Problem Set 01 · MATLAB Calculations

Instructions

1. Read the programming standards document included in the Assignment Files for this assignment. You must follow all course programming standards guidelines for this and all subsequent assignments.
2. Read each problem carefully before starting your work. You are responsible for following all instructions within each problem.
3. Each problem will ask you to create one or more files, which you will submit via Blackboard. Use problem-specific templates when provided. Name your files to match the format in the table below, where *yourlogin* is your Purdue Career Account login.

Item	**Type**	**Template Provided**	**Deliverable to include in Submission**
Problem 1: Script – Road Salt	Individual	Yes	☐ PS01_salt_*yourlogin*.m
Problem 2: Vectors – Lift Crane	Individual	Yes	☐ PS01_craneCalcs_*yourlogin*.m
Problem 3: Matrices – Brake Data	Individual	Yes	☐ PS01_brakematrix_*yourlogin*.m

4. Save all files to your Purdue career account in a folder specific to PS01.
5. When you are ready to submit your assignment,
   - Compress all the deliverables listed in Step 3 into one zip file and name it **PS01_*yourlogin*.zip**. Be sure that you
     i. Only compress files using **.zip** format. No other compression format will be accepted.
     ii. Only include deliverables. Do **not** include the problem document, blank templates, etc.
   - Submit the zip file to the Blackboard drop box for PS01 before the due date.
   - You can resubmit the assignment as many times as needed until the due date, but **only your final submission will be graded.**
6. After grades are released for this assignment, access your feedback via the assignment rubric in the My Grades section of Blackboard.

Notes Before You Start

Go to **Getting Help** > **MATLAB Resources** in your ENGR 132 Blackboard course to find extra help for using MATLAB. The file *MATLAB Reference 1: Introduction to MATLAB* may be helpful on this assignment.

Problem 1: Script – Road Salt

Individual

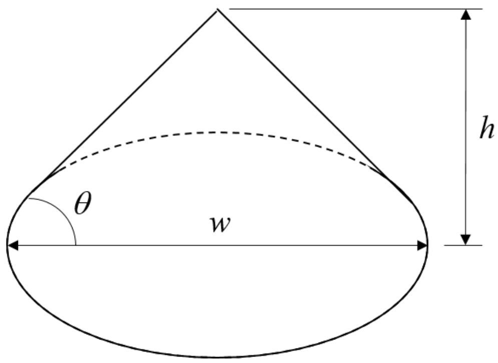
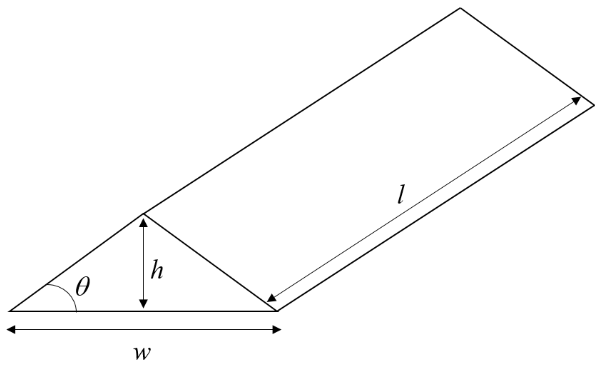
Learning Objectives

Below are learning objectives that may be used to assess your work on this problem. Use the links to find the full evidence lists for each topic.

Calculations	01.01 Perform algebraic computations with scalars
	01.02 Employ order of operations to perform calculations
	01.03 Use built-in functions to perform algebraic and trigonometric calculations
Variables	02.01 Apply MATLAB rules and "good" programming conventions to name variables
	02.02 Assign a scalar to a variable
Scripts	04.01 Create a script that adheres to programming standards
	04.02 Execute a script from the MATLAB Command Window
Text Displays	05.01 Format text output for technical presentation
	05.02 Suppress output from a command once debugging is complete

Problem Setup

You are an engineering intern for the Indiana Department of Transportation (INDOT). They are studying their storage capabilities for road salt, which they apply to icy roads. Salt is stored in large piles. The piles' shapes are either conical or windrow, which is a triangular prism shape.

Figure 1: Conical Pile Geometry 	Figure 2: Windrow Pile Geometry 

Road salt has known properties. Salt poured into a pile will naturally form sides that slope at an angle of 32 degrees (θ). This angle is known as the angle of repose. The salt's density is 80 pounds per cubic foot.

INDOT wants to study the storage of salt in uniform piles. Your supervisor gives you the following information about each pile type.

