```
>> Lecture 5
>>
>>          -- Special Topic: Text Processing
>>
```

(Most) Common Codec: ASCII[2]

- Everything in the computer is encoded in binary.
- ASCII is a character-encoding scheme originally based on the English alphabet that encodes 128 specified characters into the 7-bit binary integers (see the next page).
- Unicode[1] became a standard for the modern systems from 2007.
  - Unicode is backward compatible with ASCII because ASCII is a subset of Unicode.

[1] See Unicode 8.0 Character Code Charts.

[2] Codec: coder-decoder; ASCII: American Standard Code for Information Interchange, also see http://zh.wikipedia.org/wiki/ASCII.

Hex	Dec	Char		Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char
0x00	0	NULL	null	0x20	32	Space	0x40	64	@	0x60	96	`
0x01	1	SOH	Start of heading	0x21	33	!	0x41	65	A	0x61	97	a
0x02	2	STX	Start of text	0x22	34	"	0x42	66	B	0x62	98	b
0x03	3	ETX	End of text	0x23	35	#	0x43	67	C	0x63	99	c
0x04	4	EOT	End of transmission	0x24	36	$	0x44	68	D	0x64	100	d
0x05	5	ENQ	Enquiry	0x25	37	%	0x45	69	E	0x65	101	e
0x06	6	ACK	Acknowledge	0x26	38	&	0x46	70	F	0x66	102	f
0x07	7	BELL	Bell	0x27	39	'	0x47	71	G	0x67	103	g
0x08	8	BS	Backspace	0x28	40	(	0x48	72	H	0x68	104	h
0x09	9	TAB	Horizontal tab	0x29	41	)	0x49	73	I	0x69	105	i
0x0A	10	LF	New line	0x2A	42	*	0x4A	74	J	0x6A	106	j
0x0B	11	VT	Vertical tab	0x2B	43	+	0x4B	75	K	0x6B	107	k
0x0C	12	FF	Form Feed	0x2C	44	,	0x4C	76	L	0x6C	108	l
0x0D	13	CR	Carriage return	0x2D	45	-	0x4D	77	M	0x6D	109	m
0x0E	14	SO	Shift out	0x2E	46	.	0x4E	78	N	0x6E	110	n
0x0F	15	SI	Shift in	0x2F	47	/	0x4F	79	O	0x6F	111	o
0x10	16	DLE	Data link escape	0x30	48	0	0x50	80	P	0x70	112	p
0x11	17	DC1	Device control 1	0x31	49	1	0x51	81	Q	0x71	113	q
0x12	18	DC2	Device control 2	0x32	50	2	0x52	82	R	0x72	114	r
0x13	19	DC3	Device control 3	0x33	51	3	0x53	83	S	0x73	115	s
0x14	20	DC4	Device control 4	0x34	52	4	0x54	84	T	0x74	116	t
0x15	21	NAK	Negative ack	0x35	53	5	0x55	85	U	0x75	117	u
0x16	22	SYN	Synchronous idle	0x36	54	6	0x56	86	V	0x76	118	v
0x17	23	ETB	End transmission block	0x37	55	7	0x57	87	W	0x77	119	w
0x18	24	CAN	Cancel	0x38	56	8	0x58	88	X	0x78	120	x
0x19	25	EM	End of medium	0x39	57	9	0x59	89	Y	0x79	121	y
0x1A	26	SUB	Substitute	0x3A	58	:	0x5A	90	Z	0x7A	122	z
0x1B	27	FSC	Escape	0x3B	59	;	0x5B	91	[	0x7B	123	{
0x1C	28	FS	File separator	0x3C	60	<	0x5C	92	\	0x7C	124	|
0x1D	29	GS	Group separator	0x3D	61	=	0x5D	93	]	0x7D	125	}
0x1E	30	RS	Record separator	0x3E	62	>	0x5E	94	^	0x7E	126	~
0x1F	31	US	Unit separator	0x3F	63	?	0x5F	95	_	0x7F	127	DEL

