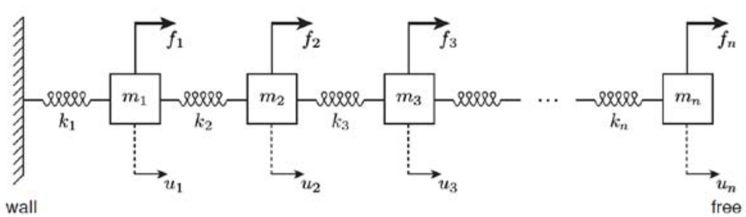
Sparsity

Origin

system of n springs :



Few connections :

mass i connected only to springs $i, i+1$;
spring i connected only to masses $i-1, i$.

row perspective on sparsity :

$$\sum \text{forces on mass } 1 = 0$$

$$\Rightarrow f_1 - k_1 u_1 + k_2(u_2 - u_1) = 0 \; ,$$

$$\sum \text{forces on mass } 2 = 0$$

$$\Rightarrow f_2 - k_2(u_2 - u_1) + k_3(u_3 - u_2) = 0 \; ,$$

$$\sum \text{forces on mass } i = 0 \;\; (i \neq 1, \; i \neq n)$$

force due to spring i

$$\Rightarrow f_i \underbrace{- k_i(u_i - u_{i-1}) + k_{i+1}(u_{i+1} - u_i)}_{\text{force due to springs on mass } i \text{ depends only on } u_{i-1},\, u_i, \text{ and } u_{i+1}} = 0 \; ,$$

force due to spring $i-1$

$$\sum \text{forces on mass } n = 0$$

$$\Rightarrow f_n - k_n(u_n - u_{n-1}) = 0 \; .$$

or column perspective

$$\underset{n \times n}{\underset{K}{\begin{pmatrix} k_1+k_2 & -k_2 & & & & \\ -k_2 & k_2+k_3 & -k_3 & & 0 & \\ & -k_3 & k_3+k_4 & -k_4 & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & & & -k_n \\ & & & & -k_n & k_n \end{pmatrix}}} \underset{n \times 1}{\underset{u}{\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix}}} = \underset{n \times 1}{\underset{f}{\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-1} \\ f_n \end{pmatrix}}}$$

Symmetric

Positive-Definite

Tri-diagonal matrix:

each row has at most 3 non-zero entries; sparsity*

non-zero entries on main diag or main ± 1 diags.**

* retained in higher dimensions

** NOT retained in higher dimensions

Storage

A: n×n sparse matrix

Let

nnz(A) ≡ # (non-zero entries of A) say $O(n)$

$\ll n^2$ if A is sparse.

Introduce $I(\cdot)$, $J(\cdot)$, $V_A(\cdot)$:

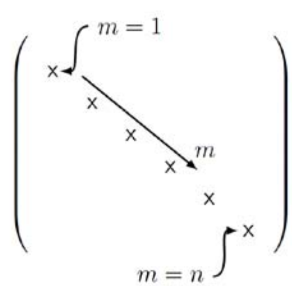
nnz × 1

$I(m), J(m),\ 1 \le m \le \mathrm{nnz}(A)$:
indices $i = I(m),\ j = J(m)$
for which $A_{ij} \neq 0$

$V_A(m),\ 1 \le m \le \mathrm{nnz}(A)$:
value of $A_{I(m),J(m)}$

completely specifies A

Example: A ≡ Identity Matrix



$$\left.\begin{array}{llll} m=1: & I(1)=1,\ J(1)=1 & V_A(1)=1 \\ m=2: & I(2)=2,\ J(2)=2 & V_A(2)=1 \\ & \vdots & \\ m=n: & I(n)=1,\ J(n)=n & V_A(n)=1 \end{array}\right\}$$

nnz(A) = $n \ll n^2$

Exercise: A tri-diagonal — $n=5$; $k_i = 1$, $1 \le i \le n$.

MATLAB: B from string in tension — spy, nnz; spalloc, issparse DEMO

B: tri-diagonal; spy, nnz: mathematical; spalloc, issparse: "declared"

Matrix-Vector Product

$$w = Av$$

A : $n \times n$ sparse matrix   given

v : $n \times 1$ given

w : $n \times 1$

Definition:

$$w_i = \sum_{j=1}^{n} A_{ij} v_j \,, \quad 1 \le i \le n. \qquad O(n^2) \text{ FLOPs}$$

Sparse equivalent: $O(\mathrm{nnz}(A))$ FLOPs

$w = \texttt{zeros}(n, 1)$

for $m = 1{:}\ \texttt{nnz}(A)$

$$w(I(m)) = w(I(m)) + \underbrace{\underset{A_{I(m),J(m)}}{V_A(m)} \times v(J(m))}_{\text{all non-zero contributions to } \sum_j A_{ij} v_j}$$

end

A tri-diagonal: $2 \cdot (3n - 2)$ FLOPs

MATLAB: w = A*v IF issparse(A) = true; otherwise...

