Dimensional Analysis Project

Requirements

MATLAB 2024a or newer

Backstory

You are investigating a novel process you want to implement as part of a microfluidic lab-on-a-chip. You have information regarding a variety of process parameters under many different conditions. You know several parameters that are critical to the process behavior and one of note that you want to be able to assess as a function of the others.

Your first task is to ***evaluate all potential pi groups*** and ***assess their functional relationship*** to the critical dependent parameter. You will use the statistical results to ***assess which pi groups are critical*** to the process behavior, since they will be important to manage for the final product.

Your second task is to determine the necessary parameter values for a cheap ***scale model prototype*** that can be used to easily and precisely interrogate the process operating constraints.

Deliverables (check-in during PSS 3/6 and 3/7; due 3/14/2025 at 11:59 p.m.)

1. **Part 1: All pi groups** determined by parameter set and assigned basis parameters.
2. **Part 2: Three distinct fit models for the dependent pi data** given the independent pi data, determined from the provided data and the pi group structures determined previously — the specific models are discussed in detail in the following project documentation.
3. **Part 3: Best fit model and the critical pi groups** based on the statistical results of the model fit data determined previously.
4. **Part 4: Scale model analysis** of critical pi groups using the <u>first</u> observation data set — scale the geometry as indicated in the following project documentation and provide a set of other parameter values that will result in analogous behavior.

Overview of Steps

Each student will be given a unique numerical key) to input into a MATLAB function as follows:

```
Project_data = ProjectDataGenerator(key);
```

The function will output a cell array with the following format:

Measured Parameter Units	A table of parameters with associated units.
Measurements	A table of parameters with associated values. Each column is a distinct parameter. Each row is a distinct observation (set of measurements).
Basis Parameters	A list of parameter names to use as your basis set for devising your pi groups.
Dependent Parameter	The parameter associated with the pi group you want to define as a function of the other pi groups.

The information contained in the cell array will be necessary for your analyses:

- You will use the *Measured Parameter Units* to build the matrix needed to calculate your pi group structures.
- You will use the *Measurements* to calculate the values of the pi groups for each observation and fit those data to the models outlined later in this document.
- You will use the provided *Basis Parameters* to determine which set of pi group structures you generate.
- You will use the *Dependent Parameter* to determine which pi group will be the dependent variable for your data fits.

Part 1: Finding Pi Groups

Deliverables:

- **Work/code for generating your pi groups**
- **Formula for each pi group**

The first stage of this project is determining all the (possible) pi groups. To do this, you must first construct a matrix of all (10) parameter dimensions. **The first columns of the matrix should be the (3 or 4) basis parameters indicated in your cell array**. This will ensure the intended pi group outcomes.

The general matrix structure should be as follows (*matrix values are indicated in red*):

	Parameter 1	**…**	**Parameter 10**
Mass Exponent	*Parameter 1 mass exponent*	…	*Parameter 10 mass exponent*
Length Exponent	*Parameter 1 length exponent*	…	*Parameter 10 length exponent*
Time Exponent	*Parameter 1 time exponent*	…	*Parameter 10 time exponent*
Temperature Exponent	*Parameter 1 temperature exponent*	…	*Parameter 10 temperature exponent*

If you have three basis parameters, then one dimension is not covered by the bases and must be eliminated. Likewise, any parameters dependent on that extra dimensions are irrelevant — and should be removed from your matrix.

After this setup phase, you can then determine the reduced row echelon form of the matrix — in MATLAB, you can use the **rref()** function (*please review the MATLAB documentation for this function and remember the order of your unit columns matters*). Mechanically, this is a Gauss-Jordan elimination. The resulting values in the non-basis columns will be the exponents of the basis parameters whose product yields the non-basis parameter units.

To help you interpret the reduced row echelon form and generate appropriate pi group formulas, consider the following:

Given:

$$\text{Basis Parameters:} \quad a \quad b \quad c$$

$$\text{Non-Basis Parameter:} \quad p$$

$$\text{Non-Basis Parameter Column:} \quad \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

$$[x] \stackrel{\text{def}}{=} \text{units of } x$$

Then:

$$[p] = [a]^u[b]^v[c]^w$$

Which yields these pi group structures:

$$\Pi_i = \frac{a^{u_i}b^{v_i}c^{w_i}}{p_i}$$

Part 2: Fitting the data

Deliverables:

- **Work/code for fitting your pi data**
- **Fit coefficients for each model (raw and scaled, where applicable)**
- **Fit statistics for each model — p-values for coefficients and overall fit**

A list of `Steps (1-8)` is provided to help you organize your operations. First, you will calculate pi group data:

Step 1:	Use pi group formulas from Part 1 to calculate all pi group data from the provided measurements *(you should have 200)*
Step 2:	Identify (and isolate) your independent and dependent pi data — i.e., the data independent from and dependent upon the target dependent variable. **The pi group containing your "dependent" parameter is to be treated as your dependent variable.**

Recommendation: Given the widely varying scales of different parameters, rather than fitting the raw data, ***I would suggest first scaling the data. Simply divide each set of pi group data by the maximum value in that set.*** This puts all values on the same general scale and makes fitting a much simpler computational process. After fitting the data, you can then readjust your predicted coefficients according to each scale factor.

- *For a* ***multilinear fit***, all coefficients are multiplied by the dependent scale factor, and each parameter coefficient is divided by the relevant independent scale factor.
- *For an* ***exponential fit***, the first coefficient is multiplied by the dependent scale factor, and each parameter coefficient is divided by the relevant independent scale factor.
- *For a* ***power fit***, the first coefficient is multiplied by the dependent scale factor, then it is further multiplied by the product of the inverse of each independent scale factor raised to the power of its corresponding coefficient.

