

SURFACE-ROUGHNESS CORRECTIONS TO EXTREME ULTRAVIOLET
THIN-FILM REFLECTANCE MEASUREMENTS

by

Elise Martin

A senior thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Bachelor of Science

Department of Physics and Astronomy

Brigham Young University

April 2007

Copyright © 2007 Elise Martin

All Rights Reserved

BRIGHAM YOUNG UNIVERSITY

DEPARTMENT APPROVAL

of a senior thesis submitted by

Elise Martin

This thesis has been reviewed by the research advisor, research coordinator,
and department chair and has been found to be satisfactory.

Date

R. Steven Turley, Advisor

Date

Eric Hintz, Research Coordinator

Date

Scott D. Sommerfeldt, Chair

ABSTRACT

SURFACE-ROUGHNESS CORRECTIONS TO EXTREME ULTRAVIOLET THIN-FILM REFLECTANCE MEASUREMENTS

Elise Martin

Department of Physics and Astronomy

Bachelor of Science

There are several traditional methods of accounting for the way that the roughness of a surface changes how light reflects. This paper introduces a new method based on arbitrarily exact solutions of Maxwell's equations solved computationally for reflectance from rough surfaces. This method leads to a correction factor for the changed reflectance and is compared to the oft-used Debye-Waller factor. Where the Debye-Waller factor contains only a quadratic factor in the exponential, the computational fitting of reflectance data demonstrates the need for cubic and linear terms in addition to a quadratic term.

ACKNOWLEDGMENTS

I would like to thank all those who made this work possible. Especially deserving is Dr. Turley, who put in countless hours of work and unending amounts of patience to help me understand and complete this research. My thanks to the BYU Thin Films group, those who provided funding, and all others who helped me with the details of this project. I am also extremely grateful to my parents for supporting me in all my endeavors and for the encouragement and love that they always give me. And to my Heavenly Father, for making such an interesting world and letting me study it.

Contents

Table of Contents	i
List of Figures	ii
1 Introduction	1
1.1 Applications of Optics in the Extreme Ultraviolet	1
1.2 Finding Optical Constants of Thin Films	2
1.3 Effects of Roughness on Reflectance Measurements	3
2 Finding a Correction Factor	6
2.1 Standard Correction Factors	6
2.2 Creating a Computational Model of Rough Surfaces	8
2.3 Finding a Correction Factor for a Perfect Conductor	8
3 Conclusions	11
3.1 Comparison of the Debye-Waller Correction Factor to Computational Data	11
3.2 Future Research	29
Bibliography	31
A Matlab Codes	33
Index	44

List of Figures

1.1	Reflection and transmission diagram	3
1.2	TEM image of a Scandium film	4
1.3	Plane wave reflection	5
2.1	Fitted reflectance	7
2.2	Roughness frequency	9
3.1	Reflectance versus angle	12
3.2	Rough Surface	12
3.3	Electric field reflected from a rough surface	13
3.4	$-\ln(R)$ as a function of qh	15
3.5	Fit for roughness frequency of 1	16
3.6	Fit for roughness frequency of 3	17
3.7	Fit for roughness frequency of 5	18
3.8	Fit for roughness frequency of 7	19
3.9	Fit for roughness frequency of 9	20
3.10	Fit for roughness frequency of 11	21
3.11	Fit for roughness frequency of 13	22
3.12	Fit for roughness frequency of 15	23
3.13	Fit for roughness frequency of 17	24
3.14	Fit for roughness frequency of 19	25
3.15	Fit of the cubic coefficient	26
3.16	Fit of the quadratic coefficient	27
3.17	Fit of the linear coefficient	28

Chapter 1

Introduction

1.1 Applications of Optics in the Extreme Ultraviolet

The extreme ultraviolet (EUV) wavelength range is becoming increasingly important to many areas of science and technology, such as space-based astronomy and photolithography. Various celestial objects, such as the sun, emit light in the extreme ultraviolet wavelength range. In order to observe these emissions, optical instruments are needed that work well in the EUV. The importance of EUV astronomy was shown when an observation satellite exclusively for EUV astronomy was launched by NASA in 1992. The mission completed an all-sky survey as well as the first EUV detection of extragalactic objects [1]. Astronomical observations in the EUV, specifically magnetosphere imaging of the Earth, were also performed by the 2000 IMAGE mission. The satellite that made these observations contained an EUV mirror created by researchers at BYU [2].

EUV wavelengths are also important to photolithographic processes. Photolithography is used to create computer chips. Patterns are etched on wafers by reflecting

light off of a larger mask and through reducing optics onto the wafer. EUV lithography is beneficial because its smaller wavelength size results in the capability of producing smaller patterns. This involves the use of multilayer mirrors and other optics that must work well with EUV light [3].

In order to make the special EUV optics that are needed for these applications, the optical constants of the materials must be known. The problem with making effective optics in the EUV is that most materials are extremely absorptive at these wavelengths. To reflect well, multilayer mirrors are needed so that the reflection of each layer will be constructively added. To optimize the reflectivity of these mirrors, both the real and the imaginary parts of the index of refraction must be known for the material that comprises each layer.

1.2 Finding Optical Constants of Thin Films

The optical constants of materials are found through reflectance and transmission of light incident on a surface. Reflectance and transmission are related to the Fresnel coefficients which, in turn, can be used to find the real and imaginary parts of the index of refraction.

For s-polarized light reflecting from a single interface,

$$R = |F|^2 \quad (1.1)$$

$$F = \frac{E_r}{E_i} = \frac{n_i \sin \theta_i - n_t \sin \theta_t}{n_i \sin \theta_i + n_t \sin \theta_t}. \quad (1.2)$$

where R is reflectance, F is the Fresnel coefficient, n_i is the index of refraction of the incident material (vacuum in this case), n_t is the index of refraction of the material in question, E_r is the reflected electric field, E_i is the incident electric field, θ_i is the angle of incidence, and θ_t is the transmitted angle.

