Unit 2:

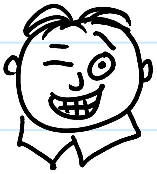
Linear Algebra

Solving systems of linear equations -- the how and why.

LA-01: Google Page Rank

Goal: To introduce what Google page rank is, and one way to compute it.

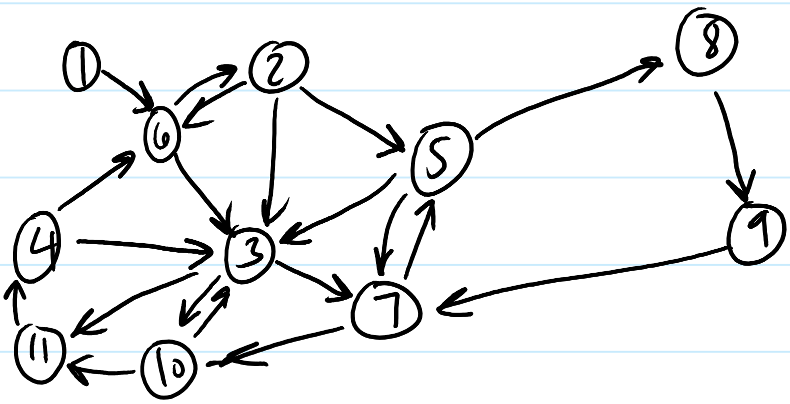
Meet Randy, the random web surfer.
His job is to visit web page after web page…
just surf the web.
Each time he clicks to a new web page, he looks at the links on the page, and chooses one randomly (hence his name).



If we give Randy enough time, he will eventually visit all the pages on the web… more than once if we let him continue.

But he will not visit all the pages with equal frequency. For example, he will probably load the UWaterloo CS home page more often than the web page "Enjoyable Dental Procedures". The dental page might only be linked to from Prof. Orchard's home page, but the UW CS web page is linked to by many pages.

Consider this tiny web, where arrows indicate outgoing links:



Intuitively, it seems that Randy will visit 3 more often than 9.

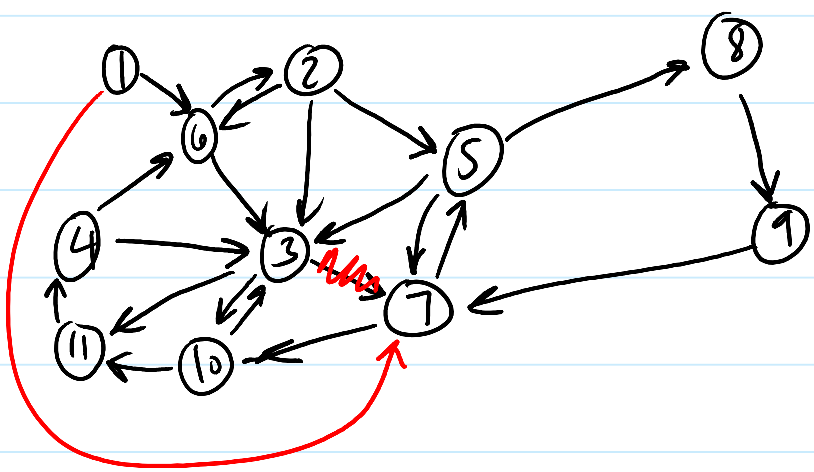
Code for Randy

```
%% Random walk
Init; % Set up a graph adjacency matrix in G
N = size(G,1); % number of nodes
```

```
visits = zeros(N,1); % counts number of visits

numSteps = 10000; % number of steps to take
j = ceil(rand * N); % start at a randomly-choosen node

for t = 1:numSteps
   visits(j) = visits(j) + 1; % increment counter for node j
   outLinks = find(G(:,j)==1); % list outgoing links for node j
   numOutLinks = length(outLinks);

   j = outLinks( ceil(rand*numOutLinks) ); % choose one randomly
end

plot(1:N, visits);
```

One way to measure the value or importance of a web page is **what fraction of the time a random surfer will spend there.** Notice that it is better to be linked to by other important web pages since Randy will visit those more often, and hence visit your page that much more frequently.

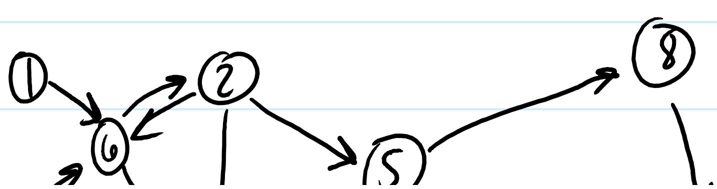
Try removing the link from $3 \to 7$, and replace it with $1 \to 7$.

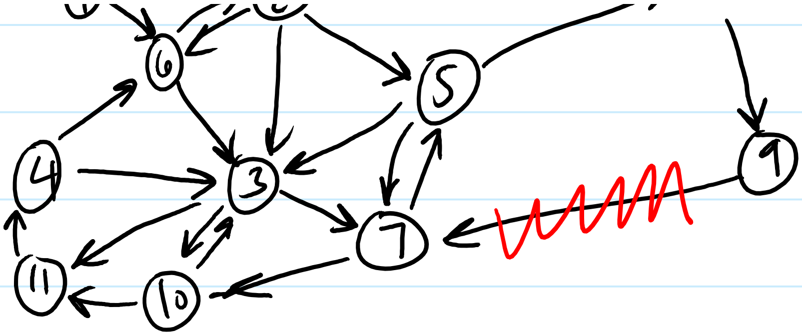


The importance of 7 decreases.

Problem: Terminal Pages

What if Randy reaches a page that has no outgoing links?



