Engr 101 Project 3: Siting a Wind Farm

Project Due: October 5, 2021

Engineering intersections: Naval Architecture & Marine Engineering/Computer Science

Implementing this project provides an opportunity to work with data in files, automating an analysis, and producing a set of graphs to support your analysis.

The autograded portion of the final submission is worth 100 points, the Environmental Summary portion is worth 10 points, and the style and comments grade is worth 10 points.

You may work alone or with a partner. Please see the syllabus for partnership rules.

Project Roadmap

This is a big picture view of how to complete this project. Most of the pieces listed here also have a corresponding section in this document that goes into more detail.

1. **Read through the project specification (DO THIS FIRST)**
   This is a major project with a lot of details for you to monitor. Skim through this whole document first to get a sense of the project, then go back and read for detail.
2. **Make sure you understand the 5 design constraints and how to implement them**
   You need to interpret each design constraint and convert into MATLAB code. Check in office hours as to whether you've interpreted the constraints correctly.
3. **Implement the `analyzeWindFarm` function**
   Evaluate the design constraints one by one and confirm your results against their test case. Then, compose your own test cases using different parameters to ensure your function will not crash.
4. **Implement the `makePlots` function**
   Ensure you display every detail in each plot. You can make each plot individually, if you want, before assembling the summary figure.
5. **Verify that your code can reproduce the data in the Test Case in the Appendix**
6. **Check that your function is sufficiently and correctly commented**
7. **Submit `analyzeWindFarm.m`, and `makePlots.m` to the Autograder.**
   Submit `environmentalSummary.pdf` and Honor Pledge to Gradescope.

Purpose

The purpose of this project is to challenge you to perform a detailed analysis of a large data set and generate summary graphics to support your analysis. **A strength of the MATLAB programming language is that it can support data analysis/display of complex data sets.** Producing visual aids and interpreting visual data is a critical engineering skill.

Background

Now that our settlements on Proxima b have been established for several decades and there are sufficient mining and manufacturing resources, we can start thinking about producing and installing more environmentally friendly sources of power. An obvious choice is wind power. However, this planet is mostly water -- most of the accessible land is already given over to farming, industry, and cities (Fig. 1). Consequently, the population decides to install an offshore wind farm (Fig. 2).

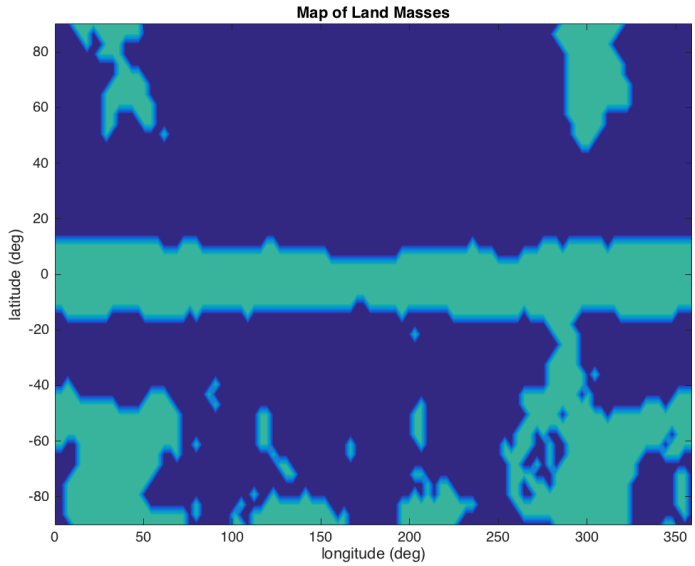


Figure 1. Distribution of land masses on the Proxima b planet (green = land, blue = water). The planet is only 29% land, and all of this is already in use for farms, industry, and cities.

Figure 2. An offshore wind farm off the coast of Germany. Offshore wind farms must withstand very harsh environmental conditions: winds, waves, and corrosion from salt water. (photo source)

Your engineering company is hired to design a tool to aid in locating the optimal location of this offshore wind farm. The wind turbine company you are partnering with tells you there are 5 design constraints[1] on the location of an offshore wind turbine:

1. The global-model-based average wind speed at the proposed site must be within a specified minimum-maximum range
2. The global-model-based average wave height at the proposed site must be less than a specified maximum value
3. The buoy measurement of the wave height at the proposed site must be less than the specified maximum wave height in constraint #2 for a given percentage of the time (the higher the percentage, the more conservative is the constraint)
4. The deck height of the wind turbine's base must be higher than any potential rogue waves, as estimated by the buoy-measured wave heights.
5. The standard deviation of the buoy wave height measurements must be less than 5% of the average global wave height.

To begin your investigation, your company deploys several buoys at locations you *believe* may be viable locations for a wind farm... but whether these locations are suitable for a wind farm depends on actual environmental conditions and the values that are assigned to the constraints. The buoys have gathered data for a year, and now you must process the buoy data to determine whether any of these locations are suitable for a wind farm.

[1] These constraints are described in more detail in Task 1.

Your Job

In this project, your job is to evaluate a **single** potential wind farm location (i.e. a single buoy location). You will do this by calculating whether the location passes all of the assigned constraints. You must produce a detailed and clear figure that summarizes each of the environmental conditions at the single potential wind farm location.

Note

Your functions must be general purpose -- in other words, they should work correctly with *any* provided buoy data, regardless of the number of measurements and chosen constraint values. This makes it possible to process any given buoy location on the planet. *Your functions only need to handle a single buoy location and a given set of parameters, though.*

Engineering Concepts

There are two high level engineering concepts that will be used in this project: *design constraints* and *engineering graphics*. These two terms are briefly explained in the next two sections.

Design Constraints

Design and constraints go hand-in-hand (Fig. 3), although constraints are not always evil things that stifle creativity. A design constraint is, literally and simply, a limitation on some aspect of whatever it is you are trying to create. Design constraints should not be arbitrary; they should address a specific concern about how the product or service will work or be used. Some examples of design constraints are shown in Table 1.

When you evaluate a design, especially a prototype of your design, you must state whether your design meets each of these constraints. Some constraints can be met right away (e.g., choice of programming language), but you won't know if you meet other constraints (e.g., maximum false positive rate for medical test) until after you have a prototype and start running test cases. Continually checking your design against its constraints is crucial to the design process. If you forget to check your design constraints often, you might invent something cool, spend weeks building it, more weeks checking it, more weeks documenting it, and then find out it's not what you need because it fails all the constraints. Since you have to check the constraints so many times, it makes sense to (semi) automate the process by writing a computer program to analyze your constraints for you. This way, every time you change something, you can just re-run your computer program to check the constraints.

