# Applied Finite Element Analysis

M. E. Barkey

Aerospace Engineering and Mechanics

The University of Alabama

# Course Objectives

- To introduce the graduate students to finite element analysis concepts, methods, and best practices in applications,

- To highlight solution techniques that will be useful in research and industrial applications.

# Course ‘Style’

This will be an applied course, meaning that you will learn how to use computer programs for finite element analysis.

This course is meant to complement a theory based course in which you would learn the mathematical foundations of finite element analysis.

The more you put into the course, the more you will get out of it.   Computers software comes easy for some people, but are more difficult for others.  You will need to put in enough time outside of class to make progress.

# Required Background

- Basic computer knowledge,
- basic course in stress analysis/materials,
- graduate standing.

# FEA Projects

- Use software to complete basic analysis types including:

basics of mesh building
linear static analysis
non-linear material analysis (small deformation)
non-linear material/structural analysis (large deformation)
mode shape, eigenvalue analysis
composite analysis
heat transfer analysis.

# Evaluation/Grades

Succinct written reports approximately every two weeks.

The content and format of the reports will be discussed in class.

Final evaluation (TBD).

# What is FEA?

Finite Element Analysis is a technique in which a structure is sub-divided into a (finite) number of small pieces (elements) that are effectively like springs.

The springs can be
one-dimensional (rods, bars, beams),
two-dimensional (triangle or quadrilateral plates/shells),
three-dimensional (cubes, tetrahedrons), or
special purpose elements (e.g. connector elements).

# What is 'FEA' today?

Today, what is known as FEA is usually part of a 'multi-physics' simulation software package that can combine materials in various phases and length scales with prescribed general kinematic/dynamic behavior.

# The Spring Equation

To motivate our understanding of FEA, it is useful to think of a one dimensional linear spring:

F = k Delta

where F is the force in the spring, k is the stiffness of the spring, and Delta is the displacement of the spring.

(refer to notes on the board)

# The Spring Equation con’t

The equation can be inverted to find displacement if the force is specified:

Delta = F/k

(refer to notes on the board)

# General spring 'element'

It is easy to imagine our spring that is fixed at one end, but let's generalize our spring so that both ends can move or displace.

The ends of the spring are called 'nodes'. The spring itself is called the 'element'.

(refer to notes on the board)

# Multiple Springs

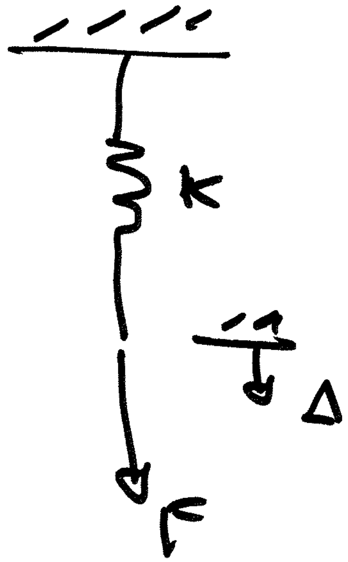
Consider two springs with spring constants k1 and k2 joined together.

With these two springs, we have two ‘elements’ and three nodes.

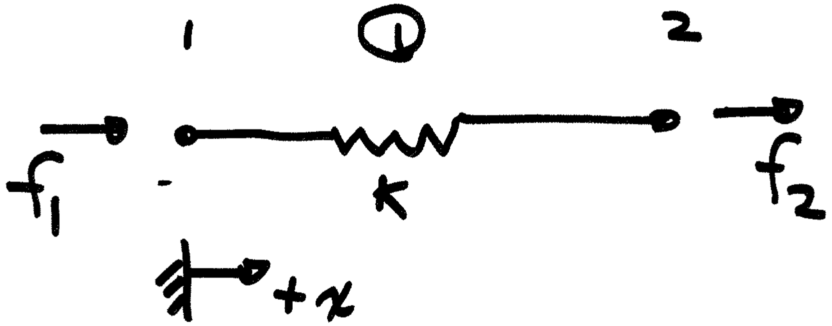
(refer to notes on the board)

