

# CE 311K: Control flow - Branching and Iterations

Krishna Kumar

University of Texas at Austin

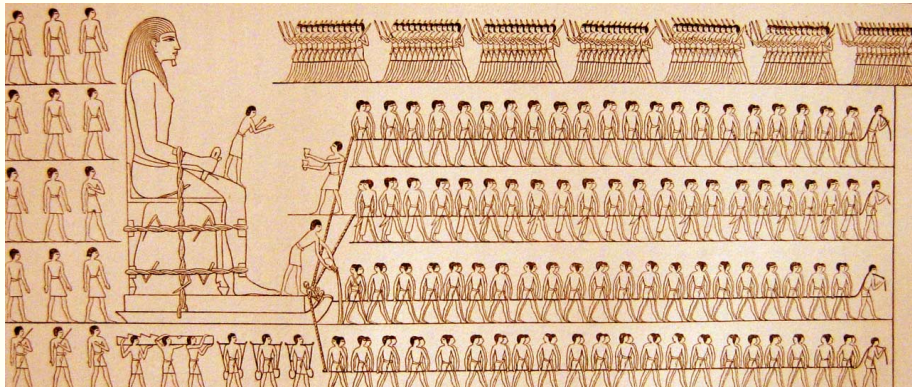
krishnak@utexas.edu

Fall 2020

1 Numerical solution of a sliding block

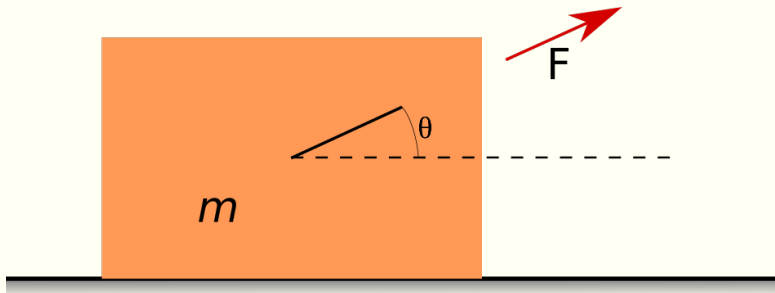
2 Bisection method

# What is the optimal angle to pull the statue?



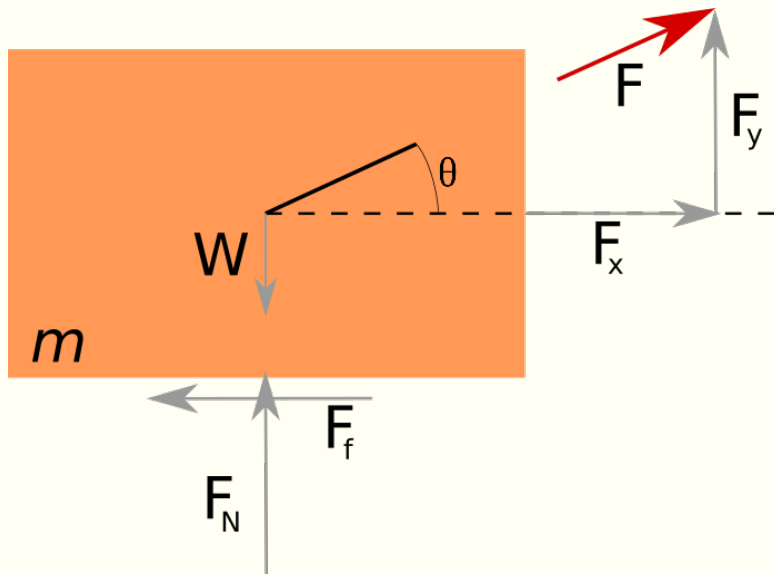
A wall painting from the tomb of Djehutihotep (credit: martinhumanities.com)

# Numerical solution of a sliding block: Approximation



What is the optimal angle to pull the block applying the least amount of force?

