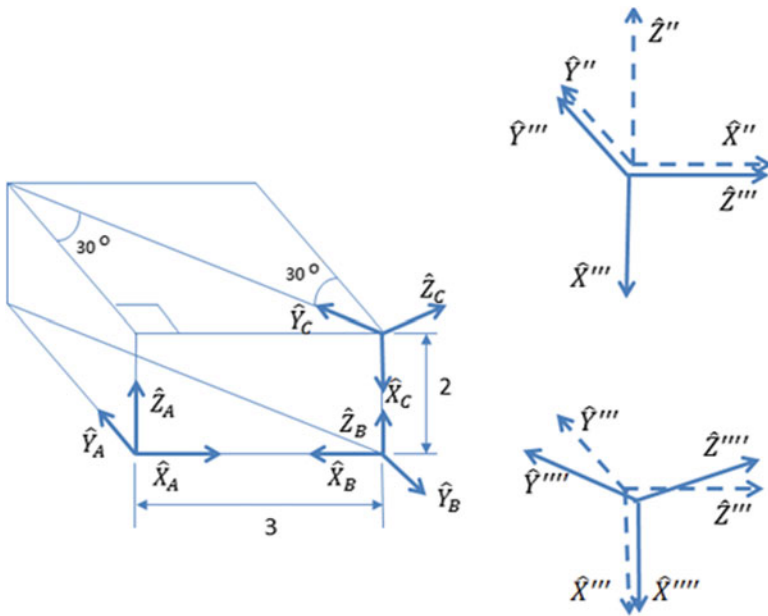
Notice that (3.23) and (3.25) are identical. Therefore, the resulting rotational matrix of the $z-y-x$ Euler angle is exactly the same as that from the rotation taken in the opposite direction of $x-y-z$ fixed angle. The $z-y-x$ Euler angle notation is more popular in industry, especially in CAD and robotics industries because of easier understanding by the moving frames and easier implementation in real-world applications.

Rotational Matrix

- Three rotations taken about fixed axes yield the same final orientation matrix as the same three rotations taken in opposite order by the axes of the moving frame in $z-y-x$ Euler angle rotation.
- Although ${}^A_B R$ transforms a vector in the frame of {B} to a vector in the frame of {A}, it expresses the rotational transformation of the frame {A} to the frame {B}.
- The prime symbol represents moving frames

Sample Problem 3.4

For given three frames in the figure below, define ${}^A_C T$ and ${}^A_B T$.



Solution

${}^A_C T$ is a general coordinate transformation matrix that transforms a vector in the frame of {C} to the frame of {A}. However, it expresses the coordinate transformation of the frame {A} to the frame {C}. By using the z-y-x moving coordinate, the frame {A} has to move along the \hat{X}_A by 3 units followed by 2 units along the \hat{Z}_A , followed by 90° rotation around $\hat{Y}_{A'}$, followed by -30° rotation around $\hat{X}_{A''}$. In mathematical formula,

$$\begin{aligned}
 {}^A_C T &= D_x(3) \cdot D_z(2) \cdot R_y(90) \cdot R_x(-30) \\
 &= \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c90 & 0 & s90 & 0 \\ 0 & 1 & 0 & 0 \\ -s90 & 0 & c90 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c-30 & -s-30 & 0 \\ 0 & s-30 & c-30 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & -0.5 & 0.866 & 3 \\ 0 & 0.866 & 0.5 & 0 \\ -1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

$$\begin{aligned}
 {}^A_B T &= D_{z'}(3) \cdot R_{z'}(180) \\
 &= \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c-180 & -s180 & 0 & 3 \\ s180 & c180 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} -1 & 0 & 0 & 3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

3.7 Transformation of a 3D Model

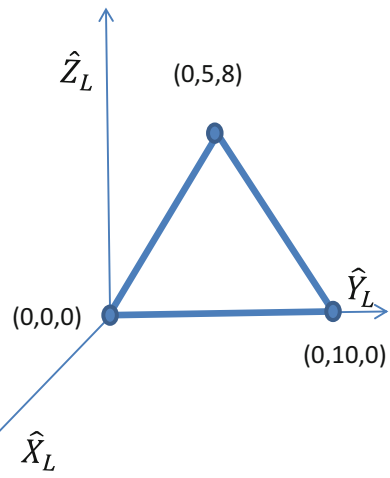
In this section, we investigate a question as to how a 3D shape is expressed on a screen by translational and rotational transformation. Let's assume that there is a triangle created as a part of an engineering design. Figure 3.7 shows a triangle created in a local frame, "L".

Three points that constitute the triangle are defined and given as;

$${}^L P_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad {}^L P_2 = \begin{bmatrix} 0 \\ 10 \\ 0 \end{bmatrix}, \quad {}^L P_3 = \begin{bmatrix} 0 \\ 5 \\ 8 \end{bmatrix},$$

Suppose the design engineer wants to manipulate the triangle following the transformation sequence as below.

Fig. 3.7 Triangle in 3D space



Translate it by (10, 20, 30) along x , y , and z axes respectively

Rotate it by 45° around z' -axis

Rotate it by 30° around y'' -axis

Rotate it by 60° around x''' -axis respectively

Then the rotational transformation matrix will be

$$\begin{aligned} {}^G_L R_{z'y'x'}(45, 30, 60) &= \begin{bmatrix} c45c30 & c45s30s60 - s45c60 & c45s30c60 + s45s60 \\ s45c30 & s45s30s60 + c45c60 & s45s30c60 - c45s60 \\ -s30 & c30s60 & c30c60 \end{bmatrix} \\ &= \begin{bmatrix} 0.612 & -0.047 & 0.789 \\ 0.612 & 0.659 & -0.435 \\ -0.5 & 0.75 & 0.433 \end{bmatrix}. \end{aligned}$$

Therefore, the complete transformation matrix will become as below.

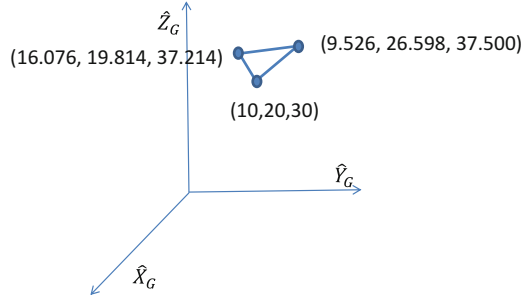
$${}^G_L T = \begin{bmatrix} 0.612 & -0.047 & 0.789 & 10 \\ 0.612 & 0.659 & -0.435 & 20 \\ -0.5 & 0.75 & 0.433 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since the complete transformation matrix is defined for the given manipulation sequence, each coordinate of the triangle after transformation can be calculated by the transformation matrix, ${}^G_L T$, with respect to the global coordinate, "G," such that,

$$\begin{aligned} {}^G P_1 &= {}^G_L T \cdot {}^L P_1 = \begin{bmatrix} 0.612 & -0.047 & 0.789 & 10 \\ 0.612 & 0.659 & -0.435 & 20 \\ -0.5 & 0.75 & 0.433 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 1 \end{bmatrix} \\ {}^G P_2 &= {}^G_L T \cdot {}^L P_2 = \begin{bmatrix} 0.612 & -0.047 & 0.789 & 10 \\ 0.612 & 0.659 & -0.435 & 20 \\ -0.5 & 0.75 & 0.433 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 10 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 9.526 \\ 26.598 \\ 37.500 \\ 1 \end{bmatrix} \\ {}^G P_3 &= {}^G_L T \cdot {}^L P_3 = \begin{bmatrix} 0.612 & -0.047 & 0.789 & 10 \\ 0.612 & 0.659 & -0.435 & 20 \\ -0.5 & 0.75 & 0.433 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 5 \\ 8 \\ 1 \end{bmatrix} = \begin{bmatrix} 16.076 \\ 19.814 \\ 37.214 \\ 1 \end{bmatrix} \end{aligned}$$

The resulting triangle after transformation is shown in the global coordinate (see Fig. 3.8).

Fig. 3.8 Transformed triangle shown in global coordinate



The same method can be used for visualization purpose. For instance, if one wants to see the object from a different angle, he or she can generate the rotational angle around each axis and plot the projection of the 3D shape on x - y plane to display the result on the screen after transformation [6].

Sample Problem 3.5

Draw the projection of the triangle in Fig. 3.7 on y - z plane and rotate the triangle by 45° around z -axis and plot it on the y - z plane of the global coordinate.

Solution

Since ${}^G_L T = R_z(45)$,

$${}^G_L T = \begin{bmatrix} c45 & -s45 & 0 & 0 \\ s45 & c45 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.707 & -0.707 & 0 & 0 \\ 0.707 & 0.707 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Therefore,

$${}^G P_1 = {}^G_L T \cdot {}^L P_1 = \begin{bmatrix} 0.707 & -0.707 & 0 & 0 \\ 0.707 & 0.707 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$${}^G P_2 = {}^G_L T \cdot {}^L P_2 = \begin{bmatrix} 0.707 & -0.707 & 0 & 0 \\ 0.707 & 0.707 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 10 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -7.071 \\ 7.071 \\ 0 \\ 1 \end{bmatrix}$$

$${}^G P_3 = {}^G_L T \cdot {}^L P_3 = \begin{bmatrix} 0.707 & -0.707 & 0 & 0 \\ 0.707 & 0.707 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 5 \\ 8 \\ 1 \end{bmatrix} = \begin{bmatrix} -3.535 \\ 3.535 \\ 8 \\ 1 \end{bmatrix}$$

The result is shown in Fig. 3.9.

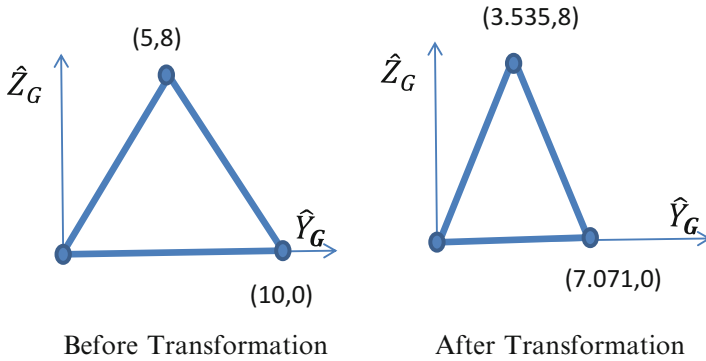


Fig. 3.9 Transformation of a triangle

3.8 Perspective Projection

To make a 3D model more realistic on a computer screen, people often use perspective projection, by which a 3D model is represented by a virtual vanishing point. Mathematically, perspective projection is achieved by applying a weighting factor to the entire x and y coordinate values of an object in regard to the relative distance of z to the virtual vanishing point. Let's assume that we set the screen coordinate with u , v , and w , while the corresponding data point of a 3D model is represented by x , y , and z coordinates (see Fig. 3.10). Since, an operator will watch the 3D model projected on the screen by u , v screen coordinates, we can achieve the effect of perspective projection by adjusting the u , v coordinates by the depth of each point of model from the observer.

In order to achieve the perspective projection on a computer screen, a simple linear relationship between actual location versus projected location needs to be established by the geometry in Fig. 3.10 (right). For instance, a dot closer to the observer will be projected on a higher location on the screen, while a dot far away will be projected at a lower location, thus an object in close vicinity will look larger on the screen. If we use the triangle in Fig. 3.10 (right), the following balance equation is true.

$$Z_e : v = Z_e - Z : y \tag{3.26}$$

Therefore, the following is also true.

$$v = \frac{Z_e}{Z_e - Z} y = \left(\frac{1}{1 - Z/Z_e} \right) y \tag{3.27}$$

If we define a projection factor g such that,

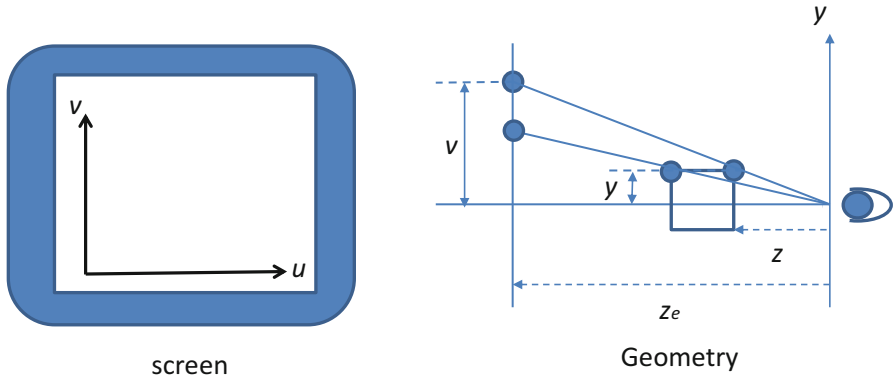


Fig. 3.10 Perspective projection

$$\begin{aligned}
 x &= g \cdot u \\
 y &= g \cdot v \\
 z &= g \cdot w
 \end{aligned}
 \tag{3.28}$$

where,

$$g = 1 - \frac{1}{Z_e} \cdot Z,
 \tag{3.29}$$

then, we can define a transformation matrix between $x, y, z,$ and u, v, w such that:

$$\begin{bmatrix} u \\ v \\ w \\ g \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{Z_e} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}
 \tag{3.30}$$

Therefore, a general coordinate transformation matrix in (3.25) can be incorporated in the above equation so that,

$$\begin{bmatrix} u \\ v \\ w \\ g \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{Z_e} & 1 \end{bmatrix} \cdot \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma & 0 \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma & 0 \\ -s\beta & c\beta s\gamma & c\beta c\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (3.31)$$

Notice that actual coordinates of u and v will be later obtained by $u \cdot g, v \cdot g$ respectively for perspective view display.

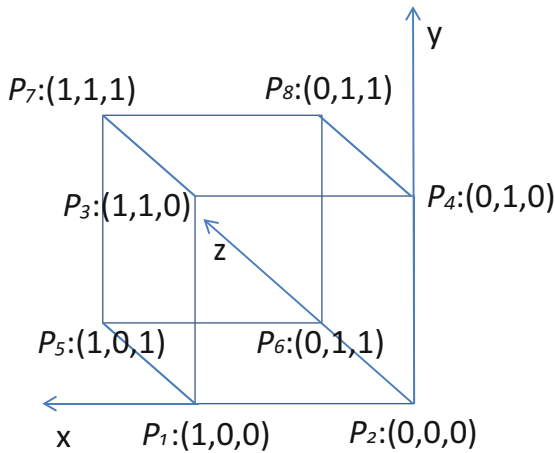
Sample Problem 3.6

Draw the projection of a cube in the figure below on $u-v$ plane with the z_e value of 10.

Solution

Since $z_e = 10$, the transformation matrix becomes,

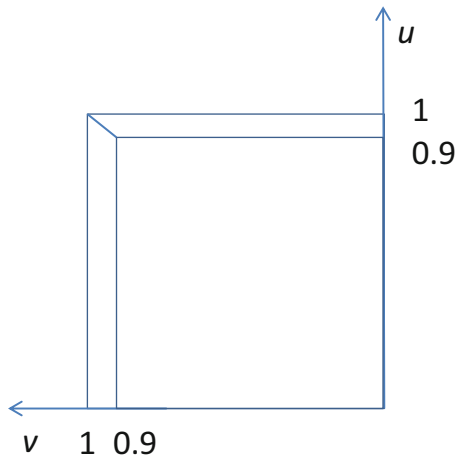
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{10} & 0 \end{bmatrix}$$



Therefore, each point of the cube on $u-v$ screen will be calculated as below.

$$\begin{aligned}
 P_1 : g &= 1 - \frac{1}{10} \cdot 0 = 1, & \begin{bmatrix} u \cdot g \\ v \cdot g \end{bmatrix} &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
 P_2 : g &= 1 - \frac{1}{10} \cdot 0 = 1, & \begin{bmatrix} u \cdot g \\ v \cdot g \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
 P_3 : g &= 1 - \frac{1}{10} \cdot 0 = 1, & \begin{bmatrix} u \cdot g \\ v \cdot g \end{bmatrix} &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
 P_4 : g &= 1 - \frac{1}{10} \cdot 0 = 1, & \begin{bmatrix} u \cdot g \\ v \cdot g \end{bmatrix} &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
 P_5 : g &= 1 - \frac{1}{10} \cdot 1 = 0.9, & \begin{bmatrix} u \cdot g \\ v \cdot g \end{bmatrix} &= \begin{bmatrix} 0.9 \\ 0 \end{bmatrix} \\
 P_6 : g &= 1 - \frac{1}{10} \cdot 1 = 0.9, & \begin{bmatrix} u \cdot g \\ v \cdot g \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
 P_7 : g &= 1 - \frac{1}{10} \cdot 1 = 0.9, & \begin{bmatrix} u \cdot g \\ v \cdot g \end{bmatrix} &= \begin{bmatrix} 0.9 \\ 0.9 \end{bmatrix} \\
 P_8 : g &= 1 - \frac{1}{10} \cdot 1 = 0.9, & \begin{bmatrix} u \cdot g \\ v \cdot g \end{bmatrix} &= \begin{bmatrix} 0 \\ 0.9 \end{bmatrix}
 \end{aligned}$$

Therefore, the projection of the cube on $u-v$ plane will be as shown below.



3.9 3D Modeling Schemes

Various 3D modeling techniques are invented and available in design industry. Recently, advanced modeling techniques are invented to represent realistic entities in nature especially for entertainment industry. The majority of 3D modeling techniques are based on three basic modeling schemes: wireframe geometry, surface modeling, and solid modeling.

Combinations of any of two or all of three are used in many design processes. We focus our discussion in three basic modeling schemes and study advantages and disadvantages to understand the optimal use of each modeling scheme in specific applications. Some examples of each modeling scheme are depicted in Fig. 3.11.

3.9.1 Wireframe Geometry

Wireframe modeling is the most fundamental 3D modeling scheme. Most of the other modeling schemes are evolved from the wireframe geometry. Therefore, understanding the fundamentals of wireframe model is of importance in 3D modeling. In wireframe scheme, geometry is defined as a series of lines and curves representing the edges of an object. Wireframe modeling has several strengths. First it is computationally the most straightforward technique. Since a 3D model is

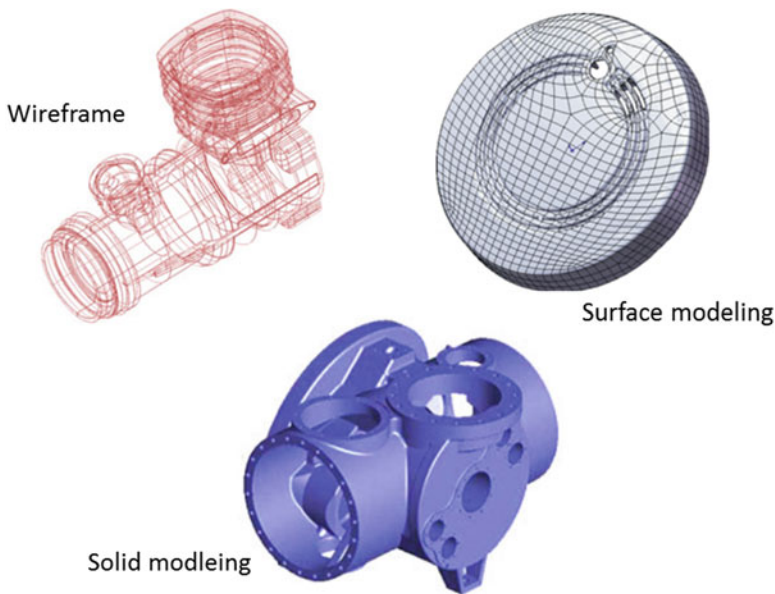


Fig. 3.11 Three basic 3D modeling schemes [1–3]

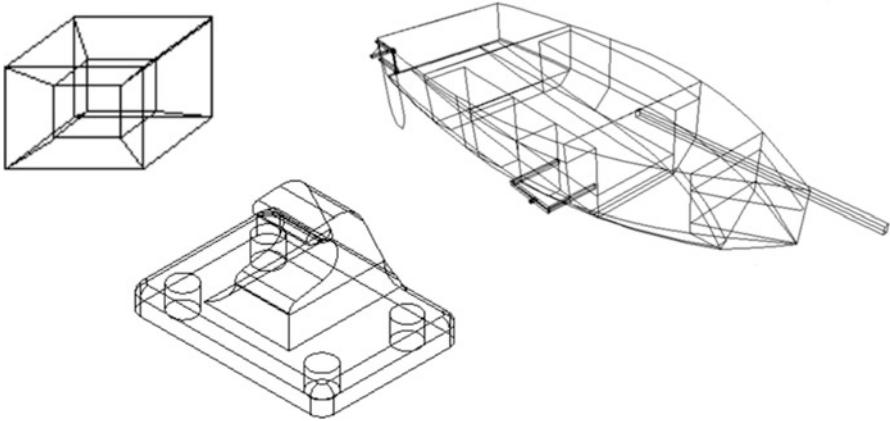
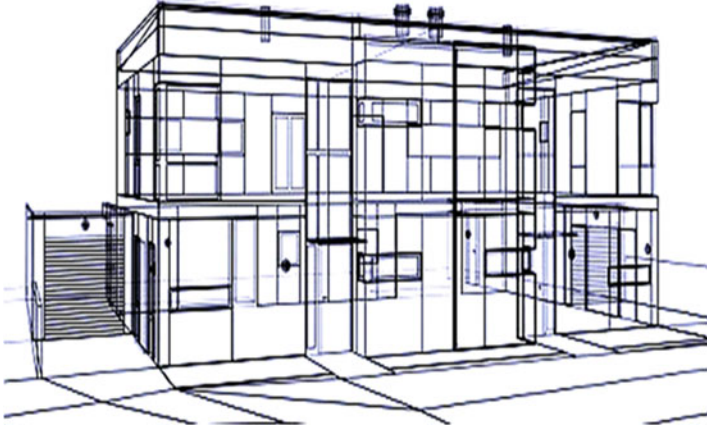


Fig. 3.12 Wireframe models

represented by a collection of wires and curves, the entire 3D model can be reproduced rapidly on a screen. In addition, it is the most economical in terms of time and memory requirements. Thanks to today's computational power of personal computers and common graphical accelerators, this may not be a big advantage anymore. However, if a massive collection of objects has to be handled for a complex design, then, it still has a merit over other 3D modeling schemes, especially for real-time animation. In order to make it a useful scheme, however, hidden lines of a wireframe model have to be removed because of the ambiguity in representation.

Hidden line removal is computationally intensive by an old fashioned computer, but not too much of a burden for modern computers unless an object is extremely complex. For instance, two drawings in Fig. 3.12 introduce confusion as to which part is empty and which part is solid. Without hidden line removal, the drawing causes difficulties in understating the complete shape of an object. In addition, the wireframe model alone is deficient in pictorial representation since it can't express any face but only expresses edges of an object. Therefore other engineers will have difficulties in interpretation of a complex wireframe model. Another weakness of the wireframe model is that it is limited in the ability to calculate mechanical properties such as surface area, volume, weight calculations, or geometrical intersections for overlapping or tolerance calculations. Without knowing the empty area of the part in the drawing, it is impossible to calculate volume or weight of the object.

Another example of the wireframe scheme is shown below. Obviously it is a drawing of a building. Since all of the wires are represented as a collection of wires without hidden lines removed, identification of feature locations such as doors or windows is very challenging.



Wireframe Model: Strength and Weakness

- Strength
 - Computationally the most straightforward
 - Most economical in terms of time and memory requirements
 - Geometry is defined as a series of lines and curves representing the edges
 - Useful for visualization of simple shapes, animation of simple mechanism
- Weakness
 - Ambiguity in representation (no merit without hidden line removal – computationally intensive)
 - Deficiencies in pictorial representation (difficulty in complex model interpretation)
 - Limitation in the ability to calculate mechanical properties, or geometric intersections
 - Limited scheme as a basis for manufacturing or analysis

3.9.2 Surface Representation

As a remedy to the deficiency of the wireframe scheme, surface representation scheme has been invented. In surface representation scheme, a 3D object is represented by specifying some or all of the surfaces on the component. Technically, each object is constructed from surface edges and curves on the surface, therefore the surface modeling is mixed with or developed from wireframe model. Although it is

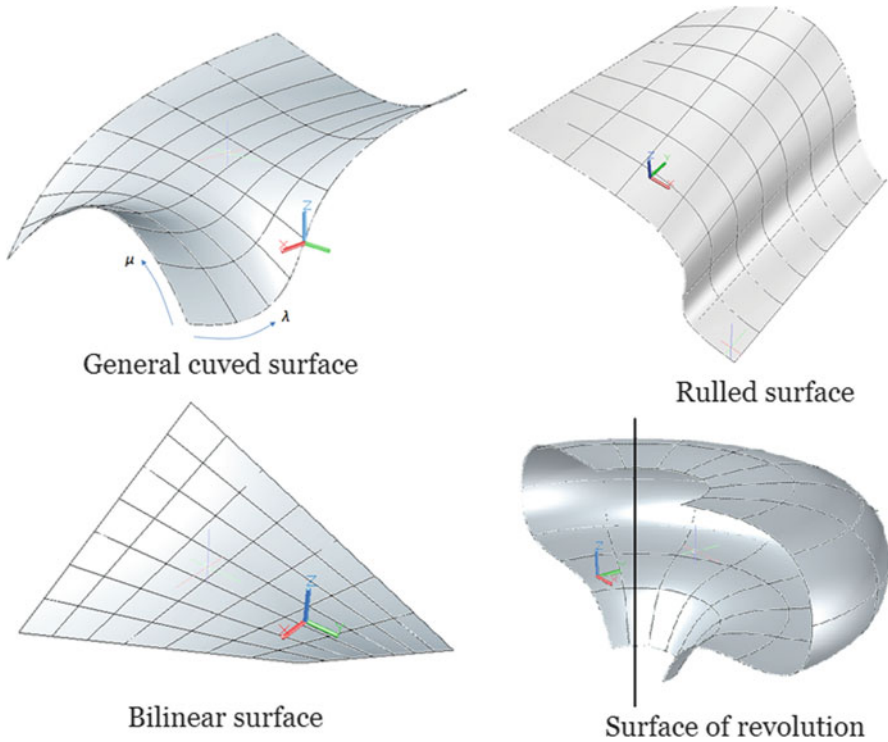


Fig. 3.13 Basic surface modeling technique

evolved from the wireframe scheme, the surface modeling scheme offers significant advantages over wireframe models in the links to manufacture and to engineering analysis. Several basic surface modeling techniques are depicted in Fig. 3.13, while some advanced surface modeling techniques are shown in Fig. 3.14.

Surface representation is more computationally demanding than wireframe and it requires advanced technical skills in construction and use of the created models. In addition, similar to the wireframe model, there is no merit without hidden surfaces removal, which is computationally very intensive for complex objects. Although surface representation has multiple advantages over the wireframe scheme, it still falls short of critical functions as a modeling scheme for design. First it does not establish connectivity information between surfaces or it is difficult to create such information. Second, objects are represented as a simple collection of surfaces. Therefore, no higher level information about the solid object is retained in the surface model data. For instance, if one created a human body by surface representation, the final outcome will be a simple collection of different surfaces with no connectivity information needed to constitute a human body.

In other words, it is a collection of surfaces just to represent different parts and shapes with different surface equations without knowing which surface belongs to

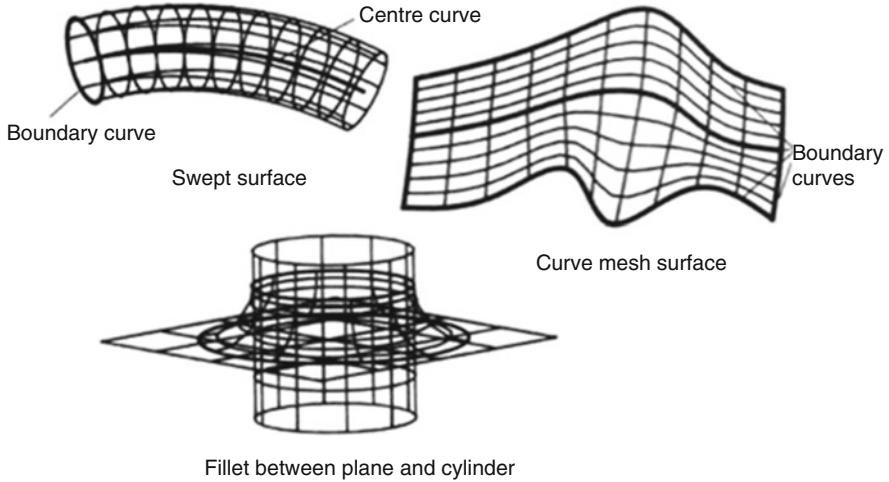
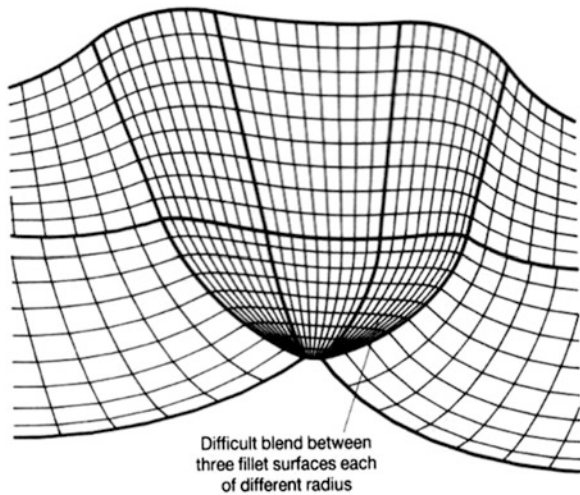


Fig. 3.14 Advanced surface modeling technique

Fig. 3.15 Difficult blending of multiple surfaces



what part of human body or how each surface is connected together. This shortage led to a new invention of the 3D representation, called solid modeling.

In addition to all of the aforementioned weaknesses of the surface modeling, another difficulty arises when the surface modeling is used to create a complex shape or blending of multiple surfaces. In Fig. 3.15, one example is depicted to demonstrate the difficulties of blending several surfaces to create a 3D object. The difficulty arises when all ten surfaces have to be connected in a way that they form a smooth surface without any discontinuity in terms of connection at all points as well as the corresponding tangents and curvatures from each surface at the point of contact have to match each other. It is a challenging task to come up with



Fig. 3.16 Wireframe versus surface model [4]

exact surface equations that satisfy all of the matching requirements at the point of contact along the edges between surfaces.

The figure above is an example of comparing wireframe versus surface model (Fig. 3.16). While vague in details by the wireframe, surface model represents the shape precisely to the extent most of the features can be easily recognized.

Surface Model: Strength and Weakness

- Strength
 - Unambiguous representation of a 3D model
 - Significant advantages over wireframe models in the links to manufacture and to engineering analysis
- Weakness
 - More computationally demanding than wireframe
 - More skill in construction and use
 - No merit without hidden surfaces removal (computationally very intensive)
 - No connectivity between surfaces
 - Objects are represented as a simple collection of surfaces (No higher level information about the solid object)
 - Some geometries are difficult to represent using surface modeling schemes

3.9.3 Solid Modeling

Solid modeling is a relatively modern 3D modeling scheme born of necessity for higher level representation of today's elegant and complex products. It is initiated by Woodwark and further advanced thereafter. Woodwark's proposal was about inventing a successful 3D modeling scheme to, first, generate a complete and unambiguous 3D parts. And second, the modeling scheme has to be appropriate for the world of engineering objects. Finally, it has to be practical to use with existing computers. Based on these three criteria, Woodwark proposed a solid modeling technique, which has two general ramifications: one is boundary representation and another is constructive solid geometry (CSG) method.

Woodwark's Proposal of a Successful Modeling Scheme

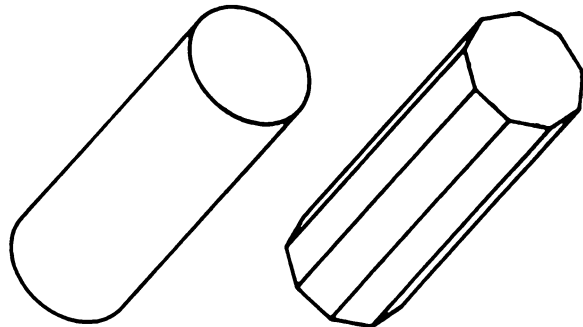
- Complete and unambiguous
- Appropriate for the world of engineering objects
- Practical for existing computers

3.9.3.1 Boundary Representation

Boundary representation, also known as B-rep, is one of the solid modeling techniques. B-rep, first, decomposes a 3D model into a collection of small faces. For instance, a solid cylinder can be decomposed into a collection of small faces as shown in Fig. 3.17. In B-rep scheme, due to the simplicity of the plane of each face, connectivity information is easily added in the data structure of a 3D model. In addition, identification of the solid side of each face can also be made clear. Because of the way it represents a 3D solid object, it sometimes called as a flat face representation.

The most important advantage of using the B-rep is that it stores information about the faces and edges of a model explicitly as an evaluated form. This means that outermost surfaces of a 3D model represented by B-rep are always identifiable

Fig. 3.17 Boundary representation of a solid cylinder



since not only the solid side of each facet is stored, but the location of each facet is also defined so all facets can be automatically prioritized in the order to represent each part of an object. This is an important feature not only for the visualization, but also for manufacturing.

3.9.3.2 Constructive Solid Geometry

Unlike the Boundary representation, in CSG, models are constructed as a combination of simple solid primitives (see Fig. 3.18). Set-theoretic model (Boolean operation) is applied on simple solid primitives to establish a complete model. For instance, the union between two primitives includes all points that belong to both objects in space. Intersection, however, only encompasses only points that are in common (see Fig. 3.19). Thanks to the well-established logic of the set theory, CSG models tend to be more robust, meaning it is less prone to numerical or computational errors. The logic behind the CSG allows us to take performance advantage of membership tests such as collision, overlapping parts, and inclusion, etc.

Although CSG solved many issues of previous 3D modeling schemes, it also has some drawbacks from the perspective of modern 3D design requirements. First, in CSG, models are stored as a combination of Boolean logic and locations in 3D, thus they are stored in an unevaluated form, meaning edges and faces (or outermost surfaces) from the combination of the primitives have to be computed when required. Because of such extra calculation burden, CSG has to suffer attendant performance penalty.

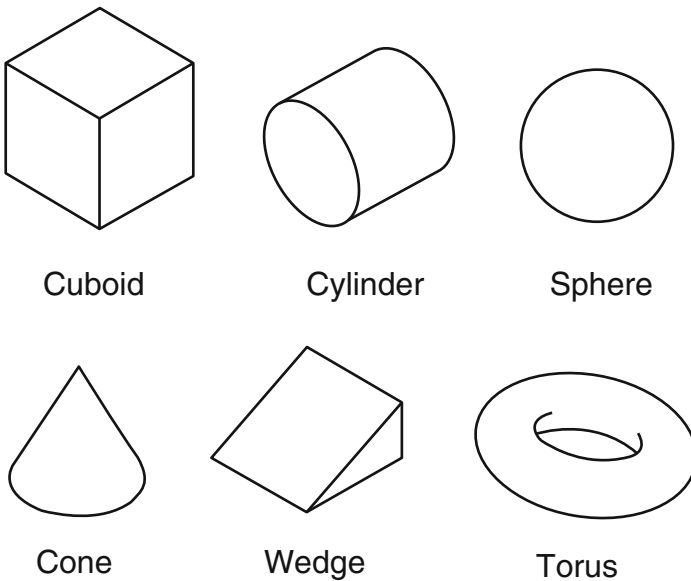
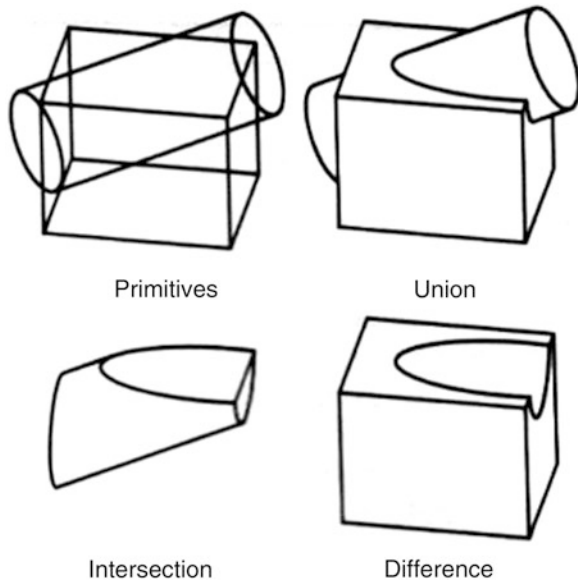


Fig. 3.18 Solid primitives

Fig. 3.19 Boolean operation with solid modeling



In addition, smooth combination between two primitives may be difficult or time consuming. For instance, fillet blends between two objects have to be either user defined or computed to generate a surface from the position and orientation information of two objects in space. But for the most part, modeling more complex forms often found in modern industry is impossible by using CSG. For instance, aerodynamic automobile body panels or irregular body shapes of creatures in nature may not be expressed efficiently by CSG technique.

Because of such deficiency of the CSG, hybrid modeling methods such as dual representation becomes more popular in industry. Dual representation is a hybrid of CSG and B-rep modeling techniques. CSG is used for approximation analysis such as collision detection, Boolean operations, and B-rep is used for quick display purposes because it is more flexible in visualization. In more recent trends, B-rep becomes more dominant than CSG because CSG is somewhat limited in visualization of modern product's geometry. Mixed approaches of B-rep with surface modeling and wireframe become more popular as well, thus nonhomogeneous approach and manifold modeling becomes dominant trends in 3D modeling (see Fig. 3.20).

Furthermore, with solid-surface hybrid modeling, engineers uses the Boolean operation to combine both solids and surfaces. A highly complex shape can be made by a hybrid model using a simple Boolean operation. For instance, a complex vehicle model is created by hybrid modeling with single Boolean operation in Fig. 3.21. In modern engineering, designers are facing more challenges as customers' demands for design diversify significantly. The innovative design methods such as solid-surface hybrid modeling are the answers.

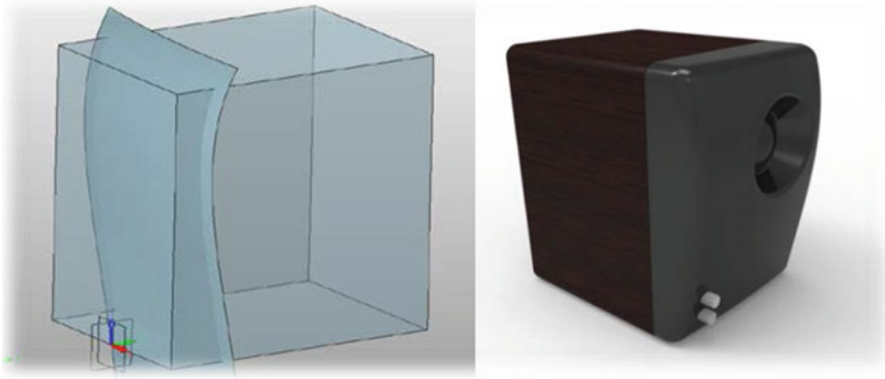
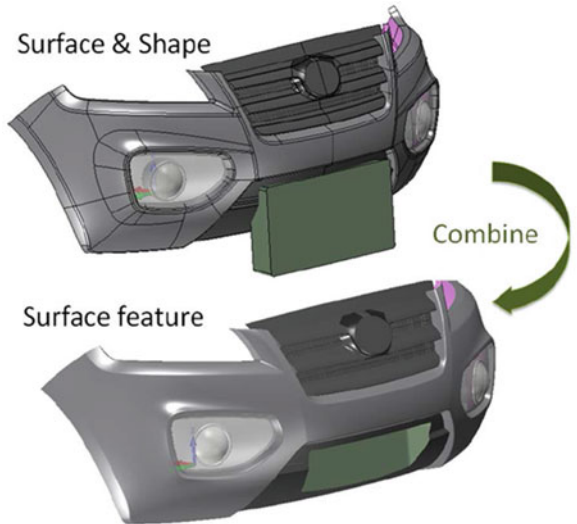


Fig. 3.20 Hybrid model between solid and surface model (cutting solid by surface model)

Fig. 3.21 Boolean operation by using solid to combine the surface [5]



In the future, more advanced technologies will be invented to overcome design challenges of modern society. The outcomes of the design in the future will be based on hybrid modeling of varied technologies such as wireframe, surface modeling, solid modeling, depending on applications and problem domains. As a designer, it is critical to understand current technologies, as well as to think out of the box for being creative to deal with design challenges in modern society.

Constructive Solid Geometry: Strength and Weakness

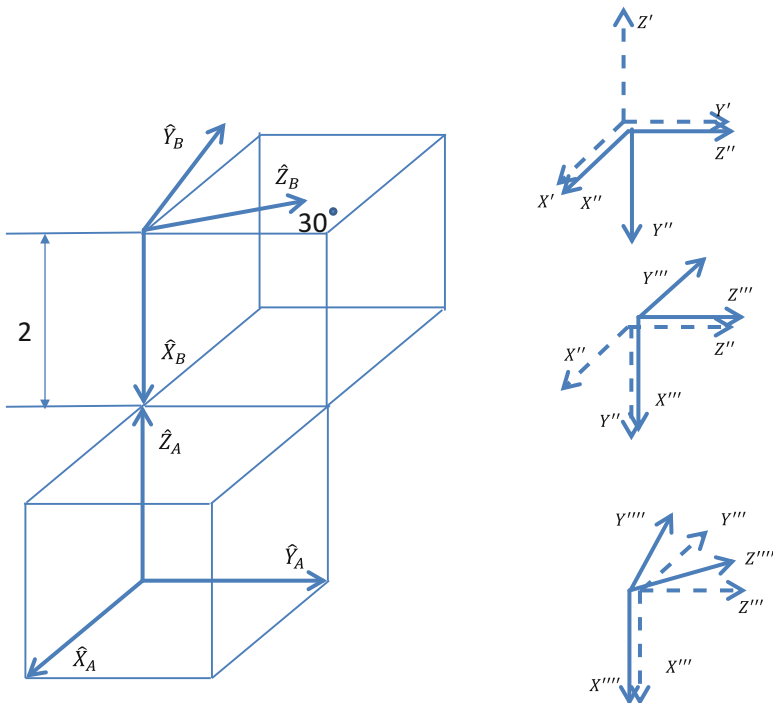
- Strength
 - Models are constructed as a combination of simple solid primitives
 - Set-theoretic model (Boolean operation)
 - CSG models tend to be more robust
 - Performance advantage for membership test
- Weakness
 - Models are stored in an unevaluated form

Edges and faces (or outermost surfaces) from the combination of the primitives have to be computed when required
Attendant performance penalty

 - Smooth combination between two primitives may be difficult or time consuming (fillet blends. . .)
 - More complex forms are impossible to model using CSG (automobile body panels. . .)

Exercise Problem 3.1

For given three frames in the figure below, define ${}^A_B T$.



Solution

$$\begin{aligned}
{}^A T &= D_z(2) \cdot R_{x'}(-90) \cdot R_{z'}(90) \cdot R_{x'}(-30) \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c-90 & -s-90 & 0 \\ 0 & s-90 & c-90 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c90 & -s90 & 0 & 0 \\ s90 & c90 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\quad \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c-30 & -s-30 & 0 \\ 0 & s-30 & c-30 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0.866 & 0.5 & 0 \\ 0 & -0.5 & 0.866 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

Exercise Problem 3.2

Write a Matlab program that transforms a dot (1, 2, 3) in x - y - z coordinate system for the following coordinate transformation.

1. Translate by (10, 20, 30) along x , y , and z axes.
2. Rotate by 30° around z' -axis.
3. Rotate by 50° around y' -axis.
4. Rotate 60° around x' -axis.

Solution

```

% define a dot
px=1
py=2
pz=3
% translation along x, y, z
x=10
y=20
z=30
% rotational angle
alpha=30
beta=50
gamma=60

```

```

% sin
sa=sin(alpa*3.141592/180)
sb=sin(beta*3.141592/180)
sg=sin(gamma*3.141592/180)
% cos
ca=cos(alpa*3.141592/180)
cb=cos(beta*3.141592/180)
cg=cos(gamma*3.141592/180)

t=[ca*cb ca*sb*sg-sa*cg ca*sb*cg+sa*sg x;sa*cb sa*sb*sg
+ca*cg sa*sb*cg-ca*sg y;-sb cb*sg cb*cg z;0 0 0 1]
p=[px;py;pz;1]

t*p

```

Exercise Problem 3.3

For the given four points that constitute a square on y - z plane, determine transformed square if the model has to be translated by (10, 20, 30) along x , y , and z axes respectively and rotated by 50° around z' -axis, followed by 80° around y' -axis, and 25° around x' -axis respectively.

$${}^L P_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, {}^L P_2 = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, {}^L P_3 = \begin{bmatrix} 0 \\ 5 \\ 5 \end{bmatrix}, {}^L P_4 = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$$

Solution

Since,

$$\begin{aligned}
{}^G R_{z'y'x'}(50, 80, 25) &= \begin{bmatrix} c50c80 & c50s80s25 - s50c25 & c50s80c25 + s50s25 \\ s50c80 & s50s80s25 + c50c25 & s50s80c25 - c50s25 \\ -s80 & c80s25 & c80c25 \end{bmatrix} \\
&= \begin{bmatrix} 0.112 & -0.427 & 0.897 \\ 0.133 & 0.901 & 0.047 \\ -0.985 & 0.073 & 0.157 \end{bmatrix}
\end{aligned}$$

Therefore,

$${}^G P_1 = {}^G T \cdot {}^L P_1 = \begin{bmatrix} 0.112 & -0.427 & 0.897 & 10 \\ 0.133 & 0.901 & 0.047 & 20 \\ -0.985 & 0.073 & 0.157 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 10.000 \\ 20.000 \\ 30.000 \\ 1 \end{bmatrix}$$

$$\begin{aligned}
 {}^G P_2 &= {}^G_L T \cdot {}^L P_2 = \begin{bmatrix} 0.112 & -0.427 & 0.897 & 10 \\ 0.133 & 0.901 & 0.047 & 20 \\ -0.985 & 0.073 & 0.157 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 5 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 7.8663 \\ 24.505 \\ 30.365 \\ 1 \end{bmatrix} \\
 {}^G P_3 &= {}^G_L T \cdot {}^L P_3 = \begin{bmatrix} 0.112 & -0.427 & 0.897 & 10 \\ 0.133 & 0.901 & 0.047 & 20 \\ -0.985 & 0.073 & 0.157 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 5 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 12.350 \\ 26.567 \\ 31.154 \\ 1 \end{bmatrix} \\
 {}^G P_4 &= {}^G_L T \cdot {}^L P_4 = \begin{bmatrix} 0.112 & -0.427 & 0.897 & 10 \\ 0.133 & 0.901 & 0.047 & 20 \\ -0.985 & 0.073 & 0.157 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 14.487 \\ 22.060 \\ 30.787 \\ 1 \end{bmatrix}
 \end{aligned}$$

Exercise Problem 3.4

For the given parallelogram on y - z plane in space, determine transformed model if it has to be translated by $(10, 20, 30)$ along x , y , and z axes respectively and rotated by 50° around z' -axis, followed by 60° around y' -axis, and 20° around x' -axis respectively.

$${}^L P_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad {}^L P_2 = \begin{bmatrix} 0 \\ 4 \\ 0 \end{bmatrix}, \quad {}^L P_3 = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix}, \quad {}^L P_4 = \begin{bmatrix} 0 \\ 5 \\ 3 \end{bmatrix}$$

Solution

$$\begin{aligned}
 {}^G_L R_{z'y'x'}(50, 80, 25) &= \begin{bmatrix} c50c60 & c50s60s20 - s50c20 & c50s60c20 + s50s20 \\ s50c60 & s50s60s20 + c50c20 & s50s60c20 - c50s20 \\ -s60 & c60s20 & c60c20 \end{bmatrix} \\
 &= \begin{bmatrix} 0.321 & -0.529 & 0.785 \\ 0.383 & 0.831 & 0.404 \\ -0.866 & 0.171 & 0.470 \end{bmatrix}
 \end{aligned}$$

Therefore,

$$\begin{aligned}
 {}^G P_1 &= {}^G_L T \cdot {}^L P_1 = \begin{bmatrix} 0.321 & -0.529 & 0.785 & 10 \\ 0.383 & 0.831 & 0.404 & 20 \\ -0.866 & 0.171 & 0.470 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 20 \\ 30 \\ 1 \end{bmatrix} \\
 {}^G P_2 &= {}^G_L T \cdot {}^L P_2 = \begin{bmatrix} 0.321 & -0.529 & 0.785 & 10 \\ 0.383 & 0.831 & 0.404 & 20 \\ -0.866 & 0.171 & 0.470 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 4 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 7.882 \\ 23.324 \\ 30.684 \\ 1 \end{bmatrix} \\
 {}^G P_3 &= {}^G_L T \cdot {}^L P_3 = \begin{bmatrix} 0.321 & -0.529 & 0.785 & 10 \\ 0.383 & 0.831 & 0.404 & 20 \\ -0.866 & 0.171 & 0.470 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 11.826 \\ 22.042 \\ 31.581 \\ 4 \end{bmatrix} \\
 {}^G P_4 &= {}^G_L T \cdot {}^L P_3 = \begin{bmatrix} 0.321 & -0.529 & 0.785 & 10 \\ 0.383 & 0.831 & 0.404 & 20 \\ -0.866 & 0.171 & 0.470 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 5 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 9.708 \\ 25.365 \\ 35.265 \\ 1 \end{bmatrix}
 \end{aligned}$$

Exercise Problem 3.5

For the given triangle on y - z plane, determine the transformed model if it has to be translated by (5, 6, 9) along x , y , and z axes respectively and rotated by 45° around z' -axis, followed by 20° around y' -axis, and 30° around x' -axis respectively.

$${}^L P_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad {}^L P_2 = \begin{bmatrix} 0 \\ 4 \\ 1 \end{bmatrix}, \quad {}^L P_3 = \begin{bmatrix} 0 \\ 7 \\ 4 \end{bmatrix}$$

Answer

$${}^G P_1 = \begin{bmatrix} 5 \\ 6 \\ 9 \\ 1 \end{bmatrix}, \quad {}^G P_2 = \begin{bmatrix} 3.597 \\ 8.789 \\ 11.693 \\ 1 \end{bmatrix}, \quad {}^G P_3 = \begin{bmatrix} 3.812 \\ 10.557 \\ 15.544 \\ 1 \end{bmatrix}$$

Exercise Problem 3.6

For the given rectangle on $y-z$ plane, determine the transformed model if it has to be translated by (4, 13, 8) along x , y , and z axes respectively and rotated by 90° around z' -axis, followed by 45° around y' -axis, and 85° around x' -axis respectively.

$${}^L P_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad {}^L P_2 = \begin{bmatrix} 0 \\ 7 \\ 0 \end{bmatrix}, \quad {}^L P_3 = \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix}, \quad {}^L P_4 = \begin{bmatrix} 0 \\ 7 \\ 10 \end{bmatrix}$$

Answer

$${}^G P_1 = \begin{bmatrix} 4 \\ 13 \\ 8 \\ 1 \end{bmatrix}, \quad {}^G P_2 = \begin{bmatrix} 3.3899 \\ 17.9309 \\ 12.9309 \\ 1 \end{bmatrix}, \quad {}^G P_3 = \begin{bmatrix} 13.962 \\ 13.616 \\ 8.616 \\ 1 \end{bmatrix}, \quad {}^G P_4 = \begin{bmatrix} 13.352 \\ 18.547 \\ 13.547 \\ 1 \end{bmatrix}$$

Other Exercise Problems

1. A frame {B} is rotated relative to frame {A} around Z-axis by 60° . For a given ${}^B P$ vector such that;

$${}^B P = \begin{bmatrix} 3.0 \\ 4.0 \\ 0.0 \end{bmatrix},$$

Find the vector ${}^A P$.

2. A frame {B} is rotated relative to frame {A} around Z-axis by 45° . For a given ${}^B P$ vector such that;

$${}^B P = \begin{bmatrix} 7.0 \\ 10.0 \\ 0.0 \end{bmatrix},$$

Find the vector ${}^A P$.

3. For a given vector, ${}^B P$, below, if a frame {B} is rotated relative to {A} around Z-axis by 45° , and translated 15 units in X_A and 10 units in Y_A , describe the matrix T for the general transformation between {A} and {B}. In addition, obtain the resultant vector ${}^A P$ by the general transformation.

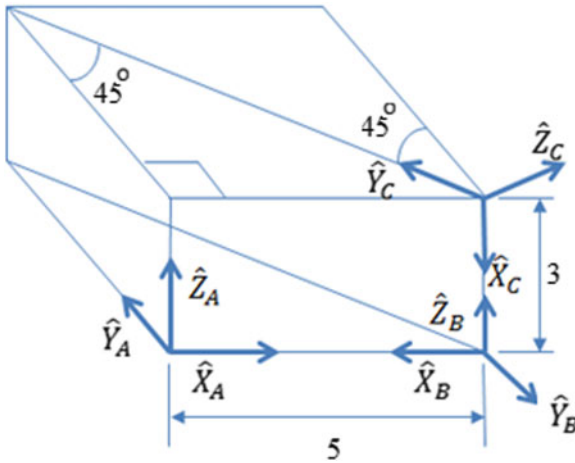
$${}^B P = \begin{bmatrix} 1.0 \\ 2.0 \\ 3.0 \\ 1 \end{bmatrix}$$

4. For a given vector, ${}^B P$, below, if a frame {B} is rotated relative to {A} around X-axis by 60° , and translated 10 units in X_A and 20 units in Z_A , describe the

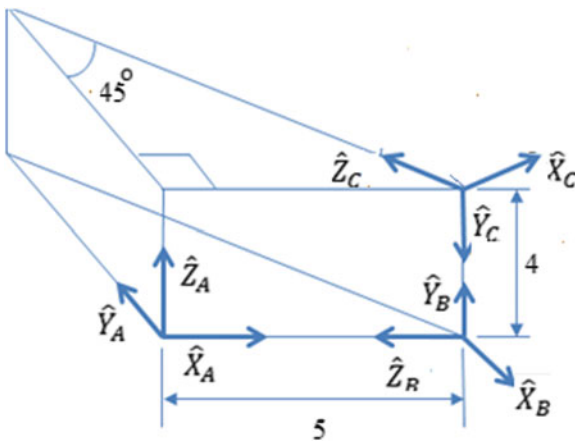
matrix T for the general transformation between $\{A\}$ and $\{B\}$. In addition, obtain the resultant vector ${}^B P$ by the general transformation.

$${}^B P = \begin{bmatrix} 10.0 \\ 15.0 \\ 20.0 \\ 1 \end{bmatrix}$$

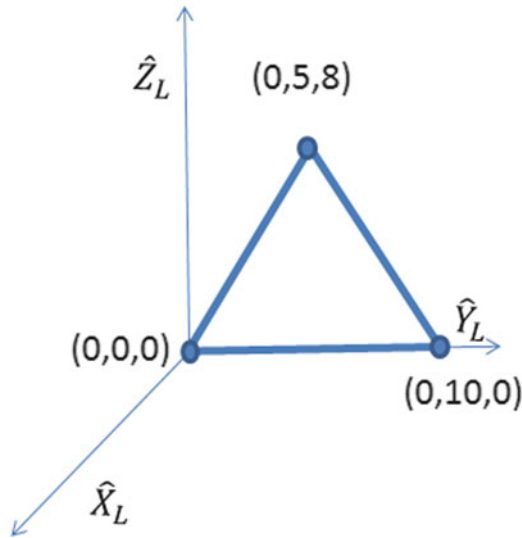
5. For given three frames in the figure below, define ${}^C T$ and ${}^A T$.



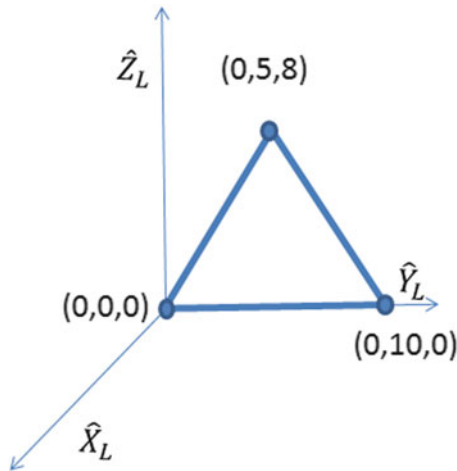
6. For given three frames in the figure below, define ${}^C T$ and ${}^A T$.



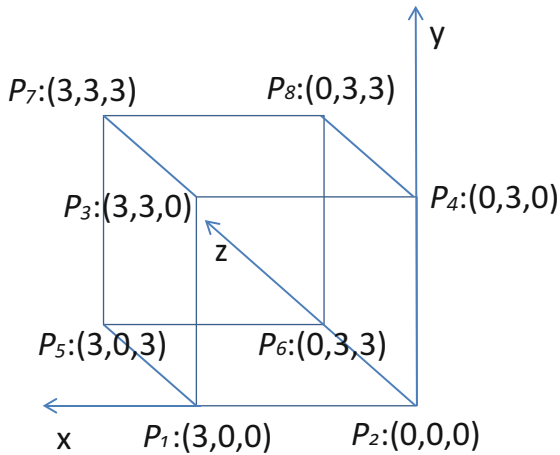
- 7. Rotate the triangle in the figure by 45° around x -axis and plot it on the $y-z$ plane of the global coordinate.



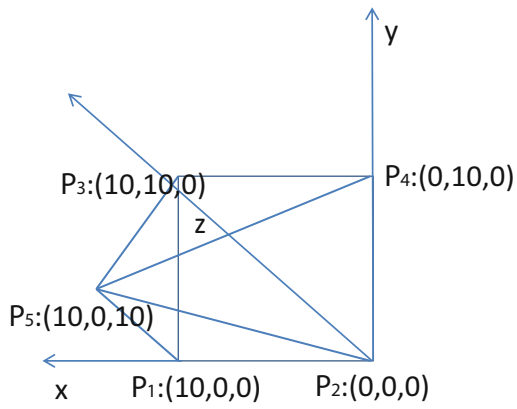
- 8. Rotate the triangle in the figure by 60° around y -axis and plot it on the $z-x$ plane of the global coordinate.



9. Draw the projection of a cube in the figure below on $u-v$ plane with the z_e value of 20.



10. Draw the projection of a cube in the figure below on $u-v$ plane with the z_e value of 20.



References

1. www.qtemfg.com
2. www.solidtrainer.com
3. www.3dscanco.com
4. <http://www.bigobjectbase.com/capture-process>
5. Solid-surface hybrid modeling: future trends of 3D CAD modeling. ZW3D CAD/CAM white paper, 2013. <http://zw3d.zwcad.org>
6. Craig JJ (2004) Introduction to robotics: mechanics and control. Prentice Hall, Upper Saddle River

Chapter 4

Parametric Line and Curve Theory

The Big Picture

Discussion Map

You need to understand the theory behind the graphical representation of elements such as line, curve, and surface for 3D modeling.

Discover

Understand the representation of lines by parametric line.

Understand the cubic spline curve theory.

Understand the B-spline curve theory.

Understand the basic surface theory.

When we draw a line on a paper, we need two points and a ruler or a guide to make a straight line. On a computer screen, the way the computer plots a line is to change the color of the corresponding points on the screen to represent a line. In order to do so, the computer must know of the coordinate of each point that belongs to the line. This also means that we need to tell the computer the equation of a line so that the computer changes the color of dots on the screen one by one from the start point to the end point. The general approach used in computer graphics is to set up a parametric equation whereby a variable changes to indicate a point on the line. Therefore, unlike the way we draw graphic elements on a paper, more specific guidelines are needed for a computer to represent 3D models.

4.1 Data Structure

One of the most efficient forms of data structure for 3D modeling in computer graphics is a matrix. A computer, once the procedure is logically well defined, can handle very complex matrix calculations with ease. There are, though, cases the computer may struggle with, such as singularity problems. However, these problems can be avoided with proper procedures. In this section, we will focus on understanding the fundamental difference between the representation by computer graphics and hand drafting, followed by parametric representation for computational modeling. As introduced earlier in Chap. 3, a computer understands a point as a column vector such as,

$$P(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ 1 \end{bmatrix}, \quad (4.1)$$

where t is a parameter to specify a point in space. Once an equation is given for each element of the point, then $p(t)$ can be defined specifically in space. The fourth element in the column is called a scaling factor. Each element of the point is parameterized, thus the term parametric lines. For a simple straight line in a plane, we use an equation as below.

$$y = a \cdot x + b \quad (4.2)$$

However, computer does not understand the general form of a line equation. Therefore, we change the equation above to a more general form such as,

$$a \cdot x + b \cdot y + c = 0 \quad (4.3)$$

Now this equation can be expressed in a matrix form as below.

$$[a \quad b \quad c] \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \quad (4.4)$$

Likewise, a plane equation in space can be expressed as below in a matrix form as well.

$$[a \quad b \quad c \quad d] \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0, \quad \text{or} \quad P \cdot R = 0 \quad (4.5)$$

where P stands for the row vector and the R stands for the column vector respectively. The first three vectors of P in (4.5) is a normal vector of a surface, while the fourth element, d , represents the distance of the plane from the origin of a coordinate system. One thing noticeable for the expression above is that any multiplication of a scaling factor to P or R does not change the geometric meaning of the surface. Another important property of the matrix notion of a line is that it can be transformed into a new line equation easily by coordinate transformation. For instance, let's assume the vector, R , in (4.5) is for a local coordinate system, "L." If a transformation matrix, ${}^G_L T$ is defined between two coordinate systems, "L" and "G," then, R' for a global coordinate system will be such that,

$$R' = {}^G_L T \cdot R \quad (4.6)$$

Since the relationship of $P' \cdot R' = 0$ has to be maintained in the global coordinate, the vector P' has to be,

$$P' = P \cdot {}^G_L T^{-1}. \quad (4.7)$$

So that,

$$P' \cdot R' = P \cdot {}^G_L T^{-1} \cdot {}^G_L T \cdot R = P \cdot R = 0$$

Since the transformation matrix is orthogonal, (4.3) become,

$$P' = P \cdot {}^G_L T^T. \quad (4.8)$$

4.2 Parametric Line

As mentioned earlier, computer understands a line as a collection of points that belong to the line. For two points in space, a parametric line can be defined as below.

$$R(\lambda) = \lambda \cdot R_1 + (1 - \lambda) \cdot R_2 = [R_1 \quad R_2] \cdot \begin{bmatrix} \lambda \\ 1 - \lambda \end{bmatrix} \quad (4.9)$$

The above equation is a parametric form of a line in computer graphics. For a line on a 2D plane, (4.9) becomes,

$$\begin{bmatrix} x(\lambda) \\ y(\lambda) \end{bmatrix} = \lambda \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + (1 - \lambda) \cdot \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \quad (4.10)$$

The parameter, λ , changes from 0 to 1 indicating a discrete point along the line. Notice that $R(\lambda)$ becomes R_1 when λ is equal to zero and R_2 when λ is equal to one.