Conical Pile	**Windrow Pile**
width (w) = 19.50 meters height $(h) = \frac{w \tan \theta}{2}$ $volume = \frac{\pi w^2 h}{12}$	width (w) = 19.50 meters length (l) = 100 meters height $(h) = \frac{w \tan \theta}{2}$ $volume = \frac{whl}{2}$

Your task is to create a script that will

- calculate the height, volume, and weight of salt in a single conical pile,
- calculate the height, volume, and weight of salt in a single windrow pile,
- calculate how much salt, in metric tons, can be stored in 10 conical piles,
- calculate how much salt, in metric tons, can be stored in 2 windrows,
- print the height and weight of one conical pile to the Command Window,
- print the height and weight of one windrow pile to the Command Window, and
- print to the Command Window the weight of the 10 conical piles and the 2 windrows.

Your supervisor provided you with a script template, which has commented sections that you must use. Your supervisor expects your calculations and variable assignments **to follow programming standards** so that other employees can easily understand and apply the script to other projects.

Problem Steps

1. **Prepare a script file.**
    a. Open the MATLAB template file **PS01_salt_template.m**. Rename the file by replacing ***template*** with your Purdue Career Account login.
    b. Complete the header information (see Programming Standards for help).
    c. Read and understand the commented sections. You must write your code in the appropriate sections.
2. **Write the script.** Follow the instructions in the script comments. Complete all the required steps. Follow programming standards.
3. **Finalize the script.** Run and debug your script until it works properly. Once your script is running properly, you should only see your printed statements in the Command Window. If you see output from your variable assignments or calculations outside the print statements, then you need to suppress the output of those variables.
4. **Alter the script.** Your supervisor asks you to change your script. Based on a change in space requirements, they now want to look at piles with the following geometry:

Conical Pile	Windrow Pile
w = 35.00 meters number of piles = 5	w = 22.25 meters l = 50 meters

    a. Change the appropriate variable values in your script to reflect the new geometry for each pile.
    b. Save and re-run your script to get the new results.
5. Debug as necessary and finalize your script for submission.

Reference: http://www.saltinstitute.org/wp-content/uploads/2013/09/Salt-Storage-Handbook-2015.pdf

Problem 2: Vectors – Lift Crane

Individual

Learning Objectives

Below are learning objectives that may be used to assess your work on this problem. Use the links to find the full evidence lists for each topic.

Calculations	01.01 Perform algebraic computations with scalars
	01.02 Employ order of operations to perform calculations
	01.04 Perform element-by-element operations
	01.05 Identify when and explain why an intended element-by-element operation will not execute properly
Variables	02.03 Assign a row or column vector to a variable using multiple methods
Arrays	03.02 Copy a single element from an array and assign it to a variable
Scripts	04.01 Create a script that adheres to programming standards
	04.02 Execute a script from the MATLAB Command Window

Problem Setup

You work for a manufacturing company that owns an interior crane system. The cranes lift objects of various masses onto platforms at a factory and can operate up to 12 hours a day. You have the following data about crane performance at the factory. Your task is to perform some vector calculations using the data.

Table 1. Crane Information

	Lifted mass, kilograms (m)	Lifting distance, meters (d_L)	Lifting time, seconds (t_L)	Return time, seconds
Crane 1	500	1.2	1.2	2.4
Crane 2	600	4.2	2.4	4.8
Crane 3	700	3.0	2.0	4.0
Crane 4	800	1.5	1.5	3.5
Crane 5	1000	2.5	3.0	9.0

Problem Steps

1. **Prepare your script file**.
   a. Open the MATLAB template file **PS01_craneCalcs_template.m**. Rename the file by replacing ***template*** with your Purdue Career Account login.
   b. Complete the header information (see Programming Standards for help).
   c. You must write your code in the appropriate sections and **follow good programming standards** throughout the script.
2. **Initialize the vectors and scalars**. In the `INITIALIZATION` section of the template, create the following variables using the information in the Problem Setup:
   a. A column vector of the masses, named `liftMass`

b. A column vector of the lifting distances, named `liftDist`

c. A column vector of the lifting times, named `liftTime`

d. A column vector of the return times, named `retTime`

e. A scalar value of the operating hours, named `opHrs`

f. Also assign the acceleration due to gravity, which is 9.81 m/s^2, to a variable named `grav`.

3. **Perform calculations**. In the `CALCULATIONS` section of the template, use the initialized variables to perform vector operations that will produce a single vector for each of the following equations

   a. Lifting velocity of an object, in meters per second: $V_L = d_L/t_L$

   b. Force on an object due to gravity, in newtons (N): $F_g = mg$

   c. Power necessary to lift an object, in watts (W): $P_L = (F_g * d_L)/t_L$

   d. The number of lifts possible in a 12-hour day, if one lift includes both the lifting time and return time.