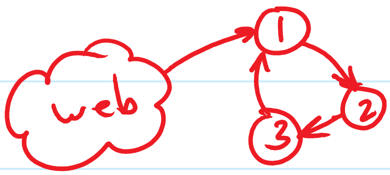


Randy gets stuck and the random walk does not give a good ranking of web page importance.

Solution: Teleportation

When a web page has no out links, **jump to another web page that is chosen randomly.**

Problem: The above issue is actually more insidious. Consider...



This is a terminal branch, and can be very difficult to detect.

Solution: Random teleportation

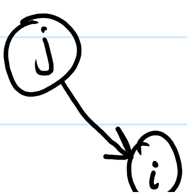
At each page, we **jump to a random web page with probability** $(1-\alpha)$.

$\therefore$ we will follow an out-link with probability α.

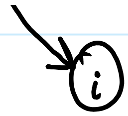
Mathematical Formulation of Random Surfer Ranking

Recall matrix $G \in \{0,1\}^{R \times R}$

$$G_{ij} = \begin{cases} 1 & \text{if node } j \text{ links to node } i \\ 0 & \text{otherwise} \end{cases}$$



$\qquad \bigg|\; 0$ otherwise



Consider, instead, matrix $P \in [0,1]^{R \times R}$

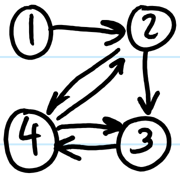
$$P_{ij} = \begin{cases} \frac{1}{\deg(j)} & \text{if node } j \text{ links to node } i \\ 0 & \text{otherwise} \end{cases}$$

where $\deg(j)$ is the # of outlinks coming from page j.

Then P_{ij} is the probability of following a link to node i given that you are at node j.

$$P_{ij} = \Pr(i \mid j) \qquad \text{(conditional probability)}$$

eg.



$$P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 1 & 0 \end{bmatrix}$$

Instead of following a single surfer, we can track the progress of an infinite number of surfers (sorry Randy) using the matrix P.

eg. If all our surfers start at ② . . .

i.e. $x = [0 \;\; 1 \;\; 0 \;\; 0]^T$

$$Px = [0 \;\; 0 \;\; \tfrac{1}{2} \;\; \tfrac{1}{2}]^T$$

$$P^2x = P(Px) = [0 \;\; \tfrac{1}{4} \;\; \tfrac{1}{4} \;\; \tfrac{1}{2}]^T$$

$$P^3x = [0 \;\; \tfrac{1}{4} \;\; \tfrac{3}{8} \;\; \tfrac{3}{8}]^T \;\; \ldots$$

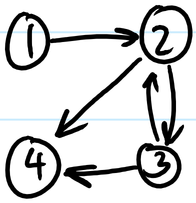
$\therefore$ after 3 clicks, $\frac{3}{8}$ of the surfers (37.5%) will be at node 3.

$\swarrow = \Sigma|x|$

Notice that $\|P^n x\|_1 = \|x\|_1$. All the surfers are accounted for. This is because each column of P redistributes all the surfers. Note also that all entries of P are non-negative. Each column of P is a probability distribution, and hence each column adds to 1.

$$\sum_{i=1}^{R} P_{ij} = 1$$

<u>Terminal Nodes</u>: Consider



$$P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$$

The column corresponding to a terminal node does not add to 1; it adds to 0.

$$Px = P[0 \;\; 1 \;\; 0 \;\; 0]^T = [0 \;\; 0 \;\; \tfrac{1}{2} \;\; \tfrac{1}{2}]^T$$

$$P^2x = [0 \;\; \tfrac{1}{4} \;\; 0 \;\; \tfrac{1}{4}] \Rightarrow \text{only } 50\% \text{ left}$$

$$P^3x = [0 \;\; 0 \;\; \tfrac{1}{8} \;\; \tfrac{1}{8}] \Rightarrow 33\% \text{ left}$$

$$\vdots$$

$$P^{20}x = \cdots \Rightarrow 0.0001\% \text{ left}$$

The matrix equivalent of the teleportation effect can be included...

$$P' = P + \frac{1}{R} e d^T$$

$$e = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \updownarrow R \qquad d_j = \begin{cases} 1 & \text{if node } j \text{ has no outlinks} \\ 0 & \text{otherwise} \end{cases}$$

In our example,

In our example,

$$P' = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{4} \\ 1 & 0 & \frac{1}{2} & \frac{1}{4} \\ 0 & \frac{1}{2} & 0 & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$$

To address the issue of terminal branches, we add a background randomness.

$$M = \alpha P' + (1-\alpha)\frac{1}{R} e e^T$$

$$(1-\alpha)\begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}$$

This guarantees that no surfer gets stuck in a subnet of the web.

M is called a Markov Transition Matrix.

Goal: To learn what a Markov Transition Matrix is, and what its properties are.

Markov Transition Matrices

Def'n A matrix Q is a Markov matrix if

$$0 \le Q_{ij} \le 1 \quad \text{and} \quad \sum_i Q_{ij} = 1 \quad \text{(sum down cols)}$$

Clearly, M (from the previous lecture) is a Markov matrix.

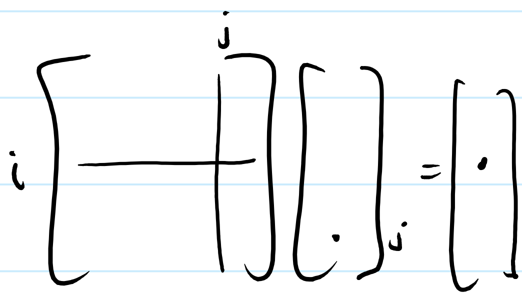
Def'n A vector q is a probability vector if

$$0 \le q_i \le 1 \quad \text{and} \quad \sum_i q_i = 1$$

Multiplying a Markov matrix by a probability vector yields another probability vector.