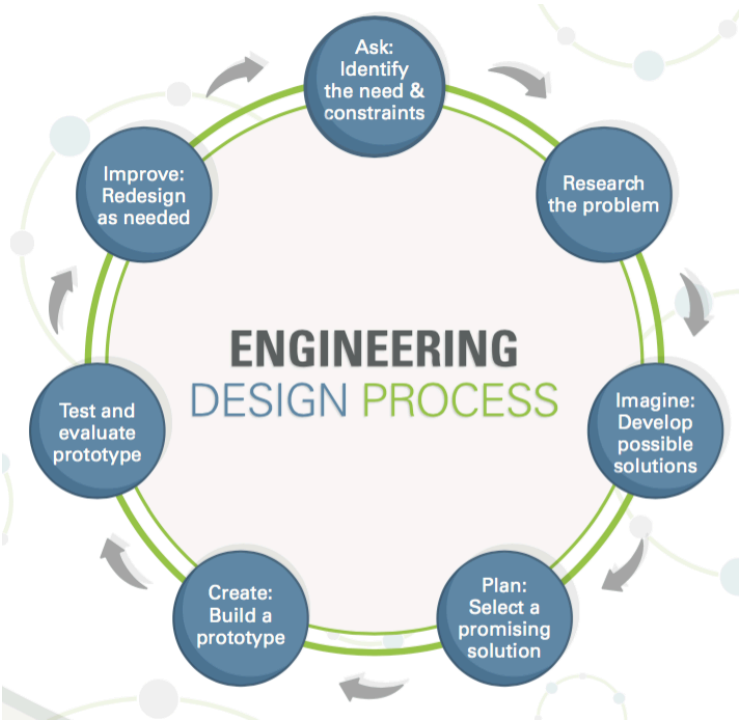


Figure 3. A description of the engineering design process. Note that the top of the cycle (where you would start) includes the constraints of the process. (photo source)

Table 1. Some examples of design constraints in engineering. These are not the sole reasons for these constraints; all of these constraints have many reasons, once you start thinking about the entire system.

design constraint	reason for constraint
maximum acceleration of a car	user comfort; safety during operation
programming language	run-time speed; support; availability
min-max range of aisle width on airplane	passenger comfort vs. revenue and safety
maximum delivery time for packages	minimum profit/revenue required
maximum false positive rate for medical test	minimize stress to people
maximum allowable toxins in wastewater treatment plant effluent	environmental health
minimum cargo capacity for a ship	minimum profit/revenue required

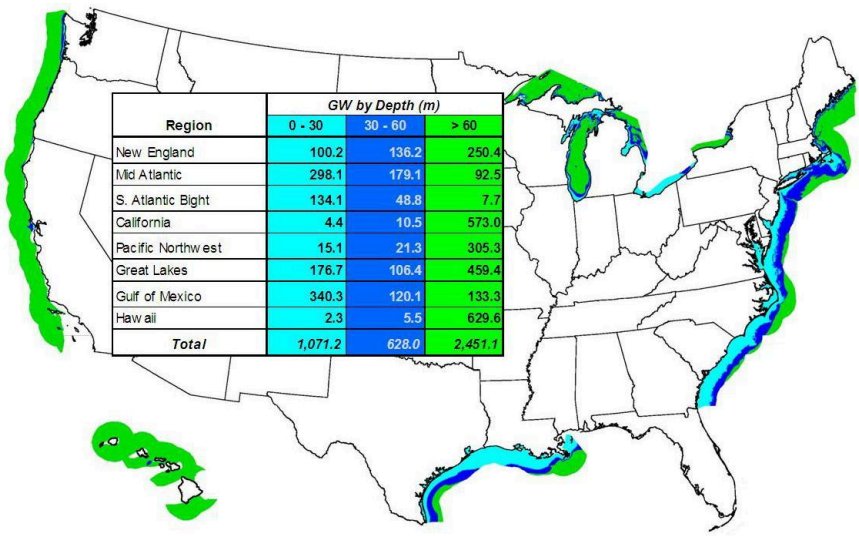
Engineering Graphics

An engineering graphic is any picture, figure, graph, or video that conveys your engineering information to someone else. They include, but are not limited to, engineering drawings, which are highly accurate drawings used to manufacture items such as circuit boards, gears, 3D printed parts, etc.

Engineering graphics are incredibly important. You can often convey something much faster and clearer if you use a picture or a video. All the project specifications for this class would be much harder to understand without any sort of graphics to explain what's going on!

This video summary of James Cameron's dive to the Mariana Trench is an excellent example of how to describe a highly scientific expedition in just 2 minutes using small words. In this article on the potential of offshore wind energy production, figures are used liberally to explain where wind turbines could be effective (Fig. 4) and the progression of types of wind turbines (Fig. 5), and concepts of future wind turbines (Fig. 6). If you have several graphs that are summarizing a complicated analysis, you can plot them together in one figure (Fig. 7) to improve the final product and to keep the data together.

Note the differences in all of the visuals. Some are drawings/cartoon-like (Figs. 4 & 5), some are renderings from 3D models (Fig. 6), and some are traditional graphs (Fig. 7). You can use any or all of these types of graphics (and others!) to help communicate your ideas to someone else.



Region	GW by Depth (m)		
	0 - 30	30 - 60	> 60
New England	100.2	136.2	250.4
Mid Atlantic	298.1	179.1	92.5
S. Atlantic Bight	134.1	48.8	7.7
California	4.4	10.5	573.0
Pacific Northwest	15.1	21.3	305.3
Great Lakes	176.7	106.4	459.4
Gulf of Mexico	340.3	120.1	133.3
Hawaii	2.3	5.5	629.6
Total	*1,071.2*	*628.0*	*2,451.1*

Figure 4. United States offshore wind resource by region and depth.
Photo source: National Renewable Energy Laboratory.

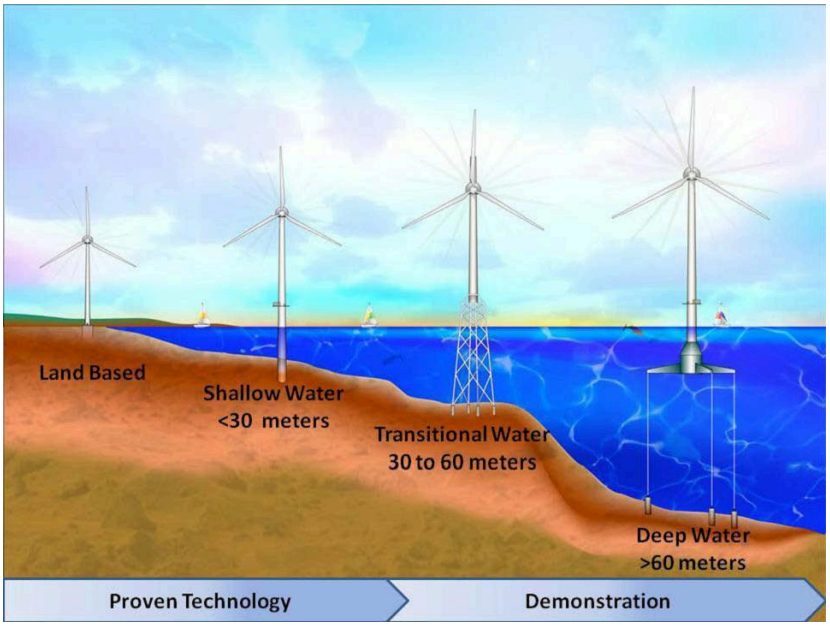


Figure 5. Progression of expected wind turbine evolution to deeper water. Photo source: National Renewable Energy Laboratory.