# Numerical solution of a sliding block: Forces



# Numerical solution of a sliding block: Forces

$$F_x = F \cos \theta \quad \& \quad F_y = F \sin \theta$$

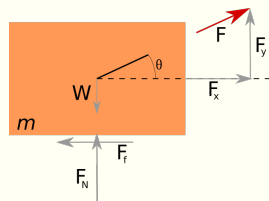
$$F_f = \mu \cdot F_N = \mu \cdot W - \mu F_y = \mu mg - \mu F \sin \theta$$

$$\text{Vertical forces } \sum F_{\text{vert}} \uparrow: F_y + F_N - W = 0$$

$$F_N = mg - F \sin \theta$$

$$\text{Horizontal forces } \sum F_{\text{hor}} \rightarrow: F_x - F_f = 0$$

$$F \cos \theta - \mu mg + \mu F \sin \theta = 0$$



$$F = \frac{\mu \cdot mg}{(\cos \theta + \mu \sin \theta)}$$

# Numerical solution of a sliding block: Compute force

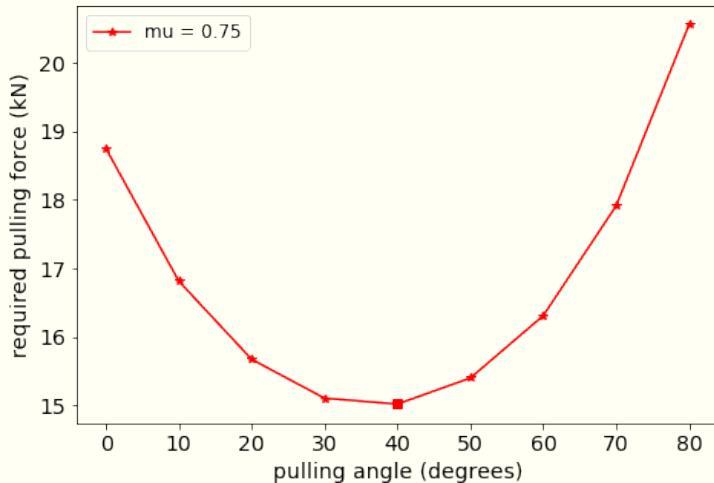
- Given  $W = 25\text{ kN}$  (2500 kg),  $\theta = 45^\circ$  and  $\mu = 0.75$  ( $35^\circ$ ):

$$F = \frac{0.75 \times 25}{\cos(45) + 0.75 \sin(45)} = 15.15 \text{ kN}.$$

- Given  $W = 25\text{ kN}$  (2500 kg) and  $\mu = 0.75$ , what's the optimum  $\theta$ ?

## Numerical solution of a sliding block: Optimal theta?

Given  $W = 25 \text{ kN}$  (2500 kg) and  $\mu = 0.75$ , what's the optimum  $\theta$ ?





# Lists

- A list is a sequence of data. (mutable)
- An 'array' in most other languages is a similar concept, but Python lists are more general than most arrays as they can hold a mixture of types.
- A list is constructed using square brackets:

```
>>> a = [0, 10, 20, 30, 40, 50, 60, 70, 80]
>>> print(a)
[0, 10, 20, 30, 40, 50, 60, 70, 80]
>>> type(a)
<class 'list'>
>>> len(a)
10
>>> a.append(90)
>>> print(a)
[0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
```

# Iterating through a list: for loops

Looping over each item in a list (or more generally a sequence) is called 'iterating'. We iterate over the members of the lab group using the syntax:

```
for each item in list do
    print(item)
```

```
for item in list:
    print(item)
```

 **Indentation matters in python!**

## CE 311K: Control flow

- └ Numerical solution of a sliding block
  - └ Iterating through a list: for loops

Looping over each item in a list (or more generally a sequence) is called 'iterating'. We iterate over the members of the lab group using the syntax:

```
for each item in list do  
    print(item)
```

```
for item in list:  
    print(item)
```

▲ Indentation matters in python!

Variables defined inside a for loop is not accessible outside.

# np.arange()

We need a new library called 'numpy', we are going to import the new library. `import numpy as np` The `np.arange()` returns a sequence of numbers:

`range(stop)`

*stop*: Number of integers (whole numbers) to generate, starting from zero.  
eg. `np.arange(3)` yields a sequence of `[0, 1, 2]`.

