2.1 Equilibrium

- Framework of linear equilibrium
  - (1) three steps:

$$u \xrightarrow{A} e \xrightarrow{C} w \xrightarrow{A^T} f$$

$$\underbrace{f}_{\substack{\text{input}\\(\text{force})}} = A^T w = A^T C e = \underbrace{A^T C A}_{\substack{K\\(\text{stiffness})}} \underbrace{u}_{\substack{\text{output}\\(\text{displacement})}}$$

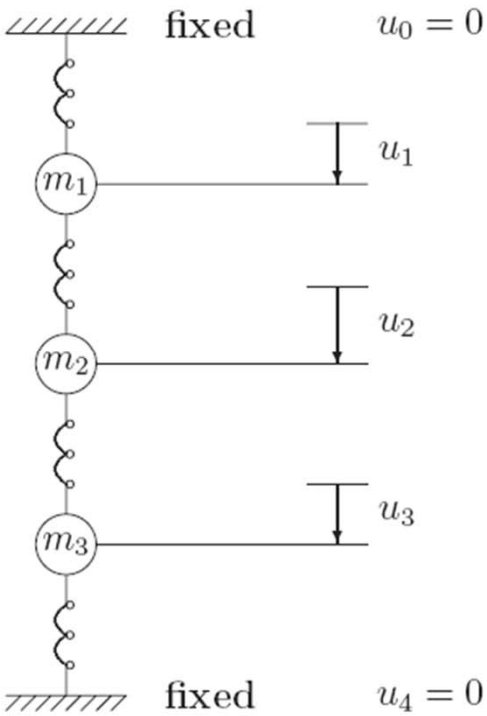
  - Modeling problem: to find $K = A^T C A$
  - Computing problem: to solve $Ku = f$
  - (2) minimum principles: minimize the total potential energy
- Extension
  - Nonlinear problem: when input becomes large
  - Time-dependent problem: linear but dynamic

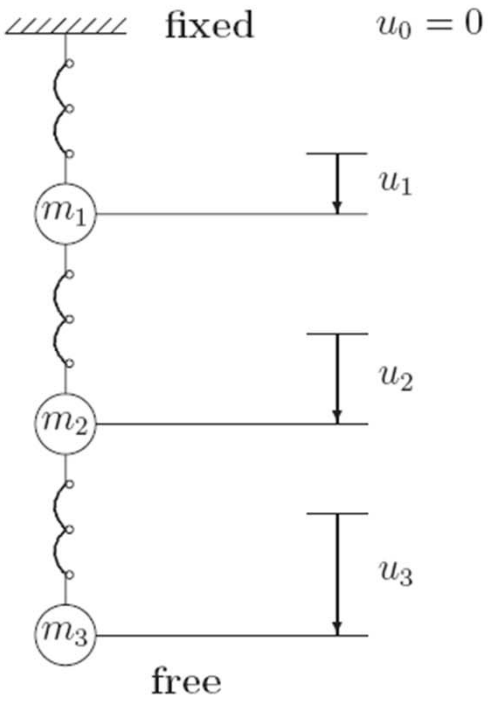
Examples

- Steady state matrix equation: Ku =f
  - A line of springs: the spring can be stretched or compressed
  - Best solution of Au = b: least squares regression in statistics
  - A network with flows on the edges: driven by potential differences
- Dynamic problems: differential equation in time
  - The springs are oscillating
  - The circuit has inductors and capacitors and alternating current
  - New measurements require us to update the statistics

$$\frac{du}{dt} = Ku - f \quad \text{or} \quad M\frac{d^2u}{dt^2} + Ku = f(t)$$

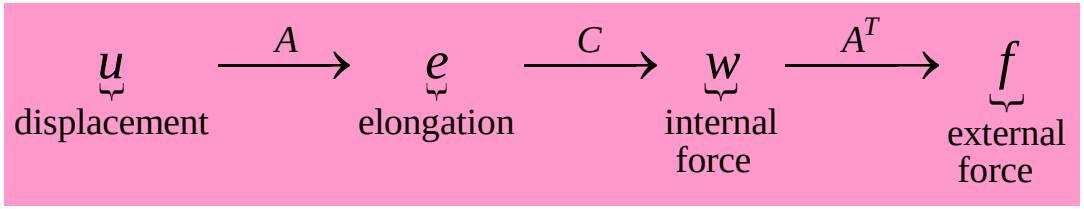
A Line of Springs (1)





	fixed-fixed	fixed-free	free-free
A	4 x 3	3 x 3	2 x 3
A^TCA (C=I)	K_3	H_3	B_3

A Line of Springs (2)



	spring	elasticity
Step 1	Difference in displacements	Kinematic equation
Step 2	Hooke's law	Constitutive law
Step 3	Balance equation	Static equation for equilibrium

