Obtaining and Characterizing a Power Spectral Density Plot

Northwestern University

Nicolas Guerra
Student of Professor Mel Ulmer
Summer of 2020

Contents

1 Introduction **1**

2 One Dimensional PSD **1**

3 Two Dimensional PSD **1**

4 Deriving an Analogous 2D PSD from a 1D PSD **2**

5 Deriving an Analogous 1D PSD from a 2D PSD **3**

6 Comparing the Smoothness of a Surface to the ATST **3**

7 Understanding the Olympus Confocal Microscope **7**

8 Conclusion **8**

9 MATLAB Code for Obtaining a 1D PSD **9**

10 MATLAB Code for Obtaining a 2D PSD **10**

1 Introduction

The purpose of this report is to document how to obtain a one dimensional and two dimensional power spectral density plot of surface roughness data. Moreover, this report will also show how to find an analogous 2D PSD from a 1D PSD, or vice versa, and compare it to the specifications of the Advanced Technology Solar Telescope, ATST, to determine if the surface measured is "smooth." The procedure for both PSDs consists of fast Fourier transforms and non-linear least-squares power law fits. Lastly, Our understanding of the Olympus confocal microscope's way of creating a 1D PSD will be covered.

2 One Dimensional PSD

In order to obtain a 1D PSD, take note of the sampling frequency, f_s in units of μm^-1, used to measure the height data in units of μm. After confirming the height data is in units of μm, take the fast Fourier transform, FFT, of the data set. Since we are only interested in the amplitude of the FFT, take the magnitude of the FFT. Because the second half of the FFT is a mirror image, only use the first half of the data which will be called X_{mag}. The frequency axis of X_{mag} should range from 0 μm^-1 to f_{max}, where f_{max} is equal to half the size of original FFT divided by the product of the size of the original data, N, and f_s.

$$f_{max} = \frac{length(FFT)}{2 * N * f_s} \tag{1}$$

The 1D PSD will then be equal to two times X_{mag} squared divided by the product of N and f_s.

$$1D\ PSD = \frac{2 * (X_{mag})^2}{N * f_s} \tag{2}$$

The MATLAB code for this procedure can be found on Page 9.

3 Two Dimensional PSD

This section is more about how to use Kashmira Nakhoda's code [1] and how to obtain a 2D PSD rather than giving an explanation on how to find a 2D PSD by hand. In order to obtain a 2D PSD, take note of the pixel size going in the x and y directions, px and py, and the wavelength, lambda, used to measure the height data. The units of px, py, and lambda should be in μm. Using MATLAB, call Kashmira Nakhoda's function psd_2D_calculation.m [1]. The PSD returned will have units of μm^4 and frequency will be in units of μm^-1. Note, however,

[1] Nakhoda, Kashmira "Prediction of the BRDF with Microfinish Topographer Roughness Measurements" University of Arizona, 2013

that the frequency axis presented will be the square root of the sum of the two frequencies in each direction squared.

$$f = \sqrt{f_x^2 + f_y^2} \tag{3}$$

4 Deriving an Analogous 2D PSD from a 1D PSD

If one wanted to compare a 2D PSD with a 1D PSD, one can convert their 1D PSD to a 2D PSD. To start, a 1D PSD plot of surface roughness in units of μm^3 can be characterized by the following power law where k_{1D} and n_{1D} are constants, and f_x is the frequency in the direction of the one dimensional data.

$$1D\ PSD = \frac{k_{1D}}{(f_x)^{n_{1D}}} \tag{4}$$

Using a non-linear least-squares solver, one can fit the above power law to the 1D PSD data and obtain their characterizing constants k_{1D} and n_{1D}. One can use the MATLAB code on Page 10 to find these constants. Note that in order to decrease influence from outliers in the beginning of the PSD, the MATLAB program takes in the index number of where to start the fit.These constants can then be converted into their analogous two dimensional version through the following equations.

