# Dynamics

math Motivation: previous strong form: $\underbrace{AE\frac{d^2u}{dx^2}+bA=0}$

what's missing? any dependence on time

physical motivation: when do you need to account for time?

1. things are moving
2. things are not steady-state, or configuration changes over time
3. loading changes over time
4. material response is time (or rate) - dependent
   (rubbery materials)
5. wave propagation → occurs from shock loading or rapid changes in input
   → car crash
   → explosions

---

Equation Characteristics

1. We will solve a wave equation: 2nd derivative in time
   → information takes time to propagate from source
   → GIGO: garbage in, garbage out

Add inertia to previous strong form:

$$\underbrace{\frac{d}{dx}(A\sigma)+bA}_{\text{old equation}} = \underbrace{\rho A\frac{d^2u}{dt^2}}_{\text{instead of } \emptyset}$$

$\rho$ = density

$\frac{d^2u}{dt^2}$ = acceleration $\quad \frac{du}{dt}$ = velocity

weak form stays the same

this is new, now we'll derive:

L = length of bar

$$\int_0^L \delta u\, \rho A \frac{d^2u}{dt^2}\,dx$$

FEM Approxim: $u(x,t) = \sum_{I=1}^{\#\text{ nodes}} N_I(x)\, d_I(t)$

$\delta u(x) = \sum_{I=1}^{\#\text{ nodes}} N_I(x)\,\delta d_I$ ← weight function same as before

substitute FEM approx + weight into weak form:

$$\int_0^L (N\delta d)^T \rho A N \frac{d^2 d_I}{dt^2}\,dx$$

$$\delta d^T \int_0^L N^T \rho A N\,dx\, \frac{d^2 d_I}{dt^2}$$

get FEM equations of motion

$$\underbrace{\left[\int_0^L N^T N \rho A\,dx\right]\frac{d^2 d}{dt^2}}_{\text{kinetic, or inertial, force} \Rightarrow F^{kinetic}} = \underbrace{\int_0^L N^T bA\,dx + F\delta u(L)}_{F^{ext}} - \underbrace{\left[\int_0^L B^T AE B\,dx\right] d}_{F^{int}}$$

Analogy to Newton's equations of motion: $\underbrace{ma = F^{ext} - F^{int}}_{\text{"semi-discrete" momentum equation}}$

$$\underbrace{ma = F^{ext} - F^{int}}$$

"semi-discrete" momentum equation

→ discretized in space

→ not discretized in time: solve for accelerations, but need to integrate accelerations twice to get displacement

$\left.\begin{array}{l}\int a = v \\ \int v = d\end{array}\right\}$ will describe algorithm for this later

→ also called: An Initial Boundary Value Problem

$$d_I(t=0) = ?$$
$$\dot{d}_I = v_I(t=0) = ?$$

1D Mass Matrix:

$$\int_0^L \underline{N^T N \rho A dx} \rightarrow \text{units: } \frac{kg}{m^3} \cdot m^2 \cdot m = \underline{\underline{kg}}, \text{ why called mass matrix}$$

$\xi$

$$\int_{-1}^{1} N^T(\xi) N(\xi) \rho A \frac{dx}{d\xi} d\xi \qquad N(\xi) = \begin{bmatrix} \frac{1-\xi}{2} & \frac{1+\xi}{2} \end{bmatrix}$$

$$M = \rho A \int_{-1}^{1} \underbrace{\begin{bmatrix} \frac{1-\xi}{2} \\ \frac{1+\xi}{2} \end{bmatrix} \begin{bmatrix} \frac{1-\xi}{2} & \frac{1+\xi}{2} \end{bmatrix}} \frac{L^e}{2} d\xi$$

① integrand is quadratic (because 2 linear elements multiplied together)

→ need 2 quadrature points

② $N^T N$ → M will be symmetric

$$M = \frac{\rho A L^e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \rightarrow$$

① "consistent mass matrix"

② all elements > 0

→ assembly: for two elements $M = \frac{\rho A L^e}{6} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix}$ coupled here

→ downside: $Ma = f^{ext} - f^{int}$

M is not diagonal → requires a matrix inverse

→ so: ad hoc mass lumping

→ create diagonal matrix M: $M_{II} = \sum_J M_{IJ}$ (add up mass of each row, put answer at diagonal)

ex. 2-element case:

$$M = \frac{\rho A L^e}{6} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix} \Rightarrow \frac{\rho A L^e}{6} \begin{bmatrix} 3 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

$$\Rightarrow \text{simplify this} \Rightarrow \frac{\rho A L^e}{2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

what have we done?