Consider $p = \underset{\text{Markov mat.}}{M}\,\underset{\text{prob. vect.}}{x}$

$$\sum_i p_i = \sum_i [Mx]_i$$
$$= \sum_i \sum_j M_{ij} x_j$$
$$= \sum_j x_j \sum_i M_{ij}$$
$$= \sum_j x_j = 1$$



$\therefore$ p is a prob vector.

Page Rank

Let v^n be a value vector indicating the value, or importance, of

Page Rank

Let p^n be a value vector indicating the value, or importance, of each page on the web. Without loss of generality (WLOG), we scale it so that

$$\sum_i [p^n]_i = 1$$

Thus, you can also think of it as a distribution of random surfers on the web.

If we allow the surfers one more click, then we get

$$p^{n+1} = M p^n \qquad (1)$$

We are looking for a steady-state solution... a fixed-point solution of (1)

i.e. $p = Mp$

This is actually an eigenvalue problem,

$$p = Mp \iff (I - M)p = 0$$

Eigenvalues & Eigenvectors

The eigenvectors, x, of a square matrix Q satisfy

$$Qx = \lambda x$$

for some scalar $\lambda \in \mathbb{C}$. This λ is called an eigenvalue.

To find eigenvalues, we can form the charactersitic polynomial.

$$Qx = \lambda x$$

$$\Rightarrow (\lambda I - Q)x = 0 \qquad (2)$$

Any nontrivial (ie. not entirely zero) x that satisfies (2) is an eigenvector. For there to be a nontrivial solution, the matrix $\lambda I - Q$ has to be singular.

eigenvector. For there to be a nontrivial solution, the matrix $\lambda I - Q$ has to be singular.

$$\Rightarrow \det(\lambda I - Q) = 0$$

The determinant is a polynomial in λ called the characteristic polynomial. The roots of the char. poly. are the eigenvalues. The x that solves $(\lambda I - Q)x = 0$ is the corresponding eigenvector.
In our case, we are looking for an eigenvetor corresponding to an eigenvalue of 1,

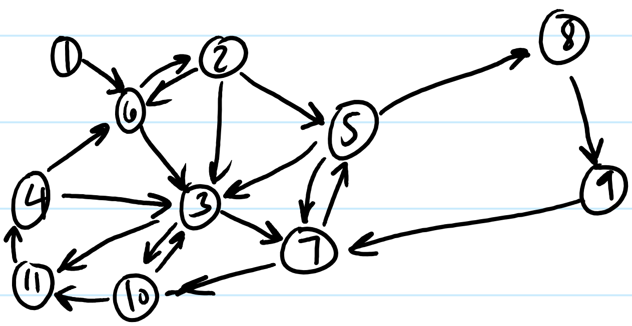
$$(1 I - M)p = 0 \qquad (3)$$

It is not feasible to use standard algebraic methods to solve

Instead, watch this...

```
Init;

D = diag(1./sum(G));
P = G * D;
N = size(P,1);

x = ones(N,1) / N; % uniform distribution
for n = 1:20
    x = P * x;
    plot(x);
    hold on;
    pause(0.5);
end

hold off;
```

Recall...



Goal: To develop the theory behind why the random surfer algorithm works.

Recall:

- We can encode the behaviour of a cohort of random surfers in the Markov matrix
- If p is a probability vector representing the distribution of surfers (or the value of each web page), then no matter what the initial p is, it seems that

$$\lim_{n\to\infty} M^n p \longrightarrow \text{steady-state}$$

i.e. if $p^\infty = \lim_{n\to\infty} M^n p$

$\Rightarrow M p^\infty = p^\infty$ (p^∞ is a fixed point of the mapping M)

How can we know for sure that this is the case?

$\boxed{\text{Thm: Every Markov matrix } Q \text{ has } 1 \text{ as an eigenvalue.}}$

Pf: The matrix Q^T has the same eigenvalues as Q since they have the same characteristic polynomial.

Let $e = [1 1 \cdots 1]^T$.

$Q^T e = e \Rightarrow 1$ is an e-value for Q^T

$\therefore \lambda = 1$ is an e-value for Q. ■

Comment: OK, so now we know that there is a solution to $Mp = p$.
But how do we explain convergence?

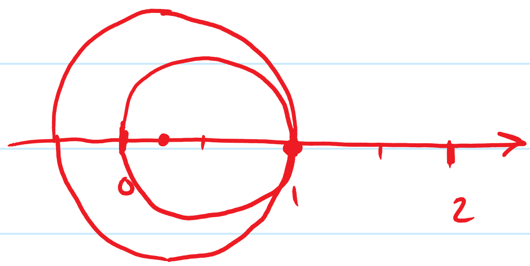
$\boxed{\begin{array}{l}\textbf{Thm: } \text{Every (possibly complex) e-value } \lambda \text{ of a Markov matrix } Q \text{ satisfies} \\ \qquad\qquad |\lambda| \leq 1 .\end{array}}$

Pf: To prove this, we'll use another theorem called the Gershgorin Circle Theorem.