# Assemble the stiffness matrix and apply boundary conditions

(refer to notes on the board)




$$F = k\Delta \qquad \Delta = \frac{F}{k}$$




$$\Delta = u_2 - u_1$$

$$f_1 + f_2 = 0 \quad (\Sigma F = 0)$$

$$f_1 = -f_2$$

$$f_1 = -k(u_2 - u_1)$$

$$f_2 = k(u_2 - u_1)$$

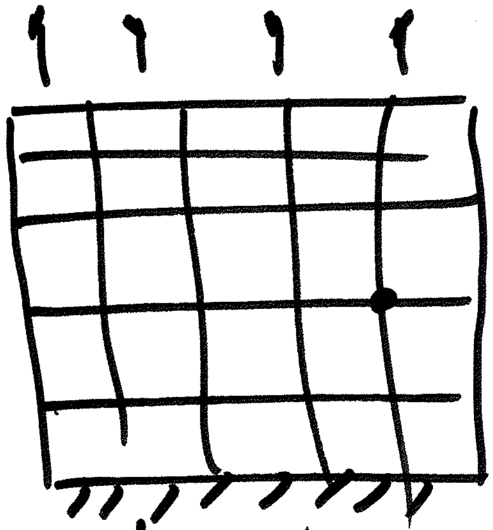
spring eqn to find force

$$f_1 = -ku_2 + ku_1 = ku_1 - ku_2$$

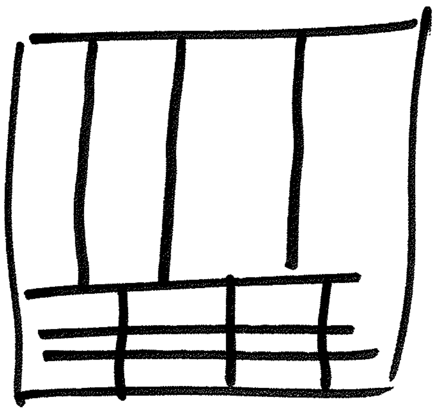
$$f_2 = ku_2 - ku_1 = -ku_1 + ku_2$$

$$\begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix} = \underbrace{\begin{bmatrix} k & -k \\ -k & k \end{bmatrix}}_{\text{Stiffness matrix for a spring element}} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

$$\underline{F} = \underline{\underline{K}}\ \underline{u} \quad \text{in matrix form}$$



Good Meshes are connected to other nodes



NO!

The following slides are from

FEA_FEM.pdf

http://www.cs.ubbcluj.ro/~alibal/Teaching/Simulation

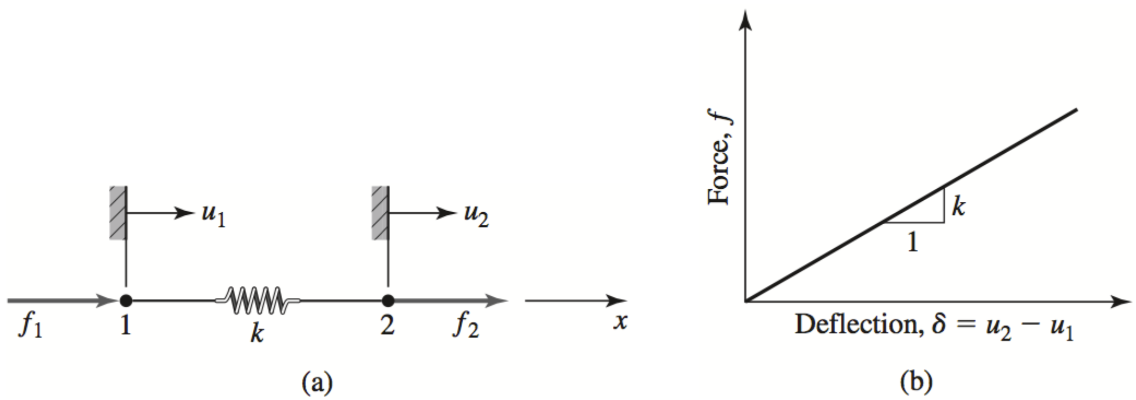
author unknown.

# Stiffness Matrix

The primary characteristics of a finite element are embodied in the element ***stiffness matrix***. For a structural finite element, the stiffness matrix contains the geometric and material behavior information that indicates the resistance of the element to deformation when subjected to loading. Such deformation may include axial, bending, shear, and torsional effects. For finite elements used in nonstructural analyses, such as fluid flow and heat transfer, the term *stiffness matrix* is also used, since ***the matrix represents the resistance of the element to change when subjected to external influences***.