Gaussian Elimination and Back Substitution

$Au = f$ f given

A : tri-diagonal $n \times n$ matrix

MATLAB: u = A\f IF issparse(A) = true; otherwise...

matrix:

$$A = \begin{pmatrix} \times & \times & & & & & & \\ \times & \times & \times & & & 0 & & \\ & \times & \times & \times & & & & \\ & & \times & \times & \times & & & \\ & & & \times & \times & \times & & \\ & 0 & & & \times & \times & \times & \\ & & & & & \times & \times & \times \\ & & & & & & \times & \times \end{pmatrix}$$

main + 1 diagonal

main − 1 diagonal

main diagonal

Gaussian Elimination

$\boxed{\text{visually}}$ DEMO

to form $\tilde{U}(k=1)$, $\hat{\hat{f}}(k=1)$: 3 + 2 FLOPs

AND $\tilde{U}(k=1)$ preserves all zeros of upper (A)

to form $\tilde{U}(k=2)$, $\hat{\hat{f}}(k=1)$: 3 + 2 FLOPs

AND $\tilde{U}(k=2)$ preserves all zeros of upper (A)

$\vdots$

} n rows

---

to form $U, \hat{f}$: $\boxed{5n}$ FLOPs vs dense $O(n^3)$

AND U preserves all zeros of upper (A)

Back Substitution

$$\underbrace{\begin{pmatrix} \times & \times & & & & & & \\ & \times & \times & & & 0 & & \\ & & \times & \times & & & & \\ & & & \times & \times & & & \\ & & & & \times & \times & & \\ & 0 & & & & \times & \times & \\ & & & & & & \times & \times \\ & & & & & & & \times \end{pmatrix}}_{U} \underbrace{\begin{pmatrix} u_1 \\ \text{3 FLOPs} \\ u_2 \\ \vdots \\ u_{n-2} \\ \text{3 FLOPs} \\ u_{n-1} \\ \text{3 FLOPs} \\ u_n \\ \text{1 FLOP} \end{pmatrix}}_{u} \underset{=}{=} \underbrace{\begin{pmatrix} \hat{f}_1 \\ \hat{f}_1 \\ \vdots \\ \hat{f}_{n-2} \\ \hat{f}_{n-1} \\ \hat{f}_n \end{pmatrix}}_{\hat{f}}$$

to find u: $\boxed{3n}$ FLOPs vs dense $O(n^2)$

DEMO - B

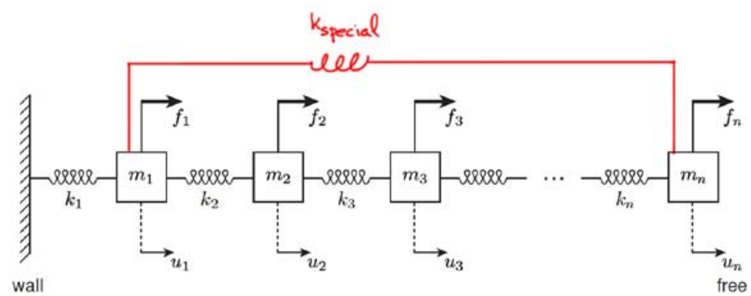
Gaussian Elimination and Back Substitution

$Au = f$ given f

A tri-diagonal/cyclic $n \times n$ matrix

MATLAB: $u = A \backslash f$ IF issparse(A) = true; otherwise...

system of springs



new matrix terms : before

$\sum$ forces on mass $1 = 0$

$$\Rightarrow f_1 - k_1 u_1 + k_2(u_2 - u_1) = 0 \,,$$

$\sum$ forces on mass $2 = 0$

$$\Rightarrow f_2 - k_2(u_2 - u_1) + k_3(u_3 - u_2) = 0 \,,$$

$\sum$ forces on mass $i = 0$ $(i \neq 1,\ i \neq n)$

force due to spring i

$$\Rightarrow f_i - \underbrace{k_i(u_i - u_{i-1}) + k_{i+1}(u_{i+1} - u_i)}_{\text{force due to springs on mass } i \text{ depends only on } u_{i-1},\, u_i,\, \text{and } u_{i+1}} = 0 \,,$$