It is relatively easy to determine these adjustments by plugging in the modifications to each parameter and grouping them with the most convenient coefficient.

Step 3: Normalize your pi data by dividing each set by their respective maximum value *(you will need to keep track of the normalization factors for Step 7)*

Next, you will fit the data to three different possible equation structures:

- Multilinear: $c_0 + \sum_{i=1}^{n}(c_i \Pi_i) = c_0 + c_1\Pi_1 + c_2\Pi_2 + \cdots + c_n\Pi_n$
  - *You will perform this in two different ways (see* **`Step 4`** *and* **`Step 7`***)*
- Exponential: $c_0 \exp(\sum_{i=1}^{n}(c_i \Pi_i)) = c_0 \exp(c_1\Pi_1 + c_2\Pi_2 + \cdots + c_n\Pi_n)$
- Power Law: $c_0 \prod_{i=1}^{c}(\Pi_i^{c_i}) = c_0 \Pi_1^{c_1} \Pi_2^{c_2} \cdots \Pi_n^{c_n}$

You can use whatever software you like, but either MATLAB or Python is recommended, and specific details for using MATLAB functions are provided below.

`Step 4:` `Perform a multilinear fit on the data` *`(be sure to save the fit results)`*

fitlm(independent pi group data, dependent pi group data, “linear”**)**

You may further include names for the predictor variables (independent pi groups) and response variable (dependent pi group). I refer you to the MATLAB documentation for this and for greater detail on the function usage and behavior.

`Step 5:` `Perform an exponential fit on the data` *`(be sure to save the fit results)`*

fitnlm(independent pi group data, dependent pi group data, exponential model function, initial guess at coefficients**)**

You may further include names for the predictor variables (independent pi groups) and response variable (dependent pi group). I refer you to the MATLAB documentation for this and for greater detail on the function usage and behavior.

The exponential model function should be defined as:

```
@(b,x)b(1)*exp(x*b(2:end))
```

I recommend the initial guess be based on a linear fit to the log of the independent data:

fitlm(log(independent pi group data**)**, dependent pi group data, “linear”**)**

Just be sure to update the intercept (first coefficient, `b(1)`) to be the exponential of the estimate — `exp(b(1))`.

`Step 6:` `Perform a power fit on the data` *`(be sure to save the fit results)`*

fitnlm(independent pi group data, dependent pi group data, power model function, initial guess at coefficients**)**

You may further include names for the predictor variables (independent pi groups) and response variable (dependent pi group). I refer you to the [MATLAB documentation] for this and for greater detail on the function usage and behavior.

The power model function should be defined as:

```
@(b,x)b(1)*prod(x.^(b(2:end)'),2)
```

I recommend the initial guess be based on a linear fit of the log of the dependent data to the log of the independent data:

fitlm(log(independent pi group data**)**, **log(**dependent pi group data**)**, “linear”**)**

Just be sure to update the intercept (first coefficient, `b(1)`) to be the exponential of the estimate — `exp(b(1))`.

Step 7: Perform a multilinear fit on the data **using a nonlinear fitting method** *(be sure to save the fit results)*

fitnlm(independent pi group data, dependent pi group data, linear model function, initial guess at coefficients**)**

You may further include names for the predictor variables (independent pi groups) and response variable (dependent pi group). I refer you to the [MATLAB documentation] for this and for greater detail on the function usage and behavior.

Using what you learned creating model functions for the exponential and power models, define an appropriate model function for a linear fit. You must also define an appropriate initial guess (or use a random coefficient vector, if you cannot come up with one).

You should confirm that the results match that of the basic multilinear fit function.

If you did not use MATLAB, and your approach to **Step 4** *already required you to define your model function, you may skip this step.*

Step 8: *(Follow-Up From Step 3)* Rescale fit coefficients for each fit to obtain the relevant coefficients for the raw pi group data

Part 3: Choosing the best fit and deciding on the relevant pi groups

Deliverables:

- **Work/code for choosing model fits and critical pi groups**
- **Best model choice with justification**
- **Critical pi groups with justification**

Once you have fit the data to all three models, you need to decide which model is the best. You may use some form of model p-value (e.g., the chi-squared model evaluation provided by MATLAB), but we recommend looking at AIC values, as these are relatively commonly used for comparing distinct models — lower AIC is better. MATLAB provides these as part of its fit model:

```
fit.ModelCriterion.AIC
```

Once you have decided on your preferred fit, you can evaluate the importance of each independent parameter. Look at the p-values associated with each coefficient. Use a Bonferroni correction based on the number of independent parameters given an initial alpha of 0.01. For example, three independent parameters would mean an alpha cutoff of 0.01/3 or 0.0033. Let only the pi groups with p-values less than the assigned cutoff be considered critical.

If you scaled your data prior to fitting, you should notice that (in general) lower p-values correspond to larger (raw) coefficient magnitudes — given the same value range, the parameters explaining the data best are given greater weight (larger coefficients).

Part 4: Devising a scale model

Deliverables:

- **Work/code for creating your scale model**
- **Parameter values chosen to match pi groups at both scales and their justifications with real-world examples**

Let the first observation in your data set be the setup of interest. Let's assume that some aspect of this is inconvenient for laboratory study. Instead, you want to run your studies in a scale model using more convenient material and kinematic properties. For a scale model to be effective, all the independent pi groups must have the same values at both scales.

Assume you are creating a model at 10x scale (all lengths are multiplied by 10, areas by 100, and volumes by 1000). Provide reasonable values for other parameters that would match the independent pi groups at both scales. Where possible, justify your value choices with examples of similar real-world material/system properties. If such justification does not seem possible, explain your reasoning.