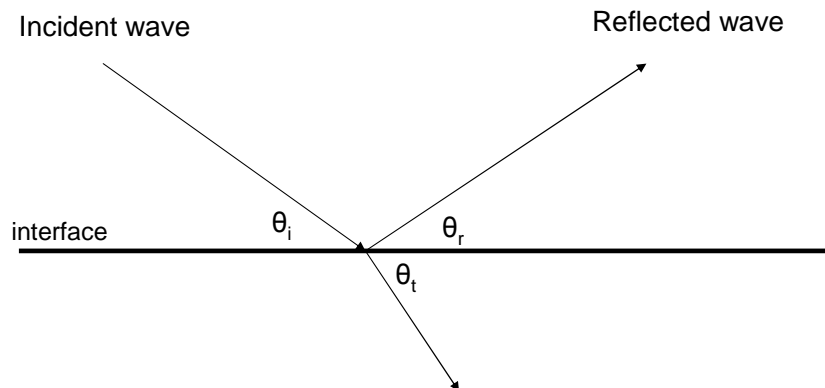


Figure 1.1 Reflection and Transmission Diagram. This is a diagram of an electromagnetic wave interacting with an interface where θ_i is the angle of incidence, θ_r is the angle of reflection and θ_t is the angle of transmission

Eq. (1.2), which has been taken from Ref. [4] and modified for angles taken from grazing, can be solved for n_t , the complex index of refraction, which affects the wavelength and the attenuation of the wave in the material. Once the index of refraction is known, it can be used to see if the material would be useful for building multilayer mirrors or other optical components that are used in the extreme ultraviolet.

Effective multilayer mirrors must reflect very well EUV wavelengths. To do this, the mirror must not be very absorptive, there must be a large difference between the indices of refraction of the alternating layers, and the phases must be matched so that the waves will interfere constructively.

1.3 Effects of Roughness on Reflectance Measurements

Unfortunately, the thin films used to find optical constants are not ideal. One trait of thin films that affects reflection, which affects the optical constants that are solved

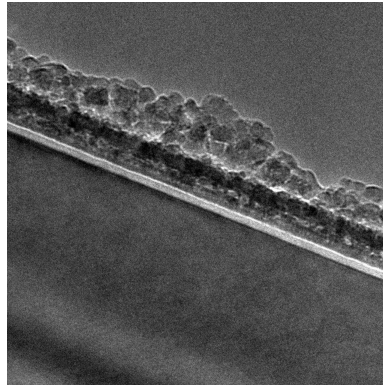


Figure 1.2 TEM Image of a Scandium Film. This TEM cross-sectional image of a scandium oxide film, taken at BYU, shows the roughness of the film. The top layer seen residue from preparing the surface for the TEM. The actual surface is the dark layer beneath the top layer, which shows significant roughness.

from reflectance, is the surface roughness of the film. Stearns notes that roughness reduces reflectivity and creates non-specular scattering [5]. When the incident wave has a wavelength that is small compared to the dominant spatial frequency, then the reflection can be traced using geometric optics. If the surface is anything but completely smooth, then light rays will reflect off at different angles due to the differing angles on the surface. This is called non-specular scattering. When this happens, some of the light does not enter the finite detector, and the perceived intensity is less than what is actually reflected. Also, when the wavelength is large compared to the dominant spatial frequency, then the light can be treated as a plane wave. If this plane wave is incident on a rough surface, where it impinges on atoms at different heights, then each part of the plane wave will be reflected at a different phase. See Fig. 1.3. This will also cause the reflected intensity to be less because the reflected wave will interfere destructively. Both of these effects inhibit the measurements needed for finding the index of refraction. In order to calculate reliable optical constants from measured reflectance data, a correction factor must be found that compensates for

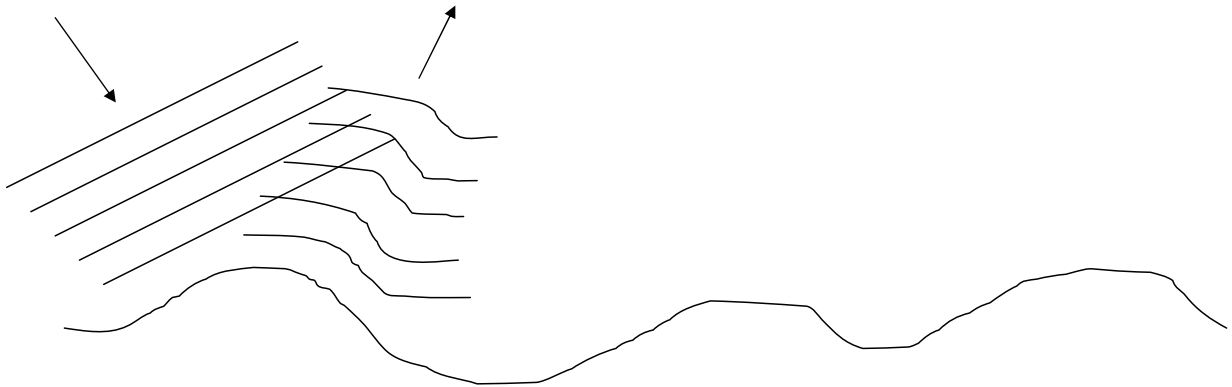


Figure 1.3 Plane Wave Reflection. A plane wave will reflect from a rough surface with different phases.

the roughness of the film. This paper shows a new method of correcting for roughness that is an improvement on previously used correction factors.

Chapter 2

Finding a Correction Factor

2.1 Standard Correction Factors

There are several standard correction factors that take into account the affect of surface roughness on reflectance. The Debye-Waller factor attempts to correct for the roughness of a single interface through the following equation,

$$R = R_0 \exp \left[- \left(\frac{4\pi\sigma}{\lambda} n \sin \theta \right)^2 \right] \quad (2.1)$$

where R_0 is the reflectance without roughness, σ is the RMS roughness, n is the complex index of refraction, and θ is measured from grazing (see Fig. 1.1). The Debye-Waller factor [6] assumes a Gaussian distribution of roughness heights on the surface.