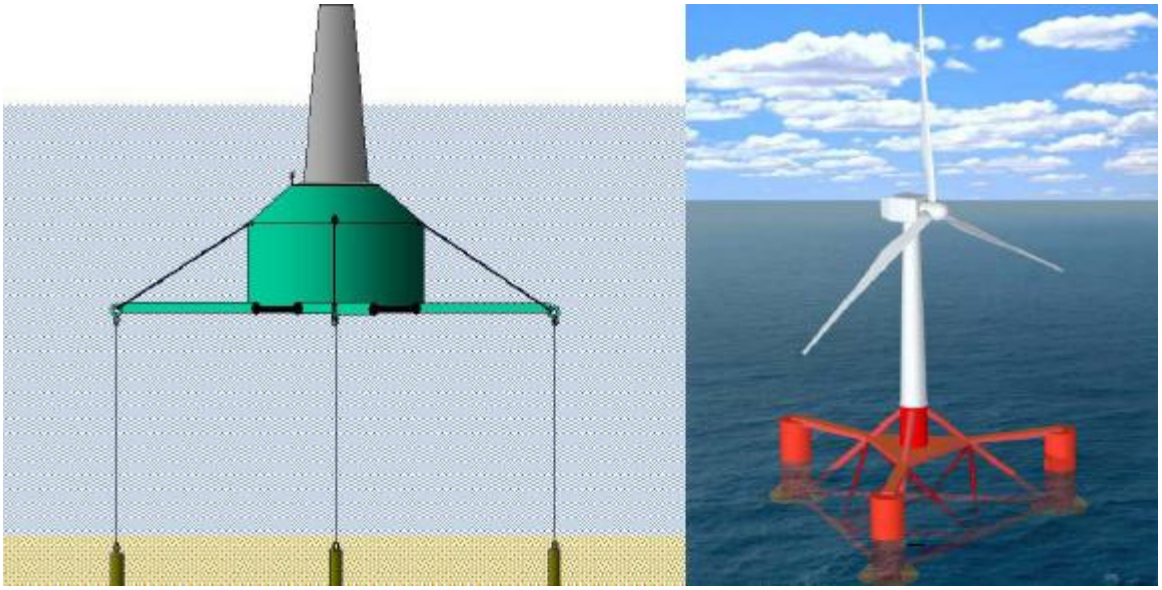
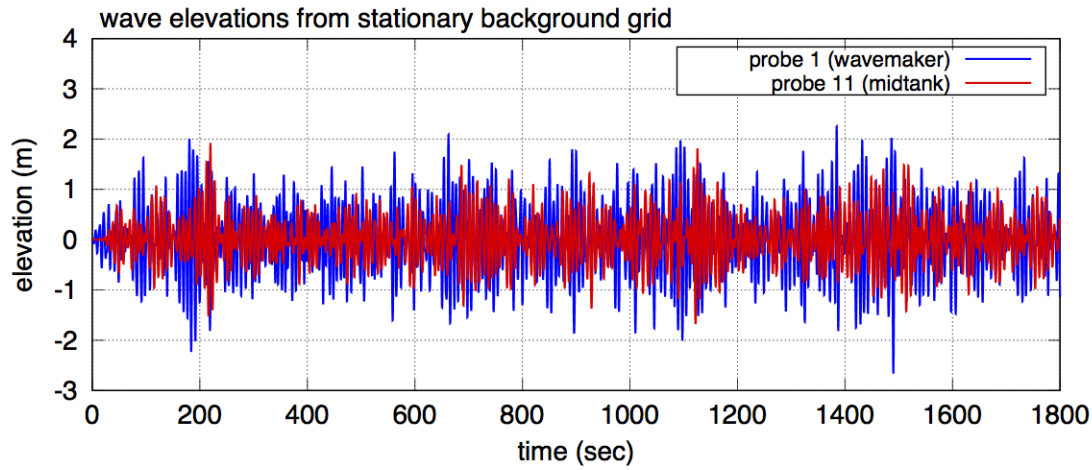
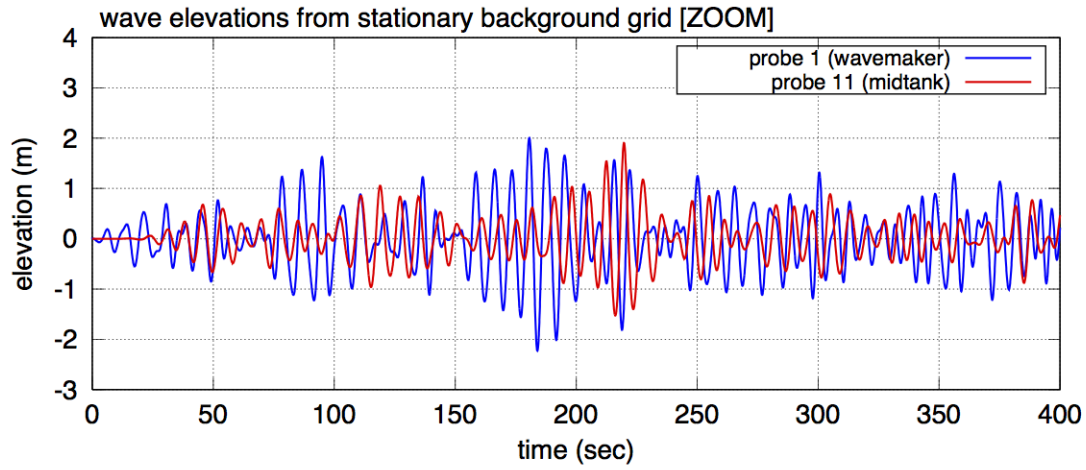
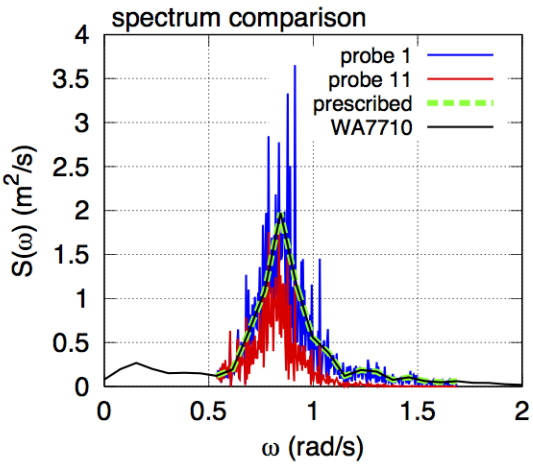


Figure 6. Floating wind turbine platform configurations, one designed by NREL and a Dutch Tri-Floater concept. Photo source: National Renewable Energy Laboratory.







		Waves2Foam		
	WA7710	prescribed	probe 1	probe 11
H_s	3.2174	2.8926	2.7615	1.9130
T_p	7.4473	7.4473	6.8840	7.5156
max	--	--	2.2510	1.9058
min	--	--	-2.6608	-1.6784
mean	--	--	0.0004	0.0083
stdev	--	--	0.6928	0.4807

Figure 7. A single figure produced in MATLAB that contains multiple and different plots and a table of data. This figure ensures that related information (the individual graphs) are grouped together and viewed at the same time. Photo Source: Laura Alford.

Program Tasks

There are 2 primary tasks for your program: 1) evaluate the constraints on the proposed wind farm location, and 2) produce a figure that summarizes the environmental conditions at the proposed location. These two tasks are described in more detail in the next two sections. The "parameters" referred to in the task descriptions are defined in the Required Functions Section.

Task 1: Evaluate Constraints (`analyzeWindFarm.m`)

Your function will use MATLAB's "compound return" capability to return 5 true or false logical values corresponding to whether each of the 5 design constraints is met at the location specified. The location is specified by a latitude/longitude index pair which can be accessed using the function parameter `filenameBuoy`. The function must return the values *in the following order*:

1. Constraint 1: Is the global-model-based average wind speed at the proposed location within the min-max range (parameters 4 & 5)?
2. Constraint 2: Is the global-model-based average wave height at the proposed location < max value (parameter 6)?
3. Constraint 3: Is the local buoy measurement of wave height < max value (parameter 6) at least XX% (parameter 7) of the time?
4. Constraint 4: Are all potential rogue wave heights < deck height (parameter 8)?
5. Constraint 5: Is the standard deviation of the buoy wave height measurements < 5% of the global-model-based average wave height?

These constraints are described in more detail in the next sections. You will also need to input data from several `.csv` files (see Input Files).

Constraint 1: Global-Model-Based Average Wind Speed

The global-model-based average wind speed is the estimated average wind speed at a given latitude-longitude coordinate based on a mathematical model of the wind speed across the planet. These types of models (such as this one) are incredibly useful in any design process as they can provide a prediction of wind speeds at locations where ground truth data does not exist (recall: ground truth is measured data we trust), so we can use them as a first guess for our design.

