1 Introduction

1.1 Finite Element Analysis and Its Procedure

While there is no "standard" definition of finite element method (FEM) or finite element analysis (FEA), the general consensus in the literature is that finite element analysis is a general purpose computational technique for obtaining numerical solutions to mathematical models of physical problems. Many physical problems can be described using a (or a set of) mathematical equation(s).

Compared to other numerical methods, the most significant advantage of FEM is its versatility, which lies in two aspects: (1) it is versatile for arbitrary complex geometries, (2) it is versatile for a wide range of physical problems including structural mechanics, heat transfer, fluid dynamics, electromagnetics, and problems in many other areas. The versatility of the method can be attributed to a few characteristics of the method:

- It has a well-established mathematical foundation. The method itself is closely related to the fundamental problems of variational calculus, that is, finding a unique function that minimizes a functional that has this function as a variable. Since this fundamental problem is essentially the fundamental problem of many engineering problems, a method of finding the solution to this fundamental problem is essentially a method of solving all these engineering problems. Furthermore, its convergence properties and error bounds can be obtained by using functional analysis methods. For these reasons, the behavior and performance of finite element solutions are relatively well understood and the accuracy of the solutions is guaranteed by its convergence properties.
- The finite element discretization, approximation, and numerical integration schemes are simple yet powerful numerical schemes that work well on irregular geometries. The implementation of finite element models into computer codes is straightforward and can easily be modularized, making the software development tasks less complex.
- The compact support of the shape functions enables small interaction distances among the nodes, i.e., the nodes in an element are only affected by those in the neighboring elements. This property leads to very sparse coefficient matrices in the linear systems generated by the finite element discretization. In addition, in most cases, the coefficient matrices are symmetric due to the "bilinear" form in the finite element formulation. Linear systems with sparse and symmetric coefficient

matrices require small memory storage and can be solved efficiently by using special algorithms suitable for symmetric matrices.

Due to the simplicity and versatility of the FEM, since the first commercial code NASTRAN (short for "NASA structural analysis") released in 1969, FEA techniques and tools have been quickly adopted and widely used in many of the manufacturing and consumer goods industries. Among those, the automotive, aerospace and defense, and electrical and electronics industries are the ones whose product design and development rely heavily on FEA. In parallel, the FEA software and service sector has grown into a billion dollar industry itself. It has been forecast that the global FEA software market is to expand at a compound annual growth rate of 9.1% from 2016 to 2024 (Goldstein 2019). The market is anticipated to reach USD 2.6 billion by the end of 2024.

So what does FEA involve exactly? When we talk about FEA, depending on the context, we may actually refer to very different scenarios. There is a very large difference in the meaning of "FEA" when a design engineer is performing a finite element design analysis for an engineering design of a product (e.g. an auto part), and when a software development engineer is conducting finite element analysis for a class of engineering problems for the purpose of developing finite element models and implementing a computational solver as the final product of his/her line of work. Figure 1.1 illustrates this difference. In essence, the design engineer is a user of FEA software tools. A design engineer is only concerned with how good his/her design of a physical product or solution to a particular engineering problem is in terms of performance and cost. In comparison, the FE developer focuses on the finite element method itself, works on building, mathematically and numerically, and implementing FE model(s), and typically deals with not just one but a class of engineering problems. In other words, the FE developer develops finite element models for a class of engineering problems and implements them into a finite element software package for design engineers to use. From this point of view, this book is for "FE developers." We discuss the basic theories, principles, techniques, and methods that constitute the foundation and building blocks of the finite element method. The intended audience of this book are senior undergraduate and graduate students who would like to become FE developers or want to acquire an under-the-hood understanding of FEM, as well as junior FE development professionals who need a systematic overview of the method. This book does not discuss how to use commercial FE packages to perform FEA for various engineering problems. Instead, we focus on the mathematical formulation, computational modeling, and implementation aspects of the FEM for several classes of engineering problems, namely linear elasticity, heat transfer, and fluid dynamics problems.

