<<< Vectors & Vector Space

> vector - object comprised of magnitude & direction

· starts at origin, given coordinate of its tip

· transpose - $(r_1, \dots, r_n)^T = \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$

· vector addition - $\vec{u} + \vec{v} = [\, u_1+v_1, \dots, u_n+v_n \,]^T$

· scalar multiple - $\alpha\vec{u} = [\, \alpha u_1, \dots, \alpha u_n \,]^T$

→ Vector Space

> objects that addition & scalar multiple are defined in:

A1. If $\vec{u}$ and $\vec{v}$ are in V, then $\vec{u}+\vec{v}$ is defined and in V

A2. Commutativity : $\vec{u}+\vec{v} = \vec{v}+\vec{u}$

A3. Associativity : $\vec{u} + (\vec{v}+\vec{w}) = (\vec{u}+\vec{v}) + \vec{w}$

A4. Additive identity : there is a vector $\vec{0} \in V$ such that $\vec{u} + \vec{0} = \vec{u}$ for all $\vec{u} \in V$

A5. Additive inverse : for all $\vec{u} \in V$, there is a vector $-\vec{u} \in V$ such that $\vec{u} + (-\vec{u}) = \vec{0}$

M1. If α is scalar, $\alpha\vec{u}$ is defined and in V

M2. Distributive & Associative property

$$\alpha(\vec{u}+\vec{v}) = \alpha\vec{u} + \alpha\vec{v}$$
$$(\alpha+\beta)\vec{u} = \alpha\vec{u} + \beta\vec{u}$$
$$\alpha\beta\vec{u} = \alpha(\beta\vec{u})$$

M3. $1\vec{u} = \vec{u}$

$0\vec{u} = \vec{0}$

$(-1)\vec{u} = -\vec{u}$

> field of scalar - $\mathbb{R}$ or $\mathbb{C}$

→ Examples of Vector Spaces

· $\mathbb{R}^n$

· space of polynomials of degree 2 or less : $a_0 + a_1 x + a_2 x^2$

> subset - a collection of some/all of elements of V

> subspace - a subset that's a vector space closed under definitions

· e.g. $(c, c)^T$

Linear Combination

> linear independent - if the only scalars $c_1, \dots, c_m$ for which $c_1\vec{v_1} + \dots + c_m\vec{v_m} = \vec{0}$ are $c_1 = \dots = c_m = 0$

> linear dependent - there exist scalars $c_1, \dots, c_m$, not all zero for $c_1\vec{v_1} + \dots + c_m\vec{v_m} = \vec{0}$

> linear combination - sum of scalar multiple of vectors (LC)

$$c_1\vec{v_1} + \dots + c_m\vec{v_m} \equiv \sum_{j=1}^{m} c_j\vec{v_j}$$

> nontrivial linear combination - LC that the scalars are not all zero

· vectors are linear dependent if there is a nontrivial LC equal to zero vector

Span

> span of vectors $\vec{v_1}, \dots, \vec{v_m}$ - set of all linear combinations of $\vec{v_1}, \dots, \vec{v_m}$

· If $\vec{v_1}, \dots, \vec{v_m}$ is in V, then span $(\vec{v_1}, \dots, \vec{v_m})$ is subspace of V

· $\vec{v_1}, \dots, \vec{v_m}$ span all of V if every vector $\vec{v} \in V$ can be written as LC of $\vec{v_1}, \dots, \vec{v_m}$.

Basis

> basis - $\vec{v_1}, \dots, \vec{v_m}$ are linear independent and span V

For $\mathbb{R}^n$, any set of n linear independent vectors is a basis of V

· e.g. $(1,0)^T$, $(0,1)^T$ basis of $\mathbb{R}^2$

· Not: · $(1,0)^T, (0,1)^T, (1,1)^T$ not basis of $\mathbb{R}^2$: not linear independent

· $(1,1)^T, (2,2)^T$ not basis of $\mathbb{R}^2$: do not span V

Standard Basis & Dimension

· infinite choices of basis for V → all basis have same # of elements

> dimension - number of vectors in any basis of V

· e.g. $\mathbb{R}^2$ dimension = 2 : $(1,0)^T, (0,1)^T$

dim(polynomial degree ≤ 2) = 3 : $1, x, x^2$

> standard basis - in $\mathbb{R}^n$, the set of basis vectors $\vec{e_1}, \dots, \vec{e_n}$ where $\vec{e_j}$ has 1 in position j and 0 everywhere else.

Coordinate

· Given a basis for V, any vector $\vec{v} \in V$ can be written uniquely as LC of the basis vectors

$$\vec{v} = \sum_{j=1}^{m} c_j\vec{v_j}$$

> coordinate - the scalars $c_1, \dots, c_m$ corresponding to basis $\vec{v_1}, \dots, \vec{v_m}$

<<< Dot Product

> dot product (inner product) (scalar product) of $\vec{u}$ and $\vec{v}$ in $\mathbb{R}^n$:

$$\vec{u} \cdot \vec{v} \equiv \langle \vec{u}, \vec{v} \rangle = \sum_{j=1}^{n} u_j v_j$$

> inner product - any rule that takes a pair of vectors $\vec{u}$ and $\vec{v}$ and maps them to a real number denoted $\langle \vec{u}, \vec{v} \rangle$

1. $\langle \vec{u}, \vec{v} \rangle = \langle \vec{v}, \vec{u} \rangle$
2. $\langle \vec{u}, \vec{v} + \vec{w} \rangle = \langle \vec{u}, \vec{v} \rangle + \langle \vec{u}, \vec{w} \rangle$
3. $\langle \vec{u}, \alpha \vec{v} \rangle = \alpha \langle \vec{u}, \vec{v} \rangle$ for real number α
4. $\langle \vec{u}, \vec{u} \rangle \geq 0$ with equality if and only if $\vec{u} = \vec{0}$.

> norm - $\|\vec{u}\| = \langle \vec{u}, \vec{u} \rangle^{\frac{1}{2}} = \sqrt{\sum_{j=1}^{n} u_j^2}$

<<< Orthonormality

> orthogonal - inner product of two non-zero vector is zero: $\langle \vec{u}, \vec{v} \rangle = 0$
- standard basis vector for $\mathbb{R}^n$ are orthogonal

> orthonormal - orthogonal vectors that have norm of 1
- standard basis vector for $\mathbb{R}^n$ are orthonormal
  - $\langle \vec{e}_i, \vec{e}_j \rangle = 0$, $i \neq j$
  - $\|\vec{e}_i\| = 1$, $i = 1, \ldots, n$

<<< Gram-Schmidt Algorithm

- construct orthonormal set $\vec{q}_1, \ldots, \vec{q}_m$ from linearly independent vectors $\vec{v}_1, \ldots, \vec{v}_m$
- $\tilde{\vec{q}}_j = \vec{v}_j - \sum_{i=1}^{j-1} \langle \vec{v}_j, \vec{q}_i \rangle \vec{q}_i$
- $\vec{q}_j = \frac{\tilde{\vec{q}}_j}{\|\tilde{\vec{q}}_j\|}$

e.g. $\tilde{\vec{q}}_1 = \vec{v}_1$ $\qquad \tilde{\vec{q}}_2 = \vec{v}_2 - \langle \vec{v}_2, \vec{q}_1 \rangle \vec{q}_1$

$\vec{q}_1 = \frac{\tilde{\vec{q}}_1}{\|\tilde{\vec{q}}_1\|}$ $\qquad \vec{q}_2 = \frac{\tilde{\vec{q}}_2}{\|\tilde{\vec{q}}_2\|}$

Matrices & Linear System

> Matrix - rectangular array of elements

> Gaussian elimination - eliminating variables by adding / subtracting multiples of one equation to another