To evaluate this constraint, import the global-model-based average wind speed from a `.csv` file (see Input Files for detail on this file), find the wind speed corresponding to the buoy location (lat/lon index pair), and then evaluate:

Is this wind speed ≥ specified minimum (parameter 4)?
AND
Is this wind speed ≤ specified maximum (parameter 5)?

Return this answer as the first value in your compound return of the function.

Constraint 2: Global-Model-Based Average Wave Height

The global-model-based average wave height is the estimated average wave height at a given latitude-longitude pair based on a mathematical model of the wave heights across the planet. To evaluate this constraint, import the global-model-based average wave height from a `.csv` file, find the wave height corresponding to the buoy location (lat/lon pair), and then evaluate:

$$\text{Is this wave height} < \text{specified maximum (parameter 6)?}$$

Return this answer as the second value in your compound return of the function.

Constraint 3: Buoy Measured Wave Height Exceedance

Constraint 2 checked only if the *average* wave height was less than a given maximum. We now need to verify how often the *actual* wave heights are less than that given maximum. The actual wave heights must be less than the given maximum for a given percentage of time (a measure of risk). The buoy measured wave height is a time series of wave heights measured by a buoy located at a given latitude-longitude pair. To evaluate this constraint, import the buoy measured wave height from a `.csv` file, giving you wave heights vs. time. Then, evaluate:

$$\text{Is } \frac{\#\,(\text{wave heights} < \text{specified maximum (parameter 6)})}{\text{total \# wave heights}} > \text{wave height risk (parameter 7)?}$$

Return this answer as the third value in your compound return of the function.

Constraint 4: Risk of a Rogue Wave

Another danger to the wind farm is a rogue wave at the proposed location. Rogue waves are very large waves that appear suddenly and without warning; they can cause extreme damage to structures and put human lives at risk. A rogue wave is defined as a wave that is twice as high as the significant wave height (significant wave height is what is recorded by the buoy). If the deck that supports the wind turbine is higher than the rogue wave, then the risk of damage and potential loss of life is minimized. Therefore, we need to check whether the deck height is greater than the height of any potential rogue waves. To evaluate this constraint, multiply all the buoy-measured wave heights by 2 to estimate the height of "rogue waves",

$$\text{rogue wave heights} = 2 \times \text{buoy-measured wave heights}$$

then evaluate:

$$\text{Are all potential rogue wave heights} < \text{deck height (parameter 8)?}$$

Return this answer as the fourth value in your compound return of the function.

Constraint 5: Standard Deviation of Buoy Wave Height

The proposed location needs to have wave heights that are fairly consistent. The standard deviation of the buoy measured wave heights can be used to quantify consistency. To evaluate this constraint, calculate the standard deviation of the buoy measured wave heights. Then, evaluate:

Is the standard deviation of the buoy-measured wave heights < 5% of the global-model-based average wave height at the buoy lat-lon?

Return this answer as the fifth value in your compound return of the function.

Task 2: Summarize Environmental Conditions (`makePlots.m`)

You also need to produce a figure visually summarizing the conditions at the proposed site of the wind farm. This visual summary must include 5 plots (see Fig. 8):

1. Map of global-model-based average wind speed across the planet
2. Map of global-model-based average wave height across the planet
3. Map of *all* potential wind farm locations (i.e. all lat-lon combinations) based only on constraints 1 & 2 with the proposed location (i.e. the buoy's lat-lon location) marked with a red square
4. Histogram of buoy measured wave heights
5. \\\\\\\Time series of buoy measured wave heights compared to average wave height at that location

Plots 1 & 2 are general, high-level maps that would be the same for all potential buoy locations. You include them here to give context to your audience. Plot 3 shows your audience where the potential wind farm is located (it corresponds with the buoy location, remember). Plots 4 & 5 summarize the local wave conditions at that wind farm/buoy location.

In this function, you are required to input data from several `.csv` files (see Input Files) and create one `.pdf` file (see Output Files).

Note

When calling the `print()` function to save your environmental summary as a `.pdf`, use the `'-dpdf'` and `'-fillpage'` options. The first option tells MATLAB to save it as a `.pdf` file and the second option tells MATLAB to stretch/shrink the overall figure such that it fills one US-standard letter sized page. See the MATLAB documentation for `print` for more details.

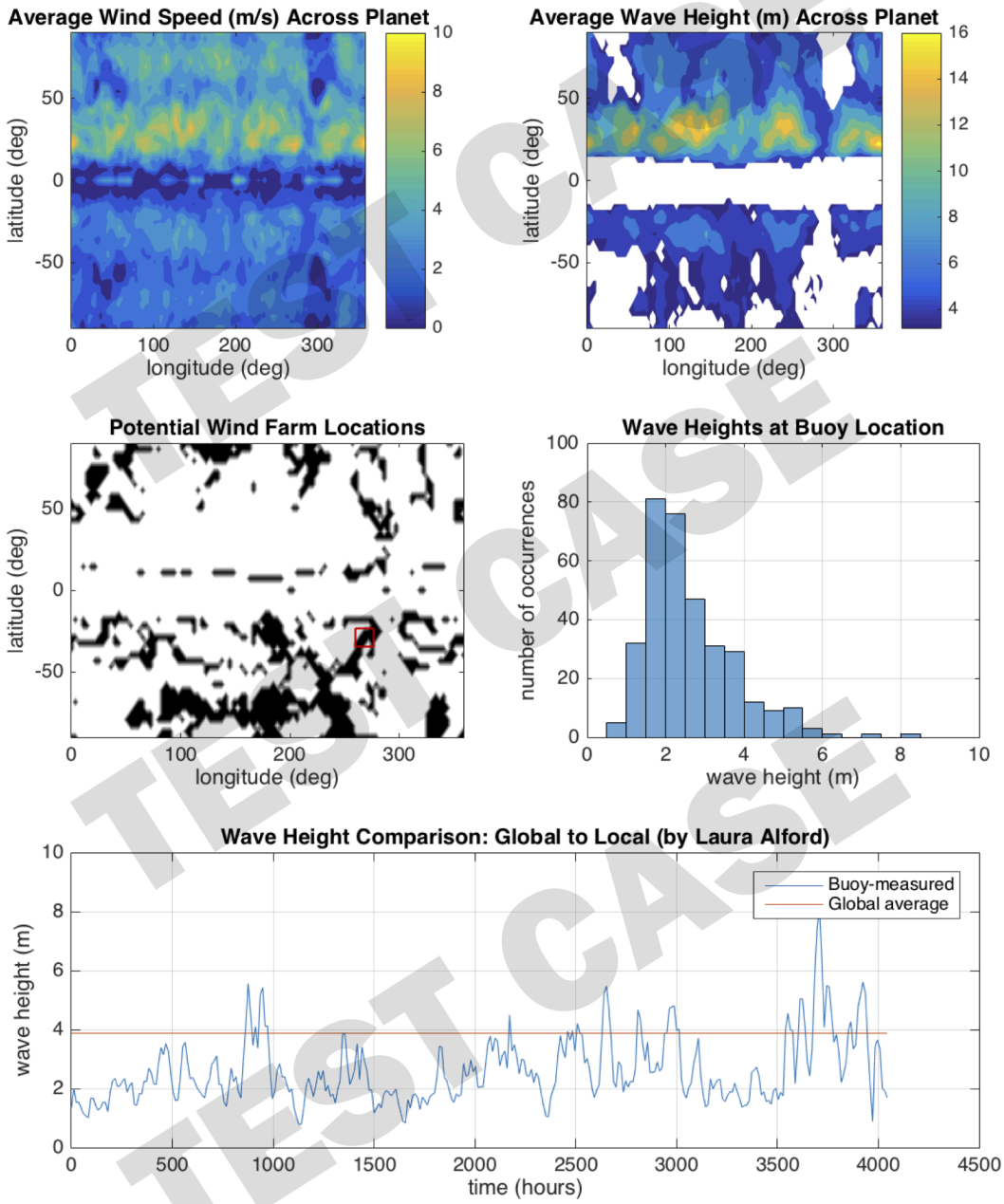


Figure 8. A single figure created by MATLAB that visually summarizes the conditions at the proposed site of the wind farm. Note that it starts at the global level and then works its way down to the local level. **You will need to replace the "Laura Alford" part in the title of the bottom-most plot with your name for this assignment (you wouldn't normally put your name in a title like this, but we're doing it here for grading purposes). Your figure will not have TEST CASE across the top of it, either (that is just for this document).**