Parametric line representation facilitates collision check between two lines. For instance, if two parametric line equations are given as below, we can check if they are crossing each other or not by solving for two parameters.

$$\begin{bmatrix} x(\lambda) \\ y(\lambda) \end{bmatrix} = \lambda \cdot \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + (1 - \lambda) \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad (4.11)$$

$$\begin{bmatrix} x(\mu) \\ y(\mu) \end{bmatrix} = \mu \cdot \begin{bmatrix} x_4 \\ y_4 \end{bmatrix} + (1 - \mu) \cdot \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} \quad (4.12)$$

The idea is that if we set them equal, assuming that they are crossing each other, the values of two parameters have to be in the valid region. Since the valid region of a parameter is from zero to one, if either one of them is not in between zero and one, conclusively they are not crossing each other. For two equations given above, if we set them equal, and rearrange them for two parameters, we obtain,

$$\lambda \cdot R_2 + (1 - \lambda) \cdot R_1 = \mu \cdot R_4 + (1 - \mu) \cdot R_3 \quad (4.13)$$

$$(R_2 - R_1) \cdot \lambda - (R_4 - R_3) \cdot \mu = R_3 - R_1 \quad (4.14)$$

In matrix form,

$$\begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} \cdot \lambda - \begin{bmatrix} x_4 - x_3 \\ y_4 - y_3 \end{bmatrix} \cdot \mu = \begin{bmatrix} x_3 - x_1 \\ y_3 - y_1 \end{bmatrix} \quad (4.15)$$

Therefore, the solution of the above equation for two parameters, λ and μ should yield values from zero to one if they collide each other.

Sample Problem 4.1

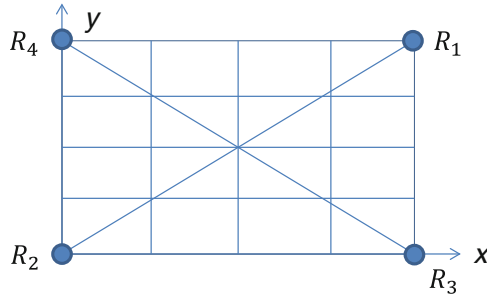
For two lines defined by two points for each line below, plot them on a plane and visually observe if they collide each other. Verify your observation by solving for two parameters associated with two parametric line equations.

$$\text{Line 1 : } R_1 = \begin{bmatrix} 4 \\ 3 \end{bmatrix}, \quad R_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\text{Line 2 : } R_3 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}, \quad R_4 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

Solution

By visual observation, two lines are colliding each other. Now if we establish the matrix equation of (4.15) with the given point data, we obtain the equation below.



$$\begin{bmatrix} 0 & -4 \\ 0 & -3 \end{bmatrix} \cdot \lambda - \begin{bmatrix} 0 & -4 \\ 3 & 0 \end{bmatrix} \cdot \mu = \begin{bmatrix} 4 & -4 \\ 0 & -3 \end{bmatrix}$$

If we solve the above equation for λ and μ we obtain 0.5 for both λ and μ . Since both parameters are in the valid region, two lines have to collide each other. This result is equivalent to the visual observation as shown in the figure above.

4.3 Cubic Spline Curve

Generating a smooth curve on a computer screen is demanding in modern industry and is challenging for design engineers. The most convenient way of generating a smooth curve on the computer screen is to indicate multiple seed points and let the computer connect them with a nice and smooth curve. A spline curve is defined as a collection of piecewise curves that connect multiple control points, also known as knots, in sufficiently smooth fashion. Although the definition of “sufficiently smooth fashion” is a bit ambiguous, it is the form of an elastic curve that is close to a metallic band if it is connected with all the seed points. We define initially seeded points as control points. If piecewise curves of the total curve are expressed by polynomial equations, we call it a polynomial spline curve. If the polynomial equation is a cubic polynomial, then we call it a cubic spline curve. We cast in this section a question as to how to obtain piecewise cubic polynomial curve equations by which a smooth elastic curve is described. In addition, we will review some advantages and disadvantages of a cubic spline curve later in the section.

Cubic spline curve is also known as Hermite spline curve. Each segment of the overall curve will be expressed by a cubic polynomial equation in a Cartesian coordinate system. Equations are parametric equations so that the value of the parameter will change from zero to one. Therefore, the total number of equations required is the same as the total number of piecewise curves to complete a spline curve (see Fig. 4.1). We start developing a curve equation for each segment based on geometric constraints so that piecewise curves will constitute a smooth spline curve.

Fig. 4.1 Cubic spline curve

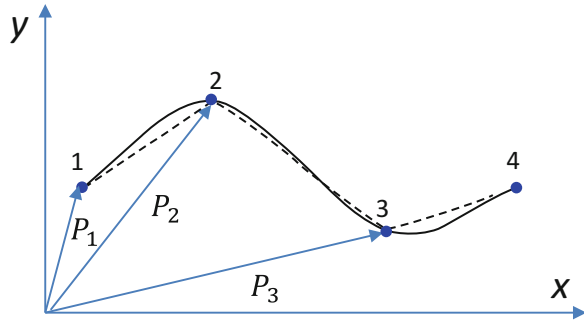
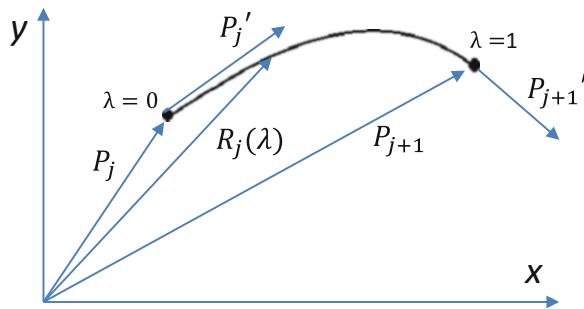


Fig. 4.2 Piecewise curve



In order to create a cubic polynomial equation for each segment, we start with a piecewise curve to develop a general form of the equation (see Fig. 4.2). P_j and P_{j+1} in the figure below are generalized vectors that indicate start and end dots on the piecewise curve, while P'_j and P'_{j+1} are the tangential vectors to the curve at each control point respectively. $R_j(\lambda)$ in the figure is a cubic polynomial equation by which a location vector of each dot is specified by the parameter, λ .

By definition, each tangent vector is defined as below.

$$P'_j = \left. \frac{dP_j}{d\lambda} \right|_{\lambda=0}, \quad P'_{j+1} = \left. \frac{dP_j}{d\lambda} \right|_{\lambda=1} \tag{4.16}$$

Notice that the value of λ is zero at the start point and one at the end point. Now, since each element of the vector R is a cubic polynomial equation of the parameter, λ , three equations are required to express R such that,

$$\begin{aligned} x_j(\lambda) &= a \cdot \lambda^3 + b \cdot \lambda^2 + c \cdot \lambda + d \\ y_j(\lambda) &= e \cdot \lambda^3 + f \cdot \lambda^2 + g \cdot \lambda + h \\ z_j(\lambda) &= i \cdot \lambda^3 + j \cdot \lambda^2 + k \cdot \lambda + l \end{aligned} \tag{4.17}$$

In matrix form, (4.17) will become,

$$\begin{bmatrix} x(\lambda) \\ y(\lambda) \\ z(\lambda) \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \lambda^3 \\ \lambda^2 \\ \lambda \\ \mathbf{1} \end{bmatrix} \quad (4.18)$$

The coefficient matrix in the above equation is called cubic polynomial coefficient matrix. The equation above can be simplified as below.

$$\mathbf{R}_j(\lambda) = [\mathbf{C}_j] \cdot \begin{bmatrix} \lambda^3 \\ \lambda^2 \\ \lambda \\ \mathbf{1} \end{bmatrix} \quad (4.19)$$

where \mathbf{C}_j is called cubic polynomial coefficient. Now the primary task narrows down to determine all of the components in the cubic polynomial coefficient matrix. To that end, we need to collect boundary conditions by geometric constraints of the piecewise curve to solve for the polynomial coefficients. Two boundary conditions can be easily identified for the curve. One is the location relevant boundary condition, and the other is relevant to the tangent of the curve at the boundary. For the location specific boundary condition, the following two boundary conditions have to be met.

$$\mathbf{R}_j(\mathbf{0}) = \mathbf{P}_j, \quad \mathbf{R}_j(\mathbf{1}) = \mathbf{P}_{j+1}$$

or

$$\mathbf{R}_j(\mathbf{0}) = [\mathbf{C}_j] \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} \quad (4.20)$$

$$\mathbf{R}_j(\mathbf{1}) = [\mathbf{C}_j] \cdot \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix} \quad (4.21)$$

In addition, since the derivative of the curve equation, $R(\lambda)$, becomes a tangent line equation such as,

$$\mathbf{R}'_j(\lambda) = \frac{d\mathbf{R}_j}{d\lambda} = [\mathbf{C}_j] \cdot \begin{bmatrix} 3 \cdot \lambda^2 \\ 2 \cdot \lambda \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix},$$

The following two boundary conditions have to be met as well.

$$R'_j(0) = P'_j, \quad R'_j(1) = P'_{j+1}$$

or

$$R'_j(0) = [C_j] \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad (4.22)$$

$$R'_j(1) = [C_j] \cdot \begin{bmatrix} 3 \\ 2 \\ 1 \\ 0 \end{bmatrix} \quad (4.23)$$

Now, if we collect all four equations from (4.20) to (4.23) and pack them into a single matrix equation, we obtain;

$$\left[\begin{array}{c|c|c|c} P_j & P'_j & P_{j+1} & P'_{j+1} \end{array} \right] = [C_j] \begin{bmatrix} 0 & 0 & 1 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad (4.24)$$

The above equation packed by all four boundary conditions is made possible by the fact that all four equations have the common cubic polynomial coefficient matrix. Since the goal is to find the cubic polynomial coefficient matrix, we can rearrange the above equation such that;

$$[C_j] = \left[\begin{array}{c|c|c|c} P_j & P'_j & P_{j+1} & P'_{j+1} \end{array} \right] \cdot \begin{bmatrix} 0 & 0 & 1 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}^{-1} \quad (4.25)$$

or

$$[C_j] = \left[\begin{array}{c|c|c|c} P_j & P'_j & P_{j+1} & P'_{j+1} \end{array} \right] \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}. \quad (4.26)$$

Assuming a design engineer already specified control points, P_j and P_{j+1} , in order to find the cubic polynomial coefficients, two unknown vectors of tangent, P'_j and P'_{j+1} have to be determined. In order to find two unknowns, we use geometric constraints that have to be met between piecewise curves. Considering the definition of the spline curve aforementioned, first, two consecutive piecewise curves need not only to meet each other but also to have a coincident tangent slop at the

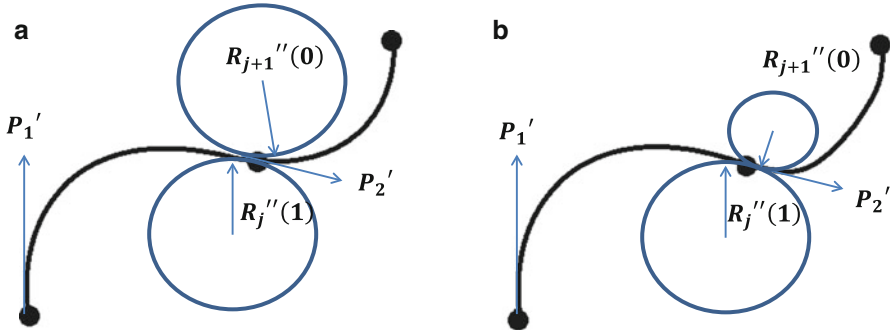


Fig. 4.3 Curvature constraint between two consecutive segments. (a) Consistent curvature (preferable). (b) Inconsistent curvature (less preferable)

point of contact. However, the first two boundary conditions does not provide further clue for the value of each tangent vector at the point of contact. Therefore, another boundary condition has to be considered. Extra boundary conditions are obtained from curvature. That is, two consecutive piecewise curves need to have a coincident curvature at the point of contact to be a smooth curve. As shown in Fig. 4.3, the spline curve on the left looks more natural compared to the one on the right since two curves have the same curvature at the point of contact.

Since the second derivative of a curve equation represents curvature, we obtain the second derivative of the curve (4.19).

$$R_j''(\lambda) = \frac{d^2 R_j}{d\lambda^2} = [C_j] \cdot \begin{bmatrix} 6 \cdot \lambda \\ 2 \\ 0 \\ 0 \end{bmatrix},$$

Then, the following equation has to be valid.

$$R_{j-1}''(1) = R_j''(0)$$

or

$$[C_j] \cdot \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \end{bmatrix} = [C_{j-1}] \cdot \begin{bmatrix} 6 \\ 2 \\ 0 \\ 0 \end{bmatrix} \tag{4.27}$$

By using (4.26), the above equation can be rewritten as;

$$\begin{aligned} & \left[\mathbf{P}_j \mid \mathbf{P}'_j \mid \mathbf{P}_{j+1} \mid \mathbf{P}'_{j+1} \right] \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \end{bmatrix} \\ &= \left[\mathbf{P}_{j-1} \mid \mathbf{P}'_{j-1} \mid \mathbf{P}_j \mid \mathbf{P}'_j \right] \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 6 \\ 2 \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

In short,

$$\left[\mathbf{P}_j \mid \mathbf{P}'_j \mid \mathbf{P}_{j+1} \mid \mathbf{P}'_{j+1} \right] \cdot \begin{bmatrix} -6 \\ -4 \\ 6 \\ -2 \end{bmatrix} = \left[\mathbf{P}_{j-1} \mid \mathbf{P}'_{j-1} \mid \mathbf{P}_j \mid \mathbf{P}'_j \right] \cdot \begin{bmatrix} 6 \\ 2 \\ -6 \\ 4 \end{bmatrix} \quad (4.28)$$

The above equation can be further simplified as an equation below.

$$\mathbf{P}'_{j-1} + 4\mathbf{P}'_j + \mathbf{P}'_{j+1} = 3(\mathbf{P}_{j+1} - \mathbf{P}_{j-1}) \quad (4.29)$$

The final equation obtained from the assumption of the same curvature establishes a relationship between tangent vectors and the control points. Notice that all the vectors on the left are unknown, while the vectors on the right hand side are known or given points to draw a spline curve. If the number of given control points are N , starting from 1, then obviously the parameter j starts from 2 to $N - 1$.

Sample Problem 4.2

Show how (4.28) becomes (4.29).

Solution

If we express the matrix equation of (4.28) to a simple algebraic equation, we obtain;

$$-6 \cdot \mathbf{P}_j - 4 \cdot \mathbf{P}'_j + 6 \cdot \mathbf{P}_{j+1} - 2 \cdot \mathbf{P}'_{j+1} = 6 \cdot \mathbf{P}_{j-1} + 2 \cdot \mathbf{P}'_{j-1} - 6 \cdot \mathbf{P}_j + 4 \cdot \mathbf{P}'_j$$

The above equation becomes;

$$2 \cdot P'_{j-1} + 8 \cdot P'_j + 2 \cdot P'_{j+1} = 6 \cdot P_{j+1} - 6 \cdot P_{j-1}$$

or

$$P'_{j-1} + 4P'_j + P'_{j+1} = 3(P_{j+1} - P_{j-1})$$

Now with the generalized equation of (4.29), if we collect the equations for all of the piecewise curves for given control points, we can establish a following matrix equation.

$$\begin{bmatrix} 4 & 1 & 0 & \dots & 0 & 0 \\ 1 & 4 & 1 & \dots & 0 & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 \\ & & \vdots & & & \\ 0 & 0 & \dots & 1 & 4 & 1 \\ 0 & 0 & \dots & 0 & 1 & 4 \end{bmatrix} \cdot \begin{bmatrix} P'_2 \\ P'_3 \\ P'_4 \\ \vdots \\ P'_{N-2} \\ P'_{N-1} \end{bmatrix} = \begin{bmatrix} 3(P_3 - P_1) - P'_1 \\ 3(P_4 - P_2) \\ 3(P_5 - P_3) \\ \vdots \\ 3(P_{N-1} - P_{N-3}) \\ 3(P_N - P_{N-2}) - P'_N \end{bmatrix} \quad (4.30)$$

Notice that all of the components on the right hand side of the above equation are known except P'_1 and P'_N . In order to simplify the above equation by removing P'_1 and P'_N on the right hand side, we add two virtual piecewise curves at both ends of the spline curve so that we obtain two more equations: one with the parameter j equal to 1 and the other one with the parameter j equal to N . In other words, two virtual piecewise curves at both end yield following two equations.

$$\begin{aligned} R''_0(1) &= R''_1(0), \quad \text{where } R''_0(1) = P'_1 \\ R''_{N-1}(1) &= R''_N(0), \quad \text{where } R''_{N-1}(1) = P'_N \end{aligned}$$

Now, if we assume $P'_1 = \mathbf{0}$ and $P'_N = \mathbf{0}$, then by (4.28), the first equation above yields the following equation.

$$\begin{aligned} -6 \cdot P_1 - 4 \cdot P'_1 + 6 \cdot P_2 - 2 \cdot P'_2 &= 0 \\ \text{or} \\ 2 \cdot P'_1 + P'_2 &= 3(P_2 - P_1) \end{aligned} \quad (4.31)$$

Likewise, by the second curvature equation, we obtain the following equation.

$$\begin{aligned} -6 \cdot P_{N-1} - 4 \cdot P'_{N-1} + 6 \cdot P_N - 2 \cdot P'_N &= 0 \\ \text{or} \\ 2 \cdot P'_N + P'_{N+1} &= 3(P_{N+1} - P_N) \end{aligned} \quad (4.32)$$

Now, if we add (4.31) and (4.32) in (4.30), and bring P'_1 and P'_N to the left hand side of the equation, then (4.30) becomes,

$$\begin{bmatrix} 2 & 1 & 0 & \dots & 0 & 0 \\ 1 & 4 & 1 & \dots & 0 & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 \\ & & \vdots & & & \\ 0 & 0 & \dots & 1 & 4 & 1 \\ 0 & 0 & \dots & 0 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \\ \vdots \\ P'_{N-1} \\ P'_N \end{bmatrix} = 3 \begin{bmatrix} P_2 - P_1 \\ P_3 - P_1 \\ P_4 - P_2 \\ \vdots \\ P_N - P_{N-2} \\ P_N - P_{N-1} \end{bmatrix} \tag{4.33}$$

The above equation, then, is a matrix equation that contains all of the unknown tangent vectors on the left hand side and the all of the known control points on the right hand side of the equation. The unknown tangent vectors in (4.33) can be determined by solving the following matrix equation.

$$\begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \\ \vdots \\ P'_{N-1} \\ P'_N \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 & \dots & 0 & 0 \\ 1 & 4 & 1 & \dots & 0 & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 \\ & & \vdots & & & \\ 0 & 0 & \dots & 1 & 4 & 1 \\ 0 & 0 & \dots & 0 & 1 & 2 \end{bmatrix}^{-1} \cdot 3 \begin{bmatrix} P_2 - P_1 \\ P_3 - P_1 \\ P_4 - P_2 \\ \vdots \\ P_N - P_{N-2} \\ P_N - P_{N-1} \end{bmatrix} \tag{4.34}$$

Therefore, once the tangent vectors are obtained by (4.34), then we can determine the cubic polynomial coefficients, C_j , and thus we can define the cubic polynomial equations of all of the piecewise curves that belong to the complete spline curve using (4.19).

While the cubic spline curve theory introduced in this section allows us to define a smooth spline curve expressed by a third-order polynomial equation for each segment, there is an innate disadvantage in regard to the local control of the cubic spline curve. The term, local control, means a capability of being able to change the control points with minimum overall shape change of the original spline curve. This may happen when there is a disagreement between a designer and a manufacturer. While a designer wants to maintain the original design for an aerodynamic body shape of a vehicle, for instance, a manufacturer may need to change the hood shape to house a new engine which requires a larger space. For instance, two spline curves shown in Fig. 4.4 are from a set of control points that are very close to each other. However, as it is depicted in the figure, slight change of the locations of the intermittent control points results in a dramatic shape change of the spline curve. As a result, it is said that the cubic spline curve falls short of practicality in real-world applications such as engineering designs in various fields.

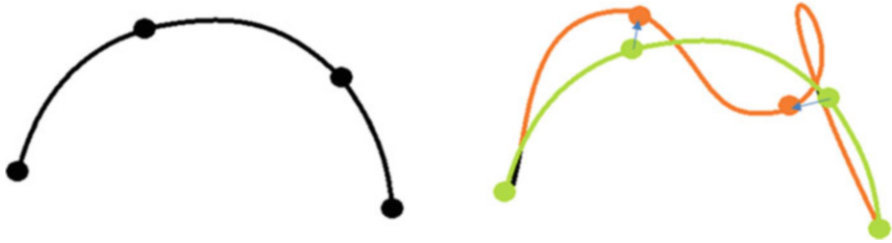


Fig. 4.4 Lack of local control

Sample Problem 4.3

For the given three control points below, determine the equation of the Hermite spline curve. Assume $P_1'' = P_3'' = \mathbf{0}$

$$P_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 4 \\ 3 \\ 1 \\ 1 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 10 \\ 5 \\ 6 \\ 1 \end{bmatrix}$$

Solution

Since $N = 3$, first, (4.34) becomes the following.

$$\begin{bmatrix} P_1' \\ P_2' \\ P_3' \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix}^{-1} \cdot 3 \begin{bmatrix} P_2 - P_1 \\ P_3 - P_1 \\ P_3 - P_2 \end{bmatrix}$$

$$\begin{bmatrix} P_1' \\ P_2' \\ P_3' \end{bmatrix} = \begin{bmatrix} 0.583 & -0.167 & 0.083 \\ -0.167 & 0.333 & -0.167 \\ 0.083 & -0.167 & 0.583 \end{bmatrix} \cdot 3 \begin{bmatrix} P_2 - P_1 \\ P_3 - P_1 \\ P_3 - P_2 \end{bmatrix}$$

Therefore, we obtain the following equation for P_1' from the first row.

$$P_1' = 0.583 \cdot 3 \cdot (P_2 - P_1) - 0.167 \cdot 3 \cdot (P_3 - P_1) + 0.083 \cdot 3 \cdot (P_3 - P_2),$$

or in matrix form,

$$\begin{aligned}
 P'_1 &= 1.75 \cdot \left(\begin{bmatrix} 4 \\ 3 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right) - 0.5 \cdot \left(\begin{bmatrix} 10 \\ 5 \\ 6 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right) + 0.25 \cdot \left(\begin{bmatrix} 10 \\ 5 \\ 6 \\ 1 \end{bmatrix} - \begin{bmatrix} 4 \\ 3 \\ 1 \\ 1 \end{bmatrix} \right) \\
 &= 1.75 \cdot \begin{bmatrix} 3 \\ 2 \\ 0 \\ 0 \end{bmatrix} - 0.5 \cdot \begin{bmatrix} 9 \\ 4 \\ 5 \\ 0 \end{bmatrix} + 0.25 \cdot \begin{bmatrix} 6 \\ 2 \\ 5 \\ 0 \end{bmatrix} = \begin{bmatrix} 2.25 \\ 2 \\ -1.25 \\ 0 \end{bmatrix}
 \end{aligned}$$

Likewise, P'_2 and P'_3 can be found as below.

$$\begin{aligned}
 P'_2 &= -0.5 \cdot \begin{bmatrix} 3 \\ 2 \\ 0 \\ 0 \end{bmatrix} + 1 \cdot \begin{bmatrix} 9 \\ 4 \\ 5 \\ 0 \end{bmatrix} - 0.5 \cdot \begin{bmatrix} 6 \\ 2 \\ 5 \\ 0 \end{bmatrix} = \begin{bmatrix} 4.5 \\ 2 \\ 2.5 \\ 0 \end{bmatrix} \\
 P'_3 &= 0.249 \cdot \begin{bmatrix} 3 \\ 2 \\ 0 \\ 0 \end{bmatrix} - 0.5 \cdot \begin{bmatrix} 9 \\ 4 \\ 5 \\ 0 \end{bmatrix} + 1.749 \cdot \begin{bmatrix} 6 \\ 2 \\ 5 \\ 0 \end{bmatrix} = \begin{bmatrix} 6.741 \\ 1.996 \\ 6.245 \\ 0 \end{bmatrix}
 \end{aligned}$$

Therefore, with all of the tangent vectors obtained so far, the cubic polynomial coefficient, C_j , can be determined by (4.25) for each segment such as;

$$\begin{aligned}
 [C_1] &= \left[P_1 \mid P'_1 \mid P_2 \mid P'_2 \right] \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 2.25 & 4 & 4.5 \\ 1 & 2 & 3 & 2 \\ 1 & -1.25 & 1 & 2.5 \\ 1 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.75 & 0 & 2.25 & 1 \\ 0 & 0 & 2.0 & 1 \\ 1.25 & 0 & -1.25 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

$$\begin{aligned}
 [C_2] &= [P_2 | P'_2 | P_3 | P'_3] \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \\
 &= \begin{bmatrix} 4 & 4.5 & 10 & 6.741 \\ 3 & 2 & 5 & 1.996 \\ 1 & 2.5 & 6 & 6.245 \\ 1 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -0.759 & 2.259 & 4.5 & 4 \\ -0.004 & 0.004 & 2.0 & 3 \\ -1.255 & 3.755 & 2.5 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

With two cubic polynomial coefficients, C_1 and C_2 , now we determine the cubic polynomial equation of each segment by (4.19) such as;

$$\begin{aligned}
 R_1(\lambda) &= \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.75 & 0 & 2.25 & 1 \\ 0 & 0 & 2.0 & 1 \\ 1.25 & 0 & -1.25 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \lambda^3 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}, \\
 R_2(\lambda) &= \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.759 & 2.259 & 4.5 & 4 \\ -0.004 & 0.004 & 2.0 & 3 \\ -1.255 & 3.755 & 2.5 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \lambda^3 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}.
 \end{aligned}$$

From the matrix equation above, the cubic spline curve equations for the first segment are as follows.

$$\begin{aligned}
 x_1(\lambda) &= 0.75 \cdot \lambda^3 + 2.25 \cdot \lambda + 1 \\
 y_1(\lambda) &= 2.0 \cdot \lambda + 1 \\
 z_1(\lambda) &= 1.25 \cdot \lambda^3 - 1.25 \cdot \lambda + 1
 \end{aligned}$$

Cubic spline curve equations for the second segment are as follows.

$$\begin{aligned}
 x_2(\lambda) &= -0.759 \cdot \lambda^3 + 2.259 \cdot \lambda^2 + 4.5 \cdot \lambda + 4 \\
 y_2(\lambda) &= -0.004 \cdot \lambda^3 + 0.004 \cdot \lambda^2 + 2.0 \cdot \lambda + 3 \\
 z_2(\lambda) &= -1.255 \cdot \lambda^3 + 3.755 \cdot \lambda^2 + 2.5 \cdot \lambda + 1
 \end{aligned}$$

Sample Problem 4.4

By using the cubic spline curve equation from the Problem 4.1, produce the spline curve using Excel spreadsheet.

Solution

As shown in the spreadsheet below, the first column contains the values of λ varying from zero to one increasing by 0.1. The incremental interval may change depending on the accuracy requirement. Second, third, and fourth columns are for x , y , and z values of each segment.

	A	B	C	D
1	First segment			
2	λ	x	y	z
3	0	1	1	1
4	0.1	1.22575	1.2	0.87625
5	0.2	1.456	1.4	0.76
6	0.3	1.69525	1.6	0.65875
7	0.4	1.948	1.8	0.58
8	0.5	2.21875	2	0.53125
9	0.6	2.512	2.2	0.52
10	0.7	2.83225	2.4	0.55375
11	0.8	3.184	2.6	0.64
12	0.9	3.57175	2.8	0.78625
13	1	4	3	1

	A	B	C	D
14	Second segment			
15	λ	x	y	z
16	0	4	3	1
17	0.1	4.471831	3.200036	1.286295
18	0.2	4.984288	3.400128	1.64016
19	0.3	5.532817	3.600252	2.054065
20	0.4	6.112864	3.800384	2.52048
21	0.5	6.719875	4.0005	3.031875
22	0.6	7.349296	4.200576	3.58072
23	0.7	7.996573	4.400588	4.159485
24	0.8	8.657152	4.600512	4.76064
25	0.9	9.326479	4.800324	5.376655
26	1	10	5	6

Below is the set of equations for cells assigned to x , y , and z .

$$\text{Cell 'B4'} = 0.75*A4^3 + 0*A4^2 + 2.25*A4 + 1$$

$$\text{Cell 'C4'} = 0*A4^3 + 0*A4^2 + 2*A4 + 1$$

$$\text{Cell 'D4'} = 1.25*A4^3 + 0*A4^2 - 1.25*A4 + 1$$

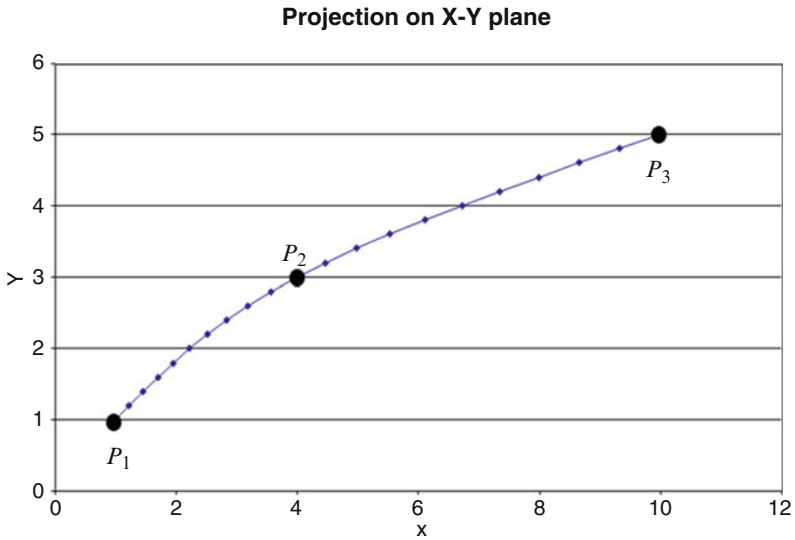
The rest of the cells are filled by simply dragging each of these cells down to the last row to copy the formula. Below is the formulation of the first row for the second segment.

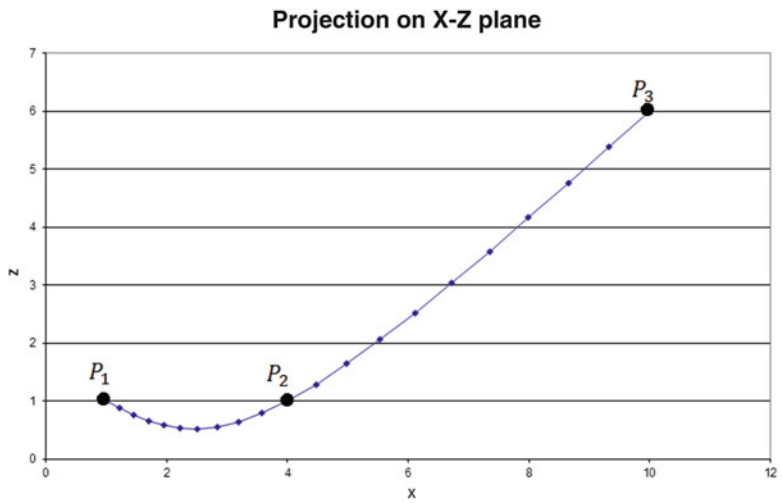
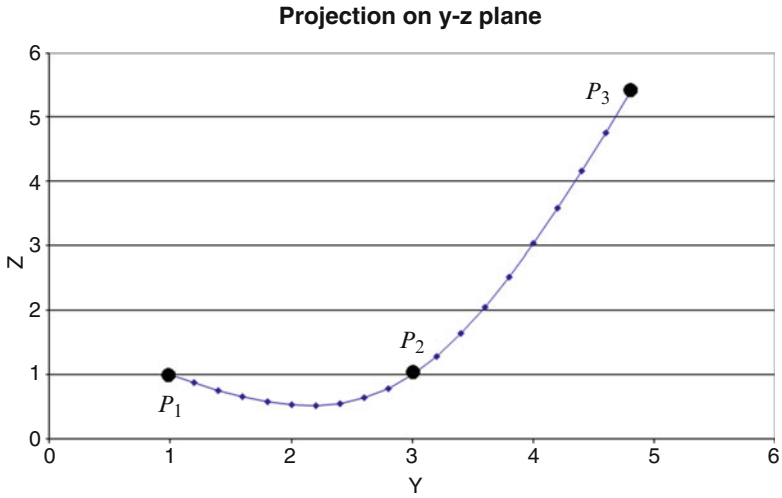
$$\text{Cell 'B17'} = -0.759*A17^3 + 2.259*A17^2 + 4.5*A17 + 4$$

$$\text{Cell 'C17'} = -0.004*A17^3 + 0.004*A17^2 + 2*A17 + 3$$

$$\text{Cell 'D17'} = -1.255*A17^3 + 3.755*A17^2 + 2.5*A17 + 1$$

Projection plots of the obtained cubic spline curve are shown in the figures below.

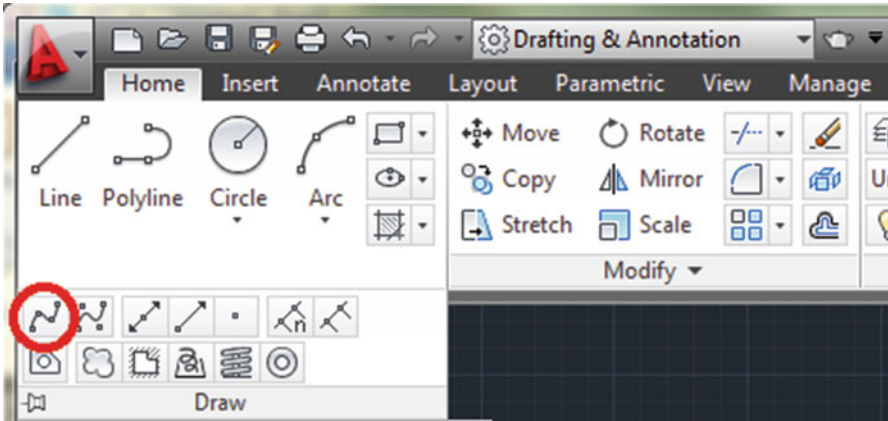




Sample Problem 4.5

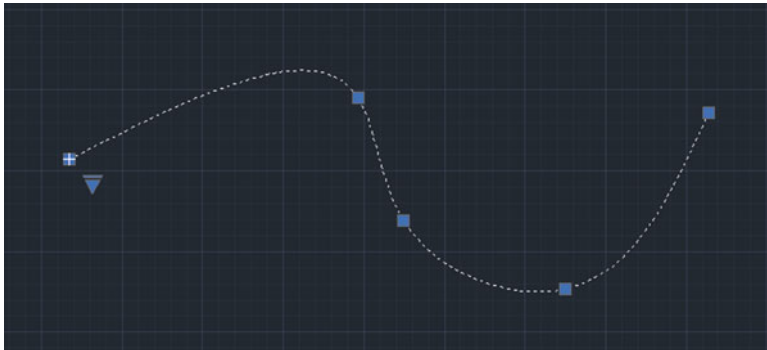
Draw a Hermite spline curve with AUTOCAD (AutoCAD version 2011 is used).

Solution



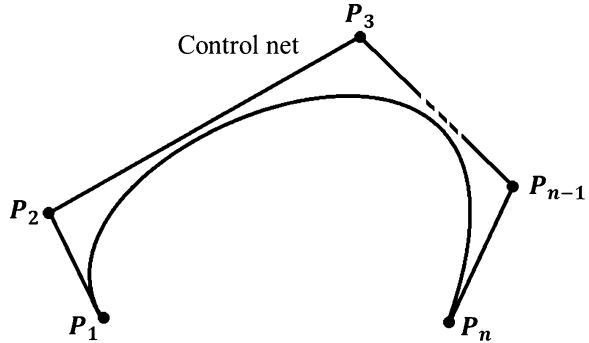
1. Make sure that the workspace is selected for “Drafting & Annotation.”
2. Click on the “Draw” panel to expand it.
3. Select the “cubic spline curve” in the draw panel (marked with a circle in the above figure).
4. Click control points on the model space.
5. Push “Enter” key to finish operation

Notice that the curve passes each and all control points you selected.



An example of a cubic spline curve

Fig. 4.5 Control net concept in Bezier spline curve



4.4 Bezier Spline Curve

Bezier spline curve is born of necessity to obtain a complete local control in spline curve design. As discussed earlier, Hermite spline curve falls short of local control (see Sect. 4.3) and, therefore, is limited in industrial applications. Pierre Bezier, then an engineer at Renault Company in France, invented a mathematical means to express a spline curve to achieve better local control in spline curve design. He became a leader in the transformation of design and manufacturing through mathematics and computing tools into CAD and 3D modeling. Eventually his team developed the UNISURF CAD system in 1968. UNISURF was a pioneering surface CAD/CAM system, designed to assist with car body design and tooling, which entered full use at the company in 1975. In order to achieve the complete local control, Bezier spline uses the concept of control net by which the spline curve is guided along the straight line segments of the control net (see Fig. 4.5).

Bezier spline curve is based on the concept of the blending function, by which all of the piecewise curves are connected in an elastic curve formation. Multiple blending functions will be formulated depending on the number of control points by (4.35) below.

$$B_{N,j}(s) = \frac{N!s^j(1-s)^{N-j}}{j!(N-j)!} \quad (4.35)$$

N is the number of blending functions, which is equal to the number of control points minus one. The index, j , is the parameter for each blending function, varying from 0 to N . Each blending function of Bezier spline curve is mathematically formulated to make different influences to each control point. The index, s is a parameter that is varying from 0 to 1 equally for all blending functions. For instance, if there are five control points, N will be 4 and j will vary from 0 to 4. Therefore, there will be five blending functions as below.

$$\begin{aligned}
 B_{4,0}(s) &= \frac{4!S^0(1-s)^4}{0!4!} = (1-s)^4 \\
 B_{4,1}(s) &= \frac{4!S^1(1-s)^3}{1!3!} = 4 \cdot s \cdot (1-s)^3 \\
 B_{4,2}(s) &= \frac{4!S^2(1-s)^2}{2!2!} = 6 \cdot S^2(1-s)^2 \\
 B_{4,3}(s) &= \frac{4!S^3(1-s)^1}{3!1!} = 4 \cdot S^3 \cdot (1-s) \\
 B_{4,4}(s) &= \frac{4!S^4(1-s)^0}{4!0!} = S^4
 \end{aligned}$$

Now, if we plot each blending function, varying the parameter, s , from 0 to 1, we obtain five graphs as below.

As shown in Fig. 4.6, each blending function has different level of influence to each control point. For instance, the first blending function has the highest influence to the first control point, while the second blending function has the highest influence to the second control point, and so on. Once all of the blending functions are defined, then we combine them all in one formula as shown below.

$$R(s) = \sum_{j=0}^N B_{N,j}(s) \cdot p_j, \tag{4.36}$$

or in matrix form,

$$\begin{bmatrix} x \\ y \\ z \\ \mathbf{1} \end{bmatrix} = \sum_{j=0}^N B_{N,j}(s) \cdot \begin{bmatrix} x_j \\ y_j \\ z_j \\ \mathbf{1} \end{bmatrix}. \tag{4.37}$$

If we expand (4.37) for the case of five control points, then we obtain following three equations.

Fig. 4.6 Blending functions of Bezier spline curve

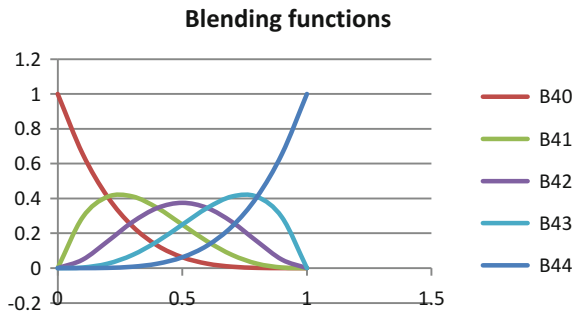
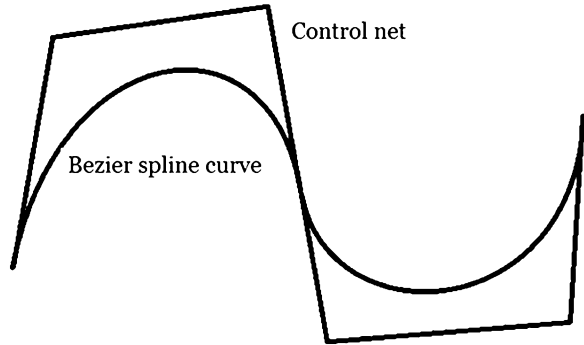


Fig. 4.7 Bezier spline curve and control net



$$\begin{aligned}
 x &= B_{4,0}(s) \cdot x_0 + B_{4,1}(s) \cdot x_1 + B_{4,2}(s) \cdot x_2 + B_{4,3}(s) \cdot x_3 + B_{4,4}(s) \cdot x_4 \\
 y &= B_{4,0}(s) \cdot y_0 + B_{4,1}(s) \cdot y_1 + B_{4,2}(s) \cdot y_2 + B_{4,3}(s) \cdot y_3 + B_{4,4}(s) \cdot y_4 \\
 z &= B_{4,0}(s) \cdot z_0 + B_{4,1}(s) \cdot z_1 + B_{4,2}(s) \cdot z_2 + B_{4,3}(s) \cdot z_3 + B_{4,4}(s) \cdot z_4
 \end{aligned}$$

Bezier spline curve has some unique properties. First, we define the degree of Bezier spline curve as the number of control points. If we created five control points, then the degree of the Bezier spline curve is five. Second, unlike the Hermite spline curve, the curve is not formulated by geometric constraints, but it is formed naturally by the influence of each blending function to each control point. Third, the Bezier spline curve is “gently” guided by a control net, but does not interpolate between each control point. This is because of the “variation diminishing” property of the curve by the blending functions. Forth, as shown in Fig. 4.7, the Bezier spline curve only meets the first and the last control points. In addition, the tangent of the curve matches the control net only at the first and the last control points. Another interesting property is that the curve always stays at concave side of the control net. Sometimes, the curve crosses the control net to stay on the concave side (see Fig. 4.7). Finally, the Bezier spline curve is invariant under coordinate transforms and keeps its original shape intact.

One of the critical disadvantages of the Bezier spline curve is that it does not pass control points except the first and the last control points. While this would be a disadvantage of the Bezier spline curve, the complete local control imparts a big advantage for various engineering applications such as vehicle or airplane body design. In order to minimize the disadvantage, there are two techniques often used to gain more control on the shape of the Bezier spline curve. The first technique is to use different number of control points. This allows us to change the shape of the Bezier spline curve dramatically. As shown in Fig. 4.8, the more the number of control points used on the control net, the closer the shape between the control net and the spline curve.

The second technique is to control the number of clicks on the same control point. The larger the number of clicks on a control point, the closer the spline curve to that control point is (see Fig. 4.9). In addition to the blending function of Bezier

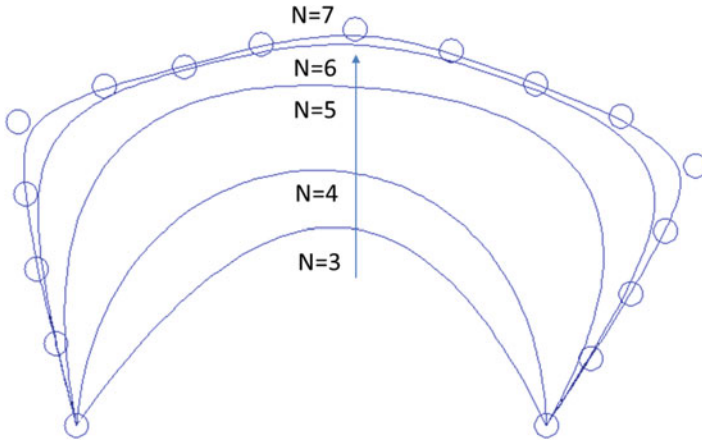


Fig. 4.8 Control of Bezier spline curve with number of control points

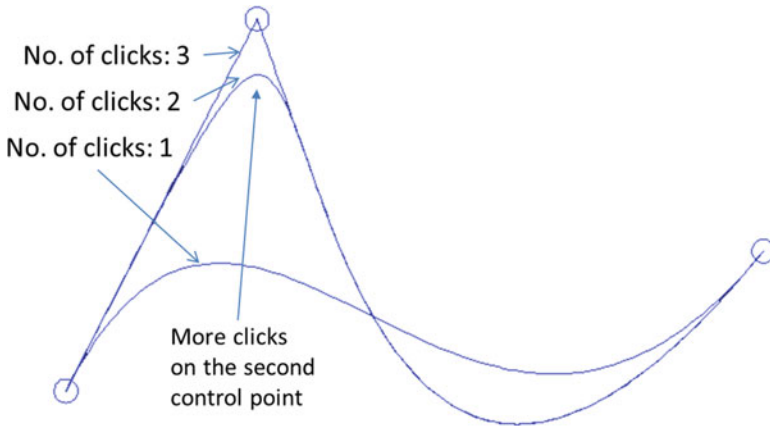


Fig. 4.9 The effect of number of clicks on the second control point

spline curve, other forms of blending functions are invented. The first one is a square-shaped blending function as shown in Fig. 4.10. Its mathematical expression can be formulated as below.

$$B_{1,i} = \begin{cases} 1 & \text{for } \lambda_i \leq \lambda \leq \lambda_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

Another form of the blending function is a triangular blending function as shown in Fig. 4.11. Its mathematical expression can be formulated as below.

Fig. 4.10 Square blending function

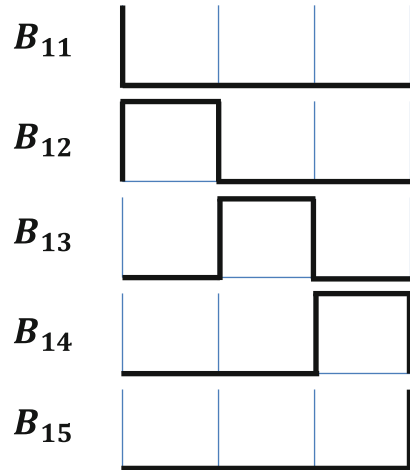
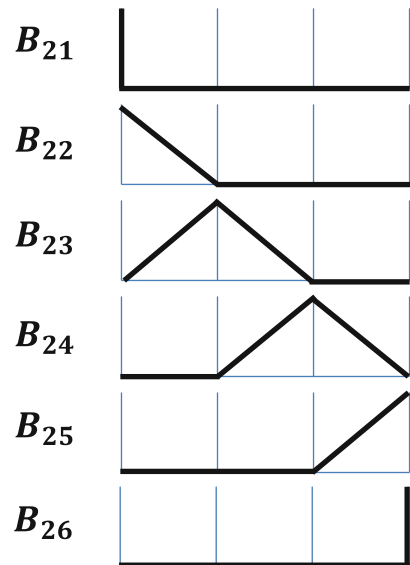


Fig. 4.11 Triangle blending function



$$B_{j,i} = \frac{(\lambda - \lambda_i) \cdot B_{j-1,i}}{\lambda_{j-1+i} - \lambda_i} + \frac{(\lambda_{i+j} - \lambda) \cdot B_{j-1,i+1}}{\lambda_{i+j} - \lambda_{i+1}}$$

Other blending functions can be invented to impart a unique property specific to a certain application.

Important Properties of Bezier Spline Curve

1. The degree of the Bezier spline curve is defined as the number of control points.
2. The Bezier spline curve is “generally” guided by control net, but does not interpolate between each control points.
3. The curve is “variation diminishing.”
4. $R(s)$ matches P_0 and P_{n-1} .
5. Tangent, R' matches control net at P_0 and P_{n-1} .
6. The generated curve stays on concave side of the control net.
7. Curve is invariant under transforms.

Sample Problem 4.6

For given three control points below, determine the equation of the Bezier spline curve.

$$P_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 3 \\ 7 \\ 4 \\ 1 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 8 \\ 9 \\ 1 \\ 1 \end{bmatrix}$$

Solution

Since there are three control points, N becomes 2, and j changes from 0 to 2. Three blending functions will become;

$$B_{2,0}(s) = \frac{2!S^0(1-s)^2}{0!2!} = (1-s)^2$$

$$B_{2,1}(s) = \frac{2!S^1(1-s)^1}{1!1!} = 2 \cdot s \cdot (1-s)$$

$$B_{2,2}(s) = \frac{2!S^2(1-s)^0}{2!0!} = S^2$$

Therefore, equations for x , y , and z of the Bezier spline curve are

$$x = (1-s)^2 \cdot x_0 + 2 \cdot s \cdot (1-s) \cdot x_1 + S^2 \cdot x_2$$

$$y = (1-s)^2 \cdot y_0 + 2 \cdot s \cdot (1-s) \cdot y_1 + S^2 \cdot y_2$$

$$z = (1-s)^2 \cdot z_0 + 2 \cdot s \cdot (1-s) \cdot z_1 + S^2 \cdot z_2$$

Now with the equations of the Bezier spline curve, we obtain the values of x , y , and z in Excel spreadsheet, changing the parameter, s , from 0 to 1. Projection graphs

on x - y , y - z , and z - x planes are also depicted using Excel spreadsheet. The above equations for x , y , and z need to be encoded in three columns under “R.”

Formulation of each blending function in Excel spreadsheet is as follows.

$$‘C4’ = \text{FACT}(\$C\$1)*\$A4^B4*(1 - \$A4)^{(\$C\$1-B4)}/(\text{FACT}(B4)*\text{FACT}(\$C\$1 - B4))$$

$$‘H4’ = \text{FACT}(\$C\$1)*\$A4^G4*(1 - \$A4)^{(\$C\$1-G4)}/(\text{FACT}(G4)*\text{FACT}(\$C\$1 - G4))$$

$$‘M4’ = \text{FACT}(\$C\$1)*\$A4^L4*(1 - \$A4)^{(\$C\$1-L4)}/(\text{FACT}(L4)*\text{FACT}(\$C\$1 - L4))$$

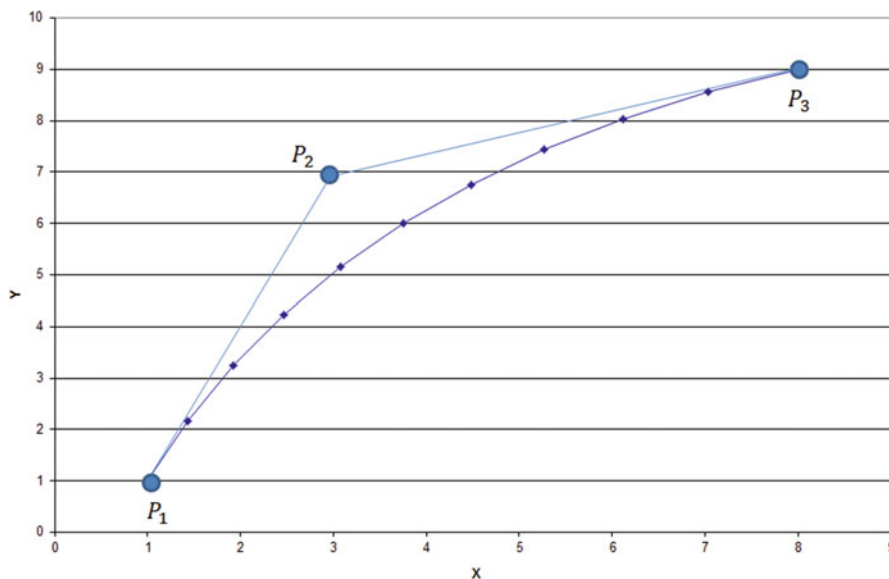
Formulation of x , y , and z in Excel spreadsheet is as follows.

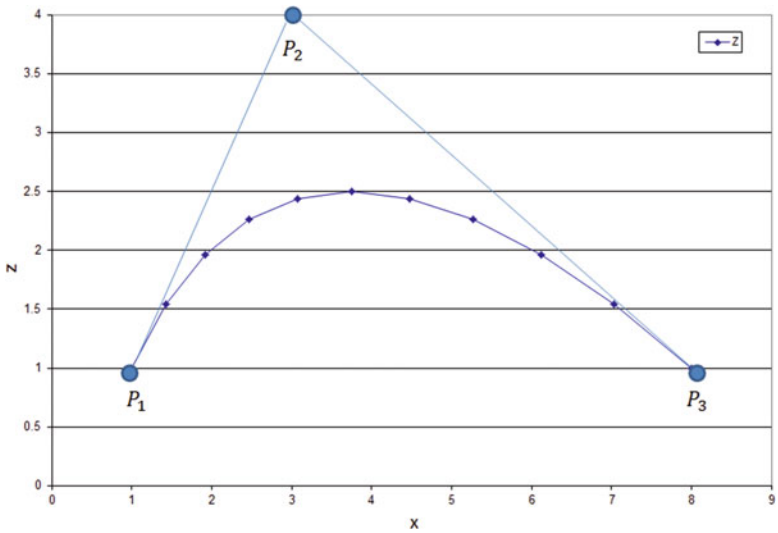
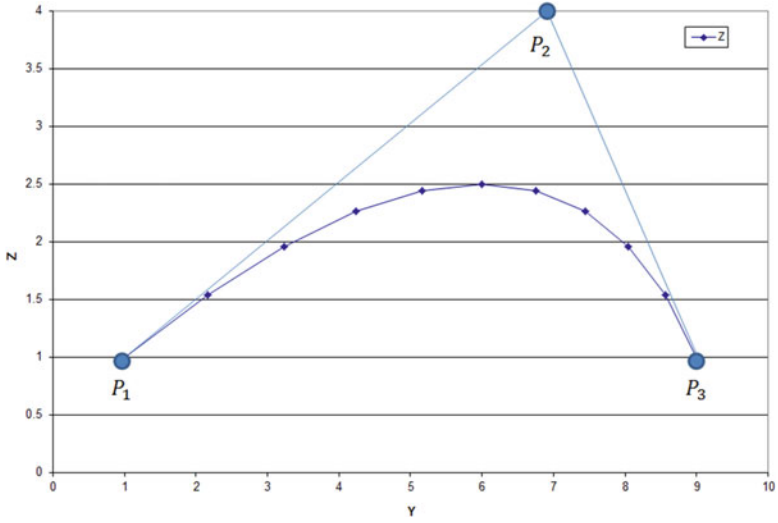
$$‘Q4’ = C4*D4 + H4*I4 + M4*N4$$

$$‘R4’ = C4*E4 + H4*J4 + M4*O4$$

$$‘S4’ = C4*F4 + H4*K4 + M4*P4$$

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1																			
2		N		2															
3	S	J	B20	X	P1			P2			P3			R	X	Y	Z		
4	0	0	1	1	1	1	1	0	3	7	4	2	0	8	9	1	1	1	1
5	0.1	0	0.81	1	1	1	1	0.18	3	7	4	2	0.01	8	9	1	1.43	2.16	1.54
6	0.2	0	0.64	1	1	1	1	0.32	3	7	4	2	0.04	8	9	1	1.92	3.24	1.96
7	0.3	0	0.49	1	1	1	1	0.42	3	7	4	2	0.09	8	9	1	2.47	4.24	2.26
8	0.4	0	0.36	1	1	1	1	0.48	3	7	4	2	0.16	8	9	1	3.08	5.16	2.44
9	0.5	0	0.25	1	1	1	1	0.5	3	7	4	2	0.25	8	9	1	3.75	6	2.5
10	0.6	0	0.16	1	1	1	1	0.48	3	7	4	2	0.36	8	9	1	4.48	6.76	2.44
11	0.7	0	0.09	1	1	1	1	0.42	3	7	4	2	0.49	8	9	1	5.27	7.44	2.26
12	0.8	0	0.04	1	1	1	1	0.32	3	7	4	2	0.64	8	9	1	6.12	8.04	1.96
13	0.9	0	0.01	1	1	1	1	0.18	3	7	4	2	0.81	8	9	1	7.03	8.56	1.54
14	1	0	0	1	1	1	1	0	3	7	4	2	1	8	9	1	8	9	1





Sample Problem 4.7

Draw a Bezier spline curve with AUTOCAD.

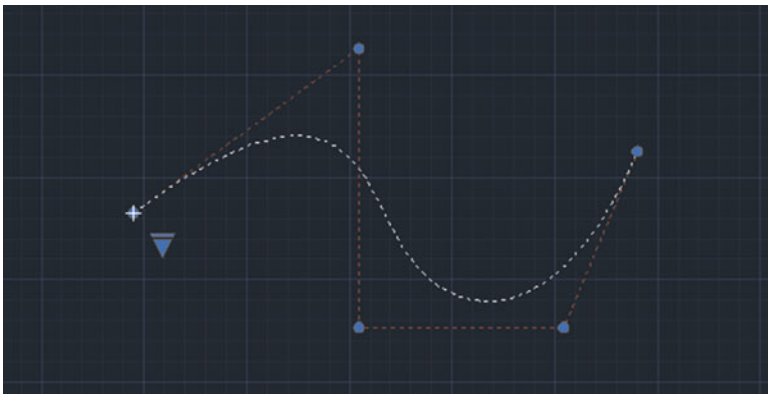
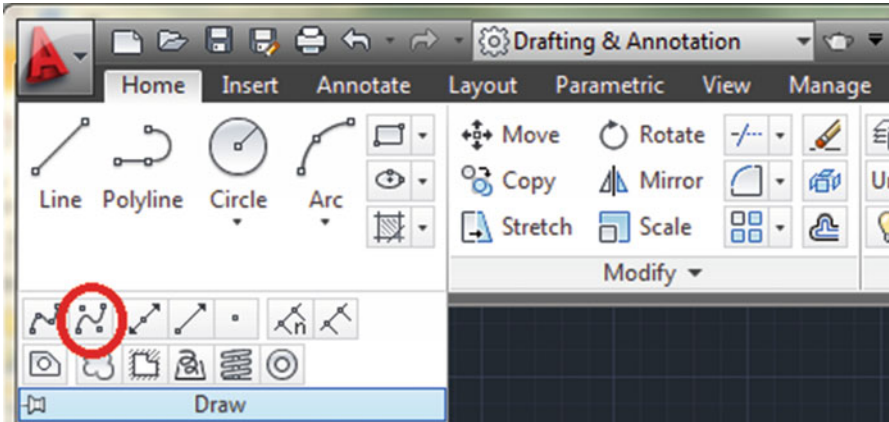
Solution

Make sure that the workspace is selected for “Drafting & Annotation.”

1. Click on the “Draw” panel to expand it.

2. Select the “Bezier spline curve” in the draw panel (marked with a circle in the above figure).
3. Click control points on the model space.
4. Push “Enter” key to finish operation

Notice that the curve is embraced by a control net.



An example of a Bezier spline curve

4.5 Surface Theory

A curved surface is a three-dimensional shape that forms elastically smooth surface created by multiple control points in space. One can think of a surface as an expansion of two curves into two directions perpendicular to each other in space (see Fig. 4.12). Two parameters are required to represent a curved surface in space.