- $A\vec{x} = \vec{b}$, where $b_i = \sum_{j=1}^{n} a_{ij}x_j$, $i=1,\dots,m$

  ($m\times n$ matrix) (n-vector) (m-vector)

$$\begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x+5y+9z \\ 2x+6y+10z \\ 3x+7+11z \\ 4x+8y+12z \end{bmatrix}$$

4×3 3×1 4×1

- $AB = C$, where $c_{ik} = \sum_{j=1}^{n} a_{ij}b_{jk}$, $i=1,\dots,m$, $k=1,\dots,p$

  ($m\times n$ matrix) ($n\times p$ matrix) ($m\times p$ matrix)

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 3 \\ 2 & 2 \end{bmatrix}$$

2×3 3×2 2×2

- number of column of A = number of rows of B
- $AB \neq BA$
- Matrix addition: elementwise addition
- Matrix multiplication:

$$AB = \begin{bmatrix} A\begin{bmatrix} b_{1i} \\ \vdots \\ b_{mi} \end{bmatrix}, \cdots, A\begin{bmatrix} b_{1p} \\ b_{mp} \end{bmatrix} \end{bmatrix} \quad (m\times p)$$

Existance & Uniqueness of Solutions

> homogenous - $\vec{b} = \vec{0}$ in $A\vec{x} = \vec{b}$, that is, $A\vec{x} = \vec{0}$
  - trivial solution: $\vec{x} = \vec{0}$
  - nontrivial solution: solutions where not all entries are zero

> inhomogenous - $\vec{b} \neq \vec{0}$ in $A\vec{x} = \vec{b}$

> range $(A) = \{A\vec{x} : \vec{x} \in \mathbb{R}^n\} \subset \mathbb{R}^m$ (the set of all things one can get from applying A)
  - all LC of columns of A
  - span of columns of A

> rank (A) - dim (range (A))

<<< Existance & Uniqueness of Solutions

$A\vec{x}=\vec{b}$

- $\vec{b}=0$ (homogeneous)
  - trivial solution $\vec{x}=0$
  - infinite solutions
- $\vec{b}\neq 0$ (inhomogeneous)
  - no solution
    - $\vec{b}$ is not in range of A
  - one solution (unique)
    - $\vec{b}$ is in range of A
    - homogeneous system has only trivial solution
  - infinite solutions
    - $\vec{b}$ is in range of A
    - homogeneous system has infinite soln
    - in form $\hat{y}+\vec{y}$, where $\hat{y}$ = particular soln, $\vec{y}$ = all soln of homo system

→ nxn matrix A

- homo system $A\vec{x}=\vec{0}$
  - trivial soln $\vec{x}=\vec{0}$ if columns of A are independent
  - the column of A spans $\mathbb{R}^n$, forming basis
- Inhomo system $A\vec{x}=\vec{b}$
  - only one unique solution

<<< Inverse of Matrix

> identity matrix I - 1 on diagonal, 0 elsewhere

- $AI=IA=A$
- $I\vec{x}=\vec{x}$

> invertible - for A, if there exist a matrix B such that $BA=I$

> Inverse - the matrix B satisfies $BA=I$, denoted by A^{-1}

- inverse is unique.
- If A is invertible, $A\vec{x}=\vec{b}$ has unique soln, columns of A linearly independent
  - $A^{-1}(A\hat{x}) = (A^{-1}A)\hat{x} = I\hat{x} = \hat{x} = A^{-1}\hat{b}$

> nonsingular - invertible matrix

> Singular - not invertible

→ Conclusion of nxn matrix

- equivalent
  - A is invertible (nonsingular)
  - columns of A are linearly independent
  - $A\vec{x}=\vec{0}$ has only trivial soln $\vec{x}=\vec{0}$
  - $A\vec{x}=\vec{b}$ has unique soln
  - rows of A linearly independent
  - $\det(A)\neq 0$

<<< Transpose

> transpose of A - matrix whose (i,j) entry is the (j,i) entry of A, denoted A^T

- $(AB)^T = B^T A^T$
- proves that row of A are linear independent if nxn A is invertible.

<<< Determinant

- $\det(A) = \det\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = a_{11}a_{22} - a_{12}a_{21}$
- $\det(A) = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} \det(A_{ij}) = \sum_{i=1}^{n} (-1)^{i+j} a_{ij} \det(A_{ij})$ for $(n-1)\times(n-1)$ A_{ij} matrix by deleting row i column j
  (for some i) (for some j)

> Laplace expansion

→ Cramer's Rule

- Solving nxn nonsingular A for $A\vec{x} = \vec{b}$
- $A \xleftarrow{j} \vec{b}$ - matrix that replaces jth column by vector $\vec{b}$.
- jth entry of solution $\vec{x}$ is

$$x_j = \frac{\det(A \xleftarrow{j} \vec{b})}{\det(A)} \qquad j = 1, \ldots, n$$

- $\pm m \times 2^E$, $1 < m < 2$

> Single precision (32 bits): 1 bit sign, 8 bit exp, 23 bit significand

→ hidden bit representation, not store the 1 before binary pt, since it's always 1.

> machine precision ϵ - gap between 1 and the next larger fltng pt #.

- single precision: $(1 + 2^{-23}) - 1 = 2^{-23} \approx 1.2e-7$

> double precision (64 bits): 1 sign, 11 exp, 52 significand

- $\epsilon = (1 + 2^{-52}) - 1 = 2^{-52} \approx 2.2e-16$

→ toy system

$1.\ b_1\ b_2$ $\quad \exp(0, 1, -1)$

$1.00 \times 2^0 = 1$	$1.01 \times 2^0 = \frac{5}{4}$	$1.10 \times 2^0 = \frac{3}{2}$	$1.11 \times 2^0 = \frac{7}{4}$
$1.00 \times 2^1 = 2$	$1.01 \times 2^1 = \frac{5}{2}$	$1.10 \times 2^1 = 3$	$1.11 \times 2 = \frac{7}{2}$
$1.00 \times 2^{-1} = 1/2$	$1.01 \times 2^{-1} = \frac{5}{8}$	$1.10 \times 2^{-1} = \frac{3}{4}$	$1.11 \times 2^{-1} = \frac{7}{8}$

- $\epsilon = (1 + 0.01) - 1 = 0.01 = 2^{-2} = \frac{1}{4}$
- abs gap between # becomes large as it moves away from origin
  - relative gap still ok.
- gap between smallest positive #&0 >> that of smallest & next smallest pos. #

~~$|1 + 2^{-2} - 0| = 1 + 2^{-2}$~~ $\quad |1.00 \times 2^{-1} - 1.01 \times 2^{-1}|$

$|1.00 \times 2^{-1} - 0| = 2^{-1}$ >> $= \underbrace{0.01}_{\epsilon} \times 2^{-1} = \underbrace{2^{-2}}_{\epsilon} \times 2^{-1} = 2^{-3}$

- the smallest pos. # in any system is 1×2^{-E},
- the next smallest pos. # is $(1 + \epsilon) \times 2^{-E}$

NOTE: gap between two smallest # is the machine precision $\epsilon \times 2^{-E}$

- gap between smallest & next smallest is ϵ times that of 0 & smallest.