‘

Figure Guidelines

To produce the summary figure in Fig. 8, you will need to use the `subplot()` function to put multiple plots in one figure. Arrange your plots as shown in Fig. 9 (plot numbers correspond to those listed at the beginning of the Task 2 section). Your program must save the figure as `environmentalSummary.pdf`.

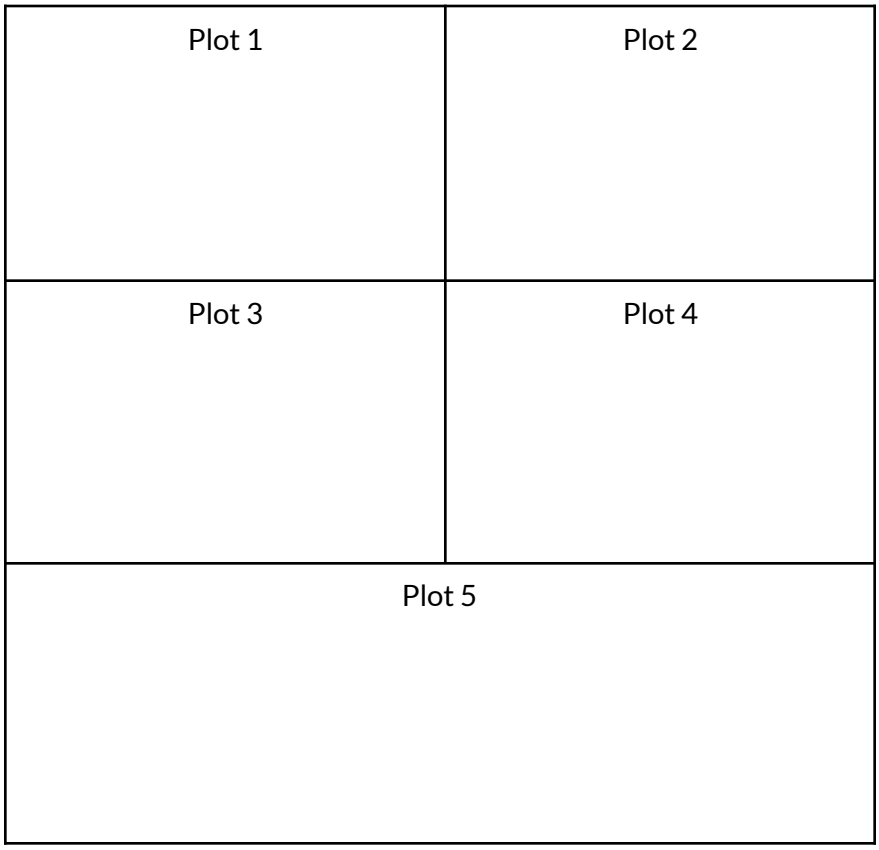


Figure 9. The layout for the individual plots in the `environmentalSummary.pdf` file.

Note

When you compare your `environmentalSummary.pdf` to the one in Fig. 8, it's okay if there are small differences, e.g. tick mark spacing is slightly different, aspect ratio of the graphs is slightly different, overall size is slightly different. MATLAB renders things a little differently depending on the computer. We will grade only those details explicitly specified on the next page.

Details for each plot are:

1. Map of average wind speed across the planet (measured globally)
    a. Use the `contourf` function to create a map of the average wind speed vs. latitude and longitude (tip: use `meshgrid()`)
    b. Do NOT plot the mesh lines (e.g. set `'LineStyle','none'`)
    c. Use the parula colormap
    d. Make sure all labels, color bars, legends, etc. match Fig. 8
    e. Use default MATLAB font sizes
2. Map of average wave height across the planet (measured globally)
    a. Use the `contourf` function to create a map of the average wave height vs. latitude and longitude (tip: use `meshgrid()`)
    b. Do NOT plot the mesh lines (e.g. set `'LineStyle','none'`)
    c. Use the parula colormap
    d. Make sure all labels, color bars, legends, etc. match Fig. 8
    e. Use default MATLAB font sizes
3. Map of all potential wind farm locations based on constraints 1 & 2 with potential location of wind farm (i.e. the buoy location) marked with a red square
    a. Use the `contourf` function to create a map of all potential wind farm locations based on constraints 1 & 2 vs. latitude and longitude
        i. Latitude-longitude combinations that meet BOTH constraints should be given a value of 1
        ii. Latitude-longitude combinations that do NOT meet both constraints should be given a value of 0
    b. Do NOT plot the mesh lines (e.g. set `'LineStyle','none'`)
    c. Plot the buoy location with a red square with a marker size of 12 points
    d. Use the gray colormap in reverse
    e. Make sure all labels, color bars, legends, etc. match Fig. 8
    f. Use default MATLAB font sizes
4. Histogram of buoy measured wave heights
    a. Show the grid
    b. Make sure all labels, color bars, legends, etc. match Fig. 8
    c. Use default MATLAB font sizes
5. Time series of buoy measured wave heights
    a. Plot the buoy measured wave heights vs. time
    b. Plot the average wave height (from the global measurements for the buoy's lat-lon) across all time
    c. Show the grid
    d. Make sure all labels, color bars, legends, etc. match Fig. 8
        i. **Note: Replace 'Laura Alford' with your name (or names if working with a partner) in Plot 5**
    e. Use default MATLAB font sizes

Program Description

Before writing your function, go to Project 3 on the Google drive and download the files in the "Starter Files" folder. The .csv files (`lat.csv, lon.csv, windSpeedTestCase.csv, waveHeightTestCase.csv, buoyTestCase.csv`) will be used in writing and testing your functions. Starter code for your two functions (`analyzeWindFarm()` and `makePlots()`) are also available.

In general, you need to write a MATLAB program that:

1. Passes in 8 parameters corresponding to various information you need to evaluate the wind farm (see Required Functions for more detail)
2. Inputs data from several `.csv` files
3. Analyzes data and evaluates the constraints
4. Draws a figure that summarizes the environmental conditions at the proposed wind farm location and saves it as a `.pdf` file
5. Return 5 values (in a compound return) that correspond to whether or not each of the 5 constraints is passed

Note: The above list can be considered to be the *start* of a program design for this project. We highly encourage you to expand this program design into a detailed list of steps that you need to implement for this project. There is a test case in the Appendix to help you with testing your program.

Terminology

In this program description, the following terms will be used:

- **Required Functions:** Functions required by the project. These functions must use the interface defined (i.e. you have to use the function header provided to you).
- **Function Header:** the line of code in your `.m` file that defines the function's name, parameters, and return values.
- **Parameters:** Variables that are passed into your function when it is called. These variables are defined in the function header and may be assumed to exist when the function is called.
- **Return Values:** Variables that are returned by your function to the caller (e.g. your main program, the autograder, etc.). These variables are defined in the function header.
- **Input Files:** Files that will be used by your function, such as a `.csv` file. "Input Files" are different from the parameters described above.
- **Output Files:** Files that will be produced by your function, such as a figure saved as `.pdf` file. "Output Files" are different from the return values defined above.

