Finite Element Vibrations and Dynamics Studies

MECH 417, Rice University, J.E. Akin, Draft 1, 4/14/20

Introduction

The extension of static FEA formulations to dynamics and vibrations can be done by at least three different approaches: 1. Apply Newton's laws to define the acceleration body forces (and then use D'Alembert's principle), 2. Add kinetic energy (KE) to the work-energy formulation, 3. Apply Galerkin's method to the differential equations of motion that include time, τ. All approaches require that time be brought into the FEA approximation. There are two choices there: classical separation of variables as a product of spatial behavior and time behavior, or utilize a four-dimensional Space-Time formulation (for which Prof. T. Tezduyar is world famous).

Interpolations

Here the classic approach will be used. Up to this point the spatial behavior has been interpolated with polynomials from time invariant nodal values of the solution, $u(x,y,z) = \boldsymbol{H}(x,y,z)\,\boldsymbol{u^e}$, but now it is necessary to recognize that all of the nodal values can change with time ($\boldsymbol{u^e} \rightarrow \boldsymbol{u^e}(\tau)$). The calculation of spatial derivatives change slightly, but new derivatives with respect to time must be added. The interpolation functions generalize to

$$u(x,y,z,\tau) = \boldsymbol{H}(x,y,z)\,\boldsymbol{u^e}(\tau)$$

with spatial gradient terms

$$\frac{\partial u(x,y,z,\tau)}{\partial x} = \frac{\partial \boldsymbol{H}(x,y,z)}{\partial x}\boldsymbol{u^e}(\tau) + 0; \qquad \frac{\partial u}{\partial y} = \frac{\partial \boldsymbol{H}}{\partial y}\boldsymbol{u^e}(\tau); \qquad \frac{\partial u}{\partial z} = \frac{\partial \boldsymbol{H}}{\partial z}\boldsymbol{u^e}(\tau)$$

a new velocity term

$$\dot{u} = \frac{\partial u(x,y,z,\tau)}{\partial \tau} = 0 + \boldsymbol{H}(x,y,z)\frac{\partial \boldsymbol{u^e}(\tau)}{\partial \tau} = \boldsymbol{H}(x,y,z)\dot{\boldsymbol{u}}^{\boldsymbol{e}}, \quad \dot{\boldsymbol{u}}^{\boldsymbol{e}} \equiv \frac{\partial \boldsymbol{u^e}(\tau)}{\partial \tau}$$

and an acceleration term

$$\ddot{u} = \frac{\partial^2 u(x,y,z,\tau)}{\partial \tau^2} = 0 + \boldsymbol{H}(x,y,z)\frac{\partial^2 \boldsymbol{u^e}(\tau)}{\partial \tau^2} = \boldsymbol{H}(x,y,z)\ddot{\boldsymbol{u}}^{\boldsymbol{e}}, \quad \ddot{\boldsymbol{u}}^{\boldsymbol{e}} \equiv \frac{\partial \boldsymbol{u^e}(\tau)}{\partial \tau^2}$$

In other words, in addition to the system nodal displacements it is necessary to store and calculate the velocity of all nodes, and the acceleration of all nodes.

Kinetic Energy

The scalar mechanical work, and the scalar potential energy used in static models. To introduce typical time dependent quantities the concept of kinetic energy (KE) will be re-formulated into a matrix expression. In physics, the kinetic energy of a mass particle is half the mass, *m*, times the velocity squared (here written as a triple product). Initially, let $u(x,\tau)$ be the axial displacement of a particle with time. Then, the velocity is $\dot{u}$ and the KE is

$$KE_{particle} = \frac{1}{2}\,\dot{u}\,m\,\dot{u} \quad \left(= \frac{1}{2}\left(m\dot{u}_x^2 + m\dot{u}_y^2 + m\dot{u}_z^2\right) \text{for a general velocity vector}\right)$$