$$k_{2D} = \frac{\Gamma[\frac{n_{1D}+1}{2}]}{2 * \Gamma[\frac{1}{2}] * \Gamma[\frac{n_{1D}}{2}]} * k_{1D} \tag{5}$$

$$n_{2D} = n_{1D} + 1 \tag{6}$$

To get the frequency axis for the analogous 2D PSD, use Equation 3. Since the 1D PSD only has one frequency axis and assuming your analogous 2D PSD is isotropic, both f_x and f_y will be equal to the 1D PSD frequency axis which should be in units of μm^-1. To state plainly:

$$f_{2D} = \sqrt{2} f_{1D} \tag{7}$$

Now, one has all the parts to calculate their corresponding 2D PSD. To do this, they will use the following formula where the input frequency is the above f_{2D} in units of μm^-1 and the output 2D PSD will be in units of μm^4

$$2D\ PSD = \frac{k_{2D}}{(f_{2D})^{n_{2D}}} \tag{8}$$

5 Deriving an Analogous 1D PSD from a 2D PSD

If one wanted to instead convert their 2D PSD into a corresponding 1D PSD, that can be done as well. Using a non-linear least-squares solver, one can fit the power law from Equation 8 to the 2D PSD data and collect the characterizing constants k_{2D} and n_{2D}. One can use the MATLAB code on Page 9 to find these constants. Note that in order to decrease influence from outliers in the beginning of the PSD, the MATLAB program takes in the index number of where to start the fit. To convert these into k_{1D} and n_{1D}, one can solve for them using Equations 5 and 6, or for a more direct approach, use the following equations.

$$k_{1D} = \frac{2 * \Gamma[\frac{1}{2}] * \Gamma[\frac{n_{1D}}{2}]}{\Gamma[\frac{n_{1D}+1}{2}]} * k_{2D} \tag{9}$$

$$n_{1D} = n_{2D} - 1 \tag{10}$$

To get the frequency axis, f_{1D}, from the 2D PSD, one must assume their 2D surface roughness data is isotropic. Solving for f_{1D} from Equation 7, one gets the following formula where f_{1D} is in units of $\mu m^{-}1$.

$$f_{1D} = \frac{f_{2D}}{\sqrt{2}} \tag{11}$$

Now, one has all the components to calculate their analogous 1D PSD. To do so, one must use Equation 4 which outputs their 1D PSD in units of μm^3.

6 Comparing the Smoothness of a Surface to the ATST

Since after reading the above sections one can characterize a surface with general constants k and n, one can understand the smoothness of the surface by comparing it to the specifications of the Advanced Technology Solar Telescope. For the purposes of explaining, a silicon surface will be compared to the ATST. Using the MATLAB code on Page 10, the frequency and 2D PSD of a silicon surface is plotted below.

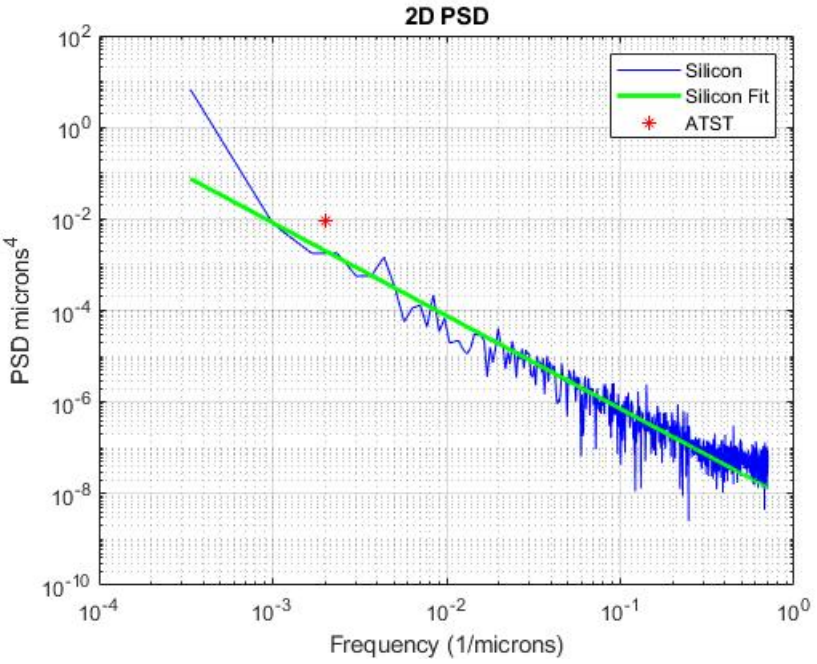

Figure 1: 2D PSD of a silicon surface and its power law fit compared to the specifications of the ATST

The specifications of the ATST is .009 μm^4 at a frequency of .002 μm^-1, and the fit constants k_{2D} and n_{2D} are $6.68 * 10^{-9} \mu m^4$ and 2.03, respectively. Seeing how close the 2D PSD of silicon is to the ATST, one can use the k_{2D} and n_{2D} acquired from the silicon data to justify whether a surface is "smooth." Now, let's say one had only a slice of the silicon data, so they only have a 1D PSD at hand. One can convert that 1D PSD into a 2D PSD and still compare against the ATST. To show what is being said, here is a 1D PSD of a slice of the 2D silicon data and its fit.