Required Functions

There are 2 required functions for this project, one for each Program Task. The functions and their parameters and return values are defined below.

Notes

- Do NOT place plotting functions in `analyzeWindFarm()`. If you do, you will automatically receive a zero for all test cases related to `analyzeWindFarm()`.
- The `analyzeWindFarm()` function does not need to use the `lat.csv` and `lon.csv` files.

```
function [ c1, c2, c3, c4, c5 ] = analyzeWindFarm( filenameWind, filenameWave,
filenameBuoy, windSpeedMin, windSpeedMax, waveHeightMax, waveHeightRisk,
deckHeight )

% Function to complete Task 1. Evaluates the 5 constraints on the location of a
% wind farm.
%
%   parameters:
%          filenameWind: a string that names the file containing the
%                        global-model-based average wind speed
%                        (i.e. 'windSpeedTestCase.csv')
%          filenameWave: a string that names the file containing the
%                        global-model-based average global wave heights
%                        (i.e. 'waveHeightTestCase.csv')
%          filenameBuoy: a string that names the file containing the time
%                        series of wave heights measured by the buoy
%                        (i.e. 'buoyTestCase.csv')
%          windSpeedMin: for constraint 1 -- minimum wind speed (m/s)
%          windSpeedMax: for constraint 1 -- maximum wind speed (m/s)
%         waveHeightMax: for constraints 2 & 3 -- maximum wave height (m)
%        waveHeightRisk: for constraint 3 -- maximum wave height risk (%)
%            deckHeight: for constraint 4 -- height of the deck that supports
%                        the turbine base (m)
%
%   return values:
%                    c1: boolean values corresponding to whether the wind
%                        farm location passes constraint #1
%                    c2: boolean values corresponding to whether the wind
%                        farm location passes constraint #2
%                    c3: boolean values corresponding to whether the wind
%                        farm location passes constraint #3
%                    c4: boolean values corresponding to whether the wind
%                        farm location passes constraint #4
%                    c5: boolean values corresponding to whether the wind
%                        farm location passes constraint #5
```

```
function [ ] = makePlots( filenameWind, filenameWave, filenameBuoy,
windSpeedMin, windSpeedMax, waveHeightMax )

% Function to complete Task 2. Creates a figure with multiple plots that
% summarizes the environmental conditions for a wind farm.  Saves figure as
% a .pdf file.
%
%   parameters:
%          filenameWind: a string that names the file containing the
%                        global-model-based average wind speed
%                        (i.e. 'windSpeedTestCase.csv')
%          filenameWave: a string that names the file containing the
%                        global-model-based average global wave heights
%                        (i.e. 'waveHeightTestCase.csv')
%          filenameBuoy: a string that names the file containing the time
%                        series of wave heights measured by the buoy
%                        (i.e. 'buoyTestCase.csv')
%          windSpeedMin: for constraint 1 -- minimum wind speed (m/s)
%          windSpeedMax: for constraint 1 -- maximum wind speed (m/s)
%         waveHeightMax: for constraint 2 -- maximum wave height (m)
%
%   return values: none
%
```

Input Files

These particular functions require a LOT of data, and these data are scattered across many files. Within your functions, **use `csvread()` to import data from the following files**:

1. Latitude in degrees from the `lat.csv` file (this file will always be the same)
2. Longitude in degrees from the `lon.csv` file (this file will always be the same)
3. Global-model-based average wind speed data from a `.csv` file (this file will change, based on parameter #1); rows correspond to the latitudes in the `lat.csv` file, columns correspond to the longitudes in the `lon.csv` file
4. Global-model-based average wave height data from a `.csv` file (this file will change, based on parameter #2); rows correspond to the latitudes in the `lat.csv` file, columns correspond to the longitudes in the `lon.csv` file; NaN (Not-a-Number) corresponds to a lat-lon combination that is on land
5. Buoy data from a `.csv` file (this file will change, based on parameter #3); lat-lon corresponds to the INDEX of the lat-lon vectors (not the lat-lon in degrees)

Note

Before you start coding, open these `.csv` files (you can use either a text editor or a spreadsheet program to view what's in the files) and understand what is in them and how it is organized.

These files must be in the same directory as your `analyzeWindFarm.m` and `makePlots.m` files, so the filename you give to `csvread()` is what comes through as either parameter 1, 2, or 3 (meaning, you don't need to tell MATLAB to look for the `.csv` files in another directory). Remember, you can tell the `csvread()` function where to start reading data. Take care to organize your data properly in MATLAB after you read it! Naming your variables appropriately (i.e. giving the variable a name that represents exactly what data it has) helps with the organization of your program.

Output Files

There is one output file for the `makePlots.m` function: an image file named `environmentalSummary.pdf`. Once you have created the summary figure described in Task 2, save it using the `print()` function.

Note

Your `makePlots()` function needs to create the `environmentalSummary.pdf` file when it runs. This is so you could use it in a presentation or report. However you should NOT submit `environmentalSummary.pdf` to the Autograder. Instead, **submit `environmentalSummary.pdf` to Gradescope.**

Hints for a Successful Program

Acceptable Functions

Due to the structure of the Autograder, there are certain limitations on which functions you can use and how you need to use them:

- In the summary figure, use only **`contourf()`**, **`histogram()`**, and **`plot()`** to make your graphs.
- Only use **`csvread()`** to read in data from the .csv files.

If you fail to follow these guidelines, your functions will not run in the Autograder.

Submission and Grading

This project has three deliverables: the `analyzeWindFarm.m` and `makePlots.m` functions and the `environmentalSummary.pdf` file. Grades are broken down into 3 categories: the autograder, the environmental summary picture, and the style and comments. You will also be required to sign the Honor Pledge.

After the due date, this project will be graded in three parts. First, the Autograder will be responsible for evaluating your submission. Second, one of our graders will evaluate your submission for style and commenting and will provide a maximum score of 10 points. Third, a grader will also evaluate the correctness of the `environmentalSummary.pdf` file that is submitted to Gradescope and will provide a maximum score of 10 points. Thus the maximum total number of points on this project is 120 points. Some test cases on the autograder are public (meaning you can see your output compared to the solution as well as the inputs used to test your submission) while many are private (you are simply told pass/fail and the reason). The breakdown of points for each category is described in the next sections.

Autograder

Maximum Score: 100 points

Submit the `analyzeWindFarm.m` and `makePlots.m` functions to the Autograder (autograder.io) for grading. The autograder provides you with a final score (out of 100 points) as well as information on *a subset of its test cases*. We provide this autograder to give you a sense of your current progress on the project.

Submitting files to the Autograder

Note: You don't *have* to wait until you have all the files ready before submitting to the Autograder. You can submit a subset of files and get feedback on the test cases related to those files. However, **to receive full credit, you need to submit all files and pass all test cases within a single submission.**

Project scoring and feedback on the Autograder

You are limited to 5 submissions on the Autograder per day. After the 5th submission, you are still able to submit, but all feedback will be hidden other than confirmation that your code was submitted. The autograder will only report a subset of the tests it runs. It is up to you to develop tests to find scenarios where your code might not produce the correct results.

You will receive a score for the autograded portion equal to the score of your best submission. Your latest submission with the best score will be the code that is style graded. For example, let's assume that you turned in the following:

Submission #	1	2	3	4
Score	50	100	100	50

Then your grade for the autograded portion would be 100 points (the best score of all submissions) and Submission #3 would be style graded since it is the latest submission with the best score. Please refer to the syllabus for more information regarding partner groups and general information about the Autograder.