① mass of each element = $\rho A L^e$

→ for 1 element $M = \frac{\rho A L^e}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ ... ½ mass went to left node, ½ the other node (split mass equally between nodes)

② mass is preserved

③ mass matrix is diagonal ... means all of the equations are uncoupled, so can solve each individually, don't need other equations to get current

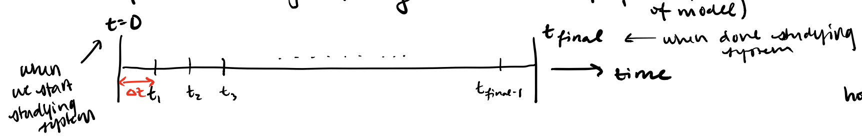
↳ we say: this allows parallel computing as equations are uncoupled

↳ linear (perfect) scaling for computational efficiency

↳ one downside to doing dynamics for very large systems

---

Time Steps + Integrating in Time (physical time of model) ... Dynamic Part of the Problem




how big of a time step Δt can we use in between pts we sample? time step determines how many sampling points!

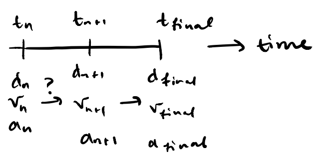
=> time discretization: only see FEM solution at discrete points in time

time step Δt: determines how many sampling points in time needed

Goals that we want to resolve:

① figure out time limits on Δt

② given $d_n, v_n, a_n$, how do you find $d_{n+1}, v_{n+1}, a_{n+1}$ (i.e. displacement, velocity, and acceleration at next time step)




partial answer to ①: limits on Δt

→ take momentum equation and ignore body force

$$\cancel{A}E\frac{d^2u}{dx^2} = \rho \cancel{A}\frac{d^2u}{dt^2}$$

acceleration: $\frac{d^2u}{dt^2} = \frac{E}{\rho}\frac{d^2u}{dx^2}$

Define variable $c$ = "wave speed" $= \sqrt{E/\rho}$ units of m/s (v)

→ Stiffness and density of material tells you how fast elastic waves will propagate in your structure

Can determine a "critical" time step $\Delta t_{crit} = \frac{L^e}{c}$ gives units of seconds

(like shaking bar in sinusoidal way, matlab ex.)

$\Delta t_{crit} = L^e\sqrt{\rho/E}$ → what does this mean? if use too large of time step, can lose information that need to capture

↑ estimate for what your critical time step is

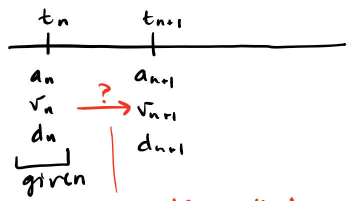
Time step must be small enough to allow a single FE to capture a characteristic wavelength passing through that element

→ if shrink element, time step must also get smaller

This critical time step will depend on type of mass matrix used (consistent vs. lumped)

---

Time Integration: goal is to systematically update $d, v, a$ for each time step

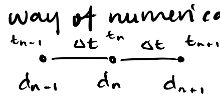



how do we find value at next time step

Use a simple algorithm called the "Newmark Predictor-Corrector Integrator" (also called "Explicit Central-Difference")

