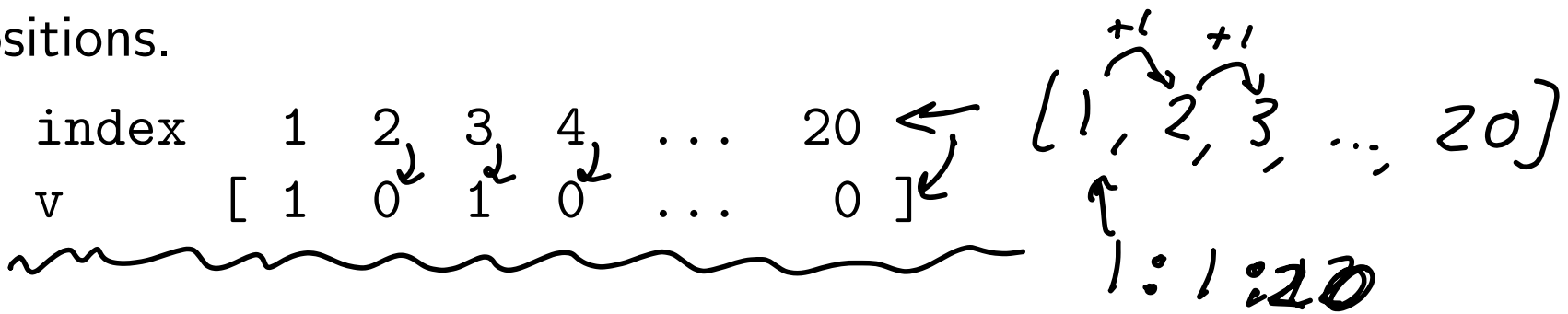
Lecture 2: Vectors and Matrices

Math 98

Vectors: Warm-up

use "mod"

1. Create a length 20 vector with 1s in odd positions and 0s in even positions.



2. Now put the 1s in even positions and 0s in odd positions.
3. ~~Put 1s at multiples of 3 and 0s otherwise.~~

Vectors: Arithmetic

We can do arithmetic with vectors and scalars, or with pairs of vectors. When working with pairs of vectors the symbols $*, /, \hat{}$ are reserved for matrix operations while element-wise operations use $.*, ./, .\hat{}$ instead.

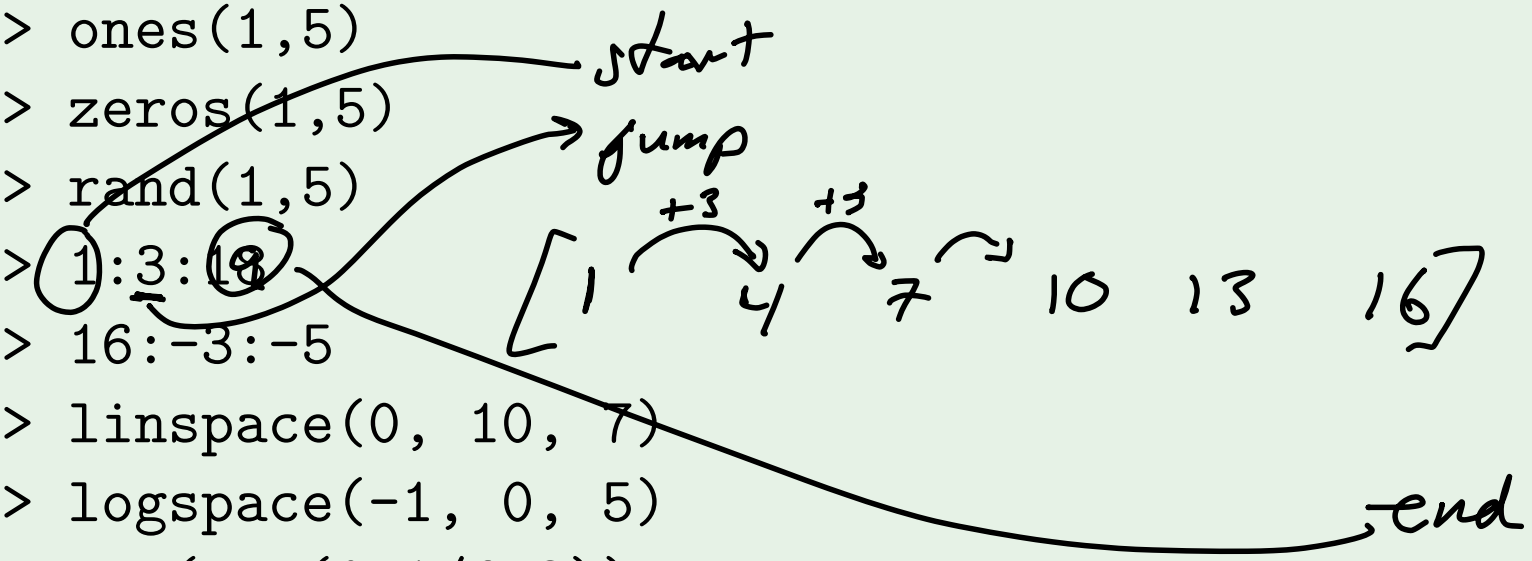
```
>> v + 2
ans =
    3   7   0
>> v.^2
ans =
    1   25   4
```