   Each calculation should produce a resulting vector that has one element for each crane. Only use period characters when they are required to get an element-by-element result; do not use a period character if an element-by-element result can be found without it.

   In the comment for each result, **include the vector dimension as well as units**.

4. **Add new information and calculations to the script**. Maintaining the cranes' operation is necessary for you company. They wants to invest money to create a long-term crane repair fund. Use the compound interest formula when interest is calculated once per year

$$A = P(1 + r)^t$$

   Where A is the value of the investment including interest, P is the principal (the initial amount of money put into the fund), r is the annual interest rate in decimal form, and t is the number of years of the investment.

   You need to examine the possible fund values over time using vector operations.

   a. You first need to define the following vectors:

      i. Principal: A row vector of 11 equally-spaced elements from \$1000 to \$2000

      ii. Interest rates: A row vector of 11 elements that starts at 0.1 and decreases to 0.025 in increments of 0.0075

      iii. Investment time: A row vector of 11 elements that starts at 1 year and increments by 2 until the final element is 21 years

      NOTE: Use at least two different methods of vector creation in step 4.a.

   b. Perform the following calculations using vector operations and the three vectors in Step 4.a. Only hardcode constants in the equations. Do not hardcode a scalar value that is an element in one of the vectors; use vector indexing to identify the single element to use in the calculation. Only use period characters when they are required to get an element-by-element result; do not use a period character if an element-by-element result can be found without it.

      i. If the principal is \$1000 and the interest rate is 0.025, then what is the total value of the investment fund for each year in the investment time vector?

      ii. If the principal is \$1500 and the time of investment is 5 years, what is the total value of the investment for each potential interest rate?

      iii. If the interest rate is 0.07 and the time of investment is 17 years, then what is the total value of the investment for each initial principal amount?

5. Debug as necessary and finalize your script for submission.

Problem 3: Matrices – Brake Data

Individual

Learning Objectives

Below are learning objectives that may be used to assess your work on this problem. Use the links to find the full evidence lists for each topic.

Calculations	01.03 Use built-in functions to perform algebraic and trigonometric calculations
Variables	02.04 Assign a matrix to a variable
Arrays	03.01 Convert a row vector to a column vector (or vice versa)
	03.02 Copy a single element from an array and assign it to a variable
	03.03 Copy an array from a larger array and assign it to a variable
	03.04 Concatenate arrays
	03.05 Replace elements of arrays
	03.06 Reference an array using appropriate indexing
Scripts	04.01 Create a script that adheres to programming standards
	04.02 Execute a script from the MATLAB Command Window

Problem Setup

You have experimental data for a mechanical brake design. The experiment tested the stopping time, in seconds, for the brake under three (3) different conditions. An analyst performed the test four (4) times. The results are in Table 1.

Table 1. Experimental brake data, in seconds

	Test 1	**Test 2**	**Test 3**	**Test 4**
Condition 1	0.59	0.62	0.6	0.6
Condition 2	0.97	0.91	0.98	0.95
Condition 3	1.25	1.15	1.1	1.12

You will use this table of data to practice matrix manipulations using MATLAB.

Problem Steps

1. **Prepare a script file.**
    a. Open the MATLAB template file **PS01_brakematrix_template.m**. Rename the file by replacing ***template*** with your Purdue Career Account login.
    b. Complete the header information.
    c. Review the comments and commands in the template.
    d. Remember to follow good programming standards throughout the script. Remember that array indices are not considered hardcoding.

2. **Add the experimental data to the script.** Define a matrix in the INITIALIZATION section of the code that recreates the table above. Assign the matrix to a variable named `brakeData`.
3. **Comment the provided matrix manipulation commands.** A set of matrix manipulations have been included in the script template under MATRIX MANIPULATIONS. Run the script, and then add a comment on each line explaining what that command does. If a line produces a MATLAB error, then explain the error and suppress the whole line of code with a `%`.
4. **Interpret matrix manipulations.** In the CALCULATIONS section, perform the matrix manipulations necessary to complete the tasks below. Each task requires only one line of code. Use descriptive variable names, comment each line, and do not hardcode any value that is part of `brakeData`. Run your script to check your work.
    a. Assign all the stopping times in Test 2 to one variable.
    b. Use array indexing to assign the stopping time of Condition 3 in Test 4 to one variable.
    c. Replace the first condition of Test 4 with a corrected value of 0.61 using array indexing.
    d. You have a 5th test to add to the data. Create a column vector with the following stopping times, in seconds.

	Test 5
Condition 1	0.58
Condition 2	0.93
Condition 3	1.2

    e. Create a new data matrix that concatenates the Test 5 vector to the right of the `brakeData` matrix.
5. Debug as necessary and finalize your script for submission.