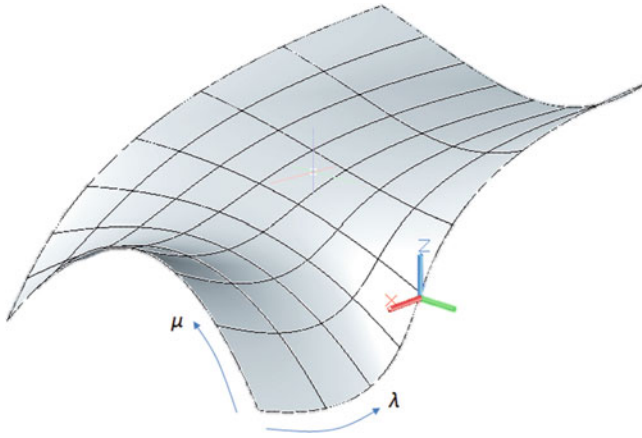


Fig. 4.12 A general curved surface

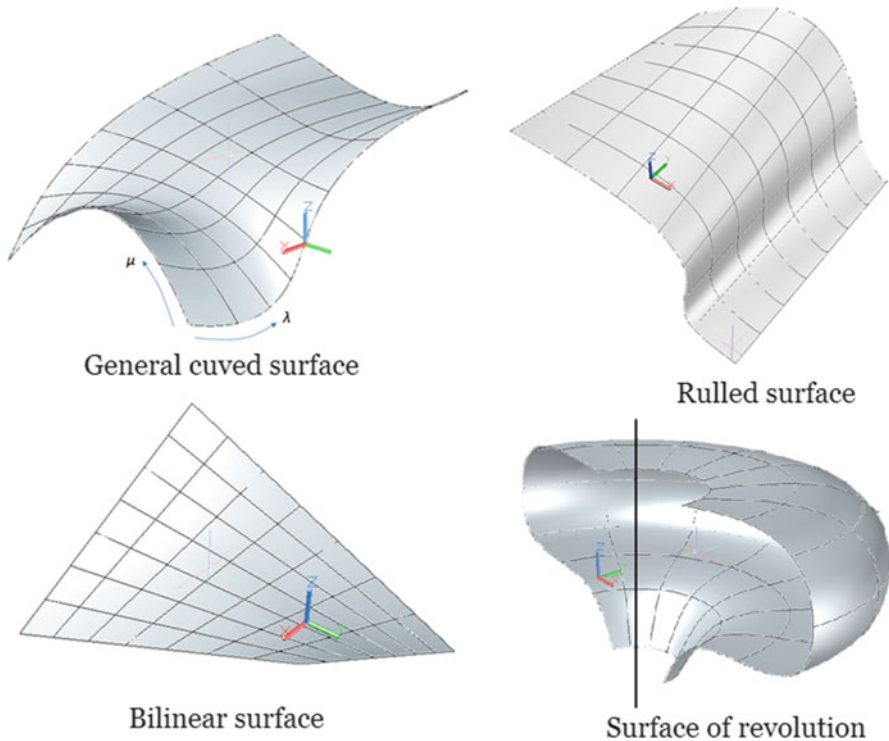


Fig. 4.13 Various curved surfaces

Various methods are available to create different types of surfaces including ruled surface, bilinear surface, general curved surface, swapped surface, or surface of revolution, etc. (see Fig. 4.13). In order to draw a general curved surface, more than four different curves are required in space, while a ruled surface and a bilinear

surface are interpolated surfaces between two curves in space. A swapped surface and a surface of revolution require a boundary curve and a reference curve (or a line) to be created in space. In this section, we limit our discussion to bilinear surface, ruled surface, and general curved surface.

4.5.1 Bilinear Surface

Bilinear surface is the simplest form of a surface that can be easily formulated mathematically. It starts with two straight lines created in space. Since each line requires two points, four points are required (see Fig. 4.14). A bilinear surface equation can be represented by a linear combination of two straight lines. The result is a twisted surface in space. If two space lines are parallel, then the surface becomes a simple plane.

Since a bilinear surface is a linear combination of two straight lines, the surface equation is expressed by a parametric linear combination of two lines such that:

$$R(\lambda, \mu) = [\text{line}(\lambda)_{\mu=0}, \text{line}(\lambda)_{\mu=1}] \cdot \begin{bmatrix} 1 - \mu \\ \mu \end{bmatrix} \quad (4.38)$$

or

$$R(\lambda, \mu) = [R(\lambda, 0), R(\lambda, 1)] \cdot \begin{bmatrix} 1 - \mu \\ \mu \end{bmatrix} \quad (4.39)$$

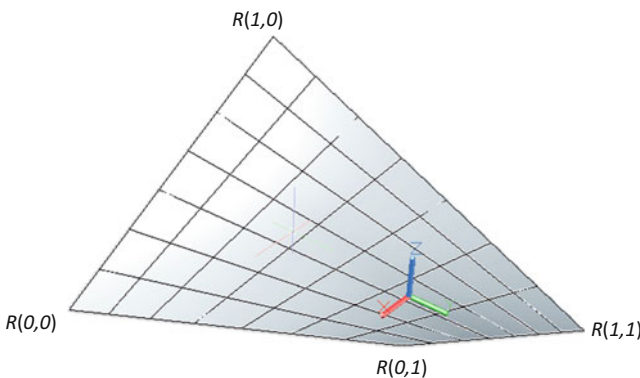


Fig. 4.14 Bilinear surface

By using (4.9), the above equation becomes;

$$R(\lambda, \mu) = \begin{bmatrix} 1 - \lambda & \lambda \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R}(0, 0) & \mathbf{R}(0, 1) \\ \mathbf{R}(1, 0) & \mathbf{R}(1, 1) \end{bmatrix} \cdot \begin{bmatrix} 1 - \mu \\ \mu \end{bmatrix} \tag{4.40}$$

In order to facilitate linear combination, (4.9), the parametric line equation, has been transposed to facilitate multiplication of two matrices. $\mathbf{R}(0, 0)$ and $\mathbf{R}(1, 0)$ are the points of the first line, while $\mathbf{R}(0, 1)$ and $\mathbf{R}(1, 1)$ are the points for the second line respectively.

Sample Problem 4.8

For two lines, each of which is defined by two points in space, generate a bilinear surface.

$$\begin{aligned} \text{Line 1 : } \mathbf{R}(0, 0) &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, & \mathbf{R}(0, 1) &= \begin{bmatrix} 0 \\ 50 \\ 100 \\ 1 \end{bmatrix}, \\ \text{Line 2 : } \mathbf{R}(1, 0) &= \begin{bmatrix} 50 \\ 0 \\ -100 \\ 1 \end{bmatrix}, & \mathbf{R}(1, 1) &= \begin{bmatrix} 100 \\ 100 \\ 0 \\ 1 \end{bmatrix} \end{aligned}$$

Solution

If we expand (4.40), it becomes;

$$\begin{aligned} R(\lambda, \mu) &= [(1 - \lambda)\mathbf{R}(0, 0) + \lambda\mathbf{R}(1, 0)](1 - \mu) + [\lambda\mathbf{R}(0, 1) + \lambda\mathbf{R}(1, 1)] \cdot \begin{bmatrix} 1 - \mu \\ \mu \end{bmatrix} \\ &= [(1 - \lambda)\mathbf{R}(0, 0) + \lambda\mathbf{R}(1, 0)] \cdot (1 - \mu) + [(1 - \lambda)\mathbf{R}(0, 1) + \lambda\mathbf{R}(1, 1)] \cdot \mu \end{aligned}$$

or

$$\begin{aligned} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} &= \left[(1 - \lambda) \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{0,0} + \lambda \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{1,0} \right] \cdot (1 - \mu) \\ &+ \left[(1 - \lambda) \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{0,1} + \lambda \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{1,1} \right] \cdot \mu \end{aligned}$$

Therefore, the equation of each element will become;

$$x = x_{0,0} \cdot (1 - \lambda)(1 - \mu) + x_{1,0} \cdot \lambda(1 - \mu) + x_{0,1} \cdot (1 - \lambda)\mu + x_{1,1} \cdot \lambda\mu$$

$$y = y_{0,0} \cdot (1 - \lambda)(1 - \mu) + y_{1,0} \cdot \lambda(1 - \mu) + y_{0,1} \cdot (1 - \lambda)\mu + y_{1,1} \cdot \lambda\mu$$

$$z = z_{0,0} \cdot (1 - \lambda)(1 - \mu) + z_{1,0} \cdot \lambda(1 - \mu) + z_{0,1} \cdot (1 - \lambda)\mu + z_{1,1} \cdot \lambda\mu$$

or

$$x = 50 \cdot (1 - \lambda)\mu + 100 \cdot \lambda\mu$$

$$y = 50 \cdot \lambda(1 - \mu) + 100 \cdot \lambda\mu$$

$$z = 100 \cdot \lambda(1 - \mu) - 100 \cdot (1 - \lambda)\mu$$

Now with the equations of the bilinear surface, we put together an Excel spreadsheet, changing the parameters, λ and μ from 0 to 1 respectively. The projection graphs on x - y , y - z , and z - x planes are also illustrated. Below is the spreadsheet expression of the x component.

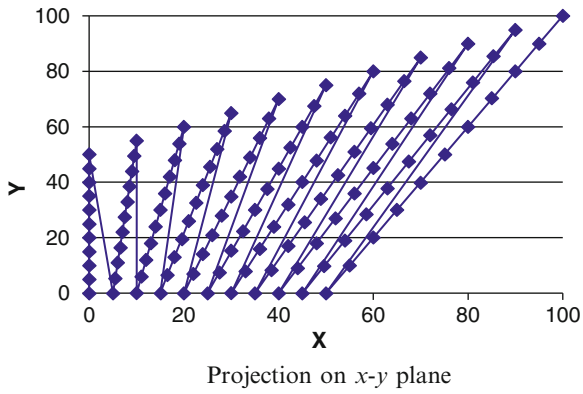
Cell 'C9' = (1 - B9)*(1 - A9)*\$C\$3 + (1 - B9)*A9*\$G\$3 + B9*(1 - A9)*\$E\$3 + B9*A9*\$I\$3

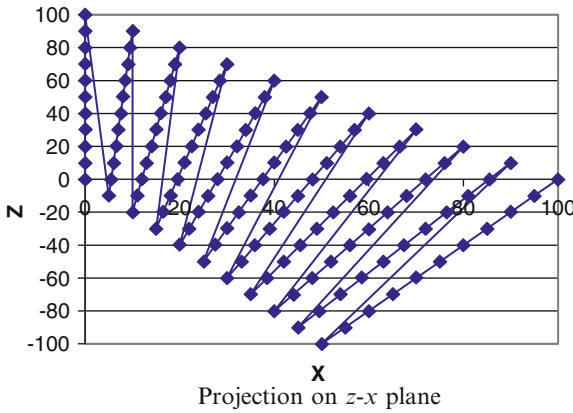
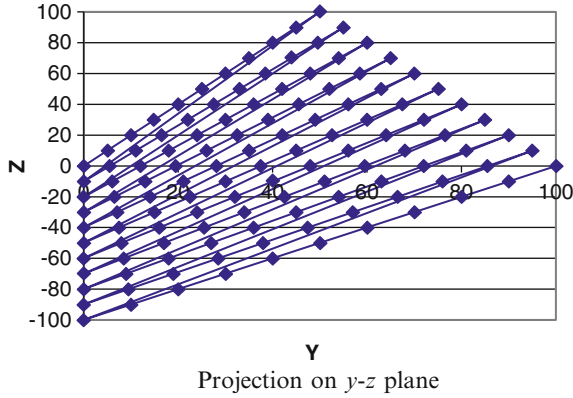
Cell 'D9' = (1 - B9)*(1 - A9)*\$C\$4 + (1 - B9)*A9*\$G\$4 + B9*(1 - A9)*\$E\$4 + B9*A9*\$I\$4

Cell 'E9' = (1 - B9)*(1 - A9)*\$C\$5 + (1 - B9)*A9*\$G\$5 + B9*(1 - A9)*\$E\$5 + B9*A9*\$I\$5

	A	B	C	D	E	F	G	H	I
1	Bilinear surface								
2									
3		x00	0	x01	0	x10	50	x11	100
4		y00	0	y01	50	y10	0	y11	100
5		z00	0	z01	100	z10	-100	z11	0

7	A	B	C	D	E
8	Lamda	Mu	x	y	z
9	0	0	0	0	0
10	0	0.1	0	5	10
11	0	0.2	0	10	20
12	0	0.3	0	15	30
13	0	0.4	0	20	40
14	0	0.5	0	25	50
15	0	0.6	0	30	60
16	0	0.7	0	35	70
17	0	0.8	0	40	80
18	0	0.9	0	45	90
19	0	1	0	50	100
20	0.1	0	5	0	-10
21	0.1	0.1	5.5	5.5	0
22	0.1	0.2	6	11	10
⋮					
25	1	0.7	85	70	-30
26	1	0.8	90	80	-20
27	1	0.9	95	90	-10
28	1	1	100	100	0





4.5.2 Ruled Surface

Similar to the bilinear surface, a ruled surface can be represented by a linear combination of two curved lines. The result is a twisted surface interpolated in space between two curved lines whose matching points are connected by a straight line (see Fig. 4.15). Since a ruled surface is a linear combination of two curved lines, the surface equation can be expressed by a parametric linear combination such that;

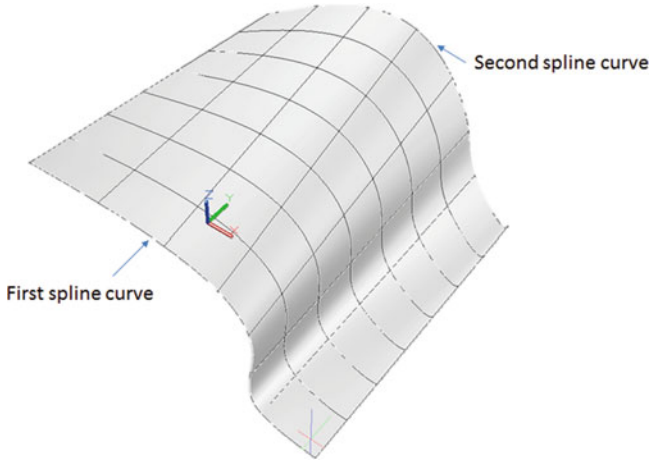


Fig. 4.15 Ruled surface

$$R(\lambda, \mu) = [\text{Curve}_1(\lambda), \text{Curve}_2(\lambda)] \cdot \begin{bmatrix} 1 - \mu \\ \mu \end{bmatrix} \tag{4.41}$$

or

$$R(\lambda, \mu) = [R(\lambda)_1, R(\lambda)_2] \cdot \begin{bmatrix} 1 - \mu \\ \mu \end{bmatrix} \tag{4.42}$$

Sample Problem 4.9

For two space curves given below, generate a ruled surface.

The first curve:

$$\begin{aligned} x(\lambda) &= 0.75 \cdot \lambda^3 + 2.25 \cdot \lambda + 1 \\ y(\lambda) &= 2.0 \cdot \lambda + 1 \\ z(\lambda) &= 1.25 \cdot \lambda^3 - 1.25 \cdot \lambda + 1 \end{aligned}$$

The second curve:

$$\begin{aligned} x(\lambda) &= 0.5 \cdot \lambda^3 + 6.259 \cdot \lambda^2 + 0.5 \cdot \lambda + 8 \\ y(\lambda) &= 10 \cdot \lambda^3 + 0.004 \cdot \lambda^2 + 2.0 \cdot \lambda - 5 \\ z(\lambda) &= -1.255 \cdot \lambda^3 + 7 \cdot \lambda^2 + 2.5 \cdot \lambda \end{aligned}$$

Solution

If we expand (4.42), it becomes;

$$R(\lambda, \mu) = R(\lambda)_1 \cdot (1 - \mu) + R(\lambda)_2 \cdot \mu$$

Therefore, the equation of each element will become;

$$x = (0.75 \cdot \lambda^3 + 2.25 \cdot \lambda + 1) \cdot (1 - \mu) + (0.5 \cdot \lambda^3 + 6.259 \cdot \lambda^2 + 0.5 \cdot \lambda + 8) \cdot \mu$$

$$y = (2.0 \cdot \lambda + 1) \cdot (1 - \mu) + (10 \cdot \lambda^3 + 0.004 \cdot \lambda^2 + 2.0 \cdot \lambda - 5) \cdot \mu$$

$$z = (1.25 \cdot \lambda^3 - 1.25 \cdot \lambda + 1) \cdot (1 - \mu) + (-1.255 \cdot \lambda^3 + 7 \cdot \lambda^2 + 2.5 \cdot \lambda) \cdot \mu$$

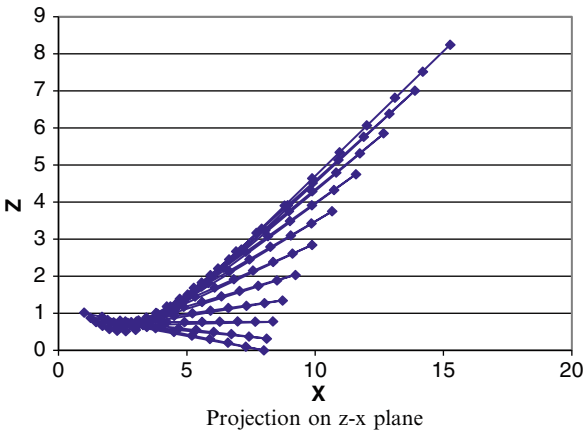
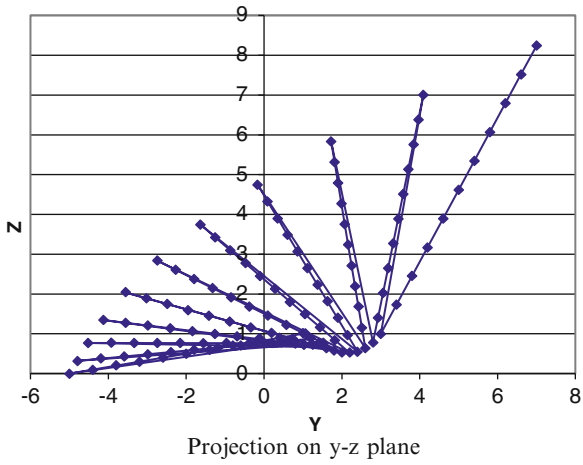
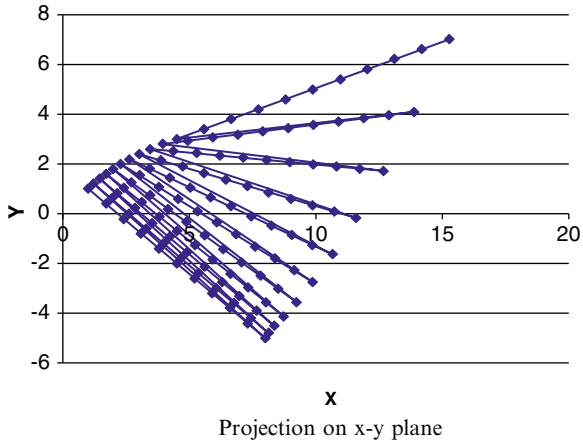
Now with the equations of the ruled surface, we put together an Excel spreadsheet, changing the parameter, λ and μ from 0 to 1 respectively. The projection graphs on x - y , y - z , and z - x planes are also depicted. Below is the spreadsheet expression of the x , y , and z components.

Cell 'C9' = $(0.75 \cdot A9^3 + 2.25 \cdot A9 + 1) \cdot (1 - B9) + (0.5 \cdot A9^3 + 6.259 \cdot A9^2 + 0.5 \cdot A9 + 8) \cdot B9$

Cell 'D9' = $(2.0 \cdot A9 + 1) \cdot (1 - B9) + (10 \cdot A9^3 + 0.004 \cdot A9^2 + 2.0 \cdot A9 - 5) \cdot B9$

Cell 'E9' = $(1.25 \cdot A9^3 - 1.25 \cdot A9 + 1) \cdot (1 - B9) + (-1.255 \cdot A9^3 + 7 \cdot A9^2 + 2.5 \cdot A9) \cdot B9$

	A	B	C	D	E
7			:		
8	Lamda	Mu	x	y	z
9	0	0	(.75+8)*B9	1	1
10	0	0.1	1.7	0.4	0.9
11	0	0.2	2.4	-0.2	0.8
12	0	0.3	3.1	-0.8	0.7
13	0	0.4	3.8	-1.4	0.6
14	0	0.5	4.5	-2	0.5
15	0	0.6	5.2	-2.6	0.4
16	0	0.7	5.9	-3.2	0.3
17	0	0.8	6.6	-3.8	0.2
18	0	0.9	7.3	-4.4	0.1
19	0	1	8	-5	0
20	0.1	0	1.22625	1.2	0.87625
21	0.1	0.1	1.914934	0.601004	0.8204995
			:		
127	1	0.8	13.1072	6.2032	6.796
128	1	0.9	14.1831	6.6036	7.5205
129	1	1	15.259	7.004	8.245



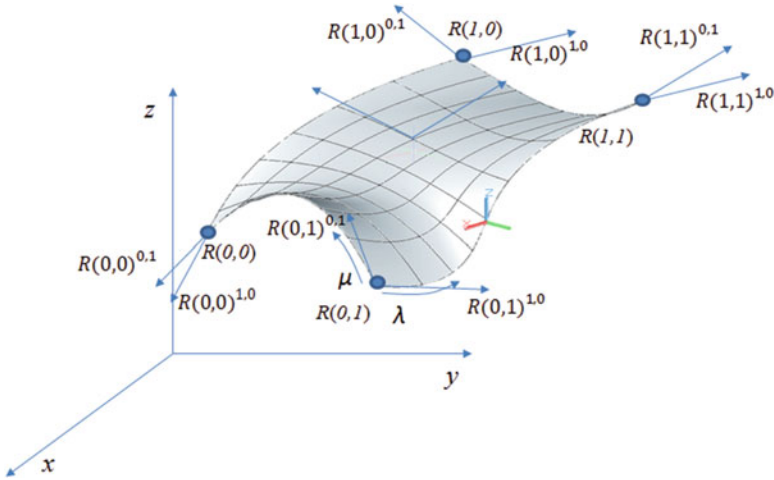


Fig. 4.16 Curved surface

4.5.3 General Curved Surface

A general curved surface is a freeform surface that represents an elastically smooth surface defined by more than four curves in space. There are two most popular surfaces prevailing in engineering applications: BiCubic surface and NURBS surface. In order to define a BiCubic surface, four control points are required. In addition, two tangent vectors are required for each curve to define the slope of the curve at both ends, thus eight tangent vectors are required (see Fig. 4.16). Mathematical definition of each tangent in the figure is shown below.

$$R(\lambda, \mu)^{1,0} = \frac{\partial R}{\partial \lambda}$$

$$R(\lambda, \mu)^{0,1} = \frac{\partial R}{\partial \mu}$$

$$R(\lambda, \mu)^{1,1} = \frac{\partial R}{\partial \lambda \partial \mu}$$

A general curved surface is a collection of a basic surface that is uniquely defined by four space curves. The basic surface is often called “Coon’s patch,” which is a linear combination of two Hermite spline curves. We start with the cubic polynomial coefficient from the Hermite spline curve (4.26).

$$[C_j] = [P_j | P'_j | P_{j+1} | P'_{j+1}] \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

or

$$[C_j] = [P_j | P_{j+1} | P'_j | P'_{j+1}] \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

Notice that P'_j and P_{j+1} switched their position for convenience. Notice also that,

$$\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}^{-1},$$

Therefore,

$$R(\lambda) = [C_j] \cdot \begin{bmatrix} \lambda^3 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix} = [P_j | P_{j+1} | P'_j | P'_{j+1}] \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \lambda^3 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}$$

To outline the Coon's patch, let us define a column vector of B as below,

$$\begin{bmatrix} B_1(v) \\ B_2(v) \\ B_3(v) \\ B_4(v) \end{bmatrix} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix}. \tag{4.43}$$

Then, by using (4.43), we obtain the following equation.

$$R(\lambda) = [R(0) | R(1) | R(0)' | R(1)']_{\lambda} \cdot \begin{bmatrix} B_1(\lambda) \\ B_2(\lambda) \\ B_3(\lambda) \\ B_4(\lambda) \end{bmatrix} \tag{4.44}$$

If we define another cubic spline curve with the different parameter, μ , then we obtain another equation as below.

$$R(\mu) = \left[R(0) \mid R(1) \mid R(0)' \mid R(1)' \right]_{\mu} \cdot \begin{bmatrix} B_1(\mu) \\ B_2(\mu) \\ B_3(\mu) \\ \mathbf{1} \end{bmatrix} \tag{4.45}$$

Notice that the control points and the tangent vectors in each equation are different. Now, if we combine them linearly by the following equation, we obtain the first form of the Coon’s patch.

$$R(\lambda, \mu) = R^T(\lambda) \cdot R(\mu) \tag{4.46}$$

or

$$= [B_1(\lambda) \ B_2(\lambda) \ B_3(\lambda) \ B_4(\lambda)] \begin{bmatrix} R(0) \\ R(1) \\ R(0)' \\ R(1)' \end{bmatrix}_{\lambda} \left[R(0) \mid R(1) \mid R(0)' \mid R(1)' \right]_{\mu} \cdot \begin{bmatrix} B_1(\mu) \\ B_2(\mu) \\ B_3(\mu) \\ \mathbf{1} \end{bmatrix} \tag{4.47}$$

The above equation is a matrix equation that can be simplified to the following matrix form.

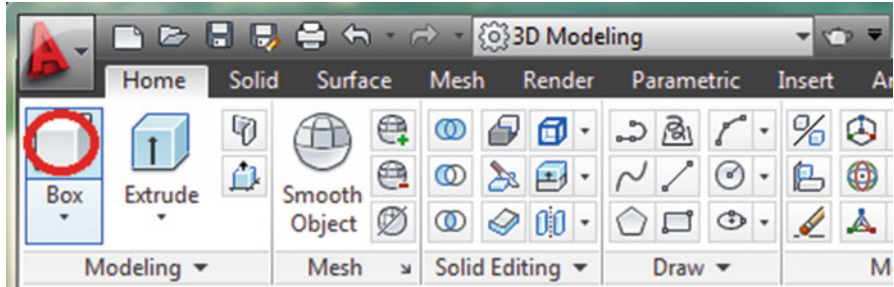
$$= [B_1(\lambda) \ B_2(\lambda) \ B_3(\lambda) \ B_4(\lambda)] \begin{bmatrix} R(0, 0) & R(0, 1) & R(0, 0)^{0,1} & R(0, 0)^{0,1} \\ R(1, 0) & R(1, 1) & R(0, 0)^{0,1} & R(0, 0)^{0,1} \\ R(0, 0)^{1,0} & R(0, 1)^{1,0} & R(0, 0)^{1,1} & R(0, 0)^{1,1} \\ R(1, 0)^{1,0} & R(1, 1)^{1,0} & R(0, 0)^{1,1} & R(0, 0)^{1,1} \end{bmatrix} \begin{bmatrix} B_1(\mu) \\ B_2(\mu) \\ B_3(\mu) \\ \mathbf{1} \end{bmatrix} \tag{4.48}$$

The partial derivative terms for both λ and μ in the above equation is often called a degree of twist or changing rate of the tangent of the surface, which are set to be zero unless otherwise particularly specified. In order to generate a free form of a surface, the above Coon’s patch can be added on to create a complex surface. For instance, there are 49 Coon’s patches in Fig. 4.16. Coon’s patch, in general, is used to create a patch to smoothly connect several surface edges in space. Therefore, the matrix coefficients of the Coon’s patch have to be known a priory. To that end, all of the components other than four control points in (4.48) have to be calculated by proper geometric constraints following a similar procedure introduced in Sect. 4.3. The study of determining the coefficients of Coon’s patch is skipped since it is beyond our scope of study.

Sample Problem 4.10

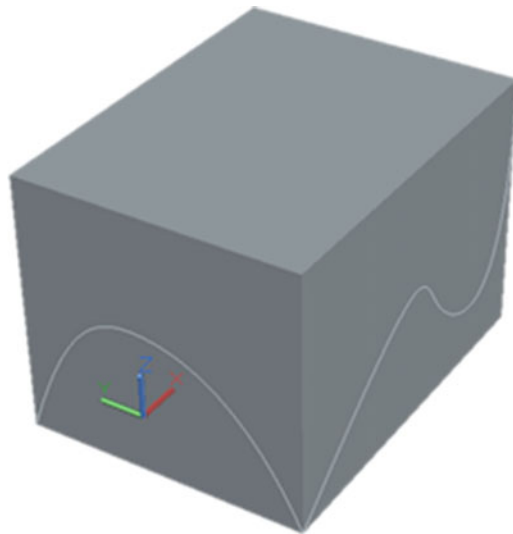
Create a general curved surface with AUTOCAD.

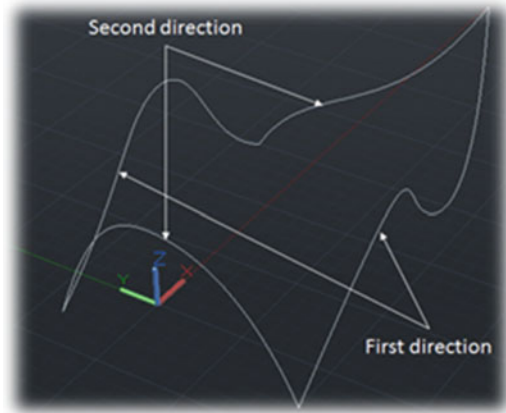
Solution



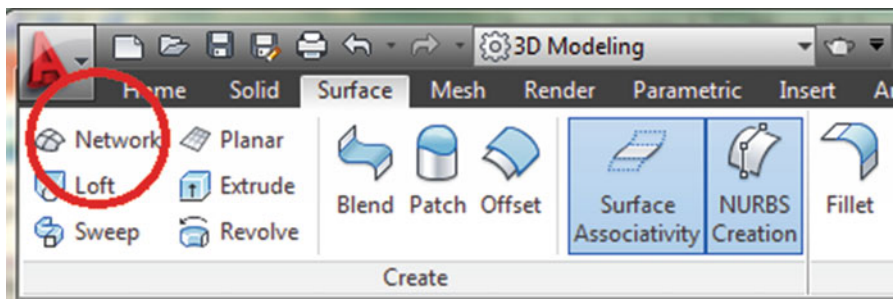
Make sure that the workspace is selected for “3D Modeling.”

1. Click on the “Home” panel and select Box.
2. Draw a 3D box.
3. Draw four curves on the face of each side, making sure that each end meets each other.

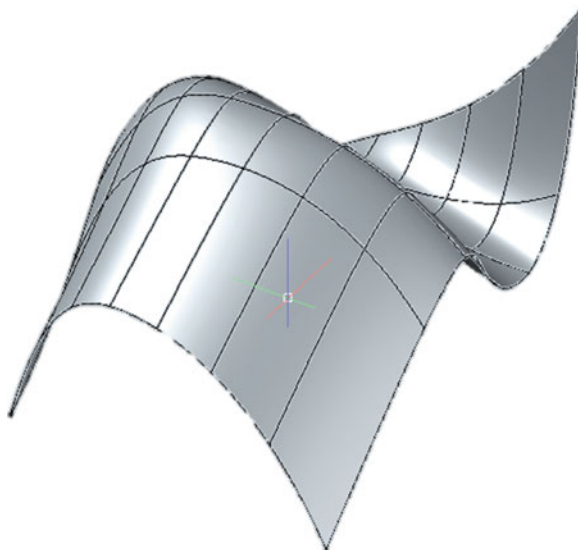




4. Click on “Surface” tag and select the “Network” in the “Create” panel



5. Select first and second directions by clicking the corresponding lines indicated in the above wireframe, and enter. As a result, a freeform surface is obtained as below.



Exercise Problem 4.1

For the given three control points below, draw the Hermite spline curve in x - y , y - z , and z - x planes. Assume $P''_1 = P''_3 = \mathbf{0}$

$$P_1 = \begin{bmatrix} 2 \\ 4 \\ 3 \\ 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 3 \\ 1 \\ 2 \\ 1 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 1 \end{bmatrix}$$

Solution

By using (4.34), we obtain the following equations for P'_1 , P'_2 and P'_3 .

$$P'_1 = 1.75 \cdot \begin{bmatrix} 1 \\ -3 \\ -1 \\ 0 \end{bmatrix} - 0.5 \cdot \begin{bmatrix} -1 \\ -2 \\ 1 \\ 0 \end{bmatrix} + 0.25 \cdot \begin{bmatrix} -2 \\ 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1.75 \\ -4 \\ -1.75 \\ 0 \end{bmatrix}$$

$$P'_2 = -0.5 \cdot \begin{bmatrix} 1 \\ -3 \\ 1 \\ 0 \end{bmatrix} + 1 \cdot \begin{bmatrix} -1 \\ -2 \\ 1 \\ 0 \end{bmatrix} - 0.5 \cdot \begin{bmatrix} -2 \\ 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ -1 \\ -0.5 \\ 0 \end{bmatrix}$$

$$P'_3 = 0.249 \cdot \begin{bmatrix} 1 \\ -3 \\ 1 \\ 0 \end{bmatrix} - 0.5 \cdot \begin{bmatrix} -1 \\ -2 \\ 1 \\ 0 \end{bmatrix} + 1.749 \cdot \begin{bmatrix} -2 \\ 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -2.749 \\ 2.002 \\ 3.247 \\ 0 \end{bmatrix}$$

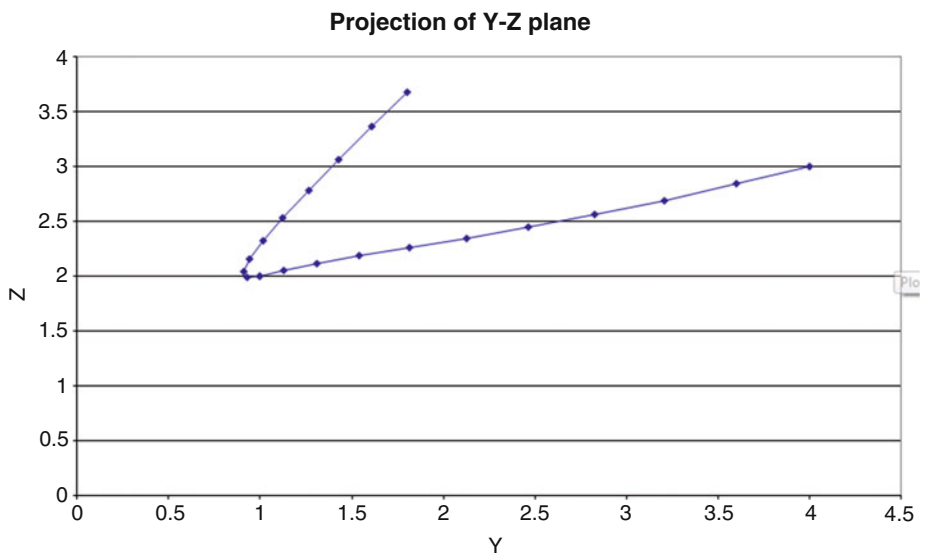
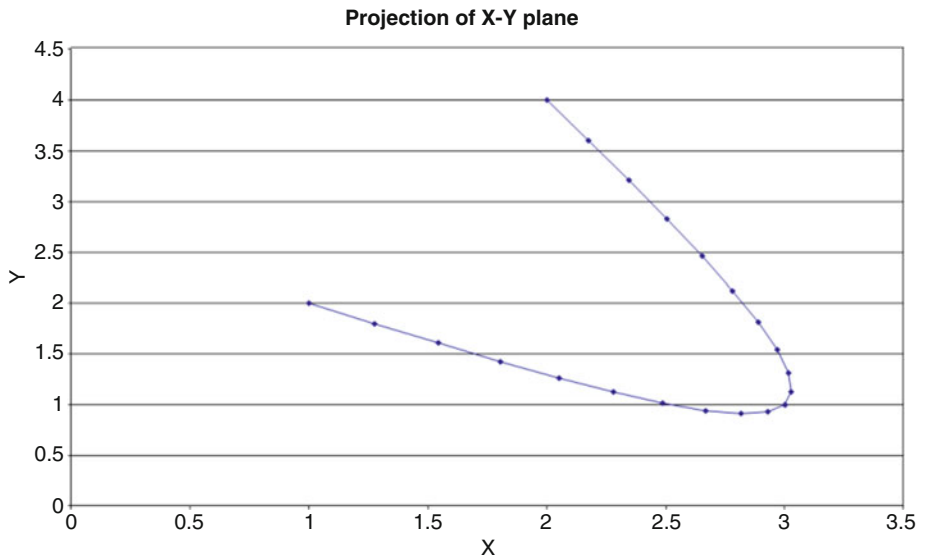
Therefore, with the tangent vectors, the cubic polynomial coefficient, C_j , can be determined by (4.25).

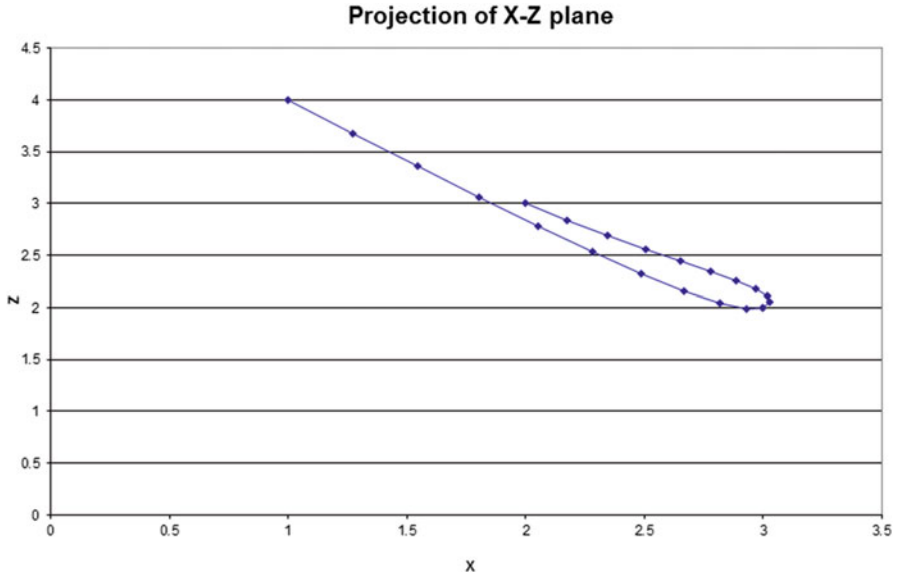
$$[C_1] = \begin{bmatrix} P_1 & P'_1 & P_2 & P'_2 \end{bmatrix} \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 1.75 & 3 & -0.5 \\ 4 & -4 & 1 & -1 \\ 3 & -1.75 & 2 & -0.5 \\ 1 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -0.75 & 0 & 1.75 & 2 \\ 1 & 0 & -4 & 4 \\ -0.25 & 1 & -1.75 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Likewise,

$$[C_2] = \begin{bmatrix} 0.751 & -2.251 & -0.5 & 3 \\ -0.998 & 2.998 & -1 & 1 \\ -1.253 & 3.753 & -0.5 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$





Exercise Problem 4.2

For the given three control points below, determine cubic polynomial coefficients of the Hermite spline curve. Assume $P'_1 = P'_3 = 0$

$$P_1 = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 6 \\ 3 \\ 2 \\ 1 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 3 \\ 2 \\ 8 \\ 1 \end{bmatrix}$$

Solution

By using (4.34), we obtain P'_1, P'_2 and P'_3 .

$$P'_1 = \begin{bmatrix} 5.75 \\ 0.25 \\ -4 \\ 0 \end{bmatrix}, \quad P'_2 = \begin{bmatrix} -2.5 \\ -1.5 \\ 8 \\ 0 \end{bmatrix}, \quad P'_3 = \begin{bmatrix} -4.75 \\ -1.25 \\ 8 \\ 0 \end{bmatrix}$$

Therefore, with the tangent vectors, the cubic polynomial coefficient, C_j , can be determined by (4.25).

$$[C_1] = \begin{bmatrix} -4.75 & 3 & 5.75 & 2 \\ -10.75 & 10 & 0.25 & 3 \\ 8 & -6 & -4 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[C_2] = \begin{bmatrix} -1.25 & 0.75 & -2.5 & 6 \\ 0.75 & 1.25 & -1.5 & 3 \\ 4 & -6 & 8 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Exercise Problem 4.3

For the given eight control points below, determine the equations of the blending functions for a Bezier spline curve.

$$P_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 5 \\ 5 \\ 7 \\ 1 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 5 \\ 6 \\ 4 \\ 1 \end{bmatrix}, \quad P_4 = \begin{bmatrix} 4 \\ 7 \\ 8 \\ 1 \end{bmatrix}, \quad P_5 = \begin{bmatrix} 6 \\ 7 \\ 9 \\ 1 \end{bmatrix}$$

Solution

Since there are five control points, N becomes 4, and j changes from 0 to 4. Five blending functions will become;

$$B_{4,0}(s) = \frac{4!S^0(1-s)^4}{0!4!} = (1-s)^4$$

$$B_{4,1}(s) = \frac{4!S^1(1-s)^3}{1!3!} = 4 \cdot s^1 \cdot (1-s)^3$$

$$B_{4,2}(s) = \frac{4!S^2(1-s)^2}{2!2!} = 6 \cdot s^2 \cdot (1-s)^2$$

$$B_{4,3}(s) = \frac{4!S^3(1-s)^1}{3!1!} = 4 \cdot s^3 \cdot (1-s)^1$$

$$B_{4,4}(s) = \frac{4!S^4(1-s)^0}{4!0!} = s^4$$

Exercise Problem 4.4

For the given eight control points below, determine the equations of the blending functions of a Bezier spline curve.

$$\begin{aligned}
 P_1 &= \begin{bmatrix} 2 \\ 3 \\ 1 \\ 1 \end{bmatrix}, & P_2 &= \begin{bmatrix} 3 \\ -4 \\ 3 \\ 1 \end{bmatrix}, & P_3 &= \begin{bmatrix} 1 \\ 2 \\ 1 \\ 1 \end{bmatrix}, & P_4 &= \begin{bmatrix} 2 \\ -4 \\ 3 \\ 1 \end{bmatrix}, & P_5 &= \begin{bmatrix} 1 \\ 3 \\ 2 \\ 1 \end{bmatrix}, \\
 P_6 &= \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix}, & P_7 &= \begin{bmatrix} 2 \\ -4 \\ 1 \\ 1 \end{bmatrix}, & P_8 &= \begin{bmatrix} 3 \\ 4 \\ 2 \\ 1 \end{bmatrix}
 \end{aligned}$$

Solution

Since there are eight control points, N becomes 7, and j changes from 0 to 7, eight blending functions will become;

$$\begin{aligned}
 B_{7,0}(s) &= \frac{7!S^0(1-s)^7}{0!7!} = (1-s)^7 \\
 B_{7,1}(s) &= \frac{7!S^1(1-s)^6}{1!6!} = 7 \cdot s^1 \cdot (1-s)^6 \\
 B_{7,2}(s) &= \frac{7!S^2(1-s)^5}{2!5!} = 21 \cdot s^2 \cdot (1-s)^5 \\
 B_{7,3}(s) &= \frac{7!S^3(1-s)^4}{3!4!} = 35 \cdot s^3 \cdot (1-s)^4 \\
 B_{7,4}(s) &= \frac{7!S^4(1-s)^3}{4!3!} = 35 \cdot s^4 \cdot (1-s)^3 \\
 B_{7,5}(s) &= \frac{7!S^5(1-s)^2}{5!2!} = 21 \cdot s^5 \cdot (1-s)^2 \\
 B_{7,6}(s) &= \frac{7!S^6(1-s)^1}{6!1!} = 7 \cdot s^6 \cdot (1-s)^1 \\
 B_{7,7}(s) &= \frac{7!S^7(1-s)^0}{7!0!} = s^7
 \end{aligned}$$

Exercise Problem 4.5

For the given three control points below, draw Hermite spline curves in x - y , y - z , and z - x planes. Assume $P_1'' = P_3'' = \mathbf{0}$

$$P_1 = \begin{bmatrix} -3 \\ 5 \\ 2 \\ 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 9 \\ 7 \\ -2 \\ 1 \end{bmatrix}, \quad P_3 = \begin{bmatrix} -4 \\ 3 \\ 5 \\ 1 \end{bmatrix}$$

Exercise Problem 4.6

For the given three control points below, determine cubic polynomial coefficients of the Hermite spline curve. Assume $P_1' = P_3' = \mathbf{0}$

$$P_1 = \begin{bmatrix} 3 \\ -2 \\ 4 \\ 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 5 \\ 1 \\ -3 \\ 1 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 4 \\ 2 \\ 1 \\ 1 \end{bmatrix}$$

Exercise Problem 4.7

For the given three control points below, determine the equation of a Bezier spline curve.

$$P_1 = \begin{bmatrix} 3 \\ 1 \\ 4 \\ 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 5 \\ 6 \\ 7 \\ 1 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 6 \\ -2 \\ 3 \\ 1 \end{bmatrix}$$

Exercise Problem 4.8

For the given five control points below, determine the equations of the blending functions of a Bezier spline curve.

$$P_1 = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 3 \\ -1 \\ 2 \\ 1 \end{bmatrix}, \quad P_3 = \begin{bmatrix} -4 \\ 3 \\ 1 \\ 1 \end{bmatrix}, \quad P_4 = \begin{bmatrix} 3 \\ -5 \\ 2 \\ 1 \end{bmatrix}, \quad P_5 = \begin{bmatrix} 10 \\ 4 \\ 1 \\ 1 \end{bmatrix}$$

Exercise Problem 4.9

Below are four spatial points by which two spatial lines are defined. Draw the bilinear surface equation and generate projections of the surface on x - y , y - z , and z - x planes.

$$R(\mathbf{0}, \mathbf{0}) = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \quad R(\mathbf{0}, \mathbf{1}) = \begin{bmatrix} \mathbf{10} \\ -\mathbf{5} \\ \mathbf{20} \\ \mathbf{1} \end{bmatrix}, \quad R(\mathbf{1}, \mathbf{0}) = \begin{bmatrix} \mathbf{5} \\ \mathbf{0} \\ \mathbf{10} \end{bmatrix}, \quad R(\mathbf{1}, \mathbf{1}) = \begin{bmatrix} \mathbf{10} \\ \mathbf{10} \\ \mathbf{10} \\ \mathbf{1} \end{bmatrix}$$

Exercise Problem 4.10

For two space curves given below, generate a ruled surface.

The first curve:

$$\begin{aligned} x(\lambda) &= 0.5 \cdot \lambda^3 + \mathbf{0.3} \cdot \lambda \\ y(\lambda) &= 1.0 \cdot \lambda + 2 \\ z(\lambda) &= -3.5 \cdot \lambda^3 - 2 \cdot \lambda - 4 \end{aligned}$$

The second curve:

$$\begin{aligned} x(\lambda) &= 4 \cdot \lambda^3 + \mathbf{7} \cdot \lambda^2 + \mathbf{0.7} \cdot \lambda \\ y(\lambda) &= -2 \cdot \lambda^3 + \mathbf{5} \cdot \lambda^2 + \mathbf{2.5} \cdot \lambda - 3 \\ z(\lambda) &= -4.5 \cdot \lambda^3 + \mathbf{10} \cdot \lambda^2 - \mathbf{2.5} \cdot \lambda \end{aligned}$$

Chapter 5

Miscellaneous Issues in Computer Graphics for Modeling

The Big Picture

You need to understand key computer graphics techniques used in modeling.

Discover

Understand raster graphics technique.

Understand the hidden line removal technique.

- Polygon filling algorithm
- Visible surface testing
- Z-buffering algorithm
- Polygon clipping
- Z-clipping

In this section, we study several fundamental computer graphics techniques to understand 3D modeling process. We first study the basic raster graphics algorithm which is the key to understanding how a 2D model is represented on the discrete computer screen effectively. Second, we discuss the hidden line removal technique by which a 2D and 3D model will be represented unambiguously on the computer screen. The hidden line removal technique is a function to minimize the ambiguity in representation. Finally, we study a z-clipping technique for multiple 3D model representation.

5.1 Basic Raster Graphics Algorithms for Drawing in 2D

“Raster graphics” is the term for mathematical approximation of ideal primitives such as “line,” “circle,” “ellipse,” “polygon,” etc. Since a primitive in computer graphics has to be displayed on a screen by sets of pixels, an appropriate raster algorithm has to be put in place for satisfactory visualization results. Raster display invokes scan-conversion algorithms and clipping at each time an image is created or modified. The challenge in raster graphics lies in the fact that these algorithms not only must create visually satisfactory images, but also must execute as rapidly as possible. Scan-conversion algorithms usually utilize incremental methods to minimize the number of calculations. In addition for rapid execution, these algorithms employ integer rather than floating point arithmetic. Although there are various advanced floating point raster graphics packages available, we limit our study to integer point raster graphics packages to understand the fundamental raster graphics principle. Therefore, in this chapter, we study the scan-conversion algorithm first, and the clipping algorithm in conjunction with hidden line removal technique later.

5.1.1 Scan Conversion

The scan-conversion algorithm computes the coordinates of the pixels that lie on or near an ideal, infinitely thin straight line imposed on a 2D raster grid (see Fig. 5.1). Again the challenge lies in the realistic representation of an ideal line on the grid-filled screen. Although a discrete approximation of the ideal line may not be perfect, an optimal representation makes it look close to an ideal line.

Fig. 5.1 A scan-converted line showing intensified pixels as *black circles*

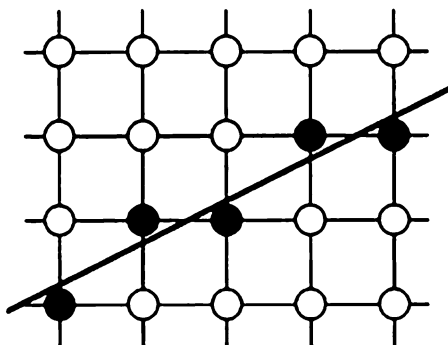
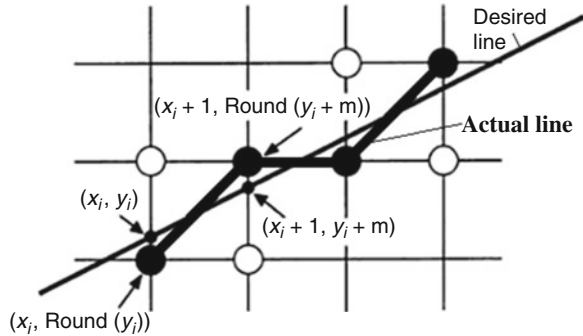


Fig. 5.2 Incremental calculation



5.1.2 The Basic Incremental Algorithm

The simplest strategy for the scan conversion of lines is to compute the slope to increment horizontal line (x) by 1 starting with the leftmost point, to calculate vertical point (y) by the slope and cross of the vertical axis, such as:

$$y_{i+1} = mx_{i+1} + B \tag{5.1}$$

Since the grid is in integer format, the y value has to be rounded, equivalent to Floor ($0.5 + y$), for the closest possible grid point of an ideal line for each given x value. From the computational standpoint, however, this approach is not efficient since each iteration requires a floating point multiplication, addition, and invocation of Floor. Instead, the process can be replaced by an incremental operation such that,

$$y_{i+1} = m(x_i + \Delta x) + B = mx_i + B + m \Delta x = y_i + m \Delta x \tag{5.2}$$

If $\Delta x = 1$ then the above equation becomes,

$$y_{i+1} = y_i + m \tag{5.3}$$

Therefore, we can avoid floating point multiply, but addition and Floor operation only as shown in Fig. 5.2.

If $|m| > 1$, a step in x creates multiple steps in y , so that the line may look disconnected along the way. Thus we must reverse the roles of x and y by assigning a unit step to y and incrementing x by $\Delta x = \Delta y/m$. If we set $\Delta y = 1$ (in order not to be disconnected, and at the same time, to minimize the number of steps along the y axis) then $\Delta x = 1/m$. One thing noticeable is that since the next y position will be found from the rounded value of the current y , the incremental operation introduces cumulative error buildup and eventually a drift away from a true Round of each value of y .

Another and yet one of the most successful incremental operations is the DDA (Digital Differential Analyzer) [3], a mechanical device that solves differential

equations by using numerical methods. It traces out successive (x, y) values by simultaneously incrementing x and y by small steps proportional to the first derivative of x and y . We consider line drawing as a collection of dots moving along the line assuming that a dot moves at a constant speed from a start point (x_s, y_s) to an end point (x_e, y_e) , if we define the time derivative of the displacement along x and y as \dot{x} and \dot{y} respectively, then

$$x(t) = x_s + \int_0^t \dot{x} dt \quad (5.4)$$

$$y(t) = y_s + \int_0^t \dot{y} dt \quad (5.5)$$

For a constant velocity,

$$\dot{x} = \frac{x_e - x_s}{T}$$

$$\dot{y} = \frac{y_e - y_s}{T}$$

where T is the total travel time. Therefore,

$$x(t) = x_s + \int_0^t \frac{x_e - x_s}{T} dt \quad (5.6)$$

$$y(t) = y_s + \int_0^t \frac{y_e - y_s}{T} dt \quad (5.7)$$

If the total interpolation time is divided into N equal time interval, and the time duration, Δt , is very small, then the integral of the constant velocity may be considered as a summation of infinitesimally small discrete velocities. Given that

$$T = N \Delta t \quad \text{and} \quad t = n \Delta t,$$

We have,

$$x(t) = x(n \Delta t) = x_s + \frac{x_e - x_s}{N \Delta t} \cdot n \Delta t = x_s + \frac{x_e - x_s}{N} \cdot n \quad (5.8)$$

$$y(t) = y(n \Delta t) = y_s + \frac{y_e - y_s}{N \Delta t} \cdot n \Delta t = y_s + \frac{y_e - y_s}{N} \cdot n \quad (5.9)$$

for $n = 0, 1, 2, 3, \dots, N$.

After every summation, the position values x and y are increased by a constant amount. For practical applications, it is, however, necessary that the rate of increase for a single calculation time unit Δt is not greater than the output resolution for a

continuous line. Hence, we need to have a condition for N that should not be less than a particular value. The value of N is, in general, rounded up to the next highest factor of ten; in this way, the division in (5.8) and (5.9) may be carried out by simply moving the decimal point to the left by one digit.

One important aspect of using (5.8) and (5.9) is to determine the value of N so that the line is not discontinuous on the output screen. For a given screen resolution, Δr , the value of N has to satisfy the following equation.

$$\max \left\{ \left| \frac{x_e - x_s}{N} \right|, \left| \frac{y_e - y_s}{N} \right| \right\} \leq \Delta r \quad (5.10)$$

For a hardware DDA, the frequency (f_0) of the calculation cycle is derived for the required slide-displacement velocity v_s :

$$f_0 = \frac{N}{T}; \quad v_s = \frac{\sqrt{[(x_e - x_s)^2 + (y_e - y_s)^2]}}{T} \quad (5.11)$$

Sample Problem 5.1

Find the appropriate value of N , interpolation time T , and Δr for the given points, P_s (10,000, -55,000) and P_e (60,000, 15,000), when Δr is 10 μm and slide velocity v_s is to be constant at 0.5 m/min.

Solution

First, by (5.10),

$$\max \left\{ \left| \frac{15,000 + 55,000}{N} \right|, \left| \frac{60,000 - 10,000}{N} \right| \right\} \leq 10 \mu\text{m} \quad (5.12)$$

Therefore, $N_{\min} \left| \frac{70,000}{10} \right| = 7000$. N_{\min} is rounded up to nearest power of ten, i.e.:

$$N = \text{MOD}_{10}\{N_{\min}\} = 10,000$$

The interpolation time T is:

$$T = \frac{L}{v} = \frac{\sqrt{[(x_e - x_s)^2 + (y_e - y_s)^2]}}{0.5 \times 10^{-6}} = 10.323 \text{ s.}$$

Sample Problem 5.2

Generate the first three and last three sequences of x and y for the Problem 5.1.

Solution

Since $N = 10,000$,

$$\frac{x_e - x_s}{N} = \frac{50,000}{10,000} = 5, \quad \frac{y_e - y_s}{N} = \frac{70,000}{10,000} = 7,$$

$$x(t) = 10,000 + 5*1, \quad 10,000 + 5*2, \quad 10,000 + 5*3, \dots, 10,000 + 5*9998,$$

$$10,000 + 5*9999, \quad 10,000 + 5*10,000.$$

$$y(t) = -55,000 + 7*1, \quad -55,000 + 7*2, \quad -55,000 + 7*3, \dots,$$

$$-55,000 + 7*9998, \quad -55,000 + 7*9999, \quad -55,000 + 7*10,000.$$

or

$$x(t) = 10,005, \quad 10,010, \quad 10,015, \dots, 59,000, \quad 59,995, \quad 60,000.$$

$$y(t) = -54,993, \quad -54,986, \quad -54,979, \dots, 14,986, \quad 14,993, \quad 15,000.$$

Notice that the resolutions for both x and y are less than $10 \mu\text{m}$, so that the line will appear to be connected. Ideally the minimum resolution should be $10 \mu\text{m}$, but the nearest power of 10 is justified by integer arithmetic.

Sample Problem 5.3

Find the resolution of x and y axes for the Problem 5.1, if N is kept as the original value of 7000.

Solution

Since $N = 7000$,

$$\frac{x_e - x_s}{N} = \frac{50,000}{7000} = 7.143, \quad \frac{y_e - y_s}{N} = \frac{70,000}{7000} = 10$$

5.1.3 Circular Interpolation Using DDA

DDA method can also be used for circular interpolation. The velocity component of the individual axis must always be generated tangentially to the given circle. If we consider a circle with its center at the origin of the coordinate system, i.e., $x_m, y_m = 0$, then we have,

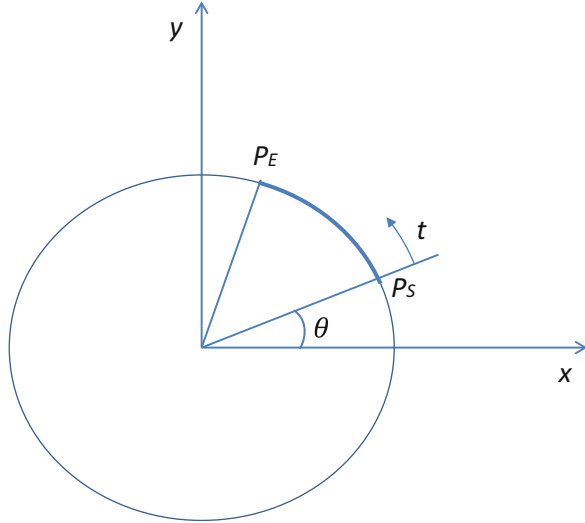
$$x = R \cdot \cos \theta \quad (5.13)$$

$$y = R \cdot \sin \theta \quad (5.14)$$

where R stands for radius (Fig. 5.3). As in linear interpolation, the location of points along the circular path have to be established as functions of time. Given that the required slide-displacement velocity is v_s , then:

$$v_s = \frac{2\pi R}{T} \quad \text{and} \quad 2\pi : \theta = T : t \quad \text{or} \quad \theta = 2\pi \frac{t}{T} \quad (5.15)$$

Fig. 5.3 Circular interpolation using DDA



Therefore,

$$x = R \cdot \cos\left(2\pi\frac{t}{T}\right) \tag{5.16}$$

$$y = R \cdot \sin\left(2\pi\frac{t}{T}\right) \tag{5.17}$$

By differentiating (5.16) and (5.17) with respect to time, we obtain the axial velocity components v_x and v_y as:

$$v_x = \dot{x} = \frac{dx}{dt} = -\frac{2\pi R}{T} \sin\left(2\pi\frac{t}{T}\right) = -\frac{2\pi}{T} y(t) \tag{5.18}$$

$$v_y = \dot{y} = \frac{dy}{dt} = \frac{2\pi R}{T} \cos\left(2\pi\frac{t}{T}\right) = \frac{2\pi}{T} x(t) \tag{5.19}$$

Note that the velocity along the x direction is related to y direction displacement and the velocity along the y direction is related to x direction displacement. Now we can update x and y displacements with the velocity components such as:

$$x(t) = x_s + \int_0^t \dot{x} dt = x_s - \frac{2\pi}{T} \int_0^t y(t) dt \tag{5.20}$$

$$y(t) = y_s + \int_0^t \dot{y} dt = y_s + \frac{2\pi}{T} \int_0^t x(t) dt \tag{5.21}$$

If we replace the integration with summation, then we have:

$$x(t) \approx x(n\Delta t) = x_s - \frac{2\pi}{N} \sum_{i=0}^n y(i\Delta t) = x_s - k \sum_{i=0}^n y(i\Delta t) \quad (5.22)$$

$$y(t) \approx y(n\Delta t) = y_s + \frac{2\pi}{N} \sum_{i=0}^n x(i\Delta t) = y_s + k \sum_{i=0}^n x(i\Delta t) \quad (5.23)$$

for $n = 0, 1, 2, 3, \dots, N$ and $k = \frac{2\pi}{N}$ with the secondary condition of

$$\max \left\{ \left| \frac{x_e - x_s}{N} \right|, \left| \frac{y_e - y_s}{N} \right| \right\} \leq \Delta r \quad (5.24)$$

However, to make (5.22) and (5.23) further simplified, N is usually set to $2\pi \times D$, where D is the diameter of the circle. Therefore, DDA has to compare the current $x(t)$ and $y(t)$ locations to check if both of them reach the destination point. The value of each axis is determined from the previously calculated coordinate position of the other axis. This mutual dependency is sometimes referred to as “loop feedback.”

