A10 · User-Defined Functions

Introduction

Assignment Goals

This assignment focuses on creating user-defined functions for engineering-based contexts.

Successful Completion

This assignment has **2** problems. The deliverables list contains everything you are expected to submit individually.

Submit Problems 1 and 2 to the Gradescope online assignment **A10 – Context Problems**		
Problem	**Type**	**Deliverables**
Problem 1: Hay Bale Ejector	Individual	☐ A10Prob1_baler_*login*.m
Problem 2: Sit-Stand Device	Individual	☐ A10Prob2_main_*login*.m ☐ A10Prob2_lengths_*login*.m ☐ A10Prob2_springs_*login*.m

1. Read *Notes Before You Start*, on **Page 1**.
2. Read each problem carefully. You are responsible for following all instructions within each problem.
    a. Each problem has a team planning component that you are expected to submit separately from your deliverables. Read the instructions within each of the problems.
3. Complete the problems using the problem-specific m-file templates provided in the assignment download. Replace *template* in the filename with your Purdue Career Account login.
4. When your work is complete, confirm your deliverables are submitted to Gradescope.

Learning Objectives & Grading

This course uses learning objectives (LOs) to assess your work. You can find a list of the course LOs on Brightspace (Content > Key Course Info > Learning Objectives). Review the assignment grading for each problem in this assignment, which starts on **Page 8**. This outlines how your work will be graded for each problem.

Notes Before You Start

Gradescope

You will submit all your deliverables to Gradescope for grading. This homework has **one** Gradescope submission assignment for the individual submissions, plus a **second** assignment for the team planning component:

- **A10 – Context Problems**: submit your deliverables for Problems 1 and 2 as an individual.
- **A10 – Team Planning**: submit your team plans for Problems 1 and 2 as a team.

Problem 1: Hay Bale Ejector

Introduction

This problem gives you practice writing a user-defined function that checks for valid inputs and returns a calculated value. Be sure to follow good programming standards in your function.

Problem

You are an agricultural engineer studying projectile motion for a hay bale ejector. You must create a user-defined function that uses the equations below to predict the maximum height the bale will travel (y) when ejected and the horizontal distance traveled when the maximum height is reached (x).

Vertical Position	Horizontal Position
$y = 1.25 + (v_0 \sin\theta)t - 0.5gt^2$	$x = (v_0 \cos\theta)t$

Where
- y = vertical position of the hay bale (m)
- x = horizontal position of the hay bale (m)
- v_0 = initial velocity of hay bale when it is ejected (m/s)
- θ = launch angle that the hay bale leaves the ejector (deg)
- t = time since the hay bale was ejected (sec)
- g = gravitational constant, 9.8 m/s^2

For this problem, you will need to calculate x and y using the time when the hay bale reaches its maximum height:

$$t = t_{max} = \frac{(v_0 \sin\theta)}{g}$$

Your function must do the following:

- Accept two input arguments: initial velocity (m/s) and launch angle (deg).
- Return two output arguments: the maximum height of the bale (m) and the horizontal distance at the maximum height (m).
- The initial velocity must be between 9.5 and 13.25 m/s, inclusive of both. The angle of launch must be between 30 and 60 degrees, inclusive of both.
  - If either input is invalid, return an error message indicating which input is invalid. Do not return any output arguments if an input is invalid.
- Display the maximum height and corresponding horizontal distance to the Command Window. Be sure to properly manage the decimal display of the values.

When your function is running properly, run it with the following inputs:

Test case	Input: initial velocity (m/s)	Input: launch angle (deg)
1	15	35
2	10	65
3	11.4	45

In the `RESULTS` section of your function, copy and paste as comments the **function call** AND the **information displayed** to the Command Window for each test case.