Characters and Strings (Revisited)

- Before R2017a, a text is a sequence of characters, just like numeric arrays.
    - For example, 'ntu'.
- Most built-in functions can be applied to string arrays.

```
clear; clc;

s1 = 'ntu'; s2 = 'csie';
s = {s1, s2};
upper(s) % output: {'NTU', 'CSIE'}
```

- Since R2017a, you can create a string by enclosing a piece of text in double quotes.[3]
  - For example, "ntu".
- You can find a big difference between characters and strings in this example:

```
clear; clc;

s1 = 'ntu'; s2 = 'NTU';
s1 + s2 % output: 188 200 202

s3 = string(s1); s4 = string(s2);
s3 + s4 % output: "ntuNTU"
```

[3] See https://www.mathworks.com/help/matlab/ref/string.html.

Selected Text Operations[4]

sprintf	Format data into string.
strcat	Concatenate strings horizontally.
contains	Determine if pattern is in string.
count	Count occurrences of pattern in string.
endsWith	Determine if string ends with pattern.
startsWith	Determine if string starts with pattern.
strfind	Find one string within another.
replace	Find and replace substrings in string array.
split	Split strings in string array.
strjoin	Join text in array.
lower	Convert string to lowercase.
upper	Convert string to uppercase.
reverse	Reverse order of characters in string.

[4]See `https://www.mathworks.com/help/matlab/characters-and-strings.html`.

Introduction to Regular Expressions[5]

- A regular expression, also called a pattern, is an expression used to specify a set of strings required for a particular purpose.
  - Check this: `https://regexone.com`.

[5] See `https://en.wikipedia.org/wiki/Regular_expression`; also `https://www.mathworks.com/help/matlab/matlab_prog/regular-expressions.html`.

Example

```
>> text = 'bat cat can car coat court CUT ct ...
    CAT-scan';
>> pattern = 'c[aeiou]+t';
>> start_idx = regexp(text, pattern)

start_idx =

       5    17
```

- The pattern 'c[aeiou]+t' indicates a set of strings:
  - c must be the first character;
  - c must be followed by one of the characters in the brackets [aeiou], followed by t as the last character;
  - in particular, [aeiou] must occur one or more times, as indicated by the + operator.

Metacharacters[6]

Operator	Definition
|	Boolean OR.
*	0 or more times consecutively.
?	0 times or 1 time.
+	1 or more times consecutively.
{n}	exactly n times consecutively.
{m, }	at least m times consecutively.
{, n}	at most n times consecutively.
{m, n}	at least m times, but no more than n times consecutively.

[6] See https://www.mathworks.com/help/matlab/ref/regexp.html.

Operator	Definition
.	any single character, including white space.
$[c_1c_2c_3]$	any character contained within the brackets.
$[\wedge c_1c_2c_3]$	any character not contained within the brackets.
$[c_1\text{-}c_2]$	any character in the range of c_1 through c_2.
\s	any white-space character.
\w	a word; any alphabetic, numeric, or underscore character.
\W	not a word.
\d	any numeric digit; equivalent to [0-9].
\D	no numeric digit; equivalent to $[\wedge 0\text{-}9]$.

Output Keywords

Keyword	Output
'start'	starting indices of all matches, by default
'end'	ending indices of all matches
'match'	text of each substring that matches the pattern
'tokens'	text of each captured token
'split'	text of nonmatching substrings
'names'	name and text of each named token

Examples

```
clear; clc;

text1 = {'Madrid, Spain', 'Romeo and Juliet', ...
    'MATLAB is great'};
tokens = regexp(text1, '\s', 'split')

text2 = 'EXTRA! The regexp function helps you ...
    relax.';
matches = regexp(text2, '\w*x\w*', 'match')
```