Explicit integrator: $a_{n+1} = f(a_n, v_n, t_n)$ *preferable*

Central-Difference: one way of numerically calculating a derivative




$$v_n = \frac{d_{n+1} - d_{n-1}}{2\Delta t}$$

Algorithm: given $d_n, v_n, a_n$

$$m a_{n+1} = f^{ext}_{n+1} - f^{int}_{n+1}$$

$$= f^{ext}_{n+1} - k d_{n+1}$$

define $d_{n+1} = d_n + \Delta t v_n + \frac{\Delta t^2}{2}\left((1-2\beta)a_n + 2\beta a_{n+1}\right)$

define $v_{n+1} = v_n + \Delta t\left((1-\gamma)a_n + \gamma a_{n+1}\right)$

→ for explicit central difference, $\beta = 0, \gamma = \frac{1}{2}$

so:

$$d_{n+1} = d_n + \Delta t v_n + \frac{\Delta t^2}{2} a_n$$

→ completely explicit, meaning, if know velocity and acceleration @ previous step, then know next displacement

$$v_{n+1} = v_n + \frac{\Delta t}{2}(a_n + a_{n+1})$$

→ depends on $a_{n+1}$, the next step? We'll need to rewrite to fix this problem

→ Integrator for $d_{n+1}$ looks like a Taylor Series in terms of $\Delta t$, truncated after terms of $\Delta t^2$ ($\Delta t^2$ is largest expansion term we have). So it's called <u>2nd Order Accurate</u>

- because truncating expansion, leaving some error (2nd order is pretty cheap and still pretty accurate)

Define Predictors:

$\tilde{d}_{n+1} = d_n + \Delta t v_n + \frac{\Delta t^2}{2} a_n$ (displacement predictor) $= d_{n+1}$

$\tilde{v}_{n+1} = v_n + \frac{\Delta t}{2} a_n$ (velocity predictor)

correction step: (corrector) $d_{n+1} = \tilde{d}_{n+1}$ ; $v_{n+1} = \tilde{v}_{n+1} + \frac{1}{2}\Delta t a_{n+1}$

## Write algorithm

- given $d_0, v_0$ (known from initial conditions)
  1. get $f_0^{ext}$ (at time $t=0$)
  2. then calculate initial acceleration $a_0$ at $t=0 \Rightarrow a_0 = M^{-1}(f_0^{ext} - Kd_0)$ — inverse of mass matrix

Loop for $i=1:\#$ of time steps

3. partial velocity update $\tilde{v}_{n+1} = v_n + \frac{\Delta t}{2} a_n$
4. enforce boundary conditions on specified nodes, i.e. set $v=0$ for specific nodes, then integrate to get $d=0$ for same nodes
5. update displacements: $d_{n+1} = \tilde{d}_{n+1} = d_n + \Delta t v_n + \frac{\Delta t^2}{2} a_n$
   $= d_n + \Delta t\, \tilde{v}_{n+1}$
6. get internal force: $f_{n+1}^{int} = Kd_{n+1}$
7. update acceleration: $a_{n+1} = M^{-1}(f_{n+1}^{ext} - f_{n+1}^{int})$
   what was just calculated in 6
8. find velocity update: $v_{n+1} = \tilde{v}_{n+1} + \frac{\Delta t}{2} a_{n+1}$

*computer program will do everything for you, just need to define boundary conditions really*

---

Stable timestep for consistent vs. lumped mass matrices
→ explicit vs. implicit time integration

Time Integration: how you update

$$\begin{matrix} d_n & & d_{n+1} \\ v_n & \longrightarrow & v_{n+1} \\ a_n & & a_{n+1} \end{matrix}$$

$$d^{n+1} = d^n + \Delta t v^n + \frac{\Delta t^2}{2}\left[(1-2\beta)a^n + 2\beta(a^{n+1})\right]$$

if $\beta = 0$, $d^{n+1} = d^n + \Delta t v^n + \frac{\Delta t^2}{2} a^n$

key: $d^{n+1} = f(d^n, v^n, a^n)$, i.e. only depends on previous time values
"EXPLICIT"

if $\beta \neq 0$, $d^{n+1} = f(d^n, v^n, a^n, a^{n+1})$, because depends on current time properties ($a^{n+1}$)
"IMPLICIT"

| EXPLICIT | IMPLICIT |
|---|---|
| 1. conditionally stable → i.e. limit on how large $\Delta t$ can be<br>2. need many timesteps, but quick because M (mass matrix) can be diagonal<br><br>*what we'll use in this class* | 1. unconditionally stable → can use larger $\Delta t$<br>2. expensive → can't diagonalize M |