For a differential mass, $dm = \rho \mathrm{dV}$, formed by the mass density, ρ, in a differential volume, dV, its KE is

$$KE_{dm} = \frac{1}{2}\,\dot{u}\,dm\,\dot{u} = \frac{1}{2}\,\dot{u}\,\rho \mathrm{dV}\,\dot{u}$$

For a differential volume in a bar element, with n_n nodes, the total kinetic energy is obtained by integrating over the total mass (volume).

$$KE^e = \frac{1}{2}\int_{V^e} (\boldsymbol{H}(x)\dot{\boldsymbol{u}}^{\boldsymbol{e}})^T\,\rho^e\,(\boldsymbol{H}(x)\dot{\boldsymbol{u}}^{\boldsymbol{e}})\,dV = \frac{1}{2}\,(\dot{\boldsymbol{u}}^{\boldsymbol{e}})^T\left[\int_{L^e} \boldsymbol{H}^T(x)\,\rho^e \boldsymbol{H}(x)\,A(x)\,dx\right]\dot{\boldsymbol{u}}^{\boldsymbol{e}}$$

The integrand is a square symmetric matrix, known as the mass matrix

$$KE^e = \frac{1}{2}\,(\dot{\boldsymbol{u}}^{\boldsymbol{e}})^T[\boldsymbol{M}^{\boldsymbol{e}}]\dot{\boldsymbol{u}}^{\boldsymbol{e}}$$

For a bar element, with $n_i = n_n\, n_g$ independent degrees of freedom, the element square mass matrix is

$$[\boldsymbol{M}^{\boldsymbol{e}}] = \int_{L^e} \boldsymbol{H}^T(x)\,\rho^e \boldsymbol{H}(x)\,A(x)\,dx$$

$$(n_i \times n_i) = (n_i \times 1)(1 \times 1)(1 \times n_i)$$

For the previous two-noded linear bar element this mass matrix becomes

$$[\boldsymbol{M}^{\boldsymbol{e}}] = \frac{\rho^e A^e L^e}{6}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix} = \frac{m^e}{6}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}$$

Note that the sum of all of the coefficients always equals the total mass of the element, $m^e = m^e(2+1+1+2)/6$. Similarly, the quadratic line element has a mass matrix of

$$[\boldsymbol{M}^{\boldsymbol{e}}] = \frac{\rho^e A^e L^e}{30}\begin{bmatrix}4 & 2 & -1\\ 2 & 16 & 2\\ -1 & 2 & 4\end{bmatrix}$$

These matrix forms were seen in past applications involving heat convection, and structures supported with an elastic foundation; and it was referred to then as the 'generalized mass matrix'. Now, this is the literal mass matrix since the included element physical data is the mass density, ρ^e. In general, the KE of an element is half the triple matrix product of its mass matrix, $[\boldsymbol{M}^{\boldsymbol{e}}]$, pre- and post-multiplied by the vector of nodal velocities of the element.

Interpolating Vector Components

Previously, the interpolation function row matrix, $\boldsymbol{H}(x)$, has been applied to any single scalar unknown (or just the x-component of the displacement vector). In general, the displacement vector has three components ($n_g = 3$):

$$\vec{\boldsymbol{u}}(x,y,z,\tau) = u\vec{\boldsymbol{\imath}} + v\vec{\boldsymbol{\jmath}} + w\vec{\boldsymbol{k}}$$

and each scalar component is interpolated as before:

$$\begin{aligned} u(x,y,z,\tau) &= \boldsymbol{H}(x,y,z)\,\boldsymbol{u}^{\boldsymbol{e}}(\tau),\\ v(x,y,z,\tau) &= \boldsymbol{H}(x,y,z)\,\boldsymbol{v}^{\boldsymbol{e}}(\tau),\\ w(x,y,z,\tau) &= \boldsymbol{H}(x,y,z)\,\boldsymbol{w}^{\boldsymbol{e}}(\tau). \end{aligned} \qquad (13.8\text{-}1)$$