[1 5 -2] ``*'' [1 5 -2]

Vectors: Creating Them

If the elements of the vector follow a pattern then we may have some quick commands that define them. Try:

```
>> ones(1,5)
>> zeros(1,5)
>> rand(1,5)
>> 1:3:18
>> 16:-3:-5
>> linspace(0, 10, 7)
>> logspace(-1, 0, 5)
>> cos(pi*(0:1/6:2))
```



1:10 ⇝ 1:1:10

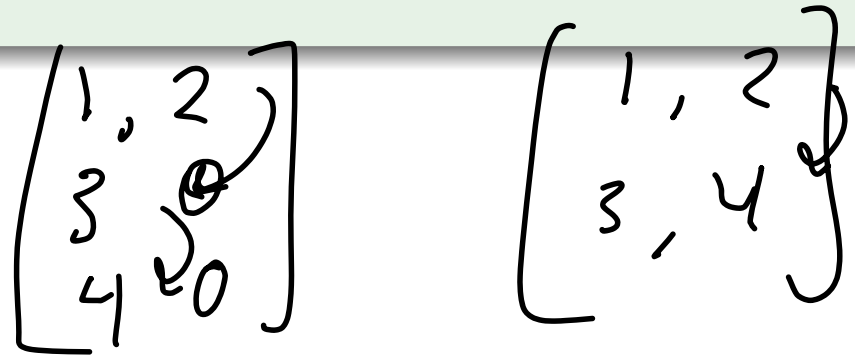
Vectors: Accessing Elements

Use the notation `v(i)` to look up the i-th element of the vector v. We can also change individual elements of `v` this way.

```
>> v = [1,3,5,8,10]; v(3)
    5
>> v(3:5)
    5   8   10
>> v(5:-1:1)                % same as fliplr(v)
    10   8   5   3   1
>> w = [1,4,2]; v(w)  % can you solve this one?
    1   8   3
>> v = [v, 6]
    1   3   5   8   10   6
>> v(3) = 2; v(9) = 7          % pad with zeros
    1   3   2   8   10   6   0   0   7
```

Vectors: Vector Operations

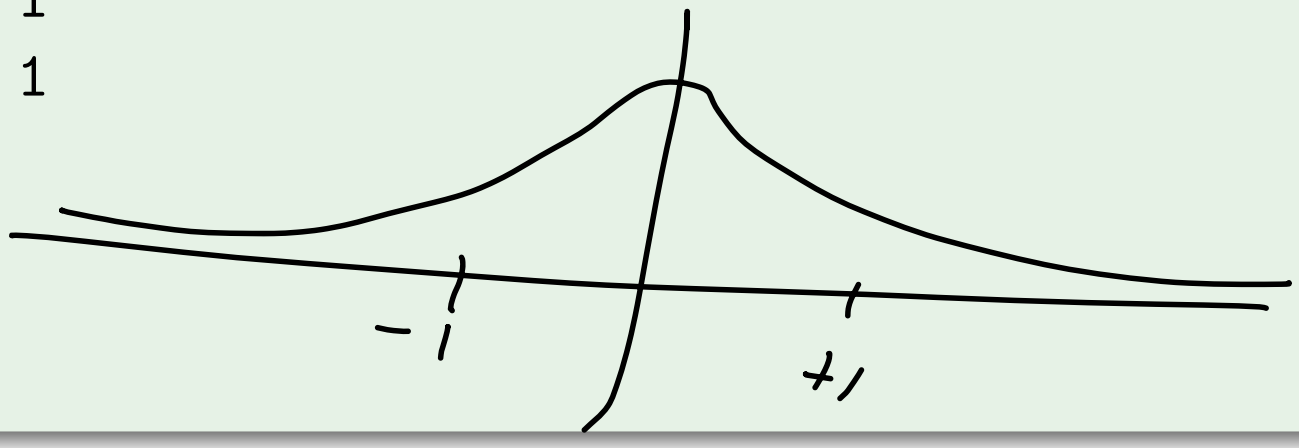
```
>> w = [1; 3; 5; 7]; w = w';
    1   3   5   7
>> w(2:end)
    3   5   7
>> w(end+1) = 500
    1   3   5   7   500
>> x = [1, 2; 3, 4]
Dimensions of arrays being concatenated are not consistent.
```



