Finite Element Method (FEAST-SMT)

Finite element method is a very popular numerical method that has found use in finding practical solutions to field problems in solid mechanics, electro-magnetics, thermal engineering etc. Earlier, solving real-life problems involved time and hence, technologies used to evolve at its own 'natural' pace. The contemporary developments in finite element method and technological breakthroughs in digital computers has complemented each other's growth and has led to revolutionary innovations in many fields of science and technology. The sophistication and user friendliness in the modern finite element software has popularized this method but has led many uninitiated users to think the results from it as the final verdict for their problem in hand.This is due to lack of discretion on their part. If asked "What is finite element method?" to many users of the software, the answer would in most of the cases describe the steps involved in operating one of the much available software. Analogously, one may ask why a person should know the combustion cycle or mechanisms of power steering of an automobile while owning it for transportation needs. Well, the question is legitimate in case of an automobile ownership, not so for the high end users like scientists, engineers and technologists using finite element software. This is because they should have the overall grasp of the problem, the solutions, validity/applicability and complete understanding of methods used to arrive at the solution, be it analytical, test or numerical. Rigour should dominate heuristics while comprehending the nature of the problem. If not, the blind choice will result is disastrous consequences in terms of resources, time and at times life.

Although, finite element method can make a good engineer better, it can make a poor engineer more dangerous . The mathematical foundations of this method are to be dwelled into for making rational conclusions while interpreting results. There are several textbooks and peer reviewed scientific publications that are available on the finite element method covering various aspects. These literatures are mathematically formidable but they are not impossible to understand. It is ironic to come across the comments that these tomes of math are "impractical" as these form the very foundations from where any finite element software is built and designed.

Before we try to understand what finite element method is, let us appreciate how modern science works in first place.

We observe several phenomena in nature and try to comprehend it by relating principal quantities involved to the net effect. In other words, the qualitative observations are translated to quantitative mathematical statements. These statements take the form of polynomial expressions, differential equations (ordinary or partial), integro-partial differential equations etc. These expressions are formulated based on some rational hypothesis, which concentrates on the cardinal influencing factors on what is being sought. Example is the Euler-Bernoulli beam. This is based on small deflection hypothesis that essentially means that the differential equation represents/captures linear behaviour and this model should seldom be used to estimate large deflection of beams.

It is important to remember that the mathematical models are only an approximation to reality. If it is possible to measure a physical quantity of interest directly, then all the convoluted mathematics and rational arguments for approximations can be gleefully abandoned. This is the reason why there is no substitute for experimental or test based results. The prohibitive cost is the drawback of tests, which may have its own pitfalls. Verifying mathematical models against real physical behaviour, from either laboratory-based experiments or in-situ observations, is an essential part of the design process .

Finding solution to these mathematical models is the next logical step. Constraints are identified before resorting to solution finding. Wherever or whenever it is possible, closed form solution is the preferred option. Analytical solutions based on variational principles are the resorted to when closed form solutions are impossible. This is true in many situations. But variational methods have their own limitations as it can give solutions only over simple domains. Numerical methods are used for finding solutions when closed form or analytical solutions are inadequate. It has to be remembered that finite element method is a numerical method for solving partial differential equations and basically is piece-wise application of variational methods such as Rayleigh-Ritz or Galerkin method.

When standard commercial packages are used for the analysis, we implicitly accept the element solution without considering the consequences of the choice. Before zeroing into a particular element (solution) the analyst/designer should carefully read the documentation on underlying theories and assumptions that has gone into the element definition. If the chosen element doesn't meet the requirement, then special purpose code has to be developed or shopped for. This process will help the analyst/designer achieve solution to the required degree of confidence.

This theoretical manual primarily dwells on:

- Mathematical principles, which is precursor to the governing relations
- Solution of eigen value and steady state problems
- Element library
- Constraints
- Material constitutive laws
- Solvers and their algorithms

The explanation though brief, it does not compromise on the essential points.
A free trial version of FEAST-*SMT*is available at a dedicated website, http://feast.vssc.gov.in, which can be downloaded, along with tutorials to get a feel of the tool. The solver is supported by a rich graphic user interface based pre/post processor to create numerical models and map the results. It takes years to master the use of any finite element software for achieving intended objectives of an analyst.
FEAST-*SMT* is validated using standard NAFEMS (National Agency for Finite Element Methods and Standards) benchmark problems. In addition to this, the development team has solved many of ISRO's structural engineering problems using PreWin/ FEAST-*SMT*as test cases. The results obtained using the present software is comparable with those obtained from industry standard proprietary software.

Euler-Bernoulli beam

$EI\frac{d^4w}{dx^4} = q(x)$; EI is the flexural rigidity; w is the transverse deflection; $q(x)$ is the loading function; $0 \leq x \leq L;$ Where, L is the length of the beam.

General mathematical overview

Solid mechanics problems can be solved using methods such as: displacement or stress function approach. In finite element method, displacement approach is preferred as the compatibility conditions are satisfied *a priori* and ease of implementing into general purpose software. Certain class of problems cannot be solved using this approach, example: materials with Poisson's ratio=0.5.
From Newton's laws of motion it can be inferred that the choice of reference coordinate is arbitrary. In most of the situations the mathematical description of the mechanical systems results in a set of coupled equations that demands a simultaneous solution to determine the unknown variables. This is due to arbitrariness of reference coordinates. Expressing the relations in terms of generalised coordinates overcomes this shortcoming as the equations are rendered independent. In many numerical solution techniques, including finite element method, a continuum is considered to be represented by a set of discrete system. Using physical coordinates will result in coupling among the discrete representations. There are procedures that can be utilized to express coupled systems of equations to independent set of equations using generalised coordinates/functions/variables. Transformation of physical coordinates to modal coordinates using eigen properties of linear time invariant system being one of them. Here, for convenience sake, without loss of generality, certain expressions are given in terms of generalised coordinates. Following sections briefly explains the standard methods used for deriving the equilibrium equations.

Hamilton's Principle

Finite element displacement method for elastostatics

Hamilton's Principle

Physical problems are in general quantified by partial differential equations that relate the physical variables governing the scope of the study.The general framework for spatial and temporal description of particles is given by Hamilton's principle as stated by equation (1).

$$\int_{t_1}^{t_2} \{\delta(T - U) + \delta W_{nc}\}\, dt = 0 \qquad (1)$$

Where, T is the kinetic energy, U is the elastic potential and W_{nc} is the work done by non-conservative forces. The limits of the time integral are arbitrary.

For mechanical systems, continuous and/or discrete, mass is associated with kinetic energy, the restoring energy is represented by the elastic strain energy and the system damping accounts for the non-conservative work done.

Lagrange's equation of motion

Application of Hamilton's principle to discrete systems results in a more convenient Lagrange's equations given by equation (2).

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) + \frac{\partial D}{\partial \dot{q}_i} + \frac{\partial U}{\partial q_i} = Q_j j = 1,2,\ldots,n \qquad (2)$$

Where, D is the dissipation function, q_i and Q_j are the generalized coordinates and forces respectively. The over dot indicates derivative with respective time.

In generalized coordinates q_i the kinetic energy, potential energy and the dissipation function are

$$T = T(\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n) \quad = \frac{1}{2}\{\dot{\boldsymbol{q}}\}^T[\boldsymbol{M}]\{\dot{\boldsymbol{q}}\} \tag{3}$$

$$U = U(q_1, q_2, \dots, q_n) = \frac{1}{2}\{\boldsymbol{q}\}^T[\boldsymbol{K}]\{\boldsymbol{q}\} \tag{4}$$

$$D = D(\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n) = \frac{1}{2}\{\dot{\boldsymbol{q}}\}^T[\boldsymbol{C}]\{\dot{\boldsymbol{q}}\} \tag{5}$$

The work done by non-conservative forces are

$$\delta W_{nc} = \sum_{j=1}^{m}\left[Q_j - \frac{\partial D}{\partial \dot{q}_j}\right]\delta q_j \tag{6}$$

Using equation (3) to (6) in (2) yields the equations of motion of a multi-degrees-of-freedom system as:

$$[\boldsymbol{M}]\{\ddot{\boldsymbol{q}}\} + [\boldsymbol{C}]\{\dot{\boldsymbol{q}}\} + [\boldsymbol{K}]\{\boldsymbol{q}\} = \{\boldsymbol{Q}\} \tag{7}$$

In general, most of the mechanical motions take place under certain constraints, which are expressed by equation (8)

$$g_j(q_1, q_2, \dots, q_n) = 0 \qquad j = 1,2,\dots,n \tag{8}$$

For linear systems the constraint equations are of the form

[G]{q}={0} (9)

G is of size $m \times n$

The column matrix *{q}* can be partitioned into as set of *(n-m)* independent displacements $\{q_1\}$ and a set of m dependent displacements $\{q_2\}$. Equation (9) is modified as

$$[G_1 \quad G_2]\begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix} = \{0\} \qquad (10)$$

Therefore the dependent displacements are

$$\{q_2\} = -[G_2]^{-1}[G_1]\{q_1\} \qquad (11)$$

Combining equation (10) and (11) i.e. the two sets of displacements

$$\{q\} = \begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix} = \begin{bmatrix} I \\ -G_2{}^{-1}G_1 \end{bmatrix}\{q_1\} = [T_G]\{q_1\} \qquad (12)$$

I is the unit matrix

The corresponding energy expressions given by equations (3) to (5), in the presence of constraints assumes the following form

$$T = \frac{1}{2}\{\dot{q}_1\}[\bar{M}]\{\dot{q}_1\} \qquad (13)$$

$$D = \frac{1}{2}\{\dot{q}_1\}[\bar{C}]\{\dot{q}_1\} \qquad (14)$$

$$U = \frac{1}{2}\{q_1\}^T[\bar{K}]\{q_1\} \qquad (15)$$

Where,

$[\bar{M}] = [T_G]^T[M][T_G]$, $[\bar{C}] = [T_G]^T[C][T_G]$, $[\bar{K}] = [T_G]^T[K][T_G]$

The virtual work done by applied forces is transformed to

$$\delta W = \sum_{j=1}^{n} Q_j \, \delta q_j = \{\delta q\}^T\{Q\}\{\delta q_1\}^T[T_G]^T\{Q\} = \{\delta q_1\}^T\{\bar{Q}\} \tag{16}$$

Where, $\{\bar{Q}\} = [T_G]^T\{Q\}$

The energy and virtual work functions are now expressed in terms of independent displacements and so may be substituted in Lagrange's equations to give the equations of motion.

$$[\bar{M}]\{\ddot{q}_1\} + [\bar{C}]\{\dot{q}_1\} + [\bar{K}]\{q_1\} = \{\bar{Q}\} \tag{17}$$

It can be concluded that, for a solid continuum the kinetic and strain energy along with the dissipation needs to be computed to determine its equations of motion utilizing Lagrange's equation. For static cases, the principle of minimum potential energy is invoked to arrive at the state expressions.

Finite element displacement method for elastostatics

A solid continuum is discretised into simple geometrical domains known as elements, where the governing differential equations are solved. The solution is sought at specified discrete locations within the element extent, known as nodes, in most cases they are at the element boundaries. The discrete solutions from various nodes of the element are used for interpolating the dependent variable at any arbitrary location within the element. In this section, a general description of the displacement method is given. The user can refer several published literature in this regard.

Principle of minimum potential energy

It states that "Of all possible displacement states a body can assume that satisfy compatibility and specified kinematic conditions, the state that satisfies the equilibrium equations makes the potential energy assume a minimum value".

For a linearly elastic body with conservative loads, the expression for potential energy is

$$\Pi = U + \Omega \quad (1)$$

Where, U is the strain-energy and Ω is potential of all external loads. The potential energy is a function of the degrees of freedom q_i ie., $\Pi = \Pi(q_1, q_2, \dots, q_n)$. Using principle of stationary potential energy, where equilibrium prevails when $d\Pi = 0$, is expressed as

$$d\Pi = \frac{\partial \Pi}{\partial q_1} dq_1 + \frac{\partial \Pi}{\partial q_2} dq_2 + \cdots + \frac{\partial \Pi}{\partial q_n} dq_n = 0 \quad (2)$$

Utilizing equation (2), *n* equations solved for obtaining values of degrees of freedom. The general form of equation (1) is

$$\Pi = \int_V \left(\frac{1}{2}\{\boldsymbol{\varepsilon}\}^T[\mathbf{D}]\{\boldsymbol{\varepsilon}\} - \{\boldsymbol{\varepsilon}\}^T[\mathbf{D}]\{\boldsymbol{\varepsilon}_0\} + \{\boldsymbol{\varepsilon}\}^T[\boldsymbol{\sigma}_0]\right) dV - \int_V \{\mathbf{u}\}^T\{\mathbf{F}\} dV - \int_S \{\mathbf{u}\}^T\{\boldsymbol{\Phi}_S\} dS - \{\mathbf{q}\}^T\{\mathbf{P}\}$$

(3)

The terms in equation (3) are strain energy, body forces, work done by surface traction and discrete loads respectively.To derive the finite element characteristics of a solid continuum in terms of displacements, it is assumed in the following form,

$$\mathbf{u} \approx \hat{\mathbf{u}} = \sum \mathbf{N}_i u_i^e = [\mathbf{N}_1 \quad \mathbf{N}_2 \quad \dots \quad \dots \quad \mathbf{N}_n] \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{Bmatrix}^e = \mathbf{N}\mathbf{u}^e$$

(4)

Where, **N** is the vector of prescribed functions of position and $\mathbf{u}^e$ the nodal displacements of an element.

In general,

$$\mathbf{N}_i(X_j) = \delta_{ij}$$

(5)

This can be satisfied by choosing linear functions of $X \in \mathbb{R}$.

The strains at all points within the element are determined by

$$\boldsymbol{\varepsilon} = \mathbf{L}\,\mathbf{u}$$

(6)

Where, **L** is a suitable linear operator.

The approximate values of strains are

$$\hat{\boldsymbol{\varepsilon}} = \mathbf{B}\,\mathbf{u}$$

(7)

and **B=L N**

For linear elastic behaviour of the material the constitutive relations are given as

$$\boldsymbol{\sigma} = \mathbf{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) + \boldsymbol{\sigma}_0 \quad (8)$$

The nodal forces are statically equivalent to the boundary stresses and distributed loads on the element and are given by

$$\mathbf{f}^e = \begin{Bmatrix} \mathbf{f}_1^e \\ \mathbf{f}_2^e \\ \vdots \\ \mathbf{f}_n^e \end{Bmatrix} \quad (9)$$

The distributed forces are

$$\boldsymbol{b} = \begin{Bmatrix} b_{x_1} \\ b_{x_2} \\ \vdots \\ b_{x_n} \end{Bmatrix} \quad (10)$$

Invoking virtual work principle, which states that sum of internal and external work through virtual displacement, is zero;

The internal work done per unit volume by stresses and distributed forces is

$$\delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma} - \delta\mathbf{u}^T\boldsymbol{b} = \delta\mathbf{u}^T\left(\mathbf{B}^T\boldsymbol{\sigma} - \mathbf{N}^T\boldsymbol{b}\right) \quad (11)$$

The work done by the nodal forces are

$$\delta\mathbf{u}^T\mathbf{f}^e \quad (12)$$

Summing equations(11) and (12), integrating over the volume of the element, and factoring the virtual displacements,

$$\mathbf{f}^e = \int_{V_e} \mathbf{B}^T \boldsymbol{\sigma}\, dV_e - \int_{V_e} \mathbf{N}^T \boldsymbol{b}\, dV_e \qquad (13)$$

This statement is valid for any stress-strain relations. Equation (12) can be expressed as

$$\mathbf{f}^e = \mathbf{K}^e \mathbf{u}^e + \boldsymbol{Q}^e \qquad (14)$$

Where,

$$\mathbf{K}^e = \int_{V_e} \mathbf{B}^T \mathbf{D}\, \mathbf{B}\, dV_e \qquad (14\ a)$$

$$\boldsymbol{Q}^e = -\int_{V_e} \mathbf{N}^T \boldsymbol{b} dV_e - \int_{V_e} \mathbf{B}^T \mathbf{D}\, \boldsymbol{\varepsilon}_0 dV_e + \int_{V_e} \mathbf{B}^T \boldsymbol{\sigma}_0\, dV_e \qquad (14\ b)$$

Capabilities in FEAST-SMT

FEAST-SMT is a completely new solver implemented using C++ with OO technology. The solution to system's governing differential equations is obtained at discrete locations by FEAST-SMT solver by Sub-structured and Multi-Threaded (SMT) techniques. This has resulted in considerable reduction of solution time, which is comparable with industry standard software. The fundamental architecture of SMT can be easily extended to parallel processing of large order problems. Initial version of FEAST-SMT supported by PreWin was realized in the year 2008 and updating the software with advanced capabilities/features is continued by a dedicated team with the objective of making the software on par with industry standard software packages. The present version of FEAST-SMT (R1.2014) has the following capabilities.

1. Linear static analysis

2. Linear elastic stability analysis

3. Linear dynamic analysis

   a. Free vibration (undamped)

   b. Transient

      i. Newmark beta method

   c. Frequency response

      i. Base excitation

      ii. Effective mass computation

   d. Random response

Sub-structured and Multi-Threaded (SMT)

Sub-structuring is a well-established method in finite element method. In this method the complete structure is subdivided into number of sub-structures called super elements. Each substructure is defined by its boundary and internal nodes that are 'condensed' out. The complete structure is an assemblage of super elements that interfaces with the adjacent counterpart along the boundary (nodes). This technique considerably reduces the number of variables for solution, which directly result in significant reduction of solution time.Figure

Sub-structuring approach is advantageous due to the following reasons:

1. It splits the work involved in the calculation process into several independent and discrete packages enabling parallel processing
2. Much of the work carried out for any given sub-structure can be used again in later calculations.
3. It reduces the amount of memory required to solve the model.

