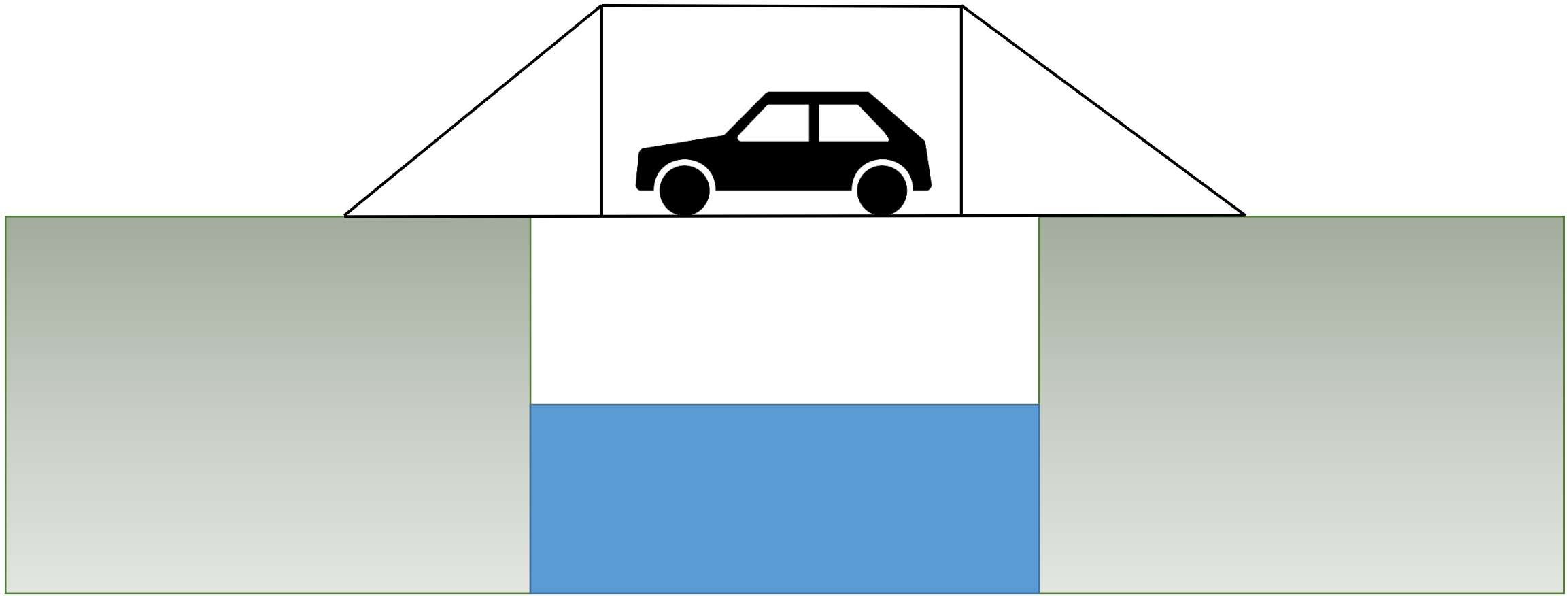
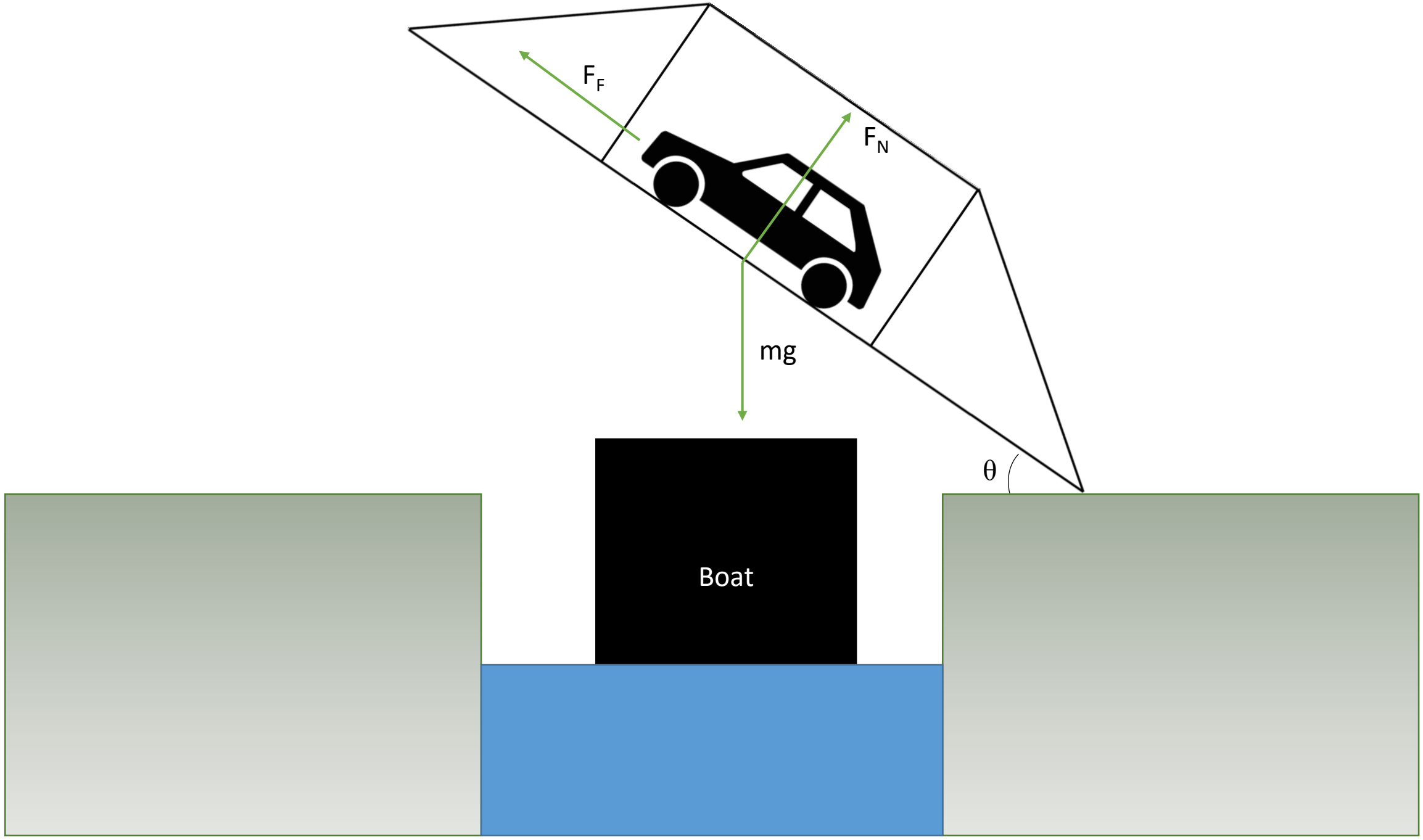