The Nevot-Croce factor [6] is similar to Debye-Waller, however it takes into account the difference in index of refraction between the two layers by taking an average of the wave vectors on either side of the interface.

$$R = R_0 \exp \left[-n_i n_t \sin \theta_1 \sin \theta_2 \left(\frac{4\pi\sigma}{\lambda} \right)^2 \right] \quad (2.2)$$

In her Honors Thesis [7], Niki Farnsworth shows that the Debye-Waller factor

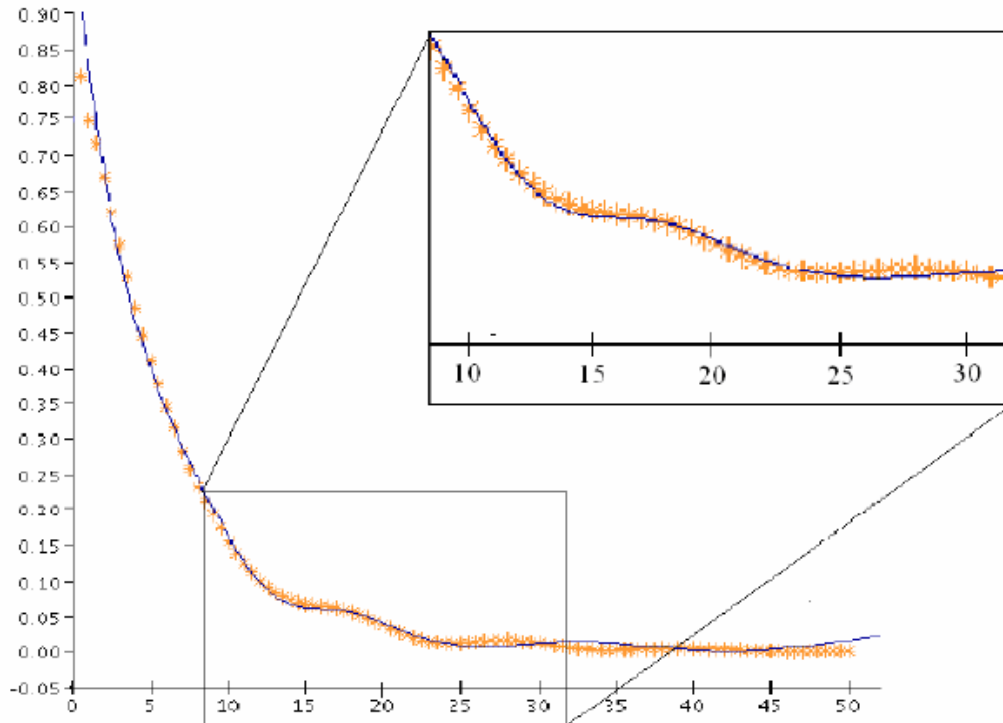


Figure 2.1 Fitted Reflectance. Fitted reflectance measurement using the finite differences approximation. Taken from reference [7].

improves fits of reflectance data at low angles and the Nevot-Croce factor improves the fits at low and high angles. She then uses a method of finite differences to account for oxidation, roughness, and linearly varying index of refraction. This improves the fits at the low and middle angles. See Fig. 2.1.

In his 1992 paper, D.G. Stearns describes a method of finding the nonspecular scattering of x-rays from rough multilayers [8]. Stearns' theory uses the first Born approximation and unfortunately is valid only for small roughness, when scattering is weak (usually at x-ray wavelengths), and at angles greater than the critical angle for total external reflection (as measured from grazing).

2.2 Creating a Computational Model of Rough Surfaces

As part of my project I used a program written by Jed Johnson [9] to find the relationship between surface roughness and reflectance. The program uses a spline to create the rough surface. In this process evenly spaced points are smoothly connected by cubic polynomials. See Fig. 2.2. This accounts for the effects of neighboring atoms on each other, which the Debye-Waller and Nevot-Croce factors do not include. Both Debye-Waller and Nevot-Croce assume a Gaussian distribution of roughness, however if the atoms affect the height of their neighbors, which they do in physical reality, then the roughness height frequency and the roughness spatial frequency will not be completely Gaussian. Therefore, a new factor is needed in order to more accurately adjust for roughness, and this factor can be found by using Johnson's program and solving for the arbitrarily exact reflectance of light from rough surfaces.

2.3 Finding a Correction Factor for a Perfect Conductor

Johnson's program finds arbitrarily exact solutions to Maxwell's equations for rough surfaces and then solves for the reflection. To do this he derives the Helmholtz equation from Maxwell's equations in source-free material. He solves the Helmholtz equation for the electric field by using the definition of the Green's function at a point on the boundary. He then integrates over the entire boundary to find the scattered electric field. Unlike Stearns', this program can be used with any amount of roughness, and range of angles, any wavelength, and total reflection (as in the case of a perfect conductor).