- Properties of $K = A^T C A$
  - Tridiagonal
  - Symmetric
  - Positive definite
  - Full matrix of K^{-1} with all positive entries

when $C = I$,

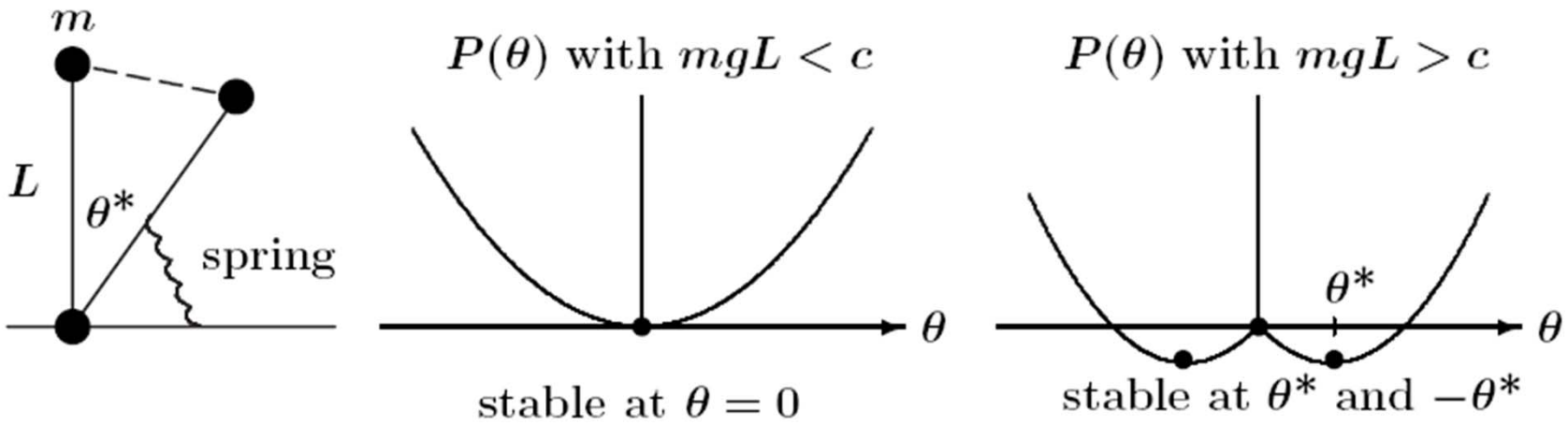
$$K \underbrace{= A^T A}_{\substack{\text{assemble} \\ \text{applied matheatics}}} \underbrace{= LL^T}_{\substack{\text{break up} \\ \text{computational mathematics}}}$$

Minimum Principles

$$P(u) = \frac{1}{2}u^T K u - u^T f = \frac{1}{2}\left(u - K^{-1}f\right)^T K\left(u - K^{-1}f\right) - \frac{1}{2}f^T K^{-1} f$$

Minimize $P(u)$ by solving $\dfrac{dP}{du} = 0$ or $\nabla P = 0$ or $\dfrac{\delta P}{\delta u} = 0$

$$P_{\min} = -\frac{1}{2}f^T K^{-1} f = -\frac{1}{2}f^T u$$



Stability? Second derivative global minimum local minimum

2.2 Oscillation by Newton's Law

$$F = ma \Leftrightarrow f - Ku = Mu_{tt} \rightarrow Mu_{tt} + Ku = f$$

$$f = 0 : u_{tt} = M^{-1}Ku \text{ (eigenvalue problem)}$$

- Finite differences with time steps Δt
  - Explicit leapfrog method: short fast steps
  - Implicit trapezoidal rule: larger slow steps
- Explicit finite differences
  - Stability
- Implicit trapezoidal rule: (new+old)/2
  - First order
  - Second order