IEEE Floating Point Arithmatic

- consistant representation of floating pt #
- correctly rounded arithmatics
- consistent & sensible treatment of exceptions
  - 0
  - $\pm \infty$ (n/0)
  - NaN (0/0)

$\exp_{(10)}$	$\exp_{(2)}$	#	comment
0	0....00	$\pm (0.b_1 \ldots b_{52}) \times 2^{-1022}$	0 / subnormal #
1	0.....01	$\pm (1.b_1 \ldots b_{52}) \times 2^{-1022}$	floating pt #
2	0.....10	$\pm (1.b_1 \ldots b_{52}) \times 2^{-1021}$	
1023	01111...11	$\pm (1.b_1 \ldots b_{52}) \times 2^{0}$	
2046	111.....10	$\pm (1.b_1 \ldots b_{52}) \times 2^{1023}$	actual exp + 1023
2047	1111...111	$\pm \infty$ if $b_1 = b_{52} = \cdots = 0$, NaN if otherwise	Exception

- Smallest + # : 2^{-1022}
- next smallest: $(1 + 2^{-52}) \times 2^{-1022}$
- gap: $2^{-52} \times 2^{-1022}$

Rounding

- Round down - largest fp# $\leq x$
- Round up - smallest fp# $\geq x$
- Round to $\emptyset$ - { round down if $x > 0$ / round up if $x < 0$
- Round to nearest

→ Absolute Rounding Error

> abs error = $|\text{round}(x) - x|$

- double precision : abs error $\leq 2^{-52} \times 2^{+E}$ any rounding mode $< \epsilon \times 2^{+E}$

  abs error $\leq 2^{-53} \times 2^{+E}$ round to nearest $\leq \frac{\epsilon}{2} \times 2^{+E}$

→ Relative Rounding Error

> rel error = $\frac{|\text{round}(x) - x|}{|x|} = \frac{\text{abs error}}{|x|} < \frac{\epsilon \times 2^E}{m \times 2^E} < \frac{\epsilon}{m} \leq \epsilon$

- any rounding mode $< \frac{\epsilon}{m} \leq \epsilon$
- round to nearest $\leq \frac{\epsilon}{2}$

- $\text{round}(x) = x(1+\delta)$, where $|\delta| < \epsilon$ for any rounding
  $|\delta| < \frac{\epsilon}{2}$ for round to nearest

→ Floating Point Operations

- $a \oplus b = \text{round}(a+b) = (a+b)(1+\delta_1)$
- $a \ominus b = \text{round}(a-b) = (a-b)(1+\delta_2)$
- $a \otimes b = \text{round}(ab) = (ab)(1+\delta_3)$
- $a \oslash b = \text{round}(a/b) = (a/b)(1+\delta_4)$

} $|\delta| < \epsilon$ for any rounding
$|\delta| < \frac{\epsilon}{2}$ for round to nearest,
$i = 1, \ldots, 4$

<<< Errors in Scientific Computation

- physical prob $\xrightarrow{\text{approx.}}$ math model
- math model $\xrightarrow{\text{approx.}}$ numeric model
- numeric model input data measurement error
- computer rounding error

> abs error = $|\hat{y} - y|$

> rel error = $\frac{|\hat{y} - y|}{|y|}$, y is real value, $\hat{y}$ is computed value

<<< Conditioning of Problem

> conditioning - how sensitive the answer is to small changes in input

> abs condition number $C(x)$: $|\hat{y} - y| = C(x)\,|y\,\hat{x} - x|$

> rel condition number $K(x)$: $\left|\frac{\hat{y} - y}{y}\right| = K(x)\left|\frac{\hat{x} - x}{x}\right|$

- $C(x) = \left|\frac{\hat{y} - y}{\hat{x} - x}\right| = \left|\frac{f(\hat{x}) - f(x)}{\hat{x} - x}\right| = |f'(x)|$

- $K(x) = \left|\frac{y - y}{y}\right| \left|\frac{x}{\hat{x} - x}\right| = \left|\frac{y - y}{\hat{x} - x} \cdot \frac{x}{y}\right| = \left|\frac{x\,f'(x)}{f(x)}\right|$

<<< Stability of Algorithm

> stable - algorithm that achieves level of accuracy defined by the conditioning of prob

> unstable - algorithm that gets unnecessarily inaccurate results due to roundoff

> forward error - output error

> backward error - input error

<<< Matrix Multiplication

- $A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$

- $A^T = \begin{bmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \cdots & a_{nm} \end{bmatrix}$

> Symetric - $A = A^T$

- $\vec{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$

- $A\vec{b} = \begin{bmatrix} a_{11}b_1 + \cdots + a_{1n}b_n \\ \vdots \\ a_{m1}b_1 + \cdots + a_{mn}b_n \end{bmatrix}$

<<< LU Decomposition

> $A \equiv LU$

- $A\vec{x} = \vec{b}$

  $LU\vec{x} = \vec{b}$

> $U\vec{x} \equiv \vec{y}$

- $L\vec{y} = \vec{b}$

$L_2 L_1 A = U$

$A = (L_2 L_1)^{-1} U$

$L = (L_1 L_2)^{-1}$

<<< Gaussian Elimination

- If A is $n \times n$ non-singular, for n-vector $\vec{b}$, $A\vec{x} = \vec{b}$ has unique soln $\vec{x}$
  - replacement - add to one row a multiple of another row
  - interchange - interchange two rows
  - scaling - multiply all entries by a nonzero constant
- after Gaussian elimination → upper triangular, solve by back sub.

→ Gaussian Elimination as Factorization

$L_2 L_1 A = U$ $\quad$ U: upper Δ matrix

let $L = (L_2 L_1)^{-1}$ $\quad$ L: lower Δ matrix

$A = LU$

$A\vec{x} = \vec{b}$

$LU\vec{x} = \vec{b}$

let $U\vec{x} = \vec{y}$

$L\vec{y} = \vec{b}$

→ Algorithm of Gaussian Elimination w/o partial pivoting

```
for j = 1 : n-1          % columns (for the pivots)
    for i = j+1 : n      % rows below j
        mult = A(i,j) / A(j,j);
        % A(i,:) = A(i,:) - mult * A(j,:);     % more than necessary
        A(i,j:n) = A(i,j:n) - mult * A(j,j:n);
        b(i) = b(i) - mult * b(j);
    end
end
```

- may fail - the (j,j) entry must be non-zero

<<< Operation Counts

- $\oplus \ominus \otimes \oslash$
- incrementing int loop index not counted
- Gaussian elimination operating on all elements

$$\sum_{j=1}^{n-1} \sum_{i=j+1}^{n} (2n+3) = (2n+3)\frac{n(n-1)}{2} = n^3 + O(n^2)$$

- Gaussian elimination operating efficiently on nonzero elements

$$\sum_{j=1}^{n-1} \sum_{i=j+1}^{n} [2(n-j)+5] = 2\sum_{k=1}^{n-1} k^2 + 5\sum_{k=1}^{n-1} k = 2\frac{(n-1)n(2n-1)}{6} + 5\frac{(n-1)n}{2}$$

$$= \frac{2}{3}n^3 + O(n^2)$$

<<< Operation Count (Cont.)

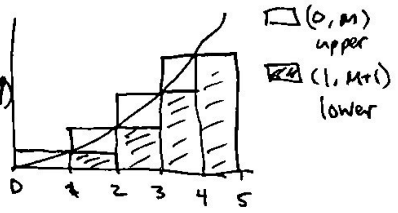
→ order equation

$$\sum_{n=1}^{3} n^p =$$

$$\int_0^m x^p\,dx \le 1^p + 2^p + \cdots + m^p \le \int_1^{m+1} x^p\,dx \qquad (m \in \mathbb{Z},\ p \in \mathbb{Z}_+)$$

upper bound: $= \frac{m^{p+1}}{p+1}$

lower bound: $= \frac{(1+m)^{p+1}-1}{p+1} = \frac{m^{p+1}}{p+1} + O(m^p)$



$$\sum_{n=1}^{m} n^p = \frac{m^{p+1}}{p+1} + O(m^p)$$

<<< LU Factorization

- use LU to solve systems w/ same coeff. matrix but diff right-hand side vector $\vec{b}$
- w/o extra storage

→ LU Factorization w/o pivoting

```
for j=1:n-1                                  % columns
    for i = j+1 : n                          % rows
        mult = A(i,j) / A(j,j)
        A(i, j+1:n) = A(i,j+1:n) - mult * A(j, j+1:n);   % upper Δ matrix U
        A(i,j) = mult;                       % lower Δ matrix L
    end
end
```