Exercise: Listing Filtered Files

```
clear; clc;

file_list = dir;
filenames = {file_list(:).name};
A = regexp(filenames, '.+\.m', 'match');
mask = cellfun(@(x) ~isempty(x), A);
cellfun(@(f) fprintf('%s\\%s\n', pwd, f{:}), A(mask))
```

Example: By Names

- You can associate names with tokens so that they are more easily identifiable.
- For example,

```
>> str = 'Here is a date: 01-Apr-2020';
>> expr = '(?<day>\d+)-(?<month>\w+)-(?<year>\d+)';
>> mydate = regexp(str, expr, 'names')

mydate =

      day: '01'
    month: 'Apr'
     year: '2020'
```

Exercise: Web Crawler

- Write a script which collects the names of html tags by defining a token within a regular expression.
- For example,

```
>> str = '<title>My Title</title><p>Here is some ...
    text.</p>';
>> pattern = '<(\w+).*>.*</\1>';
>> [tokens, matches] = regexp(str, pattern, ...
    'tokens', 'match')
```

More Regexp Functions

- See **regexpi**, **regexprep**, and **regexptranslate**.

```
>> Lecture 6
>>
>>      -- Special Topic: File Operations & other I/O
>>
```

Spreadsheets: Excel/CSV Files (Revisited)

- The command **xlsread**(*filename*) reads excel files, for example,

```
[~, ~, raw] = xlsread("2330.xlsx");
```

- By default, it returns a numeric matrix.
- The text part is the 2nd output, separated from the numeric part.
- You may consider the whole spreadsheet by using the 3rd output (stored in a cell array).
- Note that you can use $\sim$ to drop the output value.

More Tips for Excel Files

- You can specify the range.
  - For example, the string argument `"B:B"` is used to import column B.
  - If you need a single value, say the cell B1, just use `"B1:B1"`.[7]
- You could specify the worksheet by the sheet name[8] or the sheet number.
- You could refer to the document for more details.[9]

---

[7] Contribution by Mr. Tsung-Yu Hsieh (MAT24409) on August 27, 2014.
[8] The default sheet name is "工作表".
[9] See https://www.mathworks.com/help/matlab/ref/xlsread.html.

Mat Files[10]

- Recall that I/O is costly.
- To save time, you may consider **save** matrices to the disk; for example,

```
data1 = rand(1, 10);
data2 = ones(10);
save('trial.mat', 'data1', 'data2');
```

- You can use **load** to fetch the data from mat files.

```
load('trial.mat');
```

[10] See https://www.mathworks.com/help/matlab/ref/save.html.

Selected Read/Write Functions

- For text data, see https://www.mathworks.com/help/matlab/text-files.html.
  - Try **dlmread**, **dlmwrite**, **csvread**, **csvwrite**, **textread**/**textscan**.
- For images, see https://www.mathworks.com/help/matlab/images_images.html.
- For video and audio, see https://www.mathworks.com/help/matlab/audio-and-video.html.

Selected File Operations[11]

cd	Change current folder.
pwd	Identify current folder.
ls	List folder contents by chars.
dir	List folder contents by structures.
exist	Check existence of variable, script, function, folder, or class.
mkdir	Make new folder.
visdiff	Compare two files or folders.

[11]See `https://www.mathworks.com/help/matlab/file-operations.html`.

Example: Pooling Data from Multiple Files[12]

```
clear; clc;

cd('./stocks'); % enter the folder
files = dir; % get all files in the current folder
files = files(3 : end); % drop the first two
names = {files(:).name}; % get all file names
filter = endsWith(names, '.xlsx'); % filter by .xlsx
names = names(filter);

pool = cell(length(names), 2);
for i = 1 : length(names)
    [~, ~, raw] = xlsread(names{i});
    pool(i, :) = {names{i}(1 : 4), raw};
end
save('data_pool', 'pool');
```

[12] Download stocks.zip.

```
>> Lecture 7
>>
>>              -- Matrix Computation
>>
```

Vectors

- Let $\mathbb{R}$ be the set of all real numbers.
- $\mathbb{R}^n$ denotes the vector space of all m-by-1 column vectors:

$$u = (u_i) = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}. \tag{1}$$

  - You can simply use the colon (:) operator to reshape any array in a column major, say $u(:)$.
- Similarly, the row vector v is

$$v = (v_i) = \begin{bmatrix} v_1 \cdots v_n \end{bmatrix}. \tag{2}$$

- We consider column vectors unless stated.