Thm: Girshgorin Circle thm.
Let B be an arbitrary matrix. Then the e-vals. λ of B are located in the union of the n disks

$$|\lambda - b_{kk}| \leq \sum_{j \neq k} |b_{jk}| .$$

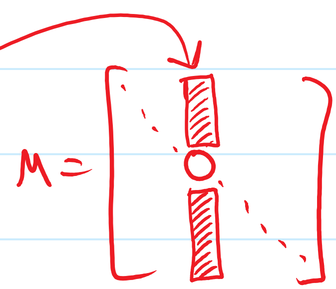
e.g. $B = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{4} \end{bmatrix}$



col 1 $\Rightarrow$ centre $=1$ radius $=0$
col 2 $\Rightarrow$ ctr $=\frac{1}{2}$ radius $=\frac{1}{2}$
col 3 $\Rightarrow$ ctr $=\frac{1}{4}$ radius $=\frac{3}{4}$

The Gershgorin thm. implies

$$|\lambda - m_{kk}| \leq \sum_{j \neq k} |m_{jk}|$$



Since $0 \leq m_{ij} \leq 1$ ...

$$|\lambda| - m_{kk} \leq \sum_{j \neq k} m_{jk}$$

$$\Rightarrow |\lambda| \leq \sum_j m_{jk} = 1 \quad \text{(sum down the } k^{th} \text{ col)}$$

$\blacksquare$

<u>Def'n</u>: A Markov matrix Q is <u>positive</u> if

$$Q_{ij} > 0 \quad \forall i,j \quad \text{(i.e. no zero elements)}$$

<u>Thm</u>: If Q is a positive Markov matrix, then there is only one eigenvector with $|\lambda| = 1$.

<u>Pf</u>: Isn't it obvious?!

$\blacksquare$

Comment: Now we know that there is only <u>one</u> solution to $Mp = p$.

<u>Thm</u>: If M is a positive Markov matrix, then

$$\lim_{n \to \infty} M^n p^0$$

converges to a unique vector p^∞ for any initial probability vector p^0.

<u>Pf</u>: With relatively few assumptions, we can represent p^0 using an eigenvector basis

$$p^0 = c_1 x_1 + \sum_{l=2}^{R} c_l x_l$$

where we order the e-values by decreasing magnitude s.t. $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_R|$. Hence, x_1 is the unique e-vector of $\lambda_1 = 1$.

Notice that

$$Mp^0 = \lambda_1 c_1 x_1 + \sum_{l=2}^{R} \lambda_l c_l x_l$$

$$\vdots$$

$$(M)^n p^0 = \lambda_1^n c_1 x_1 + \sum_{l=2}^{R} \lambda_l^n c_l x_l$$

and that $\lambda_1^n = 1$, but $\lambda_l^n \to 0$ for $l \neq 1$.

This is because $|\lambda_l| < 1$ for $l \neq 1$.

Thus,

$$\lim_{n\to\infty} (M)^n p^0 = c_1 x_1 = p^\infty .$$

$\blacksquare$

Interpretation:

No matter what the initial distribution, simple iteration converges to the steady-state. Moreover, the rate of convergence depends on the second-largest eigenvalue; if λ_2 is close to 1, then convergence is slow. This is because the convergence results from waiting for the other eigen-components $\{\lambda_2, \ldots \lambda_R\}$ to decay down close to zero.

<u>Implementation</u>

To find the Page Rank "importance" score for each web page, we can start with some probability vector p^0 and iterate,

$$M^n p^0 \approx p^\infty \quad \text{for sufficiently large } n$$

Recall:

$$M = \alpha P' + (1-\alpha)\frac{1}{R} e e^T$$

$$= \alpha\left(P + \frac{1}{R} e d^T\right) + (1-\alpha)\frac{1}{R} e e^T$$

$$M = \alpha P' + (1-\alpha)\frac{1}{R} e e^T$$

$$= \alpha\left(P + \frac{1}{R} e d^T\right) + (1-\alpha)\frac{1}{R} e e^T$$

So, $$Mp = \alpha\left(P + \frac{1}{R} e d^T\right)p + (1-\alpha)\frac{1}{R} e \underbrace{e^T p}_{=1}$$

$d^T p$ is a scalar, $O(R)$ to compute

$[1\,1 \cdots 1]p = \sum_i p_i$

$$= \alpha P p + \frac{\alpha}{R} e (d^T p) + \frac{1-\alpha}{R} e$$

Since P is sparse, it can be stored and applied in $O(R)$.
Thus, Mp takes $O(R)$ flops.

Goal: To learn how to efficiently solve triangular linear systems.

Solving Triangular Systems

A triangular matrix has zeros either above the diagonal, or below the diagonal.

eg. $\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix}$ is ... $\begin{bmatrix} a & 0 & 0 & 0 \\ b & c & 0 & 0 \\ d & e & f & 0 \\ g & h & i & j \end{bmatrix}$ is...

$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$ is...

Triangular matrices occur in certain matrix factorizations, and are a useful type of matrix.

Solving Upper-Triangular Systems: Back Substitution

$Ux = z$ U is upper-Δ matrix

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & & u_{1N} \\ 0 & u_{22} & \cdots & & u_{2N} \\ \vdots & & \ddots & & \\ \vdots & & & u_{N-1,N-1} & u_{N-1,N} \\ 0 & \cdots & & 0 & u_{N,N} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ \vdots \\ z_N \end{bmatrix}$$

Start with the last row

$$u_{N,N}\, x_N = z_N \implies x_N = \frac{z_N}{u_{N,N}}$$

Now it's easy to solve for x_{N-1} in the second-last row.

$$u_{N-1,N-1}\, x_{N-1} + u_{N-1,N}\, x_N = z_{N-1}$$

$$\Rightarrow \quad x_{N-1} = \frac{z_{N-1} - u_{N-1,N}\, x_N}{u_{N-1,N-1}}$$

The i-th row...

$$u_{ii}\, x_i + u_{i,i+1}\, x_{i+1} + \cdots + u_{iN}\, x_N = z_i$$

$$\Rightarrow \quad x_i = \frac{z_i - \left(u_{i,i+1}\, x_{i+1} + \cdots + u_{iN}\, x_N\right)}{u_{ii}}$$

$$x_i = \frac{z_i - \sum_{j=i+1}^{N} u_{ij}\, x_j}{u_{ii}}$$

<u>Back Substitution Algorithm</u> (a.k.a. Back Solve)
(see page 101 in the course notes)

<u>Complexity</u>
For each i, the j-loop performs $2(N-i)$ flops (floating-point operations). Together with $\div\, u_{ii}$ $\Rightarrow$ flops $= 2(N-i)+1$