→ Solving $L\vec{y} = \vec{b}$ for $\vec{y}$

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \Rightarrow \quad y_1 = b_1 / l_{11} \quad \text{(fwd sub)}$$

$$y_2 = (b_2 - l_{21} y_1)\, l_{22}$$

$$y_i = \left(b_i - \sum_{j=1}^{i} l_{ij} y_j\right) / l_{ii}$$

→ MATLAB code : L has unit diagonal & lower Δ.

```
function y = lsolve(L, b)
    n = length(b);
    for i = 1:n
        y(i) = b(i);
        for j = 1 : i-1
            y(i) = y(i) - L(i,j) * y(j);
        end
    end
end
```

- no divided by diagonal term since have unit diagonal

→ Operation Count

$$\sum_{i=1}^{n} \sum_{j=1}^{i-1} 2 = n^2 + O(1)$$

Partial Pivoting

- avoids (j,j) entry being zero
- > pivoting - process of interchanging rows
- > pivot element - nonzero entry a_{kj} $(k>j)$
- If no nonzero entries of/below main diagonal in column j, matrix is singular
  - no soln
  - infinite soln

  } Error
- > partial pivoting: search for largest entry in the column in abs value & uses as pivot element
  - multiplier $< = 1$
  - $O(n^2)$ finding largest entry for each column
  - additional data movement cost

Theorem Every $n \times n$ nonsingular matrix A can be factored in the form $A = PLU$, where P is a permutation matrix, L is a unit lower Δ matrix, U is an upper Δ matrix

→ MATLAB Code

```
for j = 1 : n-1                          % columns
    [pivot, k] = max(abs(A(j:n, j)));    % largest value as pivot
    if pivot == 0                        % pivot = 0 → singular
        disp("Matrix is singular")
        break
    end
    temp = A(j,:);
    A(j,:) = A(k+j-1,:);
    A(k+j-1,:) = temp;
    tempb = b(j); % exchange rows of A & b
    b(j) = b(k+j-1);
    b(k+j-1) = b(j);
    for i = j+1 : n                      % make A(i,j) = 0
        mult = A(i,j) / A(j,j);
        A(i, j:n) = A(i, j:n) - mult * A(j, j:n);
        b(i) = b(i) - mult * b(j);
    end
end
```

<<< Banded Matrix

> symmetric matrix - if A satisfies $A = A^T$

> positive definite - if symetric matrix has all eigenvalues positive

- symmetric positive define matrix does not need pivot

> Cholesky decomposition - $A = LL^T$

> Strict Diagonal Dominance - $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ for all i, the abs of diagonal entry is greater than the sum of abs of off diagonal entries

> Banded - $a_{ij} = 0$ if $|i-j| > m$, where $m \ll n$ is the half bandwidth

- half bandwidth m
- full bandwidth $2m+1$
- eg. tridiagonal matrix - diagonal & one above & below, 0 elsewhere.

tridiagonal matrix in Gaussian Elimination: $3(n-1)$ ops

$$\begin{bmatrix} a_1 & b_1 & 0 & 0 & 0 \\ b_1 & a_2 & b_2 & 0 & 0 \\ 0 & b_2 & a_3 & b_3 & 0 \\ 0 & 0 & b_3 & a_4 & b_4 \\ 0 & 0 & 0 & b_4 & a_5 \end{bmatrix} \sim \begin{bmatrix} a_1 & b_1 & 0 & 0 & 0 \\ 0 & \tilde{a}_2 & b_2 & 0 & 0 \\ 0 & 0 & \tilde{a}_3 & b_3 & 0 \\ 0 & 0 & 0 & \tilde{a}_4 & b_4 \\ 0 & 0 & 0 & 0 & \tilde{a}_5 \end{bmatrix}$$

$\tilde{a}_2 = a_2 - \frac{b_2}{a_1} b_1$

(3 ops) each row 5

- GE of banded matrix is $2m^2 n$

<<< Implementation Considerations of High Performance

- Gaussian elimination { fetch data $3(n-j+1)$; arithmatics $2(n-j+1)$

- BLAS

BLAS 1 - vector operations, least effident $\vec{y} \leftarrow \alpha\vec{x} + \vec{y}$

- $3n$ memory access
- $2n$ floating pt operation

} ratio: $\frac{2}{3}$ flop/ma ← memory access

BLAS 2 - matrix vector operation, intermediate efficiency, $\vec{y} \leftarrow A\vec{x} + \vec{y}$

- $n^2 + 3n$ memory access
- $2n^2$ floating pt operation

} ratio: 2 flop/ma

BLAS 3 - matrix-matrix operation, highest efficiency, $C \leftarrow AB + C$

- $4n^2$ memory access
- $2n^2$ floating pt op.

} ratio: $\frac{n}{2}$ flop/ma

Cholesky Algorithm

- For A that's symmetric positive definite (SPD)

$A = LU = \mathcal{L}\mathcal{L}^T$ where L is lower Δ matrix with 1 on diagonal
$= LD\tilde{U}$ U is upper Δ matrix
$= LDL^T$ $\mathcal{L}$ is lower Δ matrix with nonzeros on diagonal
$= L D^{\frac{1}{2}} D^{\frac{1}{2}} L^T$ D is diagonal matrix of U
$= \mathcal{L}\mathcal{L}^T$ $\tilde{U}$ is upper Δ matrix with 1 on diagonal
- $\tilde{U} = L^T$
- $DU = \tilde{U}$

Ex1 $A = \begin{bmatrix} 3 & 1 \\ 1 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{3} & 1 \end{bmatrix}\begin{bmatrix} 3 & 1 \\ 0 & \frac{14}{3} \end{bmatrix} = \begin{bmatrix} \sqrt{3} & 0 \\ \frac{\sqrt{3}}{3} & \frac{\sqrt{14}}{3} \end{bmatrix}\begin{bmatrix} \sqrt{3} & \frac{\sqrt{3}}{3} \\ 0 & \frac{\sqrt{14}}{3} \end{bmatrix}$

$\begin{bmatrix} 3 & 1 \\ 1 & 5 \end{bmatrix} \sim \begin{bmatrix} 3 & 1 \\ 0 & \frac{14}{3} \end{bmatrix}$
A ↑ U ↑

$L_1 = \begin{bmatrix} 1 & 0 \\ -\frac{1}{3} & 1 \end{bmatrix}$ $L_1^{-1} = \begin{bmatrix} 1 & 0 \\ \frac{1}{3} & 1 \end{bmatrix}$ ← L ↑

LU decomposition
- keep track of mult in L
- take inverse of L_i as

$D = \begin{bmatrix} 3 & 0 \\ 0 & \frac{14}{3} \end{bmatrix}$ $\tilde{U} = L^T = \begin{bmatrix} 1 & \frac{1}{3} \\ 0 & 1 \end{bmatrix}$

$A = LD\tilde{U} = LDL^T = \begin{bmatrix} 1 & 0 \\ \frac{1}{3} & 1 \end{bmatrix}\begin{bmatrix} 3 & 0 \\ 0 & \frac{14}{3} \end{bmatrix}\begin{bmatrix} 1 & \frac{1}{3} \\ 0 & 1 \end{bmatrix}$

$= (LD^{\frac{1}{2}})(D^{\frac{1}{2}}L^T) = \begin{bmatrix} 1 & 0 \\ \frac{1}{3} & 1 \end{bmatrix}\begin{bmatrix} \sqrt{3} & 0 \\ 0 & \frac{\sqrt{14}}{3} \end{bmatrix}\begin{bmatrix} \sqrt{3} & 0 \\ 0 & \frac{\sqrt{14}}{3} \end{bmatrix}\begin{bmatrix} 1 & \frac{1}{3} \\ 0 & 1 \end{bmatrix}$

$= \mathcal{L}\mathcal{L}^T = \begin{bmatrix} \sqrt{3} & 0 \\ \frac{\sqrt{3}}{3} & \frac{\sqrt{14}}{3} \end{bmatrix}\begin{bmatrix} \sqrt{3} & \frac{\sqrt{3}}{3} \\ 0 & \frac{\sqrt{14}}{3} \end{bmatrix}$

Cholesky Algorithm
- find diagonal matrix D
- calculate $\tilde{U} = L^T$
- follow formula $LD^{\frac{1}{2}}D^{\frac{1}{2}}L^T$

Methods for Solving $A\vec{x} = \vec{b}$

→ Gaussian Elimination (& LU Decomposition)
- PLU factorization: $\frac{2}{3}n^3$ flops
- solve system: $2n^2$ flops

→ Inversion of Matrix
- append identity matrix to right of A and reduce A to identity matrix, the original identity matrix becomes A^{-1}: $[A|I] \sim [I|A^{-1}]$
- matrix inversion: $\frac{8}{3}n^3$ flops ← reduce A to upper Δ, apply multiplication to $\vec{e}_1, \vec{e}_2$
- solve system: $2n^2$ flops $\frac{2}{3}n^3 + n(2n^2) = \frac{8}{3}n^3$

<< Methods of Solving $A\vec{x}=\vec{b}$ (Cont.)