Matrices

- $\mathrm{M}_{m \times n}(\mathbb{R})$ denotes the vector space of all m-by-n real matrices, for example,

$$A = (a_{ij}) = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}.$$

- Complex vectors/matrices[13] follow similar definitions and operations introduced later, simply with some care.

[13] Matlab treats a complex number as a single value.

Transposition

```
>> A = [1 i];
>> A' % Hermitian operator; see any textbook for ...
     linear algebra

ans =

    1.0000 + 0.0000i
    0.0000 - 1.0000i

>> A.' % transposition of A

ans =

    1.0000 + 0.0000i
    0.0000 + 1.0000i
```

Arithmetic Operations

- Let a_{ij} and b_{ij} be the elements of the matrices A and $B \in \mathrm{M}^{m \times n}(\mathbb{R})$ for $1 \leq i \leq m$ and $1 \leq j \leq n$.
- Then $C = A \pm B$ can be calculated by $c_{ij} = a_{ij} \pm b_{ij}$. (Try.)

Inner Product[14]

- Let $u, v \in \mathbb{R}^m$.
- Then the inner product, denoted by $u \cdot v$, is calculated by

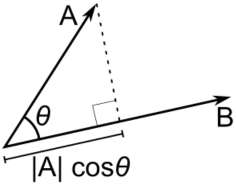
$$u \cdot v = u'v = [u_1 \cdots u_m] \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix}.$$

```
clear; clc;

u = [1; 2; 3];
v = [4; 5; 6];
u' * v    % normal way; orientation is important
dot(u, v) % using the built-in function
```

[14] Akaa dot product and scalar product.

- Inner product is also called projection for emphasizing its geometric significance.



- Recall that we know

$$u \cdot v = 0$$

if and only if these two are orthogonal to each other, denoted by

$$u \perp v.$$

Generalization of Inner Product

- Let $x \in \mathbb{R}$, $f(x)$ and $g(x)$ be real-valued functions.
- In particular, assume that $g(x)$ is a basis function.[15]
- Then we can define the inner product of f and g on $[a, b]$ by

$$\langle f, g \rangle = \int_a^b f(x)g(x)dx.$$

[15] See https://en.wikipedia.org/wiki/Basis_function, https://en.wikipedia.org/wiki/Eigenfunction, and https://en.wikipedia.org/wiki/Approximation_theory.

- For example, Fourier transform is widely used in engineering and science.
  - Fourier integral[16] is defined as

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

    where $f(t)$ is a square-integrable function.
  - The Fast Fourier transform (FFT) algorithm computes the discrete Fourier transform (DFT) in $O(n \log n)$ time.[17,18]

[16] See https://en.wikipedia.org/wiki/Fourier_transform.
[17] Cooley and Tukey (1965).
[18] See https://en.wikipedia.org/wiki/Fast_Fourier_transform.

Matrix Multiplication

- Let $A \in \mathrm{M}_{m \times q}(\mathbb{R})$ and $B \in \mathrm{M}_{q \times n}(\mathbb{R})$.
- Then $C = AB$ is given by

$$c_{ij} = \sum_{k=1}^{q} a_{ik} \times b_{kj}. \tag{3}$$

- For example,

$$\overset{4\times 2 \text{ matrix}}{\begin{bmatrix} a_{11} & a_{12} \\ \cdot & \cdot \\ a_{31} & a_{32} \\ \cdot & \cdot \end{bmatrix}} \overset{2\times 3 \text{ matrix}}{\begin{bmatrix} \cdot & b_{12} & b_{13} \\ \cdot & b_{22} & b_{23} \end{bmatrix}} = \overset{4\times 3 \text{ matrix}}{\begin{bmatrix} \cdot & x_{12} & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & x_{33} \\ \cdot & \cdot & \cdot \end{bmatrix}}$$