Sum over i

$$\text{Total flops} = \sum_{i=1}^{N} \left(2(N-i)+1\right) = \sum_{i=1}^{N} 2N - 2i + 1$$

$$= 2N^2 + N - 2\sum_{i=1}^{N} i$$

$$= 2N^2 + N - \frac{\cancel{2}\, N(N+1)}{\cancel{2}}$$

$$= 2N^2 + N - \frac{\cancel{2} N(N+1)}{\cancel{2}}$$

$$= 2N^2 + \cancel{N} - N^2 - \cancel{N}$$

$$= \boxed{N^2}$$

Forward Substitution

$Lx = z$ L is $N \times N$ lower-Δ

$$\begin{bmatrix} l_{11} & 0 & \cdots & & \\ l_{21} & l_{22} & 0 & \cdots & \\ \vdots & & & & \\ l_{N1} & l_{N2} & \cdots & & l_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}$$

From 1^{st} row,

$$\Rightarrow x_1 = \frac{z_1}{l_{11}}$$

i^{th} row

$$l_{i1} x_1 + l_{i2} x_2 + \cdots + l_{i,i-1} x_{i-1} + l_{ii} x_i = z_i$$

$$\Rightarrow x_i = \frac{z_i - (l_{i1} x_1 + \cdots + l_{i,i-1} x_{i-1})}{l_{ii}}$$

$$x_i = \frac{z_i - \sum_{j=1}^{i-1} l_{ij} x_j}{l_{ii}}$$

Gaussian Elimination

To solve a system of linear equations

$$Ax = b \qquad \text{eg.} \begin{bmatrix} 1 & 1 & 2 \\ -1 & -2 & 3 \\ 3 & -7 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 1 \\ 10 \end{bmatrix}$$

one can use Gaussian elimination.

1) Form the augmented matrix.

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 8 \\ -1 & -2 & 3 & 1 \\ 3 & -7 & 4 & 10 \end{array}\right]$$

2) Perform linear row operations to get an upper-Δ form

$$\begin{array}{r} \\ ② + ① \\ ③ - 3① \end{array} \left[\begin{array}{ccc|c} 1 & 1 & 2 & 8 \\ 0 & -1 & 5 & 9 \\ 0 & -10 & -2 & -14 \end{array}\right] \Rightarrow \begin{array}{r} \\ \\ ③ - 10② \end{array} \left[\begin{array}{ccc|c} 1 & 1 & 2 & 8 \\ 0 & -1 & 5 & 9 \\ 0 & 0 & -52 & -104 \end{array}\right] \Rightarrow \begin{array}{r} \\ -② \\ \frac{-1}{52}③ \end{array} \left[\begin{array}{ccc|c} 1 & 1 & 2 & 8 \\ 0 & 1 & -5 & -9 \\ 0 & 0 & 1 & 2 \end{array}\right]$$

3) You might have been taught to follow this with more row operations to get a diagonal matrix.

$$\Rightarrow \cdots \Rightarrow \left[\begin{array}{ccc|c} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2 \end{array}\right] \quad \therefore \text{solution is } \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

A better way is to use back substitution after step 2.

Solve

$$\begin{bmatrix} 1 & 1 & 2 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ -9 \\ 2 \end{bmatrix}$$

Goal: To review how row operations can be used to reduce and solve a linear system.

LU Factorization

Any square matrix A can be factored into a product of an upper-triangular and lower-triangular matrices such that

$$LU = PA$$

P is a permutation matrix used to swap rows.

LU factorization is essentially the same as Gaussian elimination.

pivot element

new pivot element

e.g. $\begin{bmatrix} 2 & 3 & -1 \\ 4 & 6 & -1 \\ -2 & 2 & -3 \end{bmatrix} \Rightarrow \begin{matrix} ② - \frac{4}{2}① \\ ③ + \frac{+2}{2}① \end{matrix} \begin{bmatrix} 2 & 3 & -1 \\ \boxed{0} & 0 & 1 \\ \boxed{0} & 5 & -4 \end{bmatrix} \Rightarrow \ldots$

In this case, there is no way to use the pivot element to get rid of the 5. So, swap rows 2 and 3.

$$\begin{bmatrix} 2 & 3 & -1 \\ 0 & 5 & -4 \\ 0 & 0 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{\text{permutation matrix}} \begin{bmatrix} 2 & 3 & -1 \\ 0 & 0 & 1 \\ 0 & 5 & -4 \end{bmatrix}$$

We will go over the details later, but for now I'll simply state that LU factorization takes $\mathcal{O}(N^3)$ flops.

Applications of LU Factorization

a) Solving $Ax = b$

$$Ax = b \Rightarrow PAx = Pb \Rightarrow L\underbrace{Ux}_{z} = Pb$$

Two steps to compute x

1) Solve $Lz = Pb$ for z (N^2)
2) Solve $Ux = z$ for x (N^2)

Gaussian elimination = $\boxed{\text{LU factorization}}$ + $\boxed{\text{forward sub + back sub}}$

$\mathcal{O}(N^3)$ $\mathcal{O}(N^2)$

b) Solving $AX = B$

Suppose X and B each have M columns.

$$A\left[x_1 \middle| \cdots \middle| x_M\right] = \left[b_1 \middle| \cdots \middle| b_M\right]$$

1) Factor: $LU = PA$ $\mathcal{O}(N^3)$
2) Solve: $\left.\begin{array}{l} LUx_1 = Pb_1 \\ \vdots \\ LUx_M = Pb_M \end{array}\right\}$ $\mathcal{O}(N^2)$ each

Take-home message: Do the expensive LU factorization once, and use it for each of the M systems.

