A11 · Complex Loops

Introduction

Assignment Goals

Two skills questions will give you practice working with complex looping structures. You will then work with complex loops using engineering context.

Successful Completion

This assignment has **3** problems. The deliverables list contains everything you are expected to submit individually.

Submit Problems 1 and 2 to the Gradescope online assignment **A11 – Skills Problems**		
Problem	**Type**	**Deliverables**
Problem 1: MATLAB Skills – Removing Loops	Individual	☐ A11Prob1_noLoop_*login*.m ☐ Requested results and information
Problem 2: MATLAB Skills – Nested Loops	Individual	☐ A11Prob2_nested_*login*.m ☐ Requested results and information
Submit Problem 3 to the Gradescope programming assignment **A11 – Context Problems**		
Problem	**Type**	**Deliverables**
Problem 3: Pixel Filtering	Individual	☐ A11Prob3_pixelFilters_*login*.m ☐ Supporting file(s): ○ A11Prob3_figure_*login*.png ○ Data_pixels.csv

1. Read *Notes Before You Start*, on **Page 2**.
2. Read each problem carefully. You are responsible for following all instructions within each problem.
    a. Problem 3 has a team planning component that you are expected to submit separately from your deliverables. Read the instructions within that problem.
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 and your team plans 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 9**. This outlines how your work will be graded for each problem.

Notes Before You Start

Helpful MATLAB Commands

Learn about the following built-in MATLAB commands, which might be useful in your solutions:

`end` (as an index value in an array)

String Arrays in MATLAB

MATLAB allows you to create a string array, which is an array where each element in the array is a string instead of a number. It is important that you use proper formatting in a string array. Enter this command into MATLAB:

```
purdue_astronauts = ["Neil Armstrong","Eugene Cernan","Loral OHara","Scott Tingle"]
```

The double quotes are required; single quotes will not create a string array. You can use array commands, such as `length` and `size`, on string arrays.

Gradescope

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

- **A11 – Skills Problems**: submit your deliverables for Problems 1 and 2
- **A11 – Context Problems**: submit your deliverables for Problem 3
- **A11 – Team Planning**: submit your team plan for Problem 3 as a team

Problem Generator File

In the assignment folder, you will see a file named **A11_skills.p**. This is a MATLAB function file that generates problem information for each skill problem in this assignment.

Need help using this file? Refer to the *Notes Before You Start* of A04 or A05, or review the A00 Activity from Class 1B to see fully-worked examples.

Problem 1: MATLAB Skills – Removing Loops

Introduction

This problem allows you to practice removing a `for` loop with vector indexing from code and replacing it with appropriate vector operations. You will also use skills from relational and logical operations to identify elements for the calculations.

Problem

The instruction text will provide you with the code for a `for` loop with vector indexing. This code assigns a vector of values, and then the loop moves through each element in the vector. Depending on the value of the element, the code will perform one of two possible calculations. The calculation is then added in the index location of the final results vector, named `newvec`.

You must write a script that starts with the same original vector variable, `vec`, and will produce the same final vector as `newvec`, **without** using loops (for loops or while loops) and without using selection structures. Name the final vector of your no-loop code `newvec_noLoop`. Submit your script to Gradescope.

Instructions

1. Type this command into the MATLAB Command Window prompt:

   ```
   >> A11_skills(PUID, 1)
   ```

   Remember to replace PUID with your 8-digit Purdue University ID number (leave off the leading 00).
2. Read the written instructions that appear in the Command Window.
   a. Do not include the original code in your script. You will paste that code into the instruction text box in Gradescope as reference.
   b. Initialize `vec` in your script with the vector you were given in the instruction text.
   c. Your script must work for any vector `vec`, not just the one you were given (i.e., avoid hardcoding your solution for your specific values in `vec`).
   d. Use the included template to write the requested code.
   e. Programming standards will not be assessed in this problem. Do not include comments in your solutions. Properly name the file.
3. Submit your work in Gradescope:
   a. Open Gradescope > **A11 – Skills Problems** and find the set of boxes that belong to **Q1** you want to submit. Enter the required information along with your answer:
      - ☐ **Function call**. Copy the command that you entered at the command prompt to call the function and paste the full command into this box. Be sure your PUID is included.
      - ☐ **Instruction text**. Copy the instruction text that is displayed in the Command Window. Paste it into this box. Include all text provided.
      - ☐ **Solutions**. Submit your script. Follow any additional instructions provided.
   b. When you have entered all the required information for the question, click the **Save Answer** button.