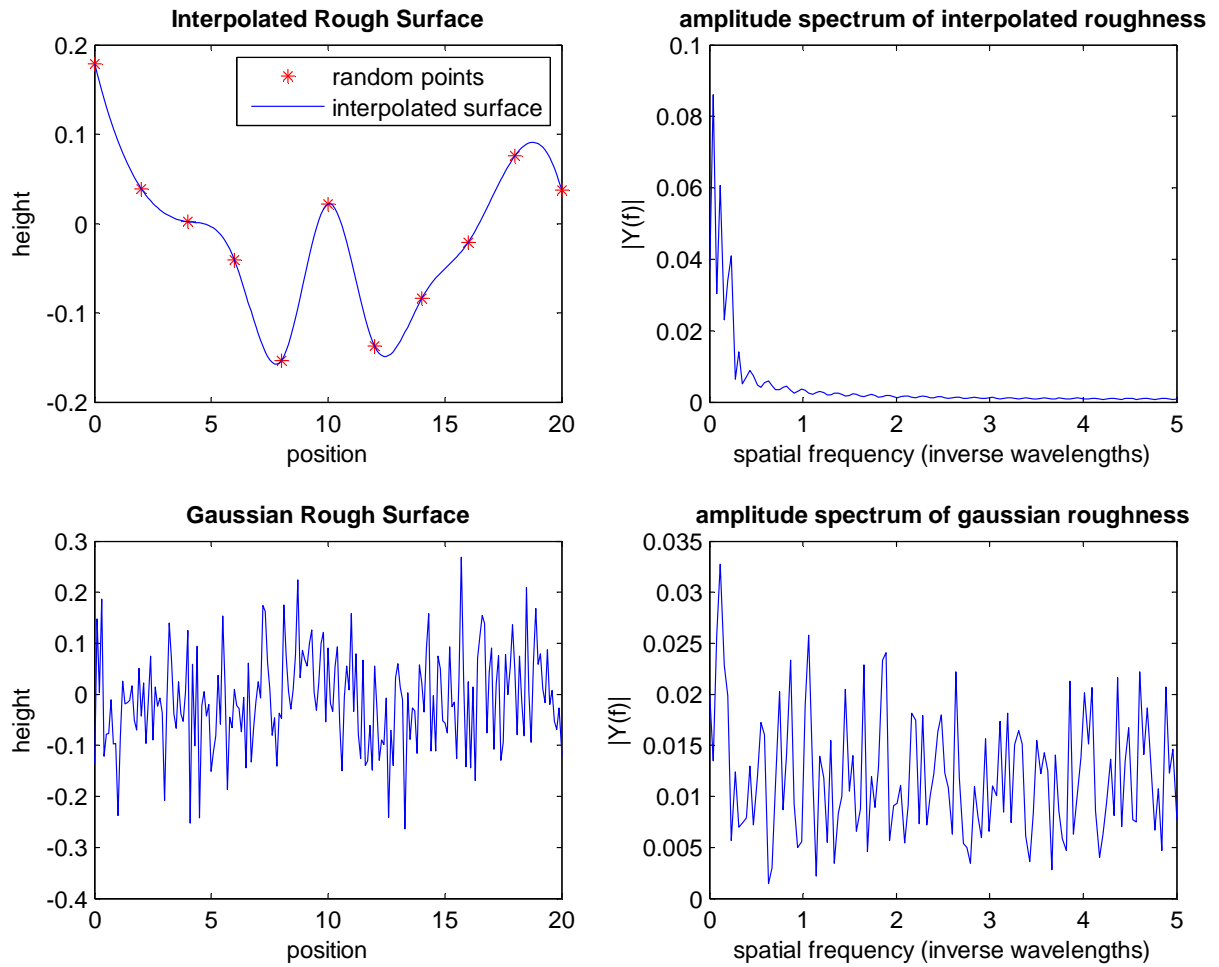


Figure 2.2 Roughness frequency. The graph of interpolated roughness shows Gaussian random points connected by a cubic spline. The accompanying amplitude spectrum shows that there is very little high spatial frequency. However, the Gaussian surface is just Gaussian random points connected directly. The accompanying amplitude spectrum shows that there is much more high spatial frequency.

From this program I can find the ratio of the electric field of a rough surface to a smooth surface of an s-polarized perfectly conducting surface, and then take the square to find the reflectance. Then I can vary the roughness height and roughness spacial frequency and fit the data to obtain a factor that corrects for roughness.

Chapter 3

Conclusions

3.1 Comparison of the Debye-Waller Correction Factor to Computational Data

To find a correction factor and compare it to Debye-Waller, I use Johnson's program and find the electric fields of s-polarized light reflecting off of a perfect conductor. I input the starting angle, the angle range, the frequency of the roughness, the height of the roughness, the length of the mirror and the number of patches. Wavelength is set at one. By running this program I get the electric field reflected off of the surface.

I ran ten surfaces at five roughness heights and ten roughness frequencies and 16 angles to find the electric fields reflected from the rough and smooth surfaces. I then found the Fresnel coefficients by taking the ratio of the maximums of the electric fields from the rough and smooth surfaces.

$$F = \frac{\max |E_r|}{\max |E_s|} \quad (3.1)$$

Fig. 3.3 shows these electric fields. The maximum peaks are used to avoid the diffraction peaks caused by reflecting from a finite surface.

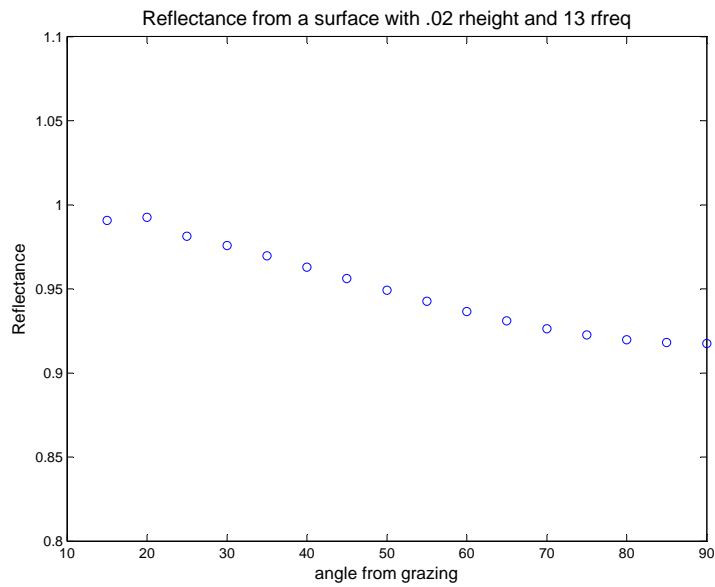


Figure 3.1 Reflectance Versus Angle. This graph shows reflectance versus angles for a surface with a roughness height of .02 wavelengths and a roughness frequency of 13 wavelengths.