MATLAB = Matrix Laboratory

Matrices: like vectors, but with more dimensions?

```
>> ones(2, 3)
     1     1     1
     1     1     1
>> ones(2)
     1     1
     1     1
```



Try also:

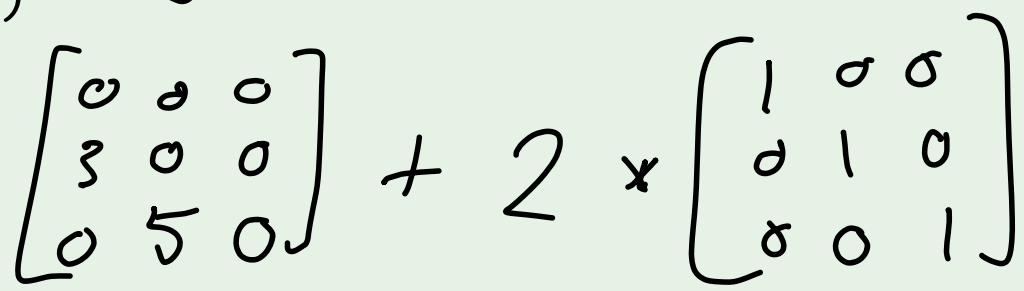
- `zeros(m,n)`
- `rand(m,n)`
- `randn(m,n)`

What happens if you enter only `m` and leave out `n`? Read the documentation to confirm.

Matrices: Special Ways to Create Them

Other commands to create special types of matrices:

```
>> eye(2)
     1     0
     0     1
>> v = [3,5]; diag(v)
     3     0
     0     5
>> diag(v,-1) + 2*eye(3)
     2     0     0
     3     2     0
     0     5     2
```



Matrices: Build Your Own

To type a a matrix into MATLAB you must:

- Begin with a square bracket, [
- Separate Elements in a row with spaces or commas
- Use a semicolon ; to seperate rows
- End with another square bracket,]

```
>> A = [1, 4, 6; 3, 6, 8; 0, 2, 6]
A =
     1     4     6
     3     6     8
     0     2     6
>> B = [1:5; 1, 3:4, 5, 6]
B =
     1     2     3     4     5
     1     3     4     5     6
```

2×5

Matrices: Accessing Entries

Accessing entries of a matrix

```
>> A(2, 3) %row, column
ans =
     8
>> A(6)
ans =
     2
```

What does the second result suggest to you about how the entries in a matrix are indexed?

Matrices: Accessing Columns, Rows, and Submatrices

Accessing rows and columns of a matrix

```
>> A(:, 2)
ans =
     4
     6
     2
>> A(3, :)
ans =
     0   2   6
```

A(1:end, 2)

A(1:3, 2)

Accessing submatrices

```
>> A(2:3, 2:3)
ans =
     6   8
     2   6
```

Matrices: Matrix Operations

As with vectors, we have lots of ways to access and manipulate the entries of matrices. Try out the following commands:

```
>> A = reshape(1:12,3,4)
>> size(A)  % also size(A,1) and size(A,2)
>> A'
>> fliplr(A) % also flipud(A)
>> A(:,[1, 3])
>> p = randperm(3); q = randperm(4); A(p,q)
>> reshape(A,2,6)
>> A(3, 3) = 1
>> A(:, 2) = A(:, 1)
```

Type `help [your command here]` to learn more!