Key Example (1)

$$u'' + u = 0 \rightarrow \left\{ \begin{matrix} u = \cos t \\ v = u' = -\sin t \end{matrix} \right\} \rightarrow u^2 + v^2 = 1$$

Exact solution travels around constant energy circle

$$\xrightarrow{\text{finite difference}} \underbrace{\left[\begin{matrix} u' = \dfrac{u_{n+1} - u_n}{h} = v \\ v' = \dfrac{v_{n+1} - v_n}{h} = -u \end{matrix} \right]}_{\text{where to evaluate the right sides?}}$$

- Forward Euler

$$\left. \begin{matrix} u' = \dfrac{u_{n+1} - u_n}{h} = v_n \\ v' = \dfrac{v_{n+1} - v_n}{h} = -u_n \end{matrix} \right\} \rightarrow \begin{bmatrix} u_{n+1} \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & h \\ -h & 1 \end{bmatrix} \begin{bmatrix} u_n \\ v_n \end{bmatrix} = G_F \begin{bmatrix} u_n \\ v_n \end{bmatrix} \rightarrow |\lambda| > 1$$

- Backward Euler

$$\left. \begin{matrix} u' = \dfrac{u_{n+1} - u_n}{h} = v_{n+1} \\ v' = \dfrac{v_{n+1} - v_n}{h} = -u_{n+1} \end{matrix} \right\} \rightarrow \begin{bmatrix} 1 & -h \\ h & 1 \end{bmatrix} \begin{bmatrix} u_{n+1} \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} u_n \\ v_n \end{bmatrix} \rightarrow \begin{bmatrix} u_{n+1} \\ v_{n+1} \end{bmatrix} = \underbrace{\frac{1}{1+h^2} \begin{bmatrix} 1 & h \\ -h & 1 \end{bmatrix}}_{G_B} \begin{bmatrix} u_n \\ v_n \end{bmatrix} \rightarrow |\lambda| < 1$$

Key Example (2)

- Trapezoidal method: centered at n+1/2

$$\left.\begin{aligned} u' &= \frac{u_{n+1}-u_n}{h} = \frac{v_n+v_{n+1}}{2} \\ v' &= \frac{v_{n+1}-v_n}{h} = -\frac{u_n+u_{n+1}}{2} \end{aligned}\right\} \rightarrow \begin{bmatrix} 1 & -h/2 \\ h/2 & 1 \end{bmatrix}\begin{bmatrix} u_{n+1} \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & h/2 \\ -h/2 & 1 \end{bmatrix}\begin{bmatrix} u_n \\ v_n \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} u_{n+1} \\ v_{n+1} \end{bmatrix} = \underbrace{\frac{1}{1+(h/2)^2}\begin{bmatrix} 1+(h/2)^2 & h \\ -h & 1-(h/2)^2 \end{bmatrix}}_{G_T}\begin{bmatrix} u_n \\ v_n \end{bmatrix} \rightarrow |\lambda| = 1$$

- Leapfrog method: explicit

$$\left.\begin{aligned} u' &= \frac{u_{n+1}-u_n}{h} = v_n \\ v' &= \frac{v_{n+1}-v_n}{h} = -u_{n+1} = -(u_n + hv_n) \end{aligned}\right\} \rightarrow \begin{bmatrix} u_{n+1} \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & h \\ -h & 1-h^2 \end{bmatrix}\begin{bmatrix} u_n \\ v_n \end{bmatrix} = G_L\begin{bmatrix} u_n \\ v_n \end{bmatrix} \rightarrow |\lambda| = 1 \text{ if } h \le 2$$

Explicit Finite Differences

$$Mu'' + Ku = f \xrightarrow{\text{Leapfrog with } \frac{\Delta^2 u}{(\Delta t)^2}} M\frac{u_{n+1} - 2u_n + u_{n-1}}{(\Delta t)^2} + Ku_n = f_n$$

$$Mu_{n+1} = \left[2 - (\Delta t)^2 K\right]u_n - Mu_{n-1} + (\Delta t)^2 f_n : \text{explicit}$$

$$u_0 = u(0), u'(0) = \frac{u_1 - u_0}{\Delta t} \rightarrow u_1 = u_0 + (\Delta t)u'(0), u_2 = \cdots$$