One of the issues for (5.22) and (5.23) is that it is set for a circle drawn counterclockwise. Therefore, the correct sign has to be determined and multiplied for plotting on the screen so that:

$$x(t) \approx x(n\Delta t) = \text{sgn}(x_s - x_e) \cdot x(n\Delta t) \quad (5.25)$$

$$y(t) \approx y(n\Delta t) = \text{sgn}(y_s - y_e) \cdot y(n\Delta t) \quad (5.26)$$

Sample Problem 5.4 [3]

Find the appropriate value of N , k and calculate $x(t)$, $y(t)$ for the given points, Ps (0, 10) and Pe (10, 0) to plot a circle by DDA interpolation.

Solution

Since $N = 2\pi \times D = 2\pi \times 20 = 125$, $k = 2\pi/N = 0.05$. If we set up an Excel spreadsheet with the following cell formula, we obtain the table shown below.

First, for x_s and y_s ,

$$F2 = B2 - 0.05*(K2)$$

$$J2 = D2 + 0.05*K2$$

Second, for correct sign of x_s and y_s ,

$$H2 = \text{SIGN}(B2 - C2)*F2$$

$$L2 = \text{SIGN}(D2 - E2)*J2$$

Then, for the past value of x_s and y_s ,

$$G3 = F2$$

$$K3 = J2$$

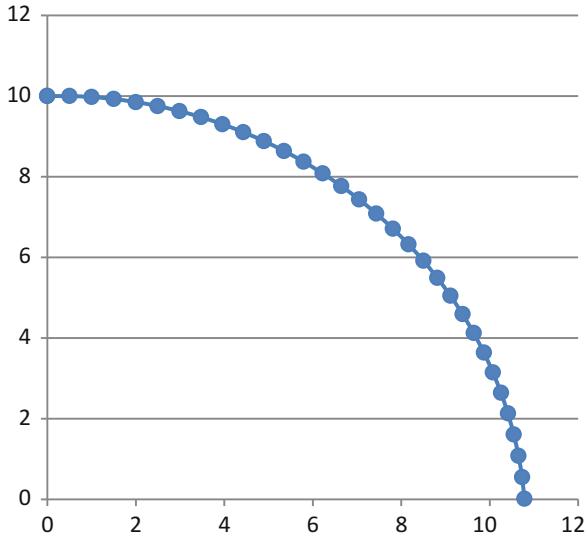
And for sums of x_s and y_s ,

$$I3 = I2 + F3$$

$$M3 = M2 + K3$$

Then, copy the cells in the second row to the rest of the rows.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Step	x_s	x_e	y_s	y_e	$x(t)$	$x(t-1)$	$sgn(x(t))$	$sum(x(t))$	$y(t)$	$y(t-1)$	$sgn(y(t))$	$sum(y(t))$
2	1	0	10	10	0	0	0	0	0	10	0	10	0
3	2	0	10	10	0	0	0	0	0	10	10	10	10
4	3	0	10	10	0	-1	0	1	-1	10	10	10	20
5	4	0	10	10	0	-1	-1	1	-2	10	10	10	30
6	5	0	10	10	0	-2	-1	2	-3	10	10	10	40
7	6	0	10	10	0	-2	-2	2	-5	10	10	10	50
8	7	0	10	10	0	-2	-2	2	-7	10	10	10	60
9	8	0	10	10	0	-3	-2	3	-10	10	10	10	70
10	9	0	10	10	0	-3	-3	3	-14	9	10	9	79
11	10	0	10	10	0	-4	-3	4	-18	9	9	9	89
12	11	0	10	10	0	-4	-4	4	-22	9	9	9	98
13	12	0	10	10	0	-5	-4	5	-27	9	9	9	107
14	13	0	10	10	0	-5	-5	5	-33	9	9	9	116
15	14	0	10	10	0	-6	-5	6	-38	8	9	8	125
16	15	0	10	10	0	-6	-6	6	-45	8	8	8	133
17	16	0	10	10	0	-7	-6	7	-51	8	8	8	141
18	17	0	10	10	0	-7	-7	7	-58	7	8	7	149
19	18	0	10	10	0	-7	-7	7	-66	7	7	7	156
20	19	0	10	10	0	-8	-7	8	-74	7	7	7	163
21	20	0	10	10	0	-8	-8	8	-82	6	7	6	170
22	21	0	10	10	0	-8	-8	8	-90	6	6	6	176
23	22	0	10	10	0	-9	-8	9	-99	5	6	5	182
24	23	0	10	10	0	-9	-9	9	-108	5	5	5	188
25	24	0	10	10	0	-9	-9	9	-118	5	5	5	193
26	25	0	10	10	0	-10	-9	10	-127	4	5	4	197
27	26	0	10	10	0	-10	-10	10	-137	4	4	4	201
28	27	0	10	10	0	-10	-10	10	-147	3	4	3	205
29	28	0	10	10	0	-10	-10	10	-157	3	3	3	208
30	29	0	10	10	0	-10	-10	10	-168	2	3	2	211
31	30	0	10	10	0	-11	-10	11	-178	2	2	2	213
32	31	0	10	10	0	-11	-11	11	-189	1	2	1	215
33	32	0	10	10	0	-11	-11	11	-200	1	1	1	216
34	33	0	10	10	0	-11	-11	11	-210	0	1	0	216



Notice that the desired end point has not reached the final target point exactly. This deviation (enlargement of circle) is caused by a systematic error in the calculation technique, because the integration in (5.20) and (5.21) has been substituted by an addition of finite values in (5.22) and (5.23). The actual value of the x -axis at the desired end point turned out to be 11. If we set the final radius as R_v and original radius as R_0 , then it may be shown that the relative error, $(R_v - R_0)/R_0$, is proportional to the number of the calculation steps and inversely proportional to the divisor N , i.e.:

$$\frac{R_v - R_0}{R_0} \sim \frac{v}{N^2}$$

5.2 Hidden Line Removal

The wireframe modeling, as discussed earlier, is the most fundamental basis of all other modeling techniques. Although useful and simple in logic, it introduces confusion in terms of various modeling aspects and in engineering applications. Thus, as mentioned in Sect. 3.9.1, it is useful only with the hidden line removal technique. Hidden line removal is a computational technique that will remove all of the lines that shouldn't be displayed. In this section, several popular hidden line removal techniques are discussed.

5.2.1 Polygon Filling Algorithm

Two popular polygon filling algorithms introduced in this section can be used for hidden line removal process. The basic idea is to test a dot to see if it is either inside or outside of a polygon. If the dot is inside of polygon than the dot won't be drawn, otherwise it will be drawn. To that end, the polygon under consideration has to be decomposed into a series of dots first and then each dot has to be tested. The first method is the sum of angle method, and the second is the scanning method.

1. Sum of angle method

This method is used to calculate the sum of angles from the dot to all of the vertices of the existing polygon. For instance, in the figure below, two dots are under inside/outside testing with respect to the existing polygon.

Now, for the given angles defined for each vertex, the following statement is true.

$$\text{If } \sum_1^6 \theta = 2\pi \text{ then the dot is inside of the polygon.}$$

$$\text{If } \sum_1^6 \theta = 0 \text{ then the dot is outside of the polygon.}$$

For instance, the sampling dot in Fig. 5.4 produces 2π , therefore, it is inside of the polygon. By such simple test strategy, each dot of new polygon will either be drawn or not drawn. The process provides accurate results, but is slow due to decomposing the polygon into a series of dots and testing each individual dot for inside/outside testing.

2. Scanning method

The scanning method is another important technique for the inclusion test. In this method, the first step of decomposition is the same as the previous method, but

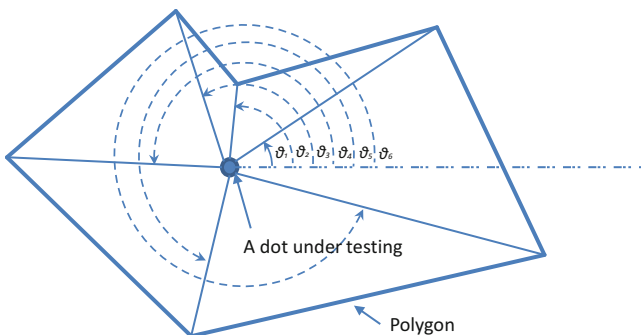


Fig. 5.4 Sum of angle method

Fig. 5.5 Scanning method

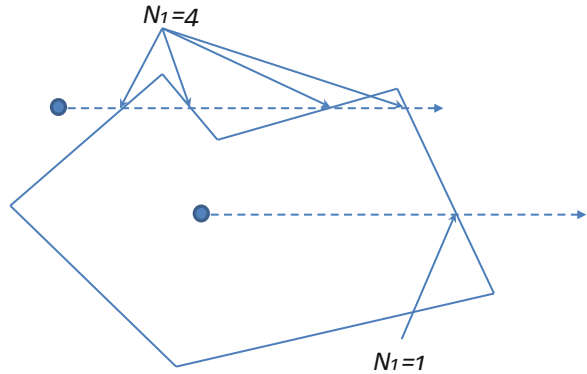
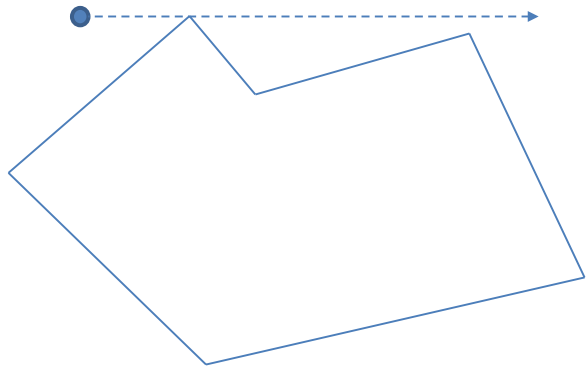


Fig. 5.6 Risky situation of the scanning method



instead uses scanning from the dot along the horizontal line through the polygon, and checks the number of crossings that occurs during the scanning process. As shown in Fig. 5.5, for instance, the number of crossings of the outside dot along the scan line is four and the number of crossings of the inside dot along the scan line is one.

For the given number of crossings along the scan line, the following statement is true.

- *If the number of crossings along the scan line is an odd number, then the dot is inside of the polygon.*
- *If the number of crossings along the scan line is an even number, then the dot is outside of the polygon.*

This method, in general, results in faster test speed compared to other testing methods, but it may cause inaccurate results. For instance, when the scan line is passing a vertex of the polygon as shown below, then the result becomes an odd number, even though the dot is outside of the polygon (Fig. 5.6).

In order to overcome such problems, each dot of the polygon has to be tested multiple times by moving the dot one pixel above and below to make sure the result is consistent. If the result is different, then the largest number of the test result will be considered as the true count of crossings. In addition, to further expedite the test, often recommended is the bounding box test, by which the testing area will be bounded only to the box that encompasses the existing polygon. This will require search for the minimum and maximum coordinate value of the shape along horizontal and vertical directions [2].

5.2.2 Visible Surface Testing

Visible surface testing, also known as Poorman's algorithm, expedites hidden line removal by sorting out faces that are facing backward from the viewer's standpoint. In boundary representation, one of the solid modeling techniques, models are expressed by a collection of facets that defines the normal vector of the outward surface. The general surface equation of the facet is given by the equation below.

$$a \cdot x + b \cdot y + c = 0 \quad (5.27)$$

Then the normal vector, n , can be defined as;

$$\hat{n} = \frac{[a \quad b \quad c]}{\sqrt{a^2 + b^2 + c^2}} \quad (5.28)$$

If we define the vector, v , as the viewer's viewing vector, then the dot product between the surface normal vector and the viewing vector produces a value depending on the angle between two vectors. Therefore, the following test can sort out invisible facets.

If $\vec{n} \cdot \vec{v} > 0$ then do not draw.

If we set the viewing vector, \vec{v} , by the vector, \vec{w} , in Fig. 5.7, such that,

$$\vec{v} = -\vec{w} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix},$$

then the above test for sorting out facets that face backward can be rephrased as;

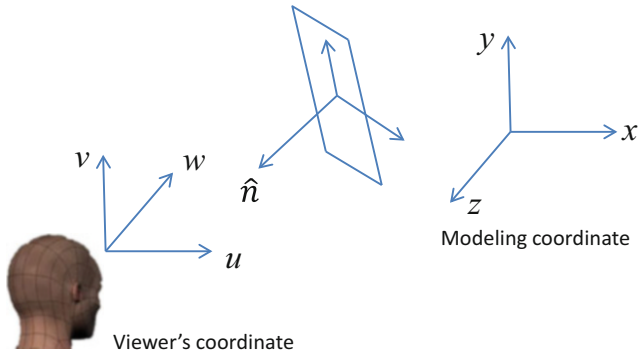


Fig. 5.7 Visible surface testing

$$\text{if } [a \ b \ c] \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -\mathbf{1} \end{bmatrix} > 0 \text{ then do not draw.} \quad (5.29)$$

Sample Problem 5.5 [1]

Find if the facet below has to be skipped by the visible surface testing for the viewer's normal vector given below.

$$\begin{aligned} -5 \cdot x + 2 \cdot y + 1 &= 0 \\ \hat{n} &= [1 \ -1 \ 1] \end{aligned}$$

Solution

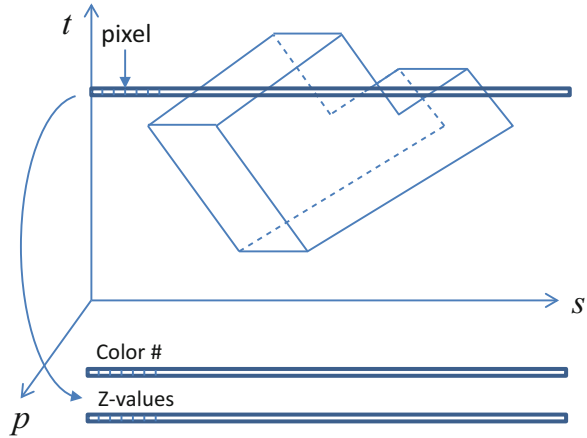
$$\vec{n} \cdot \vec{v} = [-5 \ 2 \ 1] \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} = -5 - 2 + 1 = -6 < 0$$

Therefore, the facet can't be skipped.

5.2.3 Z-Buffering Algorithm

Another efficient hidden line removal method is the z-buffering algorithm. Z-buffering is a technique based on the raster scanning method, by which the depth of each surface is determined and painted pixel by pixel to color only the object that is closest to the viewer. In principle, all of the faces of objects in the scene are identified and compared in terms of depth from the viewer for display. To that end, the Z-buffering algorithm requires a data structure that contains a color code of

Fig. 5.8 Raster scanning in z-buffering technique



each pixel as well as the z-value for the corresponding pixel (see Fig. 5.8). Therefore, the algorithm requires a 2-dimensional data table that stores the color code as well as the z-value of each pixel of the screen. This data structure will be updated as the raster scan continues.

Since it is based on raster scanning, the operation is time consuming, but it is simple and efficient for algorithm’s standpoint. Below is the outline of the z-buffering algorithm.

Z-Buffering Algorithm

1. Fill the color array by background color.
2. Apply coordinate transformation on each model to obtain screen coordinate and depth values, *p*.

$$\begin{bmatrix} s \\ t \\ p \end{bmatrix} = {}^S_G T \cdot \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

u, v, w: vertex of the model in global coordinate.
s, t, p: vertex of the model in screen coordinate.
 ${}^S_G T$: Transformation matrix between global coordinate and screen coordinate.

3. Start the line scanning from the top to bottom, left to right.
4. Select all the faces of objects in the scene where the raster scanning line passes through.
5. Use interpolation for the depth value of each pixel of the surface between edges using the interpolation equation. If *s*₁, *s*₂ are the horizontal value and *p*₁, *p*₂ are the depth values of an edge of the surface under consideration, then the depth value of a pixel on the surface can be found by;

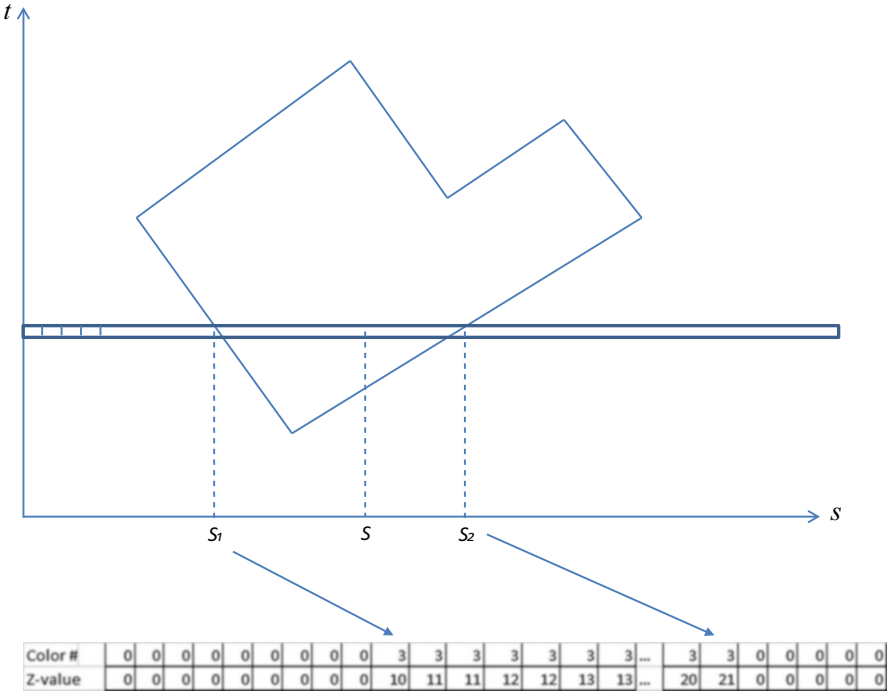


Fig. 5.9 Pixel value replacement

6.
$$p = p_1 + \frac{(p_2 - p_1)(s - s_1)}{s_2 - s_1}$$
7. Compare the obtained depth value of the model under consideration with all other objects' depth value of the surface at the same location.
8. If the current p value is larger (closer to the viewer), then replace the depth value in the table, as well as current pixel value with the corresponding surface's pixel value as shown in Fig. 5.9.
9. Continue steps from 4 to 7 for all of the faces of models in the scene.

During the z-buffering process, surface shading can be applied by using the dot product between the normal vector of the surface and the viewing vector. Based on the dot product of two vectors, the R.G.B value of each pixel is adjusted to provide shading effect.

In order to expedite the z-buffering method, it is often complemented by the bounding box strategy, whereby the scanning area will be minimized by a box that contains the only models in the scene. This will require search for the minimum and maximum coordinate value of the shape along the horizontal and vertical directions based on the screen coordinate.

5.2.4 Polygon Clipping

Polygon clipping is also known as Weiler–Atherton algorithm. The Weiler–Atherton algorithm is capable of clipping a concave polygon with interior holes to the boundaries of another concave polygon, also with interior holes. The polygon to be clipped is called the subject polygon (SP) and the clipping region is called the clip polygon (CP). Depending on the depth of each newly created polygon after the clipping process, hidden lines can be removed by skipping those new polygons behind the clip polygon. The algorithm describes both the SP and the CP by a circular list of vertices ending at the start vertex. The exterior boundaries of the polygons are described counterclockwise, and the interior boundaries or holes are described clockwise so that all of the edges are connected in the order of a linked list data structure. The boundaries of the SP and the CP may or may not intersect. If they intersect, the intersections occur in pairs.

Let us consider an example shown in Fig. 5.10. Due to the relative position of two polygons, they overlap each other. Below is the outline of the polygon clipping process summarized with the example in Fig. 5.10.

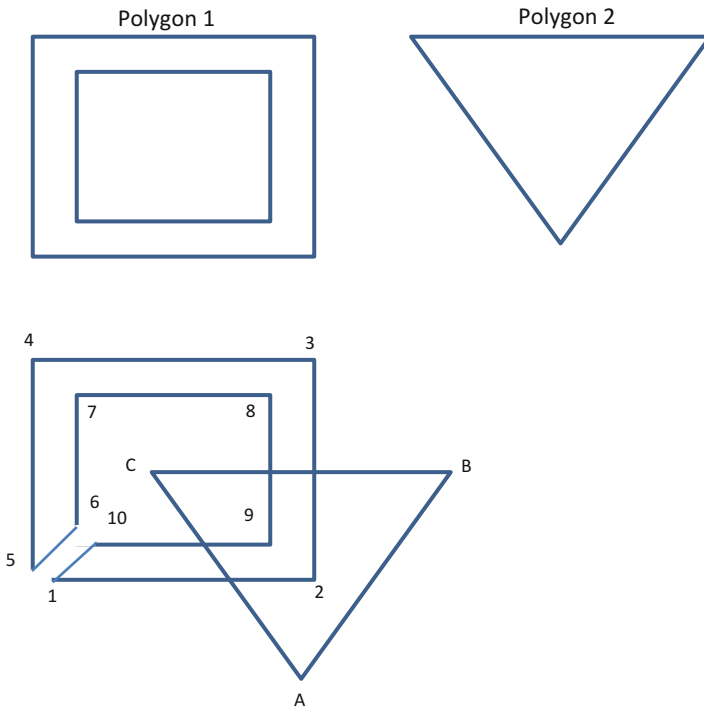


Fig. 5.10 SP (Polygon 1) and CP (Polygon 2) with the edge processing results

Fig. 5.11 Intersection point search along the SP boundary

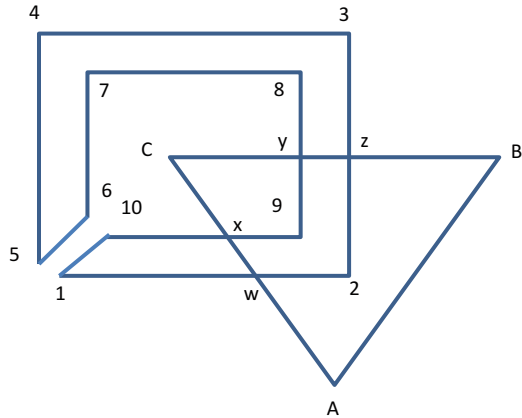
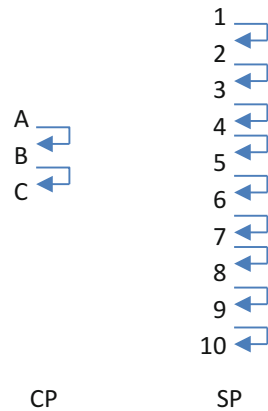


Fig. 5.12 Two linked list data structures of polygons



Step 1: The first step for the hidden line removal by polygon clipping is to form a linked list data structure of all the vertices on both SP and CP as shown in Fig. 5.11. The linked list data for SP and CP are shown in Fig. 5.12.

Step 2: Once the edge ordering process is finished, the next step is to find intersections by the overlapped region between two polygons. The newly found intersection points are marked as w, x, y, z in the example as the exterior boundary of the SP is processed. This process is continued until the starting point is reached.

Step 3: Add each intersection to the SP and CP vertex lists as shown in Fig. 5.13. Note that each intersection points are added in the order marked along the boundary on both SP and CP linked list data structures.

Step 4: Create new polygons by a vertex cross check. This process starts from the top of the linked list in SP in search of a new polygon out of two linked list data. Search is performed in clockwise circulation on the CP polygon (bottom to top) and counterclockwise circulation on the SP polygon. Jump occurs between the data structure when there is a matching vertex. For instance, the first new polygon is

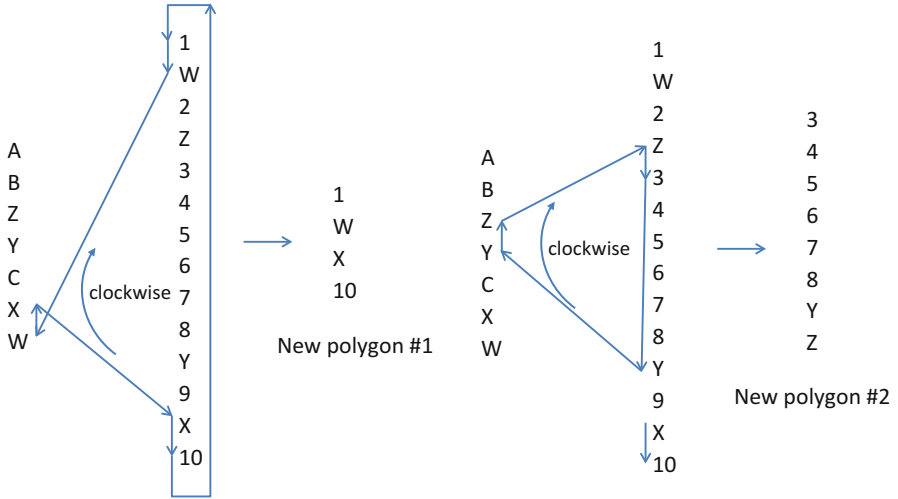
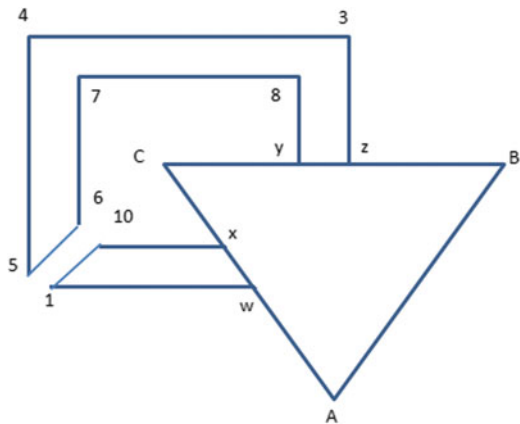


Fig. 5.13 Updated vertex lists for SP and CP

Fig. 5.14 Hidden line removal by polygon clipping



found by following the arrow lines as shown in Fig. 5.13, left. In addition, the second new polygon is found by following the loop as shown in Fig. 5.13, right.

As a result, hidden lines on SP can be removed by the polygon clipping process as shown in Fig. 5.14.

The process introduced from step 1 to step 4 above has to be performed for all of the polygons overlapping each other in the scene. In summary, the overall hidden line removal by the polygon clipping process can be outlined as below:

1. Draw the polygon closest to the viewer.
2. Z-sort all of the polygons
3. Do:

- (1) Find the next polygon (SP).
 - (2) Perform inclusion test between the current polygon and all of the previously drawn polygons. If there is no overlap, go back to step (1).
 - (3) Run the polygon clipping between current polygon and all of the previously drawn polygons.
 - (4) Draw and register new polygons created by clipping process.
4. Until (the final polygon is processed).

5.2.5 Z-Clipping

Z-clipping is another important method for 3D model development in computer graphics. It serves for the hidden line removal or for the clipping feature in CAD software. If we need to remove certain portions of an object interfered by a plane, or by an object, the interfering plane between an object and the clipping plane has to be calculated. To that end, equation of the clipping plane has to be compared with all of the planes of each object. Then, the new coordinates of the portions of an object projected on the clipping plane have to be calculated for display as well. The overall process, in general, is lengthy and mathematically intensive.

Instead, the overall process can be much simpler by using a coordinate transformation from x, y, z coordinate to u, v, w coordinate. The x, y, z coordinate is the coordinate attached on the object or a world coordinate, while u, v, w coordinate is the coordinate attached on the clipping plane. Then we can apply the following coordinate transformation to convert all of the vertices on the object into vertices in u, v, w coordinate.

$$\begin{bmatrix} {}^{uvw}P \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} {}^{uvw}R & {}^{xyz}P_{\text{BORG}} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} {}^{xyz}P \\ \mathbf{1} \end{bmatrix} \quad (5.30)$$

Let's assume that the side view of the clipping plane with a plane of an object is as shown in Fig. 5.15. First of all, we check if a line has vertices on each side of the clipping plane. This can be done by comparing the values of w in v, u, w coordinate. For instance, in Fig. 5.16, line #1 and #3 have two vertices on each side of the clipping plane. In such cases, the new coordinate of P_1 and P_3 can be found by the following parametric equation.

$$P = \lambda \cdot P_1 + (1 - \lambda) \cdot P_2 \quad (5.31)$$

Since we know the value of w' of the clipping plane as well as the value of w of P_1 and P_2 , we can find the value of λ with the above equation. Therefore, we can easily find the value of u' and v' , the new coordinate values of P_1 , on the clipping plane by the following equation.

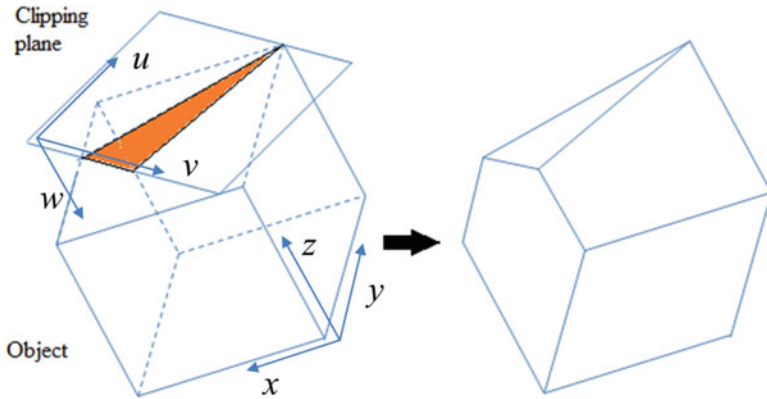
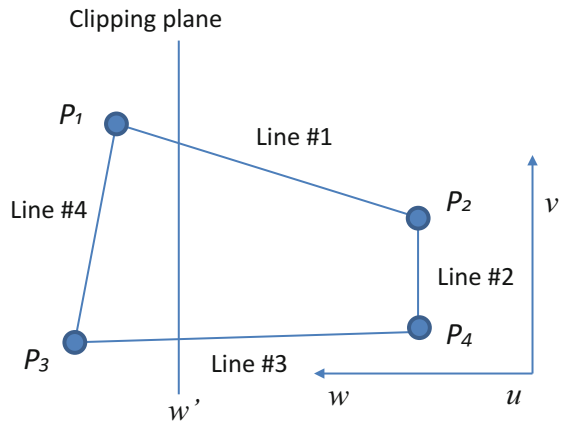


Fig. 5.15 Clipping by a plane

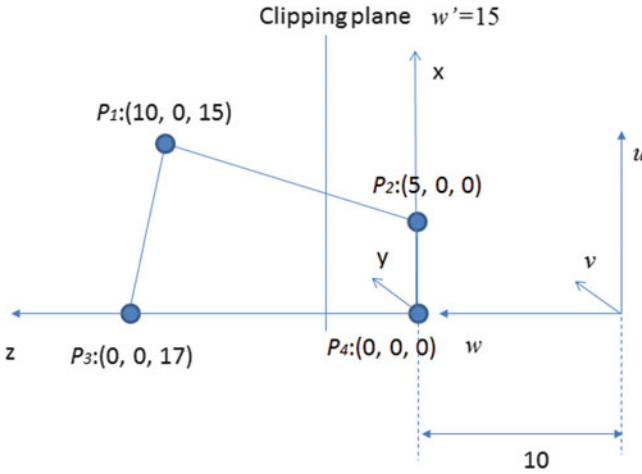
Fig. 5.16 Clipping in u, v, w coordinate



$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \lambda \cdot \begin{bmatrix} u \\ v \\ w \end{bmatrix}_1 + (1 - \lambda) \cdot \begin{bmatrix} u \\ v \\ w \end{bmatrix}_2 \tag{5.32}$$

Sample Problem 5.6

For the given shape and the z-axis coordinate of the clipping plane, obtain the new x, y, z coordinates of each vertex if affected.



Solution

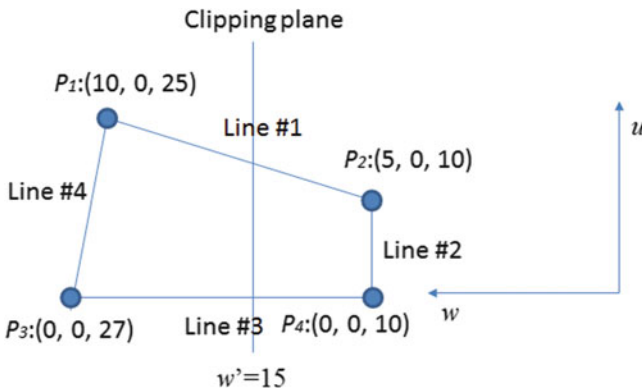
First of all, each vertex of the given shape has to be transformed into the u, v, w coordinate system. The transformation between x, y, z and u, v, w coordinates is as shown below,

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Therefore, we can obtain the new coordinate of each vertex by coordinate transformation.

$$\begin{aligned}
 P_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \\ 15 \\ 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 25 \\ 1 \end{bmatrix} \\
 P_2 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \\ 10 \\ 1 \end{bmatrix} \\
 P_3 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 17 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 27 \\ 1 \end{bmatrix} \\
 P_4 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 10 \\ 1 \end{bmatrix}
 \end{aligned}$$

The transformed shape into u, v, w coordinates will be as shown below.



By comparing the value of w axis, three lines are found affected: *Line #1*, *Line #3* and *Line #4*. Therefore, the new values of P_1 and P_3 have to be obtained. Using (5.31), we can obtain the value of λ for P_1 and P_3 as shown below.

For P_1 ,

$$P_w = \lambda \cdot P_{1w} + (1 - \lambda) \cdot P_{2w}$$

or

$$15 = \lambda \cdot 25 + (1 - \lambda) \cdot 10$$

$$\lambda = 1/3$$

Therefore, the new coordinate of P_1 will be as follows:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \frac{1}{3} \cdot \begin{bmatrix} 10 \\ 0 \\ 25 \end{bmatrix}_1 + \left(1 - \frac{1}{3}\right) \cdot \begin{bmatrix} 5 \\ 0 \\ 10 \end{bmatrix}_2 = \begin{bmatrix} 6.67 \\ 0 \\ 15 \end{bmatrix}$$

For P_2 ,

$$P_w = \lambda \cdot P_{3w} + (1 - \lambda) \cdot P_{4w}$$

or

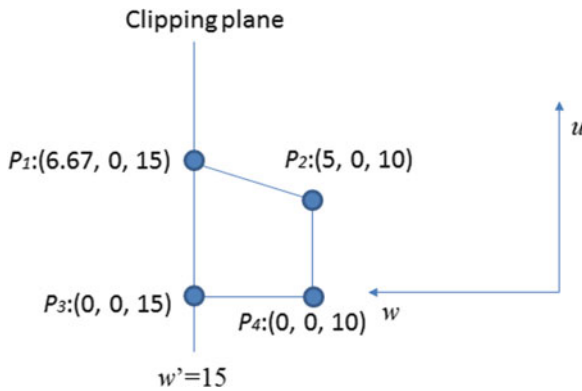
$$15 = \lambda \cdot 27 + (1 - \lambda) \cdot 10$$

$$\lambda = 5/17$$

Therefore, the new coordinate of P_3 will be as follows:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \frac{5}{17} \cdot \begin{bmatrix} 0 \\ 0 \\ 27 \end{bmatrix}_1 + \left(1 - \frac{5}{17}\right) \cdot \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix}_2 = \begin{bmatrix} 0 \\ 0 \\ 15 \end{bmatrix}$$

The given shape will be as drawn below after clipping:



Finally, if we transform the newly obtained shape back to x, y, z coordinate system, then we obtain new coordinates of P_1 and P_3 as below:

$$P'_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6.67 \\ 0 \\ 25 \\ 1 \end{bmatrix} = \begin{bmatrix} 6.67 \\ 0 \\ 15 \\ 1 \end{bmatrix}$$

$$P'_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 25 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 15 \\ 1 \end{bmatrix}$$

Exercise Problem 5.1

Find the appropriate value of N , interpolation time T , and Δt for the given points, Ps (3000, -5000) and Pe (10,000, 10,000), when Δr is 5 μm and slide velocity v_s is to be constant at 1 m/min.

Exercise Problem 5.2

Generate the first three and last three sequences of x and y for the Exercise Problem 5.1.

Exercise Problem 5.3

Find the resolution of x and y axes for the Exercise Problem 5.1, if N is kept as the original.

Exercise Problem 5.4

Find the appropriate value of N , k and calculate $x(t)$, $y(t)$ for the given points, Ps (0, 5) and Pe (5, 0) to plot a circle by DDA interpolation.

Exercise Problem 5.5

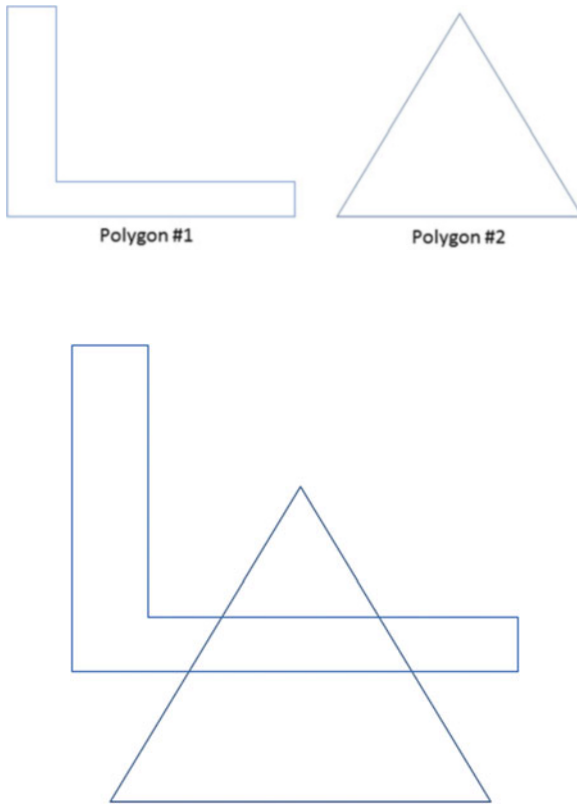
Find if the facet below has to be skipped by the visible surface testing for the viewer's normal vector given below:

$$-15 \cdot x - 7 \cdot y - 1 = 0$$

$$\hat{n} = [-1 \quad -1 \quad 1]$$

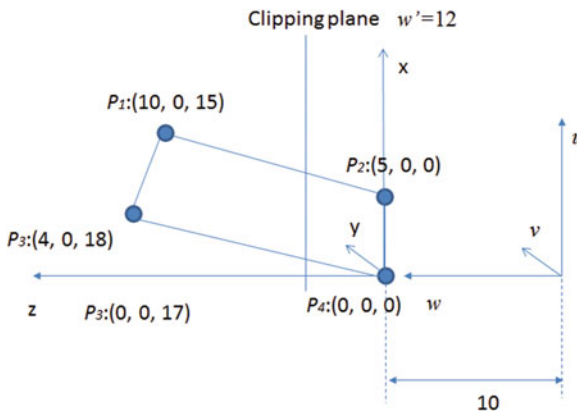
Exercise Problem 5.6

For the given two polygons below, perform polygon clipping. Assume polygon #2 is on top of polygon #1.



Exercise Problem 5.7

For the given shape and the z -axis coordinate of the clipping plane, obtain the new x, y, z coordinates of each vertex if affected.



References

1. Weck M (1984) Handbook of machine tools: automation and controls, vol 3. Wiley, Hoboken
2. Blinn J (1988) Fractional invisibility. IEEE Comput Graph Appl 8(6):77–84
3. Appel A (1967) The notion of quantitative invisibility and the machine rendering of solids. In: Proceedings ACM National Conference. Thompson Books, Washington, pp 387–393

Chapter 6

Rendering Theory

The Big Picture

Discussion Map

You need to understand the basic rendering theory and terminologies for realistic manifestation of 3D models.

Discover

Understand how a computer generated image represents a 3D model realistically

Understand how different colors are represented in computer graphics

Understand the dithering process

Understand shading theory for realistic representation

In computer graphics, rendering is a process of generating a 2D image of a 3D model. The results of such a model is called rendering. A 3D model represented in a 2D screen is a strictly defined geometry by viewpoint, texture, lighting condition, and shading information. Although there are various rendering theories, the general challenges to overcome in producing a 2D image of a 3D model lies in the question as to how to formulate a graphics pipeline along a rendering software and hardware. Phong's illumination model and Gouraud's illumination model are a few examples from other numerous rendering theories. Recently, the BRDF (bidirectional reflectance distribution function), a four-dimensional function that defines how light is reflected at an opaque surface, was introduced in academia. The function takes a negative incoming light direction, and an outgoing direction, which are both defined with respect to the surface normal and returns the ratio of reflected radiance exiting along the irradiance incident on the surface from the incoming light direction. Each direction is itself parameterized by the azimuth angle and the zenith

angle, therefore, the BRDF, as a whole, is 4-dimensional. Some results of the study in reflection phenomena on transparent objects are available as well [1].

No matter what illumination model is used as a graphic pipeline, in order to make a virtual scene looks relatively realistic under certain lighting conditions, the rendering software should solve the rendering equation. Since the rendering algorithm cannot take all of the aspects of natural lighting phenomena into account, it is rather a simplified algorithm based on a certain theory. Since rendering has uses in various areas such as architecture, video games, simulators, movies or TV visual effects, and design visualizations, each employs a different balance of features and techniques. As a product, a wide variety of rendering techniques are available tailored to each application. In this chapter, we investigate fundamental rendering components and several others, most prevailing rendering techniques used in solid modeling.

6.1 Color

What determines the color of light? Color is not a primary physical property like the temperature or pressure of a gas. We can, however, ascertain the color of an “instance” of light by physical measurements which will predict for us the eye’s response to it. The physical property of the light that gives it its color is its “spectrum” or the “plot” of distribution of the power in the light over the range of wavelengths that can affect the eye.

In computer graphics, realistic color representation asks for an exquisite, yet simple color model to mathematically formulate different colors since a computer generates different colors with Red Green Blue (RGB) numbers for each pixel. To name a few, CMYK (Cyan, Magenta, Yellow, and Key (black)), RGB color models such as HSL (Hue, Saturation, Lightness), HSV (Hue, Saturation, Value) are the examples used in computer graphics. CMYK model is popular for printers with four different inks to produce various colors on a paper, while HSL and HSV are popular models for display devices. Please refer to Table 6.1 for definitions of several important color-making attributes. In HSL and HSV models, all of the pure colors can be transformed to white by increasing brightness and to black by decreasing brightness. Therefore, a simple triangular model of the tones of color can be represented as shown in Fig. 6.1. A “pure” color can be saturated to white color by changing the level of “tints,” while the level of “shading” effect can be added with decreasing brightness.

In Physics, a specific wavelength of the observed light and the level of energy density result in a specific color. Most of the color-making attributes can be explained by the energy density versus the wavelength diagram depicted in Fig. 6.2. For instance, Hue is directly relevant to the wavelength, while the purity of a color is due to the ratio between e_1 and e_2 . In addition, the luminance is the area of the energy density. Humans are known to be able to discern 350,000 different colors and 128 Hues (pure colors) with up to 3 nm wavelength difference. Among RGB, human eyes are most sensitive to green whose peak energy density is at around 530 nm. Red and blue have their peak energy density at around 650 nm and 425 nm respectively.

Table 6.1 Color-making attributes

Hue	The attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors: red, yellow, green and blue, or to a combination of two of them
Intensity (radiance)	The total amount of light passing through a particular area
Luminance	The radiance weighted by the effect of each wavelength on a typical human observer measured in candela per square meter (cd/m^2). Often the term luminance is used for the relative luminance
Brightness	The attribute of a visual sensation according to which an area appears to emit more or less light
Lightness (value)	The brightness relative to the brightness of a similarly illuminated white
Colorfulness	The attribute of a visual sensation according to which the perceived color of an area appears to be more or less chromatic
Chroma	The colorfulness relative to the brightness of a similarly illuminated white
Saturation	The colorfulness of a stimulus relative to its own brightness

Fig. 6.1 Tones of color

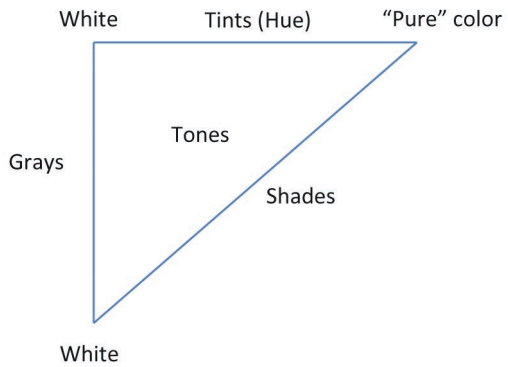
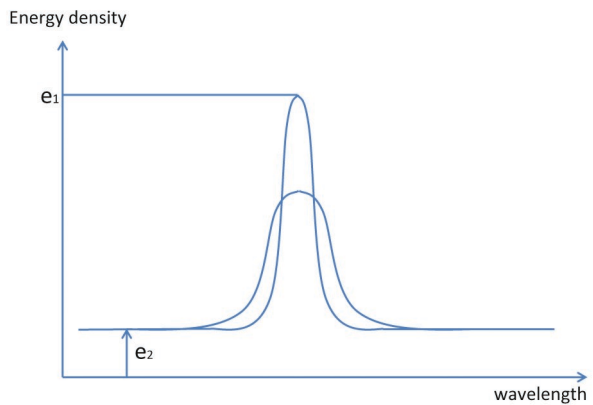


Fig. 6.2 Color attributes in Physics



6.1.1 *How the Eye Determines Color*

It has been determined that (for fairly substantial luminance), the eye observes each tiny element of the image on the retina with three kinds of “cones,” which are “photodetectors.” Each kind has a different spectral response, by which we mean a curve that tells how much “output” the cone delivers from light of a fixed “potency” at each wavelength over the visible range. When an area on the retina is bathed in light with a certain spectrum, in effect, for each of the three kinds of cones:

- The spectrum of the light is multiplied by the spectral response of the cone, meaning that, for each wavelength, the “potency” of the light at that wavelength is multiplied by the value of the spectral response at that wavelength.
- All of these products are added together, giving the output of the cone.

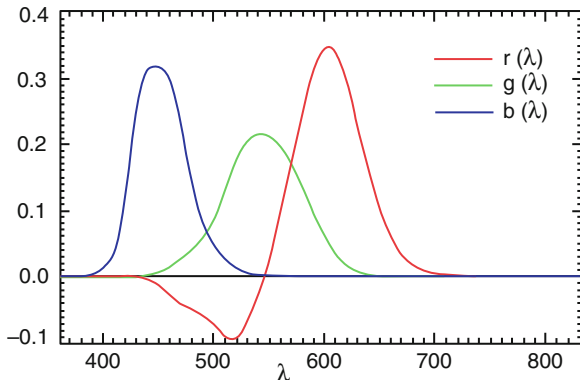
The three types of cone are known as “L,” “M,” and “S,” referring to the fact that the peaks of their spectral responses are at different wavelengths, which we arbitrarily consider to be “long,” “medium,” and “short.” The spectral response curves of the three types of cone are called \bar{l} , \bar{m} , and \bar{s} . The three parameters, noted S, M, and L, can be indicated using a 3-dimension space, called LMS color space, which is one of many color spaces which have been devised to help quantify human color vision. In the next section, we study how we could formulate eye’s perception with those three identified cones mathematically.

6.1.2 *The Color Matching Experiments*

We cannot directly determine the three response functions, \bar{l} , \bar{m} , and \bar{s} , of the human eye, mainly because we cannot put a “meter” on the outputs of the cones. In order to estimate this output, a series of experiments are performed. The basic tests use a “tristimulus” concept with three actual primary light sources. A small screen was arranged so that one half was illuminated by light of a “test color,” while the other half was illuminated by light composed of adjustable amounts of the three primaries. Now for a given series of various test colors changing from 650 to 425 nm, observers are asked to change the intensity of spectrum density that corresponds to each cone. Another word, with a particular test light (a spectrum with single wavelength) in place, the user was asked to adjust the amounts of the three primaries (700, 546.1, and 435.8 nm) until there was an exact visual match at the boundary between the two halves of the screen.

In some cases, it turned out that a negative amount of a certain primary was needed for a match. How could that be done? In such a case, that primary was added to the “test light” on the reference half of the screen, which was equivalent to a negative contribution of that primary to the “matching light.” Multiple runs were made with multiple observers, and a vast mass of data was accumulated. Analysis of this data produced a consolidated model of light matching with its known color

Fig. 6.3 CIE RGB color matching functions



by amounts of the three primary lights. This was presented in the form of three curves, one for each primary, which showed the amount of that primary needed in the “mix” to make light whose color was the same as that of “monochromatic” test light of a certain wavelength, as a function of that wavelength. These were called the CIE (International Commission on Illumination) “color matching functions,” designated r , g , and b . They are seen in Fig. 6.3.

A disadvantage of this characterization of human color response is that the “red” matching function has negative values in part of the wavelength range. In fact, it can be demonstrated that, for a “color space” with “real” primaries (that is, ones we can actually generate, and see), at least one of the curves must have negative portions. There was concern that the need to keep track of both positive and negative values could increase the risk of a misstep. Thus, the workers decided that there was a need to develop another color space whose underlying matching functions were “nowhere negative” [2]. This did not call for another round of zillions of human observations. We can take a description of a color in terms of values of the three coordinates of one color space and convert it into values of the three coordinates of another color space—coordinates that revolve around a different set of primaries (assuming that both color spaces are of the additive genre and thus both have primaries). And then we can, by another mathematical manipulation, determine what the corresponding set of matching functions would be for the “new” color space by the following linear transformation.

$$X = m_{11}R + m_{12}G + m_{13}B \quad (6.1)$$

$$Y = m_{21}R + m_{22}G + m_{23}B \quad (6.2)$$

$$Z = m_{31}R + m_{32}G + m_{33}B \quad (6.3)$$

These three new primaries are “everywhere nonnegative” but they are imaginary values. These variables become the foundation of the CIE XYZ color scheme in the field of colorimetry (Fig. 6.4).

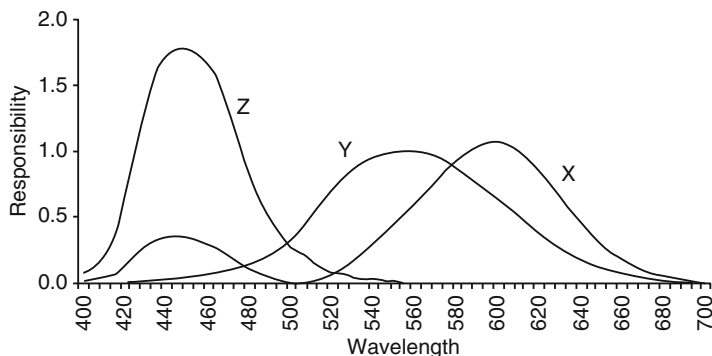


Fig. 6.4 CIE XYZ color matching functions

Based on the fact that we are free to choose from among many new color spaces in our quest for one with “everywhere nonnegative” matching functions, we realize that for any one we choose, there will be a certain set of the nine values, m_{11} – m_{13} , that make up the transformation matrix to this space from the CIE RGB space. Further, we have a lot of flexibility in that process. One of the well-known facts in colorimetry is that the sum of R, G, and B becomes white visible spectrum, which is relevant to the luminance of the color. Therefore, if we define the luminance (L) of a color we have represented by its coordinates in an additive color space of R, G, and B, we can write an equation of luminance naturally as a function of R, G, and B values such that:

$$L = k_R R + k_G G + k_B B \quad (6.4)$$

Now, since we have a freedom to choose the transformation coefficients in (6.1)–(6.3), if we choose m_{21} , m_{22} , and m_{23} , such that;

$$m_{21} = k_R, \quad m_{22} = k_G, \quad m_{23} = k_B,$$

then L will always be equal to Y . This color space is another color space different from CIE XYZ, and is named as CIE xyY, which will be discussed in more detail in the later section. One problem of the CIE XYZ color space was that, since it is an imaginary space, which is transformed from RGB space not to cause any negative value, the three “stimuli” (its primaries) cannot really be emitted, and so in fact cannot be used as actual stimuli to the eye!

6.1.3 A Cousin Color Space

The CIE XYZ color space is an additive color space: as we have seen, it revolves around the concept of describing a color by stating the amounts of three primaries that would be combined to make light of that color. (Can we make visible light,

with a real color, by mixing together “imaginary” primaries? In the laboratory, no. On paper, mathematically, yes. We’ll see how a little later.) Another important genre of color space is the luminance-chrominance color space genre. These relate more directly to the human outlook on color than the additive spaces. In a luminance-chromaticity color space, one coordinate tells us the luminance of the color, while the other two, together, tell us the chromaticity (and there can be several ways this can be organized). To allow the benefits of this in our scientific work, the wonks devised a luminance-chromaticity color space based on the CIE XYZ color space. Its chromaticity coordinates are known as x and y . They are defined this way:

$$x = \frac{X}{X + Y + Z} \quad (6.5)$$

$$y = \frac{Y}{X + Y + Z} \quad (6.6)$$

Two new variables of x and y in above equations and the luminance variable of Y become the foundation of the CIE xyY color space. Since the value of X and Y are normalized by the sum of X , Y , and Z , we often call this space the cousin color space as well. The above equations allow us to obtain the corresponding values of x and y for each wavelength of the visible spectrum in Fig. 6.4 by which the values of X , Y , and Z can be determined.

6.1.4 The CIE x - y Chromaticity Diagram

Since the coordinates, x and y , tell us the chromaticity of a color, we can use a chart with x and y as its axes to plot points that indicate chromaticity, usually called the CIE x - y chromaticity diagram.

If we vary the spectrum of the color that consists of only a single wavelength, we can obtain corresponding values of x and y , which generate a locus of “horseshoe” in x , y coordinate diagram (see Fig. 6.5). The dotted line at the bottom completes the region “enclosed” by the horseshoe. The chromaticities along it are not “spectral” (there is no light with only a single wavelength component that exhibits a color with such a chromaticity). They are called “nonspectral purples.” All colors of visible light have chromaticities represented by points inside the region bounded by the horseshoe (and the locus of nonspectral purples).

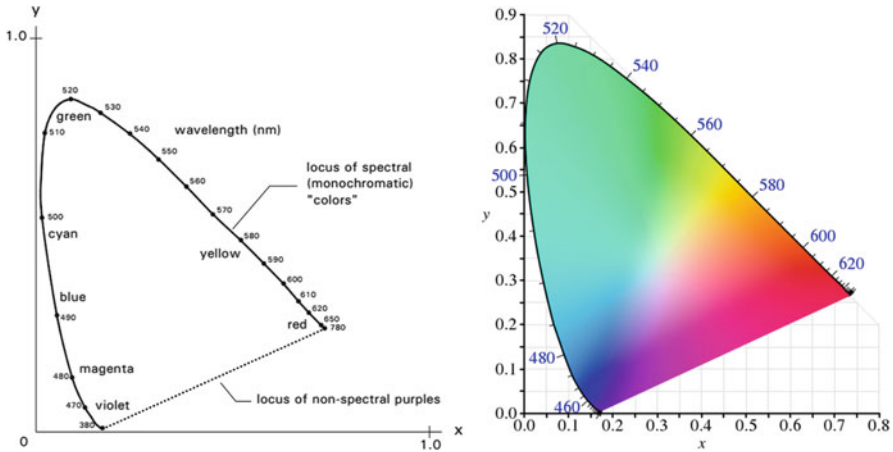


Fig. 6.5 CIE x - y chromaticity diagram (left) and CIE xyY color space diagram (right)

6.2 Color Display

While most display products today can display a staggering number of colors (more than 16 million), human can only recognize 350,000 colors. In order to be able to display true color, PC needs to use RGB color input, 8 bit each, thus for a total of 24 bits. Twenty-four bits of RGB can generate 16,777,216 ($2^8 \times 2^8 \times 2^8$) colors (see Fig. 6.6). In order to minimize the waste of memory, most of the leading LCD monitors use 6-bit approach for RGB, thus producing 262,144 ($2^6 \times 2^6 \times 2^6$) different colors. The 6-bit approach is less costly in implementation with no waste in memory, but it can't cover the total of 350,000 colors that human eyes can differentiate. In order to overcome the limit of the 6-bit approach, the FRC (frame rate control) technique is used.

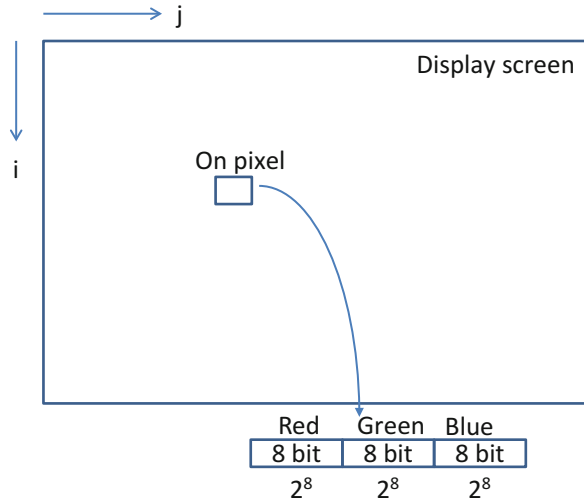
The FRC, a temporal dithering method, is a display color control system to increase the number of apparent colors by manipulating the frame rate, rather than increasing the number of bit for each pixel. As a result, three different types of monitors are available in the market. While the first one can represent the color scheme in the true sense of the term, the price tag will be much higher than the other two types.

Maximum number of colors and color reproduction method in LCD monitors

	Maximum display colors	LCD panel specs	Tone reproduction	Approximate price
1	16.77 million	8-bit	Excellent	High
2	16.77 million	6-bit + FRC	Good	Medium
3	16.19 million/16.2 million	6-bit + FRC	Fair	Low

In order to increase the number of apparent colors, the FRC technique takes advantage of afterimage effects in the human eye. For example, switching rapidly between white and red will create what the human eye perceives as a pink color.

Fig. 6.6 Display screen composition for true color display



As a result, if we apply FRC to each RGB color and change the display interval between each of the LCD panels original colors with a 4-bit FRC to generate three simulated colors between each pair of individual colors, the total number of different colors becomes 189 $((6 \text{ bit} - 1) \times 3)$ different shadings. The zero value is excluded since FRC does not cause any change on zero (blackout) value. Since 189 different shades are added on to the existing colors, 253 $(6 \text{ bit} + 189)$ different colors for each RGB value can be achieved. This leads to the total number of colors of 16,194,277 $(253 \times 253 \times 253)$ colors. The difference in picture quality between 8 bits and 6 bits plus the FRC is often not apparent on visual inspection. Under real-life conditions, factors other than the panel, such as the quality of ICs for image control, can also significantly affect picture quality.

Another approach is to use a color lookup table by which all 350,000 colors are regenerated by the 6-bit input for more realistic color regeneration (see Fig. 6.7). In this approach, 6-bit color input will be mapped into a lookup table already prepared in the monitor that may skip some of the colors producible by 8-bit approach. Therefore, the color lookup table only contains about 2 % of the 8-bit approach display, which will result in significant savings on the memory use.

6.3 Dithering

Dithering is a technique to produce different shading effects by either spatial or temporal afterimage effects. The FRC is a temporal dithering effect to increase the number of shading effects while keeping the screen resolution intact. The term, "dithering" is more often used for spatial dithering effect by combining multiple pixels to a virtual pixel. Dithering is often used to create an illusion of "color depth"

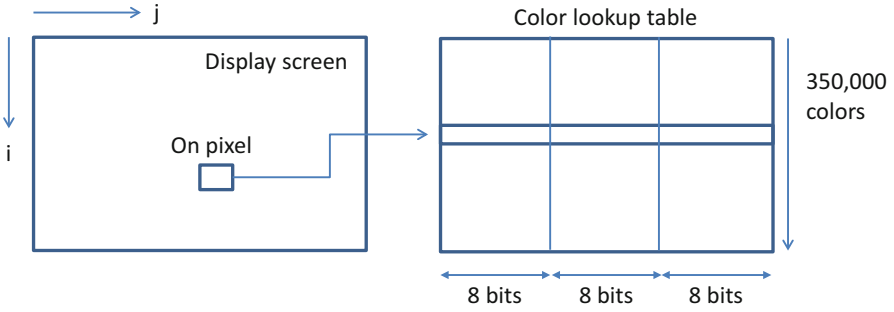


Fig. 6.7 Color lookup table approach

in images with a limited color palette. It is a technique also known as color quantization. The technique is similar to edge smoothing for image sharpening to minimize stepwise appearance. In a dithered image, colors that are not available in the palette are approximated by a diffusion of colored pixels from within the available palette. The human eye perceives the diffusion as a mixture of the colors within it. Dithered images, particularly those with relatively few colors, can often be distinguished by a characteristic graininess or speckled appearance.

By its nature, dithering introduces patterns into an image. The theory is based on the fact that the image will be viewed from such a distance that the pattern is not discernible to the human eye. Error diffusion techniques were some of the first methods to generate blue-noise (azure noise) dithering patterns. In computer graphics, the term “blue noise” is sometimes used more loosely as any noise with minimal low frequency components and no concentrated spikes in energy. This can be good noise for dithering. Retinal cells are arranged in a blue-noise-like pattern which yields good visual resolution. However, other techniques such as ordered dithering can also generate blue-noise dithering without the tendency to degenerate into areas with artifacts. For instance, if four adjacent pixels are considered to be one pixel, then there are four dithering possibilities achievable as shown in Fig. 6.8.

Dithering is not only used for increasing the number of colors or shading effect, but is also used for smoothing of a shape sloped from the horizontal lines. For the consistent result of dithering effect, ordered dithering is used in computer graphics. For instance, if four adjacent pixels are regarded as one pixel, the order of dithering is fixed as;

$$D^{(2)} = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} \tag{6.7}$$

Therefore, following the order of dithering in the above equation, the patterns below can be achieved with the number of pixels that need to be filled as shown below.