# LINEAR SPRING AS A FINITE ELEMENT

A linear elastic spring is a mechanical device capable of supporting axial loading only, and the elongation or contraction of the spring is directly proportional to the applied axial load. The constant of proportionality between deformation and load is referred to as the *spring constant, spring rate,* or ***spring stiffness k***, and has units of force per unit length. As an elastic spring supports axial loading only, we select an *element coordinate system* (also known as a *local* coordinate system) as an $x$ axis oriented along the length of the spring, as shown.




(a) Linear spring element with nodes, nodal displacements, and nodal forces.
(b) Load-deflection curve.

Assuming that both the nodal displacements are zero when the spring is undeformed, the net spring deformation is given by

$\delta = u_2 - u_1$

and the resultant axial force in the spring is

$f = k\delta = k(u_2 - u_1)$

For equilibrium,

$f_1 + f_2 = 0$ or $f_1 = -f_2$,

Then, in terms of the applied nodal forces as

$\boldsymbol{f_1 = -k(u_2 - u_1)}$

$\boldsymbol{f_2 = k(u_2 - u_1)}$

which can be expressed in matrix form as

$$\begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix} \quad \text{or} \quad [k_e]\{u\} = \{f\}$$

where

$$[k_e] = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix}$$

**Stiffness matrix for one spring element**

is defined as the element stiffness matrix in the element coordinate system (or local system), $\{u\}$ is the column matrix (vector) of nodal displacements, and $\{f\}$ is the column matrix (vector) of element nodal forces.

$$\begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix} = [k_e] \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \qquad \text{with} \qquad [k_e] = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix}$$

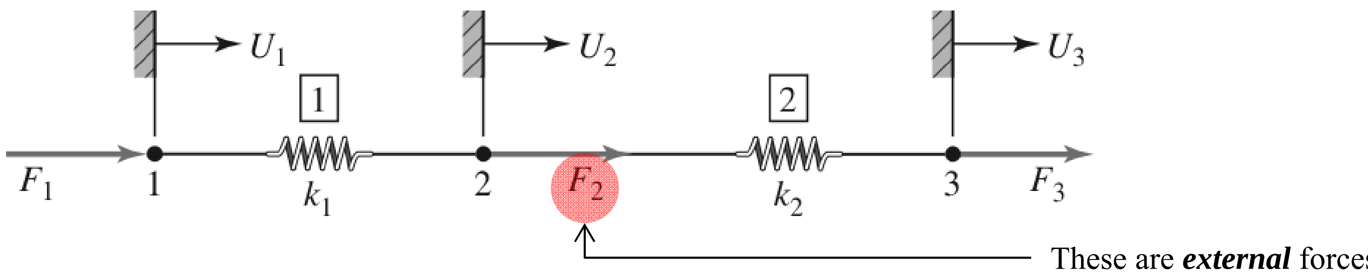
$$\text{known} \quad \{F\} = [K]\ \{X\} \quad \text{unknown}$$

The equation shows that the element stiffness matrix for the linear spring element is a ***2 × 2 matrix***. This corresponds to the fact that the element exhibits ***two nodal displacements (or degrees of freedom)*** and that ***the two displacements are not independent (that is, the body is continuous and elastic)***.