→ Cramer's Rule

· good for small, not for big

· jth component of solution $\vec{x}$: replace column j of A by $\vec{b}$, calc $\frac{\det(A_j(\vec{b}))}{\det(A)}$ ($A_j(\vec{b})$)

· determinat computation: $>n!$

→ Summary

Method	Operation Count
Gaussian Elimination	$\frac{2}{3}n^3$
Inversion of Matrix	$\frac{4}{3}n^3$
Cramer's Rule	$>n!$

<< Conditioning of Linear Systems

→ Norms

> A <u>norm</u> for vectors in a function $\|\cdot\|$ satisfying

(i) $\|\vec{v}\| \geq 0$, where $\|\vec{v}\|=0$ if and only if $\vec{v}=0$.

(ii) $\|\alpha\vec{v}\| = |\alpha|\,\|\vec{v}\|$ for any scalar α

(iii) $\|\vec{v}+\vec{w}\| \leq \|\vec{v}\|+\|\vec{w}\|$ (triangular inequality)

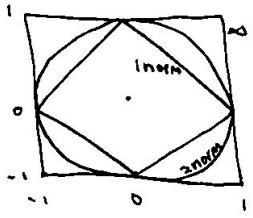
. Types of Norms

· 2-norm (Euclidean norm) - $\|\vec{v}\|_2 \equiv \sqrt{\sum_{i=1}^{n} |v_i|^2}$; $\|\vec{v}\|_2 = \langle \vec{v}, \vec{v} \rangle^{\frac{1}{2}}$, where $\langle \vec{x}, \vec{y} \rangle = \sum_{i=1}^{n} x_i y_i$ (2-norm from inner product)

· ∞-norm - $\|\vec{v}\|_\infty \equiv \max_{i=1,\dots,n} |v_i|$

· 1-norm - $\|\vec{v}\|_1 \equiv \sum_{i=1}^{n} |v_i|$

· p-norm ($p \geq 1$) - $\|\vec{v}\|_p \equiv \left(\sum_{i=1}^{n} |v_i|^p \right)^{\frac{1}{p}}$



Vectors with norms equal to 1.

> A <u>matrix norm</u> is function $\|\cdot\|$ for $m \times n$ matrix A, B

(i) $\|A\| \geq 0$, where $\|A\|=0$ if and only if $A=0$.

(ii) $\|\alpha A\| = |\alpha|\,\|A\|$ for any scalar α

(iii) $\|A+B\| \leq \|A\|+\|B\|$ (triangular inequality)

(iv) $\|AB\|$ ($m \times n$) $\leq \|A\| \cdot \|B\|$ ($n \times p$) (submultiplicative)

<<< Norms

→ Matrix Norm

· induced matrix norm: $\|A\| \equiv \max_{\|\vec{v}\|=1} \|A\vec{v}\| = \max_{\|\vec{v}\|\neq 0} \frac{\|A\vec{v}\|}{\|\vec{v}\|}$ ~ $\|A \frac{\vec{v}}{\|\vec{v}\|}\|$

· $\|A\| \geq \frac{\|A\vec{v}\|}{\|\vec{v}\|}$ → $\|A\vec{v}\| \leq \|A\| \cdot \|\vec{v}\|$ (vector norm { compatible with / subordinate to } matrix norm)

→ Matrix Norm induced by 1-norm

· maximum absolute column sum

· $\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{m} |a_{ij}|$ for mxn matrix induced by n-vector

→ Matrix Norm induced by ∞-norm

· maximum absolute row sum

· $\|A\|_\infty = \max_{i=1,\dots,n} \sum_{j=1}^{n} |a_{ij}|$ for mxn matrix induced by n-vector

→ Matrix Norm induced by 2-norm

· $\|A\|_2 = \sqrt{\text{largest eigenvalue of } A^TA}$

· square root of largest eigenvalue of A^TA.

<<< Conditioning of Linear Systems

→ Error in $\vec{b}$

· $A\vec{x} = \vec{b}$ is the exact system

· $A\hat{x} = \hat{b}$, where $\hat{b}$ is rounded, giving $\hat{x}$.

· $A(\vec{x} - \hat{x}) = \vec{b} - \hat{b}$

$\vec{x} - \hat{x} = A^{-1}(\vec{b} - \hat{b})$

$\|\vec{x} - \hat{x}\| \leq \|A^{-1}\| \cdot \|\vec{b} - \hat{b}\|$, $\boxed{C(x) \equiv \|A^{-1}\|}$

$\frac{\|A\vec{x}\|}{\|\vec{x}\|} \leq \|A\|$

$\frac{\|\vec{x} - \hat{x}\|}{\|\vec{x}\|} \leq \|A^{-1}\| \cdot \frac{\|\vec{b} - \hat{b}\|}{\|\vec{b}\|} \cdot \frac{\|\vec{b}\|}{\|\vec{x}\|} \overset{\vec{b} = A\vec{x}}{=} \|A^{-1}\| \cdot \frac{\|\vec{b} - \hat{b}\|}{\|\vec{b}\|} \cdot \frac{\|A\vec{x}\|}{\|\vec{x}\|} \leq \|A^{-1}\| \cdot \|A\| \frac{\|\vec{b} - \hat{b}\|}{\|\vec{b}\|}$

$\boxed{K(x) = \|A^{-1}\| \|A\|}$

> Condition number of nonsingular matrix - $K(x) \equiv \|A^{-1}\| \|A\|$

→ Error in A and $\vec{b}$

· Let $A\vec{x} = \vec{b}$ be exact, $A+E$ be another nonsingular matrix, $\hat{b}$ another n-vector, $\hat{x}$ satisfies $(A+E)\hat{b}$

then $\frac{\|\vec{x} - \hat{x}\|}{\|\vec{x}\|} \leq \|(A+E)^{-1}\| \cdot \|A\| \left(\frac{\|\vec{b} - \hat{b}\|}{\|\vec{b}\|} + \frac{\|E\|}{\|A\|} \right)$

$\boxed{K(A) = \|(A+E)^{-1}\| \|A\| \approx \|A^{-1}\| \|A\|}$

if $\|E\|$ is small enough so $\|A^{-1}\| \cdot \|E\| < 1$, then

$\frac{\|\vec{x} - \hat{x}\|}{\|\vec{x}\|} \leq \frac{K(A)}{1 - K(A)\|E\|/\|A\|} \cdot \left(\frac{\|\vec{b} - \hat{b}\|}{\|\vec{b}\|} + \frac{\|E\|}{\|A\|} \right)$

<<< Stability of Gaussian Elimination

- without pivoting - unstable
- with pivoting - almost always stable
  - backward stable - when solve $A\vec{x} = \vec{b}$, produce $\hat{x}$ satisfies $(A+E)\hat{x} = \hat{b}$,

$$\text{where } \frac{\|E\|}{\|A\|} = O(\varepsilon), \quad \frac{\|\hat{b} - \vec{b}\|}{\|\vec{b}\|} = O(\varepsilon)$$

: Suppose $PA = LU$, and GE w/ partial pivoting produces $\tilde{L}\tilde{U}$, then $\tilde{L}\tilde{U} = A + E$,

where $\frac{\|E\|}{\|L\|\|U\|} = O(\varepsilon)$

- there are exceptions, as proposed by Wilkinson.

