

Contents

- [BMED 3310 Computational Project - Trial Run \(jjc\)](#)
- [Part 1: Finding Pi Groups](#)
- [Calculate Pi Groups from Observation Data](#)
- [Alternative methods for exponential and power fitting](#)
- [Comparing Fits](#)
- [Readjust Coefficients](#)

BMED 3310 Computational Project - Trial Run (jjc)

```
clear all
close all
clc

% Load in Project Data
projectdata = GenerateProjectData(37241);
```

Part 1: Finding Pi Groups

Create Variables of Parameter Dimensions

```
area = [0;2;0;0];
dynamicviscosity = [1;-1;-1;0];
mass = [1;0;0;0];
massrxnrate = [1;0;-1;0];
massresistance = [0;-3;1;0];
specificheat = [0;2;-2;-1];
time = [0;0;1;0];
velocity = [0;1;-1;0];
volume = [0;3;0;0];
volumetricthermalexpansion = [0;0;0;-1];

% Create Matrix of Parameter Dimensions
parameterdimensions = [area,massrxnrate,massresistance,specificheat,dynamicviscosity,mass,time,velocity,volume,volumetricthermalexpansion];

% Determine the Reduced Row Echelon Form of the Matrix
rrmatrix = rref(parameterdimensions)
```

```
rrmatrix =

Columns 1 through 7

    1.0000         0         0         0   -0.5000    1.5000    1.5000
         0    1.0000         0         0    1.0000    1.0000         0
         0         0    1.0000         0         0    1.0000    1.0000
         0         0         0    1.0000         0         0         0

Columns 8 through 10

   -1.0000    1.5000    2.0000
         0         0         0
   -1.0000         0    2.0000
         0         0    1.0000
```

Calculate Pi Groups from Observation Data

```
base_parameter_indices = [1,4,5,6];
nonbase_parameter_indices = [2,3,7,8,9,10];
dataset = table2array(projectdata{2,2}); % Get raw dataset from project data
exponents = rrmatrix(:,5:end);

% Separate dataset into base and non-base matrices
base_data = dataset(:,base_parameter_indices);
nonbase_data = dataset(:,nonbase_parameter_indices);

% Create pi group data set
for i = 1:1:6
    pi_data(:,i) = base_data(:,1).^exponents(1,i).*base_data(:,2).^exponents(2,i).*base_data(:,3).^exponents(3,i).*base_data(:,4).^exponents(4,i)./nonbase_data(:,i)
end

% Scaling the data
scaled_pi_data = pi_data./max(pi_data);

% Separate data into dependent and independent pi data
dependent_pi_data = scaled_pi_data(:,1);
independent_pi_data = scaled_pi_data(:,2:end);
```

```
% Multilinear fit
MultilinearMdl = fitlm(independent_pi_data,dependent_pi_data);

% Get initial guess for coefficients for exponential and power models
mdl = fitlm(log(independent_pi_data), dependent_pi_data, 'linear'); % Linear fit to log of independent data

coefficient_guess = table2array(mdl.Coefficients(:,1));
coefficient_guess(1) = exp(coefficient_guess(1));

% Exponential fit
ExponentialMdl = fitnlm(independent_pi_data,dependent_pi_data,@(b,x)b(1)*exp(x*b(2:end)), coefficient_guess);

% Power fit
PowerMdl = fitnlm(independent_pi_data,dependent_pi_data,@(b,x)b(1)*prod(x.^(b(2:end)'),2),coefficient_guess);
```

Alternative methods for exponential and power fitting

Exponential Option 2

```
modelfun = @(b,x)b(1).*exp(x(:,1).*b(2)).*exp(x(:,2).*b(3)).*exp(x(:,3).*b(4)).*exp(x(:,4).*b(5)).*exp(x(:,5).*b(6)));
ExponentialMdl_2 = fitnlm(independent_pi_data,dependent_pi_data,modelfun, coefficient_guess);

% Exponential Option 3
modelfun = @(b,x)b(1).*exp(x(:,1).*b(2)+x(:,2).*b(3)+x(:,3).*b(4)+x(:,4).*b(5)+x(:,5).*b(6));
ExponentialMdl_3 = fitnlm(independent_pi_data,dependent_pi_data,modelfun, coefficient_guess);

% Writing my own power fit instead of using the one provided
modelfun = @(b,x)b(1).*(x(:,1).^b(2)).*(x(:,2).^b(3)).*(x(:,3).^b(4)).*(x(:,4).^b(5)).*(x(:,5).^b(6));
PowerMdl_2 = fitnlm(independent_pi_data,dependent_pi_data,modelfun, coefficient_guess);
```

Comparing Fits

Output Model AIC

```
display(sprintf('The multilinear model's AIC is: %f\n', MultilinearMdl.ModelCriterion.AIC));
display(sprintf('The exponential model's AIC is: %f\n', ExponentialMdl.ModelCriterion.AIC));
display(sprintf('The power model's AIC is: %f\n', PowerMdl.ModelCriterion.AIC));
```

The multilinear model's AIC is: -1979.707075

The exponential model's AIC is: -613.214238

The power model's AIC is: -718.915889

Readjust Coefficients

```
scale_factors = max(pi_data);
coefficients = MultilinearMdl.Coefficients(:,1);
rescaled_coefficients = coefficients.*scale_factors(1);
rescaled_coefficients(2:end,1) = rescaled_coefficients(2:end,1)./scale_factors(2:end)'

for i = 1:1:6

    p_value = table2array(MultilinearMdl.Coefficients(i,"pValue"));

    if p_value<0.01/5
        fprintf('Keep coefficient x%d: p-value = %e (Significant)\n', i-1,p_value);
    end
end
```

rescaled_coefficients =

6×1 table

	Estimate
(Intercept)	2.6435e-05
x1	-0.00064478
x2	1.5998e-07
x3	5.0866e-06
x4	5.8524e-07
x5	2.0294e-11

Keep coefficient x0: p-value = 1.522938e-80 (Significant)
Keep coefficient x3: p-value = 0.000000e+00 (Significant)
Keep coefficient x5: p-value = 1.467218e-239 (Significant)

.....