For explicit time integration, what is limit on $\Delta t$?

$$M\ddot{d} + Kd = 0 \qquad \ddot{d} = \frac{d^2 d}{dt^2}, \quad f^{ext} = 0,$$ so free vibration problem

$$\ddot{d} + \omega^2 d = 0 \qquad \omega = \sqrt{\frac{K}{m}}$$

Take central difference for second derivative

$$\ddot{d} = \frac{1}{\Delta t^2}\left(d_{n+1} - 2d_n + d_{n-1}\right)$$

$$\frac{1}{\Delta t^2}\left(d_{n+1} - 2d_n + d_{n-1}\right) + \omega^2 d_n = 0$$

$$d_{n+1} + d_n(\omega^2\Delta t^2 - 2) + d_{n-1} = 0$$

Seek solutions of the form $d_n = \lambda^n \rightarrow$ So $d_{n+1} = \lambda^{n+1}$, $d_{n-1} = \lambda^{n-1}$

$\rightarrow \quad \lambda^{n+1} + \lambda^n(\omega^2\Delta t^2 - 2) + \lambda^{n-1} = 0$

$\rightarrow \quad \dfrac{\lambda^{n+1} + \lambda^n(\omega^2\Delta t^2 - 2) + \lambda^{n-1} = 0}{\lambda^{n-1}}$ (divide by $\lambda^{n-1}$)

$\rightarrow \quad \lambda^2 + \lambda(\omega^2\Delta t^2 - 2) + 1 = 0$ (quadratic, solve for roots)

$\rightarrow \quad \lambda = \dfrac{2 - \omega^2\Delta t^2 \pm \sqrt{(\omega^2\Delta t^2 - 2)^2 - 4}}{2}$

$\rightarrow \quad \begin{cases} \lambda_1 = 2 - \omega^2\Delta t^2 + \sqrt{(\omega^2\Delta t^2 - 2)^2 - 4} \\ \lambda_2 = 2 - \omega^2\Delta t^2 - \sqrt{(\omega^2\Delta t^2 - 2)^2 - 4} \end{cases}$

$\rightarrow$ assume $\lambda_1 \neq \lambda_2$, ie distinct roots

Then: $d_n = c_1\lambda_1^n + c_2\lambda_2^n$ where $c_1, c_2$ = constants

$\rightarrow$ What does this mean?

- if $|\lambda_1|$ and $|\lambda_2| \leq 1$, $d_n$ (the displacement soln) decays or remains stable with time, i.e. solution is <u>stable</u>
- if $|\lambda_1|$ and $|\lambda_2| > 1$, $d_n$ grows unbounded, so solution is <u>unstable</u>

$a\lambda^2 + b\lambda + c = 0 \rightarrow \lambda_1, \lambda_2 = \frac{c}{a} = 1$ for this equation

$$\sqrt{(\omega^2\Delta t^2 - 2)^2 - 4} = \omega\Delta t\sqrt{\omega^2\Delta t^2 - 4}$$

if $\sqrt{\omega^2\Delta t^2 - 4} \geq 0$, $\lambda_1, \lambda_2 = 1$, but because have real, distinct roots, then either $|\lambda_1| > 1$ or $|\lambda_2| > 1$, making this solution <u>unstable</u>, which is not what we want!