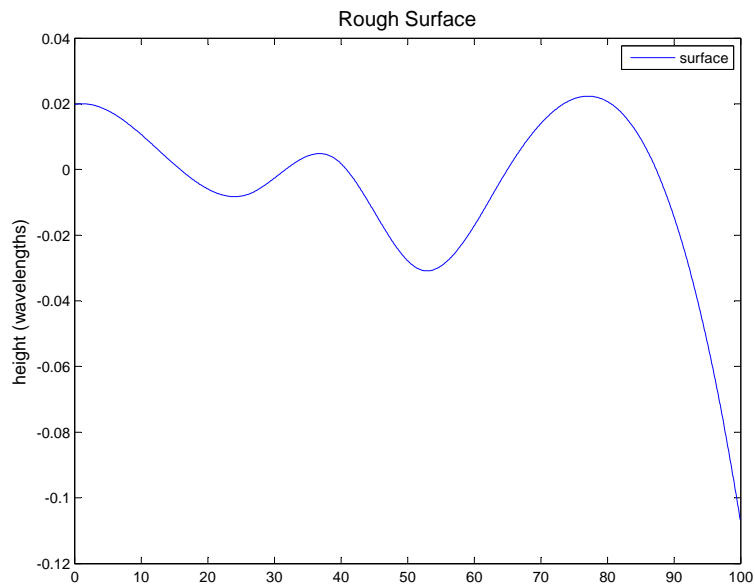


Figure 3.2 Rough Surface. This is the surface used to generate the data in the previous figure. It has a roughness height of .02 wavelengths and a roughness frequency of 13 wavelengths.

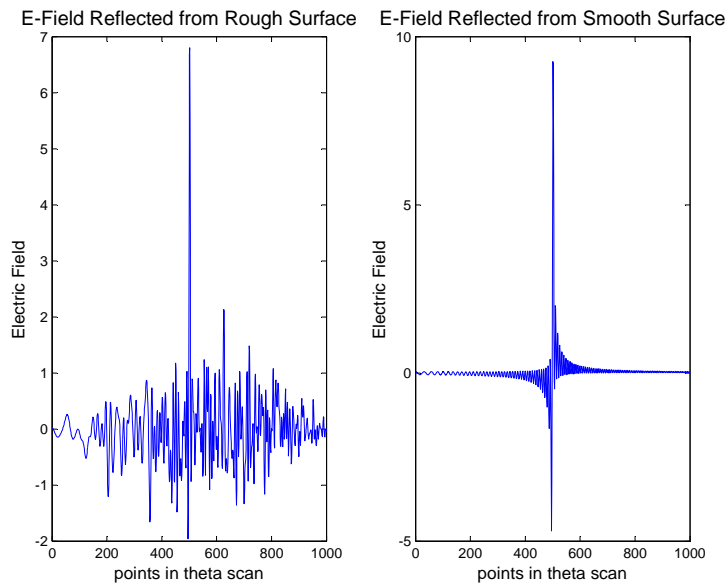


Figure 3.3 Electric Field Reflected from a Rough Surface. The left graph shows the electric field reflected from a surface with .1 roughness height and 1 roughness frequency. The right graph shows the electric field reflected from a perfectly smooth surface. The ratio of the peaks of these two is the Fresnel coefficient. The nonspecular peaks on the graph of the smooth surface are due to diffraction from the ends of the finite surface.

By squaring these Fresnel coefficients I find the reflectance of each roughness at each angle. I take the negative natural logarithm of each of these reflectances and find the mean across the ten surfaces. I also find the standard deviation of the negative natural logarithm given by:

$$\sigma = \left(\frac{1}{n-1} \sum_1^n (x_i - \bar{x})^2 \right)^{1/2}, \quad (3.2)$$

where x represents $-\ln(R)$. From this I can find the weights to use for fitting the reflectance.

$$w = \frac{1}{\sigma^2} \quad (3.3)$$

The Debye-Waller factor is $\exp(-\sigma q)^2$ where

$$q = 2 \left(\frac{2\pi}{\lambda} n \sin \theta \right) \quad (3.4)$$

and h is the roughness height. $\frac{2\pi}{\lambda}$ is the wave vector and $\sin \theta$ puts it in the perpendicular direction. In this case, λ is one because the program uses a wavelength of one, and n is one because that is the index of refraction of vacuum. So, to see if the reflectance data is well-modeled by Debye-Waller, I plot the $-\ln(R)$ against qh to see if a polynomial in qh fits the data better than Debye-Waller's single term of q^2h^2 . See Fig. 3.4. The different roughness heights lie on the same curve, which indicates that the $-\ln(R)$ is a function of qh , and not a function of the other permutations of q and h , such as q^2h^3 . Figs. (3.5-3.14) show that Debye-Waller is not the closest fit to the data, although it lies within the error bars except for roughness frequency of one. The error bars result from averaging five different surfaces for these fits. The constant term on the cubic polynomial fit has been suppressed because at a grazing angle of zero the reflectance should be 1, or complete reflectance, and $-\ln(1) = 0$.

Fitting the data with cubic polynomials for five roughness frequencies results in three coefficients that must also be fit because they depend on the roughness