When interpolating three components of a displacement vector most finite element literature uses a rectangular interpolation matrix, $[\boldsymbol{N}(x,y,z)]$; which contains the scalar interpolation terms and a majority of zero entries.

$$\vec{\boldsymbol{u}} \leftrightarrow \boldsymbol{u} = \begin{Bmatrix} u(x,y,z,\tau)\\ v(x,y,z,\tau)\\ w(x,y,z,\tau)\end{Bmatrix} = \boldsymbol{N}(x,y,z)\boldsymbol{\delta}^{\boldsymbol{e}}(\tau)$$

$$\boldsymbol{\delta}^{\boldsymbol{e}T} = \left[u_1, v_1, w_1, u_2, \cdots, \cdots, u_{n_n}, v_{n_n}, w_{n_n}\right]^e \qquad (13.8\text{-}2)$$

$$\boldsymbol{N}(x,y,z) = \left[\boldsymbol{N}_1|\boldsymbol{N}_2|\cdots|\boldsymbol{N}_{n_n}\right],\ \boldsymbol{N}_k = \begin{bmatrix} H_k(x,y,z) & 0 & 0\\ 0 & H_k(x,y,z) & 0\\ 0 & 0 & H_k(x,y,z)\end{bmatrix} \qquad (13.8\text{-}5)$$

In other words, for a solid element the mass matrix is three times larger (since $n_g = 3$) than the bar element and is written as

$$[\boldsymbol{M^e}] = \int_{V^e} \boldsymbol{N}^T\,(x,y,z)\,\rho^e(x,y,z)\,\boldsymbol{N}(x,y,z)\;\;dV$$

Of course, if the element is curved or distorted from the parametric shape then the mass matrix, like the stiffness matrix, must be integrated numerically. Also, the square mass matrix, $[\mathbf{M^e}]$, is assembled in exactly the same way as the square system stiffness matrix. Sometimes a node is attached to an object that can be approximated as a point mass. Then that mass is simply added to the diagonal of the assembled mass matrix in the rows associated with that node.

Acceleration Body Force

While the velocity leads to KE and an element mass matrix, the acceleration vector leads to a body force per unit volume due to the mass density, ρ. In static equilibrium the FEA load conditions considered before included the resultant vector due to a body force per unit volume, $\vec{\mathrm{f}} \leftrightarrow \{\boldsymbol{f}\}$. The resulting column vector was

$$\{\boldsymbol{c_f^e}\} = \int_{\Omega^e} [\boldsymbol{N^{eT}}]\,\{\boldsymbol{f^e}\}\,d\Omega$$

Newton's second law of motion gives $\vec{\boldsymbol{f}}d\mathrm{V} = \rho\,\vec{\boldsymbol{a}}\,dV$ so that $\vec{\boldsymbol{f}} = \rho\,\vec{\boldsymbol{a}} \leftrightarrow \{\boldsymbol{f}\} = \rho\{\ddot{\boldsymbol{u}}\}$ in matrix notation. The most common cases of the body force are the weight due to a constant gravitational acceleration, $\vec{\boldsymbol{f}} = \rho\vec{\boldsymbol{g}}$; the radial body force due to rotation about an axis with an angular velocity of ω, $f_r = \rho\,r\,\omega^2$; the tangential body force due to rotation about an axis with an angular acceleration of α, $f_t = \rho\,r\,\alpha$. For a time variant acceleration vector, $\overrightarrow{\boldsymbol{a}(\tau)}$, at a point in an element its value is obtained by interpolating from the nodal accelerations of the element

$$\{\ddot{\boldsymbol{u}}(\tau)\} = [\boldsymbol{N}(x,y,z)]\{\boldsymbol{a^e}(\tau)\} = [\boldsymbol{N}(x,y,z)]\left\{\ddot{\boldsymbol{\delta}^{\boldsymbol{e}}}(\tau)\right\}$$

