Problem Set 05
User-Defined Functions
Problems

Instructions:

1. This problem set contains a mix of paired and individual programming problems. Each problem has a set of deliverables that needs to be submitted. You are responsible for following the appropriate guidelines and instructions below. Create appropriately-named files as instructed.
2. Save all files to your Purdue career account in a folder specific to PS05.
3. Compress all deliverables into one zip file named **PS05_yourlogin.zip**. Submit the zip file to the Blackboard drop box for PS05 before the due date.

Problem Set Deliverables

Item	Type	Deliverable
Problem 1: Contact Lens Acceptability	Individual	PS05_contactlens_*yourlogin*.m PS05_contactlens_*yourlogin*_report.pdf Any data loaded into your code PS04_stats_io_*yourlogin1*_*yourlogin2*.m contactlens_decision.p
Problem 2: Window Blinds	Paired Individual	PS05_blind_fracs_*loginW*_*loginZ*.m PS05_blind_calcs_*loginX*_*loginY*.m PS05_blind_exec_*yourlogin*.m PS05_blind_exec_ *yourlogin*_report.pdf

Problem 1: Contact Lens Acceptability

Individual Programming

Problem Setup

Contact lenses are manufactured specifically for each patient, with lens geometry dictated by patient eye shape and vision correction needs. Two measurements are critical for properly fitting and functioning contact lenses (Fig. 1): the base curve radius (r) controls vision correction, while the lens diameter (d) ensures a proper fit on the patient's eye. Optometrists take detailed measurements of their patients and pass those measurements along to lens manufacturers. Typical base curve radii range from 7-10 mm, while typical diameters range from 13-16 mm.

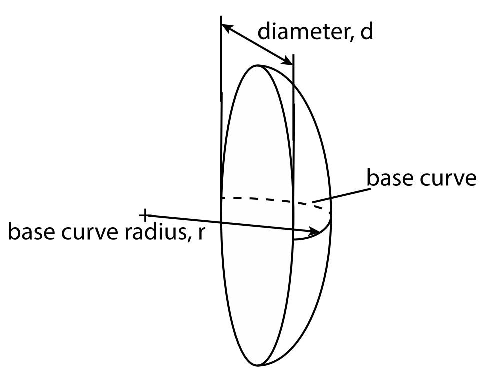


Fig 1. Contact Lens Geometry

Problem Set 05
User-Defined Functions
Problems

Contact lens manufacturers seek very precise control over lens geometry, and they routinely engage in various quality assurance (QA) practices so that they are confident that their lenses will fit properly and correct patients' vision problems. One key measure of manufacturing quality control is the ratio of standard deviation to mean value for both the base curve and the diameter for a batch of manufactured lenses. The ratios of these values must be below a threshold value, set by the manufacturer, for a batch of lenses to be deemed acceptable. We can define two 'lens ratios' (LR), for the base curve radius (BCR) and the diameter (D):

$$LR_{BCR} = \frac{\sigma_{BCR}}{\bar{r}_{BC}}; \quad LR_D = \frac{\sigma_D}{\bar{d}}$$

Here, the σ (sigma) indicates the standard deviation of a set of measurements, while the overbar notation (e.g. $\bar{x}$) indicates the mean of the measurements. QA standards for manufacturing indicate that these lens ratios must be below a very small threshold (ε, epsilon) for the lenses to be acceptable for sale to patients.

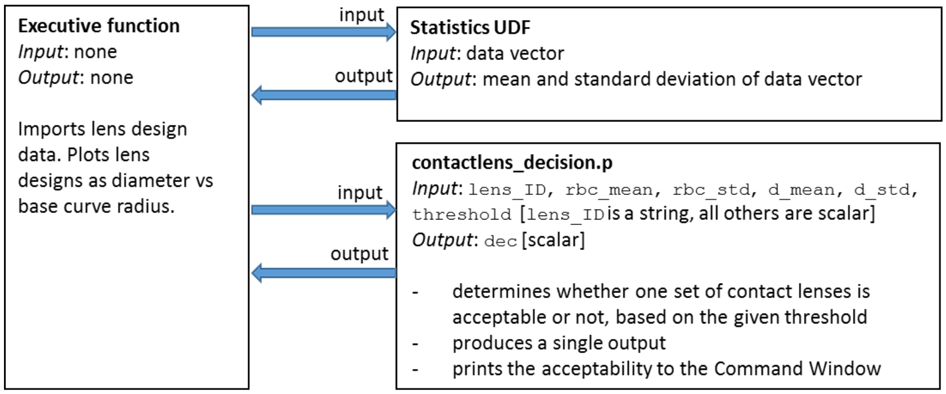
You have been supplied with a dataset, named **Data_contactlens.csv**, that contains geometry (base curve radius, diameter) for 50 contact lenses from each of four different design batches, arranged in a 50 x 8 matrix. **Open the dataset and review it**.