Instructions

1. Read through the entire problem statement.

2. **With your teammates**: develop and document a plan to solve this problem.
    a. Understand the expectations of the problem.
    b. Discuss strategies for solving the problem. This can include citing examples from class notes, drawing pictures, outlining a plan using text or pseudocode, etc. **DO NOT SHARE CODING SOLUTIONS.**
    c. Submit your plan to the team assignment in Gradescope
        1. Open the Gradescope assignment **A10 – Team Planning**.
        2. In the area for Problem 1:
            a. Enter the names of your teammates who participated in the planning.
            b. Enter a brief description of your team's plan to solve the problem. The plan should be connected to the problem and have at least 2-3 steps. It should not be a detailed explanation of every step necessary to solve the problem.
            c. If you have image files, etc., that you would prefer to share, then you may add them in the *Optional* file submission area.
        3. Save your results.
    d. Add your teammates to the submission (or double-check everyone is added if you completed this step already). Select 1 team member to submit the plan. Work together to make sure it is done correctly.
        1. Click **Submit & View Submission** at the bottom of the assignment
        2. Add all teammates to the group (Gradescope instruction link)
        3. All teammates confirm that you get an email confirming the submission and verify that you can see the submission in your Gradescope.
3. **Individually**:
    a. Complete your function, run it to get your results, paste the text results as comments into the function and save the figure window as an image.
        - The team plan is an initial start on the problem. It may not be completely correct, and you may find flaws in the plan once you start coding. You should make any individual changes that are necessary to obtain the best solution. You will be assessed on your individual solution to the problem.
    b. Cite the teammates you worked with in your function header if their help changed how you decided to solve the problem.
        1. Make sure you also completed the rest of the function header.
    c. Submit your properly named files to **A10 – Context Problems** in Gradescope.

Problem 2: Sit-Stand Device

Introduction

This problem allows you to build a set of user-defined functions that work together to solve an engineering problem. Be sure to follow good programming standards in your functions.

Problem

Introduction

A sit-to-stand assist device can serve the needs of people suffering from muscle weakness due to age or disabilities that make sit-to-stand a difficult functional task. The company you are working for is designing a passive gravity-balancing assist device for sit-to-stand motion. Fig. 1 provides an image of a prototype of such a machine.

Figure 1. Photographs of the new prototype with the subject in sit and stand positions

To make a gravity-balanced assistive device for the human body, the following procedure is used:

(i) Determine the center of mass of the system, i.e., device and the human body, using auxiliary parallelograms;

(ii) select springs to connect to the system center of mass such that the total potential energy of the system is invariant with configuration.

This method allows one to physically determine the system center of mass and connect this point to the inertially-fixed frame through springs.

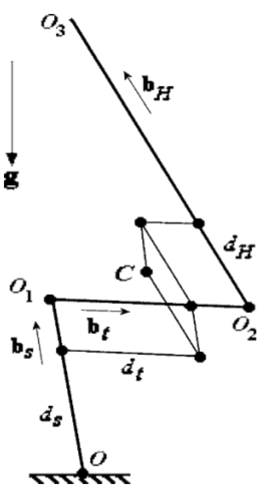


Figure 2. The 3-link human body & device with auxiliary parallelograms to determine the center of mass of the system.

Determine the center of mass of the system using auxiliary parallelograms.

To calculate the center of mass of a person using the sit-to-stand device, model the human body as three sections, each connected at a joint. The shank refers to the mass and length of the leg from the knee to ankle. The thigh refers to the mass and length of the leg from the knee to hip. The HAT (head, arms, trunk) refers to the body's mass and length above the hip. Fig. 2 shows a free-body diagram representation of the human body model. OO_1 represents the shank, O_1O_2 represents the thigh, O_2O_3 represents the HAT. The subscripts for measurements in each section are s, t, and H, respectively.

The location of the center of mass for the model of the human body shown in Fig. 3 from the point O is defined by $\mathbf{r}_{OC}$. Its expression is given by:

$$\mathbf{r}_{OC} = d_s\mathbf{b}_s + d_t\mathbf{b}_t + d_H\mathbf{b}_H \qquad \text{(eq. 1)}$$

where d_s, d_t, and d_H are scaled lengths (m) and $\mathbf{b}_s$, $\mathbf{b}_t$, and $\mathbf{b}_H$ are unit vectors in the direction of each of the body sections. The scaled-length components of $\mathbf{r}_{OC}$ can be found using the following equations:

$$d_s = \frac{1}{M}(m_t l_s + m_H l_s + m_s l_{cs}) \qquad \text{(eq. 2)}$$

$$d_t = \frac{1}{M}(m_H l_t + m_t l_{ct}) \qquad \text{(eq. 3)}$$

$$d_H = \frac{1}{M}(m_H l_{cH}) \qquad \text{(eq. 4)}$$

$$\text{and} \quad M = m_s + m_t + m_H \qquad \text{(eq. 5)}$$

where m_i is the mass of the body section in kilograms, l_i is the length of the body section in meters, l_{ci} is the distance in meters between C and the section's center of mass, and M is the total mass of the system (which assumes the mass of the device is negligible).