Multi-threading is a kind of parallel processing technique that can be used on single processor machines such as a typical desktop, especially with a processor having multi-core computing capabilities. The computations associated with each sub-structure, as shown in Figure 3:1 are assigned to separate threads and the entire tasks can be carried out without giving idle time for the processor, and it can be used to increase the interactivity of the program. Since, the operating system assigns time slices to different activities like computations of sub-structures and user operations, the user theoretically can continuously interact with the program and the sub-structure calculations are carried out in the background of the program in different threads. Thus, multi-threading offers advantage of carrying outcomputations with celerity and greater interactivity. In FEAST*SMT* static condensation and component mode synthesis with fixed interface are used. The latter technique is also known as Craig-Bampton method.

Static condensation

Dynamic sub -structuring

Static condensation

This method is adopted when static problems are considered. The stiffness, displacements and forces associated with boundary and internal nodes are identified with subscripts ***B*** and ***I*** respectively. Equation (1) is the force equilibrium statement, expressed with earlier description. The internal displacement u_I of the second part of the equation is given by equation (2). This variable is eliminated from the first relation of equation (1), which reduces it in terms of boundary displacements u_B as given in equation (4).

The coefficient of u_B is the 'condensed' matrix for the structure in terms of the super-elements or sub-structures.

$$\begin{bmatrix} k_{BB} & k_{BI} \\ k_{IB} & k_{II} \end{bmatrix} \begin{Bmatrix} u_B \\ u_I \end{Bmatrix} = \begin{Bmatrix} f_B \\ f_I \end{Bmatrix} \quad (1)$$

$$\{u_I\} = -[k_{II}]^{-1}([k_{IB}]\{u_B\} - \{f_I\}) \quad (2)$$

$$([k_{BB}] - [k_{BI}][k_{II}]^{-1}[k_{IB}])\{u_B\} = \{f_B\} - [k_{BI}][k_{BI}]^{-1}\{f_I\} \quad (3)$$

$$[K_{condensed}]\{u_B\} = \{f\} \quad (4)$$

Dynamic sub- structuring(component mode synthesis - fixed interface method)

The kinetic and strain energy of a sub-structure are given as

$$T_s = \frac{1}{2}\{\dot{u}\}_s^T[M]_s\{\dot{u}\}_s \quad (1)$$

$$U_s = \frac{1}{2}\{u\}_s^T[K]_s\{u\}_s \quad (2)$$

To reduce the number of degrees of freedom, the transformation given in equation (3) is used.

$$\{\mathbf{u}\}_s = \begin{bmatrix}\mathbf{u}_I \\ \mathbf{u}_B\end{bmatrix} = \begin{bmatrix}\boldsymbol{\phi}_N & \boldsymbol{\phi}_c \\ \mathbf{0} & \mathbf{I}\end{bmatrix}_s \begin{bmatrix}\mathbf{q}_N \\ \mathbf{u}_B\end{bmatrix}_s = [\mathbf{T}_F]_s\{\mathbf{q}\}_s \quad (3)$$

$\boldsymbol{\phi}_N$Matrix is composed of columns corresponding to the natural modes of the sub-structure with fixed interfaces, which is obtained by solving equation (4)

$$[\mathbf{K}_{II} - \omega^2\mathbf{M}_{II}]\{\boldsymbol{\phi}_I\} = 0 \quad (4)$$

In general $\boldsymbol{\phi}_N$ is a rectangular matrix with element ϕ_{ij} with $j>i$. In most of the situations only first few modes are considered as they dominate the dynamic characteristics of the structure. $\mathbf{q}_N$ are the generalized coordinates. $\boldsymbol{\phi}_c$ is the matrix of constraint modes of the sub-structure. The columns represent the values assumed by the degrees of freedom at the internal nodes for a unit value of one of the degrees of freedom at an interface boundary node. These are obtained by solving equation (5).

$$[\mathbf{K}_{II}]\{\mathbf{u}_I\} + [\mathbf{K}_{IB}]\{\mathbf{u}_B\} = 0 \quad (5)$$

and

$$\{\mathbf{u}_I\} = -[\mathbf{K}_{II}]^{-1}[\mathbf{K}_{IB}]\{\mathbf{u}_B\} \quad (6)$$

Where, $\{\mathbf{u}_I\} = \boldsymbol{\phi}_c\{\mathbf{u}_B\}$

The energy expressions are modified as

$$T_s = \frac{1}{2}\{\dot{\mathbf{q}}\}_s{}^T\{\bar{\mathbf{M}}\}_s\{\dot{\mathbf{q}}\}_s \quad (7)$$

$$U_s = \frac{1}{2}\{\mathbf{q}\}_s{}^T\{\bar{\mathbf{K}}\}_s\{\mathbf{q}\}_s \quad (8)$$

Where,

$$[\bar{\mathbf{M}}]_s = [\mathbf{T}_F]_s{}^T[\mathbf{M}]_s[\mathbf{T}_F]_s = \begin{bmatrix} \bar{\mathbf{M}}_{NN} & \bar{\mathbf{M}}_{NB} \\ \bar{\mathbf{M}}_{BN} & \bar{\mathbf{M}}_{BB} \end{bmatrix}_s \quad (9)$$

$$[\bar{\mathbf{K}}]_s = [\mathbf{T}_F]_s{}^T[\mathbf{K}]_s[\mathbf{T}_F]_s = \begin{bmatrix} \bar{\mathbf{K}}_{NN} & 0 \\ 0 & \bar{\mathbf{K}}_{BB} \end{bmatrix}_s \quad (10)$$

$$\bar{\mathbf{M}}_{NN} = \boldsymbol{\phi}_N{}^T\mathbf{M}_{II}\boldsymbol{\phi}_N \quad (11)$$

$$\bar{\mathbf{K}}_{NN} = \boldsymbol{\phi}_N{}^T\mathbf{K}_{II}\boldsymbol{\phi}_N \quad (12)$$

$\bar{\mathbf{M}}_{NN}$ and $\bar{\mathbf{K}}_{NN}$ are diagonal.

$$\bar{\mathbf{K}}_{BB} = \mathbf{K}_{BB} - \mathbf{K}_{BI}\mathbf{K}_{II}^{-1}\mathbf{K}_{IB} \quad (13)$$

Where, $\mathbf{K}_{BB}$, $\mathbf{K}_{BI}$, $\mathbf{K}_{IB}$ and $\mathbf{K}_{II}$ are partitions of K; is the stiffness matrix of the sub-structure in terms of the interface boundary degrees of freedom.

If two adjacent sub-structures are considered, the energy expressions assume the following form

$$T = \frac{1}{2}\{\dot{\mathbf{q}}\}^T[\mathbf{M}]\{\dot{\mathbf{q}}\} \quad (14)$$

$$U = \frac{1}{2}\{q\}^T[K]\{q\} \quad (15)$$

Where,

$$\{\mathbf{q}\} = \begin{bmatrix} \mathbf{q}_N^I \\ \mathbf{q}_N^{II} \\ \mathbf{u}_B \end{bmatrix}, \quad [\mathbf{M}] = \begin{bmatrix} \bar{\mathbf{M}}_{NN}^I & 0 & \bar{\mathbf{M}}_{NB}^I \\ 0 & \bar{\mathbf{M}}_{NN}^{II} & \bar{\mathbf{M}}_{NB}^{II} \\ \bar{\mathbf{M}}_{BN}^I & \bar{\mathbf{M}}_{BN}^{II} & \bar{\mathbf{M}}_{BB}^I + \bar{\mathbf{M}}_{BB}^{II} \end{bmatrix} \text{ and } [\mathbf{K}] = \begin{bmatrix} \bar{\mathbf{K}}_{NN}^I & 0 & 0 \\ 0 & \bar{\mathbf{K}}_{NN}^{II} & 0 \\ 0 & 0 & \bar{\mathbf{K}}_{BB}^I + \bar{\mathbf{K}}_{BB}^{II} \end{bmatrix}$$

The final equation of the complete structure is

$$[\mathbf{M}]\{\ddot{\mathbf{q}}\} + [\mathbf{K}]\{\mathbf{q}\} = 0 \quad (16)$$

Linear static analysis

Finite element analysis consists of determining the potential energy Π^e ,which corresponds to the integrals of equation (see equation no:3) for each and every element that are used in describing the structure. The displacement field within the element is expressed as a function of corresponding nodal values as given in equation no :4.

The total potential of the idealized (approximated) continuum composed of m elements is

$$\Pi = \sum_{i=1}^{m} \Pi_i^e \quad (1)$$

The global static equilibrium equation in terms of nodal displacement **u**,is of the form

Ku=f (2)

Where, K is the global stiffness matrix obtained from assembling the element stiffness matrix as

$$\mathbf{K} = \sum_{e=1}^{m} \mathbf{K}^e \quad (3)$$

The total load vector **f**, includes the effect of surface traction, body forces, concentrated loads, initial stresses and strains. Equation (2) gives the nodal displacement values that are back substituted appropriately to obtain strains and stresses of the structure.

Linear elastic stability analysis

One of the most important engineering analyses during sizing of structures is the elastic stability or buckling analysis. This problem is fundamentally a neutral equilibrium problem. It is a significant aspect during design phase in applications involving slender structures subjected to membrane compressive stress fields. Buckling occurs when structure converts membrane strain energy into strain energy of bending. The objective of the analysis is to determine the bifurcation buckling load, for which a reference configuration and an infinitesimally close configuration are both possible equilibrium configurations. The buckling of structure is strongly related to the effects of membrane forces that reduce the bending stiffness. In finite element analysis, the effects of membrane forces are accounted by the geometrical stiffness/ initial stress stiffness/ differential stiffness/ stability coefficient. The geometrical stiffness matrix $[\mathbf{K}_\sigma]$ accounts the membrane force effects. It depends on the geometry, displacement field and the stress state. This matrix augments the conventional structural stiffness matrix. The elemental geometrical stiffness matrix is given by

$$[\mathbf{K}_\sigma] = \int_{V_e} [\mathbf{G}]^T \begin{bmatrix} \boldsymbol{s} & 0 & 0 \\ 0 & \boldsymbol{s} & 0 \\ 0 & 0 & \boldsymbol{s} \end{bmatrix} [\mathbf{G}]\, dV \qquad (1)$$

Where, $[\boldsymbol{s}] = \begin{bmatrix} \boldsymbol{\sigma}_{x0} & \boldsymbol{\tau}_{xy0} & \boldsymbol{\tau}_{zx0} \\ \boldsymbol{\tau}_{xy0} & \boldsymbol{\sigma}_{y0} & \boldsymbol{\tau}_{yz0} \\ \boldsymbol{\tau}_{zx0} & \boldsymbol{\tau}_{yz0} & \boldsymbol{\sigma}_{z0} \end{bmatrix}$ is the initial stress tensor and $[\mathbf{G}] = \partial\, [\mathbf{N}]$, the partial derivative of the shape functions.

The bifurcation buckling is obtained by solving the eigenvalue problem,

$$([\mathbf{K}] + \lambda_{cr}[\mathbf{K}_\sigma])\{\overline{\mathbf{u}}\} = \{0\} \qquad (2)$$

Here, λ_{cr} are the eigenvalues corresponding to the load multipliers that causes elastic instability of the structure. The corresponding eigenvector $\{\overline{\mathbf{u}}\}$ is the displacement configuration assumed by the structure that is

indeterminate. If $\{\mathbf{f}\}_0$ represents the initial load vectors, the critical buckling load is given by,

$$\{\mathbf{f}\}_{cr} = \lambda_{cr}\{\mathbf{f}\}_0 \quad (3)$$

Linear dynamic analysis

The equation of motion under constraints given in equation 17 can be considered as generalization of a single degree of freedom system to a multi-degree of freedom system. The equations are coupled, characterized by the presence of off-diagonal elements of the system matrices. The mass matrix obtained from first principles is called consistent as it is formulated in the same way as the stiffness matrix. In FEASTSMT, consistent and lumped mass formulations are implemented. The lumped mass matrix consists of mass components corresponding to translation degrees of freedom and rotational mass is assumed zero. Certain industry standard proprietary software use coupled mass, which is mean of consistent and lumped mass formulations. This feature is also included inFEASTSMT. The computation of stiffness and mass matrices are explained in element's formulation.

Free vibration / eigenvalue analysis

Transient Analysis

- Newmark beta method

Frequency Response Analysis

- Base excitaiton
- Effective mass computation

Random response analysis

Free vibration/eigenvalue analysis

The free vibration analysis is used for dynamic characterization of a structure expressed by its eigenvalue and eigenvector (function in case of continuum). This helps in uncoupling of the equations of motion aiding faster response analysis. Damping is ignored for free vibration analysis to overcome the mathematical complexity. Use of proportional damping concept for decoupling damping matrix though mathematically elegant, it seldom predicts the observed values to acceptable accuracy. In most of the real life situations, the damping values considered are estimated experimentally rather than using derived values. For undamped free vibration analysis, the equation of motion is

$$[\boldsymbol{M}]\{\ddot{\boldsymbol{q}}\} + [\boldsymbol{K}]\{\boldsymbol{q}\} = \{\mathbf{0}\} \quad (1)$$

The eigenvalue problem corresponding to equation (1) is obtained by considering the oscillations to be harmonic.

$$[\mathbf{K} - \omega_i^2\mathbf{M}]\{\boldsymbol{\phi}\} = \{\mathbf{0}\} \quad (2)$$

Where, ω_i and $\boldsymbol{\Phi}$ are the eigenvalues and corresponding eigenvectors, or natural frequencies and mode shapes. The conditions for equation (2) to have non-trivial solution is

$$|K - \omega_i^2 M| = 0 \quad (3)$$

i.e., the determinant value should be zero.*M* isalways positive definite, *K* is positive definite or positive semi-definite, the eigenvalues are all real and either positive or zero. All the eigenvalues are not necessarily distinct. If the eigenvalue occurs ***m*** times, it is said to be of multiplicity ***m*** and the eigen problem is said to have multiple eigenvalues. If the eigenvalues are distinct, then corresponding to each one, there is a non-trivial solution to equation (2). These solutions are known as eigenvectors. The eigenvectors are usually mass normalized. It can be such that

$$\{\boldsymbol{\phi}\}^{\mathrm{T}}[\mathbf{M}]\{\boldsymbol{\phi}\} = \mathrm{I} \quad (4)$$

The eigenvectors for square symmetric matrices exhibit orthogonal properties.

Transient analysis

When a structure is subjected to a sudden non-periodic excitation, the ensuing oscillation is transient. i.e., the structure undergoes free vibration and it decays in the presence of positive damping. The applied excitation is for a very short duration when the definition of transient is rigorous. In reality, the transient excitation definition covers situations where the continually applied load is an arbitrarily varying function of time. There are two methods to determine the transient response, they are:

- Modal method: Free vibration response for a damped free oscillation is carried out for determining eigenvalues and eigenvectors. Using these, the physical representation of the system equations are expressed in terms of modal coordinates and solved to obtain the response.
- Direct method: The direct method has the advantage of not solving the eigen problem prior to response analysis. This method involves computation time as system matrices need to be inverted for every time step.

The equation of motion of the complete structure is given as

$$[\bar{M}]\{\ddot{q}_1\} + [\bar{C}]\{\dot{q}_1\} + [\bar{K}]\{q_1\} = \{\bar{Q}\} \quad (1)$$

For modal analysis, without damping the equation (1) is modified to

$$[\bar{M}]\{\ddot{q}_1\} + [\bar{K}]\{q_1\} = \{0\} \quad (2)$$

Equation (2) is an eigenvalue problem, from which the eigenvalue ω_i^2 and the corresponding eigenvectors $\{\boldsymbol{\phi}\}$ are determined, which are obtained by assuming the oscillations to be harmonic.

The eigenvectors exhibit orthogonal properties as given by equation (3)

$$\{\boldsymbol{\phi}\}_r^T\{\boldsymbol{\phi}\}_s = \delta_{rs} \quad (3)$$

Using the orthogonal properties and assuming modal damping factor, equation (1) can be expressed in modal coordinates as

$$\ddot{q}_r + 2\,\gamma_r\,\omega_r\,\dot{q}_r + \omega_r^2 q_r = Q_r(t) \quad (4)$$

Equation (4) corresponds to the rth mode of vibration with γ_r as the modal damping factor. Solution of equation (4) can be obtained using the convolution integral or Duhamel integral given by

$$q_r(t) = \int_0^t Q_r(\tau)\,h_r(t-\tau)\,d\tau \quad (5)$$

Where, $h_r(t)$ is the impulse response function of the system, and

$$h_r(t) = \frac{1}{\omega_{d_r}} e^{(-\gamma_r \omega_r t)} \sin(\omega_{d_r} t) \quad (6)$$

ω_{d_r} is the damped natural frequency of rth mode defined as $\omega_{d_r} = \omega_r(1-\gamma_r^2)^{\frac{1}{2}}$

The expression (5) is evaluated numerically as the loading is usually available as experimental data.The numerical solution of transient analysis can be obtained using following four methods either by modal or direct analysis.