You have also been supplied with a MATLAB p-code, which is a protected file that can be called within MATLAB but whose code you cannot read. The provided p-code is named **contactlens_decision.p**. It is a user-defined function that can determine whether or not a batch of contact lenses meets the QA threshold standard. It requires six input arguments: lens design batch ID, $\overline{r_{BC}}$, σ_{BCR}, $\bar{d}$, σ_D, and ε. It has one output argument, *dec*, which is the binary decision variable about acceptability (0 = unacceptable, 1 = acceptable). P-code does not allow you to use MATLAB's help command. The help lines that would be visible in the non-protected version of `contactlens_decision` are:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function determines whether one set of contact lenses is acceptable
% or not, based on the given threshold. It produces a single output and
% also prints the acceptability to the Command Window
%
% Function Call
%   [dec] = contactlens_decision(lens_ID,bc_mean,bc_std,d_mean,d_std,threshold)
%
% Input Arguments
%     1. lens_ID:   The lens design batch ID (string)
%     2. bc_mean:   mean of a lens design's base curve radius (mm)
%     3. bc_std:    standard deviation of a lens design's base curve radius (mm)
%     4. d_mean:    mean of a lens design's diameter (mm)
%     5. d_std:     standard deviation of a lens design's diameters (mm)
%     6. threshold: acceptability ratio between a mean measurement and its
%                   standard deviation (dimensionless)
%
% Output Arguments
%     1. dec: numerical indicator (1 for acceptable; 0 for unacceptable)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem Set 05
User-Defined Functions
Problems

You will create an executive function (i.e. a no-input, no-output function) that calls a statistics UDF from PS04 Problem 2 to determine mean and standard deviation for the contact lens measurements and calls the p-code to determine the acceptability of the lens designs. The chart below shows how the functions must interact with each other.



Note: In PS04 Problem 2, you wrote an input-output function to calculate mean and standard deviation of a data vector. Use that function in this problem. The PS04 solutions are available on Blackboard.

Problem Steps

1. Download the **PS05_contactlens_template.m**. Complete the header and save the file as **PS05_contactlens_*yourlogin1_yourlogin2*.m**.
2. Create an appropriate function definition line.
3. In the `INTIALIZATION` section, load all values that need to be hardcoded in the function.

   Hint: You can save character strings to variable names.

   Try this in MATLAB:

   ```
   >> my_uni = 'Purdue University';
   >> fprintf('I attend %s. \n',my_uni)
   ```
4. In the `LENS DESIGN PLOT` section, create a single plot with multiple datasets on one figure window to show the diameter versus base curve radius for each lens. Plot each lens design with a different color marker.
5. Run the function at this point and examine the plot. In the `ANALYSIS` section of your code, answer the following question.

   Q1: What do you expect the data points to look like for an acceptable lens design? What do you expect the data points to look like for an unacceptable lens design?
6. In the `FUNCTION CALLS` section, perform the following for each lens design:

   a. Call your input-output stats UDF from PS04 to calculate the required statistics for each lens parameter. Revise this function so that it does not display anything to the Command Window.

   b. Call `contactlens_decision.p` using

Problem Set 05
User-Defined Functions
Problems

i. the appropriate lens design batch ID value, as a string variable

ii. the calculated mean and standard deviation of both parameters for each lens in the dataset

iii. a threshold value of ε = 0.02.

7. Save your files and run your executive function. Debug your code as necessary.
8. Publish PS05_contactlens_*yourlogin*.m to a PDF file. Name the published file **PS05_contactlens_*yourlogin*_report.pdf**.

Problem 2: Window Blinds

Paired Programming: Sub UDFs

Individual: Executive function

Problem Setup

Energy conversion is an area of mechanical engineering and a concern of a number of other engineering disciplines. This area is often concerned with the heating, ventilation, and air conditioning (HVAC). To determine the heating and cooling loads of buildings, engineers need to determine the heat flows in and out of buildings. Solar radiation comes through windows. You and your team will create a set of interacting user-defined functions to determine the transmission, absorption, and reflection of solar energy through horizontal venetian blinds.

Horizontal venetian blinds transmit, absorb, and reflect solar radiation, and these radiation values can be calculated. The calculations require knowledge of the operational geometry of a venetian blind, as shown in *Figure 1*.