Broadly speaking, a complete FEA procedure can be divided into (1) formulation and pre-processing, (2) solution, and (3) post-processing stages. Here we use an example to briefly illustrate, from an internal point of view, a typical FEA procedure for solving a linear elasticity problem. Figure 1.2 shows a thin plate with holes and notches subjected to uniformly distributed tensile loads at its left and right edges. Assuming the dimensions, material properties, and loads are given, and the structure

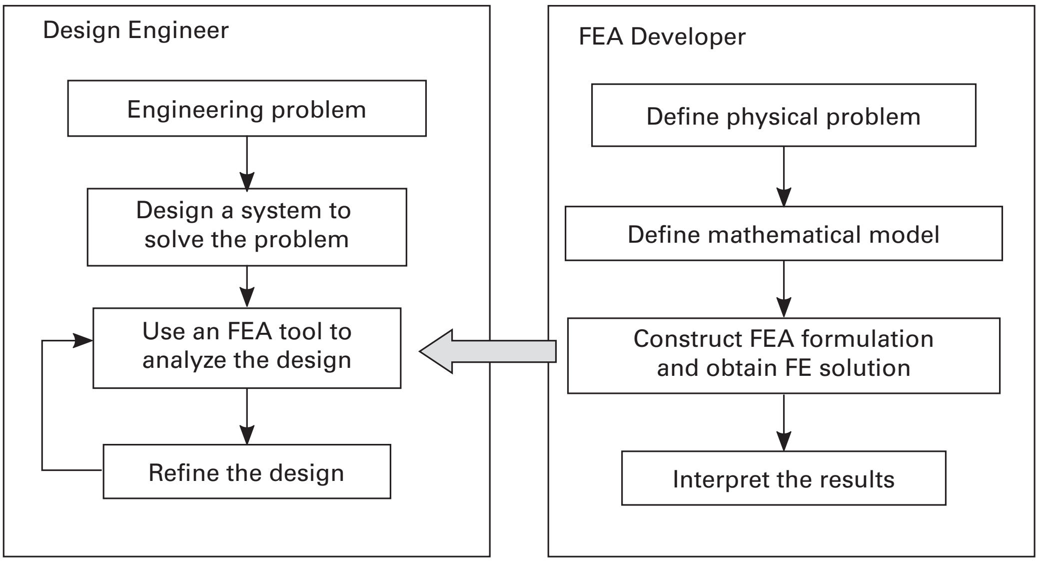

Figure 1.1 The meaning of FEA for a design engineer and an FE developer.

is under isothermal condition, the objective of FEA is to obtain the deformation and stress distribution of the plate.

Before an analysis strategy and modeling approach are determined, we first need to have a clear (and correct) definition of the physical system and problem. In this step, we would ask ourselves a set of questions:

- What type of system is this: a single or a multi-object system?
- What type of physical behavior/phenomenon are we investigating? Is it a mechanics, heat transfer, electromagnetics, acoustics, or multiphysics problem? Is it an equilibrium, steady state, or transient/dynamic problem? Is it a linear or nonlinear problem?
- What is the dimensionality of the problem: 1-D, 2-D or 3-D?
- What are the relevant physical properties of the object(s) in the system?

The answers to these questions give the definition of the physical system under consideration. In our example shown in Fig. 1.2, the system is a single structure system. We are interested in the equilibrium mechanical behavior of this structure and would like to obtain its deformation and stress profiles under the given loading condition. The structure can be treated as a two-dimensional problem due to its geometric feature (thin plate) and loading condition (in-plane loading). The plate is symmetric with respect to both x- and y-axes. If the given load is small, small displacement is expected and the theory of linear elasticity applies. The required material properties include Young's modulus and Poisson's ratio.