That gives the body force resultant as

$$\{\boldsymbol{c_f^e}(\tau)\} = \int_{\Omega^e} [\boldsymbol{N}^{\boldsymbol{T}}(x,y,z)]\,\rho^e[\boldsymbol{N}(x,y,z)]\left\{\ddot{\boldsymbol{\delta}^{\boldsymbol{e}}}(\tau)\right\}d\Omega = \int_{\Omega^e} [\boldsymbol{N}^{\boldsymbol{T}}]\,\rho^e[\boldsymbol{N}]\,d\Omega\,\left\{\ddot{\boldsymbol{\delta}^{\boldsymbol{e}}}\right\} = [\mathbf{M^e}]\left\{\ddot{\boldsymbol{\delta}^{\boldsymbol{e}}}\right\}$$

which again gives rise to the mass matrix. However, here the nodal accelerations are unknowns.

D'Alembert's principle

Recall that for static equilibrium an element with stiffness matrix $[\boldsymbol{K^e}]$, with point loads, $\{\boldsymbol{c_P^e}\}$, and gravity loading, $\left\{\boldsymbol{c_f^e}\right\}$, the matrix equations of equilibrium (before BCs) were

$$[\boldsymbol{K^e}]\{\boldsymbol{\delta^e}\} - \{\boldsymbol{c_P^e}\} - \left\{\boldsymbol{c_f^e}\right\} = \{\mathbf{0}\}$$

But now the above equation converts to an equation of motion because the body force resultants, $\left\{\boldsymbol{c_f^e}\right\}$, includes the unknown acceleration. Instead of using Newton's second law of motion, $\sum \vec{\boldsymbol{F}}\,(\tau) = m\,\vec{\boldsymbol{a}}(\tau)$, as the equations of motion, D'Alembert's principle is to rewrite this as a pseudo-static equilibrium problem

$$\sum \vec{\boldsymbol{F}}\,(\tau) - m\,\vec{\boldsymbol{a}}(\tau) = \vec{\mathbf{0}}$$

and introduce an inertial body force vector due to the acceleration, $\vec{\boldsymbol{F}}_I = -m\,\vec{\boldsymbol{a}}(\tau)$, and view the problem as static equilibrium:

$$\sum \vec{\boldsymbol{F}}\,(\tau) + \vec{\boldsymbol{F}}_I = \vec{\mathbf{0}}$$

Here, as usual, the 'inertia force', is the negative of the acceleration contribution $\vec{\boldsymbol{F}}_I = -[\mathbf{M^e}]\left\{\ddot{\boldsymbol{\delta}^{\boldsymbol{e}}}\right\}$

Matrix Equations of Motion:

The governing matrix equation is

$$[K^e]\{\delta^e\} - \{c_P^e\} + [M^e]\{\ddot{\delta}^e\} = \{0\}$$

and recognizing the external forces could depend on time, $\{c_P^e(\tau)\}$, the assembled system equation of motion is

$$[M\,]\{\ddot{\delta}\,\} + [K\,]\{\delta\,\} = \{c_P(\tau)\}$$

In addition to being modified to enforce the Dirichlet boundary conditions this system has to meet Initial Conditions (IC) on the displacements, and the velocities at $\tau = 0$. In a similar way force due to viscous damping leads a viscous damping matrix associated with the velocity that can be added for the most general matrix equations of motion

$$[M\,]\{\delta(\ddot{\tau})\,\} + [D\,]\{\delta(\dot{\tau})\,\} + [K\,]\{\delta\,(\tau)\} = \{c_P(\tau)\}$$

Simple Harmonic Motions (Vibrations)