`np.arange([start], stop[, step])`

- *start*: Starting number of the sequence.
- *stop*: Generate numbers up to, but not including this number.
- *step*: Difference between each number in the sequence.

## CE 311K: Control flow

## └ Numerical solution of a sliding block

## └ np.arange()

## np.arange()

We need a new library called 'numpy', we are going to import the new library. import numpy as np The np.arange() returns a sequence of numbers.

```
range(5)
```

stop: Number of integers (whole numbers) to generate, starting from zero. eg. np.arange(3) yields a sequence of [0, 1, 2].

```
np.arange([start], stop[, step])
```

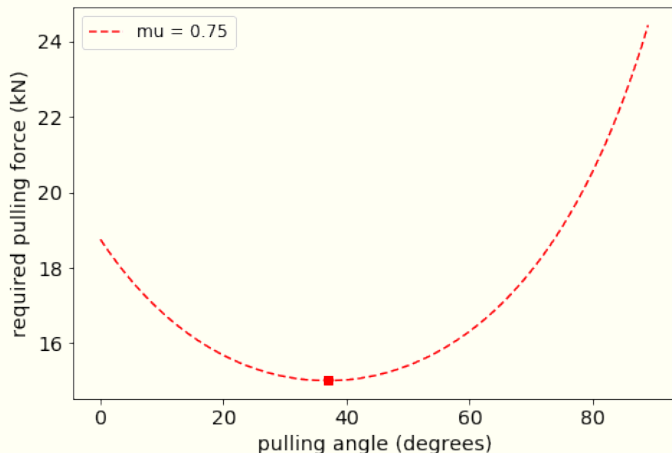
- start: Starting number of the sequence.
- stop: Generate numbers up to, but not including this number.
- step: Difference between each number in the sequence.

Note that:

- All parameters can be positive or negative.
- range() (and Python in general) is 0-index based, meaning list indexes start at 0, not 1

# Numerical solution of a sliding block: Optimal theta?

Given  $W = 25 \text{ kN}(2500 \text{ kg})$  and  $\mu = 0.75$ , what's the optimum  $\theta$ ?



② Identifying optimum requires conditional statements

# Comparison on int, float and strings

`i` and `j` are variable names and comparisons below evaluate to a Boolean

- `i > j`
- `i >= j`
- `i < j`
- `i <= j`
- `i == j`: equality test, True if `i` is the same as `j`
- `i != j`: in equality test, True if `i` is not the same as `j`

# Logic operators on bools

a and b are variable names with Boolean values

- not a: True if a is False  
False if a is True
- a and b: True if both are True.
- a or b: True if either or both are True.

---

A	B	A and B	A or B
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

---



# Designing a smart window: if condition



- An electric window opener, attached to a rain sensor and a temperature gauge, might be controlled by the following program:
- If raining: close window
- If too hot (80F): open window
- If too cold (66F): close window
- Otherwise: do nothing and leave window as it is

# Designing a smart window: if condition

```
# If raining, close the window
if raining:
    close_window()

# If the temperature is over 80 F, open window
elif temperature > 80: # else if
    open_window()

# If the temperature is below 66 F, close window
elif temperature < 66:
    close_window()

# Otherwise, do nothing and leave window as it is
else:
    continue
```