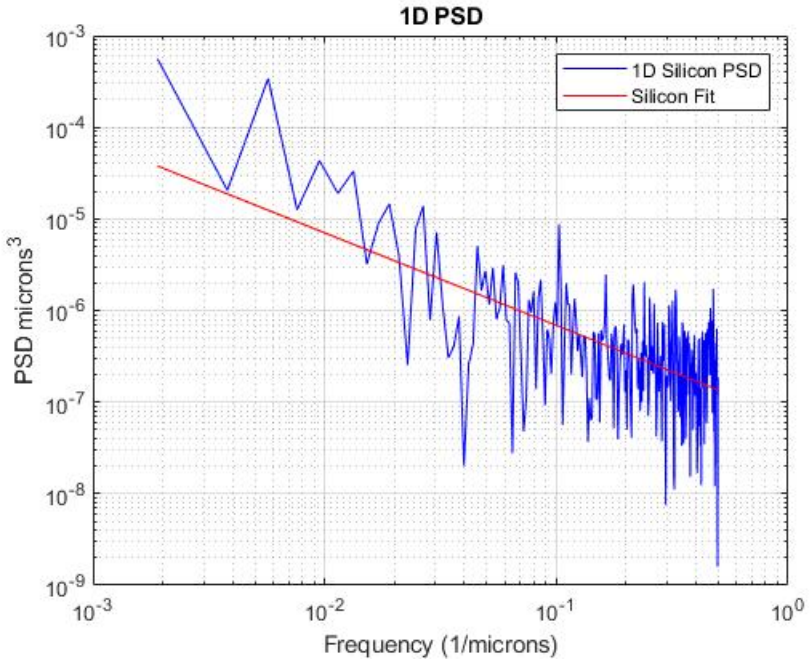

Figure 2: 1D PSD of the silicon slice and its power law fit

The fit constants for k_{1D} and n_{1D} are $6.69 * 10^{-8} \mu m^3$ and 1.01, respectively. After running these constants through Equations 5 and 6, the analogous characterizing constants are $1.07 * 10^{-8} \mu m^4$ for k_{2D} and 2.01 for n_{2D}. Let's plot these constants in the initial figure and see how they compare.

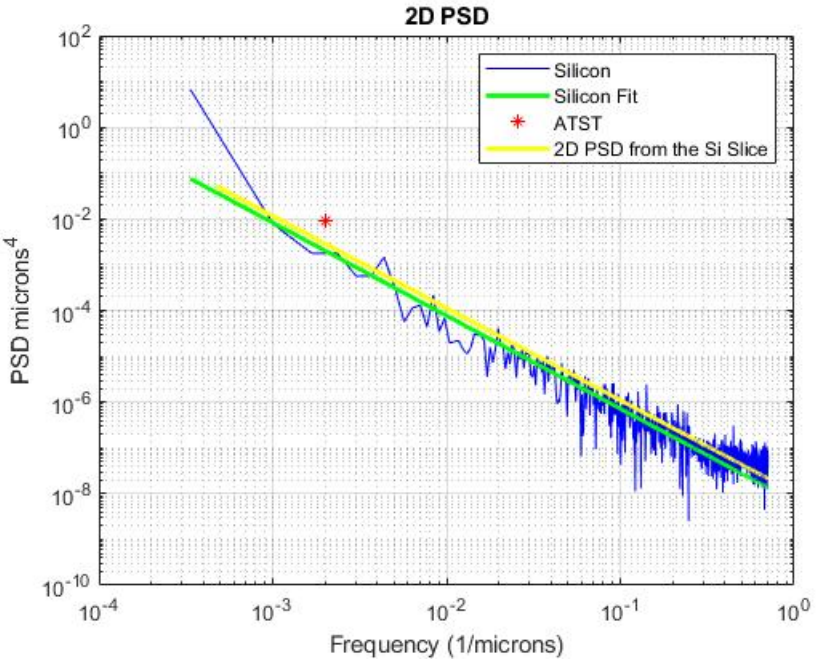

Figure 3: 2D PSD of the silicon and its power law fit compared to the ATST and the 2D PSD from the silicon slice