In physics Simple Harmonic Motions (SHM) is the special case of free, $\{c_P(\tau)\} = \{0\}$, undamped, $[D\,] = [0\,]$, sinusoidal motion. The displacement solution in that case becomes a group of solutions (eigen-solution in German)

$$\{\delta(\tau)\,\} = \{\delta_k\}\,\mathrm{Sin}\,(\omega_k\,\tau)$$

where $\{\delta_k\}$ is the amplitude vector, ω_k is the frequency of vibration for mode k, and k varies from 1 to the number if degrees of freedom in the system, $1 \le k \le n_d$. The nodal velocities and nodal accelerations are

$$\{\delta(\dot{\tau})\} = \omega_k\,\{\delta_k\}\,\mathrm{Cos}\,(\omega_k\,\tau)$$

$$\{\delta(\ddot{\tau})\} = -\,\omega_k{}^2\{\delta_k\}\,\mathrm{Sin}\,(\,\tau) = -\,\omega_k{}^2\{\delta(\tau)\,\}$$

Then, the group equations become (for$1 \le k \le n_d$)

$$\left[[K\,] - \,\omega_k{}^2\,[M\,]\right]\{\delta_k\}\,\mathrm{Sin}\,(\omega_k\,\tau) = \,\{0\}$$

But a solution only exists, for ω_k and $\{\delta_k\}$, when the determinant vanishes

$$|[K\,] - \,\omega_k{}^2\,[M\,]| = 0$$

Once the eigenvalue, ω_k, is known then the corresponding mode shape, $\{\delta_k\}$, can be found. A zero natural frequency corresponds to a rigid body motion. There are at most six rigid body modes (three translations and three rotations). The k-th mode shape vector values are relative to each other and are usually normalized. Most codes normalize those data by dividing by the largest absolute value. Then the mode shape displacements vary from -1 to +1.

Obtaining the group solutions (eigen-solutions) is much more computationally intensive than solving a linear system with the same number of unknowns. The natural frequency, or eigenvalue, of the SHM is ω_k; and the mode shapes, or eigenvectors, are the $\{\delta_k\}$. Since the solution is computationally intense, only the first five or ten of the lowest or highest eigen-solutions are usually requested. The Matlab function $[[\mathrm{Modes}],[\mathrm{Sq\_diag}]] = \mathrm{eigs}([K],[M],k_{max},'sm')$ returns two square matrices of size $k_{max} \times k_{max}$ for the smallest $('sm')$ k_{max} natural frequencies. The mode shapes $\{\delta_k\}$ are in the k-th column of the $[\mathrm{Modes}]$ matrix.

The eigenvalue $\lambda_k = \omega_k{}^2$ is k-th diagonal entry in the diagonal square matrix [Sq_diag]. For mechanical vibrations ω_k is a real positive number, but round-off error can sometimes return a small complex component so it is best use the Matlab function real to extract just the real part. If a rigid body motion is found then its natural frequency is $\omega_k = 0$ (or a tiny negative value due to round-off errors).

When the vibration is forced, $\{c_P(\tau)\} \neq 0$, or if dampening is present, $[D] \neq [0]$, ,the full equations of motion must be integrated through time to obtain the time history of the motion.

Example 14.4-1 Given: A guitar string is needed for a musical instrument. The string has both ends fixed, a mass per unit length of ρ, and has a constant tension for of T. Determine how the tension of the string affects the first natural frequency of the string. **Solution**: The first symmetric mode is the fundamental one. Approximate it using a single quadratic line element, with both ends fixed. Then $L^e = L$ and the system (and element) stiffness and mass matrices are

$$\left(\frac{T}{3L}\begin{bmatrix} 7 & -8 & 1 \\ -8 & 16 & -8 \\ 1 & -8 & 7 \end{bmatrix} - \omega^2 \frac{\rho L}{30}\begin{bmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{bmatrix}\right)\begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix} = \begin{Bmatrix} -c(0)_{NBC} \\ 0 \\ c(L)_{NBC} \end{Bmatrix}$$