## CE 311K: Control flow

└ Numerical solution of a sliding block

└ Designing a smart window: if condition

Designing a smart window: if condition

```
# If raining, close the window
if raining:
    close_window()

# If the temperature is over 80 F, open window
elif temperature > 80: # else if
    open_window()

# If the temperature is below 66 F, close window
elif temperature < 66:
    close_window()

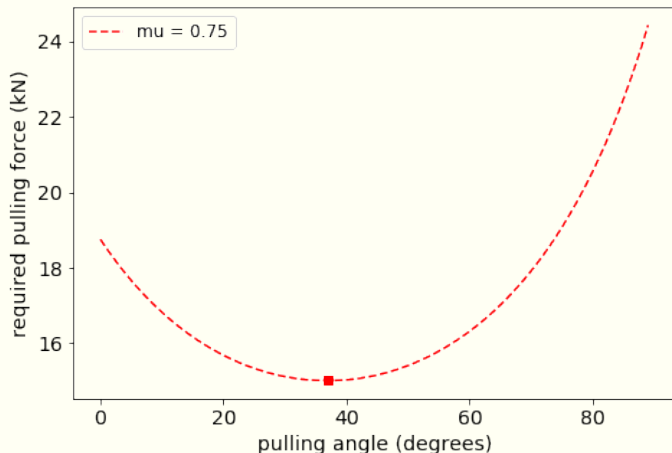
# Otherwise, do nothing and leave window as it is
else:
    continue
```

<condition> has a value True or False

evaluate expressions in that block if <condition> is True

# Numerical solution of a sliding block: Optimal theta?

Given  $W = 25 \text{ kN}(2500 \text{ kg})$  and  $\mu = 0.75$ , what's the optimum  $\theta$ ?



Identify optimum with an if conditional statement

1 Numerical solution of a sliding block

2 Bisection method

# Calculate the optimum angle to pull for a given force

- Given  $F = 17.5 \text{ kN}(1750 \text{ kg})$ ,  $W = 25 \text{ kN}$  and  $\mu = 0.75$ , what's  $\theta$ ?

$$\text{Try } \theta = 60^\circ : F = \frac{0.75 \times 25}{\cos(60) + 0.75 \sin(60)} = 16.31 \text{ kN.} (\delta = 1.19)$$

$$\text{Try } \theta = 70^\circ : F = \frac{0.75 \times 25}{\cos(70) + 0.75 \sin(70)} = 17.91 \text{ kN.} (\delta = -0.41)$$

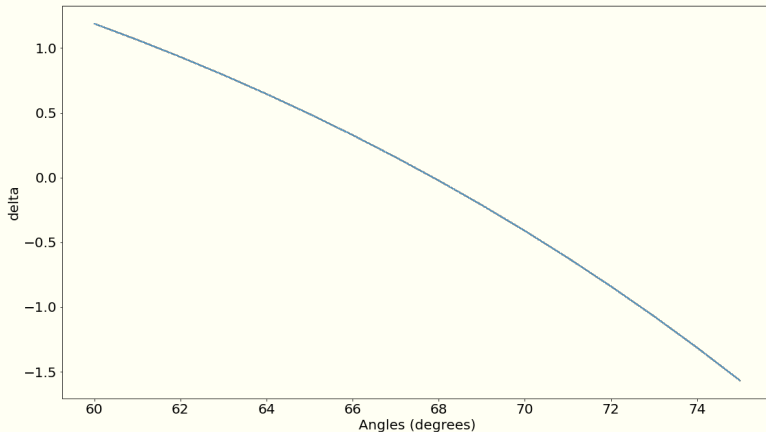
$$\text{Try } \theta = 65^\circ : F = \frac{0.75 \times 25}{\cos(65) + 0.75 \sin(65)} = 17.00 \text{ kN.} (\delta = 0.5)$$

$$\text{Try } \theta = 67.5^\circ : F = \frac{0.75 \times 25}{\cos(67.5) + 0.75 \sin(67.5)} = 17.43 \text{ kN} (\delta = 0.07).$$

This is **bisection method**!

# Calculate the optimum angle to pull for a given force

- Given  $F = 17.5 \text{ kN}(1750 \text{ kg})$ ,  $W = 25 \text{ kN}$  and  $\mu = 0.75$ , what's  $\theta$ ?



## What are the characteristics of a numerical solution?

- A numerical recipe is *a sequence of simple steps*
- *Flow of control* as each step is executed.
- Yields an *approximate* numerical answer (a finite number) for the problem
- These solutions can be very accurate
- Most answers are determined in an iterative approach (numerical method: mathematical / computer-aided technique) until a desired minimum/acceptable accuracy is obtained
- Typically, a finite set of iterations (steps) are used in the numerical method to obtain a solution. A means of determining *when to stop*.



# Numerical solution of a sliding block: Friction angles