1. Central difference method
2. The Houbolt method
3. The Newmark beta method
4. The Wilson theta method

In FEASTSMT(2014.R1) with modal analysis the Newmark beta method is used to numerically determine the integral value of equation (5)

Newmark beta method

In this method constant average acceleration is assumed, as it is unconditionally stable in comparison with linear acceleration method. The method is therefore implicit. The response at time t_{j+1} is obtained by evaluating the equation of motion:

$$m\,\ddot{q}_1 + c\,\dot{q}_1 + k\,q_1 = f_{j+1} \quad (1)$$

The responses are:

$$\ddot{q}_{1_{j+1}} = \frac{1}{\beta(\Delta t)^2}\left(q_{1_{j+1}} - q_{1_j}\right) - \frac{1}{\beta(\Delta t)}\dot{q}_{1_j} - \left(\frac{1}{2\beta} - 1\right)\ddot{q}_{1_j} \quad (2)$$

$$\dot{q}_{1_{j+1}} = \frac{\gamma}{\beta(\Delta t)}\left(q_{1_{j+1}} - q_{1_j}\right) + \left(1 - \frac{\gamma}{\beta}\right)\dot{q}_{1_j} + \Delta t\left(1 - \frac{\gamma}{2\beta}\right)\ddot{q}_{1_j} \quad (3)$$

$$q_{1_{j+1}} = q_{1_j} + \dot{q}_{1_j}\Delta t + (\Delta t)^2\left\{\left(\frac{1}{2} - \beta\right)\ddot{q}_{1_j} + \beta\,\ddot{q}_{1_{j+1}}\right\} \quad (4)$$

For $\gamma \geq \frac{1}{2}$ and $\beta \geq \frac{1}{4}\left(\gamma + \frac{1}{2}\right)^2$ the solution is unconditionally stable. The method introduces an artificial damping for values of $\gamma < \frac{1}{2}$. For good accuracy, the time step should be $\frac{\Delta t}{\tau_0} = 0.01$ or $\omega_0 \Delta t = \frac{\pi}{50}$.

Frequency Response Analysis

No mechanical system vibrates unless it is imparted some initial displacement and/or velocity. If these stimuli act initially for an infinitesimal time, the subsequent response is called free response whereas it is known as forced response if the stimuliact throughout the duration of interest. In the mathematical solution used to determine the response of such a system, these two responses are usually obtained separately as the particular solution and the complementary or homogenous solution. The sum of these two responses is referred to as the total response of the system.

A time-domain analytical model is a set of differential equations with respect to the independent variable time(t). A frequency-domain model is a set of input-output transfer functions with respect to the independent variable frequency (ω). Time domain approach is adopted when response predictions are requiredfor a time varying force. Frequency domain methods are most efficient for random vibrations and periodic loadings.

When forced response is studied in frequency domain the mathematical manipulations are relatively simple. The response of a system to an excitation expressed in the frequency domain is called its frequency response. In basic terms, the frequency response of a dynamic system is the response to a harmonic excitation. As the amplitude and the frequency of the excitation are varied, the response also changes. In this manner, the response of the system over a range of excitation frequencies is determined. Thus, for analytical purpose, the system response is studied for harmonic excitations. Although real systems are seldom excited by purely harmonic forces with only a single frequency component, analysis of the effects of such excitations on the vibrating systems provide the foundation for analyses of more complicated forms of excitation encountered in real systems, such as periodic excitations of various types containing multiple frequency components of random excitations and so forth.

Any periodic excitation can be decomposed into harmonic components using Fourier series and effect of each such harmonics can be studied independently and added to get the total response. Frequency domain

considerations are applicable even when the excitations are not periodic. For arbitrary excitations one has to resort to integral transforms such as Laplace and Fourier in frequency domain and convolution integral (Duhamel) in time domain. In fact a time signal can be transformed into its frequency spectrum through the Fouriertransform. Fora given time signal, an equivalent Fourier spectrum, which contains all the frequency (sinusoidal) components of the signal, can be determined either analytically or computationally. Hence, a time domain representation and analysis has an equivalent frequency domain representation and analysis, at least for linear dynamic systems. For this reason, and because of the periodic nature of typical vibration signals, the frequency response analysis is extremely useful in the subject of mechanical vibrations.

When subjected to dynamic forces, a structure's total response is the sum of the responses of its modes of vibration. Only a relatively small number of a structure's modes; typically those with the lowest frequencies are of concern. The response associated with higher order modes isnot enough to produce significant stress. The equation of motion of MDOF system has the form

$$[\boldsymbol{M}]\{\ddot{\boldsymbol{q}}_1\} + [\boldsymbol{C}]\{\dot{\boldsymbol{q}}_1\} + [\boldsymbol{K}]\{\boldsymbol{q}_1\} = \{\boldsymbol{Q}\} \quad (1)$$

The free vibration response of equation (1) gives the system characteristics such as the natural frequencies ω_i^2 and the associated modal matrix Φ which is formed by columns of eigenvectors $\{\boldsymbol{\Phi}\}$ corresponding to *Ith* mode. The physical displacements are expressed in terms of modal coordinates $\boldsymbol{\eta}(\mathbf{t})$ as

$$\boldsymbol{q}_1 = \boldsymbol{\Phi}\ \boldsymbol{\eta}(\mathbf{t}) \quad (2)$$

Eliminating q_1 from equation (1) and pre-multiplying by the transpose of the modal matrix Φ, decoupled equations will be of the form:

$$\bar{M}\ddot{\eta} + \bar{C}\dot{\eta} + \bar{K}\eta = Q \quad (3)$$

Where,

$$\bar{M} = \Phi^T M \Phi \quad \bar{C} = \Phi^T C \Phi \quad \bar{K} = \Phi^T K \Phi \quad \bar{Q} = \Phi^T Q$$

Both $\bar{M}$ and $\bar{K}$ are diagonal matrices. If columns of Φ are mass normalized, then $\bar{M} = I$ and $\bar{K} = \wedge I$

Where

$$\wedge = \begin{bmatrix} \omega_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \omega_n^2 \end{bmatrix}$$

If the nodal forces are harmonic, all with the same frequency, ω equation (1) reduces to

$$\ddot{\eta} + \bar{C}\dot{\eta} + \wedge \eta = Q \quad (4)$$

The steady state solution is obtained by assuming that the response is harmonic with frequency ω. The derivation is same as that of SDOF, where $e^{i\omega t}$ is factored to obtain the following relations.

$$[\wedge - \omega^2 I + i\omega\bar{C}]\eta = Q \quad (5)$$

$$\eta = [\wedge - \omega^2 I + i\omega\bar{C}]^{-1} Q \quad (6)$$

$$\boldsymbol{q_1} = \Phi[\wedge - \omega^2 I + i\omega\bar{C}]^{-1}\Phi^T\bar{Q} \quad (7)$$

$$u(i\omega) = H(i\omega)F(i\omega) \quad (8)$$

Where,

$$H(i\omega) = \Phi[\wedge - \omega^2 I + i\omega\bar{C}]^{-1}\Phi^T$$

$H_{jk}(i\omega)$ is the transfer function which represents the response in degree of freedom *j* due to a harmonic force of unit magnitude and frequency ω applied in degree of freedom *k*. The transfer function or frequency response function assumes the following two forms depending on the type of damping.

Structural damping:

$$H_{jk}(i\omega) = \sum_{r=1}^{n} \frac{\varphi_{jr}\varphi_{kr}}{(\omega_r^2 - \omega^2 + i\zeta\omega_r^2)} \quad (9)$$

In rationalized complex form

$$H_{jk}(i\omega) = \sum_{r=1}^{n} \frac{\varphi_{jr}\varphi_{kr}(\omega_r^2 - \omega^2)}{(\omega_r^2 - \omega^2)^2 + (\zeta\omega_r^2)^2} - i\sum_{r=1}^{n} \frac{\varphi_{jr}\varphi_{kr}(\zeta\omega_r^2)}{(\omega_r^2 - \omega^2)^2 + (\zeta\omega_r^2)^2} \quad (10)$$

ζ , is the material loss factor.

Proportional damping:

$$H_{jk}(i\omega) = \sum_{r=1}^{n} \frac{\varphi_{jr}\varphi_{kr}}{(\omega_r^2 - \omega^2 + i2\gamma_r\omega_r\omega)} \quad (11)$$

In rationalized complex form

$$H_{jk}(i\omega) = \sum_{r=1}^{n} \frac{\varphi_{jr}\varphi_{kr}(\omega_r^2 - \omega^2)}{(\omega_r^2 - \omega^2)^2 + (2\gamma_r\omega_r\omega)^2} - i\sum_{r=1}^{n} \frac{\varphi_{jr}\varphi_{kr}(2\gamma_r\omega_r\omega)}{(\omega_r^2 - \omega^2)^2 + (2\gamma_r\omega_r\omega)^2} \quad (12)$$

γ_r, is the modal damping ratio.The term $\boldsymbol{\varphi_{jr}\varphi_{kr}}$ is referred to as the modal constant. In FEASTSMT (2014.R1) proportional damping is implemented.

The below table summarizes expressions for obtaining displacement, velocity and acceleration response in time and frequency domain.

Response	**Gain function**	**Phase angle**
<u>Displacement</u> $\boldsymbol{u(i\omega) = H(i\omega)F(i\omega)}$ $\boldsymbol{u(t) = \lvert H(i\omega)\rvert Fe^{i(\omega t - \phi)}}$	$\boldsymbol{\lvert H(i\omega)\rvert}$	$\phi = arctan\left(\frac{Re[H_{jk}(i\omega)]}{Im[H_{jk}(i\omega)]}\right)$

<u>Velocity</u> $\dot{u}(i\omega) = [i\omega H(i\omega)]F(i\omega)$ $\dot{u}(t) = i\omega\lvert H(i\omega)\rvert F e^{i(\omega t-\phi)}$	$i\omega\lvert H(i\omega)\rvert$	
<u>Acceleration</u> $\ddot{u}(i\omega) = \left[-\omega^2 H(i\omega)\right]F(i\omega)$ $\ddot{u}(t) = -\omega^2\lvert H(i\omega)\rvert F e^{i(\omega t-\phi)}$	$-\omega^2\lvert H(i\omega)\rvert$	

Response expressions in frequency and time domain

Base excitation

In many engineering applications, one of the major modes of load transmission through a body takes place at the anchoring points/ mounts. For example, a passenger car is subjected to time varying loads during motion by the wheels; at the satellite-launch vehicle interface, the longitudinal loads are transmitted to the satellite. Apart from this, most of the vibration tests are conducted by exciting the base or the mounting points. From engineering analysis point of view, this can be viewed as imposed motion on the body. Analysis of structures under base excitation is extremely important, and it is routinely performed, in aerospace industry. The aim of the analysis is to verify the structural integrity of the space hardware, i.e., spacecraft, spacecraft antennae, electronic boxes, etc, when exposed to a sinusoidal or random base excitation environment. Besides the simulation, base-excitation tests are usually performed to qualify the structural design and also to identify the modal characteristics of the tested structure. The base excitation or imposed motion in FEASTSMT(2014.R1) is implemented using seismic mass or large mass or penalty mass concept. The degrees-of-freedom of the structure where imposed motion is to be applied, a large mass is considered. The excitation is applied as acceleration through the mass. A frequency response analysis is carried out.

Effective mass computation

Structures are subjected to enforced motion in many situations during their service life. The drive points are subjected to either displacement or acceleration as excitation rather than force for base excited structures. The dynamic response of the structure due to force is different from base motion. Testing of avionic packages and structures subjected to seismic loads are some examples of structures with support motion. It is therefore an important aspect when structural dynamic problems are considered.

The finite element model (mesh) of a structure for static analysis usually consists of large number of degrees of freedom. Using the same model for dynamic analysis can result is unacceptable time penalty and is not that essential. For base excited structures the concept of effective mass can be used to identify the most significant modes that participate in the structural oscillation. In other words, the kinetic energy contribution of mass particles into a particular mode is a measure that can be computed using effective mass. This results in a model with less degrees-of-freedom. Considering less number of modes for dynamic characterization, needless to mention, has many advantages.

This concept was introduced in the early seventies of twentieth century. Effective mass is useful in

- Definition of predominant modes having large reaction forces, especially in relation to the coupling between a spacecraft or launch vehicle with its payload
- Verification and updating of finite element models
- Vibration testing of space hardware. The concept of force limited vibration testing aimed at reducing the over testing of components, which is associated with conventional vibration testing is based on effective mass concept.

The equation of motion for a multi-degree-freedom system is given by equation (1)

$$\mathbf{M}\ddot{\mathbf{X}} + \mathbf{C}\dot{\mathbf{X}} + \mathbf{K}\mathbf{X} = \mathbf{Q} \quad (1)$$

The relative displacement when the system is subjected to base motion is

Z=X-Y (2)

Remembering that inertia forces are associated with total/absolute motion and the spring and damper with the relative motion, equation (1) is modified with no force excitation as

$$\mathbf{M}\ddot{\mathbf{Z}} + \mathbf{C}\dot{\mathbf{Z}} + \mathbf{K}\mathbf{Z} = -\mathbf{M}\,\mathbf{T}\,\ddot{\mathbf{Y}} \quad (3)$$

In equation (3) the RHS is the effective force for the system subject to base/ground motion i.e.

$$\mathbf{Q}_e = -\mathbf{M}\,\mathbf{T}\,\ddot{\mathbf{Y}} \quad (4)$$

Where the transformation matrix

$$\mathbf{T}^T = [\mathbf{T}_1^T \quad \mathbf{T}_2^T \quad \dots \quad \dots \quad \mathbf{T}_N^T]$$

$$\mathbf{T}_i = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{R} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{R} = \begin{bmatrix} 0 & (z_i - z_0) & -(y_i - y_0) \\ -(z_i - z_0) & 0 & (x_i - x_0) \\ (y_i - y_0) & -(x_i - x_0) & 0 \end{bmatrix}$$

(x_i, y_i, z_i) are the coordinates of the *Ith* node and (x_0, y_0, z_0) are the coordinates of the reference point about which the structure undergoes pitching/ rocking motion. The modal matrix is obtained by solving for the un-damped free vibration case of equation (3), given by

$$\mathbf{M}\ddot{\mathbf{Z}} + \mathbf{K}\mathbf{Z} = \mathbf{0} \quad (5)$$

Equation (6) shows the modal matrix whose column vectors corresponds to the eigen vector for each natural frequency/ eigen value of the system,

$$\Phi = \left[\begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{Bmatrix}_{\omega_1^0} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{Bmatrix}_{\omega_2^0} \cdots \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{Bmatrix}_{\omega_n^0} \right] \quad (6)$$

Assuming modal damping, the *Ith* equation of motion in the generalized coordinates q_i given in equation (8) is obtained by the substitution and pre-multiplication of expression in equation (7) with equation (5)

$$\{Z\} = [\Phi]\{q\} \quad (7)$$

$$m_i\ddot{q}_i + c_i\dot{q}_i + k_i q_i = f_i(t) \quad (8)$$

Where,

$$f_i(t) = -\Gamma_i^T \ddot{Y} \quad (9)$$

The participation factors for *Ith* mode in 6 directions are defined as

$$\Gamma_i^T = \phi_i^T M T \quad (10)$$

The matrix of participation factors is given by equation (5)

$$\Gamma^T = \Phi^T M T \quad (11)$$

$$\Gamma = [\Gamma_1, \Gamma_2, \Gamma_3, \ldots, \Gamma_n] \quad (12)$$

The participation factors of all modes in *jth* direction is given by

$$\gamma_j = \Phi^T M t_j \quad (13)$$

The eigenvectors exhibit orthogonal property and for mass normalised modal equations, the equation (14) holds.

$$\Phi^T M \Phi = I \quad (14)$$

Considering equation (14), the expression given in equation (13) is modified to relate the participation factors and the columns of transformation matrix.

$$\boldsymbol{t}_j = \boldsymbol{\Phi}\,\boldsymbol{\gamma}_j \quad (15)$$

Consider the product $\boldsymbol{t}_j^T \boldsymbol{M} \boldsymbol{t}_j$. It gives the total mass of the structure in *jth* direction and the rotational components give the mass moment of inertia passing through the point of rotation.

Therefore,

$$\boldsymbol{t}_j^T \boldsymbol{M} \boldsymbol{t}_j = \boldsymbol{\gamma}_j^T \boldsymbol{\Phi}^T \boldsymbol{M} \boldsymbol{\Phi}\, \boldsymbol{\gamma}_j = \boldsymbol{\gamma}_j^T \boldsymbol{\gamma}_j = \sum_{r=1}^{n} \boldsymbol{\Gamma}_{r,j}^2 \quad (16)$$

Equation (16) gives the total mass in *jth* direction, i.e., the summation of the square of the participation factors is the total mass in the direction; $\boldsymbol{\Gamma}_{r,j}^2$ is the effective modal mass. All the above expressions are valid only for a base fixed case to determine the effective modal mass.

Random Response Analysis

The accurate analysis, design and assessment of mechanical and structural systems, subjected to realistic dynamic environments, must consider the potential for random loads and randomness in structural and material properties. The engineering field that deals with these issues is known as random vibrations. The modern theory of random vibrations is the product of generations of works in the fields of deterministic structural vibrations, probabilistic analysis of mechanical system response, and random signal analysis. When the response of a mechanical system is forced by a random vibration excitation, the specific response to a future event cannot be computed because the dynamic environments to be realized in the future cannot be precisely predicted.

There exists large class of vibration problems for which the excitations, or forcing functions, are described statistically. Typical problems include launch vehicle buffeting during the transonic time of flight, random base-shake excitation of structures or components during qualification and/or acceptance testing, and spacecraft solar panel response analysis for acoustic excitation. The elements which generally describe this class of problem are:

- Multi-degree-of-freedom dynamic models, resulting in one or more normal modes required in any analysis.
- Multiple, random forces acting on the system, with the forces described by frequency dependent power spectral density(PSDs) functions
- Some degrees of correlation between the forces, with resulting force cross-correlations that are nonzero.

Application of statistical principles explicitly implies that very large numbers of observations/events are involved. Rich insights into the random dynamical problems can be acquired when it is investigated in frequency domain, besides the mathematical simplicity.

A fair understanding of modern theory of probability is essential to understand random vibration phenomenon. Modern theory of probability is

based on axioms propounded by Andrey Nikolayevich Kolmogrov (1903-1987). The detailed treatment on this subject can be found in standard publications.