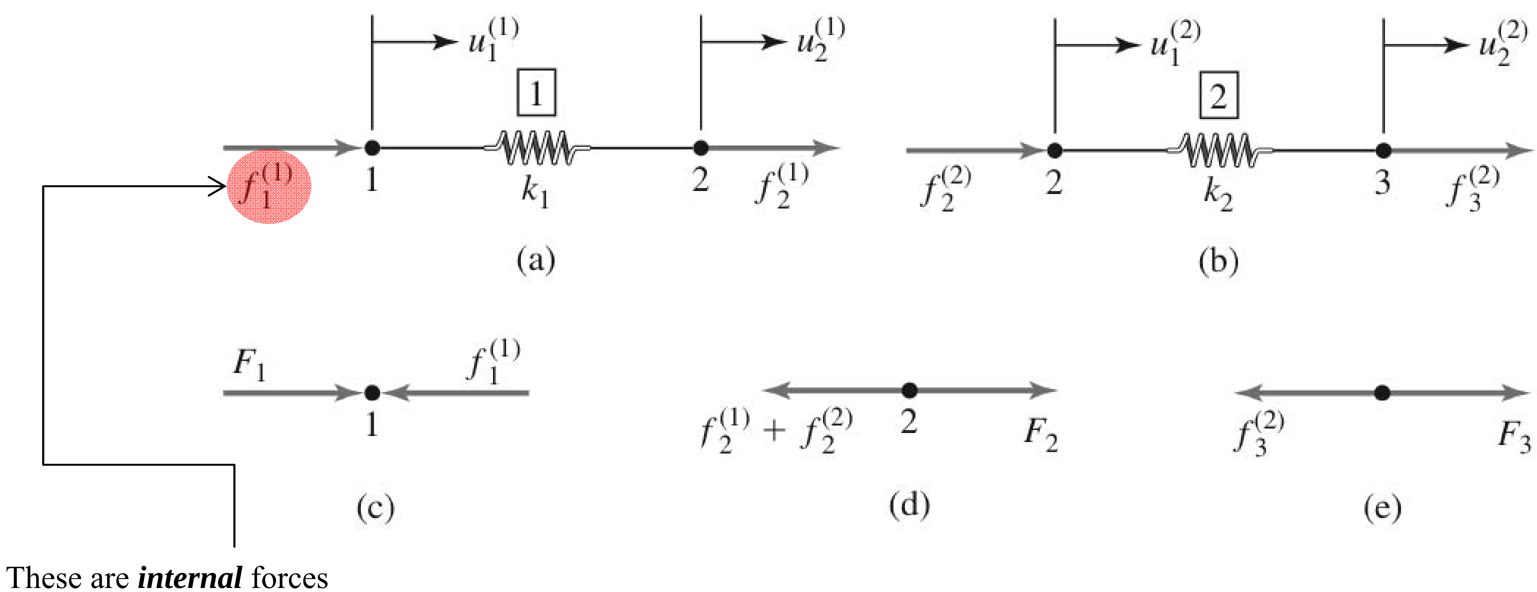
Furthermore, ***the matrix is symmetric***. This is a consequence of the symmetry of the forces (equal and opposite to ensure equilibrium).

Also ***the matrix is singular*** and therefore not invertible. That is because the problem as defined is incomplete and does not have a solution: ***boundary conditions are required***.

# SYSTEM OF TWO SPRINGS




## *Free body diagrams:*



Writing the equations for each spring in matrix form:

Superscript refers to element

$$\begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix} \begin{Bmatrix} u_1^{(1)} \\ u_2^{(1)} \end{Bmatrix} = \begin{Bmatrix} f_1^{(1)} \\ f_2^{(1)} \end{Bmatrix}$$

$$\begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} u_1^{(2)} \\ u_2^{(2)} \end{Bmatrix} = \begin{Bmatrix} f_2^{(2)} \\ f_3^{(2)} \end{Bmatrix}$$

To begin assembling the equilibrium equations describing the behavior of the system of two springs, the displacement ***compatibility conditions****,* which relate element displacements to system displacements, are written as:

$$u_1^{(1)} = U_1 \qquad u_2^{(1)} = U_2 \qquad u_1^{(2)} = U_2 \qquad u_2^{(2)} = U_3$$

And therefore:

$$\begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \end{Bmatrix} = \begin{Bmatrix} f_1^{(1)} \\ f_2^{(1)} \end{Bmatrix}$$

$$\begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} f_2^{(2)} \\ f_3^{(2)} \end{Bmatrix}$$

Here, we use the notation $f^{(j)}{}_i$ to represent the force exerted on element $j$ at node $i$.