Gradescope

Maximum Score: 10 points

Submit your `environmentalSummary.pdf` figure and a signed copy of the Honor Pledge to Gradescope. It will be graded according to the required plots and formatting specified in the Figure Guidelines. Go to Canvas and click on the "Gradescope" link on the left sidebar to access Gradescope. Follow the directions in the next sections to create and submit your assignment.

Submitting your assignment to Gradescope

This Gradescope assignment has two questions: the Environmental Summary and the Honor Pledge. The Environmental Summary is the figure created by the `makePlots()` function. Create a separate document that states the Honor Pledge and includes your signature below the Honor Pledge (if your signature does not look like your name, also print your name).

> If working alone:
>
> > *I have neither given nor received unauthorized aid on this assignment, nor have I concealed any violations of the Honor Code on this assignment.*
>
> If working with a partner:
>
> > *We have neither given nor received unauthorized aid on this assignment, nor have we concealed any violations of the Honor Code on this assignment.*

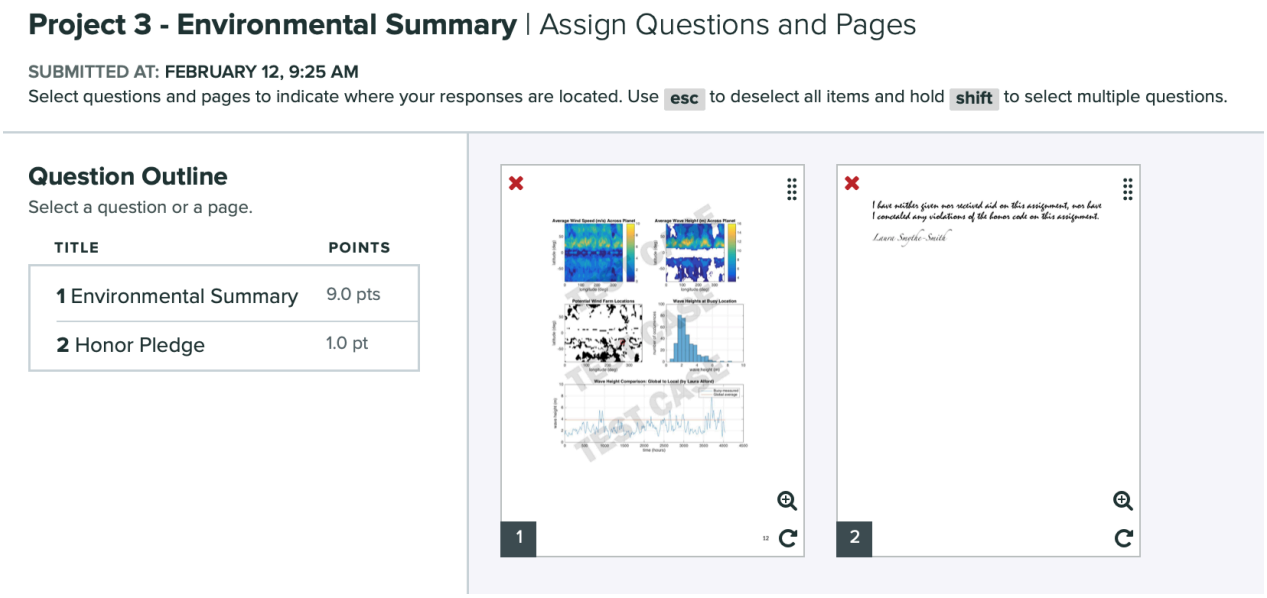
The Honor Pledge may be typed or written; your signature must be *written* and added to the Honor Pledge document; if you are working in a partnership, both partners must sign the Honor Pledge. To create the Honor Pledge document, you have several options:

1. Write the Honor Pledge and sign it on a piece of paper and then scan it to create an electronic copy; we recommend the Scannable app to get a high quality scan (low quality scans will not be accepted).
2. On a tablet or touchscreen, write the Honor Pledge, sign it, and save the file electronically. Please come to Office Hours if you need help with either of these options.
3. Type the Honor Pledge in a document on your computer and then add your signature separately.

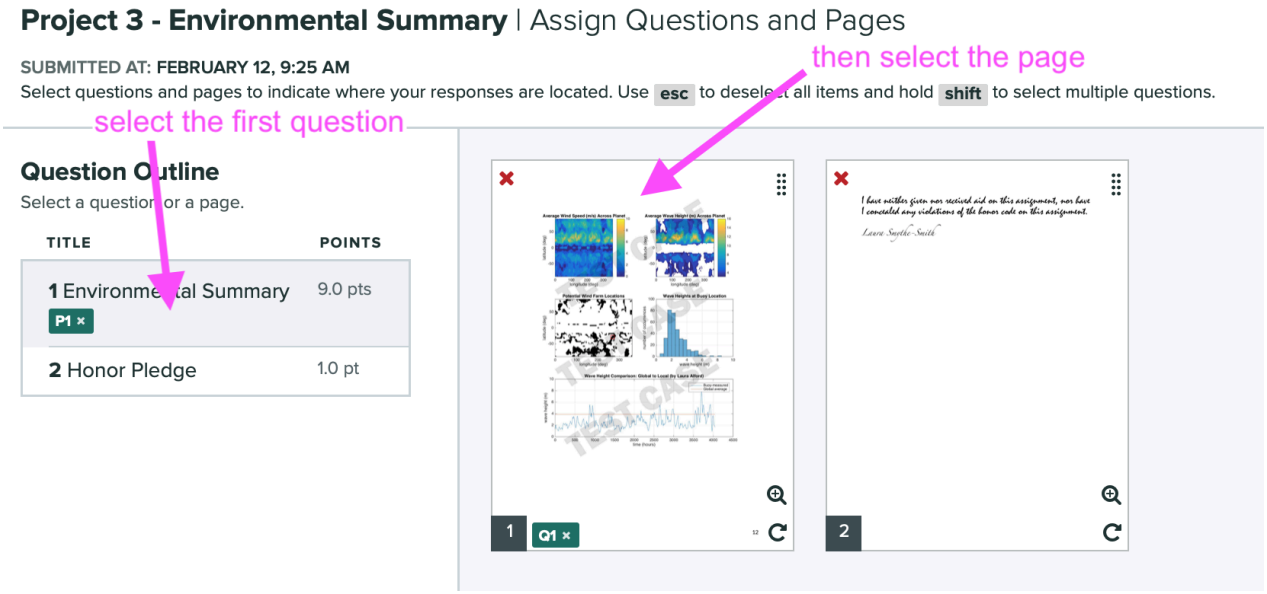
If you have any trouble creating this document, please come to office hours!