Random processes can be analysed in time or frequency domains. The structural systems' response to random environments such as displacements, velocities and accelerations are important from applications view. The information contained in time and frequency domains are equivalent. These are related by Wiener-Khinchine relations.

$$S_X(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} R_X(\tau)e^{-i\omega\tau}d\tau \quad (1)$$

$$R_X(\tau) = \int_{-\infty}^{\infty} S_X(\omega)e^{i\omega\tau}d\omega \quad (2)$$

Equation (1) and (2) forms Fourier transform pairs between spectral density and autocorrelation respectively. In other words, the spectral density $S_X(\omega)$ can be defined as the Fourier transform of autocorrelation function. The cross spectral density function $S_{XY}(\omega)$ is a more generalized form of the spectral density function, to allow cross relation between excitation in one location to another location on the structure.

$$S_{XY}(\omega) = \int_{-\infty}^{\infty} R_{XY}(\tau)e^{-i\omega\tau}d\tau \quad (3)$$

In most problems there will be a number of input uncertainties associated with the model, such as material properties, loadings, geometry, and boundary conditions etc. A probabilistic analysis, therefore, can be viewed as the process of propagating these input uncertainties through the numerical model to yield results that also have uncertainties associated with them.

In FEASTSMT (2014.R1), autocorrelation is implemented; cross-correlation is not implemented. System randomness is not considered.

For a multi-degree of freedom system, the random environment is characterized in frequency domain as spectral density quantities specified as $[S_{FF}(i\omega)]$. The spectral density of the responses of interest, are given as:

Displacement:

$$[S_{XX}(i\omega)] = [\mathbf{\Phi}][H^*(i\omega)][\mathbf{\Phi}]^T[S_{FF}(i\omega)][\mathbf{\Phi}][H(i\omega)][\mathbf{\Phi}]^T \quad (4)$$

Velocity:

$$[S_{\dot{X}\dot{X}}(i\omega)] = -\omega^2[S_{XX}(i\omega)] \quad (5)$$

Acceleration:

$$[S_{\ddot{X}\ddot{X}}(i\omega)] = -\omega^4[S_{XX}(i\omega)] \quad (6)$$

FEAST-SMT element library

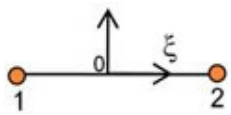
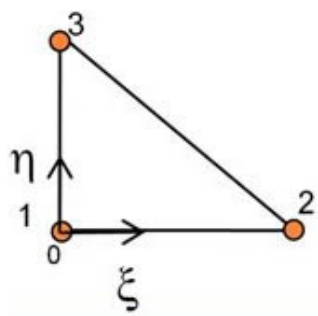
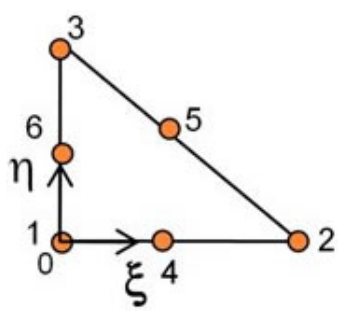
Here explains the element formulation that has been incorporated into FEAST-SMT. Presently, the elements available in FEAST-SMT are:

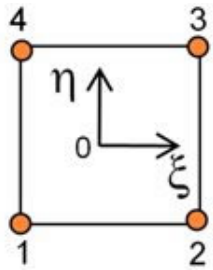
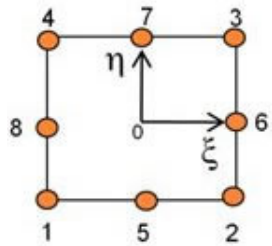
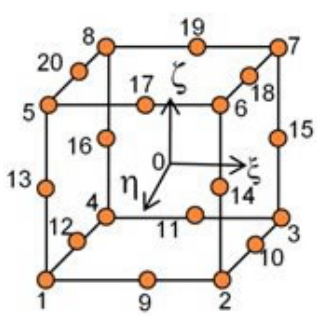
1. Plane strain element
2. Plane stress element
3. Axisymmetric element
4. Brick element
5. General beam element
6. General shell element
7. Gap element
8. Truss and Rod element
9. Spring element
10. Point mass element
11. Glue element

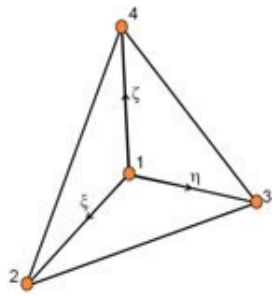
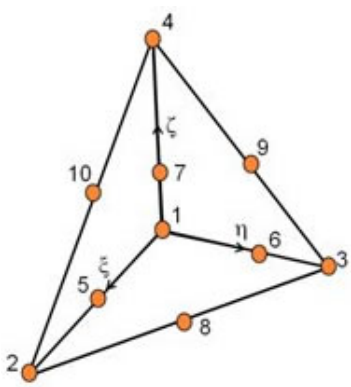
Before dwelling into the details of elements certain common features of its formulation are given. They are: element shape functions and Gauss-Legendre integration points. The total potential of the system can be obtained from the relations given in finite element displacement method for elastostatics and system matrices can be readily identified using the unique element characteristics .

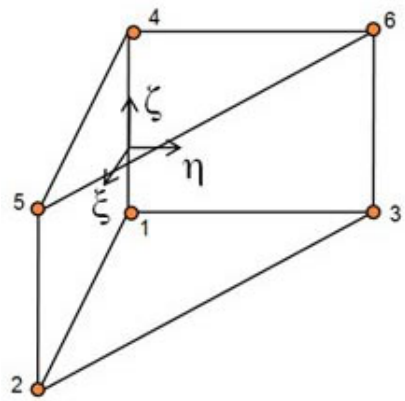
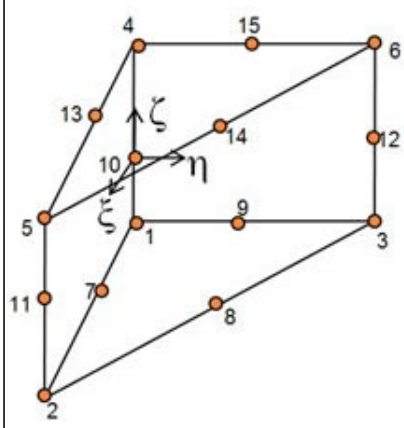
Element Interpolation Functions

The element solution is sought at nodal points. Using interpolation functions, it is possible to determine the value of dependent variable at any arbitrary location within the element. These functions for 1D, 2D and 3D elements are given below. All the interpolation functions are expressed in terms of natural coordinates or non-dimensional coordinate parameters. To numerically integrate quantities, appearing in the expression for potential energy function using Gauss-Legendre method, the order of integration, coordinates and the corresponding weights are required.See the table below.

Element geometry		Interpolation function
B1		$N_1 = (1 + \xi_i \xi)$
T3		$N_1 = (1 - \xi - \eta)$ $N_2 = \xi$ $N_3 = \eta$
T6		$N_1 = (1 - \xi - \eta)(1 - 2\xi - 2\eta)$ $N_2 = \xi(2\xi - 1)$ $N_3 = \eta(2\eta - 1)$ $N_4 = 4\xi(1 - \xi - \eta)$ $N_5 = 4\xi\eta$

		$N_6 = 4\eta(1-\xi-\eta)$
Q4		$\frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta)$
Q8		$N_i = \frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta)(\xi\xi_i+\eta\eta_i-1)$ $i = 1,2,3,4$ $N_j = \frac{1}{2}(1-\xi^2)(1+\eta\eta_j)$ $j = 5,7$ $N_k = \frac{1}{2}(1-\eta^2)(1+\xi\xi_k)$ $k = 6,8$
H20		$N_i = \frac{1}{8}(1+\xi_i\xi)(1+\eta_i\eta)(1+\zeta_i\zeta)(\xi\xi_i+\eta\eta_i+\zeta\zeta_i-2)$ $i = 1,2,3,4,5,6,7,8$ $N_j = \frac{1}{4}(1-\xi^2)(1+\eta_j\eta)(1+\zeta_j\zeta)$ $j = 9,11,17,19$ $N_k = \frac{1}{4}(1+\xi_i\xi)(1-\eta^2)(1+\zeta_i\zeta)$ $k = 10,12,18,20$ $N_l = \frac{1}{4}(1+\xi_l\xi)(1+\eta_l\eta)(1-\xi^2)$ $l = 13,14,15,16$

T4		$N_1 = (1-\xi-\eta-\zeta)$ $N_2 = \xi$ $N_3 = \eta$ $N_4 = \zeta$
T10		$N_1 = 2(1-\xi-\eta-\xi)\left(\frac{1}{2}-\xi-\eta-\xi\right)$ $N_2 = 2\xi\left(\xi-\frac{1}{2}\right)$ $N_3 = 2\eta\left(\eta-\frac{1}{2}\right)$ $N_4 = 2\zeta\left(\zeta-\frac{1}{2}\right)$ $N_5 = 4\xi(1-\xi-\eta-\xi)$ $N_6 = 4\eta(1-\xi-\eta-\xi)$ $N_7 = 4\xi(1-\xi-\eta-\xi)$ $N_8 = 4\xi\eta$ $N_9 = 4\eta\xi$ $N_{10} = 4\xi\xi$
P6		$N_i = \frac{1}{2}(1-\xi-\eta)(1+\zeta_i\zeta) \quad i = 1,4$ $N_i = \frac{1}{2}\xi(1+\zeta_i\zeta) \quad i = 2,5$

		$N_i = \frac{1}{2}\eta(1+\zeta_i\zeta) \quad i = 3,6$
P15		$N_i = \frac{1}{2}(1-\xi-\eta)(1+\zeta_i\zeta)(\zeta_i\zeta-2\zeta-2\eta) \quad i = 1,4$ $N_j = \frac{1}{2}\zeta(1+\zeta_j\zeta)(\zeta_j\zeta+2\zeta-2) \quad j = 2,5$ $N_k = \frac{1}{2}\eta(1+\zeta_i\zeta)(\zeta_i\zeta+2\eta-2) \quad k = 3,6$ $N_l = 2\xi(1-\xi-\eta)(1+\zeta_l\zeta) \quad l = 7,13$ $N_p = 2\xi\eta(1+\zeta_p\zeta) \quad p = 8,14$ $N_q = 2\eta(1-\xi-\eta)(1+\zeta_q\zeta) \quad q = 9,15$ $N_{10} = (1-\xi-\eta)(1-\zeta^2)$ $N_{11} = \xi(1-\zeta^2)$ $N_{12} = \eta(1-\zeta^2)$

Gauss integration points and weights

The following sections give the coordinate and weights for Gauss point integration for a given element topology. Gauss-Legendre quadrature of a function described in the interval(-1,+1) consists of weighted sum of function values at specified integration points. For 1-D problem the following expressions illustrate the method, which has straightforward extension to higher dimensions. The details of numerical quadrature/ cubature can be found in many standard published literatures.

Consider the integral

$$\int_a^b f(x)\ dx \qquad (1)$$

Using the following transformation

$$\xi = \frac{1}{b-a}[2x - (a+b)] \qquad (2)$$

The integral in equation (1) becomes

$$\int_{-1}^{+1} g(\xi)\ d\xi \qquad (3)$$

Equation (4) gives n point Gauss integration of equation (3),

$$\int_{-1}^{+1} g(\xi)\ d\xi = \sum_{i=1}^{n} w_i\, g(\xi_i) \qquad (4)$$

Where, w_i is the weight and ξ_i is the Gauss points. The following tables give the values of Gauss points and weights for n point integration for a specified finite element topology.

Table 1 : Integration points and weights for line, quadrilateral and hexahedron

Integration points n	**Coordinates ξ**	**Weights w_i**
1	0	2
2	$\pm\frac{1}{\sqrt{3}}$	1
3	0	$\frac{8}{9}$
	$\pm\sqrt{\frac{3}{5}}$	$\frac{5}{9}$
4	$\pm\sqrt{\frac{3-2\sqrt{\frac{6}{5}}}{7}}$	$\frac{18+\sqrt{30}}{36}$
	$\pm\sqrt{\frac{3+2\sqrt{\frac{6}{5}}}{7}}$	$\frac{18-\sqrt{30}}{36}$

Table 2 : Integration points and weights for a triangle

Integration points n	**Coordinates ξ**	**Weights w_i**
1	$\left(\frac{1}{3},\frac{1}{3}\right)$	1
2	$\left(\frac{2}{3},\frac{1}{6}\right),\left(\frac{1}{6},\frac{1}{6}\right),\left(\frac{1}{6},\frac{2}{3}\right)$	$\left(\frac{1}{3}\right)$
3	$\left(\frac{1}{2},0\right),\left(0,\frac{1}{2}\right),\left(\frac{1}{2},\frac{1}{2}\right)$	$\left(\frac{1}{3}\right)$
4	$\left(\frac{1}{3},\frac{1}{3}\right)$	$\left(-\frac{27}{48}\right)$
	$\left(\frac{3}{5},\frac{1}{5}\right),\left(\frac{1}{5},\frac{1}{5}\right),\left(\frac{1}{5},\frac{3}{5}\right)$	$\left(\frac{25}{48}\right)$

Table 3 : Integration points and weights for a tetrahedron

Integration points n	**Coordinates ξ**	**Weights w_i**
1	$\left(\frac{1}{4},\frac{1}{4},\frac{1}{4}\right)$	1

4	$(a,b,b),(b,b,b),(b,b,a),(b,a,b)$ Where, $a=\frac{5+3\sqrt{5}}{20}, b=\frac{5-\sqrt{5}}{20}$	$\frac{1}{4}$
5	$\left(\frac{1}{4},\frac{1}{4},\frac{1}{4}\right)$	$-\frac{4}{5}$
	$\left(\frac{1}{2},\frac{1}{6},\frac{1}{6}\right),\left(\frac{1}{6},\frac{1}{6},\frac{1}{6}\right),\left(\frac{1}{6},\frac{1}{6},\frac{1}{2}\right),\left(\left(\frac{1}{6},\frac{1}{2},\frac{1}{6}\right)\right)$	$\frac{9}{20}$

Table 4 : Integration points and weights for a pentahedron

Integration points n	**Coordinates** ξ	**Weights** w_i
2	$\left(\frac{1}{3},\frac{1}{3},\pm\frac{1}{\sqrt{3}}\right)$	$\frac{1}{2}$
9	$\left(\frac{1}{6},\frac{1}{6},\pm\sqrt{\frac{3}{5}}\right),\left(\frac{1}{6},\frac{2}{3},\pm\sqrt{\frac{3}{5}}\right),\left(\frac{2}{3},\frac{1}{6},\pm\sqrt{\frac{3}{5}}\right)$	$\left(\frac{5}{54}\right)$
	$\left(\frac{1}{6},\frac{1}{6},0\right),\left(\frac{1}{6},\frac{2}{3},0\right),\left(\frac{2}{3},\frac{1}{6},0\right),$	$\frac{8}{54}$
18		$\left(\frac{1}{12}\right)$

	$\left(\frac{1}{6},\frac{4}{6},\pm\sqrt{\frac{3}{5}}\right),\left(\frac{1}{6},\frac{1}{6},\pm\sqrt{\frac{3}{5}}\right),\left(\frac{4}{6},\frac{1}{6},\pm\sqrt{\frac{3}{5}}\right),$	
	$\left(\frac{1}{6},\frac{1}{6},0\right),\left(\frac{1}{6},\frac{4}{6},0\right),\left(\frac{4}{6},\frac{1}{6},0\right)$	$\frac{2}{15}$
	$\left(\frac{1}{2},\frac{1}{2},\pm\sqrt{\frac{3}{5}}\right),\left(2,0,\pm\sqrt{\frac{3}{5}}\right),\left(0,\frac{1}{2},\pm\sqrt{\frac{3}{5}}\right)$	$\frac{1}{108}$
	$\left(\frac{1}{2},\frac{1}{2},0\right),\left(\frac{1}{2},0,0\right),\left(0,\frac{1}{2},0\right)$	$\frac{2}{135}$

Element Features

Expressions for stiffness and mass matrix are given in this section.These expressions are general and true for any element topology.

Stiffness matrix

Mass matrix

Plane stress element

Plane strain element

Solid of revolution

Hexahedron element

Scalar/Spring element

Glue element

Truss and rod element

Beam element

Linear contact analysis using gap element

Kinematic constraints

Constitutive relations: orthotropic

- Plane stress
- Solid of revolution
- Solid element

Constitutive relations : incompressible materials

Stiffness matrix

Element stiffness is obtained from minimization of total potential, expressed as a function of displacement field within an element that usually is of simple geometry

$$\frac{\partial \Pi^e}{\partial U} = 0 \quad (1)$$

Where,

$$\Pi^e = \frac{1}{2}\int_{V^e} \sigma^T \varepsilon \, dV^e - \int_{V^e} F_b \, dV^e - \int_{S^e} F_S \, dS^e - \sum_i F_{p\,i} \quad (2)$$

V^e:Volume occupied by the structure; S^e:bounding surface of V^e

σ:Stress vector; ε:Strain vector;

F_b :Body force function; F_S : Surface traction function; F_p :discrete point forces

$\varepsilon = \partial U$;U:is the displacement field expressed as function of special coordinates.