force due to spring $i-1$

$\sum$ forces on mass $n = 0$

$$\Rightarrow f_n - k_n(u_n - u_{n-1}) = 0 \,.$$

. . . and after

$\sum$ forces on mass $1 = 0$

$+ k_{special}(u_n - u_1)$

$$\Rightarrow f_1 - k_1 u_1 + k_2(u_2 - u_1) = 0 \,,$$

$\sum$ forces on mass $2 = 0$

$$\Rightarrow f_2 - k_2(u_2 - u_1) + k_3(u_3 - u_2) = 0 \,,$$

$\sum$ forces on mass $i = 0$ $(i \neq 1,\ i \neq n)$

force due to spring i

$$\Rightarrow f_i - \underbrace{k_i(u_i - u_{i-1}) + k_{i+1}(u_{i+1} - u_i)}_{\text{force due to springs on mass } i \text{ depends only on } u_{i-1},\, u_i,\, \text{and } u_{i+1}} = 0 \,,$$

force due to spring $i-1$

$\sum$ forces on mass $n = 0$

$- k_{special}(u_n - u_1)$

$$\Rightarrow f_n - k_n(u_n - u_{n-1}) = 0 \,.$$

$+ k_{special}$ $\qquad\qquad$ $- k_{special}$

$$\underbrace{\begin{pmatrix} k_1+k_2 & -k_2 & & & 0 \\ -k_2 & k_2+k_3 & -k_3 & & \\ & -k_3 & k_3+k_4 & -k_4 & \\ & \ddots & \ddots & \ddots & \\ 0 & & & & -k_n \\ & & & -k_n & k_n \end{pmatrix}}_{\substack{K \\ n\times n}} \underbrace{\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix}}_{\substack{u \\ n\times 1}} = \underbrace{\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-1} \\ f_n \end{pmatrix}}_{\substack{f \\ n\times 1}}$$

Symmetric Positive-Definite

$- k_{special}$ $\qquad\qquad$ $+ k_{special}$

K is still sparse — $nnz(K) = 3n$, but

K is no longer tri-diagonal.

Gaussian Elimination and Back Substitution $\boxed{\text{visually}}$ DEMO

to form U and $\hat{f}$:

still $O(n)$ FLOPs, but constant larger;

U preserves most but not all zeros of upper(K)

FILL-IN *

skyline, optimal ordering, . . .

to solve $U u = \hat{f}$ for u:

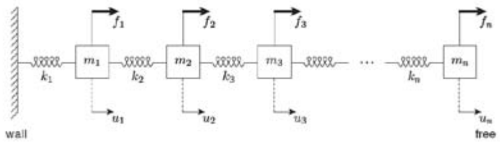
still $O(n)$ FLOPs, but constant larger.

* reason for interest in iterative solution methods

the Evil Inverse

$Au = f$ f given

A tri-diagonal $n \times n$ matrix



$\boxed{\begin{array}{l} \text{CONSTRUCT } A^{-1} \\ \text{CALCULATE } u = A^{-1}f \end{array}}$

column perspective $\rightarrow$ "density"

Write

$$A^{-1} = \begin{pmatrix} | & | & | & & | \\ p^1 & p^2 & p^3 & \cdots & p^n \\ | & | & | & & | \end{pmatrix}$$

1^{st} column of A^{-1} 2^{nd} column of A^{-1} n^{th} column of A^{-1}

such that

$$u = A^{-1}f = \begin{pmatrix} | & | & | & & | \\ p^1 & p^2 & p^3 & \cdots & p^n \\ | & | & | & & | \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$$

$$= p^1 f_1 + p^2 f_2 + \cdots + p^n f_n .$$ "one-handed"

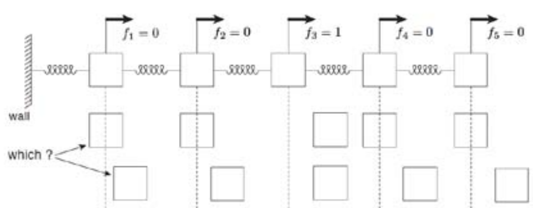
Hence,

$p^j \equiv$ displacements due to unit force applied to mass j.

p^j_i : displacement of mass i

Say for $j = 3$:

p^3



6

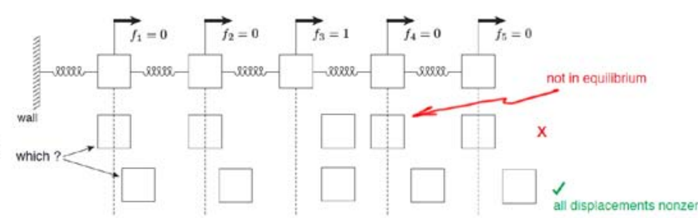
Hence,

$p^j \equiv$ displacements due to unit force applied to mass j.

p^j_i : displacement of mass i

Say for $j = 3$:

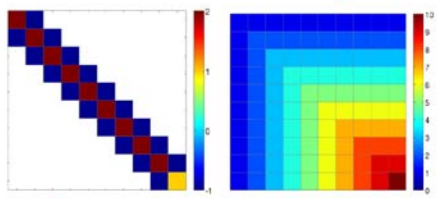
p^3



$\boxed{\Rightarrow p^j_i \text{ is non-zero for all } i, 1 \le i \le n, \text{ for all } j, 1 \le j \le n}$