Expand each equation in matrix form:

$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ 0 \end{Bmatrix} = \begin{Bmatrix} f_1^{(1)} \\ f_2^{(1)} \\ 0 \end{Bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} 0 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ f_2^{(2)} \\ f_3^{(2)} \end{Bmatrix}$$

Summing member by member:

$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} f_1^{(1)} \\ f_2^{(1)} + f_2^{(2)} \\ f_3^{(2)} \end{Bmatrix}$$

Next, we refer to the free-body diagrams of each of the three nodes:

$$f_1^{(1)} = F_1 \qquad f_2^{(1)} + f_2^{(2)} = F_2 \qquad f_3^{(2)} = F_3$$

Final form:

$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} \qquad (1)$$

Where the stiffness matrix:

$$[K] = \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix}$$

Note that the system stiffness matrix is:

***(1)*** ***symmetric***, as is the case with all linear systems referred to orthogonal coordinate systems;

***(2)*** ***singular***, since no constraints are applied to prevent rigid body motion of the system;

(3) the system matrix is simply ***a superposition of the individual element stiffness matrices*** with proper assignment of element nodal displacements and associated stiffness coefficients to system nodal displacements.

## FEA for multiple (many) elements

$$\{F\}=[K]\cdot\{U\}$$

Array of applied forces (one for each DOF)

Matrix of stiffnesses (DOF x DOF)

Array of displacements ( one for each DOF)

$\{F\}$ is "known" (loads)

$[K]$ is "known" (geometry, material properties…elements)

$\{U\}$ is to be determined (displacements)

This can be solved mathematically using a matrix inversion method

$$\{F\}=[K]\cdot\{U\} \quad\rightarrow\quad \{U\}=[K]^{-1}\{F\}$$

(first nodal quantity)

Once the displacements $\{U\}$ are known, then strains and stresses can be determined:

$$\varepsilon=\frac{\Delta u}{L}$$ (1-D …more complicated for 2-D and 3-D strains)

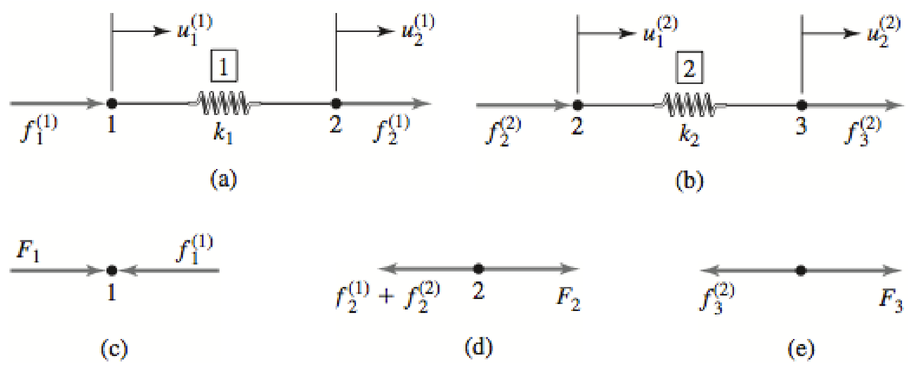
$$\sigma=E\cdot\varepsilon$$

and $FOS=\dfrac{\sigma_y}{\sigma}$

(second nodal quantities)

# Example with Boundary Conditions

Consider the two element system as described before where Node 1 is attached to a fixed support, yielding the displacement constraint $U_1 = 0$, $k_1 = 50$ lb/in, $k_2 = 75$ lb/in, $F_2 = F_3 = 75$ lb for these conditions determine nodal displacements $U_2$ and $U_3$.




Substituting the specified values into (1) we have:

Due to boundary condition

$$\begin{bmatrix} 50 & -50 & 0 \\ -50 & 125 & -75 \\ 0 & -75 & 75 \end{bmatrix} \begin{Bmatrix} 0 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ 75 \\ 75 \end{Bmatrix}$$

# Example with Boundary Conditions

Because of the constraint of zero displacement at node 1, nodal force $F_1$ becomes an unknown reaction force. Formally, the first algebraic equation represented in this matrix equation becomes:

$$-50U_2 = F_1$$

and this is known as a ***constraint equation****,* as it represents the equilibrium condition of a node at which the displacement is constrained. The second and third equations become

$$\begin{bmatrix} 125 & -75 \\ -75 & 75 \end{bmatrix} \begin{Bmatrix} U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} 75 \\ 75 \end{Bmatrix}$$

which can be solved to obtain $U_2 = 3$ in. and $U_3 = 4$ in. *Note that the matrix equations governing the unknown displacements are obtained by simply striking out the first row and column of the 3 × 3 matrix system, since the constrained displacement is zero (homogeneous)*. If the displacement boundary condition is not equal to zero (nonhomogeneous) then this is not possible and the matrices need to be manipulated differently (partitioning).