The constitutive relation is $\sigma = D\varepsilon$; D: matrix of elastic moduli

Displacement u is sought at certain discrete locations within the element domain or boundary.The displacement field can be interpolated conveniently using a suitable interpolant. Hence,

$$U^e = N^e u_i^e \quad (3)$$

Therefore, the strain is $\varepsilon = \partial U = B u_i^e$ where, $B = \partial N^e$

Using the above expressions,

$$\frac{\partial \Pi^e}{\partial U} = \int_{V^e} B^{e^T} D^e B^e \, u \, dV^e - \int_{V^e} F_b \, dV^e - \int_{S^e} F_S \, dS^e - \sum_i F_{p\,i} = 0 \qquad (4)$$

The above expression(4) is represented as:

$$K_{ij}^e u_i^e = F_j^e \quad (5)$$

Where,

$$K_{ij}^e = \int_{V^e} B^{e^T} D^e B^e \, dV^e$$

And

$$F^e = \int_{V^e} F_b \, dV^e - \int_{S^e} F_S \, dS^e - \sum_i F_{p\,i}$$

Using Gauss-Legendre numerical cubature and quadrature the global stiffness matrix is obtained by assembling the element stiffness matrices as:

$$K = \sum_{i=1}^{M} \left(\sum_{\xi=1}^{n_g} \sum_{\eta=1}^{n_g} \sum_{\zeta=1}^{n_g} [B^e]^T D^e [B^e] u_j^e w_\xi w_\eta w_\zeta \right) \quad (6)$$

$$F = \sum_{i=1}^{M} \left(\sum_{\xi=1}^{n_g} \sum_{\eta=1}^{n_g} \sum_{\zeta=1}^{n_g} F_b^e \, w_\xi w_\eta w_\zeta + \sum_{\xi=1}^{n_g} \sum_{\eta=1}^{n_g} F_s^e w_\eta + F_p \right) \quad (7)$$

Where, n_is are the number of integration points, w_is are the weights and M the total number of elements used in idealizing the solid.

Mass matrix

The kinetic energy of an element is given by

$$T^e = \frac{1}{2}\int_{V_e} \rho \left(\dot{U}^e\right)^2 dV^e \qquad (1)$$

Eliminating the element displacement field with nodal values

$$T^e = \frac{1}{2}\{\dot{u}^e\}^T[m]^e\{\dot{u}^e\} \qquad (2)$$

Where,

$$[m]^e = \int_{V_e} \rho\,[N]^T[N]dV^e$$

Plane stress element

Degrees of freedom at each node: U_x, U_Y
Material constitutive law:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{1-v^2} \begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & \frac{(1-v)}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}$$

Strain components: $\varepsilon_x, \varepsilon_y, \gamma_{xy}, \varepsilon_z$
Stress components: $\sigma_x, \sigma_y, \tau_{xy}$
Rigid body modes: U_x, U_y, R_z
To be modeled in the x-y plane

Plane strain element

Degrees of freedom at each node: U_x, U_Y
Material constitutive law:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E(1-v)}{(1+v)(1-2v)} \begin{bmatrix} 1 & \frac{v}{1-v} & 0 \\ \frac{v}{1-v} & 1 & 0 \\ 0 & 0 & \frac{(1-2v)}{2(1-v)} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad (1)$$

Strain components: $\varepsilon_x, \varepsilon_y, \gamma_{xy}$
Stress components: $\sigma_x, \sigma_y, \tau_{xy}, \sigma_z$
Rigid body modes: U_x, U_y, R_z
To be modeled in the x-y plane

Solid of revolution

Degrees of freedom at each node: U_r, U_z
Material constitutive law:

$$\begin{Bmatrix} \sigma_r \\ \sigma_\theta \\ \sigma_z \\ \tau_{rz} \end{Bmatrix} = \frac{E(1-v)}{(1+v)(1-2v)} \begin{bmatrix} 1 & \frac{v}{1-v} & \frac{v}{1-v} & 0 \\ \frac{v}{1-v} & 1 & \frac{v}{1-v} & 0 \\ \frac{v}{1-v} & \frac{v}{1-v} & 1 & 0 \\ 0 & 0 & 0 & \frac{(1-2v)}{2(1-v)} \end{bmatrix} \begin{Bmatrix} \varepsilon_r \\ \varepsilon_\theta \\ \varepsilon_z \\ \gamma_{rz} \end{Bmatrix} \quad (1)$$

Strain components: $\varepsilon_r, \varepsilon_\theta, \varepsilon_z, \gamma_{rz}$
Stress components: $\sigma_r, \sigma_\theta, \sigma_z, \tau_{rz}$
Rigid body modes: U_z
To be modeled in the first and/or fourth quadrant of the x-y plane

Hexahedron element

Degrees of freedom at each node: U_x, U_y, U_z
Material constitutive law:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{Bmatrix} = \frac{E}{2(1+v)} \begin{bmatrix} \frac{2(1-v)}{(1-2v)} & \frac{2v}{(1-2v)} & \frac{2(1-v)}{(1-2v)} & 0 & 0 & 0 \\ \frac{2(1-v)}{(1-2v)} & \frac{2(1-v)}{(1-2v)} & \frac{2(1-v)}{(1-2v)} & 0 & 0 & 0 \\ \frac{2(1-v)}{(1-2v)} & \frac{2(1-v)}{(1-2v)} & \frac{2(1-v)}{(1-2v)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} \quad (1)$$

Strain components: $\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, \gamma_{xz}, \gamma_{yz}$
Stress components: $\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{xz}, \tau_{yz}$
Rigid body modes: $U_x, U_y, U_z, R_x, R_y, R_z$

Scalar / spring element

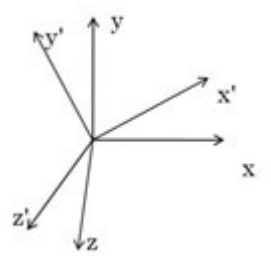
Spring elements are used to couple degrees of freedom between nodes, which needs to be specified with respect to a coordinate frame of reference. When a reference is not specified, it then implies that the stiffness are referred to global coordinate system. The physical separation of nodes has no mathematical/ computational consequences.

The element's inputs are: $k_{xx}, k_{yy}, k_{zz}, k_{\theta x \theta x}, k_{\theta y \theta y}, k_{\theta z \theta z}$

The stiffness transformation is: $[K] = [T]^T [K_e][T]$

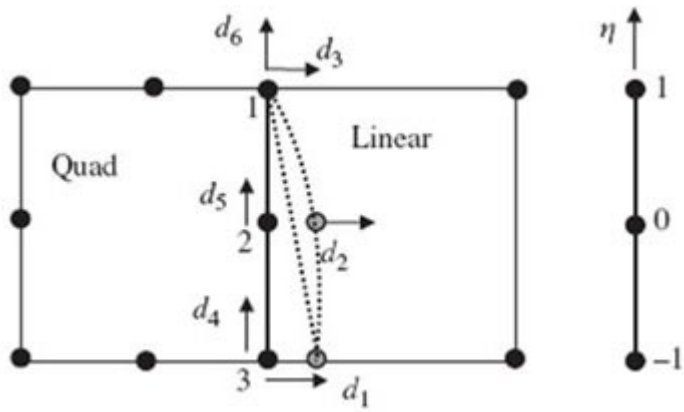
Where, $[T] = \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda \end{bmatrix}$, $\Lambda = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix}$ and the elements of Λ are direction cosines between axes as tabulated below.

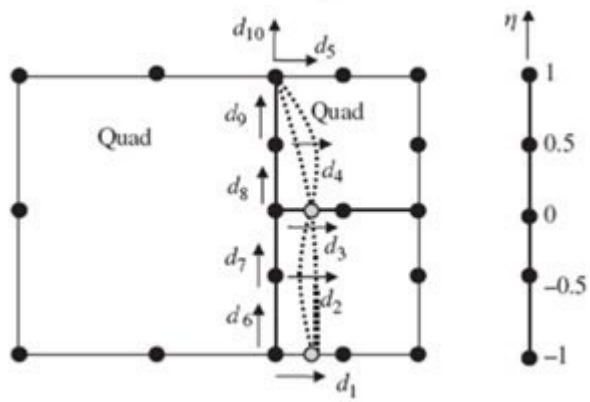
	x	**y**	**z**
x'	l_1	m_1	n_1
y'	l_2	m_2	n_2
z'	l_3	m_3	n_3



Glue element

These ‘elements’ are used when there is a need for maintaining continuity between incompatible meshes, when re-meshing can result in cost and time overruns.
The input consists of defining the nodal path as shown in following figures.





The relations among the variables are

$$d_x = 0.5(1-\eta)d_1 + 0.5(1+\eta)d_3$$

$$d_y = 0.5(1-\eta)d_4 + 0.5(1+\eta)d_6$$

$$0.5d_1 - d_2 + 0.5d_3 = 0$$

$$0.5d_4 - d_5 + 0.5d_6 = 0$$

$$d_x = -0.5\eta(1-\eta)d_1 + (1+\eta)(1-\eta)d_3 + 0.5\eta(1+\eta)d_5$$

$$d_2 = 0.25 * 1.5d_1 + 1.5 * 0.5d_3 - 0.25 * 0.5d_5$$

$$d_4 = -0.25 * 0.5d_1 + 0.5 * 1.5d_3 + 0.25 * 1.5d_5$$

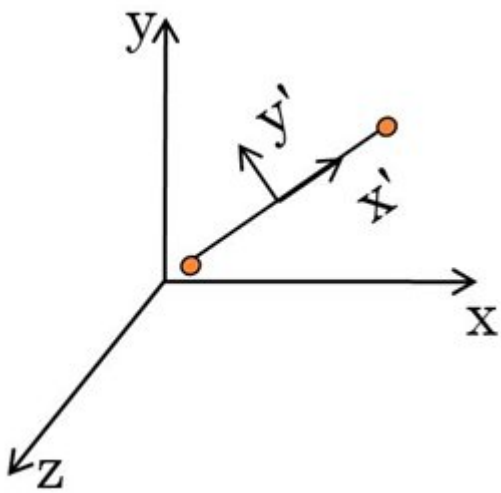
$$0.375d_1 - d_2 + 0.75d_3 - 0.125d_5 = 0$$

$$-0.125d_1 + 0.75d_3 - d_4 + 0.375d_5 = 0$$

$$0.375d_6 - d_7 + 0.75d_8 - 0.125d_{10} = 0$$

$$-0.125d_6 + 0.75d_8 - d_9 + 0.375d_{10} = 0$$

Truss and rod element



Orientation of truss or rod element in a 3-D space

The degree of freedom for truss element is along the axial direction at nodes located at either ends. For rod element the rotational degree of freedom at either node is defined to be on a plane perpendicular to its axis.

The element stiffness matrix of truss element is:

$$[K_e] = \frac{AE}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (1)$$

The element stiffness matrix of rod element is:

$$[K_e] = \frac{GJ}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (2)$$

The mass matrix for truss and rod element is:

$$[M_e] = \frac{\rho L}{12}\begin{bmatrix} 3A_1 + A_2 & A_1 + A_2 \\ A_1 + A_2 & A_1 + 3A_2 \end{bmatrix} \quad (3)$$

The stiffness and mass matrices obey, the following transformation law:

$$[K] = [T]^T [K_e][T] \quad (4)$$

$$[M] = [T]^T [M_e][T] \quad (5)$$

$[T]$ is described in scalar/spring element.

Beam element

Degrees-of-freedom for beam element consist of six translations and rotations per node. The non-zero members of the stiffness matrix are:

$$a_y = \frac{12EI_y}{L^3(1+\phi_z)}; \quad a_z = \frac{12EI_z}{L^3(1+\phi_y)}; \quad f_y = \frac{(2-\phi_z)EI_y}{L(1+\phi_z)}; \quad f_z = \frac{(2-\emptyset_y)EI_Z}{L(1+\emptyset_y)}$$

$$c_y = \frac{6EI_y}{L^2(1+\emptyset_z)}; \quad c_z = \frac{6EI_z}{L^2(1+\emptyset_y)}; \quad \emptyset_y = \frac{12EI_z}{K_y GAL^2}; \quad \emptyset_z = \frac{12EI_y}{K_z GAL^2};$$

$$e_y = \frac{(4+\emptyset_z)EI_y}{L(1+\emptyset_z)}; \quad e_z = \frac{(4+\emptyset_y)EI_z}{L(1+\emptyset_y)}$$

Where, K_y and K_z are shear factors that depends on cross-section, for circular section

$$K_y = K_z = 0.5$$

The stiffness matrix is:

$$[K_e] = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_z & 0 & 0 & 0 & c_z & 0 & -a_z & 0 & 0 & 0 & c_z \\ 0 & 0 & a_y & 0 & -c_y & 0 & 0 & 0 & -a_y & 0 & -c_y & 0 \\ 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 \\ 0 & 0 & -c_y & 0 & e_y & 0 & 0 & 0 & c_y & 0 & f_y & 0 \\ 0 & c_z & 0 & 0 & 0 & e_z & 0 & -c_z & 0 & 0 & 0 & f_z \\ -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -a_z & 0 & 0 & 0 & -c_z & 0 & a_z & 0 & 0 & 0 & -c_z \\ 0 & 0 & -a_y & 0 & c_y & 0 & 0 & 0 & a_y & 0 & c_y & 0 \\ 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\ 0 & 0 & -c_y & 0 & f_y & 0 & 0 & 0 & c_y & 0 & e_y & 0 \\ 0 & c_z & 0 & 0 & 0 & f_z & 0 & -c_z & 0 & 0 & 0 & e_z \end{bmatrix} \quad (1.0)$$

The transformation to any prescribed set of axis follows the law:

$$[K] = [T]^T [K_e][T]$$

Where,

$$[T] = \begin{bmatrix} \Lambda & 0 & 0 & 0 \\ 0 & \Lambda & 0 & 0 \\ 0 & 0 & \Lambda & 0 \\ 0 & 0 & 0 & \Lambda \end{bmatrix}; \quad \Lambda = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix}$$

and elements of Λ are described in Scalar/ spring element.

The non-zero consistent mass matrix of beam element are:

$$A_y = \frac{\frac{13}{35} + \frac{7}{10}\emptyset_z + \frac{1}{3}\emptyset_z{}^2 + \frac{6}{5}\left(\frac{r_y}{L}\right)^2}{(1+\emptyset_z)^2}; \; A_z = \frac{\frac{13}{35} + \frac{7}{10}\emptyset_y + \frac{1}{3}\emptyset_y{}^2 + \frac{6}{5}\left(\frac{r_z}{L}\right)^2}{\left(1+\emptyset_y\right)^2}$$

$$B_y = \frac{\frac{9}{70} + \frac{3}{10}\emptyset_z + \frac{1}{6}\emptyset_z{}^2 - \frac{6}{5}\left(\frac{r_y}{L}\right)^2}{(1+\emptyset_z)^2}; \; B_z = \frac{\frac{9}{70} + \frac{3}{10}\emptyset_y + \frac{1}{6}\emptyset_y{}^2 - \frac{6}{5}\left(\frac{r_z}{L}\right)^2}{\left(1+\emptyset_y\right)^2}$$

$$C_y = \frac{\left(\frac{11}{210} + \frac{11}{120}\emptyset_z + \frac{1}{24}\emptyset_z{}^2 + \left(\frac{1}{10} - \frac{1}{2}\emptyset_z\right)\left(\frac{r_y}{L}\right)^2\right)L}{(1+\emptyset_z)^2};$$

$$C_z = \frac{\left(\frac{11}{210} + \frac{11}{120}\emptyset_y + \frac{1}{24}\emptyset_y{}^2 + \left(\frac{1}{10} - \frac{1}{2}\emptyset_y\right)\left(\frac{r_z}{L}\right)^2\right)L}{\left(1+\emptyset_y\right)^2}$$

$$D_y = \frac{\left(\frac{13}{420} + \frac{3}{40}\emptyset_z + \frac{1}{24}\emptyset_z{}^2 - \left(\frac{1}{10} - \frac{1}{2}\emptyset_z\right)\left(\frac{r_y}{L}\right)^2\right)L}{(1+\emptyset_z)^2};$$

$$D_z = \frac{\left(\frac{13}{420} + \frac{3}{40}\emptyset_y + \frac{1}{24}\emptyset_y{}^2 - \left(\frac{1}{10} - \frac{1}{2}\emptyset_y\right)\left(\frac{r_z}{L}\right)^2\right)L}{\left(1 + \emptyset_y\right)^2}$$

$$E_y = \frac{\left(\frac{1}{105} + \frac{1}{60}\emptyset_z + \frac{1}{120}\emptyset_z{}^2 + \left(\frac{2}{15} + \frac{1}{6}\emptyset_z - \frac{1}{2}\emptyset_z^2 \quad\right)\left(\frac{r_y}{L}\right)^2\right)L^2}{(1 + \emptyset_z)^2};$$

$$E_z = \frac{\left(\frac{1}{105} + \frac{1}{60}\emptyset_y + \frac{1}{120}\emptyset_y{}^2 + \left(\frac{2}{15} + \frac{1}{6}\emptyset_y - \frac{1}{2}\emptyset_y^2 \quad\right)\left(\frac{r_z}{L}\right)^2\right)L^2}{\left(1 + \emptyset_y\right)^2}$$

$$F_y = \frac{\left(\frac{1}{140} + \frac{1}{60}\emptyset_z + \frac{1}{120}\emptyset_z{}^2 + \left(\frac{1}{30} + \frac{1}{6}\emptyset_z - \frac{1}{6}\emptyset_z^2 \quad\right)\left(\frac{r_y}{L}\right)^2\right)L^2}{(1 + \emptyset_z)^2};$$

$$F_z = \frac{\left(\frac{1}{140} + \frac{1}{60}\emptyset_y + \frac{1}{120}\emptyset_y{}^2 + \left(\frac{1}{30} + \frac{1}{6}\emptyset_y - \frac{1}{6}\emptyset_y^2 \quad\right)\left(\frac{r_z}{L}\right)^2\right)L^2}{\left(1 + \emptyset_y\right)^2}$$

$$\emptyset_y = \frac{12EI_z}{K_y GAL^2}; \emptyset_z = \frac{12EI_y}{K_z GAL^2} K_y = K_z = 1$$

The consistent mass matrix for the beam element is:

$$[M_e] = \rho AL \begin{bmatrix} \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 \\ 0 & A_z & 0 & 0 & 0 & C_z & 0 & B_z & 0 & 0 & 0 & -D_z \\ 0 & 0 & A_y & 0 & -C_y & 0 & 0 & 0 & B_y & 0 & D_y & 0 \\ 0 & 0 & 0 & \frac{J}{3A} & 0 & 0 & 0 & 0 & 0 & \frac{J}{6A} & 0 & 0 \\ 0 & 0 & -C_y & 0 & E_y & 0 & 0 & 0 & -D_y & 0 & F_y & 0 \\ 0 & c_z & 0 & 0 & 0 & e_z & 0 & -c_z & 0 & 0 & 0 & f_z \\ \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & B_z & 0 & 0 & 0 & D_z & 0 & A_z & 0 & 0 & 0 & -C_z \\ 0 & 0 & B_y & 0 & -D_y & 0 & 0 & 0 & A_y & 0 & C_y & 0 \\ 0 & 0 & 0 & \frac{J}{6A} & 0 & 0 & 0 & 0 & 0 & \frac{J}{3A} & 0 & 0 \\ 0 & 0 & D_y & 0 & F_y & 0 & 0 & 0 & C_y & 0 & E_y & 0 \\ 0 & -D_z & 0 & 0 & 0 & F_z & 0 & -C_z & 0 & 0 & 0 & E_z \end{bmatrix} \quad (2.0)$$

'Linear' contact analysis using gap ‘element’

It is of interest from engineering point of view to estimate the reaction forces, stress and strain states between deformable bodies when they come in contact due to the effects of tractions acting on the bodies. This type of problem is routinely encountered in many technological design situations. Theoretical treatment in classical framework, by Hertz can be found in many standard published literatures.