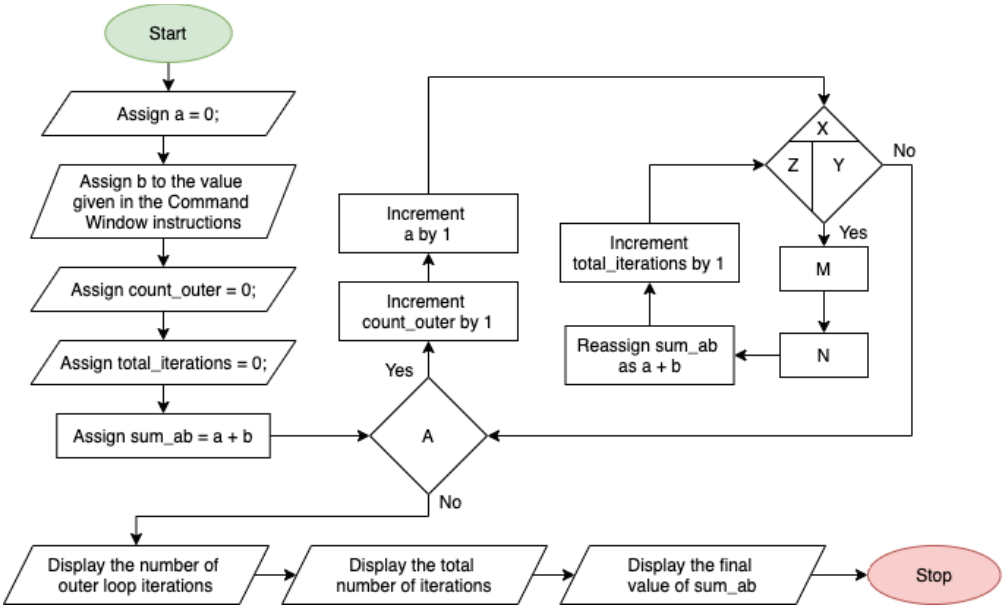
Problem 2: MATLAB Skills – For Loops

Introduction

This problem requires you to read a nested-loop flowchart and translate it into a script.

Problem

You have this partially completed flowchart:



To get the remaining instructions for this flowchart, you will need to use your p-code file, **A11_skills.p**. Match the text with the corresponding letters in the flowchart.

Instructions

1. Type this command into the MATLAB Command Window prompt:

   ```
   >> A11_skills(PUID, 2)
   ```

   Remember to replace PUID with your 8-digit Purdue University ID number (leave off the leading 00).

2. Read the instructions that display to the Command Window.
   - Use the included template to write the requested code.
   - Maintain the order of the items in the flowchart. Even if there is a different approach you could take to the solve the problem, you must follow the exact order and instructions in the flowchart.
   - Submit your script **and** displayed results to Gradescope.
   - Programming standards will not be assessed in this problem. Use the single-letter variables defined in the instruction text. Do not include comments in your solutions. Properly name your script.
3. Submit your work in Gradescope:
   a. Open Gradescope > **A11 Skills** and find the set of boxes that belong to **Q2** you want to submit. Enter the required information along with your answer:
      - ☐ **Function call**. Copy the command that you entered at the command prompt to call the function and paste the full command into this box. Be sure your PUID is included.

- ☐ **Instruction text**. Copy the instruction text that is displayed in the Command Window. Paste it into this box. Include all text provided.
- ☐ **Solutions**. Enter your solutions and script. Follow any additional instructions.

b. When you have entered all the required information for the question, click the **Save Answer** button.

Problem 3: Pixel Filtering

Introduction

You will apply your complex looping skills to an engineering context.

Problem

Image and signal arrays can require digital filtering. One filtering method is to identify an element in the array, find the median of it and its surrounding elements, and then replace the element with the median value. For example:

Replace the element in the second row, second column with the mean of it and the 8 values surrounding it:

1	8	1	3	4
7	9	8	6	2
5	10	6	0	1
2	8	9	4	6
0	2	8	6	6

$$y_{2,2} = \text{median}(1, 8, 1, 7, 9, 8, 5, 10, 6)$$

$$y_{2,2} = 7$$

1	8	1	3	4
7	7	8	6	2
5	10	6	0	1
2	8	9	4	6
0	2	8	6	6

Then repeat this for the next element in the array. For this example, we will move one element to the right (to the next horizontal element):

1	8	1	3	4
7	7	8	6	2
5	10	6	0	1
2	8	9	4	6
0	2	8	6	6

$$y_{2,3} = \text{median}(8, 1, 3, 7, 8, 6\ 10, 6, 0)$$

$$y_{2,3} = 6$$

1	8	1	3	4
7	7	6	6	2
5	10	6	0	1
2	8	9	4	6
0	2	8	6	6

Notice that the calculation used the median value of $y_{2,2}$ calculated in example 1 to find the new value for $y_{2,3}$.