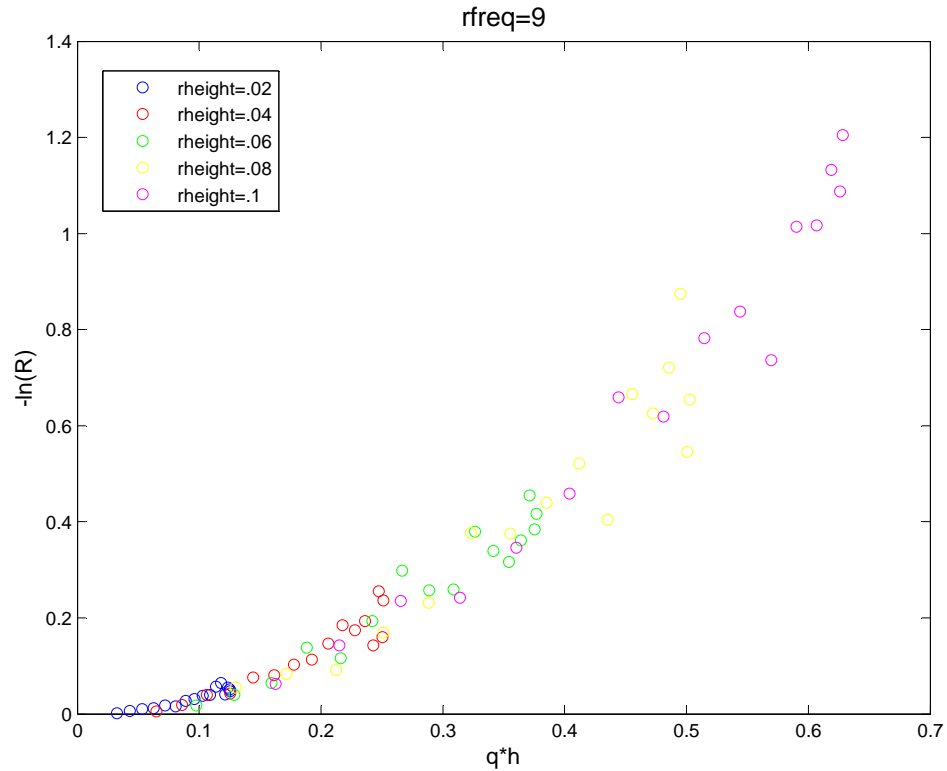


Figure 3.4 $-\ln(R)$ as a function of qh . This graph shows the negative natural logarithm of the reflectance as a function of q times the roughness height. The different colors are data points from different roughness heights. The different roughness heights lie on the same curve, which indicates that the $-\ln(R)$ is a function of qh .

frequency. Figs. (3.15-3.17) show the fits of the coefficients. The error bars result from the fit uncertainties of the reflectance fits. Error bars are obtained from the 95% confidence bounds that Matlab produces automatically by taking the difference between the maximum and minimum bounds and dividing by 4.2. This factor of 4.2 is needed because it converts the confidence bounds from the Student T distribution that Matlab automatically generates to the more useful standard deviation [10].

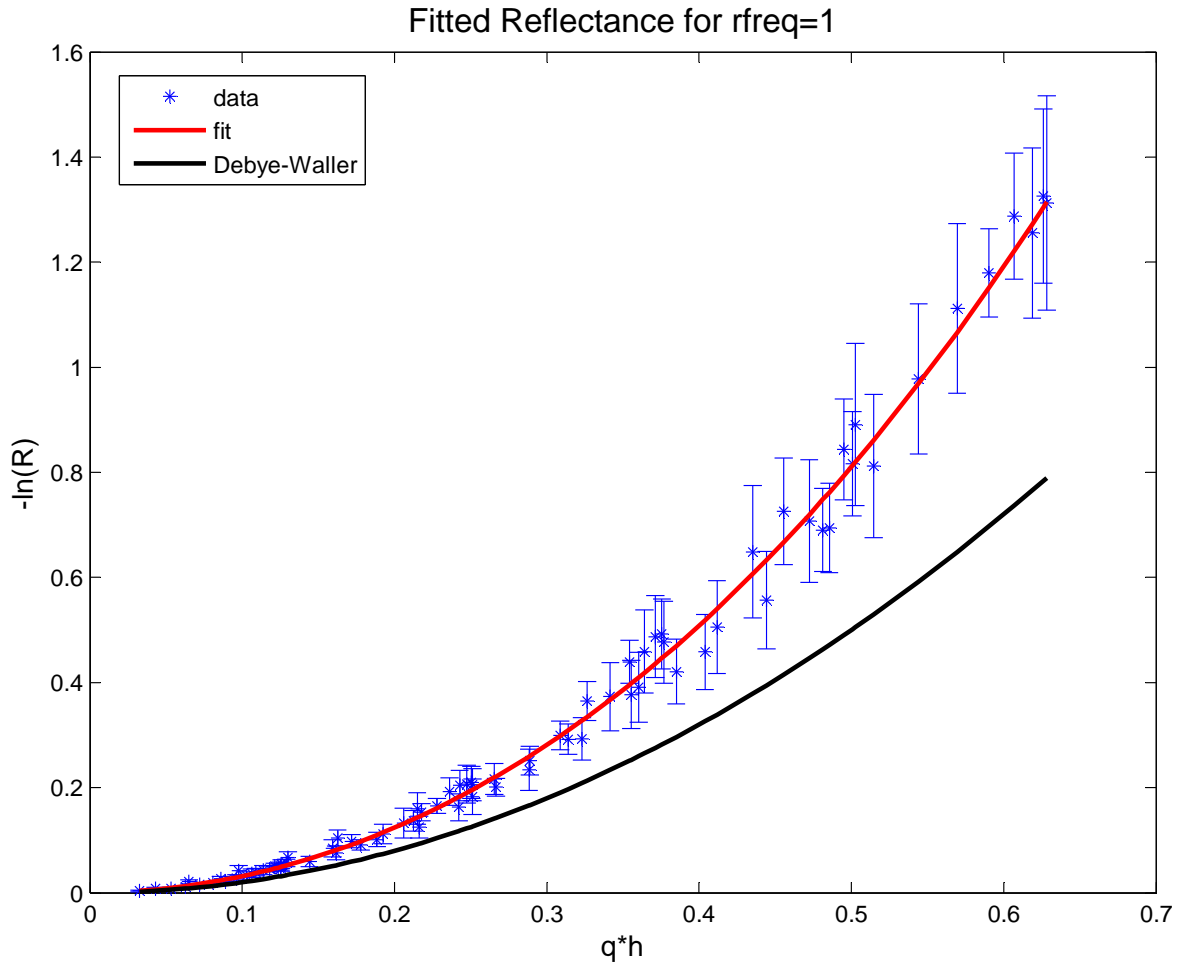


Figure 3.5 Fit for Roughness Frequency of 1. A cubic polynomial fit to reflectance data for a roughness frequency of 1.

