Day 32

- Computer representation of integers
- Integer equality checks succeed
- Representation of floating point numbers
- Floating point equality checks fail
- Finding floating point values
- Max and min
- (Exercises)

Computer Representation of Integers

Integers are stored exactly in the computer.

Uses binary

Sequence of 1's and 0's (on and off)

0b0001 = 1 in decimal

0b0010 = 2 in decimal

0b0011 = 3 in decimal

Computer Representation of Integers

You don’t need to know how binary works, just that it is exact for integers.

You will learn more about this later in your ME curriculum.

Integer Equality Checks Succeed

Because integers are stored exactly, checking for equality works fine.

```
clc
clear variables
degrees=[0:10:90];
for i=1:length(degrees)
    if degrees(i)==40
        fprintf('The %1.0fth entry is 40.\n',i);
    end
end
```

Command Window

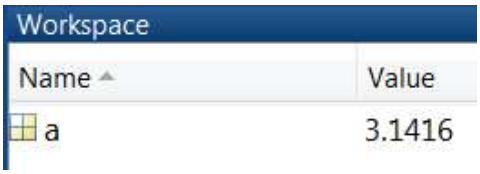
```
The 5th entry is 40.
```

Representation of Floating Point Numbers

Floating point numbers are not stored exactly.

```
a=2/3;
```

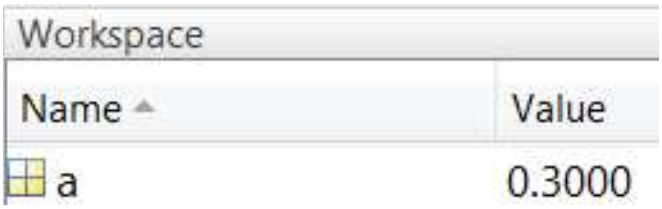

```
a=pi;
```



Representation of Floating Point Numbers

Even simple floating point numbers aren't stored exactly, even though they might appear to be.

```
a=3/10;
```

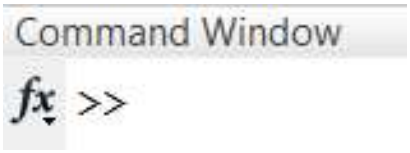


Floating Point Equality Checks Fail

The equality check with floating point doesn't work:

```
clc
clear variables
fractions=[0:0.1:1];
for i=1:length(fractions)
    if fractions(i) == 3/10
        fprintf('The %1.0fth entry is 0.3.\n',i);
    end
end
```

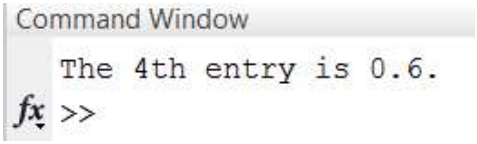
Nothing prints!



Floating Point Equality Checks Fail

Actually, the equality check works occasionally, depending on the number:

```
clc
clear variables
fractions=[0:0.2:1];
for i=1:length(fractions)
    if fractions(i) == 6/10
        fprintf('The %1.0fth entry is 0.6.\n',i);
    end
end
```



Finding Floating Point Values

We use a tolerance (range) to locate floating point values:

```
clc
clear variables
fractions=[0:0.1:1];
tol=1.0e-09;
findit=3/10;
for i=1:length(fractions)
    if (fractions(i) < (findit+tol)) && (fractions(i) > (findit-tol))
        fprintf('The %1.0fth entry is 0.3.\n',i);
    end
end
```

Small tolerance

Value to find

Command Window

```
The 4th entry is 0.3.
>>
```


Finding Floating Point Values

This code is a bit simpler but does the same thing:

```
clc
clear variables
fractions=[0:0.1:1];
tol=1.0e-09;
findit=3/10;
for i=1:length(fractions)
    if abs(fractions(i)-findit) < tol
        fprintf('The %1.0fth entry is 0.3.\n',i);
    end
end
```

Finding Floating Point Values

The `find` command is the most elegant solution. Check entire vector at once! No loop!

```
clc
clear variables
fractions=[0:0.1:1];
tol=1.0e-09;
findit=3/10;
i=find(abs(fractions-findit)<tol);
fprintf('The %1.0fth entry is 0.3.\n',i);
```


Max and Min

Review special case: to locate the index of the largest or smallest entry in a vector, use `max` or `min`.

```
clc
clear variables
x=[-2:0.1:2];
y=1-2*x+x.*x+x.*x.*x;
[miny,min_index]=min(y);
fprintf('y has a minimum value of %6.4f at x=%4.2f \n',miny,x(min_index))
```

Index for minimum y used to find the x that goes with that y.

```
Command Window
  y has a minimum value of 0.3750 at x=0.50
fx >>
```