Example

```
clear; clc;

A = randi(10, 5, 4); % 5-by-4
B = randi(10, 4, 3); % 4-by-3
C = zeros(size(A, 1), size(B, 2));
for i = 1 :  size(A, 1)
    for j = 1 : size(B, 2)
        for k = 1 : size(A, 2)
            C(i, j) = C(i, j) + A(i, k) * B(k, j);
        end
    end
end
C % display C
```

- Time complexity: $O(n^3)$.
- Strassen (1969): $O(n^{log_2 7})$.

Matrix Exponentiation

- Raising a matrix to a power is equivalent to repeatedly multiplying the matrix by itself.
  - For example, $A^2 = AA$.
- The matrix exponential[19] is a matrix function on square matrices analogous to the ordinary exponential function, more explicitly,

$$e^A = \sum_{n=0}^{\infty} \frac{A^n}{n!}.$$

- However, it is not allowed to perform A^B.

[19] See matrix exponentials and Pauli matrices.

Determinants

- Consider the matrix

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

- Then $\det(A) = ad - bc$ is called the determinant of A.
  - The method of determinant calculation in high school is a wrong way but produces correct answers for all 3×3 matrices.
- Let's try the minor expansion formula for det(A).[20]

[20] See http://en.wikipedia.org/wiki/Determinant.

Recursive Algorithm for Minor Expansion Formula

```
function y = myDet(A)

    [r, ~] = size(A);

    if r == 1
        y = A;
    elseif r == 2
        y = A(1, 1) * A(2, 2) - A(1, 2) * A(2, 1);
    else
        y = 0;
        for i = 1 : r
            B = A(2 : r, [1 : i - 1, i + 1 : r]);
            cofactor = (-1) ^ (i + 1) * myDet(B);
            y = y + A(1, i) * cofactor;
        end
    end
end
```

- It needs $n!$ terms in the sum of products, so this algorithm runs in $O(n!)$ time!
- Use **det** for determinants, which can be done in $O(n^3)$ time by using LU decomposition or alike.[21]

[21] See https://en.wikipedia.org/wiki/LU_decomposition. Moreover, various decompositions are used to implement efficient matrix algorithms in numerical analysis. See https://en.wikipedia.org/wiki/Matrix_decomposition.

Linear Systems (Transformation/Mapping)[22]

- A linear system is a mathematical model of a system based on linear operators satisfying the property of superposition.
  - For simplicity, $Ax = y$ for any input x associated with the output y.
  - Then A is a linear operator if and only if

$$A(ax_1 + bx_2) = aAx_1 + bAx_2 = ay_1 + by_2$$

    for $a, b \in \mathbb{R}$.
  - For example, $\frac{d(x^2+3x)}{dx} = \frac{dx^2}{dx} + 3\frac{dx}{dx} = 2x + 3$.
- Linear systems typically exhibit features and properties that are much simpler than the nonlinear case.
  - What about nonlinear cases?

[22] See https://en.wikipedia.org/wiki/Linear_system.

First-Order Approximation: Local Linearization

- Let $f(x)$ be any nonlinear function.
- Assume that $f(x)$ is infinitely differentiable at x_0.
- By Taylor's expansion[23], we have

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + O\left((x - x_0)^2\right),$$

where $O\left((x - x_0)^2\right)$ is the collection of higher-order terms, which can be neglected as $x - x_0 \to 0$.

- Then we have a first-order approximation

$$f(x) \approx f'(x_0)x + k,$$

with $k = f(x_0) - x_0 f'(x_0)$, a constant.