Contact analysis is a constrained minimization problem. If $\mathrm{C}(\mathbf{u})$ is a constraint function, and $\mathrm{f_R}$ the reaction at the contact interface, the Hertz-Signorini-Moreau condition in contact mechanics is given by

$$\mathrm{C}(\mathbf{u}) \geq 0, f_R \leq 0 \text{ and } f_R \mathrm{c}(\mathbf{u}) = 0 \quad (1.0)$$

This condition ensures non-penetrability of bodies in contact, when a penalty spring is introduced, which means for sufficiently large penalty parameter the solution converges. $f_R \leq 0$ indicates state of compressive type reaction induced at the contact locations. It is clear that gap ‘element’ implemented is a type of constraint rather than an element (related to finite element analysis) to perform contact analysis for small displacements and rotations.

In FEAST-*SMT*, contact capability is enabled using gap ‘element’. The analyst has to *a priori* identify the plausible locations, where contact between bodies can be expected. At these locations, gap ‘elements’ are introduced. The contact surfaces or lines must have compatible meshes. Friction is ignored.

The input specifications consist of initial gap value; gap stiffness to prevent penetration; spring stiffness to prevent rigid body mode and tolerance.

Kinematic constraints

The single point kinematic constraints can be homogenous or specified. Multi-point constraints between the degrees of freedom of nodes can be coupled by considering or omitting the separation distance.

Constitutive relations : orthotropic

In addition to element specific constitutive relations , which are reduced from 3-D isotropic stress-strain relations, orthotropic material constitutive law is enabled in FEAST*SMT* for following elements.

- Plane stress
- Solid of revolution
- Solid element

Plane stress

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{E_x}{1-\nu_{xy}\nu_{yx}} & \frac{\nu_{xy}E_y}{1-\nu_{xy}\nu_{yx}} & 0 \\ \frac{\nu_{xy}E_y}{1-\nu_{xy}\nu_{yx}} & \frac{E_y}{1-\nu_{xy}\nu_{yx}} & 0 \\ 0 & 0 & G_{xy} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{1.0}$$

Solid of revolution

$$\begin{Bmatrix} \sigma_r \\ \sigma_\theta \\ \sigma_z \\ \tau_{rz} \end{Bmatrix} = \begin{bmatrix} \frac{E_r(1-\nu_{\theta z}\nu_{z\theta})}{\Delta} & \frac{E_r(\nu_{\theta r}+\nu_{zr}\nu_{\theta z})}{\Delta} & \frac{E_r(\nu_{zr}+\nu_{\theta r}\nu_{z\theta})}{\Delta} & 0 \\ \frac{E_r(\nu_{\theta r}+\nu_{zr}\nu_{\theta z})}{\Delta} & \frac{E_\theta(1-\nu_{rz}\nu_{zr})}{\Delta} & \frac{E_\theta(\nu_{z\theta}+\nu_{z\theta}\nu_{r\theta})}{\Delta} & 0 \\ \frac{E_r(\nu_{zr}+\nu_{\theta r}\nu_{z\theta})}{\Delta} & \frac{E_\theta(\nu_{z\theta}+\nu_{z\theta}\nu_{r\theta})}{\Delta} & \frac{E_z(1-\nu_{r\theta}\nu_{\theta r})}{\Delta} & 0 \\ \frac{E_r(\nu_{zr}+\nu_{\theta r}\nu_{z\theta})}{\Delta} & \frac{E_z(1-\nu_{r\theta}\nu_{\theta r})}{\Delta} & 0 & G_{rz} \end{bmatrix} \begin{Bmatrix} \varepsilon_r \\ \varepsilon_\theta \\ \varepsilon_z \\ \gamma_{rz} \end{Bmatrix} \tag{1.0}$$

Where,

$$\Delta = 1 - (\nu_{r\theta}\nu_{\theta r} + \nu_{\theta z}\nu_{z\theta} + \nu_{rz}\nu_{zr} + \nu_{r\theta}\nu_{\theta z}\nu_{zr} + \nu_{rz}\nu_{\theta x}\nu_{z\theta})$$

Solid element

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{Bmatrix} = \begin{bmatrix} \frac{E_x(1-\upsilon_{yz}\upsilon_{zy})}{\Delta} & \frac{E_x(\nu_{yx}+\upsilon_{zx}\upsilon_{yz})}{\Delta} & \frac{E_x(\nu_{zx}+\upsilon_{yx}\upsilon_{zy})}{\Delta} & 0 & 0 & 0 \\ \frac{E_x(\nu_{yx}+\upsilon_{zx}\upsilon_{yz})}{\Delta} & \frac{E_y(1-\upsilon_{xz}\upsilon_{zx})}{\Delta} & \frac{E_y(\nu_{zy}+\upsilon_{zx}\upsilon_{xy})}{\Delta} & 0 & 0 & 0 \\ \frac{E_x(\nu_{zx}+\upsilon_{yx}\upsilon_{zy})}{\Delta} & \frac{E_y(\nu_{zy}+\upsilon_{zx}\upsilon_{xy})}{\Delta} & \frac{E_z(1-\upsilon_{xy}\upsilon_{yx})}{\Delta} & 0 & 0 & 0 \\ 0 & 0 & 0 & G_{xy} & 0 & 0 \\ 0 & 0 & 0 & 0 & G_{xz} & 0 \\ 0 & 0 & 0 & 0 & 0 & G_{yz} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix}$$

(1.0)

Where,

$$\Delta = 1 - (\nu_{xy}\nu_{yx} + \nu_{yz}\nu_{zy} + \nu_{xz}\nu_{zx} + \nu_{xy}\nu_{yz}\nu_{zx} + \nu_{xz}\nu_{yx})$$

Constitutive relations : incompressible materials

For incompressible / nearly incompressible structural materials like solid propellant grains or rubber like materials such as liner/insulation materials of rocket motors, standard displacement based finite elements are unable to capture the actual structural deformation due to volumetric locking phenomenon (i.e. , zero or nearly zero volumetric strain for the Poisson's ratio, $v \rightarrow \frac{1}{2}$). A mixed displacement/pressure element formulation is required to overcome the element locking in which the strain field is expressed in terms of deviatoric and volumetric change (due to hydrostatic pressure) of the body. The mixed Hellinger–Reissener formulation forms a basis to develop the most appropriate element. From the theory of elasticity, the stress–strain relation including thermal strain is given by,

$$\varepsilon_x - \alpha\Delta T = \frac{1}{E}\{\sigma_x - v(\sigma_y + \sigma_z)\} \quad (1)$$

$$\varepsilon_y - \alpha\Delta T = \frac{1}{E}\{\sigma_y - v(\sigma_z + \sigma_x)\} \quad (2)$$

$$\varepsilon_z - \alpha\Delta T = \frac{1}{E}\{\sigma_z - v(\sigma_x + \sigma_y)\} \quad (3)$$

$$\gamma_{xy} = \frac{1}{G}\tau_{xy} \quad (4)$$

$$\gamma_{yz} = \frac{1}{G}\tau_{yz} \quad (5)$$

$$\gamma_{zx} = \frac{1}{G}\tau_{zx} \quad (6)$$

Here E is the Young's modulus; v is the Poisson's ratio; the shear modulus,

$$G = \frac{E}{2(1+v)} \quad (7)$$

$\propto$ is the coefficient of thermal expansion and ΔT is the difference in temperature.
Components of the stress,

$$\{\sigma\} = \{\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{zx}\}^T \quad (8)$$

Components of the strain,

$$\{\varepsilon\} = \{\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}\}^T \quad (9)$$

Defining the hydrostatic pressure, volumetric strain, and bulk modulus,

$$P = \frac{1}{3}(\sigma_x + \sigma_y + \sigma_z), \quad (10)$$

$$\varepsilon_V = \varepsilon_x + \varepsilon_y + \varepsilon_z, \quad (11)$$

$$K = \frac{E}{3(1-2v)} \quad (12)$$

one can write after summing up the equations (1) to (3) in the form

$$\varepsilon_V - 3 \propto \Delta T = \frac{P}{K} \quad (13)$$

Total potential includes both the strain energy and work done due to external forces. The total potential equation formed by imposing the equation (13) as constraint will work for the compressible structural materials having the finite bulk modulus (K) and nearly incompressible materials having large bulk modulus .
The modified total potential for three dimensional body is

$$\pi(q,p) = \frac{1}{2}\int_V \{\varepsilon\}^T \{\sigma\} dV + \frac{P}{2}\int_V \left\{\varepsilon_v - 3\alpha\Delta T - \frac{P}{K}\right\} dV - \int_V \{q\}^T \{f_B\} dV - \int_{Sf} \{q\}_{Sf}{}^T \{f_{sf}\} dS \quad (14)$$

Here $\{f_B\}$ is the set of body forces acting in volume (V); $\{f_{Sf}\}$ is the set of traction forces acting on surface; $\{S_f\}$e; components of the displacement,

$$\{q\} = \{u, v, w\}^T \quad (15)$$

Expressing the strain in terms of deviatoric and volumetric strains, one can write

$$\{\varepsilon\} = \{\varepsilon_d\} + \left\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0,0,0\right\}^T \varepsilon_V \quad (16)$$

where the deviatoric strain is

$$\{\varepsilon_d\} = \left\{\left(\varepsilon_x - \frac{1}{3}\varepsilon_V\right), \left(\varepsilon_y - \frac{1}{3}\varepsilon_V\right), \left(\varepsilon_z - \frac{1}{3}\varepsilon_V\right), \gamma_{xy}, \gamma_{yz}, \gamma_{xz}\right\}^T \quad (17)$$

Similarly, the stress $\{\sigma\}$ is expressed in terms of the deviatoric stress $\{\sigma_d\}$ and the hydrostatic pressure (p) in the form

$$\{\sigma\} = \{\sigma_d\} + \{1,1,1,0,0,0\}^T P \quad (18)$$

where the deviatoric stress $\{\sigma_d\}$ is

$$\{\sigma_d\} = \{(\sigma_x - P), (\sigma_y - P), (\sigma_z - P), \tau_{xy}, \tau_{yz}, \tau_{xz}\}^T \quad (19)$$

Using equations (16) and (18), one can write equation (14) in the form

$$\pi(q,p) = \frac{1}{2}\int_V \{\varepsilon_d\}^T\{\sigma_d\}dV + \int_V P\{\varepsilon_v - 3\alpha\Delta T\}dV - \frac{1}{2}\int_V \frac{P^2}{K}dV - \int_V \{q\}^T\{f_B\}dV$$

$$- \int_{Sf} \{q\}_{Sf}^T\{f_{Sp}\}dS$$

(20)

Defining

$$\{q\} = [N]\{qi\}, \quad (21)$$

$$\{\varepsilon_d\} = [B_d]\{qi\}, \quad (22)$$

$$\{\sigma_d\} = [C_d][B_d]\{qi\}, \quad (23)$$

$$\varepsilon_v = \{B_V\}^T\{qi\}, \quad (24)$$

$$P = \{H\}^T\{p_i\} \quad (25)$$

Andputting in equation (20) and taking variation with respect to $\{qi\}$,and $\{p_i\}$, one obtains

$$\begin{bmatrix} [K_{qq}] & [K_{qp}] \\ [K_{qp}]^T & [K_{pp}] \end{bmatrix} \begin{Bmatrix} \{qi\} \\ \{Pi\} \end{Bmatrix} = \begin{Bmatrix} \{F_M\} \\ \{F_T\} \end{Bmatrix} \quad (26)$$

Where

$$[K_{qq}] = \int [B_d]^T [C_d][B_d] dV, \quad (27)$$

$$[K_{qp}] = [K_{pq}]^T = \int \{B_v\}^T\{H\} dV, \quad (28)$$

$$[K_{pp}] = -\frac{1}{K}\int_V \{H\}^T\{H\} dV, \quad (29)$$

$$\{F_M\} = \int_V [N]^T\{f_B\} dV + \int_S [N]^T\{f_{Sf}\} dS, \quad (30)$$

$$\{F_T\} = \int_V \{H\}^T 3 \propto \Delta T dV \quad (31)$$

$$\{qi\} = \{u_1, v_1, w_1, \ldots\ldots\ldots\ldots, u_n, v_n, w_n\}^T, \quad (32)$$

$\{Pi\} = \{P_1, P_2, \dots\dots\dots\dots, P_m\}^T$, (33)

n is the number of nodes in an element and m is the number of nodes in an element where the hydrostatic pressure is considered. Polynomial assumed for the hydrostatic pressure within the element is one order lower than that assumed for the displacement to satisfy equation (13).The shape function or interpolation function $[N]$ for the displacement field $\{q\}$ is defined same as the 20 node brick element, whereas the shape function $[H]$ for the hydrostatic pressure field $\{P\}$ is defined as the 8 node brick element. These functions are well defined in standard books. The matrix $[B_d]$ relates the deviotoric strain and the nodal displacements.The material matrixofthe deviotoric stress-strain relation $[C_d] = \{2G, 2G, 2G, G, G, G\}^T$ is ashear modulus (G) dependent diagonal matrix.Row matrix $\{B_y\}^T$ relates to the volumetric strain and the nodal displacements.
The displacement field, strain field and stress field for the plane strain element are

$\{q\} = \{u, v\}^T$ (34)

$\{\varepsilon\} = \{\varepsilon_x, \varepsilon_y, \gamma_{xy}\}^T$ (35)

$\{\sigma\} = \{\sigma_x, \sigma_y, \sigma_z, \tau_{xy}\}$ (36)

Strain normal to the plane is,

$\varepsilon_z = 0$ (37)

which implies that

$\sigma_z = v(\sigma_x + \sigma_y)$ (38)

Here u and v are the displacements along x and y directions.

The sub-matrices in the element stiffness matrix equation (26) for plane strain elements are

$$[K_{qq}] = \int_A [B_d]^T [C_d][B_d] dA \quad (39)$$

$$[K_{qp}] = [K_{pq}]^T = \int_A [B_v]^T \{H\} dA \quad (40)$$

$$[K_{pp}] = -\frac{1}{K} \int_A \{H\}^T \{H\} dA, \quad (41)$$

$$\{F_M\} = \int_A [N]^T \{f_B\} dA + \int_L [N]^T \{f_{Sf}\} dl, \quad (42)$$

$$\{F_T\} = \int_A \{H\}^T 3 \propto \Delta T dA \quad (43)$$

The displacement field, strain field and stress field for theaxi-symmetric solid of revolution element are

$$\{q\} = \{u, w\}^T \quad (44)$$

$$\{\varepsilon\} = \{\varepsilon_r, \varepsilon_z, \varepsilon_\theta, \gamma_{rz}\}^T \quad (45)$$

$$\{\sigma\} = \{\sigma_r, \sigma_z, \sigma_\theta, \tau_{rz}\}^T \quad (46)$$

Here u and w are the displacements along the radial (r) and axial (z) directions respectively.
The sub-matrices in the element stiffness matrix equation (26) for axi-symmetric solid of revolution elements are

$$[K_{qq}] = 2\pi \int_A [B_d]^T [C_d][B_d] r dA \quad (47)$$

$$[K_{qp}] = [K_{pq}]^T = 2\pi \int_A [B_v]^T \{H\} r dA \quad (48)$$

$$[K_{pp}] = -\frac{2\pi}{K}\int_A \{H\}^T\{H\}rdA, \quad (49)$$

$$\{F_M\} = 2\pi\int_A [N]^T\{f_B\}rdA + 2\pi\int_L [N]^T\{f_{Sf}\}rdl \quad (50)$$

$$\{F_T\} = 2\pi\int_A \{H\}^T 3 \propto \Delta T rdA \quad (51)$$

Solvers in FEAST- SMT

In finite element method, the global matrices are solved to obtain the field variables, which in the present case is displacement for static equilibrium problems and eigenvalues and vectors in case of stability and free vibration problems. In a typical real-world setting, the number of degrees of freedom is of the order 10^5 to 10^6. Efficient, accurate, reliable and fast algorithms are required to solve such matrix equations.

Solution to linear system of equations

- Cholesky Solver
- Multi-frontal Solver
- Pre-conditioned Conjugate Gradient solver
  - Pre-conditioners
    - Jacobi pre-conditioning
    - Incomplete factorization pre-conditioners
- Eigen Solvers
  - Lanczos Method
  - Progressive Simultaneous Iteration (PSI)

Solution to linear system of equations

The system of linear equations arising from a linear solution of a structure with n degrees of freedom is written as:

$$[K]\{d\} = \{f\}$$

where $[K]$ is n by n stiffness matrix, $\{d\}$ and $\{f\}$ not all non-zero are vectors of size n, which storesunknown displacements and loads respectively.