Gaussian Elimination (GE)

2×2 example

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 & (1) \\ a_{21}x_1 + a_{22}x_2 = b_2 & (2) \end{cases}$$

$$(2) - \frac{a_{21}}{a_{11}}(1) \Rightarrow \left(a_{21} - \frac{a_{21}}{a_{11}}a_{11}\right)x_1 + \left(a_{22} - \frac{a_{21}}{a_{11}}a_{12}\right)x_2$$

$$② - \frac{a_{21}}{a_{11}} ① \Rightarrow \underbrace{\left(a_{21} - \frac{a_{21}}{a_{11}} a_{11}\right)}_{0} x_1 + \underbrace{\left(a_{22} - \frac{a_{21}}{a_{11}} a_{12}\right)}_{a_{22}^{(1)}} x_2 = \underbrace{b_2 - \frac{a_{21}}{a_{11}} b_1}_{b_2^{(1)}}$$

In matrix form:
$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$② - \frac{a_{21}}{a_{11}} ① \Rightarrow \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} \\ 0 & a_{22}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \end{bmatrix}$$

For general N , the big picture of GE...

$$\underset{A}{\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix}} \rightarrow \underset{A^{(1)}}{\begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{bmatrix}} \rightarrow \underset{A^{(2)}}{\begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{bmatrix}} \rightarrow \underset{A^{(3)}}{\begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix}}$$

GE Version 1:

for i = 1:N-1

eliminate x_i from rows i+1 to N

end

At the i-th stage of GE...

$$\begin{array}{r} \\ \text{pivot row } i \rightarrow \\ \\ \text{current row } k \rightarrow \end{array} \begin{bmatrix} x & x & x & x \\ 0 & a_{ii} & x & x \\ 0 & 0 & x & x \\ 0 & a_{ki} & x & x \end{bmatrix} \xrightarrow{ⓚ - \frac{a_{ki}}{a_{ii}} ⓘ} \begin{bmatrix} x & x & x & x \\ 0 & a_{ii} & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{bmatrix}$$

$k = i+1, \ldots, N$

GE Version 2:

for i = 1:N-1

for k = i+1:N

mult = a_{ki}/a_{ii}

row(k) = row(k) - mult*row(i)

end

end

To update the entire row k:

$$\begin{array}{cl} \begin{bmatrix} 0 & \cdots & 0 & a_{ii} & \longleftarrow & a_{ij} & \longrightarrow \\ 0 & \cdots & 0 & a_{ki} & \longleftarrow & a_{kj} & \longrightarrow \end{bmatrix} & \begin{array}{l} \text{pivot row } i \\ \text{current row } k \end{array} \\ \begin{array}{ccccccc} & & & \underset{j=i}{\uparrow} & & & \underset{j=N}{\uparrow} \end{array} & \end{array}$$

for j = i+1:N

$a_{kj} = a_{kj}$ - mult*a_{ij}

end

(Note: $a_{ki} = 0$ by design)

Final GE Algorithm:

for i = 1:N-1

for k = i+1:N

```
            mult = a_ki/a_ii
            for j = i+1:N
                a_kj = a_kj - mult*a_ij
            end
            a_ki = 0
        end

    end
```

Notes: 1) The lower-triangular part is all 0.
2) We may use those elements to store the multipliers.

LA-06: LU Factorization

Goal: To find out how to compute LU factorization.

LU Factorization Algorithm

Consider the first step of GE,

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & & \\ a_{N1} & \cdots & a_{NN} \end{bmatrix} \xrightarrow[\text{operations}]{\text{row}} \begin{bmatrix} a_{11}^{(1)} & & & a_{1N}^{(1)} \\ 0 & a_{22}^{(1)} & \cdots & \\ \vdots & \vdots & & \\ 0 & a_{N2}^{(1)} & \cdots & a_{NN}^{(1)} \end{bmatrix} = A^{(1)}$$

Matrix interpretation:

row operations $\Longleftrightarrow$ matrix multiplication

i.e. $M^{(1)} A = A^{(1)}$

where

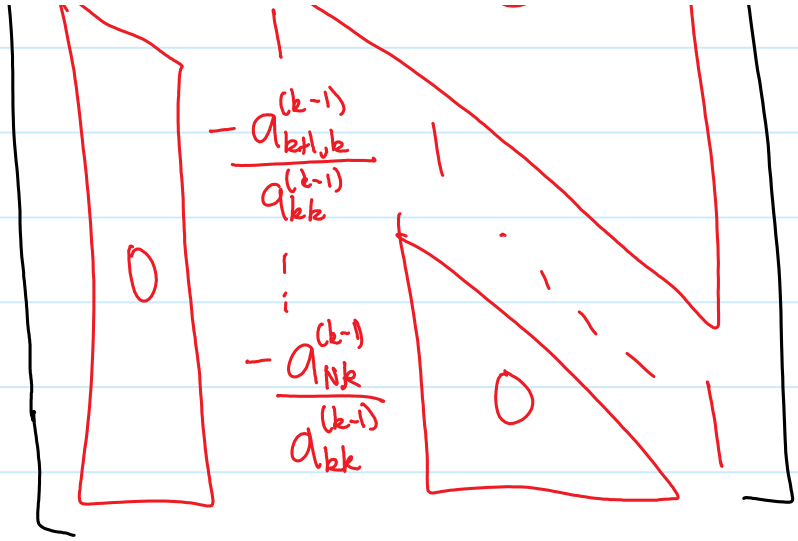
$$M^{(1)} = \begin{bmatrix} 1 & & & \\ -\frac{a_{21}}{a_{11}} & 1 & & 0 \\ \vdots & & \ddots & \\ -\frac{a_{N1}}{a_{11}} & 0 & & 1 \end{bmatrix}$$

In general, at the k-th step...

$$M^{(k)} A^{(k-1)} = A^{(k)}$$

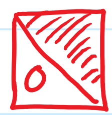
where

$$M^{(k)} = \begin{bmatrix} 1 & & & 0 \\ & \ddots & & \\ & & 1 & \\ & & & \ddots \end{bmatrix}$$



The effect of left-multiplying $A^{(k-1)}$ by $M^{(k)}$ is to eliminate x_k from rows $k+1$ to N.