[23] See https://en.wikipedia.org/wiki/Taylor_series.

Two Observations

- We barely feel like the curvature of the ground; however, we look at Earth on the moon and agree that Earth is a sphere.
- Newton's kinetic energy is a low-speed approximation (classical limit) to Einstein's total energy.
  - Let m be the rest mass and v be the velocity relative to the inertial coordinate.
  - The resulting total energy is

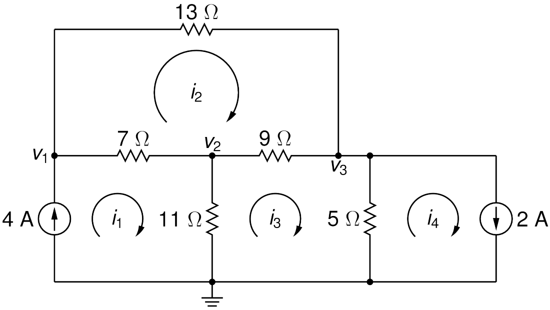
$$E = \frac{mc^2}{\sqrt{1 - (v/c)^2}}.$$

  - By applying the first-order approximation,

$$E \approx mc^2 + \frac{1}{2}mv^2.$$

Example: Kirchhoff's Laws[24]

- The algebraic sum of currents in a network of conductors meeting at a point is zero.
- The directed sum of the potential differences (voltages) around any closed loop is zero.



[24]See https://en.wikipedia.org/wiki/Kirchhoff's_circuit_laws.

General Form of Linear Equations[25]

- Let n be the number of unknowns and m be the number of constraints.
- A general system of m linear equations with n unknowns is

$$\left\{\begin{array}{cccccc} a_{11}x_1 & +a_{12}x_2 & \cdots & +a_{1n}x_n & = & y_1 \\ a_{21}x_1 & +a_{22}x_2 & \cdots & +a_{2n}x_n & = & y_2 \\ \vdots & \vdots & \ddots & \vdots & = & \vdots \\ a_{m1}x_1 & +a_{m2}x_2 & \cdots & +a_{mn}x_n & = & y_m \end{array}\right.$$

where $x_1, \ldots, x_n$ are unknowns, $a_{11}, \ldots, a_{mn}$ are the coefficients of the system, and $y_1, \ldots, y_m$ are the constant terms.

[25] See https://en.wikipedia.org/wiki/System_of_linear_equations.

Matrix Equation

- Hence we can rewrite the aforesaid equations as follows:

$$Ax = y.$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix},$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \text{ and } y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}.$$

- Finally, x can be done by $x = A^{-1}y$, where A^{-1} is called the inverse of A.

Inverse Matrices[26]

- For simplicity, let $A \in \mathrm{M}_{n \times n}(\mathbb{R})$ and $x, y \in \mathbb{R}^n$.
- Then A is called invertible if there exists $B \in \mathrm{M}_{n \times n}(\mathbb{R})$ such that

$$AB = BA = I_n,$$

  where I_n denotes a $n \times n$ identity matrix.
  - We use A^{-1} to denote the inverse of A.
  - You can use **eye**(n) to generate an identity matrix I_n.
- Use **inv**(A) to calculate the inverse of A.

[26] See `https://en.wikipedia.org/wiki/Invertible_matrix#The_invertible_matrix_theorem`.

- However, **inv**(A) may return a weird result even if A is ill-conditioned, indicates how much the output value of the function can change for a small change in the input argument.[27]
- For example, calculate the inverse of the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}.$$

  - Recall the Cramer's rule[28]: A is invertible iff $\mathbf{det}(A) \neq 0$. (Try.)
- If these constraints cannot be eliminated by row reduction, they are linearly independent.

[27] You may refer to the condition number of a function with respect to an argument. Also try **rcond**.

[28] See `https://en.wikipedia.org/wiki/Cramer's_rule`.