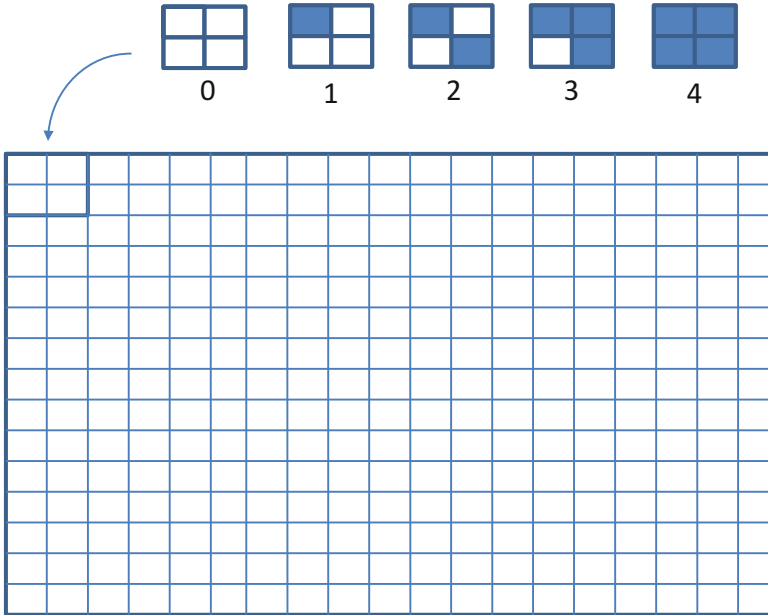
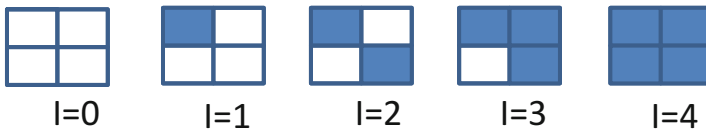


Fig. 6.8 Concept of dithering



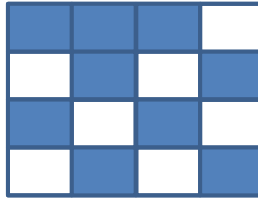
For a consistent result, the general rule of the $D^{(2N)}$ is developed based on (6.7) such that;

$$D^{(2N)} = \begin{bmatrix} 4D^{(N)} + 0 & 4D^{(N)} + 2 \\ 4D^{(N)} + 3 & 4D^{(N)} + 1 \end{bmatrix} \tag{6.8}$$

For instance, if N is equal to 2, then $D^{(4)}$ becomes:

$$\begin{aligned}
 D^{(4)} &= \begin{bmatrix} 4D^{(2)} + 0 & 4D^{(2)} + 2 \\ 4D^{(2)} + 3 & 4D^{(2)} + 1 \end{bmatrix} = \begin{bmatrix} 4 \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} + 0 & 4 \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} + 2 \\ 4 \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} + 3 & 4 \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} + 1 \end{bmatrix} \\
 &= \begin{bmatrix} \begin{bmatrix} 0 & 8 \\ 12 & 4 \end{bmatrix} & \begin{bmatrix} 0+2 & 8+2 \\ 12+2 & 4+2 \end{bmatrix} \\ \begin{bmatrix} 0+3 & 8+3 \\ 12+3 & 4+3 \end{bmatrix} & \begin{bmatrix} 0+1 & 8+1 \\ 12+1 & 4+1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix}
 \end{aligned}$$

Now, for the final matrix of the dithering order for $N=2$, if $I=9$, then the dithering effect of the 4×4 pixel will become:



6.4 Light Illumination Models

Light illumination or shading refers to the process of altering the color of an object/surface/polygon in a 3D scene, based on the surface angle with respect to the incident light and the distance from a light source to create a photorealistic effect. Shading alters the colors of faces in a 3D model based on the angle of the surface to a light source or light sources. Usually, upon rendering a scene a number of different lighting techniques will be used to make the rendering look more realistic. Different types of light sources are used to give different effects.

An ambient light source represents a fixed-intensity and a fixed-color light source that affects all objects in the scene equally. Upon rendering, all objects in the scene are brightened with the specified intensity and color following certain shading rules. This type of light source is mainly used to provide the scene with a basic view of the different objects in it. This is the simplest type of lighting to implement and model how light can be scattered or reflected many times producing a uniform effect. Ambient lighting can be combined with ambient occlusion to represent how exposed each point of the scene is, affecting the amount of ambient light it can reflect. This produces diffuse, nondirectional lighting throughout the scene, casting no clear shadows, but with enclosed and sheltered areas darkened. The result is usually visually similar to an overcast day.

There are two different commonly used shading techniques: flat shading and interpolation. Flat shading is a lighting technique used in 3D computer graphics to shade each polygon of an object based on the angle between the polygon's surface normal and the direction of the light source, their respective colors, and the intensity of the light source. It is usually used for high speed rendering where more advanced shading techniques are too computationally expensive. As a result of flat shading, all of the polygon's vertices are colored with one color, allowing differentiation between adjacent polygons. Specular highlights are rendered poorly with flat shading: If there happens to be a large specular component at the representative vertex, then brightness is drawn uniformly over the entire face. If a specular

highlight doesn't fall on the representative point, it is entirely missed. Consequently, the specular reflection component is usually not included in flat shading computation.

In contrast to flat shading, smooth shading changes the color from pixel to pixel. It assumes that the surfaces are curved and uses interpolation techniques to calculate the values of pixels between the vertices of the polygons. There are various interpolation models of light illumination such as Lambert, Gouraud, Minnaert, Oren-Nayar, Cook-Torrance, and Phong models. Most common smooth shading techniques include the Gouraud shading and the Phong shading. The Hue-Intensity-Saturation (HIS) based fusion approach is used for both approaches.

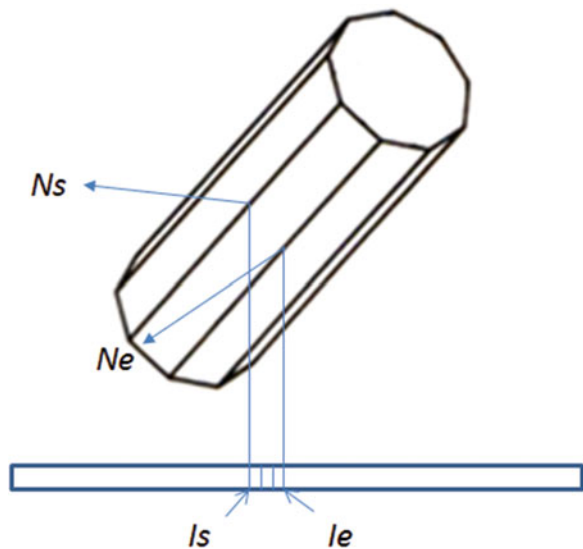
6.4.1 Gouraud Shading

In Gouraud shading, the interpolation parameter is the intensity. It first determines the normal at each polygon vertex. Then it applies an illumination model to each vertex to calculate the vertex intensity. Finally, the interpolation of the vertex intensities is applied using bilinear interpolation over the surface polygon (see Fig. 6.9).

$$I = \lambda \cdot I_e + (1 - \lambda) \cdot I_s \quad (6.9)$$

The above equation is the bilinear interpolation equation between two intensities at each vertex. I_s stands for the start intensity, while I_e stands for the ending intensity respectively. The main advantage of the Gouraud modeling is that

Fig. 6.9 Interpolation for fast rendering (Gouraud shading)



polygons, more complex than triangles, can also have different colors specified for each vertex. In these instances, the underlying logic for shading can become more intricate. However, the smoothness introduced by Gouraud shading may not prevent the appearance of the shading differences between adjacent polygons. In addition, Gouraud shading is more CPU intensive and can become a problem when rendering environments in realtime with many polygons compared to the flat shading. T-Junctions, vertices that belong to more than two facets, can sometimes result in visual anomalies. In general, T-Junctions should be avoided in Gouraud shading [8].

6.4.2 Phong Shading

Phong shading, is similar to Gouraud shading except that the normal vectors are interpolated by the bilinear interpolation. As a result, the specular highlights are computed much more precisely than in the Gouraud shading model (see Fig. 6.10). It first computes a normal vector for each vertex of the polygon. From bilinear interpolation, it computes a normal vector for each pixel (this must be renormalized each time). Now from the new normal vector at each pixel, pixel intensity will be calculated. Phong shading will paint pixels to shade corresponding to its new intensity value. The following equation is for the normal vector interpolation:

$$\hat{N} = \lambda \cdot N_e + (1 - \lambda) \cdot N_s \quad (6.10)$$

N_s stands for the start normal, while N_e stands for the ending normal respectively.

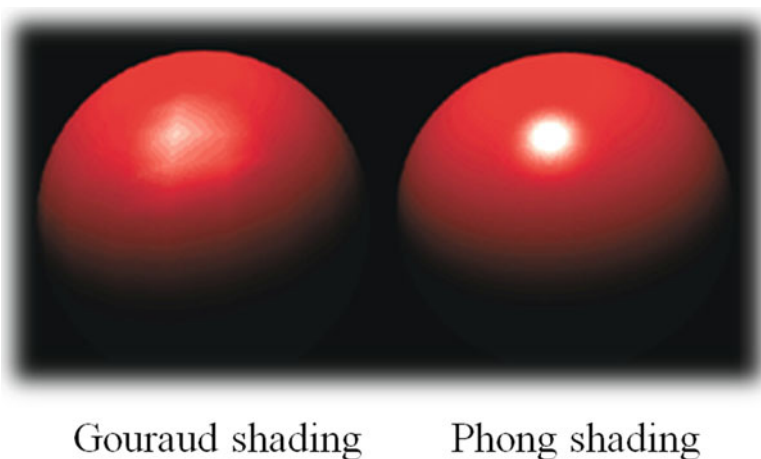


Fig. 6.10 Comparison between Gouraud shading and Phong shading

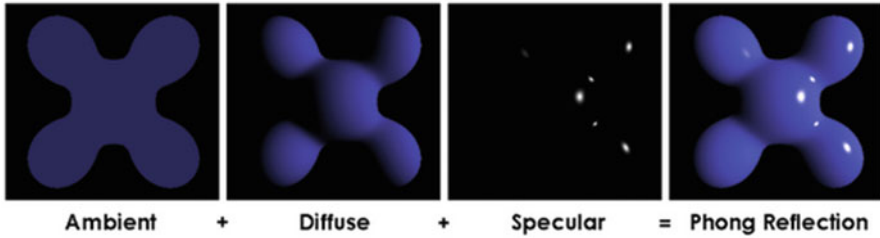
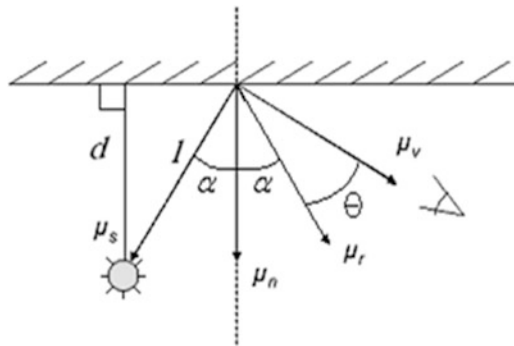


Fig. 6.11 Phong's illumination model

Fig. 6.12 Diagram for Phong's illumination model



Phong's shading model is the most official term in computer graphics for 3D rendering process. It is an empirical model of local illumination with an assumption of a light located at a certain place in the scene. According to the Phong's illumination model, diffusivity and specularity of the objects' surface plays an important role in a 3D object expression (see Fig. 6.11).

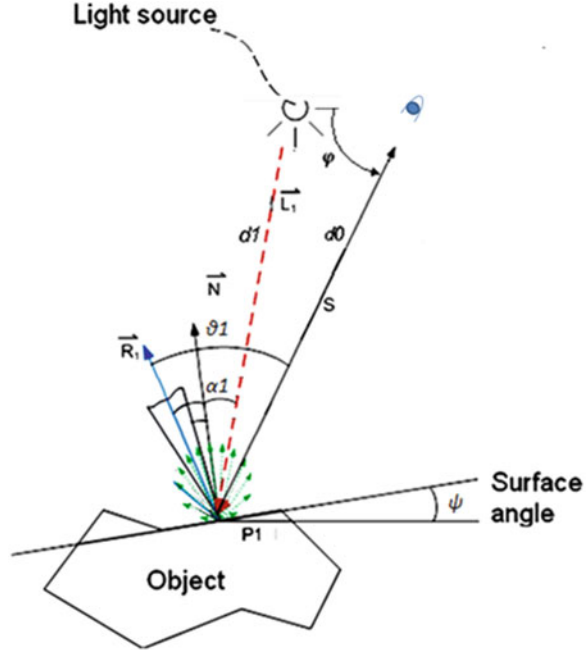
Phong's illumination model and the photometry theory [3] propose that light intensity, I , from the observer's standpoint will be given by:

$$I = C_o \left(\vec{\mu}_s \cdot \vec{\mu}_n \right) + C_1 \left(\vec{\mu}_r \cdot \vec{\mu}_v \right)^n \tag{6.11}$$

The variables C_o and C_1 are two coefficients (diffusivity and specularity) that express the reflectivity properties of the surface being sensed, and n is a power that models a specular reflected light for each material. Vectors μ_s , μ_n , μ_r , and μ_v are the light source, surface normal, reflected, and viewing vector, respectively (see Fig. 6.12).

In Fig. 6.12, θ stands for the angle between the reflected light and viewing angle. Typically, the light radiated from a light source is specified by directivity by which the distribution of radiant intensity around the light source is defined. Suppose the light model with respect to a 3D object is outlined as shown in Fig. 6.13.

Fig. 6.13 Light model



The vector N is the normal to the object's surface at the point of interest. P_1 and d_1 stand for the distance vector from the light source to the point (L_1), α_1 is the angle between the vector L_1 and the normal vector N . That said, (6.9) then becomes:

$$I = K_{\text{ambient}}I_{\text{ambient}} + \frac{I_{\text{point}}}{d^2} [K_{\text{diffuse}}(\vec{N} \cdot \vec{L}_1) + K_{\text{specular}}(\vec{S} \cdot \vec{R}_1)^n] \quad (6.12)$$

The more practical form of the equation above is the one below obtained by replacing the square term of the distance to the sum of the distance and a calibration constant, k .

$$I = K_{\text{ambient}}I_{\text{ambient}} + \frac{I_{\text{point}}}{d+k} [K_{\text{diffuse}}(\vec{N} \cdot \vec{L}_1) + K_{\text{specular}}(\vec{S} \cdot \vec{R}_1)^n] \quad (6.13)$$

The specular portion of the intensity is important, but significant only for a material with high reflectability. In the equation above, the diffuse term, in general, is not dependent on the angle of the incident light. The Phong's model becomes the Lambertian model if the specular term is ignored. Figure 6.14 depicts the Lambertian light model, where the mild change on the reflection intensity will be observed depending on the direction of the observer's eye.

The specularity term, however, will be under influence of the observer's angle significantly as well as the incidence angle of the light source. Since the energy will be concentrated more on the direction of the vector, R , the highest reflectance

Fig. 6.14 Lambertian light model

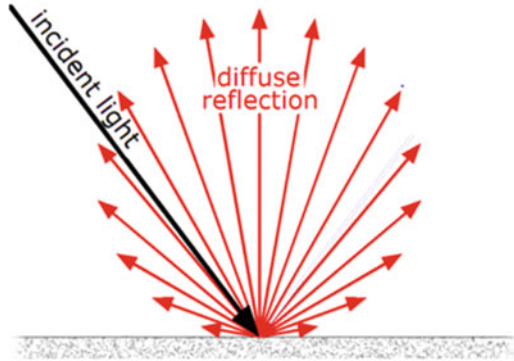


Fig. 6.15 Specularity model

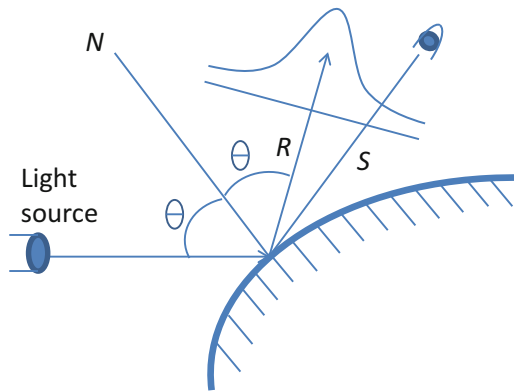
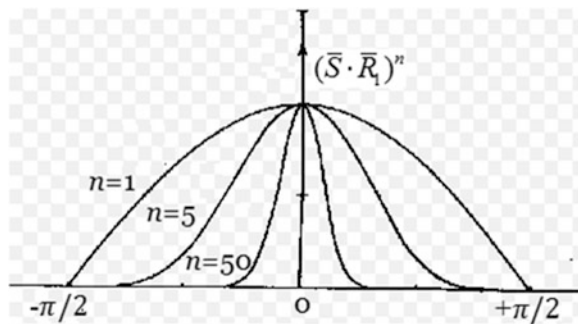


Fig. 6.16 Intensity variation



intensity will be observed if the observer’s eye is in line with the vector, R (see Fig. 6.15). The spatial distribution of the specularity concentration changes by the value, n , the parameter adjustable depending upon the texture of the object. For instance, a metallic surface may be best represented by the n value of 50, while the n value for a wooden surface may be good enough with 1. Figure 6.16 depicts intensity variation depending on the n value.

6.4.3 Other Approaches

Both Gouraud shading and Phong shading can be implemented using bilinear interpolation. Bishop and Weimer [4] proposed to use the Taylor series expansion of the resulting expression from applying an illumination model and bilinear interpolation of the normal vectors. Hence, second degree polynomial interpolation was used. This type of biquadratic interpolation was further elaborated by Barrera et al. [5], where a second-order polynomial was used to interpolate the diffusion effect of the Phong's reflection model and another second-order polynomial for the specular light.

Spherical Linear Interpolation (Slerp) was used by Kuij and Blake [6] for computing both the normal over the polygon as well as the vector in the direction to the light source. A similar approach was proposed by Hast [7], which uses Quaternion interpolation of the normals with the advantage that the normal will always have a unit length and the computationally heavy normalization is avoided.

Important Lesson (Shading Models)

- Gouraud Shading
 - Interpolate intensity values
 - Fast result
- Phong Shading
 - Interpolate normal vectors
 - Realistic result

6.5 Rendering for Shading by Shadow

When there is an object in between the light source and the target subject, there must be a shadow on the target subject by the object placed in the middle. In such cases, the illumination model needs to be modified so that the shadow would appear on the subject. In general, the shadow model is mathematically expressed as the function below.

$$I_1 = K_{\text{ambient}}I_{\text{ambient}} + K_T \cdot \frac{I_{\text{point}}}{d^2} [K_{\text{diffuse}} (\bar{N} \cdot \bar{L}) + K_{\text{specular}} (\bar{S} \cdot \bar{R})^n], \quad (6.14)$$

where K_T is the parameter that stands for the level of transparency of the object in between the light source and the target subject. However, the above equation will be

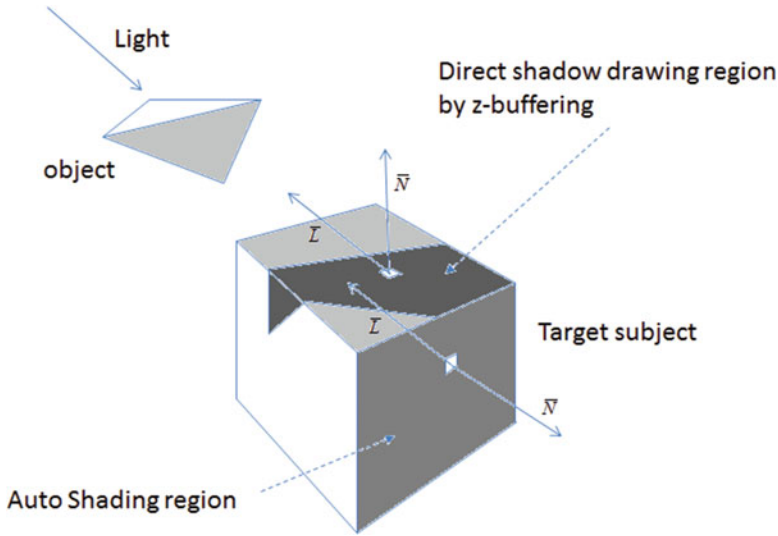


Fig. 6.17 Rendering for shading by shadow

applicable only when the dot product of two vectors, \bar{N} and \bar{L} , is negative, meaning that the angle between two vectors is larger than 90° . Since the angle is larger than 90° , the target surface does not receive the light from the light source directly, thus no shadow should be drawn on the surface. If the surface is flat, then the entire surface is considered to be an auto-shading region. In this case, the K_T becomes equal to one (see Fig. 6.17).

If the dot product of two vectors, \bar{N} and \bar{L} , is positive (the angle is less than 90°), which means that the surface is exposed directly to the light source, then the shape of the object in the middle has to be drawn directly on the surface of the target subject. This can be done by using the z-buffering technique, which is discussed in the previous chapter. The z-direction, however, will be the same as the lighting direction, not the viewing direction. Instead of removing the area covered by the object, the identified area by the z-buffering technique has to be painted by (6.14) (see Fig. 6.17).

The complete shading algorithm will be composed of following steps.

1. Identify the shadowed area by the z-buffering method
2. Apply (6.14) on the identified shadow area with the proper $K_T (<1)$ value
3. Apply (6.14) on the rest of the area with the K_T equal to 1

Important Lesson (Rendering Technique for Shading by Shadow)

- If $\bar{N} \cdot \bar{L} > 0$, then apply shading algorithm.
- If $\bar{N} \cdot \bar{L} < 0$, then apply (6.14) uniformly with the K_T equal to 1

References

1. Hertzmann A, Seitz SM (2005) Example-based photometric stereo: shape reconstruction with general, varying BRDFs. *IEEE Trans Pattern Anal Mach Intell* 27(8):1254–1264
2. Kerr DA (2010) The CIE XYZ and xyY color spaces. Issue 1
3. Phong BT (1975) Illumination for computer generated pictures. *Commun ACM* 18(6):311–317
4. Bishop G, Weimer DM (1986) Fast Phong shading. *Comput Graph (SIGGRAPH)* 20(4):103–106
5. Barrera T, Hast A, Bengtsson E (2006) Fast near Phong-quality software shading. In: *WSCG'06*, pp 109–116
6. Kuijk AAM, Blake EH (1989) Faster Phong shading via angular interpolation. *Comput Graph Forum* 8(4):315–324
7. Hast A (2005) Shading by quaternion interpolation. In: *WSCG'05*, pp 53–56
8. Gouraud H (1971) Continuous shading of curved surfaces. *IEEE Trans Comput C-20* (6):623–629

Chapter 7

Rapid Prototyping

The Big Picture

You need to understand terminologies and various 3D rapid prototyping technologies as well as fundamental physics of the layered manufacturing.

Discover

Understand terminologies.

Understand various applications of the rapid prototyping technology.

Understand various rapid prototyping processes.

Understand data structure.

Understand physics of the rapid prototyping.

While solid modeling and computer graphics offer tremendous opportunities for design verification and validation for design and manufacturing communities, nothing is considered to be better than a tangible solid model as a means for bridging the gap between all the communities in the manufacturing chain. Rapid prototyping is born of necessity to facilitate such needs for design verification and validation. Historically, the roots of rapid prototyping technology can be traced back to practices in topography and photosculpture. Within topography, Blather (1892) suggested a layered method for making a mold for raised relief paper topographical maps. The process involved cutting the contour lines on a series of plates which were then stacked layer by layer. Matsubara (1974) of Mitsubishi proposed a topographical process with a photo-hardening photopolymer resin to form thin layers stacked to make a casting mold.

Photosculpture was a nineteenth-century technique to create exact three-dimensional replicas of objects [7]. Most famously, Francois Willeme (1860) placed 24 cameras in a circular array and simultaneously photographed an object.

The silhouette of each photograph was then used to carve a replica. Morioka (1935, 1944) developed a hybrid photo sculpture and topographic process using structured light to photographically create contour lines of an object. The lines could then be developed into sheets and cut and stacked, or projected onto stock material for carving. The Munz Process (1956) reproduced a three-dimensional image of an object by selectively exposing, layer by layer, a photo emulsion on a lowering piston. After fixing, a solid transparent cylinder contains an image of the object [2].

The technologies referred to as solid freeform fabrication (SFF) are what we recognize today as Rapid Prototyping, 3D Printing, or Additive Manufacturing. Swainson (1977) and Schwerzel (1984) worked on polymerization of a photosensitive polymer at the intersection of two computer-controlled laser beams. Ciraud (1972) considered magnetostatic or electrostatic deposition with electron beam, laser, or plasma for sintered surface cladding. These were all proposed, but it is unknown until working machines were built. Hideo Kodama of Nagoya Municipal Industrial Research Institute was the first to publish an account of a solid model fabricated using a photopolymer rapid prototyping system (1981) [3]. Even at that early date, the technology was seen as having a place in manufacturing practice. A low-resolution, low-strength output had value in design verification, mold making, production jigs, and other areas. Outputs have steadily advanced toward higher specification uses [4].

7.1 Definition

The traditional manufacturing process requires multiple steps from design to inspection including manufacturing cell/material handling setup, process planning, production planning, quality control, inspection, etc. However, *rapid prototyping* is a group of techniques used to quickly fabricate a scale model of a physical part or assembly using three-dimensional computer-aided design (CAD) data [3, 5]. More concisely, it is a process of building a prototype in one step. Construction of the part or assembly is usually done using 3D printing or “additive layer manufacturing” technology [6]. The first method for rapid prototyping became available in the late 1980s and was used to produce models and prototype parts. Today, they are used for a wide range of applications [7] and are used to manufacture production-quality parts in relatively small numbers if desired without the typical unfavorable short-run economics.

Rapid Prototyping & Manufacturing (RP&M) becomes more popular in today’s industry that goes beyond the scope of the rapid prototyping. Exceeding the scope of prototype model creation, RP&M expands the possibility of the layered manufacturing into the next level, where parts for real-world engineering applications are fabricated. Titanium powder-based 3D printing technology is reported recently with many successful stories. For example, a 3D-printed bike has been fabricated with the Titanium powder [8] (see Fig. 7.1). Another exiting application

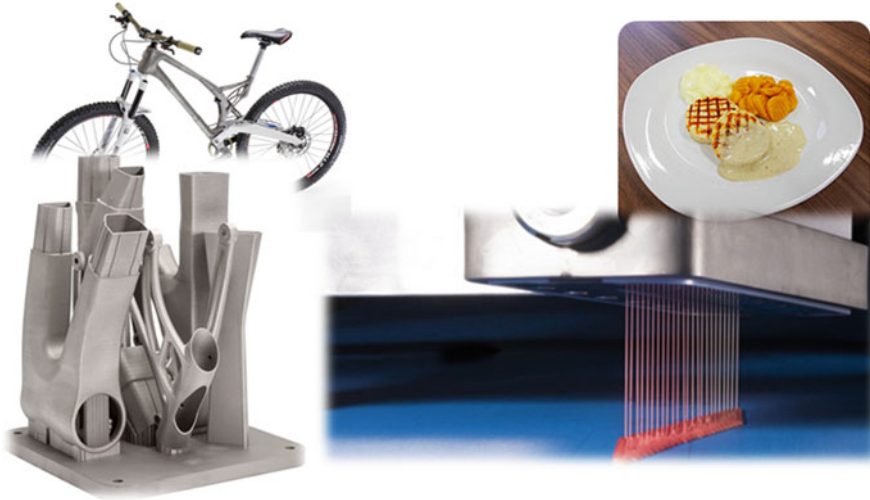


Fig. 7.1 Innovative 3D RP technology (*Left: 3D Titanium Bike, Right: 3D Food Engineering*)

Fig. 7.2 Knee joint by rapid prototyping [27]



area of the 3D printing is food industry. Softer texture of the artificial food can be fabricated in a 3D food printer (Fig. 7.1).

By 3D printing the food, not only will the 3D food printer be able to supply printed food to the elders with swallowing issues, but they may be able to compensate for dietary specifications as well [9]. As mentioned earlier, the application areas of the 3D printing technology are wide open for many years to come. Recently, it has been reported that RP&M technology is used in a biomedical application, especially for those areas where customized design is required. For instance, an RP-machined knee joint is produced to replace the injured knee for a patient (see Fig. 7.2). With a 3-D digitizer capturing external data about a patient's injured knee, a computer software can analyze the center of rotation in the patient's



Fig. 7.3 Contour crafting [28]

knee. Understanding and modeling the motion of the center of rotation in the analysis, the patient will be beneficial in obtaining a customized orthopedic device.

Another exciting area of application is 3D-manufactured building construction, also known as “Contour Crafting.” Contour Crafting (CC) is a layered fabrication technology developed by Dr. Behrokh Khoshnevis of the University of Southern California. Contour Crafting technology has great potential for automating the construction of whole structures as well as sub-components. Using this process, a single house or a colony of houses, each with possibly a different design, may be automatically constructed in a single run, embedded with all the conduits for electrical, plumbing, and air-conditioning (see Fig. 7.3). The potential applications of this technology are far reaching including but not limited to applications in emergency, low-income, and commercial housing.

Definition

Rapid prototyping is a group of techniques used to quickly fabricate a scale model of a physical part or assembly using three-dimensional computer-aided design (CAD) data. More concisely, it is a process of building a prototype in one step via layered manufacturing process.

7.2 Applications

In this section, we review the traditional application areas of the 3D printing technology in an effort to understand the trend of the 3D printing technology evolution. Ever since the concept of direct manufacturing has been coined in

early 1970s, various attempts are made to generate physical objects directly from geometric data without traditional tools. Layered manufacturing, 3D printing, desktop manufacturing, or solid freeform manufacturing are different names of 3D printing technology. RP or RP&M process includes the following three simple steps:

1. Form the cross-sections of the object
2. Lay the cross-sections layer by layer
3. Combine the layers

Among various advantages of the rapid prototyping, following is the list of most assorted advantages that draws attention to the technical innovation:

1. Converting design features to manufacturing features is unnecessary (no need for process planning)
2. Defining different setups or complex sequences of material handling is unnecessary
3. Considering clamping, jigs, or fixtures is unnecessary
4. Designing and manufacturing molds and dies are unnecessary (tool-less process)

The process planning stage is the first step in the traditional manufacturing process. Process planning is an activity of developing a manufacturing plan to convert the product design to a physical entity. It involves determining the most appropriate sequence of processing and assembly steps. All of these planning steps can be avoided in a 3D rapid prototyping process. Second, manufacturing usually involves a sequence of activities performed at different locations in the plant. Therefore, the work must be transported, stored, and tracked as it moves through the plant by material handling devices, which can also be avoidable in the rapid prototyping process. Third, jig and fixture design for each stage of manufacturing is very challenging in traditional manufacturing processes. No jig and fixture design is required in rapid prototyping process. Finally, one of the most common traditional manufacturing processes is casting either by sand or die. Mold and die design occupies large portions of the traditional manufacturing process, which is totally avoidable in rapid prototyping process, thus the name “tool-less process” is applicable.

There are many disciplines and industries willing to take advantage of new, cost-saving, fast methods of producing component parts. RP&M has become the “best practice” and the acceptable approach to “one-off” parts. Progressive companies must look past the prototyping stereotypes and develop manufacturing strategies utilizing additive manufacturing equipment, processes, and materials for high volume production. The pie-chart (Fig. 7.4) below indicates several of those industries now taking advantage of the RP&M technology and the approximate percentage of use.

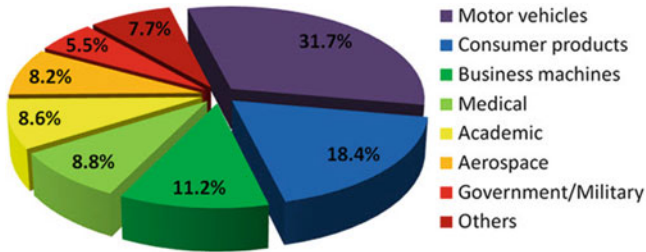


Fig. 7.4 Percent use of RP&M in various industry [29]

Overall, the rapid prototyping technology has main areas of use as listed below.

• Visual aids for engineering:	16.5 %
• Functional models:	16.1 %
• Fit and assembly:	15.6 %
• Patterns for prototype tooling:	13.4 %
• Patterns for cast metal:	9.2 %

Over seventy percent (70 %) of the total uses are given by the five categories above. This in no way negates or lessens the importance of the other uses, but obviously, visual aids, functional models, and models to prove form, fit, and function top the list. In this book, however, we will limit our study on three main areas of applications identified as below:

1. Prototypes for design evaluation
2. Prototypes for function verification
3. Models for further manufacturing processes

7.2.1 Prototypes for Design Evaluation

As mentioned earlier, nothing is better than a tangible model as a communication tool for the involved groups in the manufacturing chain, especially between a design team and a manufacturing team. Traditionally, a model as a communication tool has been the major justification for the investment in RP&M technology. Evaluating a prototype allows the production costs to be assessed and finalized. Every stage of manufacturing can be scrutinized for potential costs. If the client has set financial limits/restrictions, then alterations to the design or manufacturing processes may be inevitable. One unique example of using the PR&M technology is to make a model with a sectional view. Instead of making a whole model and cut a section to show the internal structure, it is now possible to build a model revealing the sectional view in one step with RP&M (see Fig. 7.5).

Traditionally, 3D prototyping has been done with wood-based material, which, in general, is very costly. Once a prototype model is made, the observation limit is

Fig. 7.5 Model with a sectional view [30]



Fig. 7.6 Wood prototyping [31]



constrained only by the resolution of human eyes, which are known to be second to none compared to representation on any monitor currently available. In Fig. 7.6, the traditional wood-based prototyping process has been captured for a Nissan NTX model. While wood is a good choice, it could be naturally decaying or distorted by humidity. In addition, it causes dust and health problems unless the process is tightly controlled. The new trend is now to use a 3D RP&M machine, which becomes more affordable and provides higher quality with longer period of conservation. In addition, recently added coloring function or hybrid approach with multiple materials can produce more realistic prototype models.

7.2.2 Prototypes for Function Verification

The use of RP&M, as the technology ever advanced from its inception, has been expanded to the functional verification of a designed product. Ever since it is discovered that the multiple components are produced as a body of a complete assembly, the idea of mobility analysis for kinematic models is coined and realized with 3D RP technology. This ability of making a complete assembly with no need of part assembly after the production of each part opened up such a vast application possibility with merits in production time saving and assembly cost saving. The example shown in Fig. 7.7 is a product that is produced in one step without further need of assembly of

Fig. 7.7 Kinematic chain produced as a complete assembly [32]

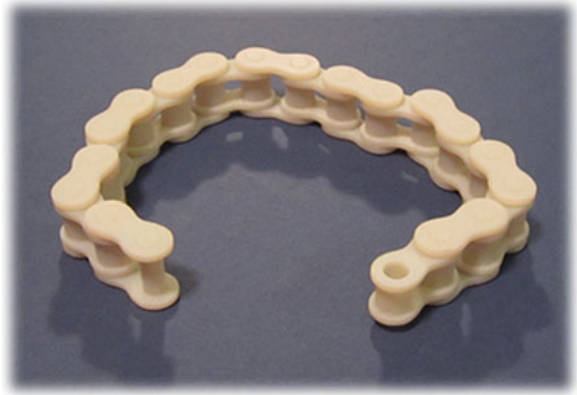


Fig. 7.8 Rapid prototyped airplane model under wind tunnel test [33]



each part. The complete chain immediately demonstrates mobility so that the kinematic aspect of the product can be checked right after the production.

In addition to the mobility test, other functional testing such as aero/hydro dynamics, stress/strain, or fatigue strength can be performed and tested on a RP-produced part to some extent. For instance, in Fig. 7.8, a rapid prototyped airplane is under wind tunnel test. In [10], an attempt is made to drastically reduce the product design cycle by providing “real experimental data” for correlation with FEA data using an RP model. To that end, homogeneity of the RP material has to be achieved to ensure a valid transfer of results from model to actual parts. As a result, RP&M enables significant savings in production cost and lead time for manufacturing, especially for a complex part.

7.2.3 *Models for Further Manufacturing Processes*

In addition to the design evaluation and function verification, RP&M parts are used for models for further manufacturing processes. For instance, an RP&M-produced support pattern can be further used for fabrication processes. In [11], a rapid prototyping technique for mold tooling is demonstrated by laser-cut laminated sheets of H13 steel, bolted or brazed together. The down-selection of materials, bonding methods and machining methods, the effect of conformal cooling channels on process efficiency, and the evaluation of a number of test tools developed for the industrial partners are also discussed. It is claimed that the cost and time advantages (up to 50 % and 54 %, respectively) of the tooling route are reported compared to traditional fabrication methods. In this section, we investigate several important uses of RP&M techniques reported for industrial manufacturing processes.

7.2.3.1 **Rapid Plaster Molding**

Rapid plaster molding (RPM) and rapid die casting are emerging technologies branched out from the RP&M technology. Plaster mold casting is a quick and relatively inexpensive way of producing aluminum and zinc castings. SLA or other rapid models can be used as master models to develop mold tools. The plaster process differs from the use of Quickcast (in which models are destroyed making each casting) in that plaster mold casting creates a foundry tool from the SLA model. Once the rubber tooling has been generated, it can be used to produce up to a thousand aluminum or zinc castings before tooling maintenance is required. Rubber foundry tooling may be cost-justified over Quickcast even if only a few Quickcast models are required, depending on part complexity. For smaller quantities (less than five pieces), plaster molds can be made directly from many types of rapid prototyping models for geometries with or without side pulls. This approach is referred to as loose pattern molding.

The plaster process has some invaluable properties, among them is the ability to produce complex, thin-walled castings with excellent surface finishes. That makes it the ideal choice for reproducing the fine and complex details often found in SLA models. Casting lead times vary from a few days to several weeks depending on part complexity. It can cast a wide range of sizes, but is the most applicable for parts that fit within a 2–24-in. cube range. Quantities ranging from two or three pieces to several thousand pieces can be produced as functional die cast prototypes. Therefore, plaster serves as a bridge process while awaiting delivery of production tooling and, in many cases, as a production process where quantity requirements do not justify the expense of hard tooling.

The first step in RPM is to create a master model. Usually, it is SLA but in many cases traditional pattern-making models are still used. The parting lines are then established and negative room temperature vulcanization (RTV) molds are developed from the model. A silicon rubber positive is made from each of the negative

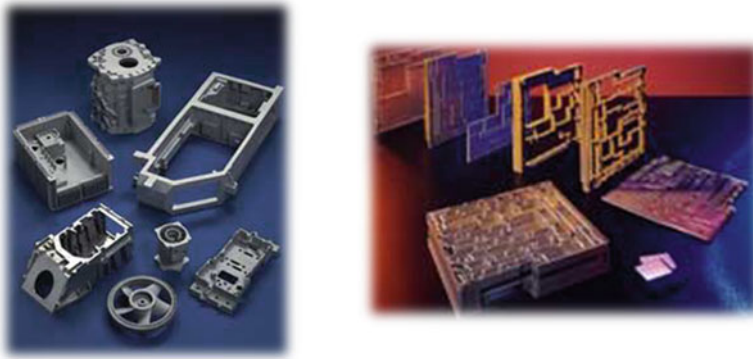


Fig. 7.9 Plaster molding (*left*) and rapid die molding (*right*) [34]

molds. Gating runner systems are added as required. Next, a liquid plaster slurry is poured over the silicon rubber patterns. Once the plaster molds have set, they are removed from the rubber patterns and baked to remove moisture.

Subsequently, molten aluminum is poured into the assembled plaster mold. Once the metal has solidified, the plaster mold is destroyed so that the part can be taken out. The reusable rubber tooling can make hundreds of molds. After the casting has passed initial inspection, gates and flash are removed and the part is now ready for secondary operations such as machining, assembly, chemical, or paint finishes. Again, for time savings in creating a few pieces, plaster molds can be made directly from the SLA pattern once parting lines have been created. Figure 7.9 shows the examples of plaster-molded aluminum parts.

Rapid Plaster Molding

SLA Master Pattern (positive) → RTV mold (negative) → RTV mold (positive) → Plaster mold (negative) → Final casting (positive)

7.2.3.2 Rapid Die Casting

Rapid die casting from cast tooling is an approach to die cast prototyping and short runs that has been around for several decades, but the advent of CAD and SLA modeling is driving a renewed interest into its application. Starting with a SLA model, H-13 steel dies are cast to a net shape in a fraction of the time required for steel cutting tool. Additional geometry can be added via secondary machining. The key benefit is that prototype parts are processed as a die casting and therefore physical and thermal transfer properties will be identical to the production part. Additionally, this process can provide considerable cost advantage over RPM for quantities over a thousand. Since the components are run in a die cast press, large numbers of parts can be manufactured in a short amount of time. This highly

accurate process is capable of casting fine detail with excellent surfaces. This approach is not recommended for use with thin, tall standing part detail or with cast-in water lines. Typical lead times range from five to eight weeks depending on part size and complexity.

In the rapid die casting process, an initial pattern is generated via SLA or CNC with shrinkage factors scaled in. The advantage of using the rapid prototyped part is that it avoids all the necessary assembly features such as bolt, nut, or rivet for complex assembly [12]. Once a RP model is created, parting lines are developed and a soft negative is created with ceramic material. The burnout process in the furnace will completely remove the plastic RP model embedded in the ceramic die, thus leaving the negative shape of the model in the ceramic die. The molten steel is then poured into the die. Cavity detail can be finished if required via CNC or EDM. The inserts are squared and fitted into a standard mold base where ejection pins are added. Gates and overflows are machined in and the cast die is ready to run in a standard die cast press. Parts can be run in any die cast alloy and finishing requires only a trim and any secondary machining that may be necessary. Figure 7.9 (right) demonstrates several dies created by RP&M technology for rapid die molding process.

RP Applications

1. Prototypes for design evaluation
2. Prototypes for function verification
3. Models for further manufacturing processes

7.3 Rapid Prototyping Processes

General RP process is composed of building multiple thin layers in sequence from the bottom to the top layer. Although various RP processes have been developed, the general layer-by-layer accumulation process (layered manufacturing) is in common.

7.3.1 General Principle

Regardless of the particular RP technology, there is a general principle for layered manufacturing. Below is a summary of general principles for layered manufacturing processes.

1. Polymerization of suitable resins by laser, other light beams, or lamps.
A liquid or a powder state polymers are common material for RP processes. Liquid polymers are supplied in resin and solid polymers are in powder form.

Laser beam is a common source of power to solidify the liquid resin and UV lights are used to bind powder type polymer.

2. Selective solidification of solid particles or powder by laser beams.

In order to form a specific shape in 3D, selective solidification is required on each layer. To that end, the initial 3D model has to be decomposed into layers from the top to bottom so that corresponding shape in each layer will be selectively solidified by a source of power.

3. Binding of liquid or solid particles by gluing or welding.

Liquid resins are photosensitive, thus it solidifies once exposed to the source of curing energy. Binding of each layer with its previous layer takes place naturally as the current layer solidifies on the surface of the previously solidified layer. Powder-type polymer requires a type of binding material to produce each layer. For instance, in the original implementations, starch and gypsum plaster fill the powder bed, the liquid “binder” being mostly water to activate the plaster. The binder also includes dyes (for color printing) and additives to adjust viscosity, surface tension, and boiling point to match print head specifications. The resulting plaster parts typically lack “green strength” and require infiltration by melted wax, cyanoacrylate glue, epoxy, etc. before regular handling.

4. Cutting and laminating the sheet materials.

Although majority of the 3D-layered manufacturing processes fall into the first three categories, LOM (laminated object manufacturing) is another innovative technology used for layered manufacturing. LOM is a rapid prototyping system developed by Helisys Inc. Cubic Technologies is now the successor organization of Helisys. In it, layers of adhesive-coated paper, plastic, or metal laminates are successively glued together and cut to shape with a knife or laser cutter. Objects printed with this technique are additionally modified by machining or drilling after printing. Typical layer resolution for this process is defined by the material feedstock and usually ranges in thickness from one to a few sheets of copy paper.

5. Melting and resolidification.

Depending on the kind of process used for 3D printing, appropriate post-cure had to be performed on the finished model. Post-cure completes the polymerization process and improves the final mechanical strength. Generally, ultraviolet radiation is used to provide curing energy to improve the layer binding and harden the finished model.

7.3.2 *Specific RP&M Processes*

Among many RP&M processes, we investigate five most popular RP&M processes in this section. They are:

- Stereolithography
- Solid ground curing
- Selective laser sintering
- 3D printing
- Laminated-object manufacturing

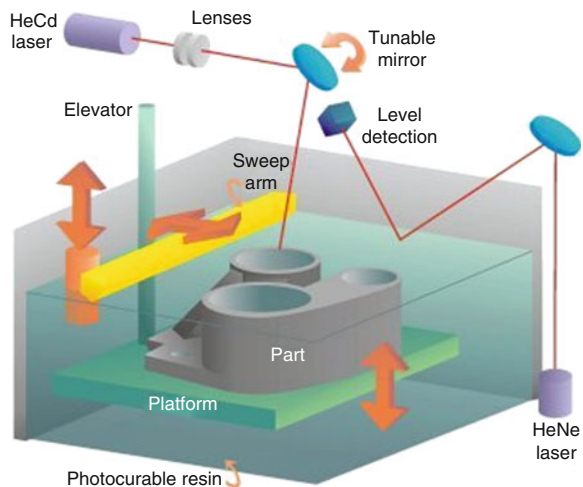
7.3.2.1 Stereolithography

The term “stereolithography” was coined in 1986 by Charles (Chuck) W. Hull, [13] who patented it as a method and apparatus for making solid objects by successively “printing” thin layers of an ultraviolet curable material layer by layer. Hull’s patent described a concentrated beam of ultraviolet light focused onto the surface of a vat filled with liquid photopolymer. The light beam draws the object onto the surface of the liquid layer by layer and uses polymerization or cross-linking to create a solid, a complex process which requires automation. In 1986, Hull founded the first company, 3D Systems Inc., to generalize and commercialize this procedure, which is currently based in Rock Hill, SC. In addition, attempts have been made to construct mathematical models of the stereolithography process and design algorithms to determine whether a proposed object may be constructed by the process or not [14].

The overall process sequence starts with liquid polymer filled in a vat (Fig. 7.10). A photosensitive polymer that solidifies when exposed to a light source is maintained in a liquid state. A platform that can be elevated is located just one layer of thickness below the top surface of the liquid polymer. The UV laser scans the polymer layer above the platform to solidify the polymer and give it the shape of the corresponding cross-section. The platform is lowered into the polymer bath to the layer thickness to allow liquid polymer to flow over the part to begin the next layer. These steps are repeated until the top layer of the part is generated. Post-curing is performed to solidify the part completely [15].

One of many challenges in stereolithography is to build a part with undercut structure, since the newly formed layer right above the undercut does not have a layer to be bonded together. Support structures are needed often to accommodate

Fig. 7.10 Principle of stereo lithography [35]



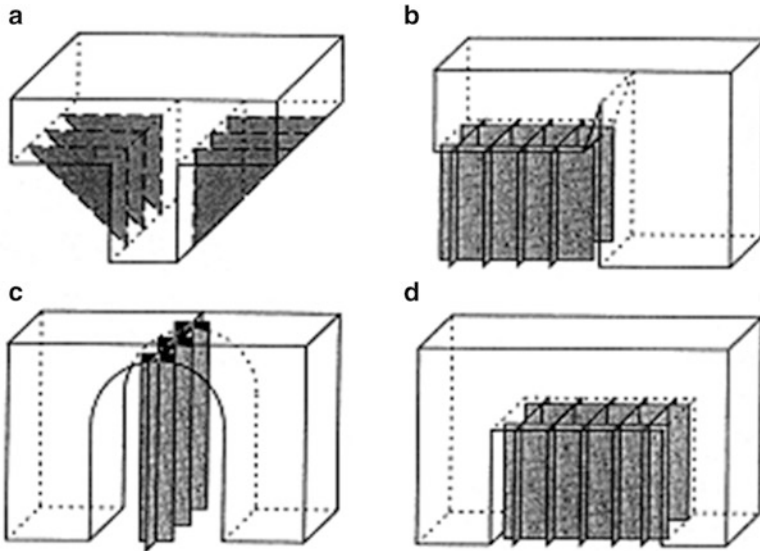


Fig. 7.11 Support structures. (a) Gusset. (b) Island. (c) Ceiling within an arch. (d) Ceiling

undercut structure of the part. Various forms of support structures are proposed to deal with undercut structures. For instance, several support structures are illustrated in Fig. 7.11. Gusset is a support structure that gradually grow laterally from the bottom to top to support a relatively small undercut. The island or ceiling is needed to provide support for relatively large undercut shapes. Arch shape, in general, are self-supportive, thus a minimal support at the center may be required.

7.3.2.2 Solid Ground Curing

Solid ground curing (SGC) is a photo-polymer-based additive manufacturing (or 3D printing) [16] technology used for producing models, prototypes, patterns, and production parts, in which the production of the layer geometry is carried out by means of a high-powered UV lamp through a mask. The key technique of SGC is the exposure of each layer of the model by means of a lamp through a mask. The processing time for the generation of a layer is independent of the complexity of the layer [17]. SGC was developed and commercialized by Cubital Ltd. of Israel in 1986 [18] in the alternative name of Solider System. While the method offered good accuracy and a very high fabrication rate, it suffered from high acquisition and operating costs due to system complexity. This led to poor market acceptance. While the company still exists, systems are no longer being sold. Nevertheless, it is still an interesting example of many technologies other than stereolithography; its predeceasing rapid prototyping process that also utilizes photo-polymer

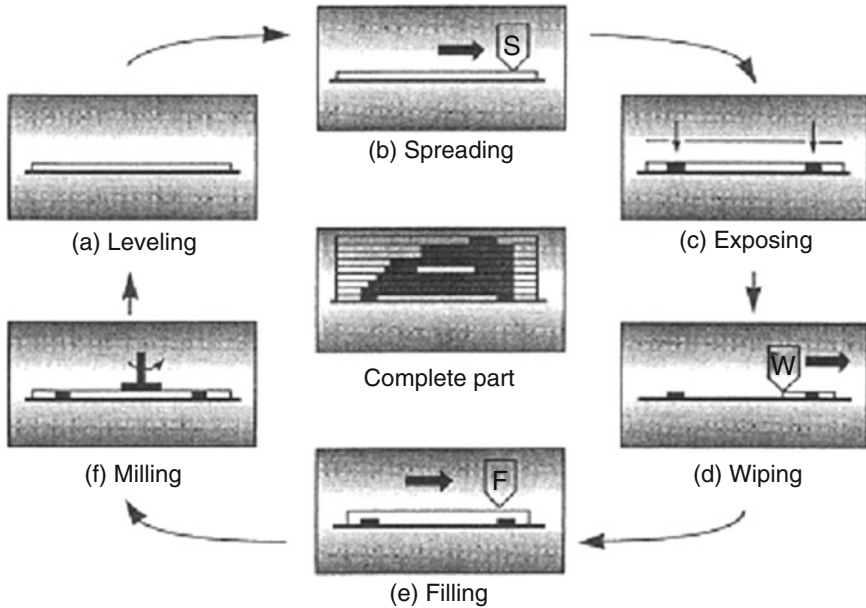


Fig. 7.12 Solid ground-curing process

materials [19]. Objet Geometries Ltd. of Israel retains intellectual property of the process after the closure of Cubital Ltd. in 2002 [18].

The overall process starts with the calculation of cross-section of each slice layer from the geometric model of a part and desired layer thickness [20]. Then, an optical mask for each layer is generated conforming to each cross-section. Once the masks are prepared, the actual building process starts with leveling of the top surface in the platform (see Fig. 7.12a). After leveling, the platform is covered with a thin layer of liquid photopolymer (Fig. 7.12b). Then, the mask corresponding to the current layer is in position over the surface of the liquid resin, and the resin is exposed to a high-power UV lamp (Fig. 7.12c). The residual liquid is removed from the workpiece by an aerodynamic wiper (Fig. 7.12d). A layer of melted wax is spread over the workpiece to fill voids, then solidified by applying a cold plate (Fig. 7.12e). The layer surface is trimmed to the desired thickness by a milling disk (Fig. 7.12f). The steps from (a) to (f) repeats until the top layer is finished. The wax is melted away upon completion of the part.

7.3.2.3 3D Printing

3D Printing is based on the ink-jet printing technology. Liquid binder is ejected instead of ink in 3D printing. Z Corporation is the first company who commercialized the technology (commonly abbreviated Z Corp.). Z Corp then was acquired by 3D Systems on January 3, 2012. In 1995, a new 3D printing technology, ZPrinting,

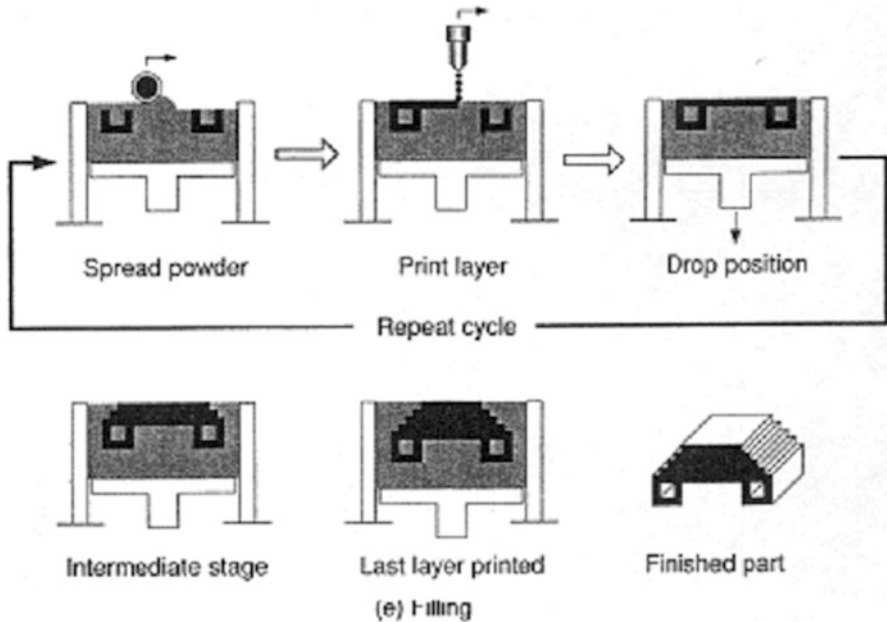


Fig. 7.13 3D printing

was developed at the Massachusetts Institute of Technology [21–23]. 3D printing or, also known as, ZPrinting relates to the z axis which adds depth to the other two axes of x, y direction. As in many other rapid prototyping processes, the part to be printed is built up from many thin cross-sections of the 3D model. In ZPrinting, an inkjet-like printing head moves across a bed of powder, selectively depositing a liquid binding material in the shape of the cross-section. A fresh layer of powder is spread across the top of the model, and the process is repeated. When the model is complete, unbound powder is automatically removed [24]. By ZPrinting, parts can be built at a rate of approximately 1 vertical inch per hour.

As shown in Fig. 7.13, the overall process of a 3D printing technology starts with a platform filled with ceramic powder. A platform is located at the height necessary for a layer of ceramic powder to be deposited on the platform to the proper thickness. The layer of ceramic powder is selectively raster-scanned with a print head that delivers a liquid binder, causing particles to adhere to each other. Then, the platform is lowered by the layer thickness to permit a new layer of powder to be deposited. The new layer is scanned, conforming it to the shape of the next surface cross-section and adhering it to the previous layer. Last two steps are repeated until the top most layer of the part is generated. A post-process heat treatment is applied to solidify the part.

7.3.2.4 Laminated-Object Manufacturing

Unlike other 3D printing technologies, layers of adhesive-coated paper, plastic, or metal laminates are successively glued together and cut to shape with a knife or laser cutter in LOM, [25]. Objects printed with this technique are additionally modified by machining or drilling after printing. Typical layer resolution for this process is defined by the material feedstock and usually ranges in thickness from one to a few sheets of copy paper [26]. LOM generates parts by laminating and laser-trimming materials that are delivered in sheet form (see Fig. 7.14). The sheets are laminated into a solid block by a thermal adhesive coating. Materials used for LOM process include paper, plastics, composites, and metal.

The overall process of LOM starts with the sheet material from the supply roll fed to the take-up roll. Each sheet is attached to the block, using heat and pressure to form a new layer (use roller). After a layer is deposited, a laser is traced on the layer along the contours corresponding to the current cross-section (only contours are scanned, hence efficient). Areas of the layer outside the contours are cross-hatched by the laser (for easy removal). These are three main steps repeated until the top layer of the part is laminated and cut. After all the layers have been laminated and cut, the result is a part imbedded within a block of supporting material (this needs to be broken into chunks) and delicate block removal process has to be followed

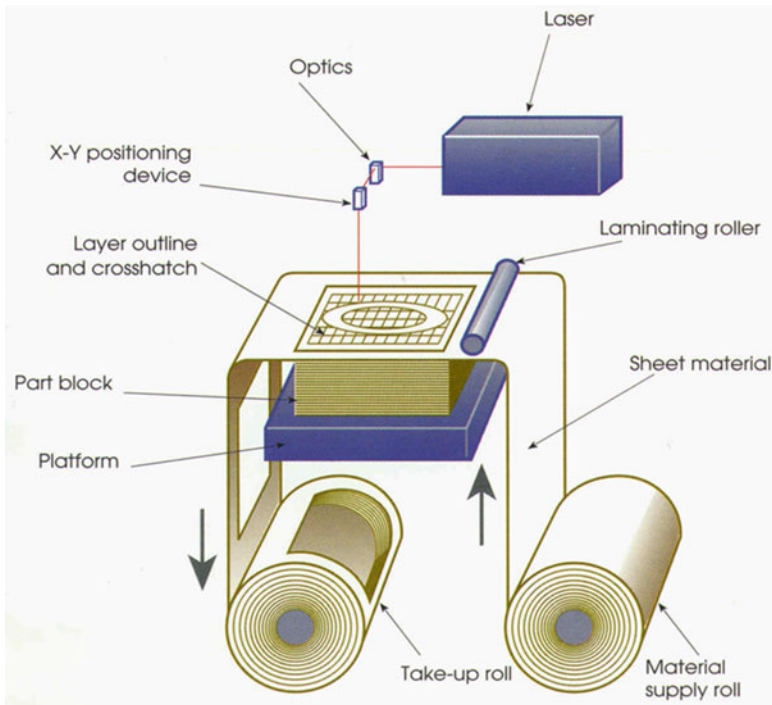


Fig. 7.14 Laminated-object manufacturing [26]

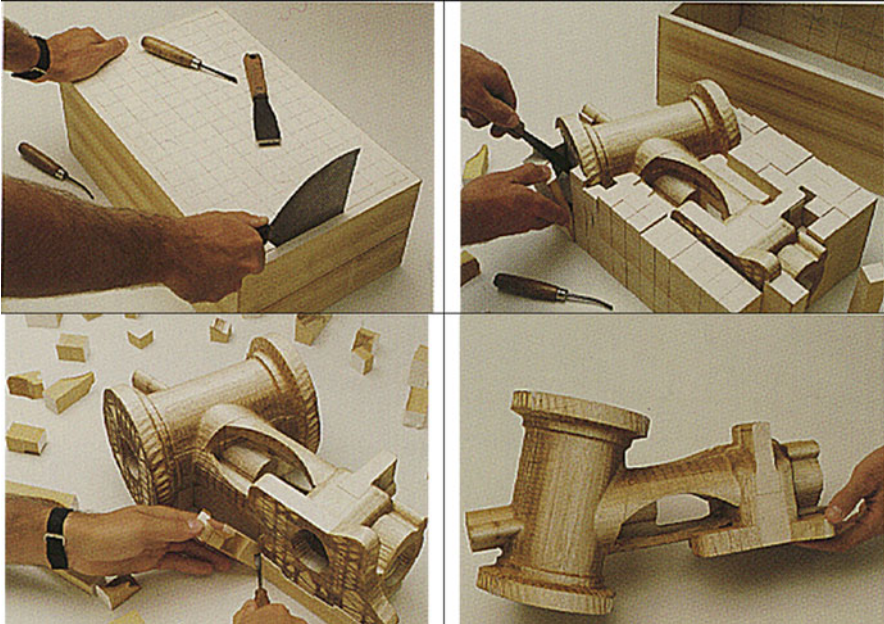


Fig. 7.15 Removing tiles in laminated-object manufacturing [26]

(Fig. 7.15). The resulting part may then be coated with a sealant to keep out of moisture.

There are several advantages in LOM compared to other RP processes. First, external support structures are not required in LOM. Second, LOM can create more stable geometry with minimal distortion since the entire structure will be embedded in a cube until the top layer is finished. Third, there is no need to worry about isolated “island contours.” Fourth, LOM is potentially the fastest technology for building parts with a high ratio of volume to surface area. Finally, resolution of the LOM is high compared to other RPs since the user can decide arbitrarily any thickness of layer by choosing the corresponding sheet material.

However, there are some significant disadvantages in LOM as well. First, the block removal is a complex task of scrapping unnecessary materials upon completion of the layer building process. In addition, careful manual cleanup process is required. A hollow structure with closed surfaces cannot be fabricated by LOM. But for the most part, parts are inhomogeneous and anisotropic in terms of physical properties because parts are formed from alternating layers of material and adhesive in vertical direction.