Select springs to connect to the system center of mass

The human body and the device can be gravity-balanced by attaching four springs to the system as shown in Fig. 3. The total potential energy of the system consists of gravitational and elastic energies due to the springs. The desired stiffness of the springs for gravity balancing of the system are as follows:

$$k = \frac{Mg}{d} \qquad \text{(eq. 6)}$$

$$k_1 = \frac{kd_s}{l_s - d_s} \qquad \text{(eq. 7)}$$

$$k_2 = k_1 = \frac{kd_s}{l_s - d_s} \qquad \text{(eq. 8)}$$

$$k_3 = \frac{kd_t}{l_t - d_t} \qquad \text{(eq. 9)}$$

where g is the gravitational constant 9.8 m/s^2 and d is an adjustable length in meters that is chosen such that we have optimal values for the stiffness of the springs and the extension of the spring k. The spring constants are in units N/m. For the purposes of our tests, we will assume d = 0.75m.

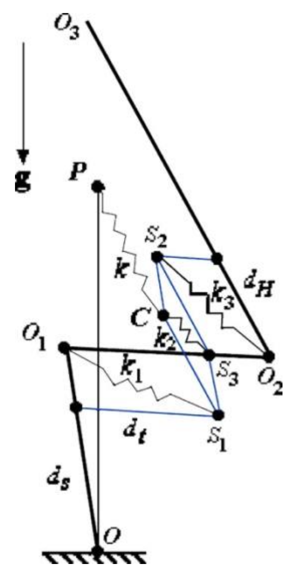


Figure 3. Spring attachments of the 3-link human body.

Problem Steps

You will create 3 functions: one main function and two subfunctions, one to calculate the scaled-length values and one to calculate the spring stiffness values. Table 1 shows a sample patient's measurements that should be used to design and test your functions.

Table 1. Sample body measurements

Section	Length of body section (m)	Mass of body section, m (kg)	Center-of-mass location (m)	Adjustable length, d (m)
Shank, *s*	$l_s = 0.421$	$m_s = 3.1$	$l_{cs} = 0.55 l_s$	0.75
Thigh, *t*	$l_t = 0.432$	$m_t = 7.39$	$l_{ct} = 0.59 l_t$	
HAT, *H*	$l_H = 0.8$	$m_H = 24.13$	$l_{cH} = 0.41 l_H$	

In your assignment files, you were given one template for this problem. Adapt the template for each of the deliverables in this problem:

- ☐ A10Prob2_main_*login*.m
- ☐ A10Prob2_lengths_*login*.m
- ☐ A10Prob2_springs_*login*.m

Note: Not every function will need every section provided in the template. Every function requires a complete header.

Scaled Length UDF – create a UDF to calculate the scaled lengths

Create a user-defined function named **A10Prob2_lengths_login.m** that:

a. Accepts 3 inputs: mass of body sections (vector), length of body sections (vector), center-of-mass locations of body sections (vector)
b. Returns 2 output arguments: the scaled lengths (vector) and total mass of the body (scalar)
c. Your function is not required to confirm that the inputs are valid.

Use the sample patient measurements to test and debug your function.

Spring Stiffness UDF – create a UDF to calculate the spring stiffness constants

Create a user-defined function named **A10Prob2_springs_*login*.m** that:

a. Accepts 3 inputs: total mass of the body (scalar), length of body sections (vector), the scaled lengths (vector)
b. Returns 1 output argument: the spring stiffness constants for the springs (vector)
c. Your function is not required to confirm that the inputs are valid.

Use the sample patient measurements to test and debug your function.

Main Function – create a UDF to call the two subfunctions and display information

Create a user-defined function named **A10Prob2_main_*login*.m** that:

a. Accepts no inputs and returns no outputs
b. Initializes the values and vectors from Table 1 to use as inputs in the subfunctions.
c. Call the two subfunctions appropriately to get the spring stiffness constants.
d. Displays the 4 spring stiffness constants to the Command Window using professional formatting.
e. Only performs the calculation to determine the l_{ci} values – all other calculations occur in the subfunctions as described above.