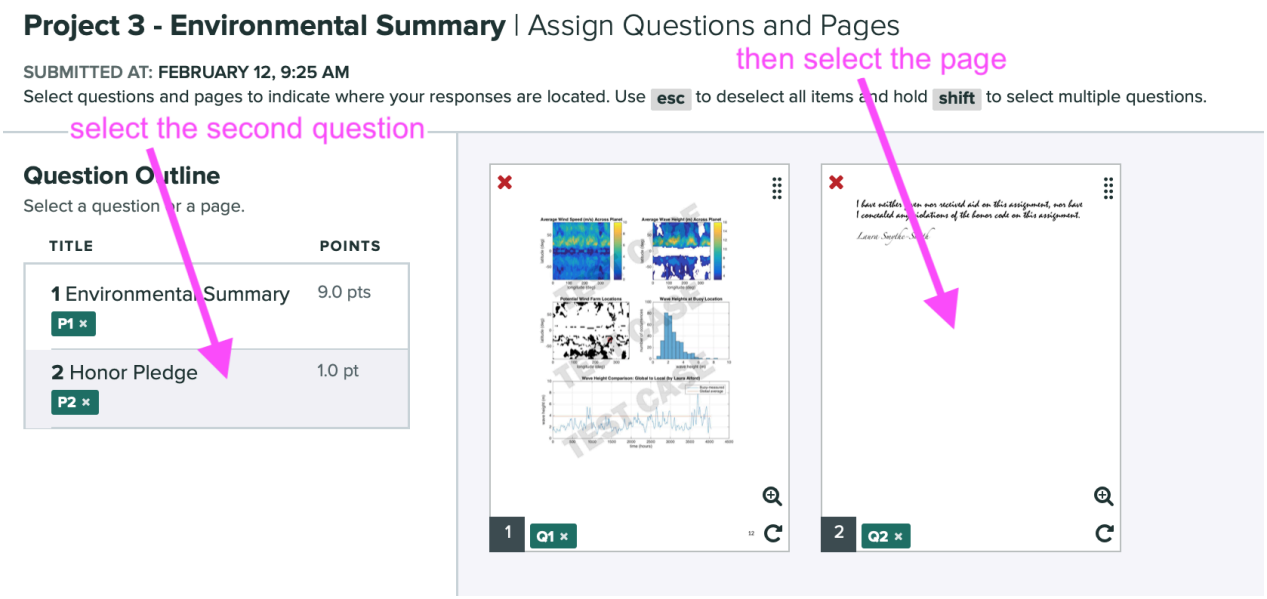
Create a document that contains both the `environmentalSummary.pdf` picture and a signed copy of the Honor Pledge. **Save this document as a single `.pdf`** (the name of this file does not matter).

When you upload your assignment, you will see something similar to what is shown below:



You now need to assign which pages go with which question:

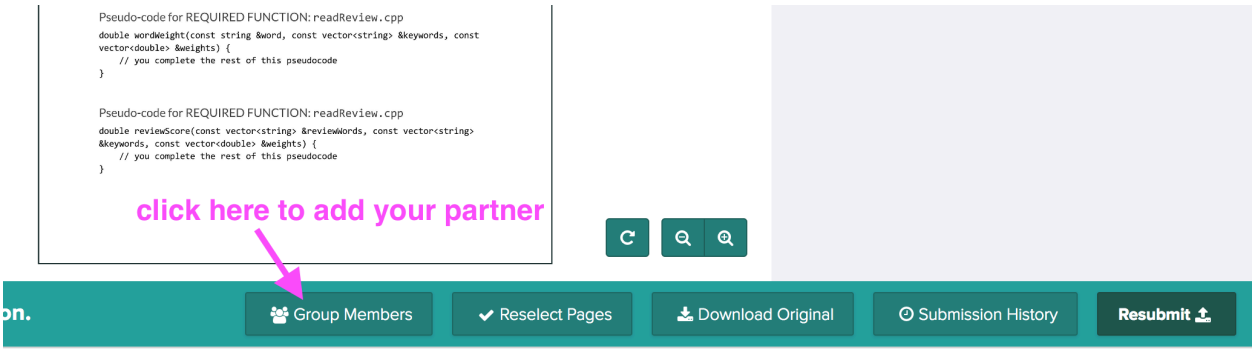




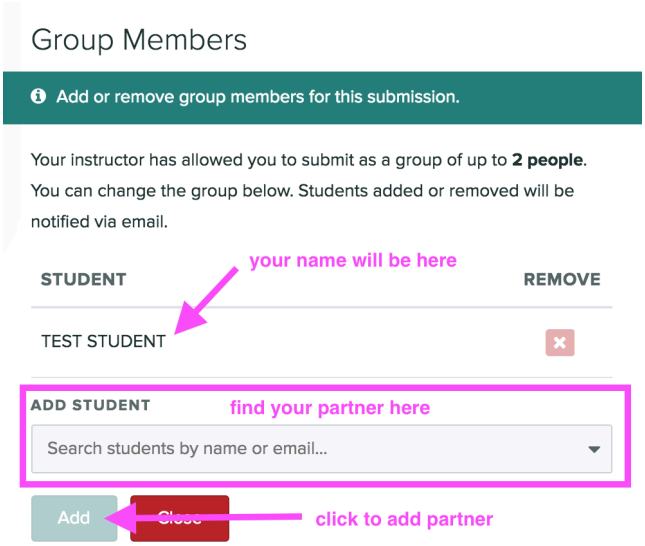
After you have assigned the correct pages to the correct questions, click the "Submit" button in the bottom right corner of the page.

Registering your partnership on Gradescope

If you are working with a partner, make sure both of your names are in the title for plot #5. Then, use "group submission" on Gradescope. Have one member of the partnership submit the `environmentalSummary.pdf` file as described above. After you submit the `.pdf` file, click "Group Members" to add your partner, as shown below:



This will pop up a little box where you can search for your partner's name and then add them to your group of two.



Your partner will receive an email verifying that they have been added to your group. Make sure your partner gets this email!

Style and Commenting

Maximum Score: 10 points

2 pts - Each submitted file has Name, Partner Uniqname (or "none"), Lab Section Number, and Date Submitted included in a comment at the top
2 pts - Comments are used appropriately to describe your code (e.g. major steps are explained)
2 pts - Indenting and white space are appropriate (including functions are properly formatted)
2 pts - Variables are named descriptively
2 pts - Other factors (Variable names aren't all caps, etc...)

Appendix: Test Case

A test case is provided for you to use in testing your program. The files used in the test case are:

- Wind speed file: `windSpeedTestCase.csv`
- Wave height file: `waveHeightTestCase.csv`
- Buoy measurement file: `buoyTestCase.csv`
- Latitude data points file: `lat.csv`
- Longitude data points file: `lon.csv`
- Summary of conditions figure: `environmentalSummaryTestCase.pdf`

These files are available in the Project 3 Google Drive folder. To test your program, call your function with the above filenames and the following case parameters:

- Minimum wind speed = 2.5 m/s
- Maximum wind speed = 7.5 m/s
- Maximum wave height = 5.5 m
- Maximum wave height risk = 80%
- Deck height = 17.2 m

See the next page for an example of a test script.

```
% Sample code to test the two functions for Project 3 -- PUT THIS IN A TEST SCRIPT
(but don't copy/paste from this document because the formatting will be weird and
make it not run in Matlab)

% Filenames for data that changes
filenameWind = 'windSpeedTestCase.csv';
filenameWave = 'waveHeightTestCase.csv';
filenameBuoy = 'buoyTestCase.csv';

% Constraint limits
windSpeedMin = 2.5;             % Minimum wind speed (m/s)
windSpeedMax = 7.5;             % Maximum wind speed (m/s)
waveHeightMax = 5.5;            % Max sig wave height (m)
waveHeightRisk = 80;            % Max sig wave height risk (%)
deckHeight = 17.2;              % Height of deck above water (m)

% Analyze this location
[c1, c2, c3, c4, c5] = analyzeWindFarm(filenameWind, filenameWave, filenameBuoy,
windSpeedMin, windSpeedMax, waveHeightMax, waveHeightRisk, deckHeight );

% Make the summary figure
makePlots(filenameWind, filenameWave, filenameBuoy, windSpeedMin, windSpeedMax,
waveHeightMax);
```

Your `analyzeWindFarm.m` function should return `[1,1,1,1,0]` as the results of evaluating each of the 5 constraints. Your `environmentalSummary.pdf` file should look like Fig. 8, with your full name (or your and your partner's name) as part of the title for Plot 5.