$\Rightarrow$ so we want $\sqrt{\omega^2\Delta t^2 - 4} < 0$ – or – $\omega^2\Delta t^2 - 4 < 0$

so then $\boxed{\Delta t < \frac{2}{\omega}}$ $\rightarrow$ will soon turn this into a more useful result

Let's connect this to FE now:

→ natural frequencies and modes

$M\ddot{d} + kd = 0$

assume $d = \bar{d} \sin \omega t$ ... sinusoidal frequency ... $\omega$ = frequency, $\bar{d}$ = amplitude of vibration

$-M\omega^2 \bar{d} \sin \omega t + k\bar{d} \sin \omega t = 0$

$(k - \omega^2 M)\bar{d} \sin \omega t = 0$

$(k - \omega^2 M)\bar{d} = 0$

$$\boxed{(k - \omega^2 M)\bar{d} = 0}$$

$\omega$ = eigenvalue → represents natural frequency of vibration

$\bar{d}$ = eigenvector → gives mode shape of vibration

Questions to think about:

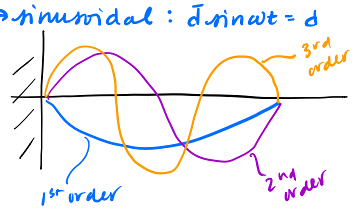
① How many natural frequencies of vibration are there?

→ infinite (in reality)

→ FEM: # of degrees of freedom (ie # nodes in 1D)

② What do vibrational modes look like?

→ sinusoidal : $\bar{d} \sin \omega t = d$




← all sine waves of different form

→ what about $\omega = 0$? What does $\omega = 0$ mean?

$\omega = 0 \longrightarrow \bar{d} = 0$ (eigenvector is zero) → means beam or bar is not displacing at all → means taking zero energy

So, $\omega = 0$ is a zero-energy mode for not properly applying boundary conditions

---

Stable timestep for different mass matrices


consistent mass matrix for 1 element: $\frac{\rho AL}{6} = \frac{M}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ bc $M = \rho AL$

lumped mass matrix : $\frac{M}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\quad k = \frac{AE}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$

→ for simplicity, say $A = E = L = M = 1$

→ consistent mass:

first eigenvalue, $\omega_1 = 0$ → rigid body mode (if don't constrain element)

second eigenvalue, $\omega_2 = 3.464\sqrt{\frac{AE}{mL}} = 3.464$

EXACT SOLN: $\pi\sqrt{\frac{AE}{mL}}$

→ lumped mass:

$\omega_1 = 0$ → rigid body

$\omega_2 = 2\sqrt{\frac{AE}{mL}} = 2$

THIS TELLS US:

1. consistent mass tends to overpredict natural frequencies
2. lumped mass tends to underpredict natural frequencies

→ Both will converge to exact with mesh refinement

consistent : $\omega_{max} = \omega_2 = 3.464\sqrt{\frac{AE}{mL}} = 3.464\sqrt{\frac{AE}{(\rho A L)L}} = \frac{3.464}{L}\sqrt{\frac{E}{\rho}} = 3.464\frac{c}{L}$

$\sqrt{\frac{E}{\rho}}$ ↑ wavespeed, $c$

lumped mass : $\omega_{max} = \omega_2 = 2\sqrt{\frac{AE}{mL}} = 2\frac{c}{L}$

from before...

consistent : $\Delta t < \frac{2}{\omega} = \frac{2}{3.464\frac{c}{L}} \Rightarrow \boxed{\Delta t < \sim 0.58\frac{L}{c}}$

lumped : $\Delta t < \frac{2}{\omega} = \frac{2}{2\frac{c}{L}} \Rightarrow \boxed{\Delta t < \frac{L}{c}}$

where $L$ is length of element, $c$ is the wave speed

Important Implications:

1. need ~40% fewer timesteps with lumped mass for same physical time to model
2. Furthermore: cheaper to invert lumped mass each timestamp

↳ * so lumped mass matrix almost always used in explicit dynamics

Physically, why is $\Delta t$ larger for lumped mass?

$\omega = \sqrt{\frac{k}{m}}$, $\Delta t < \frac{2}{\omega}$

Is effective m larger for lumped or consistent?

consis: $\frac{M}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ lump: $\frac{M}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

↑ mass isn't concentrated at the nodes

↑ all mass at nodes

so, $\omega$ larger for consistent, and $\Delta t$ is smaller

*rewatch for MATLAB demo*

→ dynamics scripts