There two main methods for solving a system of linear equations are **iterative and direct methods**. Iterative methods are scalable and require less memory compared to direct methods, which make them a better choice for solving very large problems with limited computational resources. However, the convergence of iterative methods depends on the pre-conditioner used for a problem, and the execution time is unpredictable due to their iterative nature.

The direct methods, on the other hand, factorize stiffness matrix and once the factorization is complete, the system of equations can be solved efficiently for multiple RHS vectors by forward elimination and back substitution. The **sparseness** of the system is used to minimize the arithmetic operations and data storage required for the solution. These methods have high numerical precision and guarantee the solution within a predictable amount of time if computational resources are adequate.

The linear equation solvers implemented in FEAST-*SMT* are: Cholesky, Multi-frontal and Pre-conditioned Conjugate Gradient (PCG) solver.

Cholesky Solver

This method is based on Cholesky factorization of the coefficient matrix. If stiffness matrix $[K]$ is symmetric and positivedefinite, it can be factorized as $L\,L^T$, which is called the Choleskyfactorization. It requires fewer arithmetic operations compared to the LU factorization. This method requires the full assembly of element stiffness matrix to from a global stiffness matrix $[K]$. The matrix storage format used in FEAST-*SMT* is upper skyline. Each column is stored as a part vector with individual start and end indexes. The Cholesky decomposition of upper skyline matrix is explained in following steps.

<u>Step 1 :</u> :for i = 1 to N do Steps
<u>Step 2 :</u> col = PART-VECTOR (i)
<u>Step 3</u> : ifSIZE(col) == 0 go toStep 1
<u>Step 4 :</u> :for j = col.Head()+1 toi do // Update operation
col[j] = col[j] - (PART-VECTOR (j) * col) // vector multiplication
<u>Step 5 :</u> :utDu = 0.0
<u>Step 6 :</u> :for j = col.Head()+1 to i do
q = col[j]
if(DIAGONAL(j) <DBL_EPS) return false
col[j] = col[j] / DIAGONAL(j)
utDu = utDu + (q *col[j])
<u>Step 7 :</u> :DIAGONAL(i) = DIAGONAL(i) – utDu
<u>Step 8 :</u> :Stop

Once the global stiffness matrix $[K]$ is factored, the solution is obtained by subsequent forward elimination and backward substitution.

$[K]\{d\} = \{f\}$

$\{z\} = L]^{-1}\{f\}$ Forward elimination

$\{d\} = [L]^{-T}\{z\}$Backward substitution

Multi-frontal Solver

The global stiffness matrix, $[K]$ in the Cholesky method becomes a bottleneck for large order problems of millions of unknown variables. To avoid assembling global stiffness matrix, the element connectivity information is used to create an assembly tree data structure. Each leaf nodes of the assembly is associated with one element and represent eliminated nodes (not connected with other super elements) and remaining nodes (connected with one or more other elements). The intermediate nodes represent interconnected nodes of the super elements which can be eliminated along the assembly path or updated for the parent tree node. The assembly tree shows the relationship between factorization and triangular solution operations.

In multi-frontal method, numerical factorization is reduced to a series of partial factorization operations on dense frontal matrices. The frontal matrices are formed by assembling the stiffness associated with each nodal degrees-of-freedom. They are composed of three main components: diagonal factors L_B, off-diagonal factors L_{OFF}, and Schur complement S. Once the partial factorization is completed for a frontal matrix, only S is required for the subsequent partial factorization steps. After the partial factorization of a frontal matrix, diagonal and off-diagonal factors are not required until the triangular solution. The S matrix is stored in block storage format with out-of-core storage, to handle large size of update matrices on limited memory system architectures. Figure1 and Figure 2 below shows the assembly tree structure and solution steps involved in the multi-frontal method implemented in FEAST-*SMT*.

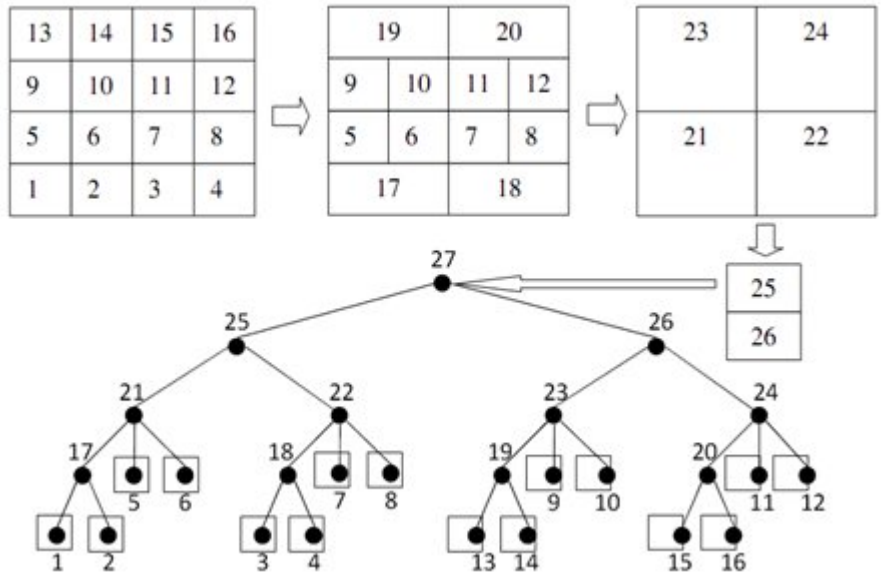


Figure 1:Assembly tree structure associated with a finite element model

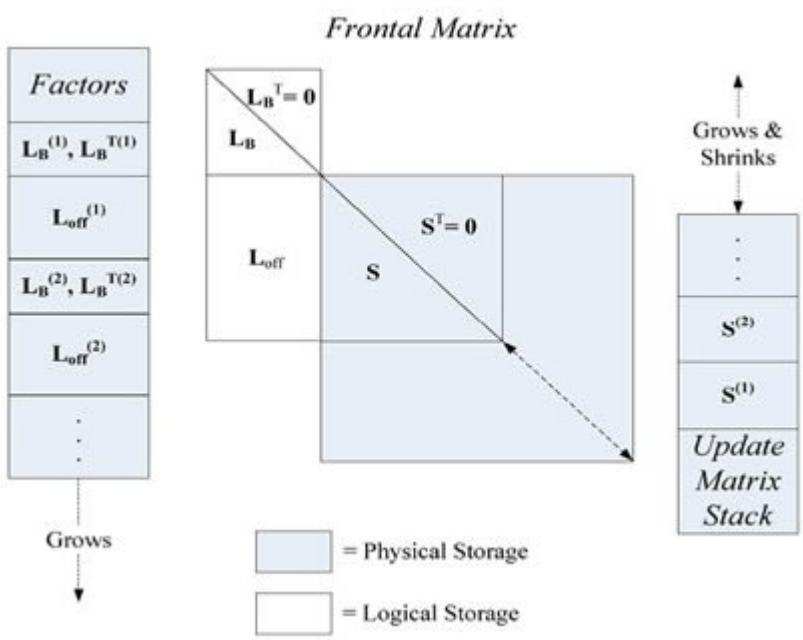


Figure2: Data structure used for multi-frontal method

Pre- conditioned Conjugate Gradient solver

The conjugate gradient method derives its name from the fact that it generates a sequence of conjugate (or orthogonal) vectors. These vectors are the residuals of the iterates. They are also the gradients of a quadratic functional, the minimization of which is equivalent to solving the linear system. CG is an extremely effective method when the coefficient matrix is symmetric positive definite, since storage for only a limited number of vectors is required.

```
Computer(0)= b - A x(0)for some initial guess x(0)
fori = 1 to size(A)do
      if (i == 1) then
      initialNorm  = ||r(0)||
      if(initialNorm == 0)return false
      else
      r(i-1)= r(i-2)- αi-1 q(i-1)
      endif
      if(||r(i-1)|| / initialNorm<ε) then
      returntrue
      endif
      solveMz(i-1)= r(i-1)
      ρi-1= r(i-1)Tz(i-1)
      if(i == 1) then
```

```
        β_{i-1} = 0.0
        p^(0) = z^(0)
        else
        β_{i-1} = ρ_{i-1} / ρ_{i-2};
        p^(i-1) = z^(i-1) + β_{i-1} p^(i-2)
        endif
        q^(i-1) = A p^(i-1)
        α_{i-1} = ρ_{i-1} / p^(i-1)T q^(i-1)
        x^(i) = x^(i-1) + α_{i-1} p^(i-1)
end
```

The method proceeds by generating vector sequences of iterates (i.e., successive approximations to the solution), residuals corresponding to the iterates, and search directions used in updating the iterates and residuals. Although the length of these sequences can become large, only a small number of vectors need to be kept in memory. Two inner products are performed in order to compute update scalars that are defined to make the sequences satisfy certain orthogonality conditions per iteration. On a symmetric positive definite linear system these conditions imply that the distance to the true solution is minimized in some norm.

The iterates $\mathbf{x}^{(i)}$ are updated in each iteration by a multiple $(\alpha_i)$of the search direction vector $\mathrm{p}^{(i)}$

$\mathbf{x}^{(i)} = \mathbf{x}^{(i-1)} + \alpha_{i-1}\mathrm{p}^{(i-1)}$ (1)

Correspondingly the residuals $\mathrm{r}^{(i)} = \mathrm{b} - \mathrm{A}\mathrm{x}^{(i)}$ are updated as

$\mathrm{r}^{(i)} = \mathrm{r}^{(i-1)} - \alpha_i\, \mathrm{q}^{(i)}$, where

$\mathrm{q}^{(i)} = A\mathrm{p}^{(i)}$ (2)

The choice $\alpha = \alpha_i = \mathrm{r}^{(i-1)T} \frac{\mathrm{r}^{(i-1)}}{\mathrm{p}^{(i)T}} A\mathrm{p}^{(i)}$ minimizes $\mathrm{r}^{(i-1)T}A^{-1}\mathrm{r}^{(i)}$ over all possible choices for α in equation (2).

The search directions are updated using the residuals

$\mathrm{p}^{(i)} = \mathrm{r}^{(i)} + \beta_{i-1}\mathrm{p}^{(i-1)}$ (3)

where the choice $\beta_i = \mathrm{r}^{(i)T} \frac{\mathrm{r}^{(i)}}{\mathrm{r}^{(i-1)T}} \mathrm{r}^{(i-1)}$ ensures that $\mathrm{p}^{(i)}$ and $A\,\mathrm{p}^{(i-1)}$ or equivalently, $\mathrm{r}^{(i)}$ and $\mathrm{r}^{(i-1)}$ are orthogonal. In fact, one can show that this choice of β_i makes $\mathrm{p}^{(i)}$ and $\mathrm{r}^{(i)}$ orthogonal to all previous $A\mathrm{p}^{(j)}$ and $\mathrm{r}^{(j)}$ respectively.

The matrix M indicates the pre-conditioner. When, $M = I$ we obtain the un-preconditioned version of conjugate gradient. The un-preconditioned conjugate gradient method constructs the ith iterate $\mathrm{x}^{(i)}$ as an element of $\mathrm{x}^{(0)} + span\,\{\mathrm{r}^{(0)}, \ \ldots, A^{-1}\mathrm{r}^{(0)}\}$ so that $(x^{(i)} - x)^T A(x^{(i)} - x)$ is minimized, where x is the exact solution of $A\mathrm{x} = b$. This minimum is guaranteed to exist in general only if A is symmetric positive definite. The preconditioned version of the method uses a different subspace for constructing the iterates, but it satisfies the same minimization property, although over this different subspace. It requires in addition that the pre-conditioner M is symmetric and positive definite.

The above minimization of the error is equivalent to the residuals $\mathrm{r}^{(i)} = b - Ax^{(i)}$ being M^{-1} orthogonal (i.e.,) $\mathrm{r}^{(i)T}M^{-1}r^{(j)} = 0\ if\ i \neq \mathrm{j}$. Since for symmetric A an orthogonal basis for the Krylov subspace span $\{\mathrm{r}^{(0)}, \ \ldots, A^{-1}\mathrm{r}^{(0)}\}$ can be constructed with only three-term recurrences, such a recurrence also suffices for generating the residuals. In the Conjugate Gradient method two coupled two-term recurrences are used; one that updates residuals is using a search direction vector, and one updating the search direction with a newly computed residual. This makes the Conjugate Gradient Method quite attractive computationally.

Pre-conditioners

The convergence rate of iterative methods depends on spectral properties of the coefficient matrix. Hence one may attempt to transform the linear system into one that is equivalent in the sense that it has the same solution, but that has more favourable spectral properties. A pre-conditioner is a matrix that effects such a transformation.
For instance, if a matrix $\boldsymbol{M}$ approximates the coefficient matrix $\boldsymbol{A}$ in some way, the transformed system

$\boldsymbol{M}^{-1}\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{M}^{-1}\boldsymbol{b}$ (1)

has the same solution as the original system $\boldsymbol{Ax} = \boldsymbol{b}$, but the spectral properties of its coefficient matrix $\boldsymbol{M}^{-1}\boldsymbol{A}$ may be more favourable. In devising a pre-conditioner, we are faced with a choice between finding a matrix $\boldsymbol{M}$ that approximates $\boldsymbol{A}$, and for which solving a system is easier than solving one with $\boldsymbol{A}$, or finding a matrix $\boldsymbol{M}$ that approximates $\boldsymbol{A}^{-1}$, so that only multiplication by $\boldsymbol{M}$ is needed. The majority of pre-conditioners fall in the first category.

Jacobi pre - conditioning

The simplest pre-conditioner consists of just the diagonal of the matrix:

$$\boldsymbol{m}_{ij} = \begin{cases} a_{ij} & if\ i = j \\ 0 & otherwise \end{cases}$$

This is known as the (point) Jacobi pre-conditioner.
It is possible to use this pre-conditioner without using any extra storage beyond that of the matrix itself. However, division operations are usually quite costly, so in practice storage is allocated for the reciprocals of the matrix diagonal.

Incomplete factorization pre - conditioners

A broad class of pre-conditioners is based on incomplete factorizations of the coefficient matrix. We call a factorization incomplete if during the factorization process certain fill elements, non zero elements in the factorization in positions where the original matrix had a zero, have been ignored. Such a pre-conditioner is then given in factored form $M = LU$ with L lower and U upper triangular. The efficiency of the pre-conditioner depends on how well M^{-1} approximates A^{-1}.

The most common type of incomplete factorization is based on taking a set S of matrix positions, and keeping all positions outside this set equal to zero during the factorization. The resulting factorization is incomplete in the sense that fill is suppressed.

The set S is usually chosen to encompass all positions (i, j) for which $a_{i,j} \neq 0$. A position that is zero in A but not so in an exact factorization is called a fill position, and if it is outside S, the fill there is said to be "discarded". Often S , is chosen to coincide with the set of nonzero positions in A, discarding all fill. This factorization type is called the $ILU(0)$ factorization: the incomplete LU factorization of level zero.

A is a large sparse square matrix, and x and b are vectorsor matrices. A and b are given and x is the unknown. Such systems of linear equations arise in awide range of scientific computing applications from various fields, in relation with numerical simulation: finite element methods, finite difference methods, or numerical optimization. There are two main classes of methods to solve such problems: direct methods, based on a factorization of A in, for example, theform LU,LDL^T,or QR; and iterative methods, in which sparse matrix-vector products are used tobuild a series of iterates, hopefully converging to the solution.

In addition to solving linear equations another important task of linear algebra is finding eigenvalues.A matrix can be considered an operator mapping a vector to another one. Foreigenvectors this mapping is a mere scaling.

Let $\boldsymbol{A}$ be a square matrix. If there is a real or complex number $\boldsymbol{\lambda}$ and a vector $\boldsymbol{x}$ such that

$\boldsymbol{Ax} = \boldsymbol{\lambda x}$ (1)

$\boldsymbol{\lambda}$ is an eigenvalue of the matrix and **x** an eigenvector. The equation (1) can also be written as

$(\boldsymbol{A} - \boldsymbol{\lambda I})\boldsymbol{x} = \boldsymbol{0}$ (2)

If the equation is to have a nontrivial solution, we must have

$\boldsymbol{det}(\boldsymbol{A} - \boldsymbol{\lambda I}) = \boldsymbol{0}$ (3)

When the determinant is expanded, we get the characteristic polynomial, the zeros of which are the eigenvalues. If the matrix is symmetric and real valued, the eigenvalues are real. Otherwise at least some of them may be complex. Complex eigenvalues appear always as pairs of complex conjugates.

Eigen solvers

Many physical models require eigen solution to determine its characteristics. This is important before studies are carried out to determine system response to stimuli. In most linear problems of solid mechanics, the resulting coefficient matrix of the system equation is square and symmetric. For such cases, the eigenvectors aid in decoupling the system matrix, which makes the solution determination faster and it can be used to represent system response as their linear combination.

Lanczos Method

The tri-diagonal reduction or is an automatic matrix reduction scheme whereby the Eigen solutions in the neighbourhood of a specified point in the Eigen spectrum can be accurately extracted from a tri-diagonal eigen value problem whose order is much lower than that of the full problem. Specifically, the order, m of the reduced problem is never greater than

$$m = 2q + 10 \quad (1)$$

Where q is the desired number of accurately computed eigenvalues.Thus, the intrinsic power of the method lies in the fact that the size of the reduced eigenvalue problem is of the same order of magnitude as the number of desired roots, even though the discretized system model may possess thousands of degrees of freedom.

From the standpoint of computational speed, the tri-diagonal reduction method is almost as fast as the Givens and Householder methods when all the Eigen solutions are calculated, and becomes increasingly more efficient as the number of required Eigen solutions is reduced. In addition, in order to avoid prohibitively long running times, both the Givens and Householder methods require the use of a relatively large computer central memory for even moderate problem sizes, while the tri-diagonal reduction method is extremely efficient with regard to core requirements.