7.3.3 RP Machine Trend

Despite various RP technologies introduced in this chapter, there are two most popular trends identified in industry. One is Fortus series RP machines (Fig. 7.16) and the other is Objet series RP machines by STRATASYS (Fig. 7.18).



Fortus series is a new line of products from originally what is known as Dimension by STRATASYS. The printing technology is fused deposition modeling (FDM) technology. Two materials (one for models, one for support) are heated in an extrusion head and deposited in thin layers on a modeling base. The material used for the Fortus series varies from ABS and thermo plastic (Fortus 250mc, Fortus 360mc) to Nylon and Ultem (Fortus 400mc, Fortus 900mc) (See Fig. 7.17). The finished surface quality is outstanding and parts from Fortus series are durable and used for real-world applications. The price range varies from 50k to 250k

depending on the model and purchase options. The support material has to be removed by a thermal furnace (Fig. 7.17) with a caustic chemical agent to dissolve the support material. The time to remove the support material varies from 20 min to 20 h, depending on the size of the product and number of parts. One of the advantages of the Fortus series is that the material cost and maintenance cost are relatively less expensive since material does not expire.

Another popular line of product is the OBJET series by STRATASYS. OBJET series uses PolyJet 3D printing technology with up to 28 μm thin layers. It is based on photosensitive polymer curing process. Several models are available in the line of product (see Fig. 7.18).

While Objet23 and 30 are entry level printers, Objet500 Connex3 is a full color 3D printer. Although there are several full color 3D printers in the market, such as ProJet X60 3D printer from 3DSYSTEMS (see Fig. 7.19), Objet series is prevailing in industry mainly because of the strength of the finished parts. ProJet series was a product line of Z-Corp developed with powder curing technology. The big advantage is that it supports full color printing at a competitive price starting from \$30k, while Objet500 Connex3 costs about \$250k. ProJet series, however, requires complicated post-curing process to solidify the finished parts with about 40 min in a curing chamber and surface waxing with glue-type epoxy material. In addition, due to the nature of powder-based parts, the finished part may be crushable, hence less popular for real-world application in industry.



Fig. 7.16 Fortus series [36]

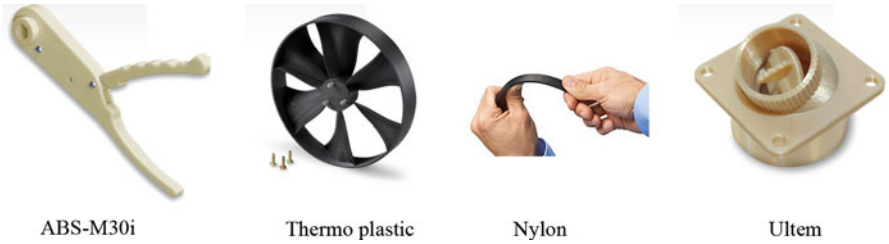


Fig. 7.17 Materials supported by Fortus series [36]



Fig. 7.18 Objet series [36]



Fig. 7.19 ProJet series [36]



Therefore, Objet series is more prevailing compared to ProJet series, though color printing capability is limited. The surface quality by the Objet series is poorer compared to the Fortus series. However, the post-processing is easy and simple. Support material can be easily removed by a peeling tool as well as a water jet clearing chamber (see the picture left). Therefore, it would be a good choice for educational purpose. The strength of the finished parts is superior to that of parts by ProJet, but inferior to that of parts by Fortus series.

The common material for Objet series is rigid white opaque material, which is photo-polymeric liquid resin. Majority of the Objet series produce products with limited color. However, Objet500 Connex3 produces true color products (see Fig. 7.20).



Fig. 7.20 Parts built by Objet series [36]

RP&M Processes

- **Stereo lithography**
- **Solid ground curing**
- Selective laser sintering
- **3D printing**
- **Laminated-object manufacturing**

7.4 Data Structure

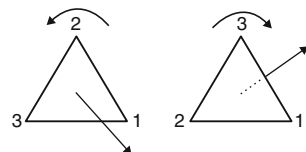
3D rapid prototyping requires a unique data structure for layered manufacturing. First, the entire 3D modeling data has to be converted into multiple layers that represent cross-sectional area of a 3D model. Each layer data contains the details of the cross-section, indicating empty region and solid region. In order to generate layer data, a 3D model has to be converted to the STL file format. That is, wireframe model, surface model, or B-rep/C-rep models have to be converted to the STL file format.

The STL file format is a standard data structure used in most of the RP&M devices. An STL file represents an object as a mesh of connected triangles, by which vertices of each triangle are listed in an order that indicates which side contains the inside volume. For instance, as depicted in Fig. 7.21, right hand rule (thumb pointing the surface normal vector, and other fingers wrapping the vertices in the ascending order) can determine which direction the surface normal vector points to.

In the example in Fig. 7.22, the text file format of a STL file is demonstrated. Each triangle will start with the keyword, “facet,” followed by a normal vector and three vertices that belong to the facet. The keywords “outerloop” and “endloop” define a block of vertices for a triangle, while the term “endfacet” defines the end of the definition of one triangle. All the STL files end with the keyword “endsolid” at the end of the file.

One example of the model conversion from a surface model to the STL file format is shown in Figs. 7.23 and 7.24. However, a question may arise as to why we need to convert the 3D model into STL file. There are several advantages for the conversion, though the STL file is not the only file format available in RP&M process. Below is the summary of the advantage of the STL file format.

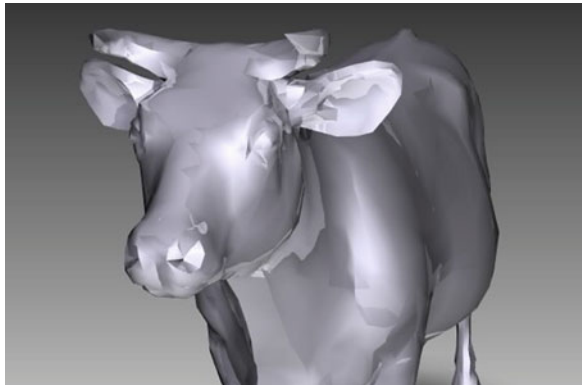
Fig. 7.21 Triangles in STL file format



```
Solid sample.stl created by Wei Feng on 15th OCT. 1994
facet normal -1.000000 0.000000 0.000000
  outer loop
    vertex 140.502634 233.993075 -38.310362
    vertex 140.502634 229.424780 -38.359042
    vertex 140.502634 242.525774 -27.097848
  endloop
endfacet
facet normal 0.903689 0.004563 0.428166
  outerloop
    vertex 134.521310 273.427873 30.342009
    vertex 134.521310 308.505852 30.715799
    vertex 140.502634 334.576026 18.369396
  endloop
endfacet
facet normal -0.903689 0.004563 0.428166
  outer loop
    vertex 140.502634 334.576026 18.369396
    vertex 140.502634 294.929752 17.946926
    vertex 134.521310 273.427873 30.342009
  endloop
endfacet
... ..
endsolid sample.stl
```

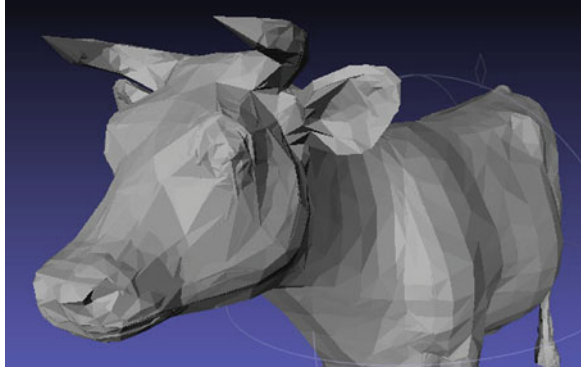
Fig. 7.22 STL file

Fig. 7.23 3D Cow model [38]



1. Easy conversion.
A 3-D model can be converted to STL format by using the standard surface triangulation algorithm. This process is simple and easy mathematically as well as practically.

Fig. 7.24 Converted 3D Cow model [39]



2. Wide range of input.

Any form of 3-D geometry can be converted to a triangulated model. Without knowing the exact model used for a certain product, a wide range of acceptance is a big benefit of STL file format.

3. Simple-slicing algorithm.

Generating layer information by slicing STL format is relatively easy. Compared to other models that may contain complex equations to describe certain curves or geometry, STL expresses the entire model as a collection of triangles.

4. Splitting STL model is easy.

STL allows splitting a model into pieces with ease. This capability is very useful when a complete model does not fit into a RP machine. Multiple models are created separately and put together later for a complete assembly.

The STL file format, however, has some disadvantage compared to other data format as well. Below is the list of the disadvantage of the STL file format.

1. Verbosity and data redundancy.

Data are redundant since the coordinates of the same vertices appear several times in the STL file. This is the disadvantage due to the simplicity of the STL file format.

2. Error due to approximation.

Since the entire model is expressed as a collection of triangles, the overall quality of representation is relatively poor since all the curves are approximated as a collection of lines.

3. Truncation errors.

Since the entire model is created with no topological information, there are truncation errors innate to the STL model.

4. Lack of information.

Due to the above reasons, not only the shape approximation loses details of curved surfaces, but also all other information such as connectivity, association, and hierarchical structure information will be lost, except basic geometry of the original model.

RP Data Format

The STL file format is a standard data structure used in most of the RP&M devices. An STL file represents an object as a mesh of connected triangles, by which vertices of each triangle are listed in an order that indicates which side contains the inside volume.

7.5 Physics Behind SFF

Selective Curing

With selective curing, a shapeless bulk material, mostly liquid polymer, is selectively solidified by exposure to an energy source of variable nature such as laser and UV light, in order to build up an object; this method involves phase change from liquid to solid. One can differentiate the selective laser-curing from selective UV-curing.

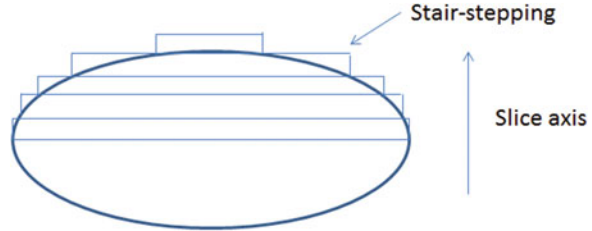
Selective laser-curing

Selective laser-curing is the most widespread of all SFF methods. Currently, more than 85 % of all the installed RP systems in service centers are based on this technology. At present, selective laser-curing comprises twelve commercial systems of which 3D System's Stereolithography process holds the largest share. It was the first and is now the most established SFF method. Stereolithography is also proved to be the process with the most extensive research background, providing a good understanding of all operating variables and their interrelationships. Each of commercial vendors differs in various aspects, which cannot be covered within the context of this book. Therefore, the field of selective laser-curing is explained on the example of 3D system's Stereolithography process.

The cross-sectional data of the sliced STL file is actuating electro-mechanical drives. These drives in turn are guiding a set of swiveling optical lenses through which a laser beam (argon-ion, helium-cadmium laser) is sent. The lenses direct the laser beam of 0.2–0.25 mm diameter onto a vat of liquid photopolymer in a point-by-point mode reaching a maximum scanning speed of 10 m/s. The epoxy or acrylate resin solidifies in those areas which were sufficiently exposed to the laser-beam. After one cross-section is completely traced, the platform on which the first layer was built is lowered incrementally by the measure of exactly one layer thickness, submerging the hardened material under the surface of the liquid in the vat. Another pass of the laser then hardens the next layer of polymer.

Scanning is accomplished in a step-by-step movement during which the laser beam is applied in micro-second impulses. Because of that and due to the Gaussian energy profile of the laser beam which means highest energy level in the center, and lowest energy level at the edges, the beam is able to penetrate the resin deeper in its center than at its edges. The polymerization of resin follows this profile, solidifying

Fig. 7.25 Vertical cross-sectional view of a 3D prototype model



tiny polymer paraboloids, also called voxels or bullets in diameter of the applied laser beam (0.2–0.25 mm).

In order to create a successful RP model in the most cost- and time-efficient way, one should understand the basic physics behind the layer manufacturing as well as principle rules for optimal use of the machine. It starts with the good understanding of the principle of the layered manufacturing. In this section, we will use the stereolithography as a 3D RP process, but the same or similar principle is applicable to other methods. The first principle of the layered manufacturing is that no matter how thin each layer is, there will be stepwise changes in shape along the vertical direction (see Fig. 7.25)

Therefore, the part orientation is one of the utmost important matters in 3D RP-produced part's quality. The accuracy of the part depends on how the object is oriented in the vat. Due to the stair stepping along the slice axis, it is recommended that the most demanding surface in quality has to be oriented horizontally. Another issue in 3D RP process is balancing between the build time and step resolution. The taller the object is in place in the vat, the more the process time would take in general. A proper compromise has to be made ahead of time. In 3D RP process, the accuracy in model representation is considered to be more important. Another issue is on minimizing the use of the support material. Depending on how a model is oriented in the vat, the amount of support material may vary significantly.

Finally, since the 3D RP process is a layered manufacturing process, the way each layer is combined will determine how sound the finished part is structurally. There are two parameters we need to consider for the structural integrity: cure depth and layer thickness. Cure depth is defined as the depth of the curing area of the current layer from the surface of the liquid resin. The cure depth is determined by the amount of energy injected in certain area of the surface by the laser beam. The more the energy is introduced on the surface, the deeper the curing depth will be. Once a laser beam is illuminated on the surface, the curing area will be in a bell shape with the transitional area at the end due to the nature of heat transfer in a liquid resin. Therefore, the energy received from the laser propagates through the bell shape region and reach the cold resin. Although most of the bell-shaped area is instantaneously solidified by the laser energy, there will be a transitional area at the edge along the bell-shaped region where two phases of solid and liquid coexist. Since the transitional area of the interface between solid and liquid is unstable, the cure depth has to be more than the layer thickness for firm connection between the current layer and the previous layer.

Important Aspects of RP Process

- Part orientation due to stair stepping along the vertical direction
- Balancing between the build time and step resolution (the taller the object is in place in the vat, the more the process time would take in general)
- Structural integrity (cure depth and layer thickness)

Cure depth

There are several parameters that determine the cure depth (Fig. 7.27). First, the most influential parameter is the intensity of the energy. Depending on how much energy is absorbed, the cure depth will be determined. The size of the laser beam also affects the cure depth significantly. The size of the laser beam not only affects the cure depth, but also the resolution of the part, thus caution for an optimal balance has to be exercised. Another important parameter is the scanning speed. Faster laser head will transfer less energy to the liquid resin, resulting in shallow cure depth. Finally, the property of the photopolymer is an important factor. A photopolymer is a polymer that changes its properties when exposed to light, often in the ultraviolet or visible region of the electromagnetic spectrum. These changes are often manifested structurally, for example, hardening of the material occurs as a result of cross-linking when exposed to light. An example is shown below depicting a mixture of monomers, oligomers, and photoinitiators that conform into a hardened polymeric material through a process called curing. Depending on the ratio of monomers, oligomers, and photoinitiators, the speed of curing will be different (Fig. 7.26).

Changes in structural and chemical properties can be induced internally by chromophores that the polymer subunit already possesses, or externally by addition of photosensitive molecules. Typically, a photopolymer consists of a mixture of multifunctional monomers and oligomers in order to achieve the desired physical properties, and therefore a wide variety of monomers and oligomers have been developed that can polymerize in the presence of light either through internal or external initiation. One of the advantages of photo-curing is that it can be done selectively using high energy light sources, for example, lasers (See Fig. 7.27). However, most systems are not readily activated by light, and in this case, a photoinitiator is required. Photoinitiators are compounds that upon radiation of light decompose into reactive species that activate polymerization of specific functional groups on the oligomers. An example of a mixture consists of monomeric styrene and oligomeric acrylates. Correct combination of all the compounds has to be thoroughly investigated for the best results in each 3D printing technology.

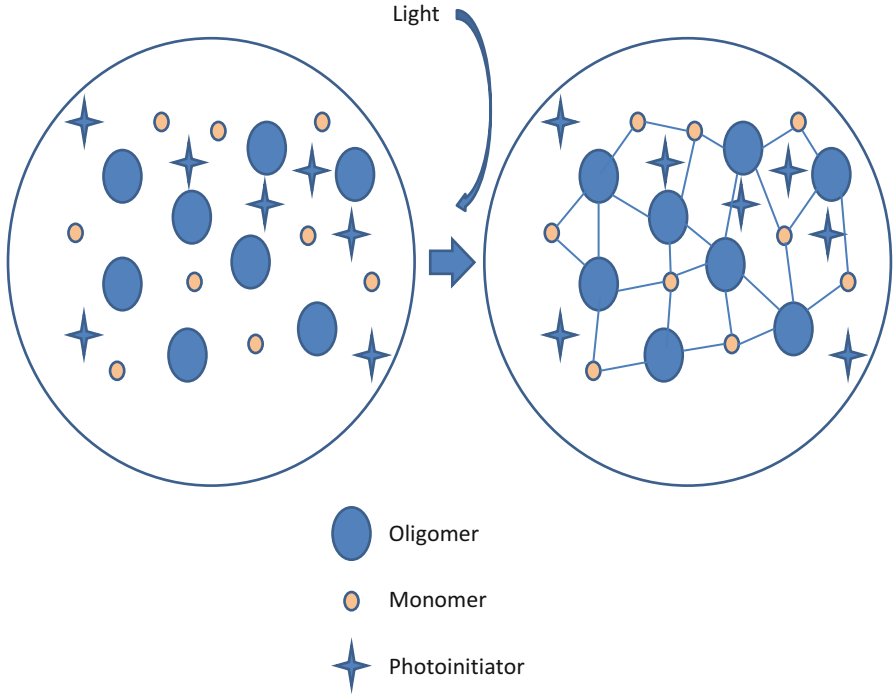


Fig. 7.26 Curing process

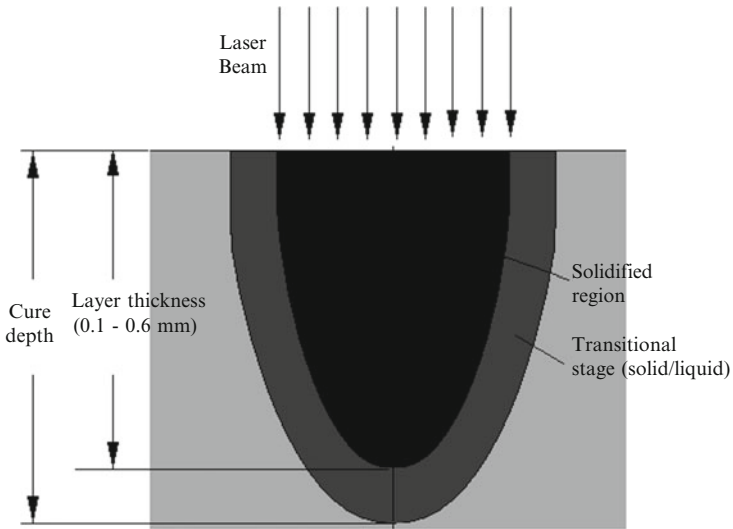


Fig. 7.27 Cure depth

Cure Depth Parameters

- Cure depth must be larger than the layer thickness
- The amount of energy absorption depends on the intensity and size of the laser beam, scanning speed, and the property of the photopolymer

7.6 Post-Processing

Once a 3D rapid prototyping model is built, the 3D structure represents the design of the model to the level of accuracy that a process can achieve. However, the finished model is, in general, not immediately usable due to several reasons such as premature surface strength and layer connectivity. The post-curing process is required to complete the polymerization process and to improve the final mechanical strength. However, depending on the specific RP&M process, post-processing varies. Therefore, a proper post-processing has to be followed, especially for those made for further manufacturing processes. In the stereo-lithography process, for example, following two-part removal and cleaning processes has to be applied.

- Together with the platform, the part is tilted on edge to drain excess liquid resin back into the vat.
- Parts are placed in a cleaning apparatus.

In the most post-curing process, ultraviolet radiation is the common source of energy. The output wavelengths need to be optimized to achieve uniform polymer post-curing. Depending on the application of the RP&M, further post-processing will be required, though removing supports are sufficient for a concept model. For instance, for practical applications such as soft tooling or master for investment casting, hand sanding, mild glass bead blasting, or some combinations are required.

Post-Processing

- Post-curing
 - Post-cure completes the polymerization process and improve the final mechanical strength.
 - Use ultraviolet radiation.
 - Need to optimize the output wavelengths to achieve uniform polymer post-curing.
- Part finishing
 - For a concept model, removing supports is sufficient.
 - For actual application (soft tooling, master for investment casting . . .), hand sanding, mild glass bead blasting, or some combinations are required.

References

1. Sculpture exhibition School of the Art Institute of Chicago. <http://blogs.saic.edu/sugs/exhibitions/artifact/>
2. JTEC/WTEC Panel Report on Rapid Prototyping in Europe and Japan, p 24
3. NSF JTEC/WTEC Panel Report-RPA. http://www.wtec.org/pdf/rp_vi.pdf
4. http://www.sme.org/uploadedFiles/Publications/ME_Magazine/2012/April_2012/April%202012%20f1%20Additive.pdf
5. eFunda, Inc. Rapid prototyping: an overview. www.Efunda.com. Accessed 14 June 2013
6. Interview with Dr Greg Gibbons, Additive Manufacturing, WMG, University of Warwick. Warwick University, Knowledge Centre. Accessed 18 Oct 2013
7. Medical applications of rapid prototyping intech open books. http://cdn.intechopen.com/pdfs/20116/InTech-medicalapplications_of_rapid_prototyping_a_new_horizon.pdf
8. <http://www.pinkbike.com/news/Worlds-first-3D-printed-bike-2014.html>
9. <http://3dprintingindustry.com/2014/04/14/3d-printed-future-food/>
10. Calvert G (1994) Rapid prototyping for experimental analysis. In: Proceedings of colloquium on “rapid prototyping in the UK”. IEE Manufacturing Division, Digest No. 1994
11. Gibbons GJ, Hansell RG, Norwood AJ, Dickens PM (2003) Rapid laminated die-cast tooling. *Assembly Autom* 23(4):372–381
12. https://www.youtube.com/watch?v=2lewK1TiQ_c
13. How stereolithography works. <http://THRE3D.com>. Accessed 4 Feb 2014
14. Asberg B, Blanco G, Bose P, Garcia-Lopez J, Overmars M, Toussaint G, Wilfong G, Zhu B (1997) Feasibility of design in stereolithography. *Algorithmica* 19(1/2):61–83
15. <http://www.youtube.com/watch?v=ygHVVKkJWII>
16. The engineer: the rise of additive manufacturing. <http://www.theengineer.co.uk/in-depth/the-big-story/the-rise-of-additive-manufacturing/1002560.article>
17. Gebhardt IA (2003) Rapid prototyping: industrial rapid prototyping system: prototyper: solid ground curing—Cubital, pp 105–109
18. Solid Ground Curing. <https://kylestetzerp.wordpress.com/2009/05/20/solid-ground-curing-sgc/>
19. Lee KW (1999) Principles of CAD/CAM/CAE systems: rapid prototyping and manufacturing: solid ground curing, pp 383–384
20. <http://www.youtube.com/watch?v=y4N4AxKQPec>
21. Origin of Company Name. Accessed 14 Jan 2011
22. Printers produce copies in 3D. BBC News. August 6, 2003. Accessed 31 Oct 2008
23. Grimm T (2004). User’s guide to rapid prototyping. SME, p 163. ISBN 978-0-87263-697-2. Accessed 31 Oct 2008
24. Sclater N, Chironis NP (2001) Mechanisms and mechanical devices sourcebook. McGraw-Hill Professional, p 472. ISBN: 978-0-07-136169-9. Accessed 3 Oct 2008
25. Karunakaran KP, Dibbi S, Shanmuganathan PV, Raju DS, Kakaraparti S (2000) Optimal stock removal in LOM-RP. *Proc Inst Mech Eng B J Eng Manuf* 214(10):947–951
26. How laminated object manufacturing works. <http://THRE3D.com>. Accessed 3 Feb 2014
27. <http://www.fractal.org/Fractal-Research-and-Products/Biomedical-Rapid-Prototyping.htm>
28. <http://www.contourcrafting.org>
29. <http://cielotech.wordpress.com/2013/11/16/rapid-prototypingrpm>
30. <http://dir.indiamart.com/pune/rapid-prototyping.html>
31. <https://www.youtube.com/watch?v=WceR92i7Q9Y>
32. <http://www.custom3dsolutions.com/gallery.html>
33. <http://www.advancedtechnologiesinc.com/uav-experimental-aircraft.php>
34. <http://www.armstrongmold.com/pages/twosteparticle.html>
35. <http://www.princeton.edu/~cml/html/research/mems.html>

36. <http://www.stratasys.com/3d-printers/production-series>
37. <http://www.3dsystems.com/3d-printers/professional/overview>
38. http://www.the3dstudio.com/product_details.aspx?id_product=522262
39. <http://rasterweb.net/raster/2011/11/14/simplifying-stl-files-with-meshlab/>

Chapter 8

Finite Element Modeling and Analysis

The Big Picture

You need to understand terminologies and two basic principles of FEM analysis: Mesh generation and mathematical analysis

Discover

- Understand terminologies
- Understand node generation
- Understand mesh generation
- Understand governing equation
- Understand analysis process

8.1 What Is FEM?

Finite element method (FEM), also known as finite element analysis (FEA) embraces broad academic areas of a distributed body analysis. One significant challenge in modern engineering is to deal with an arbitrarily shaped solid body that encompasses complex geometry, or a design with constraints, or an aerodynamic design. While most of the Newtonian physics-based solid body problems are solved by the lumped mass approach, an arbitrarily shaped body cannot be easily solved by the same approach due to the complicated geometry. In addition, more engineering parts in modern engineering are built not only with complicated geometries but also with heterogeneous material composition. The distributed mass (continuum mass) approach as a complementary method, thus, has been widely accepted in various modern engineering domains. FEM is a computer

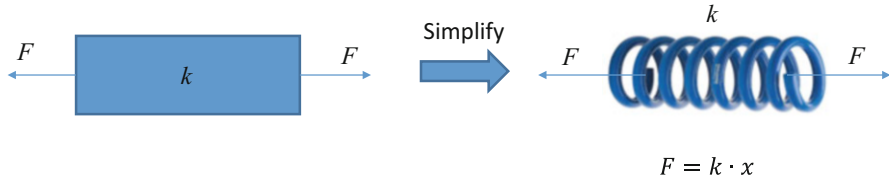


Fig. 8.1 Lumped approach

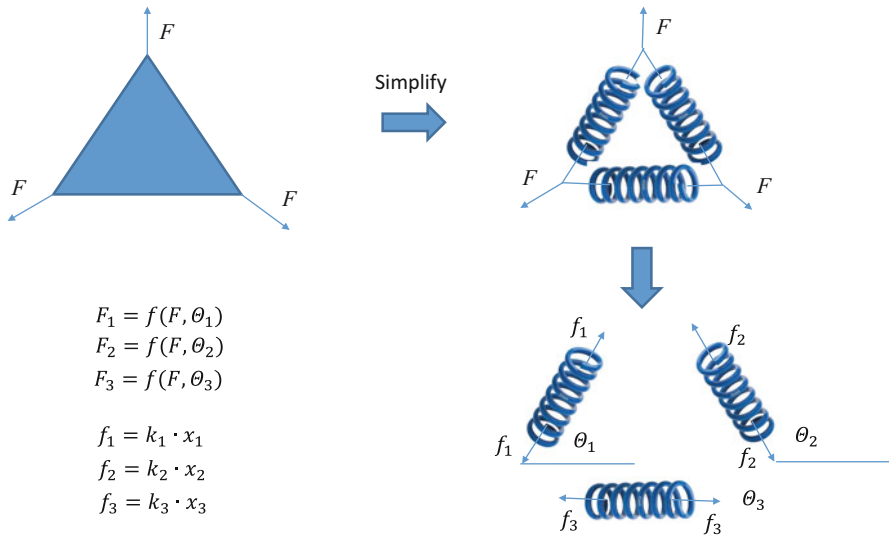


Fig. 8.2 Distributed mass approach

based approach developed for distributed mass problems where multiple governing equations are put together and solved with boundary constraints. Multiple unknowns such as displacement, temperature, vibratory modes, etc. can be calculated simultaneously in a matrix form of governing equations.

In the lumped mass approach, a rigid body will be transformed into a simple spring system where the Hook’s law is applicable (see Fig. 8.1). Once the external force and the spring constant or the elastic modulus of a rigid body is known, then the displacement can be found by the Hook’s law. In the distributed mass approach, however, a complex geometry is, first, decomposed into a multiple cross-linked spring structure called “Truss” as shown in Fig. 8.2. Then, a governing equation for each spring will be developed as a function of linearly applied force along the main member direction of each spring. Since the force on each spring is the function of the external forces and the geometric orientation of the spring, the governing equation of each spring can be solved by hook’s law. This allows us to find the displacement of each spring (x_1, x_2, x_3) by Hook’s law. In general engineering cases, the number of springs can be easily over hundred or thousand depending on the

demand of accuracy and complexity of the shape, thus computerized calculation is inevitable.

In spite of the great power of FEA, disadvantages of computer solutions must be kept in mind: they do not necessarily reveal how the stresses are influenced by important variables such as material properties and geometrical features. Furthermore, errors in input data can produce wildly incorrect results that may be overlooked by the analyst. Perhaps the most important function of theoretical modeling is that of sharpening the designer's intuition: users of finite element codes should plan their strategy toward this end, supplementing the computer simulation with as much closed form and experimental analysis as possible.

In practice, FEA usually consists of three principal steps:

1. **Preprocessing:** The user constructs a model of the part to be analyzed in which the geometry is divided into a number of discrete subregions, or elements, connected at discrete points called nodes. Certain of these nodes will have fixed displacements, and others will have prescribed loads. Creation of these models can be extremely time consuming. Therefore, commercial codes with a user-friendly graphical preprocessor are available to assist in this rather tedious chore. Some of these preprocessors can overlay a mesh on a preexisting CAD file, so that FEA can be done conveniently as part of the computerized drafting-and-design process.
2. **Analysis:** The dataset prepared by the preprocessor is used as an input to the finite element code, which constructs and solves a system of linear or nonlinear algebraic equations. The formation of the stiffness matrix is dependent on the type of problem under consideration, and this module will outline the approach for truss and linear elastic stress analysis. Commercial codes may have very large element libraries, with elements appropriate to a wide range of problem types. One of FEA's principal advantages is that many problem types can be addressed with the same code, merely by specifying the appropriate element types from the library.
3. **Postprocessing:** In the earlier days of FEA, the user would be overwhelmed by the numbers generated by the code, listing displacements and stresses at discrete positions within the model. It is easy to miss important trends and hot spots this way, and modern codes use graphical displays to assist in visualizing the results. A typical postprocessor display overlays colored contours representing stress or strain levels on the model, showing a full-colored picture similar to that of photoelastic or Moire experimental results.

Finally, it is investigator's responsibility to validate the analysis results obtained after the postprocessing. It is always risky accepting the results without validation since a small error anywhere in the analysis process or a mathematical problem such as singularity may cause unexpected errors in the result.

Overall, FEM is consisted of following four steps.

1. Convert the given model to a mesh structure
2. Generate system equations
3. Solve equations for unknowns
4. Validate the solution

In FEM, the first step is to convert a rigid body model into a truss structure that consists of multiple triangles called mesh. Due to the magnitude of countless number of meshes, the mesh generation step requires automatic mesh generation algorithms. In the following section, we will study the automatic mesh generation process.

8.2 Automatic Mesh Generation

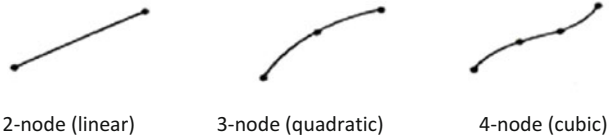
As mentioned earlier, the first step in FEM is to convert a rigid body into a truss structure with multiple meshes. Quadrilateral or triangular elements are the most popular mesh shape for three-dimensional surfaces, shells, or two dimensional geometries. For a solid geometry, tetrahedral meshing is used. When it comes to automatic mesh generation, there are two important principles ought to be addressed such as:

1. More mesh means better accuracy.
2. Element dimensionality must be the same as the problem domain.

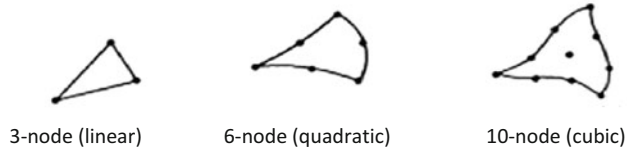
First, the more mesh the FEM program generates, the better the analysis accuracy will be. In general, a FEM package allows a user to choose a certain resolution of the mesh or provides a recommended resolution for a given model. Although better accuracy can be obtained by increasing the number of meshes in the model, taking the fact that FEM analysis costs significant CPU time and computational resources into account, computation time and resulting accuracy have to be leveraged. The most common reference is the demanding party's request for accuracy. An engineer should try not to exceed the required accuracy by controlling the mesh size to maximize the productivity.

Second, as aforementioned in the previous section, the type of mesh has to comply with the problem domain. If the target body is a beam or a link in a truss, a linear or a curved line element would suffice the need (see Fig. 8.3). Again, depending on the demanding accuracy, the element type has to be decided as proposed in the figure. Mesh generation is a two-stage operation: node generation, and mesh element generation. In node generation stage, evenly distributed nodal points are generated throughout the entire model. Even distribution and complete coverage of the target body are the important aspects of the first stage. Once nodal points are generated, then each node has to be connected to generate a truss structure in 2D objects and a tetrahedral mesh structure in 3D objects. In the next two sections, we will study how these two stages can be achieved.

Beam/truss elements



Triangular elements



Tetrahedral elements

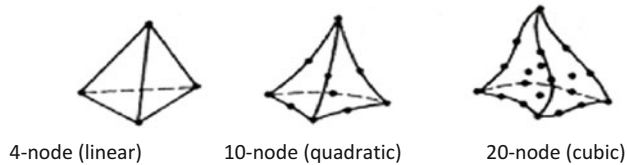


Fig. 8.3 Element dimensionality

8.2.1 Node Generation

In order to generate nodes, there are several parameters that have to be set beforehand. First, the geometric model has to be identified. Second, mesh density and element type have to be determined. There are various node generation techniques available, but we will investigate two most historically important node generation techniques: Cavendish's method and Shimada's method.

8.2.1.1 Cavendish's Method

The Cavendish's method is semiautomatic method in terms of node distribution. Although it is not popular anymore, it represents one of the early concepts of automatic node generation. It is easy to understand and well defined for 2D node generation. The Cavendish's method adds nodal points in a 2D model first by adding nodes on the boundary manually followed by interior nodes automatically. It starts with superimposing the given object with a square grid. The grid size should be correlated to the analysis resolution. Then the following iterative procedure will create nodal points inside of the object.

1. Generate one interior node randomly for each square
2. Accept the node if
 - (a) a new node falls inside the object.
 - (b) a new node has a distance greater than $r(i)$ from the boundary and previous nodes.

3. Reject the node if

- (a) failed to find an acceptable node in a limited number of attempts ((ex) 5, 6. . .) and skip the current square.

During the above procedure, multiple locations can be tested to create a nodal point in each square or an empty square may occur without a node if it fails to find one in a number of attempts set by a program. For instance, the two white dots in one of the square in Fig. 8.4 are two rejected dots since it is either too close to a previous node or to the one along the edge line. As a result of the above node distribution operation, one can obtain nodes evenly distributed along the edge and the internal area of an object.

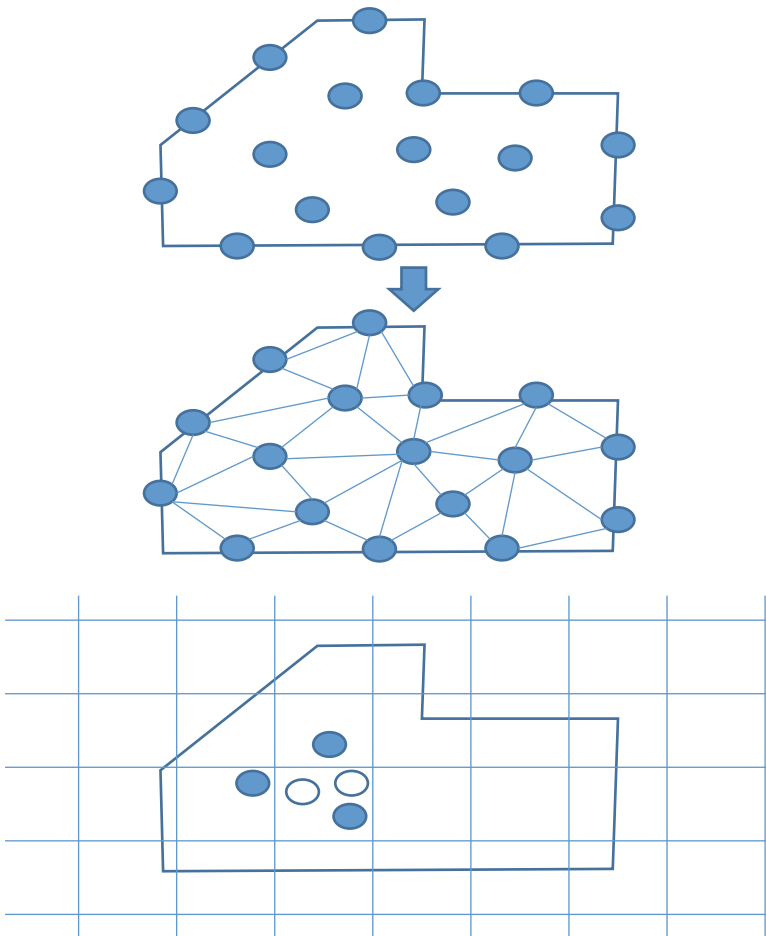


Fig. 8.4 Cavendish's method

8.2.1.2 Shimada's Method

Another popular approach used in early year of FEM is the Shimada's method, by which node and mesh generation takes place simultaneously by the force equilibrium concept. Shimada introduced a method of adding bubbles inside of the object randomly but under a certain rule.

First, the size of each bubble is determined by desired mesh density. In addition, locations of bubbles are determined to satisfy force equilibrium of reactions between bubbles. As a result, each neighboring bubbles are in equidistance from its own center to the center of each bubble in contact. Once all of the bubbles are distributed and the entire area inside of an object is filled, then the centers of all of bubbles are taken as the nodes. One example of the Shimada's method is shown in Fig. 8.5. The distribution of bubbles along edge is automatic in Shimada's method by using the two constraints:

1. Bubbles on the edge place their center along the edge line.
2. All of the reaction forces between bubbles are in equilibrium.

One most important advantage of the Shimada's method is that it not only generates nodes, but also creates mesh at the same time. Since all of the bubbles in contact can be sorted out easily, the contact information can be used to create mesh by connecting centers of bubbles. However, node generation and mesh generation are two separate stages in general. In the next section, we study two important mesh generation methods.

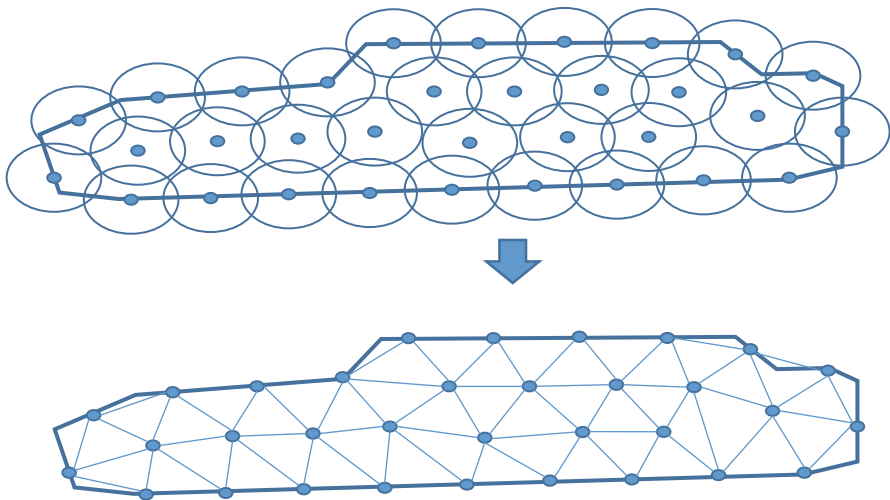


Fig. 8.5 Shimada's method

8.2.2 Mesh Generation

Mesh generation or element generation is a process of generating triangular elements with generated nodes in the previous process. There are two important aspects of element generation.

1. No elements are overlapped
2. Cover the entire object

Autonomous element generation satisfying two aspects above is not a trivial problem. Below is a list of some useful element generation methods popular in FEM.

- Lee's method
- Delaunay triangulation method
- Topology decomposition
- Geometry decomposition
- Grid-Based approach
- Mapped element approach

We limit the scope of our study to the Delaunay triangulation method and the Grid-Based approach.

8.2.2.1 Delaunay Triangulation Method

Delaunay triangulation method is one of the most popular mesh generation methods. It is based on "Voronoi diagram" or "Dirichlet tessellation" method. It divides an object into meshes with triangles for a 2D plane or tetrahedral for a 3D object. As a result of the Delaunay triangulation method, the sum of the smallest angles of all of the triangles are maximized, meaning the number of thin mesh elements will be minimized. The Voronoi diagram is mathematically well defined so that the implementation of the division algorithm is easy and produces the stable and consistent results. The definition of the Voronoi diagram is as below.

$$V_i = \left\{ x \mid |x - P_i| = |x - P_j| \quad \text{for all } i \neq j \right\}$$

Mathematically, a Voronoi diagram is a set of points that satisfies the condition of equidistance from the previously defined points. For example, P_i and P_j in the above equation are two predefined points on a plane. Then the collection of the points represented by x in the equation is a line that is in equidistance from two points to any point on the line (see Fig. 8.6).

When the number of points multiplies, multiple lines by which all neighboring points are divided will create a network-like diagram (Fig. 8.7). A Voronoi diagram of a set of N points, P_i ($i = 1, 2, \dots$) creates N polygons, V_i ($i = 1, 2, \dots$) (polyhedron in three dimensions). For example, in Fig. 8.8, ten points are equidistantly

Fig. 8.6 Voronoi diagram

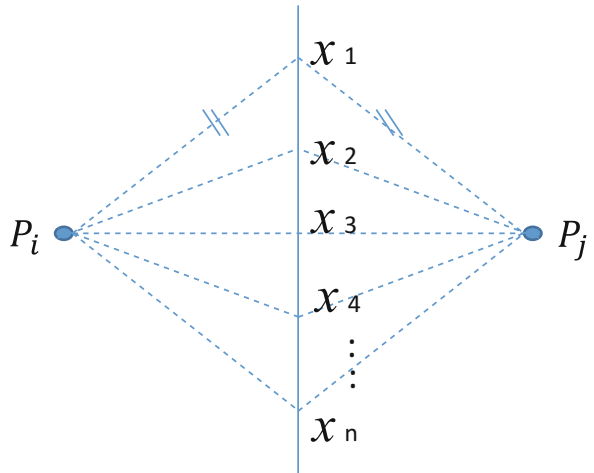


Fig. 8.7 Voronoi diagram (network)

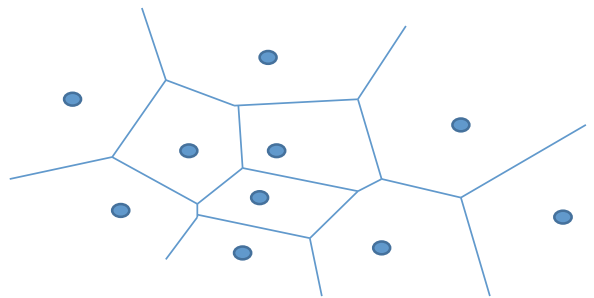
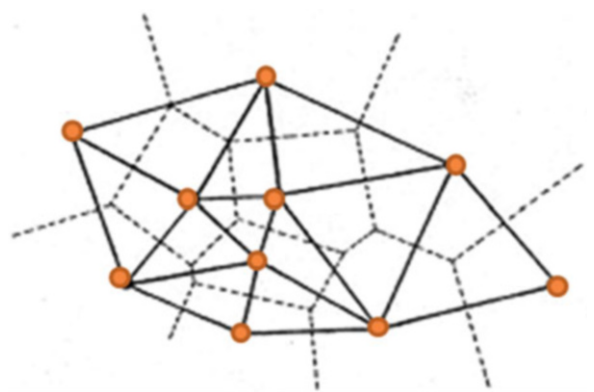


Fig. 8.8 Voronoi diagram (network)



divided by the Voronoi diagram that is composed of 10 polygons. Therefore, V_i is a convex polygon bounded by the lines bisecting perpendicularly the lines between P_i and its neighboring nodes.

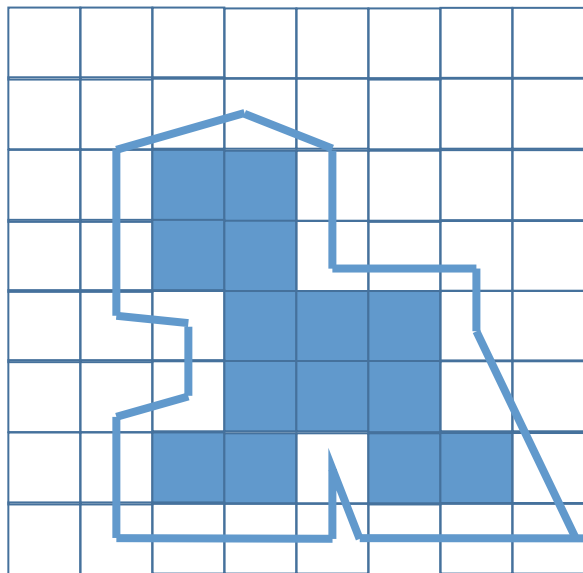
Once the Voronoi diagram is created, each mesh is registered by connecting neighboring nodes separated by V_i . That is, each side of the polygon in Voronoi diagram allows a connection between two neighboring points. This will prevent confusion as to which nodes have to be connected to each other to create a nonoverlapped mesh structure.

8.2.2.2 Grid-Based Approach

Another well-established method for automatic mesh generation is the Grid-Based approach, by which node generation step can be skipped. The basic motivation came from the fact that the grid, by its nature, seems similar to meshes (see Fig. 8.9). The actual shape of the grid is not the most favorable shape since there are two 45° corners in each triangle. Preferably, each corner has to be close to 60° for the best result. Nevertheless, a square grid is in a shape that contains two triangles so that it could provide an initial mesh structure. The initial mesh structure can be transformed into a more preferable structure via a postprocessing later.

One of the challenges is to adjust the mesh size around the object depending on shape complexity. The demand is due by optimal computational resource management. A delicate geometric change in general causes more stress concentration, resulting in failure more often compared to other open areas. To that end, the quad-tree representation technique is used in the Grid-Based approach. A quad-tree is a

Fig. 8.9 Similarity in grid and mesh



tree whose nodes either have no branch or have four branches. It is a technique conventionally used in computer graphics. Let's say we divide a picture into four sections. Those four sections are then further divided into four subsections. We continue this process, recursively dividing a square region by 4. We must impose a limit to the levels of division otherwise we could keep dividing the picture forever. Generally, the limit is imposed due to storage considerations or to limit processing time or due to the resolution of the output device.

In this section, we describe how such quad-tree can be used for FEM mesh generation process. The general procedure starts with overlapping a square grid on a given shape. Then the actual shape needs to be trimmed into a zigzag pattern for further processing. This may result in loss of accuracy. To gain a higher level of accuracy, one can increase the resolution of the grid.

The size of the Grid must be in agreement with the mesh resolution. One important aspect is that the size of the grid pattern must be in square, meaning the number of grids in the initial grid pattern along the vertical and horizontal direction must be equal. Quad-tree requires dividing a entire shape into four quadrants of same size recursively. Therefore, no matter what the shape of the object is, the initial grid pattern overlapping the given shape has to be in square shape. Once a grid is overlapped, then the total area again has to be divided into four square quadrants.

The up-side-down tree as shown in Fig. 8.10 has four branches from each node, thus the name, "quad-tree." Each quadrant has to be recursively subdivided until no partial object occupies the quadrant. Once a quadrant is either empty or full, then the node is declared as a complete node with a square shape at the quad-tree representation. The square is left empty if the quadrant is empty, while it is filled when the quadrant is full. Since the second quadrant in Fig. 8.10 is empty, no further subdivision is required, while all other quadrants still contain a partial object, thus represented as a circle in the tree structure. In the next step, the quad-tree is grown representing the result of the subdivision of each quadrant (see Fig. 8.11).

Finally, the object is divided down to the third level where no further subdivision can be made since all of the quadrants are either empty or full (Fig. 8.12). Given the

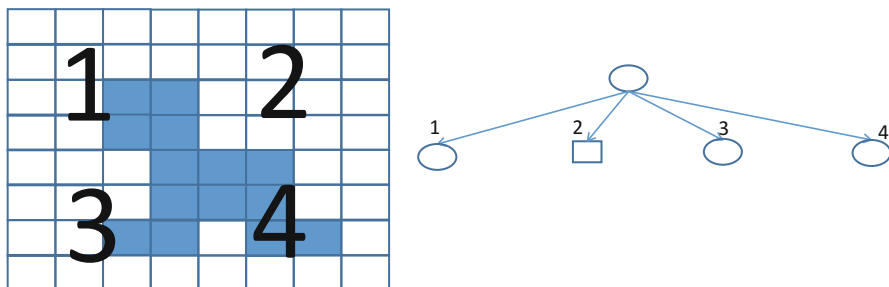


Fig. 8.10 Quad-tree representation: Step 1

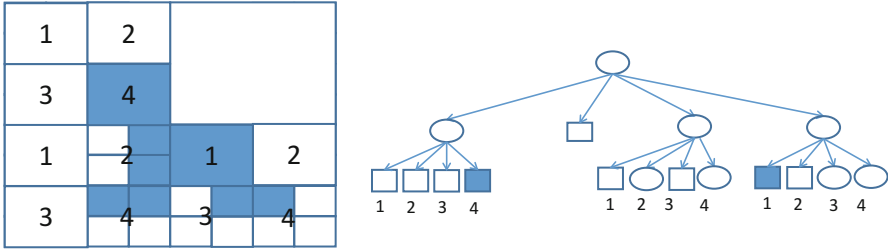


Fig. 8.11 Quad-tree representation: Step 2

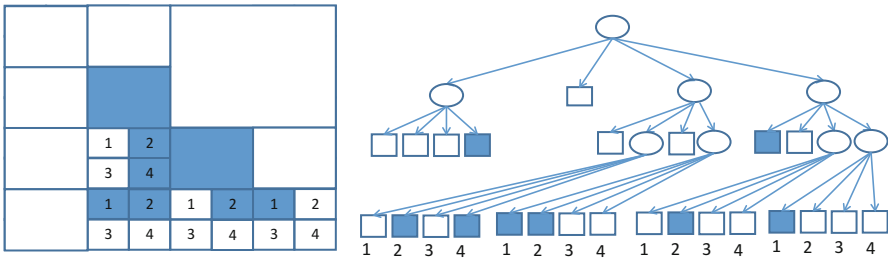


Fig. 8.12 Quad-tree representation: Step 3

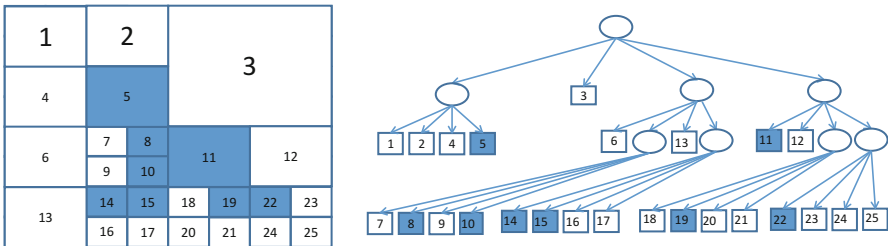


Fig. 8.13 Complete quad-tree representation

fully grown quad-tree, each square on the tree needs to be correlated to the grid pattern on the left. The result of the correlation is shown in Fig. 8.13.

Once the complete quad-tree representation is obtained, then individual grids in the complete grid structure will be transformed into triangles. For instance, the mesh structure of the given object in Fig. 8.14 is an example of triangular meshes.

One thing noticeable in the mesh structure generated by the grid-based approach is that there are more meshes around geometrically complex areas, while less meshes or larger triangles around less complicated areas. This allows us to optimize the use of computational resources, assuring optimal results with highest accuracy possible. Generally speaking, the stress level around a complex geometry is higher than that of less complex area due to the effect of stress concentration. Therefore the grid-based approach fits well with the purpose of mesh generation in FEM analysis.

Fig. 8.14 Mesh generation

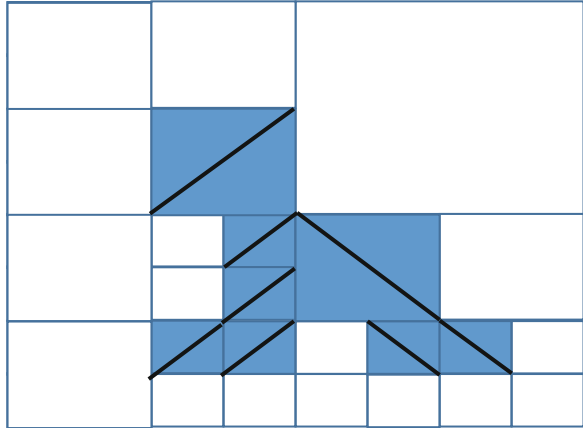
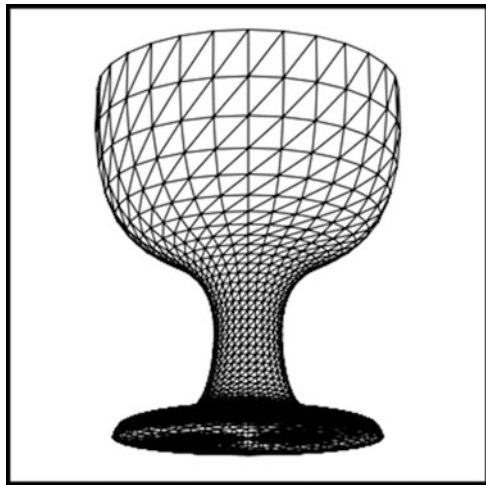


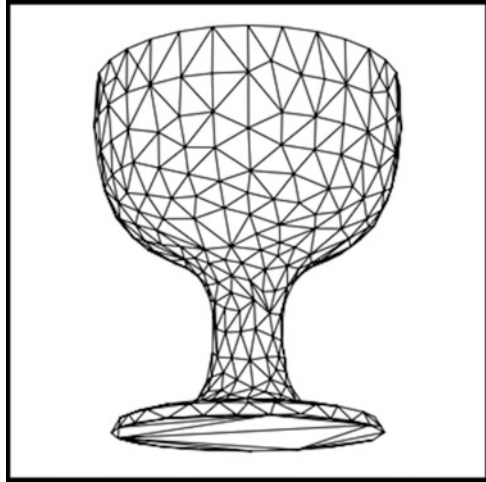
Fig. 8.15 Mesh generation by grid-based approach



8.2.3 Improvement of Mesh Quality

One can obtain reasonable results by automatic mesh generation methods introduced in the previous section. However, some methods do not generate good initial meshes. It is a common practice to use a post processing to obtain higher quality mesh after the initial mesh generation process. There are three common steps considered for the post processing. First, if the elements generated are not of the desired type, then further manual subdivision is required to obtain elements of the desired type. For instance, if a mesh in certain area is too large compared to the average size of the mesh, then further subdivision in such area will be necessary for reasonable overall accuracy. Second, if the elements do not have sizes compatible with the desired mesh density distribution, then further refinement will be necessary. Finally, if the elements obtained by automatic mesh generation process are not favorable as the example in Fig. 8.15, a mesh smoothing technique such as

Fig. 8.16 Mesh improvement by Laplacian smoothing [1]



Laplacian smoothing has to be applied (see Fig. 8.16). As shown in Fig. 8.16, the mesh smoothing process changes the shape of the triangle to more a desirable shape close to an equilateral triangle. This minimizes the sum of thin angles of all of the triangles in the mesh, which, in turn, will result in more stable and accurate stress evaluation.

What we obtained from the automatic mesh generation is a truss structure for a given solid object. In the next section, we cast a question as to how to apply equilibrium equations to find stress and strain level in a meshed truss structure.

8.3 What Is Truss?

As explained in the previous section, a rigid body is transformed into a cross-linked spring structure called “truss” for stress analysis. What is “truss” then? A truss is a structure comprising multiple triangular units constructed with straight members whose ends are assumed to be connected via a pin joint (see Fig. 8.17). Therefore, the reaction moment exerted on each joint is assumed to be minimal, thus no bending moment exists on each member. External forces and reactions to those forces are considered to act only at the nodes, resulting in forces at each joint which are either tensile or compressive aligned with member direction. Since the links in truss are considered to be pin jointed at each node, moment is disregarded in the force analysis of a truss structure. This assumption makes the force and displacement analysis simple and concise. A planar truss is a structure with all of the members and nodes lie within a two-dimensional plane, while a space truss has members and nodes expanding into a three-dimensional space.

Fig. 8.17 Truss structure



8.3.1 Matrix Approach in FEM

In this section, we introduce the matrix approach as a basis of the FEM to eventually be able to deal with a truss structure. As mentioned earlier, in a truss structure, all of the links are assumed to be connected via pin joint. As shown earlier in Fig. 8.2, since each member is assumed to be a spring element, once the force acting on each element is found, then, with the known stiffness (spring constant) of each element, the force–displacement relationship can be expressed by an algebraic equation.

$$F = k \cdot x \tag{8.1}$$

where F is the force acting on the element, k is the stiffness, and x is the resulting displacement of the member. Equation (8.1) is known as Hook’s law, widely accepted equation in static force analysis. Since there are multiple spring elements in the truss structure, FEM uses number of algebraic equations associated with each element. Instead of using numerous number of equations, however, matrix notation is used in FEM. Matrix notation represents entire group of force–displacement equations as one concise matrix equation to solve sets of simultaneous algebraic equations. For a given set of nodes $\{1, 2, 3, \dots, n\}$, we develop a matrix equation such as,

$$\{F\} = [k]\{x\} \tag{8.2}$$

where $\{F\}$ is the nodal force matrix, $[k]$ is the stiffness matrix, and $\{x\}$ is the resulting displacement matrix. The bracket symbol $\{\}$ is used for a column matrix, while $[\]$ is used for a square matrix in general. In this section, our primary focus is in the question as to how to develop (8.2) with multiple force displacement equations.

The first step toward that goal is to understand and setup a stiffness matrix for a spring element. In matrix notation, F_{1x}, F_{1y}, F_{1z} represent forces acting on the node 1 along x, y, z axes. Likewise, $d_{1x}, d_{1y},$ and d_{1z} represent displacements of the node 1 along x, y, z axes. If we collect them all in one column matrix, we obtain a complete matrix form of all of the nodes such as:

$$\{F\} = F = \begin{Bmatrix} F_{1x} \\ F_{1y} \\ F_{1z} \\ \vdots \\ F_{nx} \\ F_{ny} \\ F_{nz} \end{Bmatrix}, \quad \{d\} = d = \begin{Bmatrix} d_{1x} \\ d_{1y} \\ d_{1z} \\ \vdots \\ d_{nx} \\ d_{ny} \\ d_{nz} \end{Bmatrix} \tag{8.3}$$

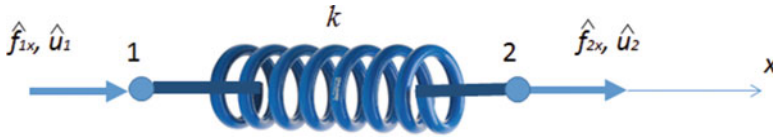


Fig. 8.18 Linear spring

Since $F = k \cdot x$, there must exist a k matrix to correlate the force column matrix and the displacement column matrix. By the nature of matrix multiplication, the k matrix will be a square matrix such as,

$$k = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1n} \\ k_{21} & k_{22} & \dots & k_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ k_{n1} & k_{n2} & \dots & k_{nn} \end{bmatrix} \quad (8.4)$$

We call the above matrix as a stiffness influence matrix. Therefore the force–displacement equation will be a matrix equation of a following form.

$$\begin{Bmatrix} F_{1x} \\ F_{1y} \\ F_{1z} \\ \vdots \\ F_{nx} \\ F_{ny} \\ F_{nz} \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1n} \\ k_{21} & k_{22} & \dots & k_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ k_{n1} & k_{n2} & \dots & k_{nn} \end{bmatrix} \begin{Bmatrix} d_{1x} \\ d_{1y} \\ d_{1z} \\ \vdots \\ d_{nx} \\ d_{ny} \\ d_{nz} \end{Bmatrix} \quad (8.5)$$

8.3.2 Force–Displacement Relationship

As the first step toward developing the above force–displacement equation, we start investigating a single linear spring member. As shown in Fig. 8.18, two nodes are assigned at both ends of the spring.