Once the physical problem is defined, the next step is defining a mathematical model to describe the physical behavior of the system. A well-defined mathematical model should faithfully represent the physics of the given problem and should contain

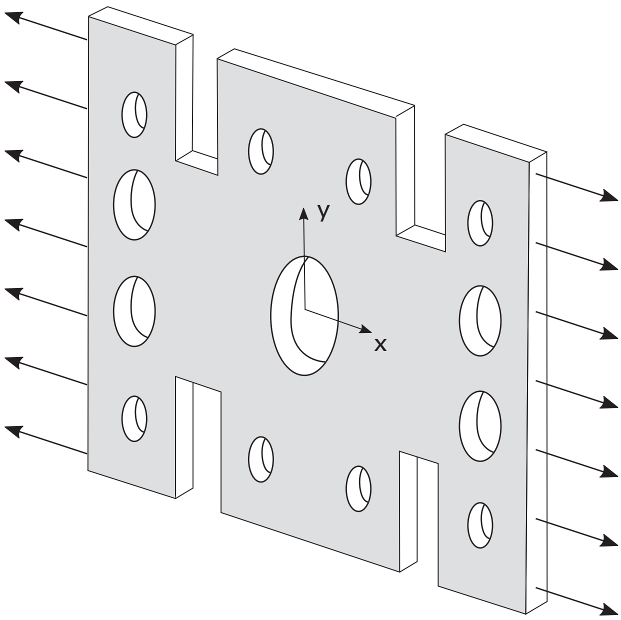

Figure 1.2 A thin plate is subjected to uniformly distributed loading on its left and right edges.

proper geometric information of the domain, loads, and boundary conditions. The mathematical equations (called governing equations) along with the loading and boundary conditions should be sufficient to guarantee a unique solution to the model.

For our example, due to the symmetry of the geometry and loads as shown in Fig. 1.2, the deformation of the plate is symmetric about the x- and y-axes. Therefore, it is only necessary to calculate the deformation and stress of a quarter of the plate and the rest are just the mirror images of the quarter. Figure 1.3 shows the computational domain (denoted as Ω) which is a quarter of the plate, geometric boundary (denoted as Γ), boundary conditions, and loads. Note that the roller-type boundary conditions at the bottom and left edges of the computational domain are the results of the symmetric displacement field along the x- and y-axes. That is, only x-displacement along the x-axis and only y-displacement along the y-axis are allowed if the displacement field is symmetric along the x- and y-axes. As the physical problem is defined as a linear static elasticity problem, the governing equation is the well-known differential equation of equilibrium

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{b} \qquad\qquad in \;\; \Omega$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\mathbf{b}$ is the body force vector. The constitutive relation is the generalized Hooke's law

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon}$ is the strain vector and $\mathbf{C}$ is the material stiffness matrix. The boundary conditions are

$u_x = 0$ along the left edge of the domain
$u_y = 0$ along the bottom edge of the domain
$\sigma_{xx} = P$ along the right edge of the domain.

The governing equations and boundary conditions altogether give a complete set of mathematical equations describing the mechanical behavior of the plate and having a unique solution.

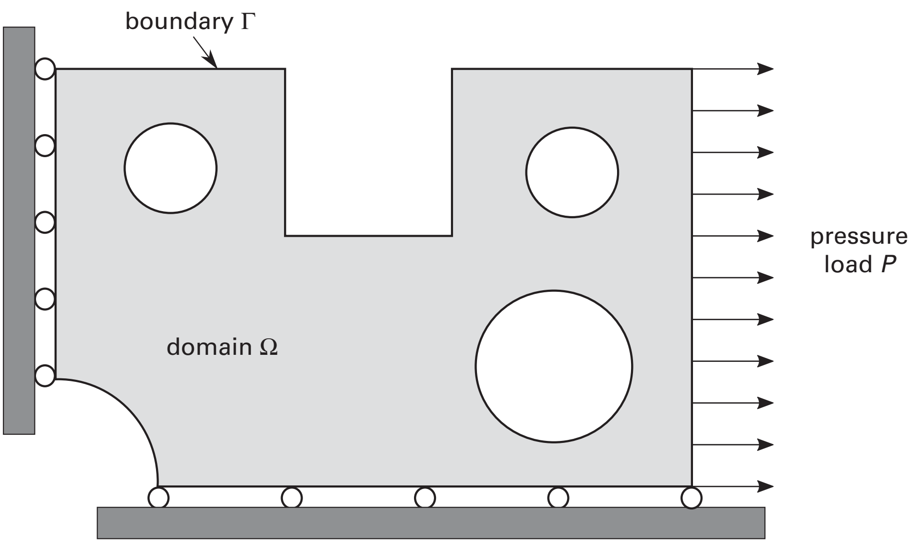

Figure 1.3 Domain of the computational model of the plate.