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & -1 & 1 & 2 \\ 0 & -1 & -1 & 2 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & -1 & 4 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 8 \end{bmatrix}$$

if $n \times n$, then last entry is 2^{n-1}

<<< Overview of Least Squares Problem

- used for overdetermined system
- plotting close fit of data pts
- minimizing square of 2-norm of residual : $\min \|\vec{b}-A\vec{x}\|_2^2 = \sum_{i=1}^{m} (b_i - \sum_{j=1}^{n} a_{ij} x_j)^2$
- for $A\vec{x}=\vec{b}$, A is $m \times n$, where $m>n$, b is m-vector, x is n-vector

$$\underset{(m\times n)}{A}\ \underset{(n\times 1)}{\vec{x}} = \underset{(m\times 1)}{\vec{b}}, \quad \text{find closest } \vec{x}$$

<<< Normal Equations : approach I

- differentiate squared error to find min : (a is element of A)

$$\frac{\partial}{\partial x_k}\|\vec{b}-A\vec{x}\|^2 = \frac{\partial}{\partial x_k}\sum_{i=1}^{m}(b_i - \sum_{j=1}^{n} a_{ij}x_j)^2$$

$$= \sum_{i=1}^{m} 2(b_i - \sum_{j=1}^{n} a_{ij}x_j)(-a_{ik}) = 0, \quad k=1,\dots,n$$

so $$\sum_{i=1}^{m} a_{ik}b_i = \sum_{i=1}^{m}\left(a_{ik}(\sum_{j=1}^{n} a_{ij}x_j)\right) \equiv \sum_{i=1}^{m} a_{ik}(A\vec{x})_i$$

since $$\sum_{i=1}^{m} a_{ik}(A\vec{x})_i = \sum_{i=1}^{m} (A^T)_{ki}(A\vec{x})_i = \sum_{i=1}^{m} (A^T)_{ki}\, b_i$$

that is, $$\boxed{A^TA\vec{x} = A^T\vec{b}}$$ normal equation

EX1 For $A\vec{x}=\vec{b}$: $\begin{bmatrix}1&1\\1&-1\\2&1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}2\\0\\4\end{bmatrix}$, $A^T=\begin{bmatrix}1&1&2\\1&-1&1\end{bmatrix}$ ← overdetermined

by $A^TA\vec{x}=A^T\vec{b}$, $\begin{bmatrix}1&1&2\\1&-1&1\end{bmatrix}\begin{bmatrix}1&1\\1&-1\\2&1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}1&1&2\\1&-1&1\end{bmatrix}\begin{bmatrix}2\\0\\4\end{bmatrix}$

$$\begin{bmatrix}6&2\\2&3\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}12\\6\end{bmatrix}$$ ← NOT overdetermined

$$\begin{cases}x_1 = \frac{9}{7}\\ x_2 = \frac{8}{7}\end{cases}$$

- drawback - 2-norm condition # of A^TA is square of A

$$\|A^TA\|_2^2 = \sqrt{\max \text{eig}((A^TA)^T(A^TA))} \qquad \text{that is, } K_2(A^TA) = K_2(A)^2$$

$$= \sqrt{\max \text{eig}((A^TA)(A^TA))}$$

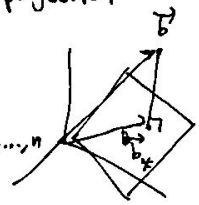
$$= \sqrt{\max \text{eig}((A^TA)^2)}$$

$$= \max \text{eig}(A^TA)$$

$$= \|A\|_2^2$$

<<< QR Decomposition : approach II

- goal - find $\vec{x}$ for $A\vec{x}=\vec{b}_*$, where $\vec{b}_*$ is closest to $\vec{b}$ in 2 norm in range(A)
- equiv goal - minimize $\|\vec{b}-A\vec{x}\|_2$, since $\|\vec{b}-\vec{b}_*\|_2 \leq \|\vec{b}-A\vec{y}\|_2$ for all $\vec{y}$
- the closest vector to a given vector from a subspace is the orthogonal projection of that vector onto that subspace.



→ Gram-Smidt Algorithm

- $\vec{q}_j = \frac{\tilde{q}_j}{\|\tilde{q}_j\|}$, where $\tilde{q}_j = \vec{v}_j - \sum_{i=1}^{j-1} \langle \vec{v}_j, \vec{q}_i \rangle \vec{q}_i$, given $\vec{v}_j$ for $j=1,\dots,n$

→ QR Decomposition (Reduced) Solving Least Squares Problem

- Given $A\vec{x}=\vec{b}$, where A is $m \times n$, where $m>n$
- Decompose A to $A=QR$, where Q has orthonormal columns, and R is upper Δ.

→ Algorithm

QR Decomposition:
1. Apply Gram-Schmidt Algorithm to A. The orthonormal vectors are columns of Q.
2. Find $R=Q^TA$.

proof:
$A=QR$
$Q^TA=Q^TQR$ ← ($Q^TQ=I$ for orthonormal matrix)
$Q^TA=R$

Use QR Decomp. Solve system:
3. Solve $R\vec{x}=Q^T\vec{b}$ for $\vec{x}$

proof:
$A\vec{x}=\vec{b}$
$QR\vec{x}=\vec{b}$
$Q^TQR\vec{x}=Q^T\vec{b}$
$R\vec{x}=Q^T\vec{b}$

$A = Q\ R$
$m\times n \quad m\times n \quad n\times n$

→ Full QR Decomposition

- First complete reduced QR Factorization
- Complete the orthonormal columns to an orthonormal basis for $\mathbb{R}^m$
- $A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$, $A: m\times n$, $Q: m\times m$, $\begin{bmatrix} R \\ 0 \end{bmatrix}: m\times n$

→ MATLAB Command

- [Q,R] = qr(A,0) for reduced QR
- [Q,R] = qr(A) for full QR

<<< Linear Fit

- Given a set of data (x_i, y_i) for $i = 1:m$
- fit linear equation $y = mx + b$ for all data pts

$$\begin{aligned} y_1 &\approx m x_1 + b \\ y_2 &\approx m x_2 + b \\ &\vdots \\ y_m &\approx m x_m + b \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_m & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

that minimizes squared error $\sum_{i=1}^{m} (y_i - (mx_i + b))^2$

- Overdetermined system can be approx. solved by { Normal equation / QR Decomposition

<<< Polynomial Fit

- Given m data pts, can fit $n-1 < m$ degree polynomial
- Find coefficients $c_0, c_1, \dots c_{n-1}$ that

$$y_i \approx c_0 + c_1 x_i + c_2 x_i^2 + c_3 x_i^3 + \cdots c_{n-1} x_i^{n-1}, \quad i = 1:m$$

so that squared error $\sum_{i=1}^{m} (y_i - \sum_{j=0}^{n-1} c_j x_i^j)^2$ is minimized.

- Solve by QR / normal eq:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$m \times n$ $\quad n \times 1$ $\quad m \times 1$

<<< Power Method

→ Power Method

> for computing the eigenvalue of the largest abs value and corresponding normalized eigenvector. ($|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$)

Given nonzero vector $\vec{w}$, set $\vec{y}^{(0)} = \frac{\vec{w}}{\|\vec{w}\|}$.