F_F
F_N
mg
θ
Boat

How do we determine if the car will slide?

- The angle the car will slide at is given by
- $\tan\theta_s = \mu_s$
  - Where μ_s is the static friction constant
- When $\theta_s \geq \theta$ the car won't move, and F_F is given as:
  - $F_F = m * g * \sin\theta$
- Note that the normal force is given as:
  - $F_N = m * g * \cos\theta$
- And thus, the three forces on the free body diagram are in equilibrium

When the car starts to slide...

- The frictional constant changes
  - It is now the kinetic frictional constant
  - Usually, $\mu_s > \mu_K$
- When $\theta_s < \theta$ the car will move, and F_F is given as:
  - $F_F = \mu_K F_N$
- Again, note that the normal force is given as:
  - $F_N = m * g * \cos\theta$
- Our three forces are no longer in equilibrium, and the car will move

Problem 1

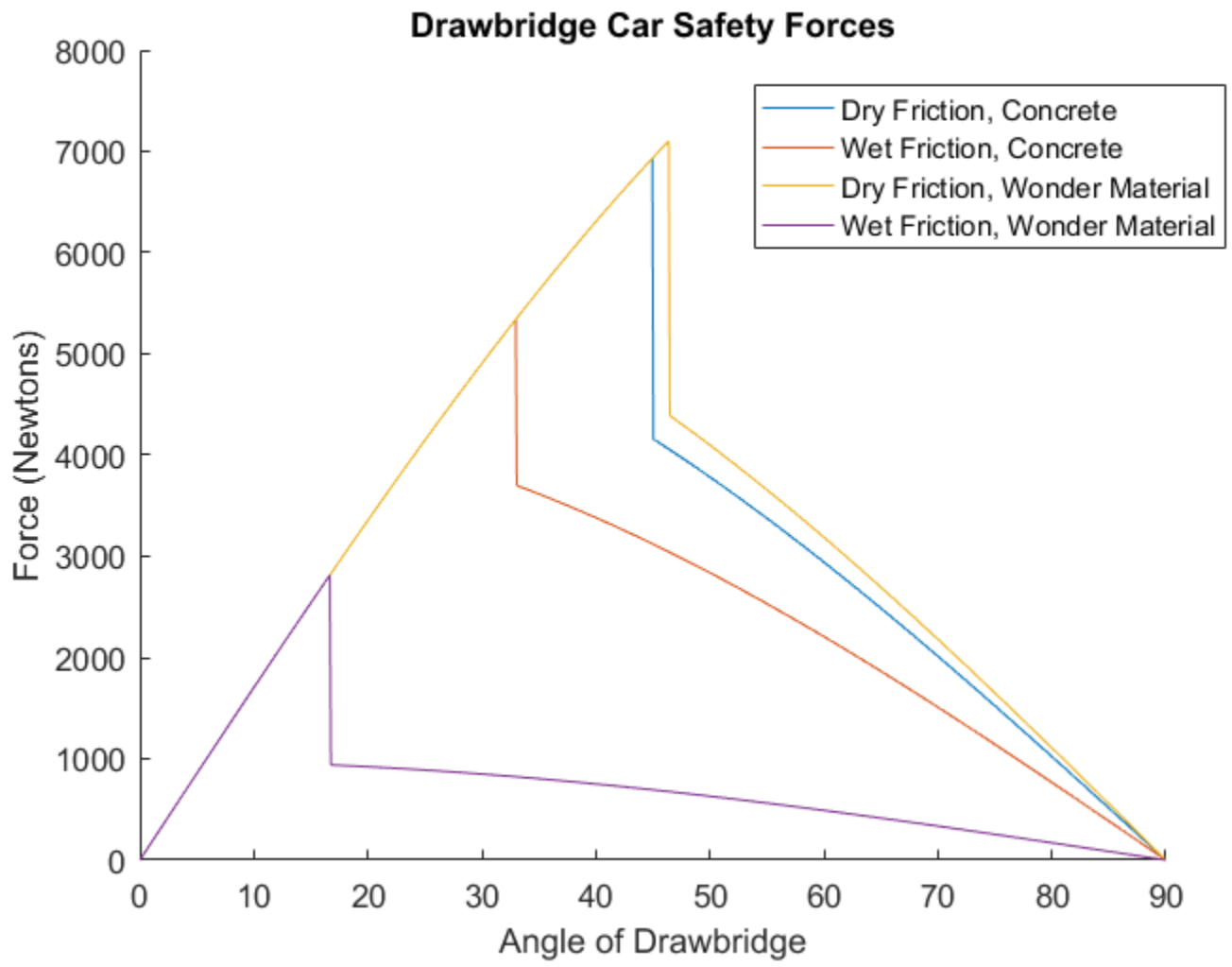
- Write a function in MATLAB that has the following inputs, in order:
  - angles → An array of angles, in degrees
  - mass → The mass of your car, in kg
  - static → μ_s (unit-less)
  - kinetic → μ_K (also unit-less)
- It should return
  - fricitonForces → An array of frictional forces for each angle in N
  - frictionForces should be the same size as angles
- To call this function, you should write
- [friction] = rec8prob1(angles, mass, static, kinetic)

Plotting for Problem 1

- Run your function with the following parameters
  - m = 1000kg
  - angles = linspace(0,90,1000)
- First, Dry Concrete
  - $\mu_s = 1.00$
  - $\mu_K = 0.60$
- Second, Wet Concrete
  - $\mu_s = 0.65$
  - $\mu_K = 0.45$
- Third, Dry “Wonder Material”
  - $\mu_s = 1.05$
  - $\mu_K = 0.75$
- Forth, Wet “Wonder Material”
  - $\mu_s = 0.30$
  - $\mu_K = 0.10$