Matrices: Solving $Ax = b$

$$x = A^{-1}b$$

Two ways to solve $Ax = b$:

```
>> x = inv(A)*b
>> x = A\b
```

$$[A \mid I] \sim [I \mid A^{-1}] \qquad [A \mid b] \sim [I \mid A^{-1}b]$$

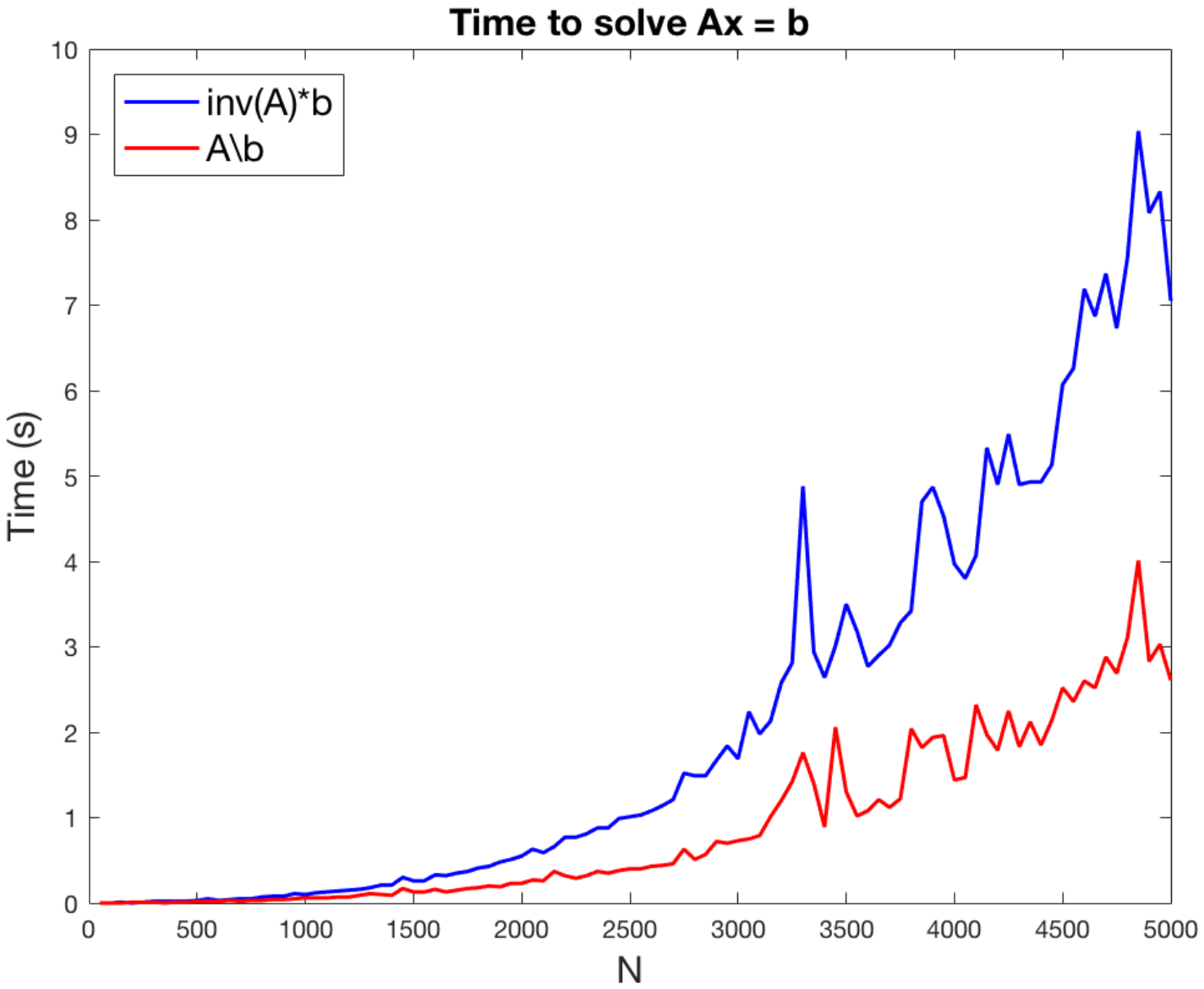
Which one should we use?

$$A^{-1} * b$$

```
P * inv(P) =
    1.0000    -0.0000
    0.0000     1.0000

P \ P =
    1.0000
         0    1.0000
```

Matrices: Solving $Ax = b$: Graph



Why Don't These Work?