The local axis, x , is defined along the direction of the spring member and all of the local forces and local displacements are defined along the positive direction of the local axis. Forces and displacements are all assigned along the positive direction of the x axis. $\hat{f}_{1x}, \hat{f}_{2x}$ are local nodal forces, while $\hat{d}_{1x}, \hat{d}_{2x}$ are local nodal displacements that determine the degrees of freedom at each node. In order to determine the relationship between force and displacement, we first consider the force acting on the node number 1. Let us assume that the nodes 1 and 2 moved by \hat{d}_{1x} and \hat{d}_{2x} respectively by the forces, \hat{f}_{1x} and \hat{f}_{2x} (see Fig. 8.19). As shown in the free body diagram (FBD), $k_{11} \cdot \hat{d}_{1x}$ and $k_{12} \cdot \hat{d}_{2x}$ are the reaction forces at each node to balance the external force acting at the node 1.

Fig. 8.19 Free body diagram for f_{1x}

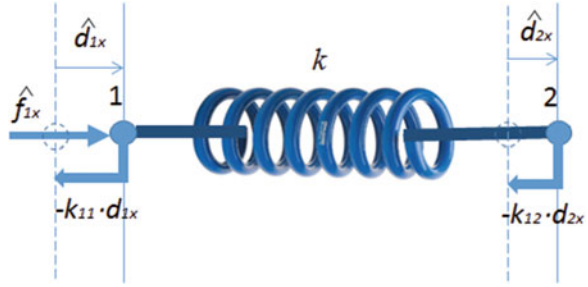
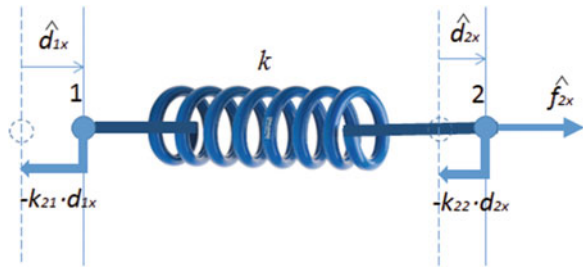


Fig. 8.20 Free body diagram for f_{2x}



The first index of k corresponds to the force acting at the first node, while the second index represents the node number. Therefore, the static force equilibrium equation between all of the forces in the FBD becomes as follows.

$$\hat{f}_{1x} - k_{11} \cdot \hat{d}_{1x} - k_{12} \cdot \hat{d}_{2x} = 0 \tag{8.6}$$

or

$$\hat{f}_{1x} = k_{11} \cdot \hat{d}_{1x} + k_{12} \cdot \hat{d}_{2x} \tag{8.7}$$

Likewise, the FBD by the force acting at node 2 is illustrated in Fig. 8.20. Then, the force balance equation by the force at node 2 becomes;

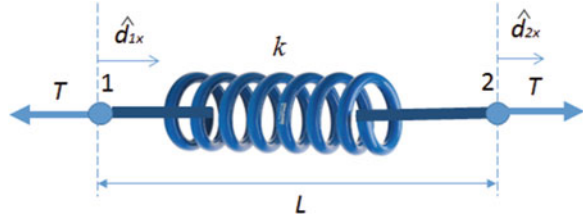
$$\hat{f}_{2x} = k_{21} \cdot \hat{d}_{1x} + k_{22} \cdot \hat{d}_{2x} \tag{8.8}$$

Now, if we combine two equations, (8.7) and (8.8), into a single matrix, then we obtain the following equation.

$$\begin{Bmatrix} \hat{f}_{1x} \\ \hat{f}_{2x} \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix} \tag{8.9}$$

Equation (8.9) is a general matrix equation that represents the relationship between force and displacement, which is applicable to any type of spring assemblage. We name the k matrix for a simple spring element above as a element stiffness matrix. In the following section we study about how to determine the stiffness matrix for a simple spring element when it is in static equilibrium state.

Fig. 8.21 Spring in static equilibrium



8.3.3 Stiffness Matrix for a Single Spring Element

In order to maintain a spring in a static equilibrium state, either tensile or compressive forces in equal magnitude have to be applied at both ends. With that in mind, we develop the element equation of a spring when it is in static equilibrium. First we consider a spring with tensile forces acting at both nodes with the same magnitude, but in opposite direction (see Fig. 8.21).

Below is a step-by-step approach to explain the entire procedure of obtaining the local stiffness matrix for a simple spring element in static equilibrium state.

Step 1: Select the element type.

We decide to use a linear spring element subjected to a tensile force at both ends.

Step 2: Select displacement function.

In order to express the absolute displacement, \hat{u} , of each node as a function of an arbitrary location, \hat{x} , we need to setup a relationship between two properties. Without knowing the exact closed form displacement at certain location beforehand, we assume that the relationship in the form of a polynomial equation, which is the most common approach, by the following equation.

$$\hat{u} = a_1 + a_2 \cdot \hat{x} \quad (8.10)$$

Equation (8.10) represents the displacement of the spring at an arbitrary location defined by x as a linear displacement along the local DOF. In matrix form, the above equation becomes;

$$\hat{u} = [1 \quad \hat{x}] \begin{Bmatrix} a_1 \\ a_2 \end{Bmatrix} \quad (8.11)$$

In order to obtain a_1 and a_2 , we use the boundary condition such as;

$$\begin{aligned} \hat{u}(0) &= \hat{d}_{1x} = a_1 \\ \hat{u}(L) &= \hat{d}_{2x} = a_2 \cdot L + \hat{d}_{1x} \end{aligned}$$

or

$$a_2 = \frac{\hat{d}_{2x} - \hat{d}_{1x}}{L}.$$

Therefore, (8.10) becomes,

$$\hat{u} = \hat{d}_{1x} + \frac{\hat{d}_{2x} - \hat{d}_{1x}}{L} \cdot \hat{x}$$

or

$$\hat{u} = \left(1 - \frac{\hat{x}}{L}\right) \hat{d}_{1x} + \frac{\hat{x}}{L} \hat{d}_{2x} \quad (8.12)$$

In matrix form,

$$\hat{u} = \left[1 - \frac{\hat{x}}{L} \quad \frac{\hat{x}}{L}\right] \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix}$$

or

$$\hat{u} = [N_1 \quad N_2] \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix} \quad (8.13)$$

where N_1 and N_2 are shape functions of displacement.

8.4 How to Develop Governing Equations?

Now we develop a governing equation for the force and displacement relationship based on the FBD in Fig. 8.21. First, the total deformation, δ , as a function of local displacement \hat{d}_{1x} and \hat{d}_{2x} can be defined such that:

$$\delta = \hat{d}_{2x} - \hat{d}_{1x} \quad (8.14)$$

Since the force is proportional to the total displacement by the spring stiffness, k ,

$$T = k \cdot \delta \quad (8.15)$$

or

$$T = k \cdot (\hat{d}_{2x} - \hat{d}_{1x}). \quad (8.16)$$

By the force and displacement relationship in (8.16), we can derive the element stiffness matrix. First of all, if we compare Figs. 8.18 and 8.21, the local forces acting at both nodes become,

$$\hat{f}_{1x} = -T, \quad \hat{f}_{2x} = T \quad (8.17)$$

By using (8.16) and (8.17),

$$T = -\hat{f}_{1x} = k \cdot (\hat{d}_{2x} - \hat{d}_{1x}) \quad (8.18)$$

$$T = \hat{f}_{2x} = k \cdot (\hat{d}_{2x} - \hat{d}_{1x}) \quad (8.19)$$

Therefore,

$$\hat{f}_{1x} = k \cdot (\hat{d}_{1x} - \hat{d}_{2x}) \quad (8.20)$$

$$\hat{f}_{2x} = k \cdot (\hat{d}_{2x} - \hat{d}_{1x}) \quad (8.21)$$

In matrix form,

$$\begin{Bmatrix} \hat{f}_{1x} \\ \hat{f}_{2x} \end{Bmatrix} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix} \quad (8.22)$$

Therefore, the stiffness matrix in (8.9) becomes;

$$\hat{k} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \quad (8.23)$$

where \hat{k} is defined as a local stiffness matrix for the element. In general case, there exist multiple spring elements in a truss structure. If we convert the local force, displacement, and stiffness equations into a global coordinate and assemble them all into a global equation, then, we obtain the following sets of equations.

$$\begin{aligned} \underline{F} = [F] &= \sum_{e=1}^N f^e \\ \underline{D} = [D] &= \sum_{e=1}^N d^e \\ \underline{K} = [K] &= \sum_{e=1}^N k^e \end{aligned}$$

If we assemble equations above, then we obtain the following equation.

$$F = K \cdot D \tag{8.24}$$

In addition, by multiplying the inverse of the stiffness matrix at both sides, we obtain the displacement for known stiffness and external force matrix acting on the truss structure.

$$D = K^{-1} \cdot F \tag{8.25}$$

The displacement matrix, D , will contain the amount of displacement of the entire spring elements in the truss structure. Finally, from the displacement matrix, we can find the stress in each spring element by the stress and strain relationship. In the next section, we will demonstrate overall FEM analysis process with an example of a simple spring assemblage.

8.5 Example of a Spring Assemblage

In this section, we expand our scope to a structure with multiple springs to demonstrate how the matrix method works in FEM analysis. Let us consider a spring assemblage with two spring elements connected in serial fashion (see Fig. 8.22)

Assuming all of the springs in the spring assemblage are linear spring (force increases proportional to the displacement), the force and displacement relationship for the first and the second springs can be rewritten respectively as:

$$\begin{Bmatrix} \hat{f}_{1x}^{(1)} \\ \hat{f}_{3x}^{(1)} \end{Bmatrix} = \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x}^{(1)} \\ \hat{d}_{3x}^{(1)} \end{Bmatrix} \tag{8.26}$$

$$\begin{Bmatrix} \hat{f}_{3x}^{(2)} \\ \hat{f}_{2x}^{(2)} \end{Bmatrix} = \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} \hat{d}_{3x}^{(2)} \\ \hat{d}_{2x}^{(2)} \end{Bmatrix} \tag{8.27}$$

where the number index on the upper right corner indicates the element number. By the compatibility requirements for the continuity, two springs remain connected before and after the deformation. Therefore the following condition must be met.

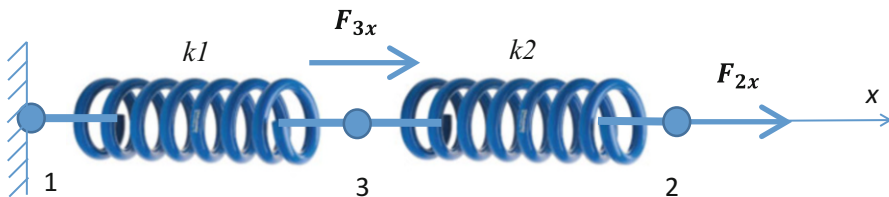


Fig. 8.22 Spring assemblage

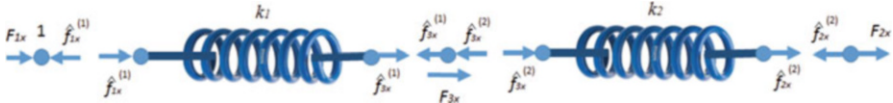


Fig. 8.23 Free body diagram

$$\hat{d}_{3x}^{(1)} = \hat{d}_{3x}^{(2)} = \hat{d}_{3x}$$

Now with the FBD of the spring assemblage, we find the governing equation for the force and displacement relationship. If we isolate all of the elements in the given system, we obtain the FBD as shown in Fig. 8.23.

Please note that each node has been isolated as well. If we apply the force equilibrium equation for each node, then we obtain the following three equations.

$$\sum F_1 = F_{1x} - \hat{f}_{1x}^{(1)} = 0 \quad (8.28)$$

$$\sum F_2 = F_{2x} - \hat{f}_{2x}^{(2)} = 0 \quad (8.29)$$

$$\sum F_3 = F_{3x} - \hat{f}_{3x}^{(1)} - \hat{f}_{3x}^{(2)} = 0 \quad (8.30)$$

If we use (8.26) and (8.27), then above three equations will turn into following.

$$F_{1x} = \hat{f}_{1x}^{(1)} = k_1 \cdot \hat{d}_{1x} - k_1 \cdot \hat{d}_{3x} \quad (8.31)$$

$$F_{2x} = \hat{f}_{2x}^{(2)} = -k_2 \cdot \hat{d}_{3x} + k_1 \cdot \hat{d}_{2x} \quad (8.32)$$

$$F_{3x} = \hat{f}_{3x}^{(1)} + \hat{f}_{3x}^{(2)} = -k_1 \cdot \hat{d}_{1x} + k_1 \cdot \hat{d}_{3x} + k_2 \cdot \hat{d}_{3x} - k_2 \cdot \hat{d}_{2x} \quad (8.33)$$

In matrix form,

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \end{Bmatrix} = \begin{bmatrix} k_1 & 0 & -k_1 \\ 0 & k_2 & -k_2 \\ -k_1 & -k_2 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \\ \hat{d}_{3x} \end{Bmatrix} \quad (8.34)$$

Therefore, the system stiffness matrix becomes as follows;

$$\underline{K} = \begin{bmatrix} k_1 & 0 & -k_1 \\ 0 & k_2 & -k_2 \\ -k_1 & -k_2 & k_1 + k_2 \end{bmatrix} \quad (8.35)$$

The system stiffness matrix has some unique properties such as,

1. K is symmetric
2. K is singular (symmetry w.r.t the diagonal). Therefore boundary condition is needed to solve for displacements.
3. The main diagonal terms of K are always positive.

8.6 Boundary Conditions

In order to solve for the unknown forces and unknown displacements in (8.34), known boundary conditions must be addressed and used, otherwise the problem is indeterministic, therefore, unsolvable. There are two types of boundary conditions: Homogeneous boundary condition and nonhomogeneous boundary condition. The homogeneous boundary condition applies when a node is fixed thus the displacement of the node is limited. Any means of complete prevention of the movement of a node will fall into this category. On the other hand, nonhomogeneous boundary condition applies when a finite nonzero value of displacement of a node is allowed.

8.6.1 Homogeneous Boundary Condition

In the previous example in Sect. 8.5, the first node is fixed on the wall, thus the homogeneous boundary condition applies on node 1. Since $\hat{d}_{1x} = 0$, (8.34) will become,

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \end{Bmatrix} = \begin{bmatrix} k_1 & 0 & -k_1 \\ 0 & k_2 & -k_2 \\ -k_1 & -k_2 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} 0 \\ \hat{d}_{2x} \\ \hat{d}_{3x} \end{Bmatrix}$$

Three unknowns are identified in the above equation. F_{1x} , \hat{d}_{2x} , and \hat{d}_{3x} . F_{1x} is the unknown reaction force at node 1 and \hat{d}_{2x} , \hat{d}_{3x} are unknown displacements at nodes 2 and 3 respectively. Since $\hat{d}_{1x} = 0$, if we remove the first row from the above equation, it becomes,

$$\begin{Bmatrix} F_{2x} \\ F_{3x} \end{Bmatrix} = \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} \hat{d}_{2x} \\ \hat{d}_{3x} \end{Bmatrix}$$

Please note that all of the variables on the left hand side of the equation are known, while the variables on the right hand side are unknown by eliminating the first row. Now by multiplying the inverse of the stiffness matrix, we obtain the following equation.

$$\begin{aligned} \begin{Bmatrix} \hat{d}_{2x} \\ \hat{d}_{3x} \end{Bmatrix} &= \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_1 + k_2 \end{bmatrix}^{-1} \begin{Bmatrix} F_{2x} \\ F_{3x} \end{Bmatrix} \\ &= \begin{bmatrix} \frac{1}{k_1} + \frac{1}{k_2} & \frac{1}{k_1} \\ \frac{1}{k_1} & \frac{1}{k_1} \end{bmatrix} \begin{Bmatrix} F_{2x} \\ F_{3x} \end{Bmatrix} \end{aligned}$$

Therefore, displacements at nodes 2 and 3 can be found as below.

$$\hat{d}_{2x} = \left(\frac{1}{k_1} + \frac{1}{k_2} \right) \cdot F_{2x} + \frac{1}{k_1} \cdot F_{3x}$$

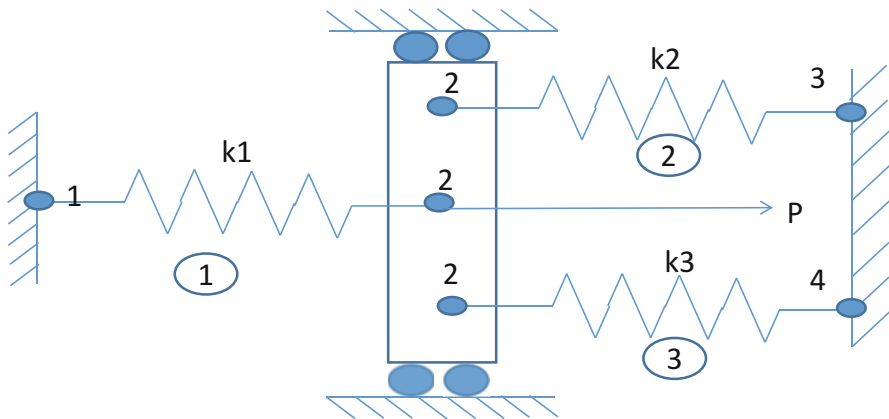
$$\hat{d}_{3x} = \frac{1}{k_1} \cdot F_{2x} + \frac{1}{k_1} \cdot F_{3x}$$

Finally, from (8.34), the unknown reaction force at node 1 will be:

$$F_{1x} = -k_1 \left(\frac{1}{k_1} \cdot F_{2x} + \frac{1}{k_1} \cdot F_{3x} \right)$$

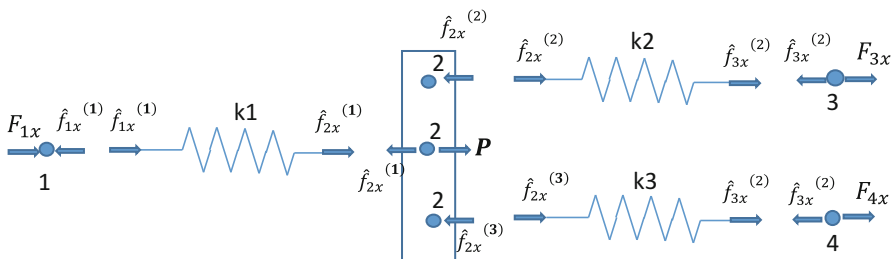
Sample Problem 8.1

For the spring assemblage below, find the displacement of the node no. 2.



Solution

First we draw a F.B.D.



The nodal force equilibrium conditions are,

$$\begin{aligned}\sum F_1 &= F_{1x} - \hat{f}_{1x}^{(1)} = 0 \\ \sum F_2 &= P - \hat{f}_{2x}^{(1)} - \hat{f}_{2x}^{(2)} - \hat{f}_{2x}^{(3)} = 0 \\ \sum F_3 &= F_{3x} - \hat{f}_{3x}^{(2)} = 0 \\ \sum F_4 &= F_{4x} - \hat{f}_{4x}^{(3)} = 0\end{aligned}$$

Now the compatibility condition at node 2 is,

$$\hat{d}_{2x}^{(1)} = \hat{d}_{2x}^{(2)} = \hat{d}_{2x}^{(3)} = \hat{d}_{2x}$$

Using the local stiffness matrix, we obtain following three matrix equations for the force and displacement relationship.

$$\begin{aligned}\begin{Bmatrix} \hat{f}_{1x}^{(1)} \\ \hat{f}_{2x}^{(1)} \end{Bmatrix} &= \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x}^{(1)} \\ \hat{d}_{2x}^{(1)} \end{Bmatrix} \\ \begin{Bmatrix} \hat{f}_{2x}^{(2)} \\ \hat{f}_{3x}^{(2)} \end{Bmatrix} &= \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} \hat{d}_{2x}^{(2)} \\ \hat{d}_{3x}^{(2)} \end{Bmatrix} \\ \begin{Bmatrix} \hat{f}_{2x}^{(3)} \\ \hat{f}_{4x}^{(3)} \end{Bmatrix} &= \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} \hat{d}_{2x}^{(3)} \\ \hat{d}_{4x}^{(3)} \end{Bmatrix}\end{aligned}$$

If we substitute above three equations to the force–displacement equilibrium equations, then we obtain the total or global equilibrium equations such as

$$\begin{aligned}F_{1x} &= k_1 \hat{d}_{1x} - k_1 \hat{d}_{2x} \\ P &= -k_1 \hat{d}_{1x} + k_1 \hat{d}_{2x} + k_2 \hat{d}_{2x} - k_2 \hat{d}_{3x} + k_3 \hat{d}_{2x} - k_3 \hat{d}_{4x} = 0 \\ F_{3x} &= -k_2 \hat{d}_{2x} + k_2 \hat{d}_{3x} \\ F_{4x} &= -k_3 \hat{d}_{2x} + k_3 \hat{d}_{4x}\end{aligned}$$

In matrix form,

$$\begin{Bmatrix} F_{1x} \\ P \\ F_{3x} \\ F_{4x} \end{Bmatrix} = \begin{bmatrix} k_1 & -k_1 & 0 & 0 \\ -k_1 & k_1 + k_2 + k_3 & -k_2 & -k_3 \\ 0 & -k_2 & k_2 & 0 \\ 0 & -k_3 & 0 & +k_3 \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \\ \hat{d}_{3x} \\ \hat{d}_{4x} \end{Bmatrix}$$

The boundary conditions are

$$\hat{d}_{1x} = \hat{d}_{3x} = \hat{d}_{4x} = 0$$

Therefore, if we remove the first, third, and fourth row from the above equation, then we obtain

$$\{P\} = [k_1 + k_2 + k_3] \{\hat{d}_{2x}\}$$

Again, all of the variables on the left hand side of the equation are known, while the unknown variables are on the right hand side. The above equation becomes,

$$P = (k_1 + k_2 + k_3) \cdot \hat{d}_{2x}$$

Therefore, the displacement of node no. 2 is

$$\hat{d}_{2x} = \frac{P}{k_1 + k_2 + k_3}$$

Now by using the force equilibrium equation we can obtain the unknown reaction forces at nodes 1, 3, and 4 such as,

$$F_{1x} = -k_1 \frac{P}{k_1 + k_2 + k_3}$$

$$F_{3x} = -k_2 \frac{P}{k_1 + k_2 + k_3}$$

$$F_{4x} = -k_3 \frac{P}{k_1 + k_2 + k_3}$$

8.6.2 Nonhomogeneous Boundary Condition

Nonhomogeneous boundary condition applies where a finite nonzero displacement is allowed at a node. Suppose a spring assemblage in the following figure, where a finite nonzero displacement, δ , is allowed (Fig. 8.24).

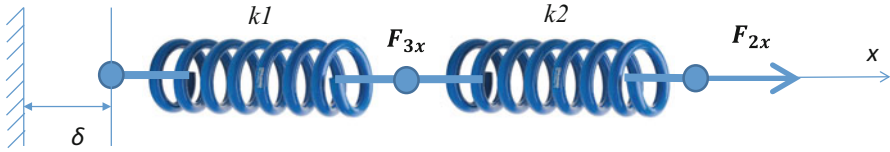


Fig. 8.24 Free body diagram

Therefore, $\hat{d}_{1x} = \delta$. Then the force displacement equation becomes,

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \end{Bmatrix} = \begin{bmatrix} k_1 & 0 & -k_1 \\ 0 & k_2 & -k_2 \\ -k_1 & -k_2 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} \delta \\ \hat{d}_{2x} \\ \hat{d}_{3x} \end{Bmatrix} \tag{8.36}$$

Since F_{1x} is unknown, we eliminate the first row from the above matrix equation. Then we obtain,

$$\begin{aligned} F_{2x} &= k_2 \hat{d}_{2x} - k_2 \hat{d}_{3x} \\ F_{3x} &= -k_1 \delta - k_2 \hat{d}_{2x} + (k_1 + k_2) \hat{d}_{3x} \end{aligned}$$

Arranging the above equation with the known variables on the left and all of the unknown variables on the right side, we obtain,

$$\begin{aligned} F_{2x} &= k_2 \hat{d}_{2x} - k_2 \hat{d}_{3x} \\ F_{3x} + k_1 \delta &= -k_2 \hat{d}_{2x} + (k_1 + k_2) \hat{d}_{3x} \end{aligned}$$

In matrix form,

$$\begin{Bmatrix} F_{2x} \\ F_{3x} + k_1 \delta \end{Bmatrix} = \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} \hat{d}_{2x} \\ \hat{d}_{3x} \end{Bmatrix} \tag{8.37}$$

Now with (8.37), the rest of problem solving steps are the same as the homogeneous boundary condition case. Therefore, for a nonhomogeneous boundary condition, in general, we need to merge the terms associated with known displacements to known forces.

8.7 Assembling the Total Stiffness Matrix by Superposition (Direct Stiffness Method)

In the previous section, we studied the standard procedure to obtain the system stiffness matrix by following a step-by-step procedure. Although the system stiffness matrix can vary depending on the configuration of the truss structure, the local

stiffness matrix in (8.23) never changes. By using this idea of consistency in local stiffness matrix, the global stiffness matrix can be assembled directly from the definition of the local stiffness matrices. Since we can construct the global stiffness matrix directly from the collection of local stiffness matrices, this method is known as a direct stiffness method. For instance, if we use the same example in Sect. 8.4, the global stiffness matrix can be constructed directly by the superposition method. First we make a frame of the global stiffness matrix consistent to the number of nodes. Since we have three nodes, the frame of the global stiffness matrix becomes as follows;

$$\underline{K} = \begin{array}{ccc|c} \mathbf{1} & \mathbf{2} & \mathbf{3} & \\ \hline a & b & c & \mathbf{1} \\ d & e & f & \mathbf{2} \\ g & h & i & \mathbf{3} \end{array} \quad (8.38)$$

The numbers around the matrix indicate node numbers. From (8.26) and (8.27), the local stiffness matrix for each node becomes as follows;

$$\underline{k}^{(1)} = \begin{array}{cc|c} \mathbf{1} & \mathbf{3} & \\ \hline k_1 & -k_1 & \mathbf{1} \\ -k_1 & k_1 & \mathbf{3} \end{array} \quad (8.39)$$

In expended form,

$$\underline{K}^{(1)} = \begin{array}{ccc|c} \mathbf{1} & \mathbf{2} & \mathbf{3} & \\ \hline k_1 & \mathbf{0} & -k_1 & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{2} \\ -k_1 & \mathbf{0} & k_1 & \mathbf{3} \end{array} \quad (8.40)$$

or

$$\begin{Bmatrix} \hat{f}_{1x}^{(1)} \\ \hat{f}_{2x}^{(1)} \\ \hat{f}_{3x}^{(1)} \end{Bmatrix} = \begin{bmatrix} k_1 & \mathbf{0} & -k_1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -k_1 & \mathbf{0} & k_1 \end{bmatrix} \cdot \begin{Bmatrix} \hat{d}_{1x}^{(1)} \\ \hat{d}_{2x}^{(1)} \\ \hat{d}_{3x}^{(1)} \end{Bmatrix} \quad (8.41)$$

Likewise, for the node number 2,

$$\underline{k}^{(1)} = \begin{array}{cc|c} \mathbf{3} & \mathbf{2} & \\ \hline k_2 & -k_2 & \mathbf{3} \\ -k_2 & k_2 & \mathbf{2} \end{array} \quad (8.42)$$

In expended form,

$$\underline{K}^{(2)} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & k_2 & -k_2 \\ \mathbf{0} & -k_2 & k_2 \end{bmatrix} \begin{matrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \end{matrix} \tag{8.43}$$

or

$$\begin{Bmatrix} \hat{f}_{1x}^{(2)} \\ \hat{f}_{2x}^{(2)} \\ \hat{f}_{3x}^{(2)} \end{Bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & k_2 & -k_2 \\ \mathbf{0} & -k_2 & k_2 \end{bmatrix} \cdot \begin{Bmatrix} \hat{d}_{1x}^{(2)} \\ \hat{d}_{2x}^{(2)} \\ \hat{d}_{3x}^{(2)} \end{Bmatrix} \tag{8.44}$$

Due to the force equilibrium between external and internal forces, the relationship between global and local forces becomes as follows.

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \end{Bmatrix} = \begin{Bmatrix} \hat{f}_{1x}^{(1)} \\ \hat{f}_{2x}^{(1)} \\ \hat{f}_{3x}^{(1)} \end{Bmatrix} + \begin{Bmatrix} \hat{f}_{1x}^{(2)} \\ \hat{f}_{2x}^{(2)} \\ \hat{f}_{3x}^{(2)} \end{Bmatrix} \tag{8.45}$$

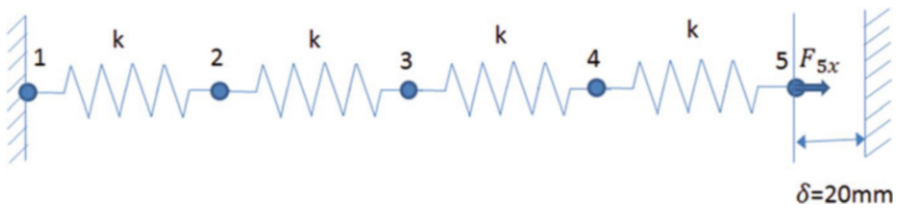
Therefore,

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \end{Bmatrix} = \begin{bmatrix} k_1 & \mathbf{0} & -k_1 \\ \mathbf{0} & k_2 & -k_2 \\ -k_1 & -k_2 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} d_{1x} \\ d_{2x} \\ d_{3x} \end{Bmatrix} \tag{8.46}$$

where d_{1x} , d_{2x} , and d_{3x} are the displacement in global coordinates. The global stiffness matrix in (8.44) can also be obtained by simply adding two matrices in (8.39) and (8.42).

Sample Problem 8.2

For the given spring assemblage shown below, find the force at node 5 where a finite displacement of 20 mm is allowed.



$k = 200 \text{ kn/m}$

Solution

Since the spring constant is identical for all of the springs, the local stiffness matrices are

$$\underline{k}^{(1)} = \begin{bmatrix} \mathbf{1} & \mathbf{2} \\ \mathbf{200} & -\mathbf{200} \\ -\mathbf{200} & \mathbf{200} \end{bmatrix} \quad \mathbf{1} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \\ \mathbf{200} & -\mathbf{200} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{200} & \mathbf{200} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \begin{matrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \end{matrix}$$

$$\underline{k}^{(2)} = \begin{bmatrix} \mathbf{2} & \mathbf{3} \\ \mathbf{200} & -\mathbf{200} \\ -\mathbf{200} & \mathbf{200} \end{bmatrix} \quad \mathbf{2} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{200} & -\mathbf{200} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{200} & \mathbf{200} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \begin{matrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \end{matrix}$$

$$\underline{k}^{(3)} = \begin{bmatrix} \mathbf{3} & \mathbf{4} \\ \mathbf{200} & -\mathbf{200} \\ -\mathbf{200} & \mathbf{200} \end{bmatrix} \quad \mathbf{3} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{200} & -\mathbf{200} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{200} & \mathbf{200} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \begin{matrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \end{matrix}$$

$$\underline{k}^{(3)} = \begin{bmatrix} \mathbf{4} & \mathbf{5} \\ \mathbf{200} & -\mathbf{200} \\ -\mathbf{200} & \mathbf{200} \end{bmatrix} \quad \mathbf{4} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{200} & -\mathbf{200} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{200} & \mathbf{200} \end{bmatrix} \quad \begin{matrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \end{matrix}$$

By using the direct stiffness method, the global stiffness matrix can be assembled as

$$\mathbf{K} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \\ \mathbf{200} & -\mathbf{200} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{200} & \mathbf{400} & -\mathbf{200} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{200} & \mathbf{400} & -\mathbf{200} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{200} & \mathbf{400} & -\mathbf{200} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{200} & \mathbf{200} \end{bmatrix} \quad \begin{matrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \end{matrix}$$

Therefore, the force–displacement relationship is

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \\ F_{4x} \\ F_{5x} \end{Bmatrix} = \begin{bmatrix} 200 & -200 & 0 & 0 & 0 \\ -200 & 400 & -200 & 0 & 0 \\ 0 & -200 & 400 & -200 & 0 \\ 0 & 0 & -200 & 400 & -200 \\ 0 & 0 & 0 & -200 & 200 \end{bmatrix} \begin{Bmatrix} d_{1x} \\ d_{2x} \\ d_{3x} \\ d_{4x} \\ d_{5x} \end{Bmatrix}$$

If we apply the boundary conditions, $\hat{d}_{1x} = 0$, $\hat{d}_{5x} = 20\text{mm}$, $F_{2x} = F_{3x} = F_{4x} = 0$, and if we transform the terms associated with known displacements to known forces, we obtain following equations.

$$\begin{aligned} F_{1x} &= -200\hat{d}_{2x} \\ 0 &= 400\hat{d}_{2x} - 200\hat{d}_{3x} \\ 0 &= -200\hat{d}_{2x} + 400\hat{d}_{3x} \\ 200\delta &= 200\hat{d}_{3x} + 400\hat{d}_{4x} \\ F_{5x} - 200\delta &= -200\hat{d}_{4x} \end{aligned}$$

Since F_{1x} and F_{5x} are unknown reaction forces, we eliminate the first and fifth equations from the above equation and rewrite it in a matrix form such as,

$$\begin{Bmatrix} 0 \\ 0 \\ 200 \times 20 \end{Bmatrix} = \begin{bmatrix} 400 & -200 & 0 \\ -200 & 400 & -200 \\ 0 & -200 & 400 \end{bmatrix} \begin{Bmatrix} d_{2x} \\ d_{3x} \\ d_{4x} \end{Bmatrix}$$

By multiplying the inverse of the stiffness matrix at both sides, we obtain the following equation.

$$\begin{Bmatrix} d_{2x} \\ d_{3x} \\ d_{4x} \end{Bmatrix} = \begin{bmatrix} 0.0038 & 0.0025 & 0.0013 \\ 0.0025 & 0.005 & 0.0025 \\ 0.0013 & 0.0025 & 0.0038 \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 4000 \end{Bmatrix}$$

Therefore, $d_{2x} = 0.005\text{ m}$, $d_{3x} = 0.01\text{ m}$, and $d_{4x} = 0.015\text{ m}$.

Unknown reaction forces can be calculated by the force–displacement equation such as

$$\begin{aligned} F_{1x} &= -200\hat{d}_{2x} = -1.0\text{ kN} \\ F_{5x} &= -200\hat{d}_{4x} + 200\delta = 1.0\text{ kN} \end{aligned}$$

8.8 Development of Truss Equation

In this section, we expand analysis scope to truss structure. Several engineering aspects have to be considered and merged with the analysis method of a spring assemblage. We first study how to derive the stiffness matrix for a truss and develop the basic concept to deal with general truss structures.

8.8.1 Derivation of the Stiffness Matrix for a Bar

Let us consider a mechanical bar shown in Fig. 8.25 to derive stiffness matrix of a truss structure.

\hat{u} in the figure represents axial displacement. By Hooks' law, the stress–strain relationship in a truss member becomes:

$$\sigma_x = E \cdot \epsilon_x \tag{8.47}$$

and

$$\epsilon_x = \frac{d\hat{u}}{dx} \tag{8.48}$$

From the force equilibrium,

$$A \cdot \sigma_x = T = \text{constant} \tag{8.49}$$

where A is the cross-sectional area of the bar. From (8.47)–(8.49),

$$\frac{d}{dx}(A \cdot \sigma_x) = \frac{d}{dx}\left(AE \cdot \frac{d\hat{u}}{dx}\right) = 0 \tag{8.50}$$

In order to develop a stiffness matrix of a bar member, we need following assumptions.

1. The bar cannot sustain shear force. Therefore, $\hat{f}_{1y} = \hat{f}_{2y} = 0$.

This assumption is realistic in that most of the bar members in a truss structure is connected by a pin joint. Even if they are not connected by a pin joint, this

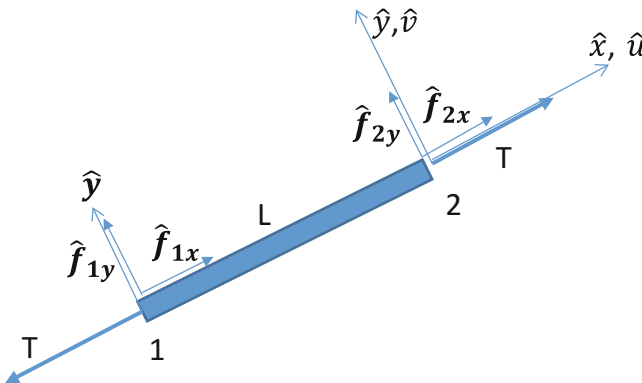


Fig. 8.25 A simple bar member of a truss

assumption is valid since the main resistance by a truss member is by either compressive or tensile stress.

2. The effect of transverse displacement is ignored, therefore $\hat{v} = 0$.

Although some degree of transverse displacement is expected, in general, the degree of transverse displacement is very small compared to axial displacement.

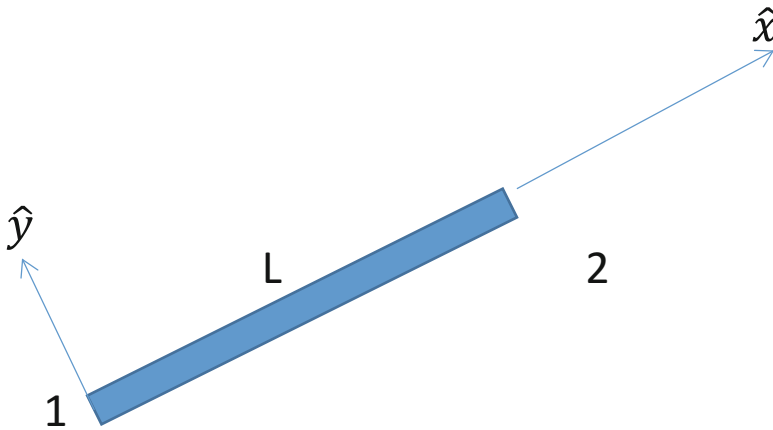
3. Hook's law in (8.47) applies to all of the members.
4. There is no intermediate applied load for all of the members.

This assumption is somehow related to the first assumption. If there is any intermediate load applied, there will be shear force and bending moment in the bar, thus the first assumption cannot be satisfied.

With all of the aforementioned assumptions, now we develop a stiffness matrix for a simple bar member by a step-by-step approach as detailed below.

Step 1: Select an element type.

We select a linear bar element whereby Hook's law is applicable as shown below



Step 2: Select a displacement function.

As we studied in the spring element, we select a linear displacement function such as,

$$\hat{u} = a_1 + a_2 \cdot \hat{x} \tag{8.51}$$

The linear displacement function is an approximation of the manner of a bar member deforms along axial direction. The constant, a_1 , represents the movement of the point 1, while a_2 indicates displacement along axial direction proportional to the relative location from the node 1.

Using absolute nodal displacements, \hat{d}_{1x} and \hat{d}_{2x} , the displacement function becomes,

$$\hat{u} = \left(\frac{\hat{d}_{2x} - \hat{d}_{1x}}{L} \right) \cdot \hat{x} + \hat{d}_{1x}$$

or

$$\hat{u} = \left(1 - \frac{\hat{x}}{L} \right) \hat{d}_{1x} + \frac{\hat{x}}{L} \hat{d}_{2x}$$

In matrix form,

$$\hat{u} = [N_1 \quad N_2] \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix}$$

where $N_1 = 1 - \frac{\hat{x}}{L}$, and $N_2 = \frac{\hat{x}}{L}$. N_1 and N_2 are often called shape function.

Step 3: Define the strain/displacement and stress/strain relationship. Using absolute displacement at each node, strain can be defined as,

$$\epsilon_x = \frac{d\hat{u}}{dx} = \frac{\hat{d}_{2x} - \hat{d}_{1x}}{L}$$

As mentioned earlier in step 1, we select a bar member that follows Hook's law. Therefore,

$$\sigma_x = E \cdot \epsilon_x = E \cdot \frac{\hat{d}_{2x} - \hat{d}_{1x}}{L}$$

Step 4: Derive element stiffness matrix and equation.

Since T (tension) is equal to $A \cdot \sigma_x$,

$$T = AE \cdot \epsilon_x = AE \cdot \frac{\hat{d}_{2x} - \hat{d}_{1x}}{L}$$

From Fig. 8.25, nodal forces are defined as,

$$\begin{aligned} \hat{f}_{1x} &= -T = \frac{AE}{L} \cdot (\hat{d}_{1x} - \hat{d}_{2x}), \\ \hat{f}_{2x} &= T = \frac{AE}{L} \cdot (\hat{d}_{2x} - \hat{d}_{1x}). \end{aligned}$$

In matrix form,

$$\begin{Bmatrix} \hat{f}_{1x} \\ \hat{f}_{2x} \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} \mathbf{1} & -\mathbf{1} \\ -\mathbf{1} & \mathbf{1} \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix} \quad (8.52)$$

Since the above matrix equation must satisfy force–displacement equation, stiffness matrix for the bar member should be;

$$\mathbf{k} = \frac{AE}{L} \begin{bmatrix} \mathbf{1} & -\mathbf{1} \\ -\mathbf{1} & \mathbf{1} \end{bmatrix} \quad (8.53)$$

Step 5: Assemble element equations to obtain global or total equation.

Once we obtain stiffness matrix for each bar member of a truss structure, then we can assemble all of stiffness equations to obtain the total stiffness equation either by conventional force equilibrium principle or by direct stiffness method. That is,

$$\begin{aligned} \underline{F} = [\mathbf{F}] &= \sum_{e=1}^N \mathbf{f}^e \\ \underline{K} = [\mathbf{K}] &= \sum_{e=1}^N \mathbf{k}^e \end{aligned}$$

Step 6: Solve for the nodal displacements.

In order to solve for nodal displacements, we need to implement boundary conditions. Appropriate boundary conditions for displacement and force have to be applied to obtain nodal displacements.

Step 7: Solve for element forces.

Once nodal displacements are all obtained, then we can estimate the strain of each element by the following equation.

$$\epsilon_x = \frac{\hat{d}_{2x} - \hat{d}_{1x}}{L}$$

As mentioned earlier, stress is linearly proportional to strain by young's modulus as long as a member is within elastic limit. Therefore, the stress exerted on each member can be found by the following equation.

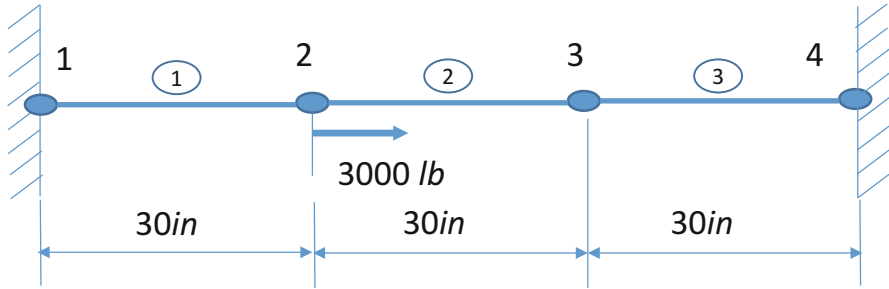
$$\sigma_x = E \cdot \epsilon_x$$

Sample Problem 8.3

For the system of bar elements shown in the figure below,

- Determine global stiffness matrix
- Find nodal displacements
- Calculate stress developed on each bar element

Let $E = 30 \times 10^6$ psi and $A = 1 \text{ in.}^2$ for elements 1 and 2, and let $E = 15 \times 10^6$ psi and $A = 2 \text{ in.}^2$ for element 3. Nodes 1 and 4 are fixed.



Solution

(a) First we find the local stiffness matrix for each bar member such as,

$$k^{(1)} = k^{(2)} = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \frac{1 \times 30 \times 10^6}{30} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = 10^6 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$k^{(3)} = \frac{2 \times 15 \times 10^6}{30} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = 10^6 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

By using the direct stiffness method, the global stiffness matrix can be assembled such as,

$$K = 10^6 \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1+1 & -1 & 0 \\ 0 & -1 & 1+1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Therefore, the force–displacement relationship is

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \\ F_{4x} \end{Bmatrix} = 10^6 \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \\ \hat{d}_{3x} \\ \hat{d}_{4x} \end{Bmatrix}$$

If we invoke boundary conditions at nodes 1 and 4,

$$\hat{d}_{1x} = \hat{d}_{4x} = 0$$

So if we substitute zero for \hat{d}_{1x} and \hat{d}_{4x} , and remove two unknown forces, F_{1x} and F_{4x} from the left hand side, then the above equation becomes,

$$\begin{Bmatrix} 3000 \\ 0 \end{Bmatrix} = 10^6 \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{Bmatrix} \hat{d}_{2x} \\ \hat{d}_{3x} \end{Bmatrix}$$

or

$$\begin{aligned} \begin{Bmatrix} \hat{d}_{2x} \\ \hat{d}_{3x} \end{Bmatrix} &= \frac{1}{10^6} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}^{-1} \begin{Bmatrix} 3000 \\ 0 \end{Bmatrix} \\ &= \frac{1}{3 \times 10^6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{Bmatrix} 3000 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 0.002 \\ 0.001 \end{Bmatrix} \end{aligned}$$

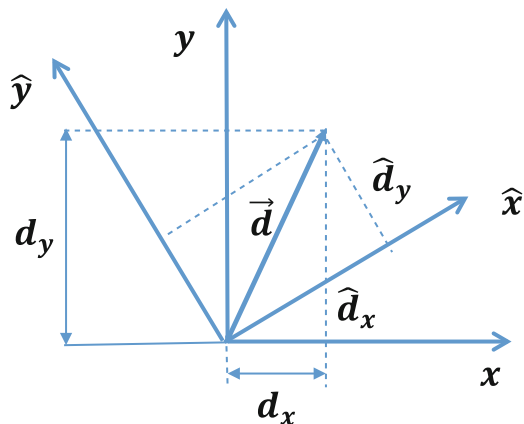
Now with the displacements, unknown external forces can be found as,

$$\begin{aligned} F_{1x} &= 10^6 (\hat{d}_{1x} - \hat{d}_{2x}) = -2000 \text{ lb} \\ F_{4x} &= 10^6 (-\hat{d}_{3x} + \hat{d}_{4x}) = -1000 \text{ lb} \end{aligned}$$

8.8.2 Transformation of Vectors in Two Dimensional Space

As shown in the previous section, force–displacement analysis on a bar member becomes feasible by using the analogy of a spring assemblage. However, for the complete analysis of a truss structure, there are other concerns that need to be addressed. One important aspect of truss structure analysis is that there are bar members slanted with arbitrary angle from horizontal line. In order to deal with all of the members slanted in a truss structure, the coordinate transformation has to be used and applied for analysis. Let us consider a point at d_x, d_y by x, y coordinate system. The same point can be also expressed by another coordinate system, \hat{x}, \hat{y} by \hat{d}_x and \hat{d}_y (Fig. 8.26).

Fig. 8.26 A simple bar member of a truss



From the figure above, if we define a vector \vec{d} that indicates the point, then \vec{d} can be expressed as,

$$\vec{d} = d_x \vec{i} + d_y \vec{j} = \hat{d}_x \hat{i} + \hat{d}_y \hat{j} \quad (8.54)$$

If we express \hat{d}_x and \hat{d}_y by d_x and d_y , then we obtain following equations.

$$\begin{aligned} \hat{d}_x &= d_x \cos \theta + d_y \sin \theta \\ \hat{d}_y &= d_y \cos \theta - d_x \sin \theta \end{aligned}$$

In matrix form,

$$\begin{Bmatrix} \hat{d}_x \\ \hat{d}_y \end{Bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{Bmatrix} d_x \\ d_y \end{Bmatrix}$$

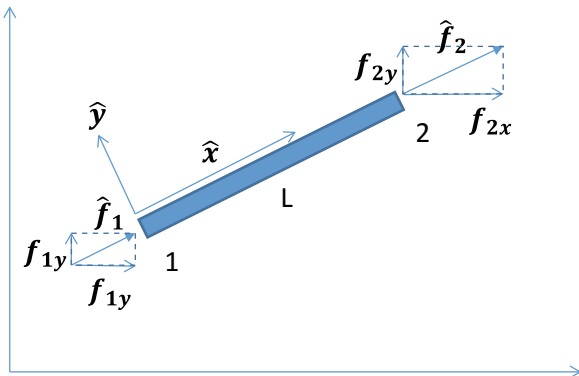
By using the above matrix equation, global stiffness matrix for a slanted bar member can be easily described.

8.8.3 Global Stiffness Matrix

Now with the coordinate transformation matrix, we derive global stiffness matrix for a slanted bar member. First, we relate global element nodal forces to global nodal displacements for a bar element arbitrarily oriented. That is, for a slanted bar, we divide nodal forces and displacements into the components in a global coordinate as shown in Fig. 8.27.

To that end, we start our derivation with the local force–displacement relationship such as,

Fig. 8.27 Local force decomposition



$$\hat{f} = \hat{k} \cdot \hat{d} \quad (8.55)$$

Up until now, a local coordinate is always defined along the main member direction. We develop a local force–displacement relationship that is applicable to an arbitrarily sloped bar member. The goal is to replace local forces and displacements with global forces and displacements to obtain stiffness matrix in a global coordinate. First of all, from the geometric relationship shown in Fig. 8.26, we relate local displacements to global displacements such as,

$$\begin{aligned} \hat{d}_{1x} &= d_{1x} \cos \theta + d_{1y} \sin \theta \\ \hat{d}_{2x} &= d_{2x} \cos \theta + d_{2y} \sin \theta \end{aligned}$$

In matrix form,

$$\begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix} = \begin{bmatrix} c & s & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & c & s \end{bmatrix} \begin{Bmatrix} d_{1x} \\ d_{1y} \\ d_{2x} \\ d_{2y} \end{Bmatrix} \quad (8.56)$$

or

$$\hat{d} = T \cdot d \quad (8.57)$$

Similarly,

$$\hat{f} = T \cdot f \quad (8.58)$$

If we substitute (8.57) to (8.55), then,

$$\hat{f} = T \cdot f = \hat{k} \cdot T \cdot d \quad (8.59)$$

If we multiply T^{-1} , then,

$$f = T^{-1} \cdot \hat{k} \cdot T \cdot d = k \cdot d \quad (8.60)$$

where $k = T^{-1} \cdot \hat{k} \cdot T$.

Since T is not a square matrix, we change (8.56) to the following form.

$$\begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{1y} \\ \hat{d}_{2x} \\ \hat{d}_{2y} \end{Bmatrix} = \begin{bmatrix} c & s & \mathbf{0} & \mathbf{0} \\ -s & c & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & c & s \\ \mathbf{0} & \mathbf{0} & -s & c \end{bmatrix} \begin{Bmatrix} d_{1x} \\ d_{1y} \\ d_{2x} \\ d_{2y} \end{Bmatrix} \quad (8.61)$$

Now we are ready to combine (8.61) with (8.52) for global force–displacement relation. However, in order to match the size of \hat{k} in (8.52) to 4×1 matrix of \hat{d} , we change (8.52) to the following equation.

$$\begin{Bmatrix} \hat{f}_{1x} \\ \hat{f}_{1y} \\ \hat{f}_{2x} \\ \hat{f}_{2y} \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{1y} \\ \hat{d}_{2x} \\ \hat{d}_{2y} \end{Bmatrix} \quad (8.62)$$

We simply added the second and fourth rows with zero shear force at each node thus all of the values in the second and fourth rows are zero. As a result, two equations (8.61) and (8.62) are ready to be put together into the following equation.

$$\begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix} \begin{Bmatrix} f_{1x} \\ f_{1y} \\ f_{2x} \\ f_{2y} \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix} \begin{Bmatrix} d_{1x} \\ d_{1y} \\ d_{2x} \\ d_{2y} \end{Bmatrix} \quad (8.63)$$

or

$$T \cdot f = \hat{k} \cdot T \cdot d$$

Since T^{-1} is equal to T^T for an orthogonal matrix, a matrix whose element vectors are orthonormal to each other (meaning dot product of each element vector is zero), (8.63) becomes,

$$\begin{Bmatrix} f_{1x} \\ f_{1y} \\ f_{2x} \\ f_{2y} \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix}^T \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix} \begin{Bmatrix} d_{1x} \\ d_{1y} \\ d_{2x} \\ d_{2y} \end{Bmatrix} \quad (8.64)$$

From the above equation, we define global k matrix such that,

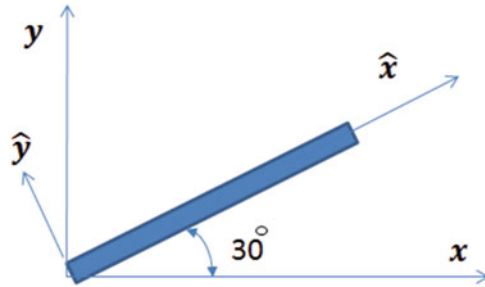
$$k = \frac{AE}{L} \begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & c & -s \\ 0 & 0 & s & c \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix}$$

or

$$k = \frac{AE}{L} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} = \frac{AE}{L} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ & s^2 & -cs & -s^2 \\ & & c^2 & cs \\ \text{Symmetry} & & & s^2 \end{bmatrix} \quad (8.65)$$

Sample Problem 8.4

The bar member shown below is a steel bar ($E = 30 \times 10^6$ psi) with 2 in.² area and 60 in. long. Find global stiffness matrix.



Solution

Since $\theta = 30^\circ$, $c = \cos 30 = \frac{\sqrt{3}}{2}$, $s = \sin 30 = \frac{1}{2}$. Therefore k is,

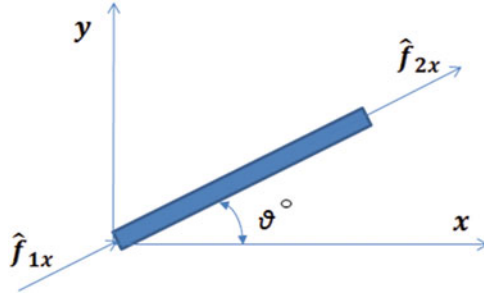
$$k = \frac{2 \times 30 \times 10^6}{60} \begin{bmatrix} \frac{3}{4} & \frac{\sqrt{3}}{4} & -\frac{3}{4} & -\frac{\sqrt{3}}{4} \\ & \frac{1}{4} & -\frac{\sqrt{3}}{4} & \frac{1}{4} \\ & & \frac{3}{4} & \frac{\sqrt{3}}{4} \\ \text{Symmetry} & & & \frac{1}{4} \end{bmatrix}$$

or

$$k = 10^6 \begin{bmatrix} 0.75 & 0.433 & -0.75 & -0.433 \\ & 0.25 & -0.433 & -0.25 \\ & & 0.75 & 0.433 \\ \text{Symmetry} & & & 0.25 \end{bmatrix}$$

8.8.4 Computation of Stress for a Bar in x-y Plane

In this section, we will consider the calculation of internally developed stress by external forces in a bar element. Let's assume a single bar member arbitrarily oriented.



For the figure above, since local force is aligned with main member direction, internally developed stress is simply local force per unit area, or

$$\sigma = \frac{\hat{f}_{2x}}{A}$$

From the local displacement and force relationship in (8.52), each local nodal force is

$$\hat{f}_{1x} = \frac{AE}{L} [1 \quad -1] \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix}$$

$$\hat{f}_{2x} = \frac{AE}{L} [-1 \quad 1] \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix}$$

Therefore, by definition, stress is

$$\hat{\sigma} = \frac{\hat{f}_{2x}}{A} = \frac{E}{L} [-1 \quad 1] \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix}$$

or

$$\hat{\sigma} = \frac{E}{L} [-1 \quad 1] \cdot \hat{d} = \frac{E}{L} [-1 \quad 1] \cdot Td$$

If we define,

$$c' = \frac{E}{L} [-1 \quad 1] \cdot T$$

Then,

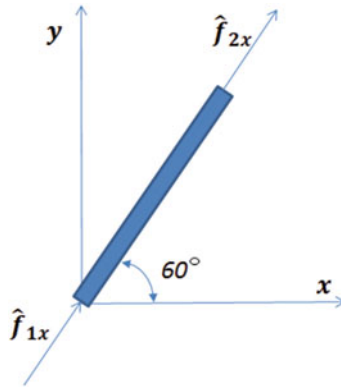
$$\hat{\sigma} = c' \cdot d$$

where

$$\begin{aligned}
 c' &= \frac{E}{L} \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{bmatrix} \\
 &= \frac{E}{L} \begin{bmatrix} -c & -s & c & s \end{bmatrix}
 \end{aligned}
 \tag{8.66}$$

Sample Problem 8.5

For the bar element below, find stress built in the bar.



$$A = 4 \times 10^{-4} \text{ m}^2, E = 210 \text{ GPa}, L = 2 \text{ m}, \theta = 60^\circ \text{ and } d = \begin{Bmatrix} 0.25 \text{ mm} \\ 0 \\ 0.5 \text{ mm} \\ 0.75 \text{ mm} \end{Bmatrix}$$

Solution

$$\begin{aligned}
 c' &= \frac{E}{L} \begin{bmatrix} -c & -s & c & s \end{bmatrix} \\
 &= \frac{210 \times 10^9}{2} \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}
 \end{aligned}$$

Therefore, stress is

$$\begin{aligned}
 \hat{\sigma} &= \frac{210 \times 10^9}{2} \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{Bmatrix} 0.25 \text{ mm} \\ 0 \\ 0.5 \text{ mm} \\ 0.75 \text{ mm} \end{Bmatrix} \\
 &= 81 \text{ Map}
 \end{aligned}$$

Sample Problem 8.6

Verify the result of the Problem 8.5 by calculating local nodal forces.

Solution

In order to find local nodal forces, we calculate local displacement first. By (8.56)

$$\begin{aligned}\hat{d}_{1x} &= 25 \text{ mm} \cdot \cos 60 + 0 \cdot \sin 60 = 0.125 \text{ mm} \\ \hat{d}_{2x} &= 0.5 \text{ mm} \cdot \cos 60 + 0.75 \text{ mm} \cdot \sin 60 = 0.9 \text{ mm}\end{aligned}$$

By (8.52),

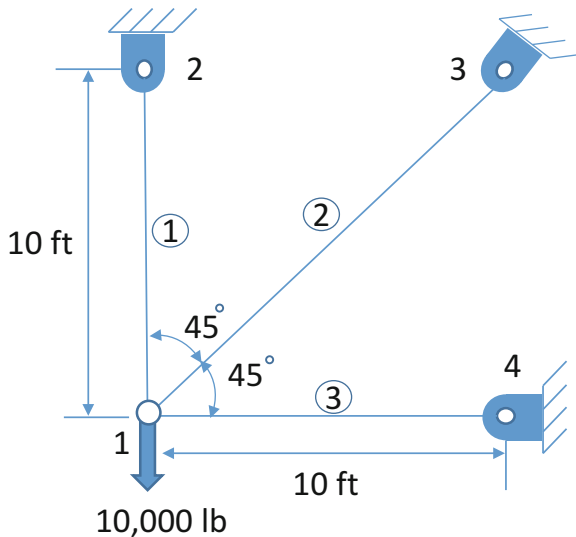
$$\begin{aligned}\begin{Bmatrix} \hat{f}_{1x} \\ \hat{f}_{2x} \end{Bmatrix} &= \frac{4 \times 10^{-4} \times 210 \times 10^9}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} 0.125 \times 10^{-3} \\ 0.9 \times 10^{-3} \end{Bmatrix} \\ &= \begin{Bmatrix} -0.775 \\ 0.775 \end{Bmatrix} \times 420 \times 10^2\end{aligned}$$

Therefore,

$$\hat{\sigma} = \frac{\hat{f}_{2x}}{A} = \frac{0.775 \times 420 \times 10^2}{4 \times 10^{-4}} = 81 \text{ Mpa}$$

8.9 Solution of a Plane Truss

Now in this section, we solve a truss problem. The truss structure under consideration is shown in the figure below.



All of the rods in the figure have the same properties as below.

$$\begin{aligned}
 A &= 2 \text{ in.}^2 \\
 E &= 3 \times 10^6 \text{ psi} \\
 L_1, L_3 &= 10\sqrt{2} = 120 \text{ in.} \\
 L_2 &= 120\sqrt{2} \text{ in.}
 \end{aligned}$$

Element data					
Element	θ	C	S	C^2	S^2
1	90	0	1	0	1
2	45	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
3	0	1	0	1	0

From the element data in the table above, we can find global stiffness matrix such as:

$$\begin{aligned}
 k^{(1)} &= \frac{2 \cdot (30 \times 10^6)}{120} \begin{bmatrix} d_{1x} & d_{1y} & d_{2x} & d_{2y} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{-1} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{matrix} d_{1x} \\ d_{1y} \\ d_{2x} \\ d_{2y} \end{matrix} \\
 k^{(2)} &= \frac{2 \cdot (30 \times 10^6)}{120\sqrt{2}} \begin{bmatrix} d_{1x} & d_{1y} & d_{3x} & d_{3y} \\ \mathbf{0.5} & \mathbf{0.5} & \mathbf{0.5} & \mathbf{0.5} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \mathbf{0.5} & \mathbf{0.5} & \mathbf{0.5} & \mathbf{0.5} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \mathbf{0.5} & \mathbf{0.5} & \mathbf{0.5} & \mathbf{0.5} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{matrix} d_{1y} \\ d_{1y} \\ d_{3y} \\ d_{3y} \end{matrix} \\
 k^{(3)} &= \frac{2 \cdot (30 \times 10^6)}{120} \begin{bmatrix} d_{1x} & d_{1y} & d_{4x} & d_{4y} \\ \mathbf{1} & \mathbf{0} & \mathbf{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{-1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{matrix} d_{1x} \\ d_{1y} \\ d_{4x} \\ d_{4y} \end{matrix}
 \end{aligned}$$

If we assemble all three stiffness matrices, we obtain total global stiffness matrix as below.

	d_{1x}	d_{1y}	d_{2x}	d_{2y}	d_{3x}	d_{3y}	d_{4x}	d_{4y}
	$0 + \frac{0.5}{\sqrt{2}} + 1 = 1.354$	$0 + \frac{0.5}{\sqrt{2}} + 0 = 0.354$	0	0	$-\frac{0.5}{\sqrt{2}} = -0.354$	$-\frac{0.5}{\sqrt{2}} = -0.354$	-1	0
	$0 + \frac{0.5}{\sqrt{2}} + 0 = 0.354$	$1 + \frac{0.5}{\sqrt{2}} + 0 = 1.354$	0	-1	$-\frac{0.5}{\sqrt{2}} = -0.354$	$-\frac{0.5}{\sqrt{2}} = -0.354$	0	0
	0	0	0	0	0	0	0	0
	0	-1	0	0	0	0	0	0
	$-\frac{0.5}{\sqrt{2}} = -0.354$	$-\frac{0.5}{\sqrt{2}} = -0.354$	0	0	$\frac{0.5}{\sqrt{2}} = 0.354$	$\frac{0.5}{\sqrt{2}} = 0.354$	0	0
	-1	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

50,000×

Therefore, total structure stiffness equation is

$$\begin{Bmatrix} 0 \\ -10,000 \\ F_{2x} \\ F_{2y} \\ \vdots \\ F_{4y} \end{Bmatrix} = K \cdot \begin{Bmatrix} u_1 \\ v_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix}$$

where u_1 and v_1 represent displacements of the node 1 along x and y axes.