$$Mu'' + Ku = f \xrightarrow{\text{first order}} \begin{cases} Mv' + Ku = f \\ u' = v \end{cases} \rightarrow \begin{cases} M\left(\frac{v_{n+1/2} - v_{n-1/2}}{\Delta t}\right) + Ku_n = f_n \\ \frac{u_{n+1} - u_n}{\Delta t} = v_{n+1/2} \end{cases}$$

$$\left.\begin{aligned} \frac{u_{n+1} - u_n}{\Delta t} &= v_{n+1/2} \\ \frac{u_n - u_{n-1}}{\Delta t} &= v_{n-1/2} \end{aligned}\right\} \rightarrow u_{n+1} - 2u_n + u_{n-1} = \Delta t\left(v_{n+1/2} - v_{n-1/2}\right)$$

Trapezoidal Rule for Second Order Equation

$$Mu'' + Ku = 0 \xrightarrow{\text{first order}} \begin{cases} Mv' + Ku = f \\ u' = v \end{cases} \rightarrow \begin{cases} M\left(\frac{v_{n+1} - v_n}{\Delta t}\right) + K\left(\frac{u_{n+1} + u_n}{2}\right) = 0 \\ \frac{u_{n+1} - u_n}{\Delta t} = \frac{v_{n+1} + v_n}{2} \end{cases}$$

$$\left.\begin{aligned} v_{n+1} - v_n &= -(\Delta t) M^{-1} K \left(\frac{u_{n+1} + u_n}{2}\right) \\ u_{n+1} - u_n &= (\Delta t)\left(\frac{v_{n+1} + v_n}{2}\right) \end{aligned}\right\} \rightarrow \begin{bmatrix} I & \frac{(\Delta t) M^{-1} K}{2} \\ -\frac{(\Delta t) I}{2} & I \end{bmatrix} \begin{bmatrix} v_{n+1} \\ u_{n+1} \end{bmatrix} = \begin{bmatrix} I & -\frac{(\Delta t) M^{-1} K}{2} \\ \frac{(\Delta t) I}{2} & I \end{bmatrix} \begin{bmatrix} v_n \\ u_n \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} I + \frac{(\Delta t)^2 M^{-1} K}{4} & 0 \\ 0 & I + \frac{(\Delta t)^2 M^{-1} K}{4} \end{bmatrix} \begin{bmatrix} v_{n+1} \\ u_{n+1} \end{bmatrix} = \begin{bmatrix} I & -\frac{(\Delta t) M^{-1} K}{2} \\ \frac{(\Delta t) I}{2} & I \end{bmatrix}^2 \begin{bmatrix} v_n \\ u_n \end{bmatrix}$$

2.3 Least Squares

$$Au = b \xrightarrow[\text{overdetermined}]{A\text{: rectangular } (m>n)} \begin{cases} \text{no solution} \\ \hat{u} \text{ (best solution)} \end{cases}$$

- Principle of least squares
  - Minimizing the total squared error

$$\text{Minimize } E = \|e\|^2 = \|b - Au\|^2 = (b - Au)^T (b - Au)$$

$$\begin{cases} A \text{ has independent columns} \rightarrow A^T A\hat{u} = A^T b \text{ (normal equation)} \\ A \text{ has dependent columns} \rightarrow QR \text{ factorization} \end{cases}$$

  - Projection of b onto the columns of A

$$A\hat{u} = p \perp e = b - p$$

Overdetermined system (m > n)	Underdetermined system (m << n)
More equations than unknowns	Fewer samples than unknowns
No solution	Infinitely many solutions
Linear regression	Gene expression analysis Bioinformatics Sparse sensing/compression
L^2 norm: small is good	L^1 norm: a lot of zeros with a few nonzeros in the right position

- Least squares by calculus: minimum
- Least squares by linear algebra: projection
  - Aû is the point in the plane that is nearest to b
- Computational least squares: how to compute û?
  - A^TA: stability?
  - A = QR
    - Gram-Schmidt
    - Householder
- Weighted least squares

2.4 Graph Model

- Graph
  - *n* nodes connected (or not) by *m* edges
  - Tree: no closed loop (m = n–1)
- Incidence matrix: A (m x n)
  - Record of all connections in the graph: topology matrix
  - Act as a difference matrix: Au (potential difference)
    - n node vector u → m edge vector Au
- Graph Laplacian: A^TA = (diagonal) – (off-diagonal)
- Weighted Laplacian: A^TCA = (node weight) – (edge weight)
- Framework
  - Three equations
  - Two equations
  - One equation