Tri-diagonal reduction method employs only a single initial shift of eigenvalues and hence usually requires only one matrix decomposition. Itconsequently tends to be much more efficient than the Inverse Power Method when more than one or two Eigen solutions are required. The tri-diagonal reduction method is implemented in FEAST-*SMT* for real eigenvalue analysis as typified by structural vibration and buckling problems.

Representing in symmetric inverse form

The eigenvalue problem

$$[\boldsymbol{K} - \lambda_a \boldsymbol{M}]\{\phi\} = \boldsymbol{0} \quad (2)$$

Is converted to a symmetric inverse form

$$[\boldsymbol{B}]\{\boldsymbol{X}\} = \boldsymbol{\Lambda}[\boldsymbol{D}]\{\boldsymbol{X}\} \quad (3)$$

Where,

$$\boldsymbol{\Lambda} = \boldsymbol{1}/(\lambda_a - \lambda_0) \quad (4)$$

The Eigen solutions constitute the set lying closest to the specified point λ_0, in the Eigen spectrum and are a shift value, which is used only in structural vibration mode applications. The definitions of the eigenvalues λ_a, the matrices $[K]$ and $[M]$, their mathematical properties depend on the type of the problem solved: structural vibration and buckling problems. The equation (2) can be re-written as

For free vibration:

$$[K - \omega^2 M]\{\varphi\} = 0 \quad (5)$$

For elastic stability:

$$[K + \lambda K^d]\{\varphi\} = 0 \quad (6)$$

Where, $[K^d]$ represents differential stiffness matrix. If the user requests vibration modes in the neighbourhood of a specified frequency, equation (5) can be written as

$$[\check{K}]\{\varphi\} = \lambda'[M]\{\varphi\} \quad (7)$$

Where,

$[\check{K}] = [K - \omega^2{}_0 M]$ and $\lambda' = \omega^2 - \omega^2{}_0$

The resulting effective stiffness matrix $[\check{K}]$ is indefinite in this case, since it possesses both positive and negative eigenvalues. This requires that a non-square root decomposition scheme be used in subsequent operations. However, $\omega_0 = 0$ is taken as a default value, or it may be specified by the user. In this case, a specified number of natural frequencies starting with the lowest will be computed. In order to utilize a more efficient Cholesky decomposition $[\check{K}]$of under these conditions, a small negative shift $\lambda_0 = -\alpha^2$ is used, yielding

$$[\check{K}] = [K + \alpha^2 M] \quad (8)$$

and

$$\lambda' = \omega^2 + \alpha^2 \quad (9)$$

The resulting effective stiffness matrix $[\check{K}]$ is always positive definite provided that the system masses generate positive kinetic energy due to any kinematically admissible rigid body motions of the structure. This requirement is always satisfied by the mass matrix in a physically well posed problem, thereby allowing a Cholesky square root decomposition to be performed when the roots are computed in the neighbourhood of zero. Since no shifting is performed in buckling problems, the effective stiffness matrix is $[\check{K}] = [K]$, which is always positive definite, again permitting the use of a Cholesky decomposition.

To facilitate computation of eigenvalues closer to the point of interest within the Eigen spectrum, inverse forms of the eigen value problems are employed, as in the case of inverse power method with shifts.

Executing tri-diagonal reduction algorithm (Lanczos)

The general form of the inverse problem given in equation (3) is $[B]\{X\} = \Lambda[D]\{X\}$, where $[B] = [M][L^{-1}][d]^{-1}[L]^{-1}[M]$ for shifted vibration modes, while $[B] = [L^{-1}][K^d][L]^{-1}$ for buckling modes. $[D] = [M]$ forshifted vibration modes, while $[D] = [I]$ for buckling modes. The equation can be re-written as

$[\bar{B}]\{X\} = \Lambda\{X\}$ (10)

Where, $[\bar{B}] = [D^{-1}][B]$

The algorithm is given below:

<u>Step 1</u>: Initialize the recurrence algorithm

Initialize the vector index to $i = 0$ and set $\{v_0\} = \{0\}$

where $\{v_0\}$ is an $(n \times 1)$null vector

<u>Step 2</u>: Generate a starting /restart vector and set $d_{i+1} = 0.0$

Construct an n-element vector {w} using pseudo random number generator.
Solve for an un-normalized trial vector from equation

$\{\bar{v}_{i+1}\} = [\dot{B}]\{w\}$

$[\dot{B}] = [L^{-1}][d]^{-1}[L]^{-1}[M]$ (Shifted vibration modes)

$[\dot{B}] = [L^{-1}][K^d][L]^{-1}$ (buckling modes)

Forward and backward passes are used to perform the inverse operations.
Normalize the above vector:

$\{V^{(0)}{}_{i+1}\} = \left[1/\{\dot{V}_{i+1}\}^T[D]\{\dot{V}_{i+1}\}\right]^{1/2}\{\dot{V}_{i+1}\}$

$[D] = [M]$ (Shifted vibration modes)

$[D] = [I]$ (Buckling modes)

Set d_{i+1} and proceed to step 5

<u>Step 3</u>: Create one approximate Trial vector and one diagonal coefficient.
The recurrence algorithm is:

$$a_{i,i} = \{v_i\}^T[B]\{v_i\}$$

$$\{\bar{v}_{i+1}\} = [\bar{B}]\{v_i\} - a_{i,i}\{v_i\} - d_i\{v_{i-1}\}$$

$$\bar{d}_{i+1} = [\{\bar{v}_{i+1}\}^T[\bar{D}]\{v_{i+1}\}]^{1/2}$$

$$\{v^0{}_{i+1}\} = 1 / d_{i+1} * \{v_{i+1}\}$$

Where

$$[B] = [\bar{D}][B]$$

And $\{v^{(0)}{}_i + 1\}$ is an approximation to new trial vector.

<u>Step 4</u>: First normalization test
The test is

$$|d_{i+1}| > 10^{2-t}|a_{i,i}|$$

Pass: Proceed directly to **step 5**.

Fail: Return to **step 2**, generate a new restart vector for $\{v^0{}_{i+1}\}$ and then proceed to **step 5.**

<u>Step 5</u>: Iterate to obtain Orthogonalized vector.
Designate $\{\bar{X}_j\}, j = 1$, f as previously calculated and stored eigenvectors. Perform the iterations,

$$\{v^{(s+1)}{}_{(i+1)}\} = \{v^{(s)}{}_{(i+1)}\} - \sum_{j=1}^{i}[\{v_j\}^T[D]\{v^{(s)}{}_{(i+1)}\}]\{v_j\}$$

$$- \sum_{j=1}^{f}[\{\bar{X}_j\}^T[D]\{\bar{v}^{(s)}{}_{(i+1)}\}]\{X_j\};\ s = 0, 1, 2 \ldots$$

until

$$1 <= j <= i|[\{v_j\}^T[D]\{v^{(s)}{}_{(i+1)}\}]| <= 10^{2-t}$$

and

$$1 <= j <= f \quad |[\{\bar{X}_j\}^T[D]\{v^{(s)}{}_{(i+1)}\}]| <= 10^{2-t}$$

$$s = 14$$

If the orthogonality criterion is satisfied, proceed to **step 6**. Otherwise, set the problem size, $m = i$ and proceed to exit.

<u>Step 6</u>: Normalize the orthogonalized Trial vector. Compute

$$\{v_{i+1}\} = \{v^{(s+1)}{}_{(i+1)}\} / [\{v_{i+1}{}^{s+1}\}^T [D] \{v_{i+1}{}^{s+1}\}]^{1/2}$$

This is the new orthogonalized and normalized trial vector.

<u>Step 7</u>: Second normalization test and creation of off-diagonal coefficient
Compute the next off-diagonal term in the reduced tri-diagonal matrix from

$$d_{i+1} = \{v_{i+1}\}^T [B] \{v_i\}$$

Verify whether the following test is met:

$$|d_{i+1}| > 10^{2-t} |a|_{i,i}$$

If it has, set $i = i + 1$ and return to **Step 3** for continuation of the recurrence algorithm. If the test fails, set $m = i$ to reduce the problem size and proceed to exit. Issue message to user,that only i modes may have been requested.

Extracteigenvalues from reduced matrix using a QR algorithm

a. The coefficients $a_{11}, a_{22}, \ldots, a_{mm}$ and $d_2, d_3, \ldots, d_m$, computed are interpreted as the following symmetric, tri-diagonal array:

$$[A] = \begin{bmatrix} a_{1,1} & d_2 & & & & & \\ d_2 & a_{2,2} & d_3 & & & & \\ & d_3 & a_{3,3} & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & d_m \\ & & & & & d_m & a_{m,m} \end{bmatrix} \quad (11)$$

b. The m^{th}order eigenvalue problem

$$[A]\{y\} = \bar{\Lambda}\{y\} \quad (12)$$

is solved for the eigen values, $\bar{\Lambda}$ and eigenvectors $\{y_i\}$ using a Q-R algorithm and eigenvector computational procedure is given in following steps

Step 1: Renumber off-diagonal indexes for convenience.

```
for  i = 1tom-1do
     d_i = d_i+1
endfor
d_m = 0.0;
```

Step 2: for l = 1 to m do Step 3 to Step 11

Step 3:iter = 0

Step 4:Look for a single small sub-diagonal element to split the matrix.

```
for j = 1 to m-1 do
     dd = abs(a_j,j) +abs(a_(j+1),(j+1))
     if((abs(d_j) + dd) == dd) break;
endfor
```

Step 5: **if** (j == l) thengo to **Step 2**

Step 6:iter = iter +1

If (iter == 30) then return ERROR;

Step 7: Form shift

```
g = (a_(l+1),(l+1) - a_l,l) / (2.0*d_l)
r = SQRT(g*g+1)
g = ((a_j,j - a_l,l) + d_l) / (g+SIGN(r,g))
```

```
s = 1.0;
c = 1.0
p = 0.0
Step 8:for i = (j-1) toldo
        f = s * d_i
        b = c* d_i
        if (ABS(f)>=ABS(g))then
            c = g/ f
            r = SQRT((c* c) + 1)
            d_{i+1} = f* r
            c *= (s=1.0/r)
        else
            s = f/g
            r = SQRT((s*s)+1.0)
            d_{i+1} = g* r
            s *= (c=1.0/ r)

        endif
        g = a_{(i+1),(i+1)} - p
        r = (a_{i,I}-g)*s+2.0*c* b
        p = s*r;
        a_{(i+1),(i+1)} = g+p;
        g = c*r-b;
```

```
            for k=1 to m do
                f = q[k,(i+1)]
                y[k,(i+1)] = s*q[k,i] +c*f
                y[k,i] = c*q[k,i] - s*f
            endfor
        endfor
Step 9:if(r == 0.0 && I > 0) then go to Step 4
Step 10:a1,1 = a1,1- p
        d1 = g
        di = 0.0
Step 11: if(m != 1) then go to step 4
Step 12:terminate the program
```

On execution of the QR program, the diagonal array $a_{i,i}$, $1 <= I <= m$ returns the eigenvalues. The eigenvectors of the tri-diagonal matrix are computed in vector $\boldsymbol{y_i}$, which is an identity at start of the program.

c. The reduced system of eigenvectors are normalized so that

$\{\boldsymbol{y}_i\}^T\{\boldsymbol{y}_i\} = \mathbf{1};\ \ i = 1,m$

Progressive Simultaneous Iteration (PSI)

In progressive simultaneous iteration process a set of vectors is used for simultaneous determination of a number of roots and associated eigenvectors. It is a novel simultaneous iteration procedure with significantly accelerated convergence characteristics. The numerical algorithm specified in the following steps for a desired NR number of roots in the vicinity of a specified shift value λ^s and associated modes are the usual requirement. With $\lambda^s = 0$, the algorithm computes the first few desired roots only. Thus the original eigenvalue program is written in the shifted form:

$$(K - \lambda^s M - (\lambda - \lambda^s)M)\varphi = 0$$

Or

$$(\bar{K} - \bar{\lambda} M)\varphi = 0$$

With $\bar{\lambda} = \lambda - \lambda^s$ and solution is sought for NR number of roots in the vicinity of λ^s.

<u>Step 1</u>: Form a set of Number of Random Trial vectors, NRT; NRT > NR (number of requested eigenvalues):

$$\hat{\varphi}1 = [\hat{\varphi}_{11}\hat{\varphi}_{12}\hat{\varphi}_{13} \dots \hat{\varphi}_1 NRT]$$

<u>Step 2</u>: Perform Cholesky factorization of $[\bar{K}]$ matrix, initially only $[\bar{K}] = LDL^T$, where L and D are a unit lower triangular and a diagonal matrix, respectively.

<u>Step 3</u>: At a typical ith iteration step, solve $[\bar{K}]\{\varphi_{i+1}\} = [\hat{M}]\{\varphi_i\}$ progressively, involving only unconverged roots, by the usual back substitution method.

<u>Step 4</u>: Estimate the magnitude of the jth eigenvalue λ^j, progressively, on unconverged roots only, as

$\lambda^j{}_{i+1} = ||\varphi_i{}^j|| \,/\, ||\varphi_{i+1}{}^j||$

If this is not the first iteration, check the convergence using the test

$(\lambda^j{}_{i+1} - \lambda_i{}^j)\,/\lambda^j{}_{i+1} <= EPSN$

Let $NR1$ be the number of leading consecutive roots that have converged; $EPSN$ is the specified norm tolerance factor. If, $NR1 = NR$ go to **Step 7**.

<u>Step 5</u>: Perform progressive mass orthonormalization on converged (NRT – NR1) vectors only:

$$\hat{\varphi}^k{}_{i+1} = \varphi^k{}_{i+1} - \sum_{l=1}^{NR1}[(\varphi^l)^T\, M\, \varphi^k{}_{i+1}]\varphi^l - \sum_{l=NR1+1}^{k-1}[(\hat{\varphi}^l{}_{i+1})^T M \varphi^k{}_{i+1}]\hat{\varphi}^l{}_{i+1}$$

$$\hat{\varphi}^k{}_{i+1} = \hat{\varphi}^k{}_{i+1} \;/\; [(\hat{\varphi}^k{}_{i+1})^T\, M\, (\hat{\varphi}^k{}_{i+1})]^{1/2}, k = NR1 + 1, NRT$$

<u>Step 6</u>: Continue iteration, go to Step 3; $\hat{\varphi}_{i+1}$ being made of $\hat{\varphi}^k{}_{i+1}$ vector:

$$\hat{\varphi}_{i+1} = [\hat{\varphi}^{NR1+1}{}_{i+1}\hat{\varphi}^{NR1+2}{}_{i+1}\hat{\varphi}^{NR1+3}{}_{i+1} \cdots\cdots \hat{\varphi}^{NR1-1}{}_{i+1}]$$

<u>Step 7</u>: Compute roots by Rayleigh-Ritz method, employing the progressive procedure involving only the unconverged roots by solving the further reduced eigenvalue problem.

$[\hat{K}_{i+1}]\{X_{i+1}\} = \Lambda[\hat{M}_{i+1}]\{X_{i+1}\}$

λ being an approximation of λ and in which

$$[\hat{K}_{i+1}] = (\hat{\varphi}_{(i+1)})^T[\hat{K}](\hat{\varphi}_{(i+1)}), [\hat{M}_{(i+1)}] = (\hat{\varphi}_{(i+1)})^T\,[\hat{M}](\hat{\varphi}_{(i+1)})$$

Noting that $[\hat{M}_{(i+1)}] = I$ because of the mass orthonormalization effected in step 5.

<u>Step 8</u>: Perform a root convergence test for each root under consideration:

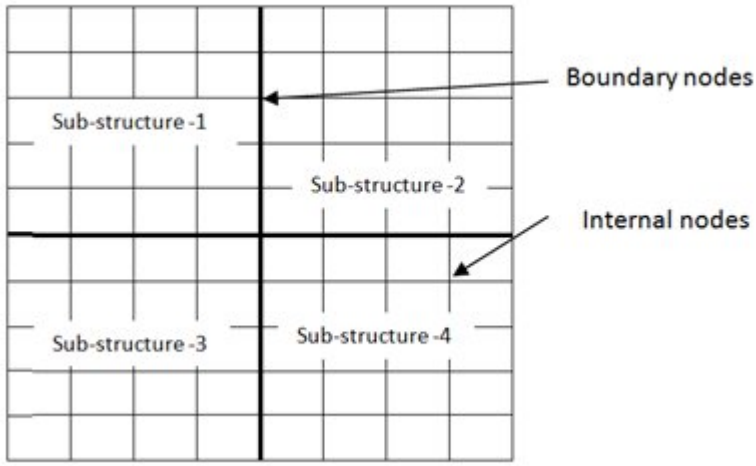
$$|(\hat{\lambda}^{j}_{i+1} - \hat{\lambda}^{j}_{i})| / |\hat{\lambda}^{j}_{i+1}| <= EPS$$

EPS being a specified root convergence factor. Update NR1 to be the number of consecutive leading roots that have converged.

<u>Step 9</u>: Recalculate eigenvectors $\hat{\boldsymbol{\varphi}}_{i+1} = \hat{\boldsymbol{\varphi}}_{i+1}\boldsymbol{X}_{i+1}$; set $\boldsymbol{\varphi}_{i+1} = \boldsymbol{\varphi}_{i+1}$**,** noting that $\boldsymbol{X}_{i+1}$ must be orthonormalized, so that $\boldsymbol{X}^{T}{}_{i+1}\boldsymbol{X}_{i+1} = \mathbf{I}$.

<u>Step 10</u>: If $NR1 < NR$, then go to step 3 and perform **Steps 3, 5, 7,8** and **9** only. Otherwise go to **Step 11**.

<u>Step 11</u>: Terminate the program



Schematic representation of sub-structure with boundary and internal nodes