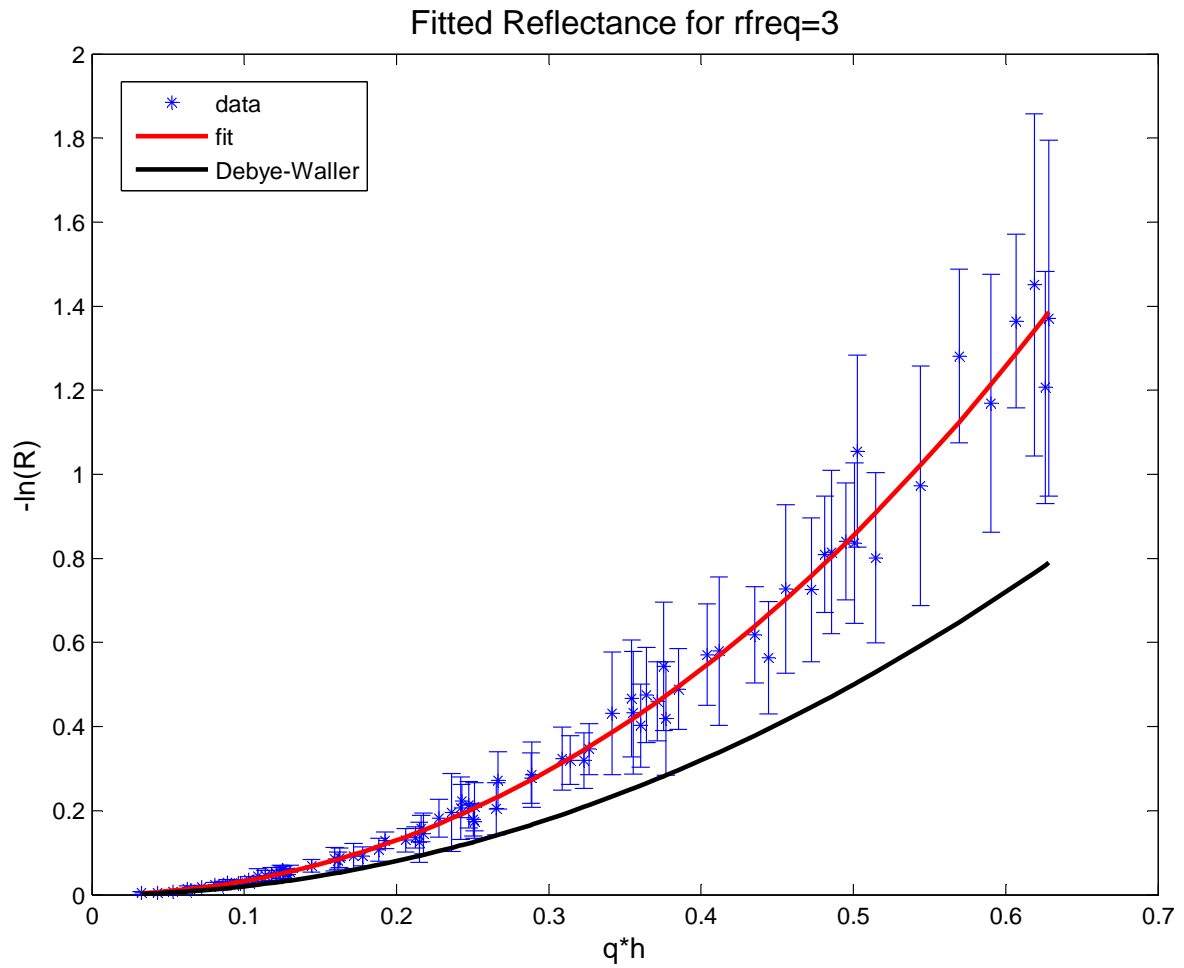


Figure 3.6 Fit for Roughness Frequency of 3. A cubic polynomial fit to reflectance data for a roughness frequency of 3.