The steps described above are categorized to be in the pre-processing stage. Next, in the solution stage, we first convert the differential governing equations into one or several integral equations called weak form. The computational domain is discretized into a set of non-overlapping small regions of primitive shapes called elements. In each element, the element vertices and sometimes also a set of points on the element edges are set to be the “nodes” on which the unknown physical variable(s) of interest is (are) to be calculated. This geometric discretization process is called meshing. A mesh of the plate in our example is shown in Fig. 1.4. Then, the unknown physical variables are approximated over each element by using functions of simple forms. Such functions are called shape functions or basis functions. The approximation process is referred to as the finite element approximation. Substituting the finite element approximation into the weak form and discretizing the weak form according to the geometric discretization, the weak form can be converted into a set of algebraic equations that can be solved numerically by using a computer.

In the solution stage, the process of (1) converting the differential governing equation(s) into weak form, (2) discretization of the computational domain and the weak form, and (3) approximating the unknown variables over the elements is referred to as developing the finite element formulation of a mathematical model. Having obtained

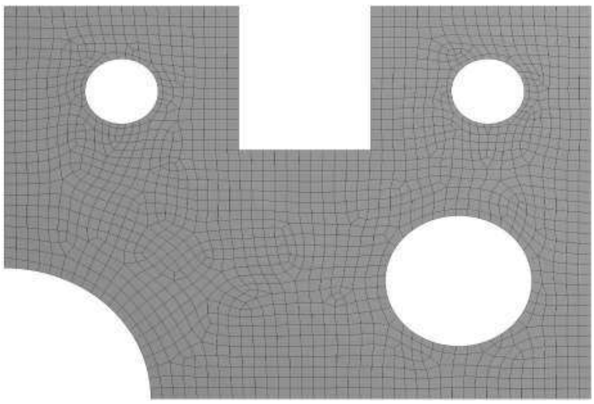

Figure 1.4 Meshing the computational model.

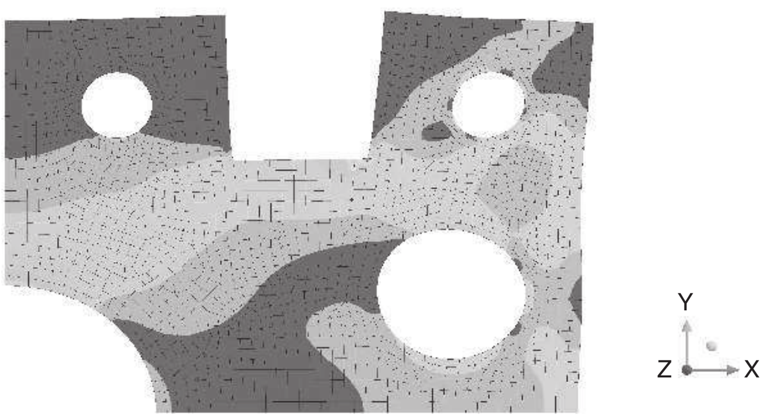

Figure 1.5 Von-Mises stress distribution in the deformed plate.

the discretized finite element formulation, by using numerical analysis methods and algorithms such as numerical integration and numerical solution of simultaneous algebraic equations, one can structure and code a program that uses the input information of geometry, physical properties, loads, and boundary conditions, and calculates the desired unknown physical variables over the nodes and elements. For our plate example, the primary unknown physical variables are the displacements. By solving the simultaneous algebraic equations obtained, the displacements of the nodes are calculated. Adding the displacements to the original positions of the nodes, the deformed shape of the plate is obtained, as shown in Fig. 1.5. After that, by using the now-known nodal displacements, strains can be obtained by calculating the derivatives of the displacements, and stresses can be calculated by using Hooke's law. The calculated strains and stresses can be visualized using contour plots. For example, the von-Mises stress distribution over the plate is visualized in Fig. 1.5. The calculation of the secondary physical quantities such as the strains and stresses in this example, and the visualization and interpretation of the numerical results all belong to the post-processing stage of FEA.