To find u_1 and v_1 we remove all of the unknowns from the left side of the equation, such that,

$$\begin{Bmatrix} 0 \\ -10,000 \end{Bmatrix} = 50,000 \begin{bmatrix} 1.354 & 0.354 \\ 0.354 & 1.354 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix}$$

Therefore,

$$\begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix} = \frac{1}{50,000} \begin{bmatrix} 1.354 & -0.354 \\ -0.354 & 1.354 \end{bmatrix} \begin{Bmatrix} 0 \\ -10,000 \end{Bmatrix}$$

Finally, u_1 and v_1 are

$$u_1 = 0.414 \times 10^{-2} \text{in.}, \quad v_1 = -1.59 \times 10^{-2} \text{in.}$$

Since we found the displacement of the node 1, we can calculate stress on each rod by (8.66) such as,

$$\hat{\sigma}^{(1)} = \frac{E}{L} [-c \quad -s \quad c \quad s] \cdot \mathbf{d}^{(1)}$$

where

$$\mathbf{d}^{(1)} = \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \end{Bmatrix}$$

Therefore,

$$\hat{\sigma}^{(1)} = \frac{30 \times 10^6}{120} \begin{bmatrix} 0 & -1 & 0 & 1 \end{bmatrix} \cdot \begin{Bmatrix} 0.414 \times 10^{-2} \\ -1.59 \times 10^{-2} \\ 0 \\ 0 \end{Bmatrix} = 3965 \text{psi}$$

Likewise,

$$\begin{aligned} \hat{\sigma}^{(2)} &= \frac{E}{L}[-c \quad -s \quad c \quad s] \cdot d^{(2)} = \frac{E}{L}[-c \quad -s \quad c \quad s] \cdot \begin{Bmatrix} u_1 \\ v_1 \\ u_3 \\ v_3 \end{Bmatrix} \\ &= \frac{30 \times 10^6}{120} \left[-\frac{\sqrt{2}}{2} \quad -\frac{\sqrt{2}}{2} \quad \frac{\sqrt{2}}{2} \quad \frac{\sqrt{2}}{2} \right] \cdot \begin{Bmatrix} 0.414 \times 10^{-2} \\ -1.59 \times 10^{-2} \\ 0 \\ 0 \end{Bmatrix} = 1471 \text{ psi} \end{aligned}$$

Finally,

$$\begin{aligned} \hat{\sigma}^{(3)} &= \frac{E}{L}[-c \quad -s \quad c \quad s] \cdot d^{(3)} = \frac{E}{L}[-c \quad -s \quad c \quad s] \cdot \begin{Bmatrix} u_1 \\ v_1 \\ u_4 \\ v_4 \end{Bmatrix} \\ &= \frac{30 \times 10^6}{120} [-1 \quad 0 \quad 1 \quad 0] \cdot \begin{Bmatrix} 0.414 \times 10^{-2} \\ -1.59 \times 10^{-2} \\ 0 \\ 0 \end{Bmatrix} = -1035 \text{ psi} \end{aligned}$$

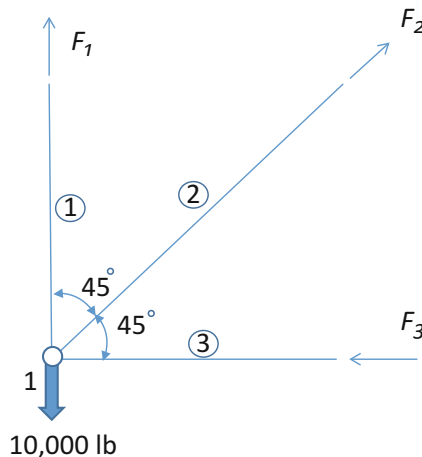
For more information in FEM analysis and examples, please refer to [2].

Sample Problem 8.7

Verify the results of the truss problem above with force equilibrium condition.

Solution

For the static force analysis, we first draw FBD as shown below.



Since the force equilibrium condition has to be satisfied,

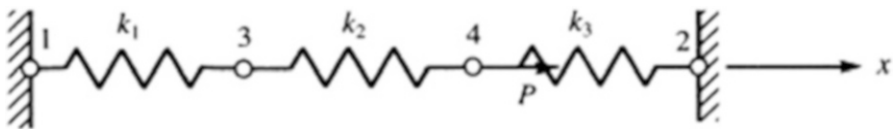
$$\sum F_x = 0, \quad \text{and} \quad \sum F_y = 0$$

Therefore,

$$\begin{aligned} \sum F_x &= F_2 \cos(45) - F_3 \\ &= \hat{\sigma}^{(2)} \cdot A^{(2)} \cos(45) - \hat{\sigma}^{(3)} \cdot A^{(3)} \\ &= 1471 \times 2 \times \frac{\sqrt{2}}{2} - 1035 \times 2 = 0 \\ \sum F_y &= F_1 + F_2 \sin(45) - 10,000 \\ &= \hat{\sigma}^{(1)} \cdot A^{(1)} + \hat{\sigma}^{(2)} \cdot A^{(2)} \sin(45) - 10,000 \\ &= 3965 \times 2 + 1471 \times 2 \times \frac{\sqrt{2}}{2} - 10,000 = 0 \end{aligned}$$

Exercise Problem 8.1

- (a) Obtain the global stiffness matrix [K] of the assemblage shown in the figure below by superimposing the stiffness matrices of the individual springs.
- (b) If nodes 1 and 2 are fixed and a force P acts on node 4 in positive x directions, find expression for the displacements of nodes 3 and 4.
- (c) Determine the reaction forces at nodes 1 and 2.



Solution

(a)

$$[k^{(1)}] = \begin{bmatrix} k_1 & 0 & -k_1 & 0 \\ 0 & 0 & 0 & 0 \\ -k_1 & 0 & k_1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$[k^{(2)}] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & k_2 & -k_2 \\ 0 & 0 & -k_2 & k_2 \end{bmatrix}$$

$$[k_3^{(3)}] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & k_3 & 0 & -k_3 \\ 0 & 0 & 0 & 0 \\ 0 & -k_3 & 0 & k_3 \end{bmatrix}$$

$$[K] = [k^{(1)}] + [k^{(2)}] + [k^{(3)}]$$

$$[K] = \begin{bmatrix} k_1 & 0 & -k_1 & 0 \\ 0 & k_3 & 0 & -k_3 \\ -k_1 & 0 & k_1 + k_2 & -k_2 \\ 0 & -k_3 & -k_2 & k_2 + k_3 \end{bmatrix}$$

(b)

$$[K] = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 + k_3 \end{bmatrix}$$

$$\{F\} = [K]\{d\}$$

$$\begin{Bmatrix} F_{3x} \\ F_{4x} \end{Bmatrix} = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 + k_3 \end{bmatrix} \begin{Bmatrix} u_3 \\ u_4 \end{Bmatrix}$$

$$\Rightarrow \begin{Bmatrix} 0 \\ P \end{Bmatrix} = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 + k_3 \end{bmatrix} \begin{Bmatrix} u_3 \\ u_4 \end{Bmatrix}$$

$$\{F\} = [K]\{d\} \Rightarrow [K^{-1}]\{F\} = [K^{-1}][K]\{d\}$$

$$\Rightarrow [K^{-1}]\{F\} = \{d\}$$

Using the adjoint method to find $[K^{-1}]$

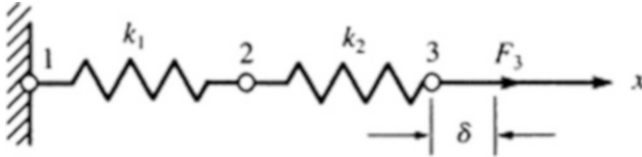
$$\begin{aligned}
C_{11} &= k_2 + k_3 & C_{21} &= (-1)^3(-k_2) \\
C_{12} &= (-1)^{1+2}(-k_2) = k_2 & C_{22} &= k_1 + k_2 \\
[C] &= \begin{bmatrix} k_2 + k_3 & k_2 \\ k_2 & k_1 + k_2 \end{bmatrix} \quad \text{and} \quad C^T = \begin{bmatrix} k_2 + k_3 & k_2 \\ k_2 & k_1 + k_2 \end{bmatrix} \\
\det[K] &= |[K]| = (k_1 + k_2)(k_2 + k_3) - (-k_2)(-k_2) \\
&\Rightarrow |[K]| = (k_1 + k_2)(k_2 + k_3) - k_2^2 \\
[K^{-1}] &= \frac{[C^T]}{\det K} \\
[K^{-1}] &= \frac{\begin{bmatrix} k_2 + k_3 & k_2 \\ k_2 & k_1 + k_2 \end{bmatrix}}{(k_1 + k_2)(k_2 + k_3) - k_2^2} = \frac{\begin{bmatrix} k_2 + k_3 & k_2 \\ k_2 & k_1 + k_2 \end{bmatrix}}{k_1k_2 + k_1k_3 + k_2k_3} \\
\begin{Bmatrix} u_3 \\ u_4 \end{Bmatrix} &= \frac{\begin{bmatrix} k_2 + k_3 & k_2 \\ k_2 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} 0 \\ P \end{Bmatrix}}{k_1k_2 + k_1k_3 + k_2k_3} \\
&\Rightarrow u_3 = \frac{k_2P}{k_1k_2 + k_1k_3 + k_2k_3} \\
&\Rightarrow u_4 = \frac{(k_1 + k_2)P}{k_1k_2 + k_1k_3 + k_2k_3}
\end{aligned}$$

(c)

$$\begin{aligned}
\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \\ F_{4x} \end{Bmatrix} &= \begin{bmatrix} k_1 & 0 & -k_1 & 0 \\ 0 & k_3 & 0 & -k_3 \\ -k_1 & 0 & k_1 + k_2 & -k_2 \\ 0 & -k_3 & -k_2 & k_2 + k_3 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} \\
F_{1x} &= -k_1u_3 = -k_1 \frac{k_2P}{k_1k_2 + k_1k_3 + k_2k_3} \\
&\Rightarrow F_{1x} = \frac{-k_1k_2P}{k_1k_2 + k_1k_3 + k_2k_3} \\
F_{2x} &= -k_3u_4 = -k_3 \frac{(k_1 + k_2)P}{k_1k_2 + k_1k_3 + k_2k_3} \\
&\Rightarrow F_{2x} = \frac{-k_3(k_1 + k_2)P}{k_1k_2 + k_1k_3 + k_2k_3}
\end{aligned}$$

Exercise Problem 8.2

For the spring assemblage shown below, determine the displacement at node 2 and the forces in each spring element. Also determine the force F_3 . Given: Node 3 displaces an amount $\delta = 1$ in. in the positive x direction and $k_1 = k_2 = 1000$ lb/in.

**Solution**

$$[k^{(1)}] = \begin{matrix} (1) & (2) \\ \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} & \begin{matrix} (1) \\ (2) \end{matrix} \end{matrix}; \quad [k^{(2)}] = \begin{matrix} (2) & (3) \\ \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} & \begin{matrix} (2) \\ (3) \end{matrix} \end{matrix}$$

By the method of superposition, the global stiffness matrix is constructed.

$$[K] = \begin{matrix} (1) & (2) & (3) \\ \begin{bmatrix} k & -k & 0 \\ -k & k+k & -k \\ 0 & -k & k \end{bmatrix} & \begin{matrix} (1) \\ (2) \\ (3) \end{matrix} \end{matrix} \Rightarrow [K] = \begin{bmatrix} k & -k & 0 \\ -k & 2k & -k \\ 0 & -k & k \end{bmatrix}$$

Node 1 is fixed $\Rightarrow u_1 = 0$ and $u_3 = \delta$

$$\{F\} = [K]\{d\}$$

$$\begin{Bmatrix} F_{1x} = ? \\ F_{2x} = 0 \\ F_{3x} = ? \end{Bmatrix} = \begin{bmatrix} k & -k & 0 \\ -k & 2k & -k \\ 0 & -k & k \end{bmatrix} \begin{Bmatrix} u_1 = 0 \\ u_2 = ? \\ u_3 = \delta \end{Bmatrix}$$

$$\begin{Bmatrix} 0 \\ F_{3x} \end{Bmatrix} = \begin{bmatrix} 2k & -k \\ -k & k \end{bmatrix} \begin{Bmatrix} u_2 \\ \delta \end{Bmatrix} \Rightarrow \begin{cases} 0 = 2ku_2 - k\delta \\ F_{3x} = -ku_2 + k\delta \end{cases}$$

$$\Rightarrow u_2 = \frac{k\delta}{2k} = \frac{\delta}{2} = \frac{1 \text{ in.}}{2} \Rightarrow u_2 = 0.5''$$

$$F_{3x} = -k(0.5'') + k(1'')$$

$$F_{3x} = \left(-1000 \frac{\text{lb}}{\text{in.}}\right)(0.5'') + \left(1000 \frac{\text{lb}}{\text{in.}}\right)(1'')$$

Internal force calculation:

Element (1)

$$\begin{aligned} \begin{Bmatrix} f_{1x}^{(1)} \\ f_{2x}^{(2)} \end{Bmatrix} &= \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{Bmatrix} u_1 = 0 \\ u_2 = 0.5'' \end{Bmatrix} \\ \Rightarrow f_{1x}^{(1)} &= \left(-1000 \frac{\text{lb}}{\text{in.}}\right) (0.5'') \Rightarrow f_{1x}^{(1)} = -500 \text{ lb} \\ f_{2x}^{(1)} &= \left(1000 \frac{\text{lb}}{\text{in.}}\right) (0.5'') \Rightarrow f_{2x}^{(1)} = 500 \text{ lb} \end{aligned}$$

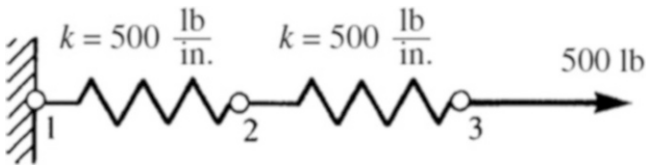
Element (2)

$$\begin{Bmatrix} f_{2x}^{(2)} \\ f_{3x}^{(2)} \end{Bmatrix} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{Bmatrix} u_2 = 0.5'' \\ u_3 = 1'' \end{Bmatrix} \Rightarrow \begin{aligned} f_{2x}^{(2)} &= -500 \text{ lb} \\ f_{3x}^{(2)} &= 500 \text{ lb} \end{aligned}$$

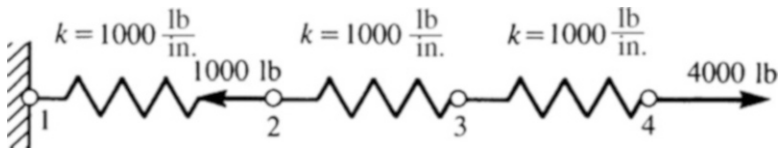
Exercise Problem 8.3–8.7

For the spring assemblages shown in figures below, determine nodal displacements, forces in each element, and reactions. Use the direct stiffness methods for all problems.

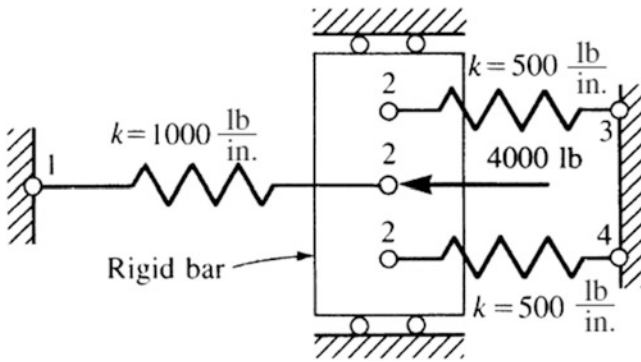
(a)



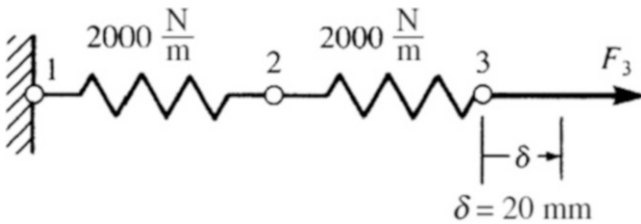
(b)



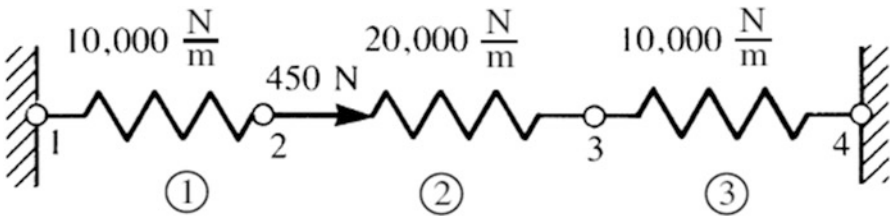
(c)



(d)



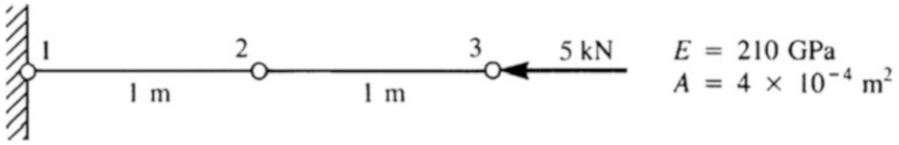
(e)



Exercise Problem 8.8–8.11

For the bar assemblages shown in the figures below, determine nodal displacements, forces in each element, and reactions. Use the direct stiffness method for these problems.

(a)

**Solution**

Element 1 – 2

$$[k_{1-2}] = 84 \times 10^6 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Element 2 – 3

$$[k_{2-3}] = 84 \times 10^6 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$\{F\} = [K]\{d\} \quad \text{and} \quad u_1 = 0$$

$$\begin{cases} F_{1x} = ? \\ F_{2x} = 0 \\ F_{3x} = -5000 \end{cases} = 84 \times 10^6 \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{cases} u_1 = 0 \\ u_2 = ? \\ u_3 = ? \end{cases}$$

$$\Rightarrow 2u_2 - u_3 = 0 \Rightarrow u_3 = 2u_2$$

$$\Rightarrow -5000 = 84 \times 10^6 [-u_2 + u_3]$$

Substituting (8.1) in (8.2), we have

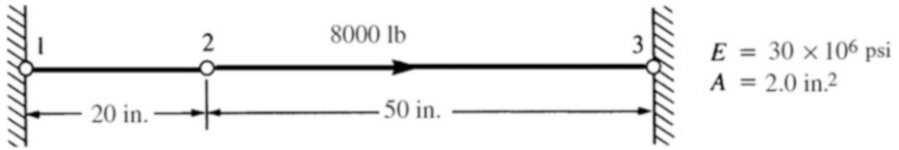
$$\frac{-5000}{84 \times 10^6} = -u_2 + 2u_2 \Rightarrow u_2 = -0.595 \times 10^{-4} \text{ m}$$

$$\Rightarrow u_3 = -1.19 \times 10^{-4} \text{ m}$$

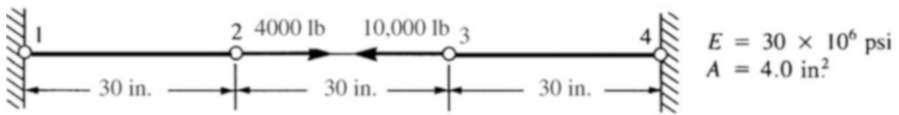
Element 1–2

$$\begin{cases} f_{1x} \\ f_{2x} \end{cases} = 84 \times 10^6 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{cases} 0 \\ -0.595 \times 10^{-4} \end{cases} \Rightarrow \begin{cases} f_{1x}^{(1)} = 5000 \text{ N} \\ f_{2x}^{(1)} = -5000 \text{ N} \end{cases}$$

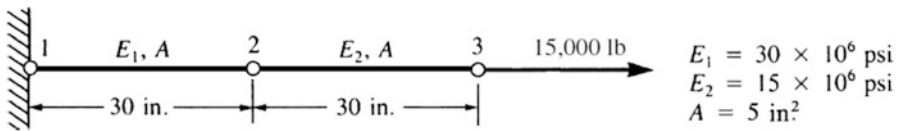
(b)



(c)



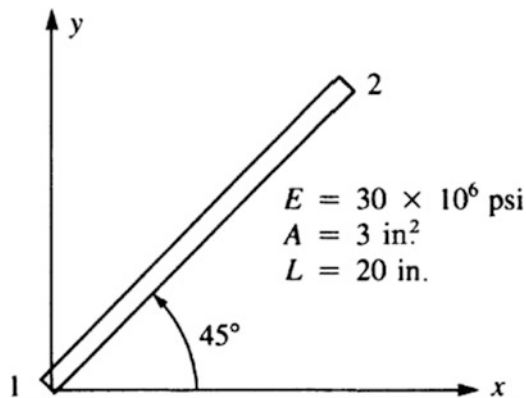
(d)



Exercise Problem 8.12–8.14

For the bar elements shown below, evaluate the global x - y stiffness matrix.

(a)



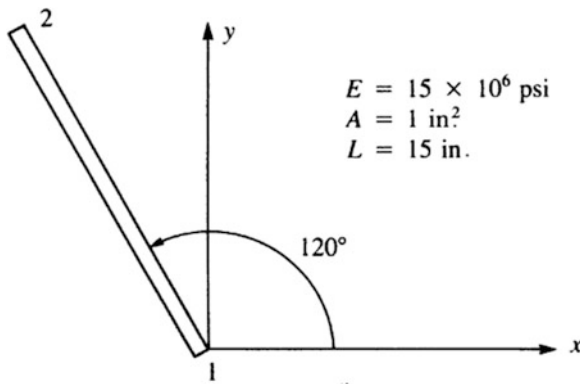
Solution

$$C = \frac{1}{\sqrt{2}}, \quad S = \frac{1}{\sqrt{2}}$$

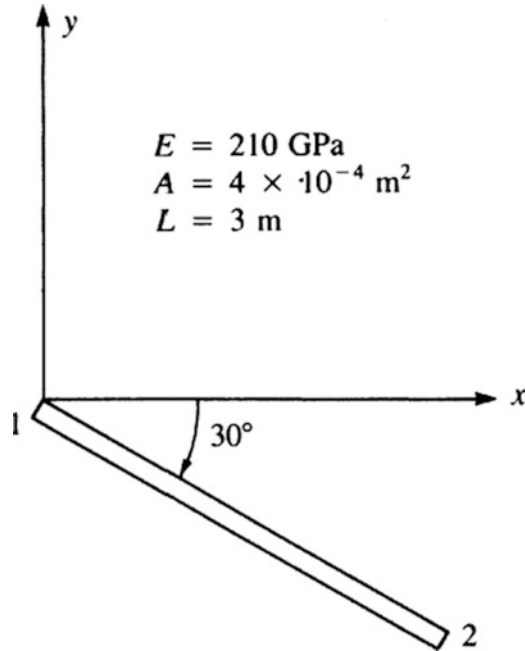
$$[K] = \frac{EA}{L} \begin{bmatrix} C^2 & CS & -C^2 & -CS \\ & S^2 & -CS & -S^2 \\ & & C^2 & CS \\ & & & S^2 \end{bmatrix}$$

$$[K] = 2.25 \times 10^6 \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \frac{\text{lb}}{\text{in.}}$$

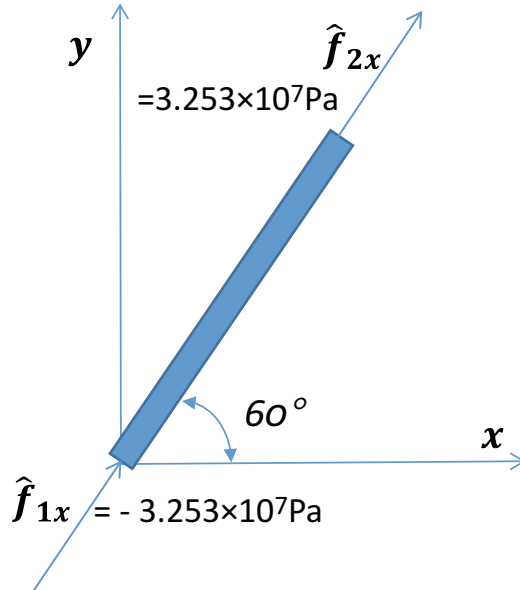
(b)



(c)

**Exercise Problem 8.15**

The bar member shown below is a steel bar ($E = 210 \text{ GPa}$) with $4 \times 10^{-4} \text{ m}^2$ area and 2 m long.



- (a) Find the global stiffness matrix.
- (b) Find the global displacement using the boundary condition below

$$d_{1x} = 0\text{m}, \quad d_{1y} = 0\text{m}.$$

- (c) Find the stress built in the bar.

Solution by Matlab

```

k = 1.0e + 07*
1.0501    1.8187   -1.0501   -1.8187
1.8187    3.1499   -1.8187   -3.1499
-1.0501   -1.8187    1.0501    1.8187
-1.8187   -3.1499    1.8187    3.1499
    
```

(b) Since $f = T^{-1} \cdot \hat{f}$ and $f = k \cdot d$

$$\begin{aligned}
 T &= \\
 &\begin{matrix} 0.5000 & 0.8660 & 0 & 0 \\ -0.8660 & 0.5000 & 0 & 0 \\ 0 & 0 & 0.5000 & 0.8660 \\ 0 & 0 & -0.8660 & 0.5000 \end{matrix} \\
 \hat{f} &= 1.0e + 07^* \\
 &\begin{matrix} -3.2530 \\ 0 \\ 3.2530 \\ 0 \end{matrix} \\
 f &= 1.0e + 07^* \\
 &\begin{matrix} -1.6266 \\ -2.8171 \\ 1.6266 \\ 2.8171 \end{matrix}
 \end{aligned}$$

Since $d_{Ix} = 0$ m, $d_{Iy} = 0$ m,

$$\begin{Bmatrix} f_{2x} \\ f_{2y} \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} c^2 & cs \\ cs & s^2 \end{bmatrix} \begin{Bmatrix} d_{2x} \\ d_{2y} \end{Bmatrix}$$

Therefore,

$$\begin{Bmatrix} d_{2x} \\ d_{2y} \end{Bmatrix} = \frac{L}{AE} \begin{bmatrix} c^2 & cs \\ cs & s^2 \end{bmatrix}^{-1} \begin{Bmatrix} f_{2x} \\ f_{2y} \end{Bmatrix}$$

0.0001
0.8943

(c)

$$\begin{aligned}
 c' &= 1.0e + 10^* \\
 &\begin{matrix} -5.2503 & -9.0931 & 5.2503 & 9.0931 \end{matrix}
 \end{aligned}$$

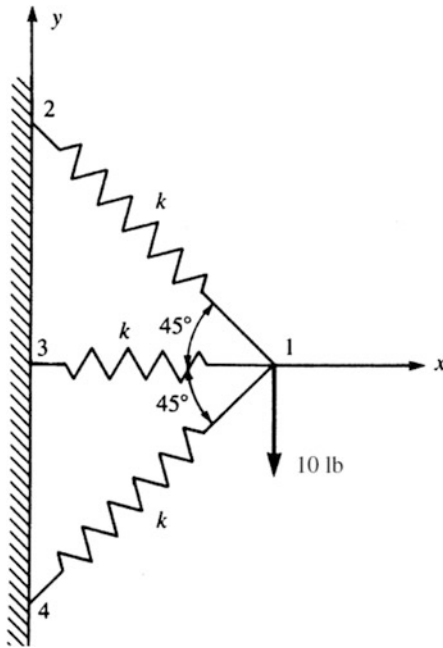
Therefore,

$$\hat{\sigma} = 8.1325e + 10\text{Pa}$$

Exercise Problem 8.16

(a) Assemble the stiffness matrix for the assemblage shown below by superimposing the stiffness matrices of the springs.

(b) Find the x, y components of deflection of node 1.



Solution

$$\{f\} = \begin{Bmatrix} f_{1x} \\ f_{1y} \\ f_{2x} \\ f_{2y} \\ f_{3x} \\ f_{3y} \\ f_{4x} \\ f_{4y} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -10 \\ f_{2x} \\ f_{2y} \\ f_{3x} \\ f_{3y} \\ f_{4x} \\ f_{4y} \end{Bmatrix} \quad \text{and} \quad \{d\} = \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix}$$

(a) For element 1-3; $\theta = 180^\circ$

$$\begin{Bmatrix} f_{1x} \\ f_{1y} \\ f_{3x} \\ f_{3y} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -10 \\ f_{3x} \\ f_{3y} \end{Bmatrix} = K \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ 0 \\ 0 \end{Bmatrix}$$

For element 1-4; $\theta = 225^\circ$

$$\begin{Bmatrix} f_{1x} \\ f_{1y} \\ f_{4x} \\ f_{4y} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -10 \\ f_{4x} \\ f_{4y} \end{Bmatrix} = \frac{K}{2} \begin{bmatrix} 1 & 0 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ 0 \\ 0 \end{Bmatrix}$$

For element 1-2; $\theta = 135^\circ$

$$\begin{Bmatrix} f_{1x} \\ f_{1y} \\ f_{2x} \\ f_{2y} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -10 \\ f_{2x} \\ f_{2y} \end{Bmatrix} = \frac{K}{2} \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ 0 \\ 0 \end{Bmatrix}$$

Total K

$$[K] = K \begin{bmatrix} 2 & 0 & -\frac{1}{2} & \frac{1}{2} & -1 & 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 1 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

(b) Applying boundary conditions

$$u_4 = v_4 = u_2 = v_2 = u_3 = v_3 = 0$$

[K] is reduced to

$$[K] = K \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{Bmatrix} f_{1x} \\ f_{1y} \end{Bmatrix} = K \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix} \Rightarrow \begin{Bmatrix} 0 \\ -10 \end{Bmatrix} K \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix}$$

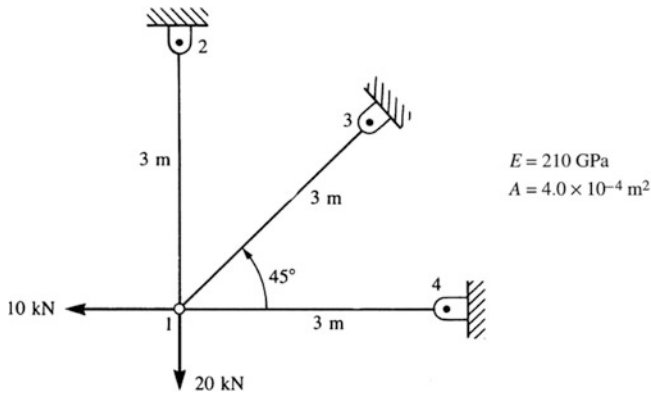
$$\Rightarrow u_1 = 0$$

$$v_1 = \frac{-10}{K}$$

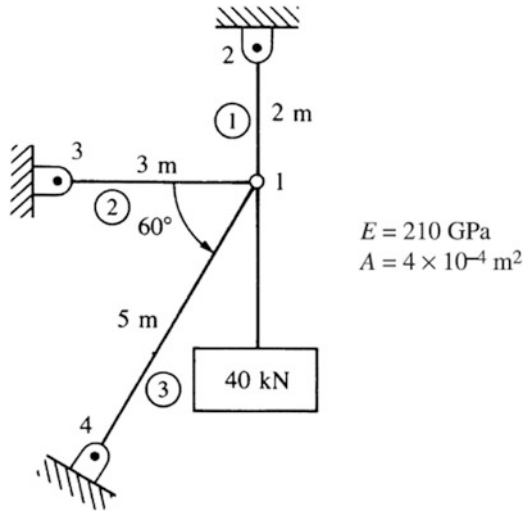
Exercise Problem 8.17–8.18

For the plane trusses shown below, determine the horizontal and vertical displacements of node 1 and the stresses in each element. All elements have $E = 210 \text{ GPa}$ and $A = 4.0 \times 10^{-4} \text{ m}^2$.

(a)



(b)



For more details of FEM methods, please refer to the article in references.

References

1. Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W (1993) Mesh optimization. TR 93-01-01, Dept. of Computer Science and Engineering, University of Washington
2. Logan DL. A first course in the finite element method, 5th ed. Cengage Learning

Appendix A: Tolerance Classification

Tolerance classification (metric values)

Basic size ^b	Loose running			Free running			Close running			Sliding			Locational clearance								
	Hole	Shaft	Fit ^b	Hole	Shaft	Fit ^b	Hole	Shaft	Fit ^b	Hole	Shaft	Fit ^b	Hole	Shaft	Fit ^b						
1	Max	1.060	0.940	0.180	0.980	0.070	H9	d9	0.980	0.070	H8	f7	0.994	0.998	0.018	H7	h6	1.010	1.000	0.016	
	Min	1.000	0.880	0.060	1.000	0.995	0.020	1.000	0.995	0.020	1.000	0.984	0.006	1.000	0.992	0.002	1.000	0.994	1.000	0.994	0.000
1.2	Max	1.260	1.140	0.180	1.180	0.070	1.225	1.180	0.070	1.214	1.194	0.030	1.210	1.198	0.018	1.210	1.200	1.200	1.200	1.200	0.016
	Min	1.200	1.080	0.060	1.200	1.155	0.020	1.200	1.155	0.020	1.200	1.184	0.006	1.200	1.192	0.002	1.200	1.194	1.200	1.194	0.000
1.6	Max	1.660	1.540	0.180	1.625	0.070	1.625	1.580	0.070	1.614	1.594	0.030	1.610	1.598	0.018	1.610	1.600	1.610	1.610	1.600	0.016
	Min	1.600	1.480	0.060	1.600	1.555	0.020	1.600	1.555	0.020	1.600	1.584	0.006	1.600	1.592	0.002	1.600	1.594	1.600	1.594	0.000
2	Max	2.060	1.940	0.180	2.025	0.070	2.025	1.980	0.070	2.014	1.994	0.030	2.010	1.998	0.018	2.010	2.000	2.010	2.010	2.000	0.016
	Min	2.000	1.880	0.060	2.000	1.955	0.020	2.000	1.955	0.020	2.000	1.984	0.006	2.000	1.992	0.002	2.000	1.994	2.000	1.994	0.000
2.5	Max	2.560	2.440	0.180	2.480	0.070	2.525	2.480	0.070	2.514	2.494	0.030	2.510	2.498	0.018	2.510	2.500	2.510	2.500	2.500	0.016
	Min	2.500	2.380	0.060	2.500	2.455	0.020	2.500	2.455	0.020	2.500	2.484	0.006	2.500	2.484	0.002	2.500	2.494	2.500	2.494	0.000
3	Max	3.060	2.940	0.180	3.025	0.070	3.025	2.980	0.070	3.014	2.994	0.030	3.010	2.998	0.018	3.010	3.000	3.010	3.000	3.000	0.016
	Min	3.000	2.880	0.060	3.000	2.955	0.020	3.000	2.955	0.020	3.000	2.984	0.006	3.000	2.992	0.002	3.000	2.994	3.000	2.994	0.000
4	Max	4.075	3.930	0.220	4.030	0.090	4.030	3.970	0.090	4.018	3.990	0.040	4.012	3.996	0.024	4.012	4.000	4.012	4.000	4.000	0.020
	Min	4.000	3.855	0.070	4.000	3.940	0.030	4.000	3.940	0.030	4.000	3.978	0.010	4.000	3.988	0.004	4.000	3.992	4.000	3.992	0.000
5	Max	5.075	4.930	0.220	5.030	0.090	5.030	4.970	0.090	5.018	4.990	0.040	5.012	4.996	0.024	5.012	5.000	5.012	5.000	5.000	0.020
	Min	5.000	4.855	0.070	5.000	4.940	0.030	5.000	4.940	0.030	5.000	4.978	0.010	5.000	4.988	0.004	5.000	4.992	5.000	4.992	0.000
6	Max	6.075	5.930	0.220	6.030	0.090	6.030	5.970	0.090	6.018	5.990	0.040	6.012	5.996	0.024	6.012	6.000	6.012	6.000	6.000	0.020
	Min	6.000	5.855	0.070	6.000	5.940	0.030	6.000	5.940	0.030	6.000	5.978	0.010	6.000	5.988	0.004	6.000	5.992	6.000	5.992	0.000
8	Max	8.090	7.920	0.260	8.036	0.112	8.022	7.960	0.112	8.022	7.987	0.050	8.015	7.995	0.029	8.015	8.000	8.015	8.000	8.000	0.024
	Min	8.000	7.830	0.080	8.000	7.924	0.040	8.000	7.924	0.040	8.000	7.972	0.013	8.000	7.986	0.005	8.000	7.991	8.000	7.991	0.000
10	Max	10.090	9.920	0.260	10.036	0.112	10.022	9.960	0.112	10.022	9.987	0.050	10.015	9.995	0.029	10.015	10.000	10.015	10.000	10.000	0.024
	Min	10.000	9.830	0.080	10.000	9.924	0.040	10.000	9.924	0.040	10.000	9.972	0.013	10.000	9.986	0.005	10.000	9.991	10.000	9.991	0.000
12	Max	12.110	11.905	0.315	12.043	0.136	12.027	11.984	0.136	12.027	11.984	0.061	12.018	11.994	0.035	12.018	12.000	12.018	12.000	12.000	0.029
	Min	12.000	11.795	0.095	12.000	11.907	0.050	12.000	11.907	0.050	12.000	11.966	0.016	12.000	11.983	0.006	12.000	11.989	12.000	11.989	0.000

16	Max	16.110	15.905	0.315	16.043	15.950	0.136	16.027	15.984	0.061	16.018	15.994	0.035	16.018	16.000	0.029
	Min	16.000	15.795	0.095	16.000	15.907	0.050	16.000	15.966	0.016	16.000	15.983	0.006	16.000	15.989	0.000
20	Max	20.130	19.890	0.370	20.052	19.935	0.169	20.033	19.980	0.074	20.021	19.993	0.041	20.021	20.000	0.034
	Min	20.000	19.760	0.110	20.000	19.883	0.065	20.000	19.959	0.020	20.000	19.980	0.007	20.000	19.987	0.000
25	Max	25.130	24.890	0.370	25.052	24.935	0.169	25.033	24.980	0.074	25.021	24.993	0.041	25.021	25.000	0.034
	Min	25.000	24.760	0.110	25.000	24.883	0.065	25.000	24.959	0.020	25.000	24.980	0.007	25.000	24.987	0.000



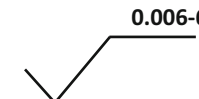

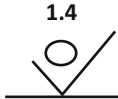
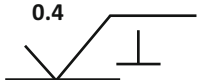
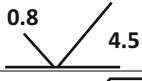
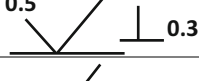



Source: From Machinery's handbook

Tolerance classification (inch values)






Nominal size range (in.)	Class LN 1			Class LN 2			Class LN 3		
	Limits of interference	Standard limits		Limits of interference	Standard limits		Limits of interference	Standard limits	
		Hole H6	Shaft n5		Hole H7	Shaft p6		Hole H7	Shaft r6
Over	To	Values shown below are given in thousandths of an inch							
0-0.12	0	+0.25	+0.45	0	+0.4	+0.65	0.1	+0.4	+0.75
	0.45	0	+0.25	0.65	0	+0.4	0.75	0	+0.5
0.12-0.24	0	+0.3	+0.5	0	+0.5	+0.8	0.1	+0.5	+0.9
	0.5	0	+0.3	0.8	0	+0.5	0.9	0	+0.6
0.24-0.40	0	+0.4	+0.65	0	+0.6	+1.0	0.2	+0.6	+1.2
	0.65	0	+0.4	1.0	0	+0.6	1.2	0	+0.8
0.40-0.71	0	+0.4	+0.8	0	+0.7	+1.1	0.3	+0.7	+1.4
	0.8	0	+0.4	1.1	0	+0.7	1.4	0	+1.0
0.71-1.19	0	+0.5	+1.0	0	+0.8	+1.3	0.4	+0.8	+1.7
	1.0	0	+0.5	1.3	0	+0.8	1.7	0	+1.2
1.19-1.97	0	+0.6	+1.1	0	+1.0	+1.6	0.4	+1.0	+2.0
	1.1	0	+0.6	1.6	0	+1.0	2.0	0	+1.4
1.97-3.15	0.1	+0.7	+1.3	0.2	+1.2	+2.1	0.4	+1.2	+2.3
	1.3	0	+0.8	2.1	0	+1.4	2.3	0	+1.6
3.15-4.73	0.1	+0.9	+1.6	0.2	+1.4	+2.5	0.6	+1.4	+2.9
	1.6	0	+1.0	2.5	0	+1.6	2.9	0	+2.0
4.73-7.09	0.2	+1.0	+1.9	0.2	+1.6	+2.8	0.9	+1.6	+3.5
	1.9	0	+1.2	2.8	0	+1.8	3.5	0	+2.5
7.09-9.85	0.2	+1.2	+2.2	0.2	+1.8	+3.2	1.2	+1.8	+4.2
	2.2	0	+1.4	3.2	0	+2.0	4.2	0	+3.0
9.85-12.41	0.2	+1.2	+2.3	0.2	+2.0	+3.4	1.5	+2.0	+4.7
	2.3	0	+1.4	3.4	0	+2.2	4.7	0	+3.5
12.41-15.75	0.2	+1.4	+2.6	0.3	+2.2	+3.9	2.3	+2.2	+5.9
	2.6	0	+1.6	3.9	0	+2.5	5.9	0	+4.5
15.75-19.69	0.2	+1.6	+2.8	0.3	+2.5	+4.4	2.5	+2.5	+6.6
	2.8	0	+1.8	4.4	0	+2.8	6.6	0	+5.0

Source: From Machinery's handbook

Appendix B: Surface Finish Symbols

	Roughness average rating (maximum) in microinches or micrometers
	Roughness average rating (maximum and minimum) in microinches or micrometers
	Maximum waviness height (first number) in mm or in. Maximum waviness spacing (second number) specified in millimeters or inches
	Amount of stock provided for material removal in millimeters or inches
	Removal of material is prohibited
	Lay direction is perpendicular to this edge of the surface
	Roughness length or cutoff rating in mm or inches below the horizontal. When no value is shown. Use 0.8 mm (0.03 in.)
	Roughness spacing (maximum) in mm or inches is placed to the right of the lay symbol
	A. Basic surface texture symbol: surface may be produced by any method
	B. Material removal by machining: indicated by horizontal bar
	C. Material removal allowance: the amount of stock (mm or in.) to be removed by machining

(continued)

	<p>D. Material removal prohibited: surface to be produced by hot finishing, casting, die casting, etc. without removing material</p>
<p>.003</p> 	<p>E. Surface texture symbol: used when values for surface characteristics are added above the horizontal or to the right</p>
<p>F. Maching symbols: the symbols below are used to recommend machining operations</p>	
	 

Index

A

Abacus, 2
ABS, 209
ACAD, 1
Additive manufacturing, 192
Allowance, 32
Alternatives, 204
Ambient occlusion, 182
Angularity, 41, 44, 45
ANSI Y14, 19–27
ANSIS, 2
Artificial food, 193
Association, 19, 96, 215, 237, 249, 253
AUTOCAD, 110, 119, 133
Auto-shading, 189
Axonometric, 23, 24, 27
Azimuth angle, 171
Azure noise, 180

B

Basic size, 32
Bead blasting, 220
Bending moment, 236, 255
Bezier spline curve, 112–115, 117, 119, 120, 139, 141
BiCubic, 130
Bidirectional reflectance distribution function (BRDF), 171
Bilateral, 27
Bilinear interpolation, 183, 184
Bilinear surface, 122
Bill of Material (BOM), 4
Bird's eye view, 26, 27
Blending function, 113

Block removal, 207, 208
Blue noise, 180
Boolean operation, 80–83
Boundary conditions, 99–101, 240, 244–249, 253, 257, 259, 281, 284
Boundary representation, 79, 80
Bounding box, 155, 158
B-rep, 79, 81, 212
Brightness, 173
B-spline curve, 93
Bubbles, 7

C

Cabinet oblique, 25
CAD, 1–6, 11–16, 51, 64, 112, 162, 192, 194, 200
CAD/CAE, 1, 12–16
CAE, 1, 2, 4, 5, 11–13, 15
Cartesian coordinate, 52, 97
Casting, 199–201
Cavalier oblique, 25
Cavendish's method, 227–229
Chain tolerancing, 28, 29
Chroma, 173
CIE, 175–178
CIE xyY, 176–178
CIE XYZ, 175, 176
Circular interpolation, 149
Circularity, 42, 43
Clearance, 32–34, 36, 39
Clearance fit, 33, 34, 36
CMYK, 172
CNC, 201
Collision check, 96

- Collision detection, 81
 - Colorfulness, 173
 - Colorimetry, 175, 176
 - Color quantization, 174, 180
 - Communication, 3
 - Compatibility, 243, 247
 - Compound transformation, 61
 - Computer Aided Design (CAD), 1–6
 - Computer Aided Engineering (CAE), 1
 - Computer graphics, 51, 93, 95, 143–168, 171, 172, 180, 182, 185, 191, 233
 - Concentricity, 40
 - Cones, 80, 174
 - Connecting rod, 229, 232
 - Connectivity, 79, 215, 219
 - Constructive solid geometry (CSG), 79–91
 - Contact area, 45
 - Continuity, 243
 - Continuum, 223
 - Contour crafting (CC), 194
 - Control net, 112
 - Control points, 97, 98, 101–105, 111–115, 117, 120, 130, 132, 133, 135, 138–141
 - Conventional method, 4
 - Cook-Torrance, 183
 - Coon's patch, 131–133
 - Coordinate, 52–55
 - Coordinate transformation, 55
 - Cousin color space, 176–177
 - Cubic polynomial coefficient, 99–101, 106, 107, 136, 138, 141
 - Cubic polynomial equation, 97, 98
 - Cubic spline curve, 97, 98, 107
 - Cure depth, 217, 219
 - Curvature, 101
 - Curve mesh surface, 77
 - Cylindricity, 42
- D**
- Datum lines, 28
 - Datum plane, 28, 29, 31
 - Datum targets, 31, 32
 - DDA interpolation, 150, 167
 - Definitive layout, 9
 - Degree of twist, 133
 - Delaunay triangulation, 8
 - Descriptive geometry, 17, 18
 - Design cycle, 198
 - Design evaluation, 196–197, 199, 201
 - Design intent, 225
 - Design process, 73, 225
 - Design prototypes, 191, 192, 194–197
 - Design validation, 191
 - Deviation, 32
 - Diagrams, 4
 - Die casting, 199–201
 - Diffusivity, 185
 - Digital differential analyzer (DDA), 145, 147–152, 167
 - Dimensionality, 226, 227
 - Dimetric, 24
 - Direct stiffness method, 27–31
 - Dirichlet tessellation, 8
 - Distributed mass, 2
 - Dithering, 179–182
 - Documentation, 9
 - Drafting errors, 225
 - Dynamic analysis, 198
 - Dynamic models, 198
- E**
- EDM, 201
 - Elastic curve, 97, 112
 - Elastic modulus, 224
 - Electromagnetic spectrum, 218
 - Electrostatic deposition, 192
 - Element type, 226, 227, 240, 255
 - Equidistance, 229, 230
 - Euler angle rotation, 63, 64
 - Evaluated form, 79
 - Exterior view, 21
- F**
- Fatigue strength, 198
 - Feature control frames, 28
 - Fillet lines, 77, 81, 83
 - Finite element method (FEM), 1–4, 7, 8, 11, 14, 15–16, 21, 64
 - First-angle projection, 21–23
 - Fixed angle rotation, 62, 63
 - Flat face representation, 79
 - Flat shading, 182
 - Flatness, 42
 - Flaws, 45
 - Floating point, 144, 145
 - Floor, 145
 - Force equilibrium, 229, 239, 247, 248, 251, 254, 257, 271, 272
 - Form tolerance, 42
 - Fortus, 209–211
 - Frame rate control (FRC), 178, 179
 - Free body diagram, 17, 22, 27
 - Full sectional view, 21

Function verification, 196–199, 201
 Functional verification, 197
 Functional models, 196
 Fundamental deviation, 33
 Fused deposition modeling (FDM), 209

G

Gaspard Monge, 17
 General curved surface, 121, 122, 130–142
 General oblique, 25
 Geometric constraints, 98, 99, 114, 133
 Geometric model, 3, 6, 17
 Geometric specifications, 17
 Geometric tolerances, 40
 Geometry decomposition, 8
 Global coordinate, 54, 55, 67, 68, 90,
 157, 242, 251, 261
 Global stiffness matrix, 250, 252, 258,
 260–264, 267, 268,
 272, 275, 281
 Gouraud, 171, 183–184, 188
 Gouraud shading, 183, 184, 188
 Gouraud's illumination, 171, 183
 Green strength, 202
 Grid-Based approach, 8, 10
 Ground's eye view, 26, 27
 Gusset, 204

H

Half sectional view, 21, 22
 Hand sanding, 220
 Heat transfer, 217
 Hermite spline curve, 97, 105, 110, 112, 114,
 131, 135, 138, 140, 141
 Hidden line removal, 74, 152, 161
 Hidden surfaces removal, 78
 Hole & shaft tolerancing, 32–40
 Homogeneity, 198
 Homogeneous boundary
 condition, 23
 Homogeneous transformation, 59
 Hook's law, 224, 237,
 254–256
 Horseshoe, 177
 HSL, 172
 HSV, 172
 Hue, 172, 173, 183
 Hue-Intensity-Saturation, 183
 Human's eye view, 25, 26
 Hybrid modeling, 81, 82

I

Ideate, 6, 7
 Illumination model, 171, 172, 182–188
 Imaginary object, 18
 Imaginary plane, 18
 Inclusion test, 153, 162
 Indeterministic, 245
 Index, 112, 239, 243
 Infiltration, 202
 Influence, 112–114, 186, 225, 238
 Ink-jet printing, 205
 Inspection, 29, 31, 179, 192, 200
 Interchangeability, 27
 Interface, 217
 Interference, 32, 34, 37, 39, 46
 Interference fit, 34, 37, 39, 46
 Internal view, 21
 International Commission on Illumination, 175
 International tolerance (IT) grade, 33
 Interpretation, 17, 74, 75
 Inverse transformation, 61
 Investment casting, 220
 Irradiance, 171
 Isometric, 24

K

Kinematic model, 197, 198

L

Lambert, 183
 Lambertian mode, 186
 Laminated object manufacturing
 (LOM), 202, 207–208, 212
 Laminating, 199, 202, 207
 Language of designer, 3
 Laplacian smoothing, 13, 14
 Laser-curing, 215–217
 Laser-trimming, 207
 Lay, 45, 195
 Layer connectivity, 219
 Layered manufacturing, 195
 Layer thickness, 203, 205, 206, 217, 219
 Least Material Condition, 29, 30
 Lee's methodm, 230
 Light illumination, 182
 Lightness, 172, 173
 Limit forms, 27
 Limits of tolerance, 32
 Line fits, 32, 34
 Liquid binder, 205

Local control, 104, 105, 112, 114
 Local coordinate, 52, 54, 55,
 95, 260
 Local stiffness matrix, 240, 242, 247,
 250, 252, 258
 Location tolerance, 40, 41
 LOM, 202, 207, 208
 Loop feedback, 150
 Loose pattern molding, 199
 Lower deviation, 33
 Luminance, 173
 Luminance-chrominance, 177
 Lumped mass, 223, 224

M

Magnetostatic deposition, 192
 Manifold modeling, 81
 Manual drawing, 23, 208, 227, 235
 Manufacturing cell, 192
 Manufacturing process, 45, 192, 194–196,
 199–202, 217, 219
 Mapped element approach, 8
 Mapping, 59–61
 Material handling, 192, 195, 198, 209
 Matrix approach, 237–238
 Maximum Material Condition (MMC), 28
 Mesh, 1, 4–14
 Mesh density, 227, 229, 235
 Mesh generation, 1, 4, 8, 13
 Metal laminates, 202, 207
 Minnaert, 183
 Mold tooling, 199
 Mongian projection, 4, 18
 Monochromatic, 175
 Monomeric styrene, 218
 Monomers, 218
 Multi-Sim, 2

N

Nodal displacements, 255, 257, 260,
 276, 277
 Node generation, 226–229, 232
 Nominal size, 32, 35
 Non-homogeneous approach, 81,
 245, 248–249
 Nonhomogeneous boundary
 condition, 26
 Nonspectral purples, 177
 Nowhere negative, 175
 NURBS, 130
 Nylon, 209

O

OBJET, 209
 Oblique projections, 23–25, 27
 Oligomeric acrylates, 218
 Oligomers, 218
 Ordered dithering, 180
 Oren-Nayar, 183
 Orthogonal matrix, 54, 95, 262
 Orthographic projection, 18, 19, 21–23,
 25, 26
 Orthographic views, 21
 Orthonormal, 262
 Overlapping, 74, 80, 159–162, 230, 233

P

Pahl and Beitz's proposal, 7
 Parametric design, 51
 Parametric equation, 93, 98, 162
 Parametric line, 95
 Part orientation, 217
 Patterns, 3, 196
 Perpendicularity, 44
 Perspective projection, 26, 27, 70
 Perspective sketches, 23
 Phong, 171, 183–188
 Phong shading, 183, 184, 188
 Phong's illumination, 171, 183–188
 Photo-curing, 218
 Photodetectors, 174
 Photoinitiators, 218
 Photopolymer, 191, 192, 203, 205, 216–219
 Photosculpture, 191
 Pictorial drawing, 19
 Pictorial projection, 19, 20, 23–27, 45, 74, 75
 Pierre Bezier, 112
 Planar geometric projections, 17
 Planar truss, 236
 Polygon clipping, 143, 159
 Polygon filling, 143
 Polyhedron, 230
 PolyJet, 209
 Polymerization, 192, 201, 203, 216, 218–220
 Polynomial spline curve, 97
 Poorman's algorithm, 155
 Potency, 174
 Post-cure, 202, 220
 Postprocessing, 3
 Preferred sizes, 37
 Preliminary layouts, 8, 9
 Preprocessing, 3
 Primaries, 25, 26, 35, 99, 172, 174–177, 237
 Primitives, 80, 81, 83, 144

- Process planning, 195
- Production planning, 192
- Productivity, 226
- Pro-E, 1
- Profile tolerance, 43

- Q**
- Quadrilateral, 4
- Quality control, 179, 192
- Quad-tree, 11, 12
- Quaternion interpolation, 188

- R**
- Radiance, 171, 173, 185
- Rapid die casting, 200
- Rapid plaster, 199
- Rapid prototyping, 5, 194
- Rapid Prototyping & Manufacturing, 192
- Raster graphics, 144
- Raster grid, 144
- Reaction moment, 236
- Redundancy, 215
- Reflectability, 171, 172, 182, 183, 185, 186, 188
- Rendering, 188–189
- Repetitive drawing, 20
- Representation, 3, 75–80
- Resolidification, 202
- Retina, 174, 180
- Rigid body, 224, 226, 236
- Room temperature vulcanization, 199
- Rotational matrix, 53, 54, 56, 57, 60, 63, 64
- Roughness, 45
- Roughness height, 45
- Roughness width, 45
- Roughness width cutoff, 45, 46
- RP&M, 192, 193, 195–199, 201–208, 212–215, 219, 220
- RTV mold, 200
- Ruled surface, 127
- Runout tolerance, 44

- S**
- Saturation, 172, 173
- Scaling factor, 94, 95
- Scan-conversion, 144
- Scanning method, 153, 154
- Sectional view, 21, 22
- Selective curing, 215
- Selective laser sintering, 202, 212
- Selective solidification, 202
- Set-theoretic mode, 80, 83
- SFF, 192, 215–219
- Sense of realism, 27
- SGC, 204
- Shear force, 254, 255, 262
- Shimada's method, 227, 229
- Singularity, 94, 226, 244
- SLA, 199–201
- Smooth shading, 183
- Soft tooling, 220
- Solid freeform fabrication, 192
- Solid ground curing, 204–205
- Solid modeling, 79
- Solidworks, 1
- Space truss, 236
- Spatial distribution, 187
- Spatial dithering, 179, 187
- Specification, 8, 9
- Spectral response, 174
- Spectrum, 172, 174, 176, 177, 218
- Specularity, 187
- Spherical linear interpolation, 188
- Spring assemblage, 21
- Standard sizes, 35
- Standardization, 17, 19, 27, 32, 35, 39, 201, 213–215, 249
- Static equilibrium, 239, 240
- Stereolithography, 203–204, 216
- Stiffness, 18–19, 27–42
- Stiffness matrix, 225, 237–245, 249–264, 267, 268, 270, 272, 275, 279, 281, 283
- Stimuli, 176
- STL, 212–216
- STL file format, 212–215
- Straightness, 27, 42, 45, 93, 94, 112, 122, 126, 144, 236
- STRATASYS, 209
- Stress, 42–44
- Stress models, 257, 264–267
- Stress/strain analysis, 198, 225, 232, 236, 243, 254, 256
- Subject polygon, 159
- Sum of angle, 153
- Superposition, 27–31
- Support structures, 203, 204
- Surface conditions, 20
- Surface finish, 27, 199
- Surface modeling, 51, 73, 75–78, 82, 212, 214
- Surface normal, 155, 171, 185, 213

Surface representation, 76
 Surface of revolution, 42, 122
 Surface texture, 19, 45, 46
 Surface theory, 120–142
 Swept surface, 77
 Symmetry tolerance, 42
 Symbols, 19, 20, 28, 29, 31, 37, 38, 40,
 41, 53, 64, 237
 System engineering, 19, 32, 224

T

Tangent vector, 77, 98, 99, 101, 102,
 104, 106, 114, 117, 130, 132,
 133, 136, 138, 148
 Tangential line, 98, 99, 101, 102,
 104, 106, 114, 117, 130, 132,
 133, 136, 138, 148
 Temporal dithering, 178, 179
 Tensile stress, 255
 Tetrahedral, 226, 227, 230
 Thermal models, 200, 207, 209
 Thermo plastic, 209
 Third-angle projection, 20–23
 3-D digitizer, 193
 3-D model, 18, 51–52, 62, 73, 79–81,
 93–94, 112, 143, 212, 214
 3D Printing, 192, 205–206
 T-Junctions, 184
 Tolerance, 7, 27–45, 32–40, 34–39, 39–40,
 40–41, 44–45
 Tolerance methods, 28
 Tolerance specification, 27, 29,
 30, 33, 34, 43
 Tolerance verification, 45
 Tolerance zone, 33
 Tool-less process 195
 Topography 191, 192
 Topology decomposition, 8
 Transform equation, 62
 Transformation arithmetic, 61–62
 Transformation matrix, 157
 Transition, 32, 34, 36
 Transition fit, 34, 36
 Triangular elements, 226, 227, 230
 Trimetric projection, 24
 Tristimulus, 174

Truss, 2, 14–19, 31–64
 TurboCAD, 1
 Type of fits, 33–35, 37, 38

U

Ultem, 209
 Undercut, 203, 204
 Unevaluated form, 80, 83
 Unilateral, 27
 UNISURF, 112
 Upper deviation, 32

V

Vanishing points, 26, 27, 69
 Variation diminishing, 114, 117
 Verbosity, 215
 Virtual assembly, 51, 103
 Virtual scene, 172, 179
 Visible surface, 143, 155, 156
 Visual aids, 196
 Voronoi diagram, 8–10

W

Wavelengths, 172–177, 219, 220
 Waviness, 45
 Waviness height, 45
 Waviness width, 45
 Weiler-Atherton, 159
 Wireframe, 73–75, 78
 Wireframe geometry, 51, 73–76, 78,
 81, 82, 135, 152, 212
 Woodward, 79
 Worm's eye view, 26, 27

Y

Young's modulus, 257

Z

Z-buffering, 143, 156
 Z-buffering algorithm, 143, 156
 Z-clipping, 143, 162
 Zenith angle, 171–172