The plots seem to agree very well. If one were to only have a one dimensional set of data, they will have a very good idea of what the 2D PSD will look like. One can compare the 2D PSD from the silicon slice with the ATST and come to the conclusion that the silicon surface measured is indeed smooth because of how near the plot is to the ATST. Let's put this silicon into practice and compare a shape memory alloy surface to it.

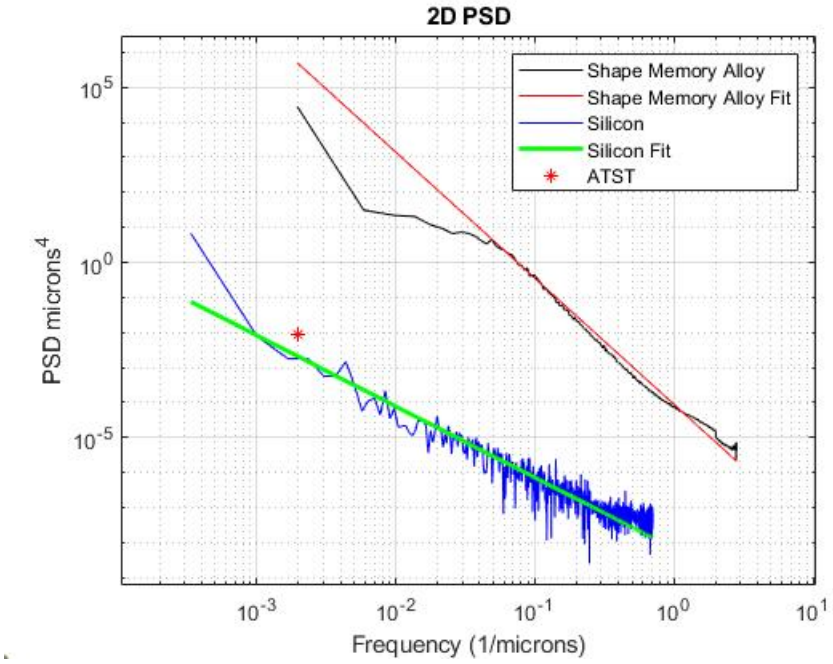

Figure 4: 2D PSD of a shape memory alloy surface plotted with silicon and the ATST

With this graph, one can see just how smooth this shape memory alloy surface really is. Because the shape memory alloy's 2D PSD is a lot higher than the 2D PSD of silicon, the shape memory alloy is not as smooth as the silicon. However, because the alloy has a clear negative slope, the smoothness of the shape memory alloy is not too bad. If the surface were a lot more rugged, the alloy's PSD would have been a lot more volatile.

7 Understanding the Olympus Confocal Microscope

In order to understand what the Olympus confocal microscope does when taking a 1D PSD, a 1D PSD of the shape memory alloy from the previous section will be taken through the MATLAB code on Page 9 and will be compared with the 1D PSD taken by the Olympus. If the outcomes are nearly the same, that shows that the MATLAB code explained in Section 2 and the Olympus share the same procedure in performing the 1D PSD.