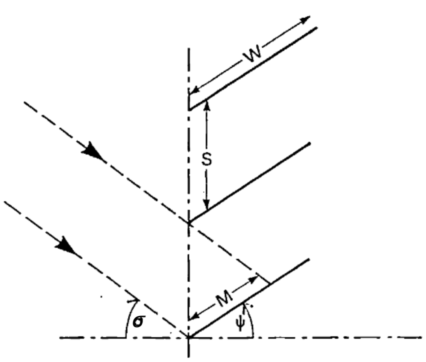


Figure 1 Operation of a blind

Table 1 Geometric parameters

Parameter	Unit	Description
W	mm	slat width
S	mm	slat spacing
M	mm	width of slat that is illuminated
σ (sigma)	degrees	vertical shadow angle
ψ (psi)	degrees	slat angle

Additionally, the blind surface material has an absorptivity constant, α (alpha). This is a dimensionless value between 0-1.

Problem Set 05
User-Defined Functions
Problems

When the slats of the blind are **fully illuminated** (i.e., when $M = W$) by incoming light, there are three fractions that are used in the transmission and absorption equations:

- **F1:** fraction of total radiation reflected by the lower slat that passed into the room when the whole width of a slat is illuminated;
- **F2:** fraction of total radiation reflected by the lower slat that is intercepted by the upper slat when the whole width is illuminated;
- **F3:** fraction of radiation reflected by the upper slat that passes into the room;

$$F_1 = \frac{1}{2}\left(1 + \frac{S}{W} - \sqrt{1 + \left(\frac{S}{W}\right)^2 + 2\frac{S}{W}\sin\psi}\right)$$

$$F_2 = \frac{1}{2}\left(\sqrt{1 + \left(\frac{S}{W}\right)^2 + 2\frac{S}{W}\sin\psi} + \sqrt{1 + \left(\frac{S}{W}\right)^2 - 2\frac{S}{W}\sin\psi} - 2\frac{S}{W}\right)$$

$$F_3 = \frac{1}{2}\left(1 + \frac{S}{W} - \sqrt{1 + \left(\frac{S}{W}\right)^2 - 2\frac{S}{W}\sin\psi}\right)$$

When the blind slats are fully illuminated, the total fraction of incident radiation transmitted by the blind is

$$T_D = 1 - \left(\frac{W}{S}\right)\left(\frac{\sin(\sigma+\psi)}{\cos\sigma}\right)\left(1 - F_1(1-\alpha) - \frac{F_2(1-\alpha)^2\left[F_3 + F_1 \cdot F_2(1-\alpha)\right]}{1 - F_2^{\,2}(1-\alpha)^2}\right)$$

The total fraction of radiation absorbed is

$$A_D = \frac{aW\sin(\sigma+\psi)}{S\left[1 - F_2(1-\alpha)\right]\cos\sigma}$$

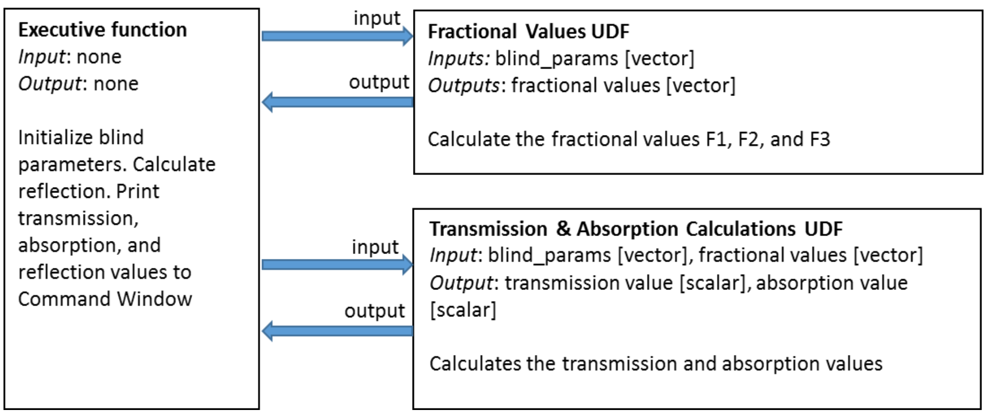
The total reflected fraction of light is

$$R_D = 1 - A_D - T_D$$

<u>Important Note:</u> In all of the above equations, angles must be in radians.