1.2 A Brief History of the Finite Element Method

The finite element method, whether in its early days or as in its current state, was not invented or developed by a single individual. The method was formed gradually by many pioneers from two sides: the mathematics side and the engineering side. The major developments before the FEM became a general purpose practical analysis tool are depicted in Fig. 1.6.

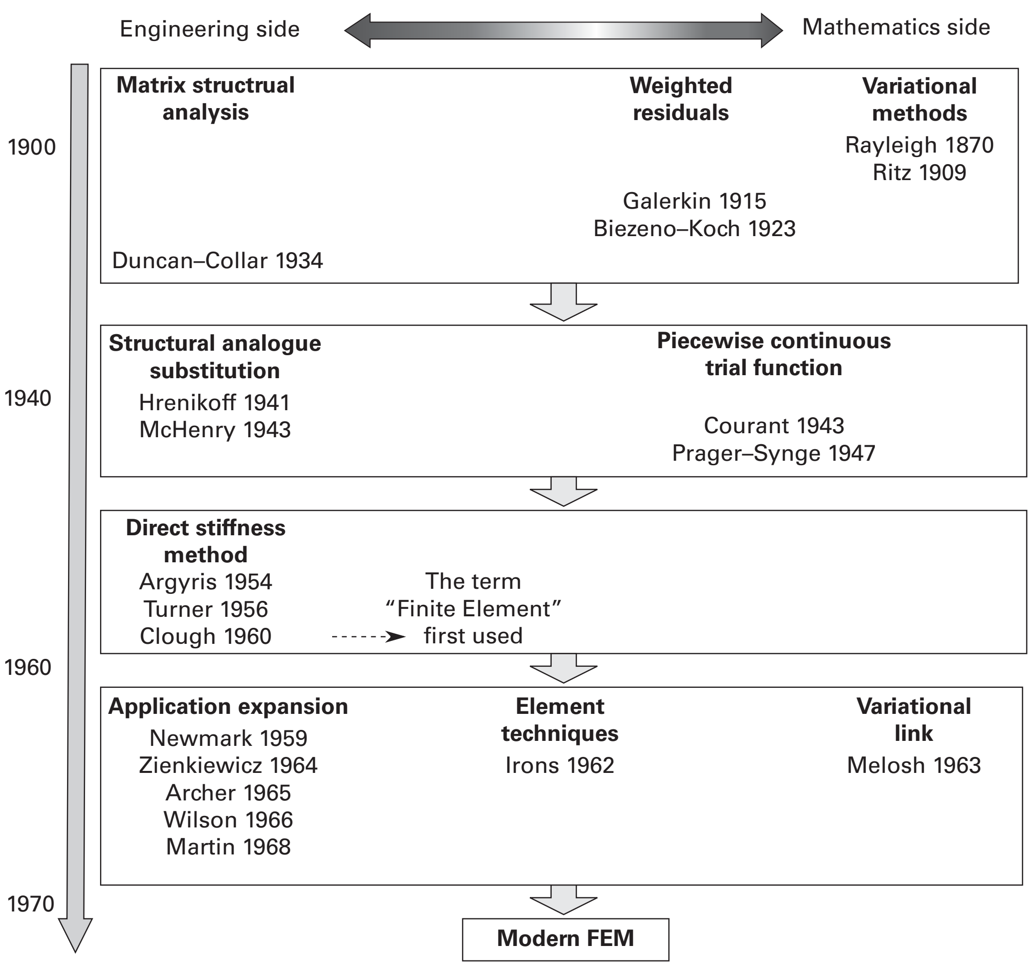

Figure 1.6 A brief history of the finite element method.