Once your main function and subfunctions are working properly, run you main function. In the `RESULTS` section of your main function, copy and paste as comments the **main function call** AND the **information displayed** to the Command Window.

Instructions

1. Read through the entire problem statement.
2. **With your teammates**: develop and document a plan to solve this problem.
    a. Understand the expectations of the problem.
    b. Discuss strategies for solving the problem. This can include citing examples from class notes, drawing pictures, outlining a plan using text or pseudocode, etc. **DO NOT SHARE CODING SOLUTIONS.**
    c. Submit your plan to the team assignment in Gradescope
        1. Open the Gradescope assignment **A10 – Team Planning**.
        2. In the area for Problem 2:
            a. Enter the names of your teammates who participated in the planning.
            b. Enter a brief description of your team's plan to solve the problem. The plan should be connected to the problem and have at least 2-3 steps.

It should not be a detailed explanation of every step necessary to solve the problem.

c. If you have image files, etc., that you would prefer to share, then you may add them in the *Optional* file submission area.

3. Save your results.

d. Add your teammates to the submission (or double-check everyone is added if you completed this step already). Select 1 team member to submit the plan. Work together to make sure it is done correctly.

1. Click **Submit & View Submission** at the bottom of the assignment
2. Add all teammates to the group (Gradescope instruction link)
3. All teammates confirm that you get an email confirming the submission and verify that you can see the submission in your Gradescope.

3. **Individually**:

a. Complete your function, run it to get your results, paste the text results as comments into the function.

- The team plan is an initial start on the problem. It may not be completely correct, and you may find flaws in the plan once you start coding. You should make any individual changes that are necessary to obtain the best solution. You will be assessed on your individual solution to the problem.

b. Cite the teammates you worked with in your function header if their help changed how you decided to solve the problem.

1. Make sure you also completed the rest of the function header.

c. Submit your properly named files to **A10 – Context Problems** in Gradescope.

Reference:

Design of a Passive Gravity-Balanced Assistive Device for Sit-to-Stand Task [https://doi.org/10.1115/1.2216732]

Using Table 1 (p. 1123), equations 4 & 5 (p. 1124) and 16 (p. 1125), Figure 4 & 5 (p.1124), Figure 12 (p. 1127)

Confirm Your Submission

You should save your progress on each question in a skills problem so that you do not lose your progress. To confirm your answers, click the **Submit & View Submission** button at the bottom of the questions in Gradescope (or select the assignment name from the Gradescope dashboard, if you have already saved your answers and navigated away from the original submission page).

Individual deliverables

Confirm that your submission for **A10 – Context Problems** includes

- ☐ The expected deliverables and results.
- ☐ Correct file names for any submitted files, including your Career Account login at the end where required.

You can resubmit your work as many times as you want, but only the final submission will be graded.

Do NOT upload any document not listed in the deliverables. Do not upload temporary versions of m-files (*.m~ or *.asv) – these files will be ignored by Gradescope and not included in your upload.

Team plans

Confirm that your submission for **A10 – Team Planning** includes

- ☐ The names of all your teammates who participated in each problem's planning;
- ☐ A brief description of the team's plan for each problem;
- ☐ All team members included in the group submission.

Assignment Grading

Your work will be graded using the evidences given in the course learning objectives. Familiarize yourself with the LOs and their evidences listed for each problem, which are below.

Find the list of the course LOs, with evidences, on Brightspace (Content > Key Course Info > Learning Objectives).