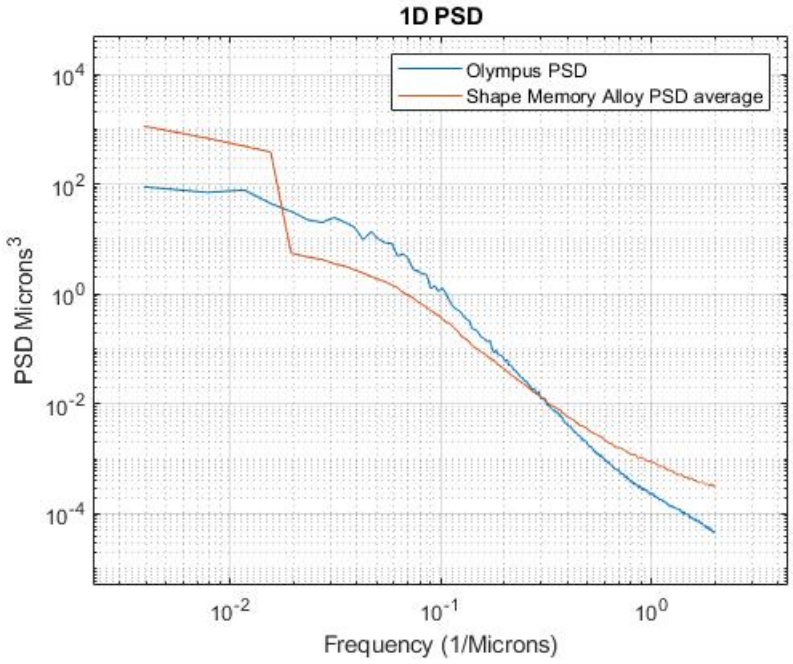

Figure 5: 1D PSD taken by Olympus compared with the 1D PSD taken by the MATLAB code

Seeing that the MATLAB PSD lands right on top of the Olympus PSD, it seems that the MATLAB code shares nearly the same understanding of performing a 1D PSD as the Olympus confocal microscope. This is a very good sign in that now one can have a good idea what exactly the Olympus is outputting.

8 Conclusion

Using the MATLAB code located below, one is able to carry out a 1D PSD and a 2D PSD of a surface roughness. If one were to only have a one dimensional slice of surface roughness data and wanted to compare it to two dimensional data, one can use a non-linear least-squares solver and Equations 5 and 6 to find an analogous 2D PSD to compare with the two dimensional data. Moreover, one can characterize a surface as smooth if its PSD goes through or near the ATST point. Lastly, one can have a very good understanding of what the Olympus confocal microscope does by understanding the 1D PSD procedure explained on Page 1.

9 MATLAB Code for Obtaining a 1D PSD

```
function [f,PSD,k,n] = psd_fit(data, fs, kpred, npred,
    start)
% MAKE INPUTS IN MICRONS

% INPUTS
% data is 1 dimensional height data (microns)
% fs is the sampling frequency (1/microns)
% kpred is your initial guess for k
% npred is your initial guess for n
% start is the index you would like the least squares fit
    to begin
%       in order to reduce influence from outliers
% OUTPUTS
% f is the frequency in units of 1/micron
% PSD is the power spectral density in units of microns^3
% k and n are the fits of the power law

x = data;
%orients data
if size(x,2) == 1
    x = x';
end
N = length(x); %size of original data

X = fft(x);
X_mag = abs(X); %magnitude of fft
new_bins = 0:length(X_mag(1:length(X)/2))-1;
f = new_bins/N*fs;

%Square fft to get PSD
PSD = 2*(X_mag.^2)/(N*fs);
PSD = PSD(1:end/2); %takes first half of PSD

%q is frequency
modelfun = @(b,q) b(1)./(q.^b(2));
%initial predictions
beta0 = [kpred,npred];
[coefficients,resnorm,~,exitflag,output] = lsqcurvefit(
    modelfun, beta0, f(start:end), PSD(start:end));

k = coefficients(1);
n = coefficients(2);
end
```

10 MATLAB Code for Obtaining a 2D PSD

```
function [f,PSD,k,n] = psd2_fit(data,px,py,lambda,kpred,
    npred,start)
% MAKE INPUTS IN MICRONS

% INPUTS
% data is 2 dimensional height data (microns)
% px is the pixel size in the x direction
% py is the pixel size in the y direction
% lambda is the wavelength used (microns)
% kpred is your initial guess for k
% npred is your initial guess for n
% start isthe index you would like the least squares fit
    to begin
%       in order to reduce influence from outliers
% OUTPUTS
% f is the frequency in units of 1/micron
% PSD is the power spectral density in units of microns^4
% k and n are the fits of the power law

roughness = data;

[f, PSD] =psd_2D_calculation(roughness,px,py,lambda);
f = f';

%q is frequency
modelfun = @(b,q) b(1)./(q.^b(2));
beta0 = [kpred,npred];
[coefficients,resnorm,~,exitflag,output] = lsqcurvefit(
    modelfun, beta0, f(start:end), PSD(start:end));

k = coefficients(1);
n = coefficients(2);
end
```