Continue this process until all elements inside the border rows and columns have had the filter applied. This type of filter can start at any corner of the matrix but must start one row and one column from the border (at $y_{2,2}, y_{4,2}, y_{2,4}$, or $y_{4,4}$ in the example matrix). This filter cannot change any element that is on the border rows or columns of the array.

Using the same type of median filter described, create a function that meets the following criteria:

- Accepts no input arguments or output arguments
- Imports the pixel values from **Data_pixels.csv** using built-in function. Do not hardcode the data values.
- Using the imported data, applies three different median filters using three different starting points. All filters must start from the original data matrix. Do not overwrite the original data array.
    - Filter 1: start at index (2,2), move left to right from top to bottom, and end at index (last row - 1, last column - 1)
    - Filter 2: start at index (last row - 1, last column - 1), move right to left from bottom to top, and end at index (2,2)
    - Filter 3: start at index (2,2), move top to bottom from left to right, and end at index (last row - 1, last column - 1)
- Display the original data array and the three different filtered arrays on a 2x2 subplot grid with this layout:

Original	Filter 2
Filter 1	Filter 3

Instead of the `plot` command, display the images with the `heatmap` command. This command must be suppressed with a semicolon. For example: if you named your original pixel data as a variable `data`, your heatmap command to display the original data will be

```
heatmap(data);
```

Include a title for each heatmap, using only the filter name or "original data". Do not add any other plot formatting – use the defaults that appear with the `heatmap` command.

Hint: remember that `median` works differently on arrays than it does on vectors.

When your function is complete, run it to generate the final figure with the filtered images. Save that figure as a PNG with the name format A11Prob3_figure_*login*.png. Replace *login* with your career account login name.

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. These 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 **A11 – Team Planning**.
        2. In the area for Problem 3:
            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, you may add them in the *Optional* file submission area.
        3. Save your results.
    d. Add your teammates to the submission. 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, 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 script header if their help changed how you decided to solve the problem.
        1. Make sure you also completed the rest of the script header.
    c. Submit your properly named files to **A11 – Context Problems** in Gradescope.

Confirm Your Submission

Problems 1 and 2

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).

Confirm that your submission for **A11 – Skills Problems** includes

- ☐ The function call and instruction text for each skills question;
- ☐ 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.

Problem 3 – Individual deliverables

Confirm that your submission for **A11 – Context Problems** includes

- ☐ The expected deliverables and results;
  - This includes m-files, data files, and image files for each problem you complete.
- ☐ 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.

Problem 3 – Team plans

Confirm that your submission for **A11 – 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. Each non-skill problem's assignment grading has a table and a flowchart. The table outlines what LOs will be used to grade your work and what point values are assigned to each evidence. The flowchart outlines the grading process that a grader will use to assess your work.

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.

Individual Problems

Problem 1

LOs: PC05, MAT06

Problem 1 is worth 6 points. There is partial credit. The partial credit may be more specific than what is in the course LOs and is based on evidences in MAT06.

You must meet the PC05 expectations for each question. If you do not meet these, you will lose additional credit.

Evidence	Penalty
PC05 (1)	Lose full credit on problem
PC05 (2)	Lose 25% of full credit on problem
PC05 (3)	Lose 25% of full credit on problem
PC05 (4)	Lose 5% of full credit on problem

Problem 2

LOs: PC05, MAT06 and MAT07

Problem 2 is worth 6 points. There is partial credit. The partial credit may be more specific than what is in the course LOs and is based on evidences in MAT06 and MAT07.

You must meet the PC05 expectations for this problem. If you do not meet these, you will lose additional credit.

Evidence	Penalty
PC05 (1)	Lose full credit on problem
PC05 (2)	Lose 25% of full credit on problem
PC05 (3)	Lose 25% of full credit on problem
PC05 (4)	Lose 5% of full credit on problem

Problem 3

LO Table

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

	PC05	EPS02	MAT05	MAT08	MAT06	MAT06	MAT06
(1)	-8	0.4	0.5	0.5	0.2	0.2	0.2
(2)	-2	0.4	0.5	1	0.3	0.3	0.3
(3)	-2	0	0	0.5	0.5	0.5	0.5
(4)	-0.4	0	0	0	0.3	0.3	0.3
(5)	0	0	0	0	0.1	0.1	0.1
(6)	0	0	0	0	0	0	0
(7)	0	0	0	0	0	0	0
(8)	0	0	0	0	0	0	0

- Will grade the nested looping structure for each filter separately.

Grading Process