At the final step: $A^{(N-1)}$ is upper-Δ



Recall that $M^{(k)} A^{(k-1)} = A^{(k)}$

For $k=1 \Rightarrow$ $M^{(1)} A = A^{(1)}$

$$M^{(2)}\left(M^{(1)}A\right) = M^{(2)} A^{(1)} = A^{(2)}$$

$$\vdots$$

$$M^{(N-1)} \cdots M^{(1)} A = A^{(N-1)} \equiv U$$

$$\Rightarrow A = \left[M^{(N-1)} \cdots M^{(1)}\right]^{-1} U$$

<u>Amazing Facts</u>

1) If B and C are lower-Δ and unit diagonal, then so is BC.
2) If B is lower-Δ and unit diagonal, then so is B^{-1}.

By fact (1), $M^{(N-1)} \cdots M^{(1)}$ is lower-Δ unit-diag.

By fact (2), $\left[M^{(N-1)} \cdots M^{(1)}\right]^{-1}$ is lower-Δ unit-diag.

By fact (2), $[M^{(N-1)} \cdots M^{(1)}]^{-1}$ is lower-Δ unit-diag.

Define $L \equiv [M^{(N-1)} \cdots M^{(1)}]^{-1}$

<u>Theorem</u>: $A = LU$

L is lower-Δ & unit diagonal

U is upper-Δ.

<u>Properties of $M^{(k)}$:</u>

1)

$$(M^{(k)})^{-1} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & 0 & \\ & & 1 & & & \\ & & \frac{a_{k+1,k}}{a_{kk}} & \ddots & & \\ & 0 & \vdots & & \ddots & \\ & & \frac{a_{Nk}}{a_{kk}} & 0 & & 1 \end{bmatrix}$$

2) $L = [M^{(N-1)} \cdots M^{(1)}]^{-1} = (M^{(1)})^{-1} \cdots (M^{(N-1)})^{-1}$

$$= \begin{bmatrix} 1 & & & 0 \\ & 1 & & \\ \frac{a_{jk}}{a_{kk}} & & \ddots & \\ = \text{multipliers} & & & 1 \end{bmatrix}$$

i.e. the k^{th} col. of L is the k^{th} col. of $(M^{(k)})^{-1}$.

eg. $\begin{bmatrix} 1 & 0 & 0 \\ \alpha & 1 & 0 \\ \beta & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \gamma & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \alpha & 1 & 0 \\ \beta & \gamma & 1 \end{bmatrix}$

Note that $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \gamma & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ \alpha & 1 & 0 \\ \beta & 0 & 1 \end{bmatrix}$ is not so straight forward.

More precisely, write L_{jk} as

$$L_{jk} = \begin{cases} 1 & \text{if } j=k \\ 0 & \text{if } j<k \\ \frac{a_{jk}}{a_{kk}} & \text{if } j>k \end{cases} \qquad \begin{matrix} k=1,\dots,N \\ j=1,\dots,N \end{matrix}$$

Example:

$$A = \begin{bmatrix} 2 & -1 & 3 \\ -4 & 6 & -5 \\ 6 & 13 & 16 \end{bmatrix} \xrightarrow[\textcircled{3} - \frac{6}{2}\textcircled{1}]{\textcircled{2} - \frac{-4}{2}\textcircled{1}} \begin{bmatrix} 2 & -1 & 3 \\ 0 & 4 & 1 \\ 0 & 16 & 7 \end{bmatrix}$$

$$\xrightarrow{\textcircled{3} - \frac{16}{4}\textcircled{2}} \begin{bmatrix} 2 & -1 & 3 \\ 0 & 4 & 1 \\ 0 & 0 & 3 \end{bmatrix} = U$$

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

Check:

$$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & 4 & 1 \end{bmatrix}\begin{bmatrix} 2 & -1 & 3 \\ 0 & 4 & 1 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 2 & -1 & 3 \\ -4 & 6 & -5 \\ 6 & 13 & 16 \end{bmatrix}$$

Goal: To stabilize LU factorization using row pivoting.

Stability of LU Factorization

In LU factorization, a problem arises when we have:

(1) a zero pivot i.e. $a_{kk}^{(k-1)} = 0$

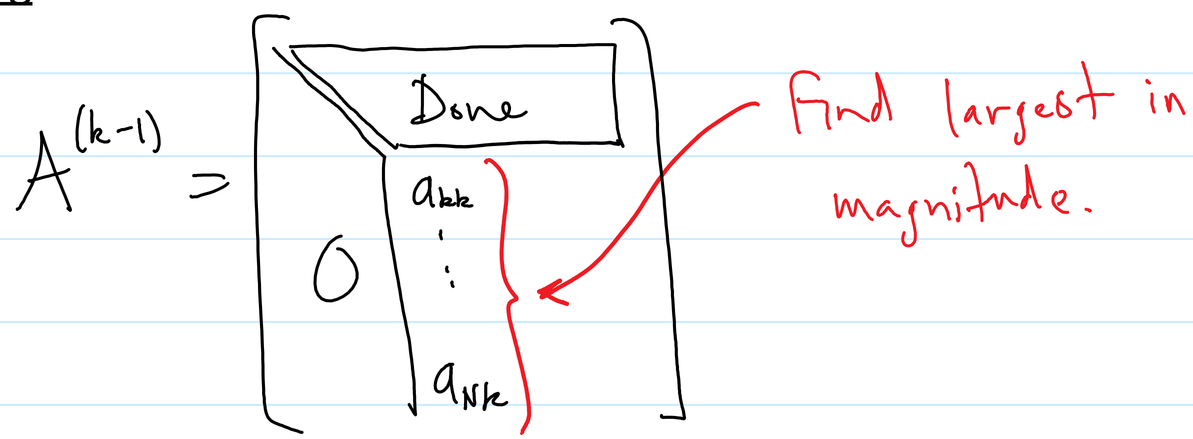
$\Rightarrow$ multipliers $\dfrac{a_{jk}}{a_{kk}}$ are undefined