Thus,

$$A^{-1} = \begin{pmatrix} p^1 & p^2 & \cdots & p^n \end{pmatrix}$$



A — forces as function of displacements is sparse

A^{-1} — displacements as function of forces is NOT sparse

operation count:

to construct A^{-1}: $O(n^2)$ FLOPs
[taking advantage of A tri-diagonal; otherwise $O(n^3)$]

to calculate $u = A^{-1} f$: $O(n^2)$ FLOPs
dense matrix-vector product

$\Rightarrow$ even if given A^{-1}, $u = A^{-1} f$ much more expensive than
$Au = f \rightarrow Uu = \hat{f} \rightarrow u$ $O(n)$

DEMO - B^{-1}

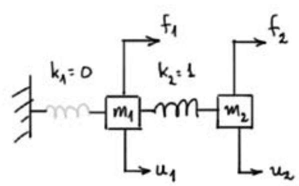
Note for small systems, non-sparse A, the inverse is not so evil and can sometimes be convenient/useful.

Breakdown and Stability

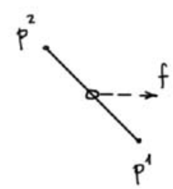
$Au = f \rightarrow Uu = \hat{f}$

$Uu = \hat{f} \rightarrow u$

breakdown: deserved



$$\Rightarrow \begin{array}{cc} p^1 & p^2 \end{array}$$
$$\underbrace{\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}}_{u} = \underbrace{\begin{pmatrix} f_1 \\ f_2 \end{pmatrix}}_{f}$$



$p^1 \parallel p^2 \Rightarrow$

$f \nparallel p^1$: no solution

$f \parallel p^1$: an infinity of solutions

If we perform Gaussian Elimination,

pivot			
$1\,u_1$	$-1\,u_2$	f_1	eqn 1
$-1\,u_1$	$1\,u_2$	f_2	eqn 2

$$1 \cdot \text{eqn } 1 + \text{eqn } 2 = 0u_1 + 0u_2 \quad \text{eqn } 2'$$

$$\Downarrow$$

$$\underset{U}{\begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}} \underset{u}{\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}} = \underset{\hat{f}}{\begin{pmatrix} f_1 \\ f_1 + f_2 \end{pmatrix}} ;$$

A singular $\Downarrow$ zero pivot

but then Back Substitution will fail:

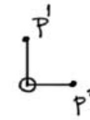
$$0 \cdot u_2 = f_1 + f_2 \quad . \qquad \div 0$$

(If $f_2 = -f_1$ then can construct non-unique solution.)

breakdown: undeserved

$$\underset{A}{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}} \underset{u}{\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}} = \underset{f}{\begin{pmatrix} f_1 \\ f_2 \end{pmatrix}}$$

$p^1 \quad p^2$



$u_1 = f_2$

$u_2 = f_1$

A non-singular

exists, unique

Gaussian Elimination:

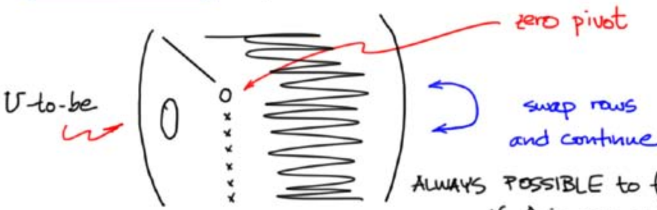
pivot				after swap
$0\,u_1$	$1\,u_2$	f_1	eqn 1	eqn 2
$1\,u_1$	$0\,u_2$	f_2	eqn 2	eqn 1

zero pivot: can not proceed, since

NO AMOUNT of eqn 1 will eliminate $1\,u_1$ of eqn 2

Simple fix: swap eqn 1 $\longleftrightarrow$ eqn 2.

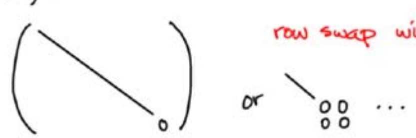
partial pivoting: general case



ALWAYS POSSIBLE to find non-zero x if A is non-singular

Note if A is SPD, no swaps will ever be necessary.

[If A is singular, will arrive at



row swap will not help]

Stability: effects of finite-precision arithmetic/round-off (fp)

model:

$$Au = f \xrightarrow{\text{round-off}} U_{fp}\, u_{fp} = f_{fp}$$

$$(A + \delta A)(u + \delta u) = (\hat{f} + \delta \hat{f}) \longrightarrow u + \delta u$$

Conditioning (condition number) of A: sensitivity of solution to perturbations

Note effects of round-off mitigated by partial pivoting: Choose largest possible × in row swap.

2.086 Numerical Computation for Mechanical Engineers
Spring 2013