Team Planning

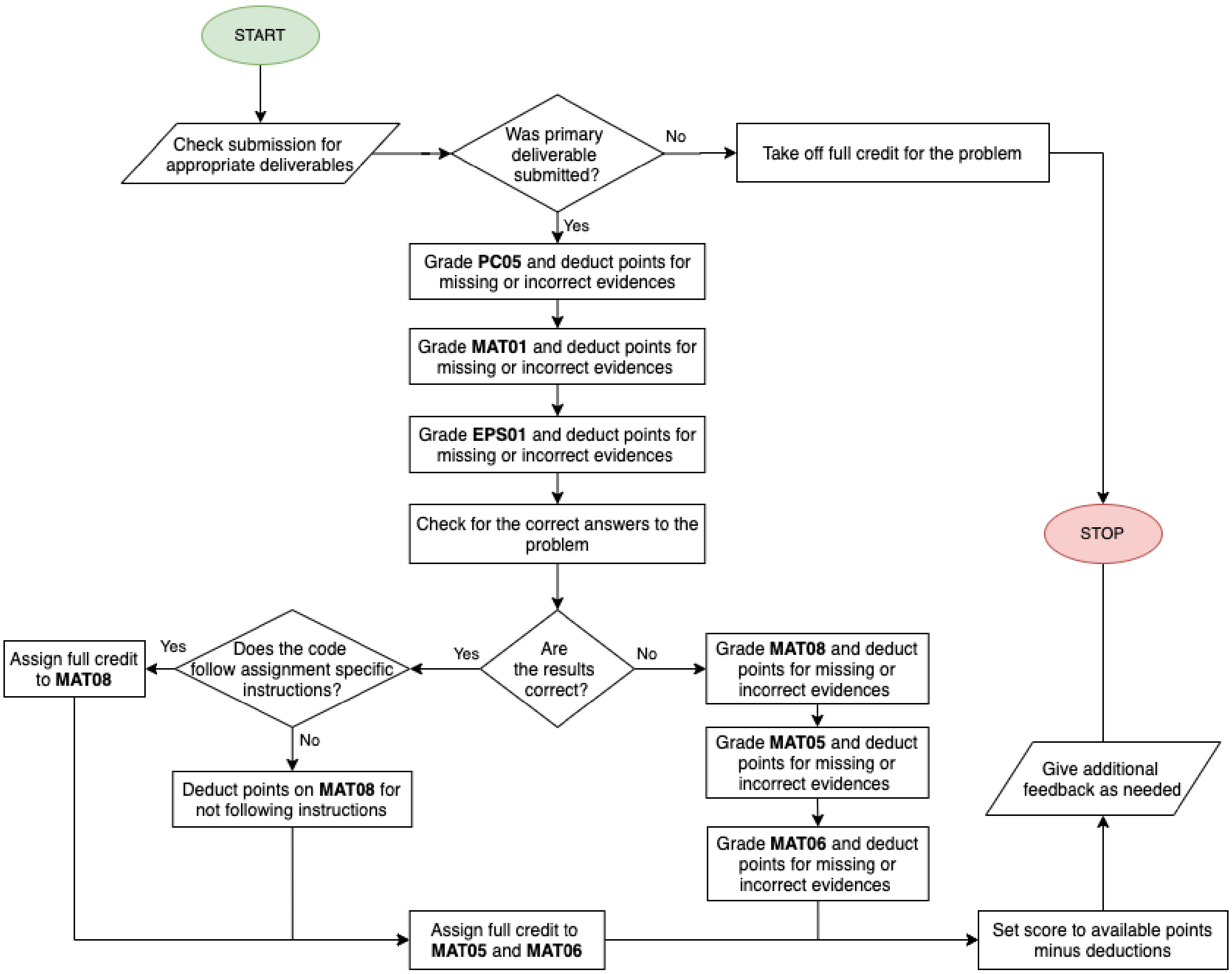
Each problem's plan is worth 1 point.

Problem 1

Note: PC05 evidences are only deductions since you are expected to follow the assignment instructions.

	PC05	MAT01	EPS01	MAT08	MAT05	MAT06
(1)	-5	0.3	0	0.3	0.3	0.2
(2)	0	0	0	0.5	0.3	0.2
(3)	-1.25	0.1	0.2	0.3	0.3	0.2
(4)	-0.25	0.1	0.2	0	0.3	0.2
(5)	0	0.1	0.2	0	0.3	0.2
(6)	0	0.1	0	0	0	0
(7)	0	0.1	0	0	0	0
(8)	0	0	0	0	0	0

Grading Process



Problem 2

Note: PC05 evidences are only deductions since you are expected to follow the assignment instructions. Note: In PC05 (8), you will lose 3 points for each subfunction missing from your submission and 4 points if the main function is missing.

	PC05	MAT01	EPS01	MAT08	MAT05	MAT03
(1)	-10	0.5	0	0.8	0.5	0
(2)	0	0	0	1	0.5	0.5
(3)	-2.5	0	0.4	0.6	1	0.5
(4)	-0.4	0.2	0.4	0	0.2	0.5
(5)	0	0.2	0.3	0	1	0
(6)	0	0.2	0	0	0	0
(7)	0	0.2	0	0	0	0
(8)	*see note	0.5	0	0	0	0

Grading Process