Enforcing the two EBCs means deleting rows one and three because only v_2 is the active displacement. Likewise, in the above sum columns one and three are multiplied by the zero Dirichlet boundary values leaving

$$\left(\frac{T}{3L}[0+16+0] - \omega^2 \frac{\rho L}{30}[0+16+0]\right)\{v_2\} = \{0\}$$

For a non-trivial solution, $v_2 \neq 0$, the determinant of the square matrix in the brackets must vanish. That happens only for specific values of $\omega_k, 1 \leq k \leq n_d$ that are equal in number to the number of free DOF in the mesh. Here, there is only one so

$$\omega_1^2 = 10\,T/\rho L^2, \quad \omega_1 = 3.1623\sqrt{T/\rho L^2}$$

The exact solution for the *k-th* mode is $\omega_k = k\,\pi\sqrt{T/\rho L^2}$, rad/sec where odd values of k are symmetric modes and even values are non-symmetric. The answer to the given question is that the frequency of the string vibration increases with the square root of its tension.

Here, the single quadratic element has only 0.66 % error in the first natural frequency. The exact first mode shape is a half-sine curve, the amplitude of which is normalized to unity. The half-sine mode shape is approximated spatially by a parabolic segment in a single element model.

ODE Equations of Motion

The same matrix equations of motion can be obtained by applying Galerkin's method to the ODE of motion. Consider the axial dynamics of an elastic bar

$$\frac{\partial}{\partial x}\left(EA\frac{\partial u}{\partial x}\right) + w(\tau) = \rho A\frac{\partial^2 u}{\partial \tau^2}$$

Multiply this relation by the solution, $u(x,\tau)$, and set that integral to zero

$$I = \int_L u(x,\tau)\left(\frac{\partial}{\partial x}\left(EA\frac{\partial u}{\partial x}\right) + w - \rho A\frac{\partial^2 u}{\partial \tau^2}\right)dx = 0$$

The last term is new, and the other matrices were formed in the prior static studies. The acceleration contribution is

$$I_m = \int_L u(x,\tau)\left(-\rho A \frac{\partial^2 u}{\partial \tau^2}\right) dx$$

Above, the interpolation with time changes were $u(x,\tau) = H(x)\, u^e(\tau)$, and

$$\frac{\partial^2 u}{\partial \tau^2} = \ddot{u} = \frac{\partial^2 u(x,\tau)}{\partial \tau^2} = \boldsymbol{H}(x)\frac{\partial^2 \boldsymbol{u^e}(\tau)}{\partial \tau^2} = \boldsymbol{H}(x)\ddot{\boldsymbol{u}}^{\boldsymbol{e}}$$

Thus, the new integral is

$$I_m^e = -\int_{L^e} \big(H(x)\, u^e(\tau)\big)^T \rho A \big(H(x)\, \ddot{u}^e(\tau)\big)\, dx$$

$$I_m^e = -u^e(\tau)^T \left[\int_{L^e} H(x)^T \rho A\, H(x)\, dx\right] \ddot{u}^e(\tau)$$

$$I_m^e = -u^e(\tau)^T [M]^e\, \ddot{u}^e(\tau)$$

This again gives rise to the mass matrix:

$$[M]^e = \left[\int_{L^e} H(x)^T \rho A\, H(x)\, dx\right]$$

Adding this new contribution to the static matrix system gives the matrix equation of motion as a second order ODE in time.

$$[M]^e \ddot{u}^e(\tau) + [K]^e u^e(\tau) = f_w^e(\tau)$$

Of course, the boundary conditions and initial conditions must be applied before this matrix ordinary differential in time can be integrated to yield a time history of the displacements, velocities, and accelerations.

Dynamic Time Histories

There are hundreds of algorithms for integrating the equations of motion. Finite element systems are typically very large and involve discontinuities in the forcing functions and/or the displacement boundary conditions. Thus, the most commonly used algorithms are one step methods. Finite difference methods in time are the oldest and most common, but newer methods using finite elements in time are also used.