Linear Independence

- Let $K = \{a_1, a_2, \ldots, a_n\}$ for each $a_i \in \mathbb{R}^m$.
- Now consider this linear superposition

$$x_1 a_1 + x_2 a_2 + \cdots + x_n a_n = 0,$$

where $x_1, x_2, \ldots, x_n \in \mathbb{R}$ are the weights.

- Then K is linearly independent iff

$$x_1 = x_2 = \cdots = x_n = 0.$$

Example: $\mathbb{R}^3$

- Let
$$K_1 = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}.$$
- It is clear that K_1 is linearly independent.
- Moreover, you can represent all vectors in $\mathbb{R}^3$ if you collect all linear superpositions from K_1.
- We call this new set a span of K_1, denoted by $\text{Span}(K_1)$.[29]
- Clearly, $\text{Span}(K_1) = \mathbb{R}^3$.

[29] See https://en.wikipedia.org/wiki/Linear_span.

- Now let

$$K_2 = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \right\}.$$

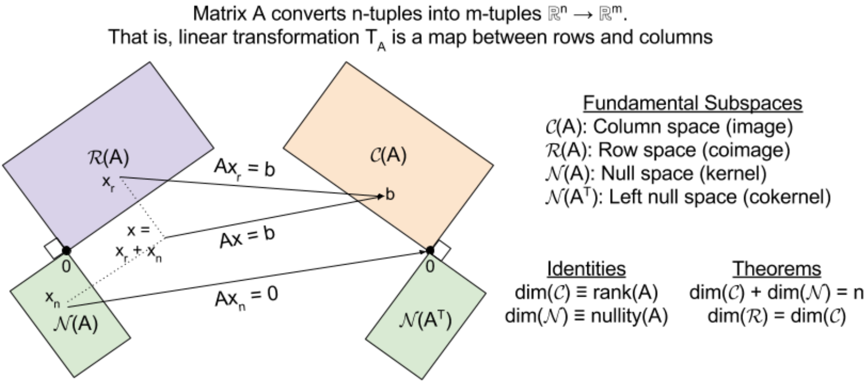
- Then K_2 is not a linearly independent set. (Why?)
- If you take one or more vectors out of K_2, then K_2 becomes linearly independent.

Basis of Vector Space & Its Dimension[30]

- However, you can take only one vector out of K_2 if you want to represent all vectors in $\mathbb{R}^3$. (Why?)
  - The dimension of $\mathbb{R}^3$ is exactly the size (element number) of K_2.
- We say that the basis of $\mathbb{R}^n$ is a maximally linearly independent set of size n.
- Note that the basis of $\mathbb{R}^3$ is not unique.
  - For example, K_1 could be also a basis of $\mathbb{R}^3$.

[30] See https://en.wikipedia.org/wiki/Basis_(linear_algebra), https://en.wikipedia.org/wiki/Vector_space, and https://en.wikipedia.org/wiki/Dimension_(vector_space).

Linear Transformation (Revisited)[31]



[31] See `https://en.wikipedia.org/wiki/Linear_map`; also see `https://kevinbinz.com/2017/02/20/linear-algebra/`.

Example: Vector Projection ($\mathbb{R}^3 \to \mathbb{R}^2$)

- Let $u \in \mathbb{R}^3$ and $v \in \mathbb{R}^2$.
- We consider the projection matrix (operator),

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

  so that $Au = v$.
- For example,

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

Solution Set to System of Linear Equations[32]

- Recall that m is the number of constraints and n is the number of unknowns.
- Now consider the following cases.
- If $m = n$, then there exists a unique solution.
- If $m > n$, then it is called an overdetermined system and there is no solution.
  - Fortunately, we can find a least-squares error solution such that $\| Ax - y \|^2$ is minimal, shown later.
- If $m < n$, then it is called a underdetermined system which has infinitely many solutions.
  - Become an optimization problem?
- For all cases,

$$x = A \setminus y.$$

[32]See https://www.mathworks.com/help/matlab/ref/mldivide.html.