Framework of Equilibrium

	mechanics	statistics	networks
u	displacement	best parameters	voltage at nodes
e=(b–)Au	elongations	errors	voltage drop
w=Ce	Hooke's law	weight	Ohm's law
$f=A^Tw$	force balance	Projection (f=0)	Kirchhoff
	$A^TCAu=f$	$A^TCA\hat{u}=A^TCb$	$A^TCAu=A^TCb-f$
source	external forces f	observations b	current f, voltage b

2.6 Nonlinear Problems

$$\left.\begin{aligned} &g_1(u_1,\ldots,u_n)=0\\ &\vdots\\ &g_n(u_1,\ldots,u_n)=0 \end{aligned}\right\} \rightarrow J_{ij}=\left[\frac{\partial g_i}{\partial u_j}\right] \xrightarrow[\text{approximation}]{\text{linear}} g(u)\approx g\left(u^0\right)+J\left(u^0\right)\left(u-u^0\right)$$

$$g\left(u^{k+1}\right)\approx g\left(u^k\right)+J\left(u^k\right)\left(u^{k+1}-u^k\right)=0$$

	iteration	Jacobian	convergence
Newton's method	$J\left(u^k\right)\left(u^{k+1}-u^k\right)=-g\left(u^k\right)$	$J^k = J(u^k)$	quadratic
Modified Newton's method → fixed point iteration	$J\left(u^0\right)\left(u^{k+1}-u^k\right)=-g\left(u^k\right)$	$J^0 = J(u^0)$	linear

$$J\left(u^0\right)\left(u^{k+1}-u^k\right)=-g\left(u^k\right)\rightarrow u^{k+1}=u^k-J\left(u^0\right)^{-1}g\left(u^k\right)=H\left(u^k\right)$$

$$\xrightarrow{u^k\rightarrow u^*} u^*=H\left(u^*\right):\ \text{fixed point}$$

$$\text{error: } u^*-u^{k+1}=H\left(u^*\right)-H\left(u^k\right)\approx H'\left(u^k\right)\left(u^*-u^k\right)\rightarrow e^{k+1}\approx ce^k$$

Variations on Newton's Method

- Damped Newton: $\alpha \Delta u^k$
- Continuation methods: homotopy method
- Inexact Newton
  - Inner iteration for each Δu^k stops early, Krylov subspaces
- Nonlinear conjugate gradients: direction, stepsize

$$\begin{cases} d^k = -g\left(u^k\right) + \beta d^{k-1} \quad \text{where} \quad \beta = \dfrac{\left(g^k\right)^T \left(g^k - g^{k-1}\right)}{\left(g^{k-1}\right)^T g^{k-1}} \\ u^{k+1} = u^k + \alpha d^k \end{cases}$$

- Quasi-Newton: approximate J

$$\left(J^{k-1} + \text{update}\right)(\Delta u) = g\left(u^k\right) - g\left(u^{k-1}\right)$$

2.7 Structures in Equilibrium

- Truss: *m* elastic bars, *N* pin joints
  - Do not bend (vs. beam): turn freely
  - One-dimensional (vs. plate)
  - Simple and straight (vs. shell)
  - Unknown displacement in a plane: $n = 2N - r$

	Stable	unstable
n columns of A	Independent rank(A) = n	Dependent rank(A) < n
solution to e = Au = 0	u = 0 (only solution) disp→stretching	Nonzero solution disp w/o stretching
$A^Tw = f$ (force balance)	Solve for every f	Not solvable

- Two types of unstable trusses: singular matrix (A^TCA)
  - Rigid motion: translate, rotate
  - Mechanism: change of shape w/o any stretching, small displacements with first order
- *m* > *n* does not guarantee *n* independent columns of A and stability
- Construction of A
  - Row: stretching of a bar
  - Column: force balance at the nodes
- $K = A^TCA$

$$K = \sum_{i=1}^{m} k_i$$

  - K_{ij} = (row *i* of A^T)(column *j* of CA)
  - Stiffness matrix for bar *i*: k_i = (column *i* of A^T)c_i(row *i* of A)