- On the same axes, plot angles vs dry concrete, wet concrete, dry wonder material, and wet wonder material
- Add a title, legend, and axis labels

Your Plot

- Note, you can drag and move your legend by clicking and holding
- It should look almost exactly like this
- You should put your NetID in the title
- Submit your figure as an image:
  - File → Save as...



Problem 2

- Let’s say we have a linear system of equations
- $ax_1 + bx_2 + cx_3 = d$
- $ex_1 + fx_2 + gx_3 = h$
- $ix_1 + jx_2 + kx_3 = l$
- Wouldn’t it be nice if we could just get the computer to solve this for us?

Matrix Math to the Rescue!

- We can rewrite that equation as **A*x = b**
- Or...

$$\begin{bmatrix} a & b & c \\ e & f & g \\ i & j & k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} d \\ h \\ l \end{bmatrix}$$

We can solve for **x**

- **$x = A^{-1}b$**
- In MATLAB, we can use the matrix division command in the place of inverting **A**
- x = inv(A)*b
- Note:
  - Orientation matters, b MUST be vertical
  - Order matters! You MUST multiply **$A^{-1}b$** not **$b'A^{-1}$**
  - This will work for most linear systems of equations (there are some exceptions)
    - You will learn about those in other classes
- You can check your answer by taking **A*x**, the answer should be **b**!

Problem 2

- Create a function that takes a the matrices **A** and **b** from ANY linear system of equations and returns **x**
  - **A** is an n x n square matrix
  - **b** is a n x 1 array
  - **x** is a n x 1 array
- You should call your function this way:
- x = rec8prob2(A,b);

Problem 2 Example

- $4x_1 + 2x_2 + 1x_3 = 11$
- $-x_1 + 2x_2 = 3$
- $2x_1 + 1x_2 + 4x_3 = 16$
- Note that if a variable isn't in equation, it is multiplied by 0
- $x_1 = 1, x_2 = 2, x_3 = 3$