On the mathematics side, much earlier than the emergence of the concept of finite element, two classes of methods were developed for solving differential equations: the method of weighted residuals and the variational method. In the weighted residual method, the unknown solution is expressed as a linear combination of a set of "trial" functions. The coefficients of the trial functions are obtained by requiring a weighted integral of the residual to be zero. Depending on the choice of the weight function, the weighted residual method can be divided into several sub-categories: collocation, Galerkin, least squares, sub-domain methods, etc. As a kind of weighted residual method, the least squares method was originated by Gauss in 1795 for least squares estimation. The least squares method uses the derivatives of the residual with respect to the unknown coefficients of the trial functions as the weight functions. In comparison, Russian engineer Galerkin (1915) used the trial functions themselves as the weight function, leading to the Galerkin weighted residual method. The sub-domain weighted residual method was first employed by Biezeno and Koch (1923) for structural stability problems. A different type of method for solving differential equations is based on the variational principle. The variational method was first used by Rayleigh (1870) and later by Ritz (1909). While developed independently, there is a close relationship between the weighted residual method and the variational method. In fact, as will be shown later in this book, the Galerkin weighted residual method is equivalent to the variational method in many cases. Although the variation and weighted residual methods enabled approximated solutions to a variety of types of differential equations, their practical use was hindered by the requirement that the trial functions must span the entire domain and satisfy the boundary conditions. The application of the methods was difficult in the analysis of large structural systems.

On the engineering side, the pioneers were structural engineers. Starting from the early 1930s, there was a trend to represent large structures with complex geometry by using smaller and simpler structures connected to each other. A representative method is the matrix structural analysis approach developed by Duncan and Collar (1934). In the early 1940s, Hrennikoff (1941) and McHenry (1943), separately, developed lattice analogies to represent continuum structures using connected beams and bars. The lattice analogy only achieved limited success. It did not draw much attention and investment until the then called direct stiffness method (DSM) was generalized by Turner et al. in 1956 (Turner 1956). Also in the 1940s, Courant (1943) used piecewise linear interpolation over 2-D triangular elements as Rayleigh–Ritz trial functions. Unfortunately, his work was not followed up immediately as it was advertised as "generalized finite differences" in his article. In 1947, Prager and Synge (1947) also proposed a concept of regional discretization similar to Courant's work. During the period of 1952 to 1964 at Boeing, Turner oversaw the creation and expansion of the DSM, a procedure that constructs the stiffness matrices for beam, truss, and two-dimensional triangular and rectangular plane stress elements, and assembles them to obtain the global structure stiffness matrix. The success of the method with the use of digital computer in 1950s prompted further development of the element stiffness based methods and resource commitment from Boeing. Influenced by Turner's work, Argyris (1955) proposed matrix structural analysis methods using energy principles. In continuing Turner's work, Clough (1960) coined the term "Finite Element" in 1960.

The decade following 1960 was the golden age of FEM development. A series of developments in theories, techniques, and methods was completed during this time, greatly enhancing the applicability and performance of the finite element method. Melosh (1963) showed that the conforming displacement based models are a form of Rayleigh–Ritz method with the minimization of potential energy, and systematized the variational derivation of stiffness elements. This started the convergence of the Argyris' dual formulation of energy methods and the DSM of Turner, and also marked the beginning of the method to solve nonstructural applications. In terms of element techniques, Irons, Zienkiewicz, and others (Ergatoudis, Irons, and Zienkiewicz 1968) invented isoparametric elements, shape functions, and the patch test. For expanding the capability of the FEM for engineering applications, Newmark (1959) developed time integration schemes for structural dynamic analysis, and Archer (1965) developed the consistent-mass matrix. Zienkiewicz, Martin, and Wilison all made substantial contributions to expand the FEM's application to a wide range of engineering areas in the 1960s. By mid 1970s, the finite element method had largely evolved into its present form.

1.3 FEA Applications, Software, and Trend

Accompanied by the development of the finite element method itself, the first set of commercial FEA packages were developed in the late 1960s and early 1970s. The well-known ones included NASTRAN, ANSYS, and ABAQUS. The first commercial version of NASTRAN FEA software was released in 1969. The first commercial version of ANSYS software was labeled version 2.0 and released in 1971. The first ABAQUS version was released in 1979.