(2) $a_{kk}^{(k-1)} \approx 0$

$\Rightarrow$ multipliers become large

$\Rightarrow$ calculations become unstable

Pivoting



Find $\displaystyle\max_{k \le j \le N} \left| a_{jk}^{(k-1)} \right| = \left| a_{k^* k}^{(k-1)} \right|$

Swap rows k^* and k, and continue.

Note: $a_{k^* k}^{(k-1)} \neq 0$

Otherwise, $a_{jk}^{(k-1)} = 0 \quad \forall\, k \le j \le N$

$\Rightarrow A$ is singular

As mentioned earlier, these row-swapping operations can be represented by matrix multiplication by a permutation matrix, P.
eg. to swap rows i & j, simply swap rows i & j in the identity matrix.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow[\text{②} \,\&\, \text{③}]{\text{Swap}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = P$$

Let's put it all together.

eg. $$A = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 16 & 64 \\ 2 & 2 & 8 \end{bmatrix}$$

$$\Big\downarrow \text{swap ① \& ②}$$

$$\begin{bmatrix} 4 & 16 & 64 \\ 1 & 1 & 1 \\ 2 & 2 & 8 \end{bmatrix} \qquad P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Big\downarrow$$

$$\begin{array}{r} \\ ② - \frac{1}{4}① \\ ③ - \frac{2}{4}① \end{array} \begin{bmatrix} 4 & 16 & 64 \\ 0 & -3 & -15 \\ 0 & -6 & -24 \end{bmatrix} \qquad L = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & & 1 \end{bmatrix}$$

$$\Big\downarrow \text{swap ② \& ③}$$

$$\begin{bmatrix} 4 & 16 & 64 \end{bmatrix} \qquad P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\downarrow \text{②} \,\&\, \text{③}$$

$$\begin{bmatrix} 4 & 16 & 64 \\ 0 & -6 & -24 \\ 0 & -3 & -15 \end{bmatrix} \qquad P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

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

$$\downarrow$$

$$③ - \frac{-3}{-6}② \begin{bmatrix} 4 & 16 & 64 \\ 0 & -6 & -24 \\ 0 & 0 & -3 \end{bmatrix}$$

$$\therefore \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 4 & 16 & 64 \\ 0 & -6 & -24 \\ 0 & 0 & -3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 4 & 16 & 64 \\ 2 & 2 & 8 \end{bmatrix}$$

$$L \quad U \quad = \quad P \quad A$$

Example:

$$A = \begin{bmatrix} -6 & 27 & -42 & -15 \\ -24 & 12 & 24 & -12 \\ -12 & 14 & -10 & 20 \\ -6 & -9 & 42 & 15 \end{bmatrix}$$

$$\Big\downarrow \text{ swap ① \& ②}$$

$$\begin{bmatrix} -24 & 12 & 24 & -12 \end{bmatrix} \qquad P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -24 & 12 & 24 & -12 \\ -6 & 27 & -42 & -15 \\ -12 & 14 & -10 & 20 \\ -6 & -9 & 42 & 15 \end{bmatrix} \qquad P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\downarrow$$

$$\begin{array}{r} \\ ② - \frac{1}{4} ① \\ ③ - \frac{1}{2} ① \\ ④ - \frac{1}{4} ① \end{array} \begin{bmatrix} -24 & 12 & 24 & -12 \\ \frac{1}{4} & 24 & -48 & -12 \\ \frac{1}{2} & 8 & -22 & 26 \\ \frac{1}{4} & -12 & 36 & 18 \end{bmatrix}$$

$$\downarrow$$

$$\begin{array}{r} \\ \\ ③ - \frac{1}{3} ② \\ \\ \end{array} \begin{bmatrix} -24 & 12 & 24 & -12 \\ \frac{1}{4} & 24 & -48 & -12 \\ \frac{1}{2} & \frac{1}{3} & -6 & 30 \\ \frac{1}{4} & -\frac{1}{2} & 12 & 12 \end{bmatrix} \qquad P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Big\downarrow \text{ swap ③ \& ④}$$

$$\begin{bmatrix} -24 & 12 & 24 & -12 \\ \frac{1}{4} & 24 & -48 & -12 \\ \frac{1}{4} & -\frac{1}{2} & 12 & 12 \\ \frac{1}{2} & \frac{1}{3} & -6 & 30 \end{bmatrix} \qquad P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} -24 & 12 & 24 & -12 \\ \frac{1}{4} & 24 & -48 & -12 \\ \frac{1}{4} & -\frac{1}{2} & 12 & 12 \\ \frac{1}{2} & \frac{1}{3} & -\frac{1}{2} & 36 \end{bmatrix}$$

$$\therefore \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -6 & 27 & -42 & -15 \\ -24 & 12 & 24 & -12 \\ -12 & 14 & -10 & 20 \\ -6 & -9 & 42 & 15 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{4} & 1 & 0 & 0 \\ \frac{1}{4} & -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & \frac{1}{3} & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} -24 & 12 & 24 & -12 \\ 0 & 24 & -48 & -12 \\ 0 & 0 & 12 & 12 \\ 0 & 0 & 0 & 36 \end{bmatrix}$$