For $k = 1, 2, \dots,$

Compute $\tilde{y}^{(k)} = A\vec{y}^{(k-1)}$

Set $\lambda^{(k)} = \langle \tilde{y}^{(k)}, \vec{y}^{(k-1)} \rangle$ [Note that $\lambda^{(k)} = \langle A\vec{y}^{(k-1)}, \vec{y}^{(k-1)} \rangle$]

Form $\vec{y}^{(k)} = \frac{\tilde{y}^{(k)}}{\|\tilde{y}^{(k)}\|}$

- only works if the largest abs(eigvalue) is strictly greater than others
- rate of convergence depend on $\left|\frac{\lambda_2}{\lambda_1}\right|$, the smaller the better

→ Power Method with Shift

> Computing the eigenvalue farthest from shift s and the corresponding normalized eigvec.

$(A - sI)\vec{v} = (\lambda - s)\vec{v}$

Given nonzero vector $\vec{w}$, set $\vec{y}^{(0)} = \frac{\vec{w}}{\|\vec{w}\|}$

For $k = 1, 2, \dots$

Compute $\tilde{y}^{(k)} = (A - sI)\vec{y}^{(k-1)}$

Set $\lambda^{(k)} = \langle \tilde{y}^{(k)}, \vec{y}^{(k-1)} \rangle + s$

Form $\vec{y}^{(k)} = \frac{\tilde{y}^{(k)}}{\|\tilde{y}^{(k)}\|}$

<<< Deflation

> deflation - process of modifying initial vector or other vectors generated by the algorithm so that they are orthogonal to the already computed eigenvectors.

- Given: A is symmetric with eigvalues $\lambda_1, \dots, \lambda_n$ satisfying $|\lambda_1| > |\lambda_2| > |\lambda_3| \geq \dots \geq |\lambda_n|$
  λ_1 and $\vec{v}_1$ found by power method.
- Let $\vec{w} = \sum_{j=1}^{n} c_j \vec{v}_j$, $\hat{w} = \vec{w} - \langle \vec{w}, \vec{v}_1 \rangle \vec{v}_1 = \sum_{j=2}^{n} c_j \vec{v}_j$

$$A^k \hat{w} = \sum_{j=2}^{n} c_j A^k \vec{v}_j = \sum_{j=2}^{n} c_j \lambda_j^k \vec{v}_j = \lambda_2^k \left[c_2 \vec{v}_2 + \underbrace{\sum_{j=3}^{n} c_j \left(\frac{\lambda_j}{\lambda_2}\right)^k \vec{v}_j}_{\lim_{k\to\infty} = 0} \right]$$

converge to λ_2

- Let $\tilde{y}^{(k)} = \hat{y}^{(k)} - \langle \tilde{y}^{(k)}, \vec{v}_1 \rangle \vec{v}_1$ to prevent going back to $\vec{v}_1$
- Can use to find all eigenpairs

Inverse Iteration

- Given A is invertible
- eigvectors of $(A-sI)^{-1}$ are same as A; the eigvalues are $(\lambda_i - s)^{-1}$ $\forall\ i = 1, \dots, n$

$$A\vec{v}_j = \lambda_j \vec{v}_j$$

$$(A-sI)\vec{v}_j = \lambda_j \vec{v}_j - s\vec{v}_j = (\lambda_j - s)\vec{v}_j$$

$$(A-sI)^{-1}(A-sI)\vec{v}_j = (A-sI)^{-1}(\lambda_j - s)\vec{v}_j$$

$$\frac{1}{\lambda_j - s}\vec{v}_j = (A-sI^{-1})\vec{v}_j$$

> computing the eigenvalue of A that is closest to shift s and the corresponding normalized ~~vecto~~ eigvector.

- Given a shift s and a nonzero vector $\vec{w}$.

Set $\vec{y}^{(0)} = \frac{\vec{w}}{\|\vec{w}\|}$

For $k = 1, 2, \dots$

Solve $(A-sI)\,\tilde{y}^{(k)} = \vec{y}^{(k-1)}$ for $\tilde{y}^{(k)}$

Set $\lambda^{(k)} = \frac{1}{\langle \tilde{y}^{(k)}, \vec{y}^{(k-1)} \rangle} + s$ [Note $\frac{1}{\lambda^{(k)} - s} = \langle (A-sI)^{-1}\vec{y}^{(k-1)}, \vec{y}^{(k-1)} \rangle$]

Form $\vec{y}^{(k)} = \frac{\tilde{y}^{(k)}}{\|\tilde{y}^{(k)}\|}$

- The max abs eigvalue of $(A-sI)^{-1}$ is the largest $\frac{1}{\lambda_j - s}$, which is the smallest $\lambda_j - s$.
- inverse iteration has faster convergence
- inverse iteration can converge to any eigvalue.

} inverse of small eigvalue in the middle can be very large

<<< Intro to Eigenvalues & Eigenvectors

> eigenvector - $\vec{v}$ that satisfies $A\vec{v} = \lambda\vec{v}$
> eigenvalue - λ that satisfies $A\vec{v} = \lambda\vec{v}$
} A is (nxn) → can be real / complex : $\lambda \in \mathbb{C}$

> characteristic polynomial - $\det(A - \lambda I) = 0$
- polynomial rootfinding can be ill-conditioned

> eigenspace of λ - $\{\vec{0}$, eigenvectors of $\lambda\}$
- if $\vec{v}$ is an eigenvector of λ, then $c\vec{v}$ is also. $(c \neq 0)$
  - $A\vec{v} = \lambda\vec{v} \Rightarrow A(c\vec{v}) = \lambda(c\vec{v})$
- if $\vec{v_1}, \ldots, \vec{v_m}$ are eigvec of λ, then linear comb. of them are also.

> geometric multiplicity of λ - dimension of eigenspace
- eigenspace of λ == nullspace $(A - \lambda I)$
  - use $(A - \lambda I)\vec{v} = 0$ to find eigenvec.

<<< Diagonalization

> diagonalizable - nxn matrix A that has n linear independent eigvec.

$$A\left(\vec{v_1}, \vec{v_2}, \ldots, \vec{v_n}\right) = \left(\lambda_1\vec{v_1}, \lambda_2\vec{v_2}, \ldots \lambda_n\vec{v_n}\right) = \left(\vec{v_1}, \vec{v_2}, \ldots, \vec{v_n}\right)\begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix}$$

$$AV = V\Lambda \quad \begin{cases} V = (\vec{v_1}, \vec{v_2}, \ldots, \vec{v_n}) \\ \Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \end{cases}$$

$$A = V\Lambda V^{-1} \qquad \Lambda = V^{-1}AV$$

<<< Similarity Transformation

> similarity transformation - $A \to W^{-1}AW$ (W is nonsingular)

Theorem 1 If A is nxn matrix with n distinct eigenvalues, then A is diagonalizable

Theorem 2 If A is real & symmetric $(A = A^T)$, then the eigvalues of A are real and A is diagonalizable via an orthogonal similar transformation : $A = Q\Lambda Q^T$, where $Q^T = Q^{-1}$, Λ is $\begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$

- not all nxn matrix have n indepedent eigenvectors
- some may have double root at an eigvalue

Theorem 3

Jordan Canonical Form

Every $n \times n$ matrix A is similar to one of the form

$$J = \begin{bmatrix} J_1 & & \\ & \ddots & \\ & & J_m \end{bmatrix},$$

where block J_i has form

$$J_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}$$

The number of linear independent eigenvectors is the number of blocks m.
The matrix is diagonalizable if and only if $m = n$.
The geometric multiplicity of an eigvalue λ_i is the number of Jordan blocks w/ λ_i.
The algebraic multiplicity of λ_i (its degree as a root of characteristic polynomial) is the sum of the orders of all Jordan blocks with eigvalue λ_i

- A tiny change in matrix can make huge change in Jordan form

Theorem 4

Schur Form

Every square matrix A can be written in the form $A = QTQ^*$, where Q is a unitary matrix, and T is upper triangular.