In its early days, FEA was mainly used in the aerospace industry and had a small footprint in civil engineering. In the 1970s, along with the advances in computer aided design and manufacturing (CAD/CAM) technology, especially the development of mathematical representations of arbitrary curved surfaces, the application of FEA quickly spread into new areas such as the automotive industry. However, before 1980, its application was still largely limited by the high cost of computers and their limited computing power. In 1964, the IBM 360 mainframe computer produced 1 MFLOPS (floating point operations per second) computing power with a price tag of \$2.5∼3M. In 1976, Cray-1, a supercomputer at that time, had 80 MFLOPS computing power at a cost of \$5∼8M. Due to this limitation, large FEA calculations could only be performed by large companies and government agencies such as national labs. Entering the 1980s, the computing power of the large computers grew very rapidly while the cost remained more or less the same. At the same time, much smaller workstations with microprocessors, such as Apollo Computer and Sun Microsystems, appeared. Such systems typically had MFLOPS performance for a cost of \$15∼100K. FEA became more affordable. In the early 1980s, FEM application became popular in the automotive industry.

Starting from the end of the 1980s, the development of smaller high performance workstations and personal computers (PCs) started to grow at a stunning speed. For example, in 1991, a business-class PC with Intel 486/33 microprocessor produces 30 MFLOPS computing performance, and costs around \$4K if equipped with 4 MB of RAM, a 200 MB hard disk and 14 inches display. This trend of exponential growth of computing power known as Moore's law has continued to the present day. Today, the computing power of a 6-core Intel Core i7 8700 desktop processor's scientific computing power reaches as much as 70 GFLOPS. That's almost 900 times more powerful than Cray-1. The price of the Intel processor is under \$400, compared to the \$5∼8M price tag for Cray-1 in 1976. The cost per GFLOPS is cut down further due to the rise of the GPUs (Graphic Processing Unit). For example, NVDIA's GeForce GTX 970 delivers 3,494 GFLOPS processing power for single precision floating point operations. The price of the unit is slightly above \$300 in 2018, which makes the cost per GFLOPs less than \$0.1. Figure 1.7 depicts the 10 orders of magnitude decrease of cost per GFLOPs over the last 60 years.

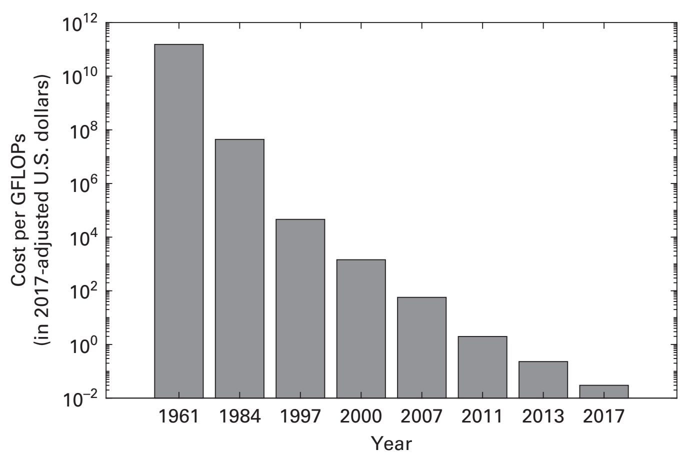

Figure 1.7 The decrease of cost per GFLOPs over the last 60 years.

Accompanied by the explosive growth of the computing power that became easily accessible to engineers and researchers, the modeling capabilities and solving power of FEM kept growing at a fast pace. The application of FEA quickly spread into disciplines and fields outside of structural mechanics. As early as 1960, FEM was applied to heat transfer, fluid mechanics, and then thermomechanical problems. For fluid dynamics problems, FE formulations were developed in parallel with the development of other numerical methods including finite difference and finite volume methods. In 1965, Gladwell (1965) first introduced FEM formulation for acoustics problems. In the early 1980s, the FEM was employed for solving electrostatic and