Problem Set 05
User-Defined Functions
Problems

Your team is divided into two pairs. Pair WZ will write a user-defined function that calculates the fraction values F_1, F_2, and F_3. Pair XY will write a user-defined function that calculates the transmission (T_D) and absorption (A_D) values. Pair WZ will share their function with Pair XY and vice versa. After you have both UDFs, you will individually write an executive function to call the UDFs and calculate the total reflected light, R_D. The chart below shows how the functions must interact with each other.



Hint: Coding large equations in one line is difficult. Calculate smaller terms and assign them to MATLAB variables; then build the equation from the assigned variables and any remaining terms.

Example: Code $x = \frac{-b+\sqrt{b^2-4ac}}{2a}$ using assigned scalar variables a, b, and c

```
% CALCULATIONS
discrim = sqrt(b^2 - 4*a*c);
denom = 2*a;
x = (-b + discrim)/denom;
```

A test blind is provided and should be used as a benchmark to test and compare to your UDF outputs.

Table 2 Test blind information

Test Blind Parameters			Test Blind Results (dimensionless)	
S	Slat Angle	55 mm	F_1	0.0728
W	Slat Width	60 mm	F_2	0.3376
α	Absorptivity Constant	0.37	F_3	0.5896
σ	Shadow Angle	25 degrees	T_D	0.0168
ψ	Slat Angle	45 degrees	A_D	0.5316

Problem Set 05
User-Defined Functions
Problems

Problem Steps

A. **Pair WZ** – create a UDF to calculate the fraction values

1. Using **PS05_blinds_subUDF_template.m**, create a user-defined function named **PS05_blind_fracs_your*loginW*_*yourloginZ*.m** that:
   a. Accepts one input argument: a set of blind parameters (S, W, α, σ, ψ) as one vector, with the angles in radians
   b. Returns one output argument: the three fractional values (F_1, F_2, and F_3) as one vector
2. Use the test blind parameters to thoroughly test and debug your function.
3. When your function is complete, share it with Pair XY

B. **Pair XY** – create a UDF to calculate the transmission and absorption fractions

1. Using **PS05_blinds_subUDF_template.m**, create a user-defined function named **PS05_blind_calcs_*yourloginX*_*yourloginY*.m** that:
   a. Accepts two input arguments: a set of blind parameters (S, W, α, σ, ψ) as one vector with angles in radians, and the fractional values (F_1, F_2, and F_3) as one vector
   b. Returns two output arguments: the total fraction of incident radiation transmitted as a scalar value, and the total fraction of radiation absorbed by the blind as a scalar value
2. Use the test blind parameters to thoroughly test and debug your function.
3. When your function is complete, share it with Pair WZ.

C. **Individually** – create an executive function to calculate the reflection fraction

Now that you have a UDF to calculate the fractional values and a UDF to calculate the transmission and absorption fraction, you can write an executive function (i.e. a no-input, no-output function) to allow you to calculate the fractions and the reflection value for any blind parameters. You have been provided with a new blind to test.

Table 3 Blind parameters

Parameter		Blind 1
S	Slat Spacing	30 mm
W	Slat Width	35 mm
α	Absorptivity Constant	0.52
σ	Shadow Angle	10 deg

Table 4 Blind settings

Parameter		Setting 1	Setting 2
ψ	Slat Angle	20 deg	30 deg

Problem Set 05
User-Defined Functions
Problems

NOTE: As the individual programmer, you are responsible for ensuring that the two sub-UDFs you will call in your executive function work properly to meet the needs of the problem and that those sub-UDFs meet the course programming standards.

1. Using *PS05_blinds_exec_template.m*, create an executive function and name it **PS05_blind_exec_*yourlogin*.m**. Using the proper code sections within the template, the executive function must

    a. Initialize the blind parameters (Table 3) and the Setting 1 slat angle (Table 4), which are then used to create a row vector named `blind_params`.

    Hint: Use MATLAB to learn about the `deg2rad` function.

    b. Call the fraction UDF and the transmission and absorption UDF,

    c. Compute the reflection, and

    d. Print the resulting transmission, absorption, and reflection values to the Command Window. The print command should display something similar to

    ```
    When the slat angle is <20/30> deg, the transmission is <value>, the absorption is <value>, and the reflection is <value>.
    ```

2. Once your code is debugged and working, run your function for the Setting 2 slat angle blind setting provided in Table 4.

3. In the `COMMAND WINDOW OUTPUT` section of your **executive function**:

    a. Paste as comments the function call and the displayed transmission, absorption, and reflection results for the Setting 1 and the Setting 2 slat angles.

4. Publish your executive function as a PDF file named **PS05_blind_exec_*yourlogin*_report.pdf** using Setting 2.

Reference: http://bse.sagepub.com/content/1/2/83.full.pdf