Theorem 5

Gerschgorin

Let A be an $n \times n$ matrix with entries a_{ij} and let r_i denote the sum of the abs values of off-diag entries in row i: $r_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$.
Let D_i denote disk in complex plane centered at a_{ii} and has radius r_i: $D_i = \{ z \in \mathbb{C} : |z - a_{ii}| \leq r_i \}$

Then all eigvalues of A lie in the union $\bigcup_{i=1}^{n} D_i$ of the Gerschgorin disks. If m of these disks are connected and disjoint from the others, then exactly m eigvalues of A lie in this connected component.

- This works similarly for column disks.

29 May 2020 Singular Value Decomposition AMATH 352

<<< Intro to SVD

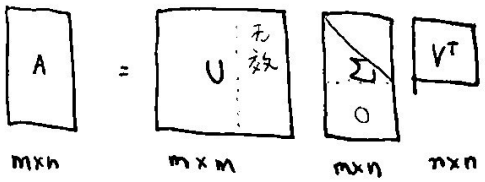
$$A = U \Sigma V^T$$

[U,S,V] = svd(A) - full SVD

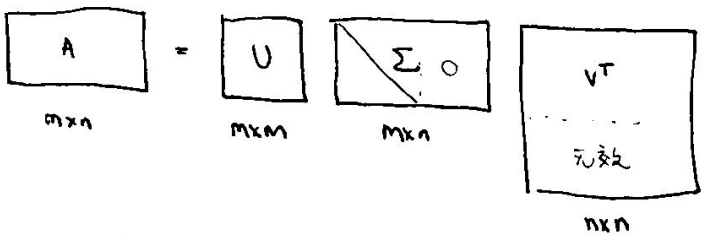
[U,S,V] = svd(A,0) - reduced SVD

- A - any mxn matrix
- U - matrix of left singular vectors, orthogonal vectors $AV = U\Sigma$
- Σ - matrix with singular values at diagonal
- V - matrix of right singular vectors, orthogonal vectors

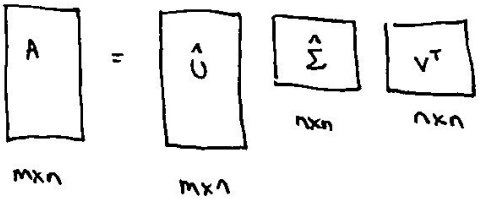
→ m×n, m>n, full SVD



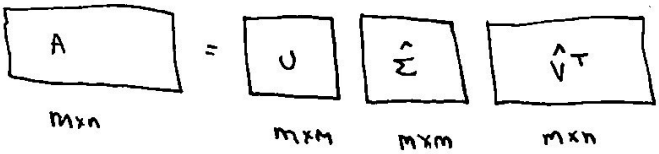
→ m<n, full SVD



→ m>n, reduced SVD



→ m<n, reduced SVD



→ ~~m=n~~ m=n

- full and reduced SVD are the same.

<<< SVD Theorem

Let A be mxn matrix,

$$\boxed{A = U\Sigma V^T}$$

where U is mxn matrix with orthonormal columns

Σ is mxn matrix with nonzeros only on its diag, where entries are real & nonneg

V is nxn matrix with orthonormal columns

is the full singular value decomposition (SVD)

> right singular vectors - columns of V

> singular values - diagonal entries of Σ

> left singular vectors - columns of U

· Right singular vectors corresponding to distinct singular values are determined up to multiplication by scalars of modulus 1 (± 1).

· Left singular vectors can be determined if right singular vectors are determined.

<<< SVD facts

· Known $A = U\Sigma V^T$, $\boxed{A^TA = V\Sigma^2V^T}$ is eigendecomposition of A^TA if A is square

$$A^TA = (U\Sigma V^T)^T(U\Sigma V^T) = (V\Sigma^TU^T)(U\Sigma V^T) = V\Sigma^2V^T$$

Since $U^TU = I$, $\Sigma^T = \Sigma$

· Singular value of A → nonnegative $\sqrt{eig(A^TA)}$

· right sgvector of A → eigvec (A^TA)

· $\boxed{AA^T = U\Sigma^2U^T}$ is eigendecomposition of AA^T

$$AA^T = (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V\Sigma^TU^T) = U\Sigma^2U^T$$

since $V^TV = I$, $\Sigma^T = \Sigma$

· singular value of A → nonnegative $\sqrt{eigval(AA^T)}$

· left singular vector of A → eigvec (AA^T)

· $AV = U\Sigma$

· $A^TU = V\Sigma$

Algorithm of Singular Value Decomposition

- Given $m \times n$ matrix A
- Get $A = U \Sigma V^T$

1. Find A^TA and AA^T
2. Calculate eigenvalue & eigenvector of A^TA
   - singular val of A = $\sqrt{\text{eigval}(A^TA)}$
   - right singular vec of A = $\text{eigvec}(A^TA)$
3. Use $AV = U\Sigma$, known A, V, Σ already, solve U.

→ Easy Matrices

1. Diagonal Matrices

   - $A = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix}$, $d_j \geq 0$, $\forall j = 1, \dots, n$

     $\Sigma = A$, $U = V = I$

   - $A = \begin{bmatrix} |d_1| & & \\ & \ddots & \\ & & |d_1| \end{bmatrix} \begin{bmatrix} \pm 1 & & \\ & \ddots & \\ & & \pm 1 \end{bmatrix}$ if some d_j is negative

     $\Sigma = \text{diag}(|d_1|, \dots, |d_n|)$, $U = I$, $I = \text{diag}(\pm 1, \dots, \pm 1)$

2. Real Symmetric Matrices

   $A = Q \Lambda Q^T$, $Q^T = Q$: $\Sigma = |\Lambda|$

3. Orthogonal Matrices

   - square matrix with orthonormal columns
   - $\Sigma = I$ since $A^TA = AA^T = I$
   - $\begin{cases} V = A^T, U = I & (AV = AA^T = I = U\Sigma) \\ V = I, U = A & (AV = A = U\Sigma) \end{cases}$

<<< Solving Linear Systems

→ Square Nonsingular Matrix

$A\vec{x} = \vec{b}$ and $A = U\Sigma V^T$

$$U\Sigma V^T \vec{x} = \vec{b}$$

$$\vec{x} = V\Sigma^{-1}U^T\vec{b} \quad \text{since } U^T = U^{-1},\ V^T = V^{-1}$$

· not usually used: Gaussian elimination faster

→ Overdetermined System ($m > n$)

$A\vec{x} \approx \vec{b}$

$$\vec{x} = V\hat{\Sigma}U^T\vec{b}$$

· not usually used: QR decomposition faster

<<< Matrix Approximation

· If $A = U\Sigma V^T$,

then $A = \sum_{j=1}^{\min(m,n)} \sigma_j \vec{u_j}\vec{v_j}^T$

· σ_j is (j,j) entry of Σ

· $\vec{u_j}$ is jth column of U

· $\vec{v_j}$ is jth column of V

· Each $\sigma_j \vec{u_j}\vec{v_j}^T$ is rank 1.

· Assume that A satisfy above approx, where $\sigma_1 > \sigma_2 \geq \dots \geq \sigma_{\min(m,n)}$.

For any $\nu < \min(m,n)$, define

$$A_\nu = \sum_{j=1}^{\nu} \sigma_j \vec{u_j}\vec{v_j}^T$$

Then A_ν is the closest rank ν matrix to A in both 2-norm and the Frobenius norm:

$$\|A - A_\nu\|_2 = \min_{\substack{B \in \mathbb{R}^{m\times n} \\ \text{rank}(B) \leq \nu}} \|A - B\|_2 = \sigma_{\nu+1}$$

$$\|A - A_\nu\|_F = \min_{\substack{B \in \mathbb{R}^{m\times n} \\ \text{rank}(B) \leq \nu}} \|A - B\|_F = \sqrt{\sigma_{\nu+1}^2 + \dots + \sigma_{\min(m,n)}^2}$$