It is also equally (if not more) informative to study things that don't work, since much of your time will be dealing with bugs and errors.

```
>> A = [1, 3, 5; 1, 2];
>> A = [1, 3, 5]; B = [1, 3]; A + B
>> A = eye(3); A(0)
>> A = [1, 3, 5]; A^2
```

Relations

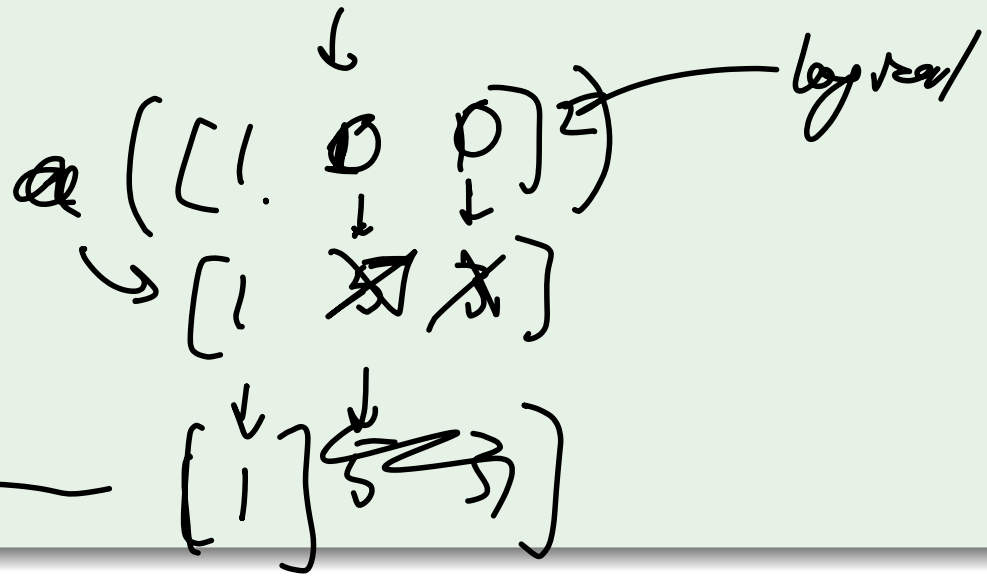
The following statements will take value 0 (if false) or 1 (if true)

- $a < b$: a less than b
- $a > b$: a greater than b
- $a <= b$: a less than or equal to b
- $a >= b$: a greater than or equal to b
- $a == b$: a equal to b (note the doubled equals sign!)
- $a \sim= b$: a not equal to b

Relations: Vectors and Matrices

You can also utilize a relation on a vector, as well as to index a vector at certain indices:

```
>> a = [1,5,3];
>> a <= 3
ans =
     1   0   1
>> b = a(a <= 3)
b =
     1   3
```



The last command selects all elements in the vector a that are less than or equal to 3.

Scripts

Okay, it's finally time to make a proper script! In the command line, enter

```
>> edit Hello.m
```

Or New → Script (Top Left)

- An m-file is a file containing a script for MATLAB—a sequence of instructions for the computer.
- The name of the file must have the format `filename.m`.
  - Filenames are case sensitive (like everything in MATLAB).
  - `filename.m` and `Filename.m` are different!
- For MATLAB to execute the file, it must be saved in the Current Directory.

Two ways to run a script:

- Open up the script and hit "Run".
- Type the name of the script into the Command Window.

input

Sometimes we want to be able to request input from the user.

```
>> input('Please input a starting value: ')
Please input a starting value:
```

Let's give it the number 3.

```
>> input('Please input a starting value: ')
Please input a starting value: 3
ans =
     3
```

We should probably assign the input to a value.

```
>> myval = input('Please input a starting value: ')
Please input a starting value: 3
myval =
```

Commenting with %

Except within a string, everything following % is a comment. Comments do not get executed when the program runs, but can make the code easier to read by providing information about its organization and usage.

Comments in the beginning lines of a program will be revealed when using the command `help`.

`help` (as well as `doc`) is also invaluable when learning how to use various MATLAB functions.

Exercise: `conversion.m`

A temperature can be converted from Fahrenheit to Celsius using the formula $c = (5/9)(f - 32)$, where c is Celsius and f is Fahrenheit. Write a script that prompts the user for a temperature in Fahrenheit, converts it to Celsius, and prints it.

↳ use "disp"

part 2

convert from C to F

exercise) create a script that takes values "m" and "n" from user & spits out an m-by-n checkerboard matrix