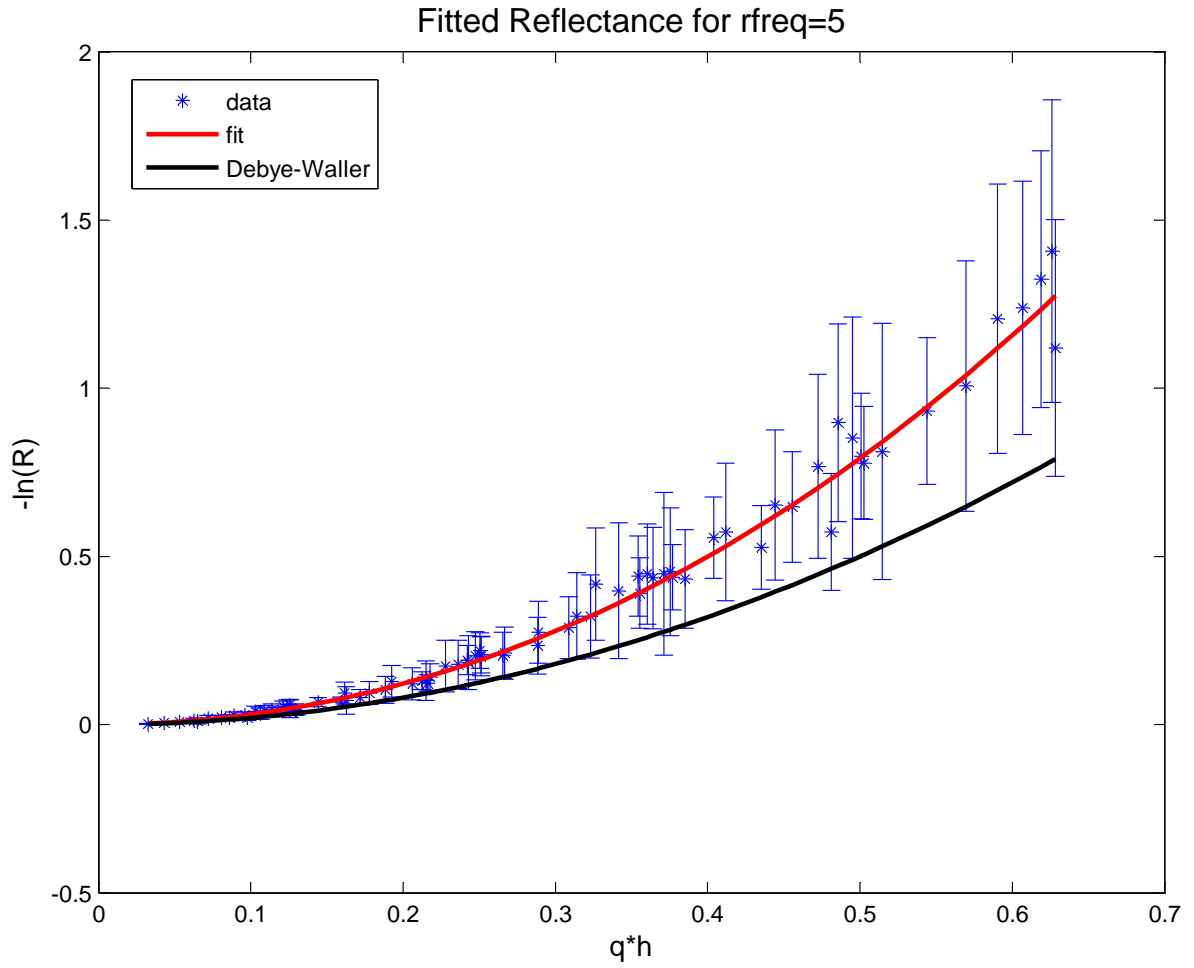


Figure 3.7 Fit for Roughness Frequency of 5. A cubic polynomial fit to reflectance data for a roughness frequency of 5.

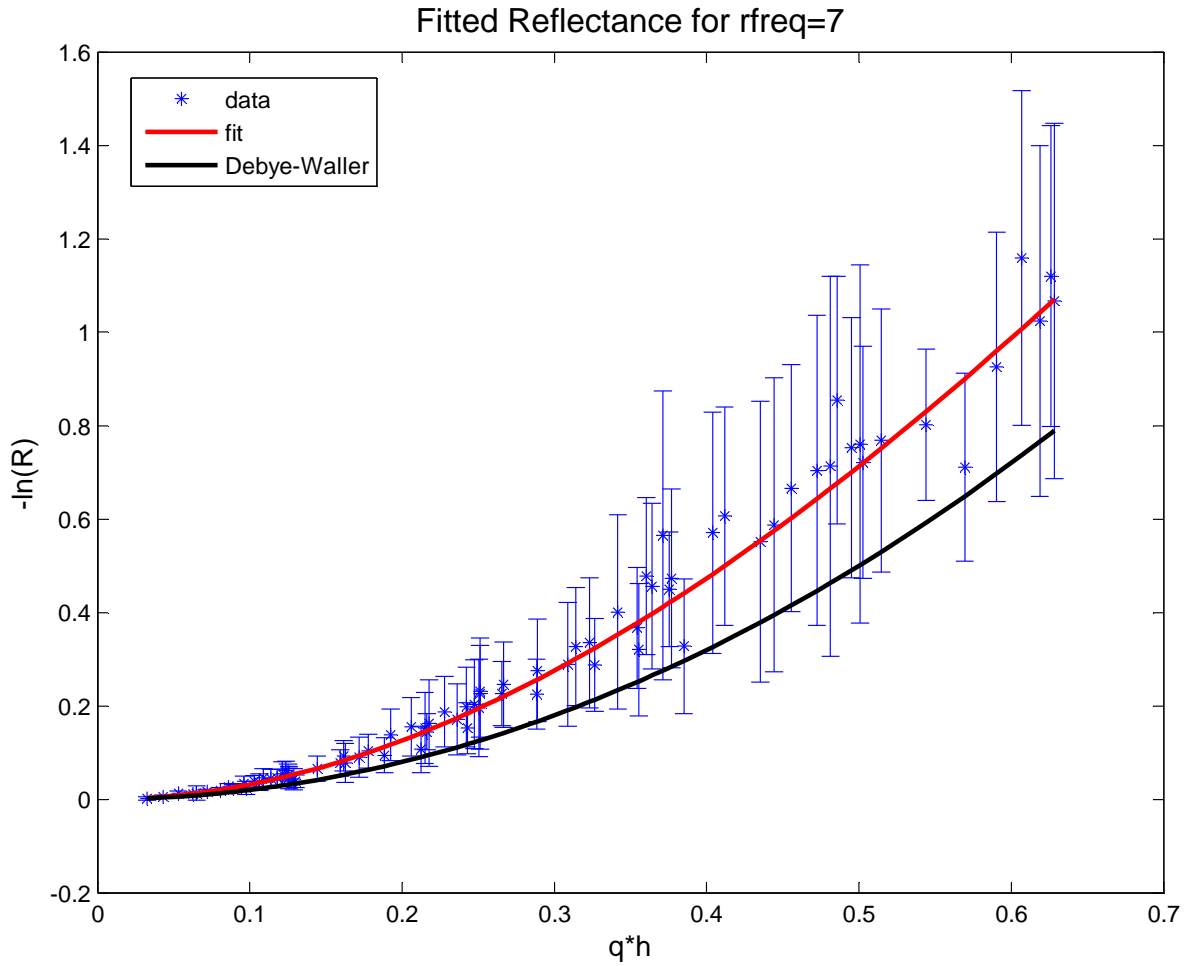


Figure 3.8 Fit for Roughness Frequency of 7. A cubic polynomial fit to reflectance data for a roughness frequency of 7.

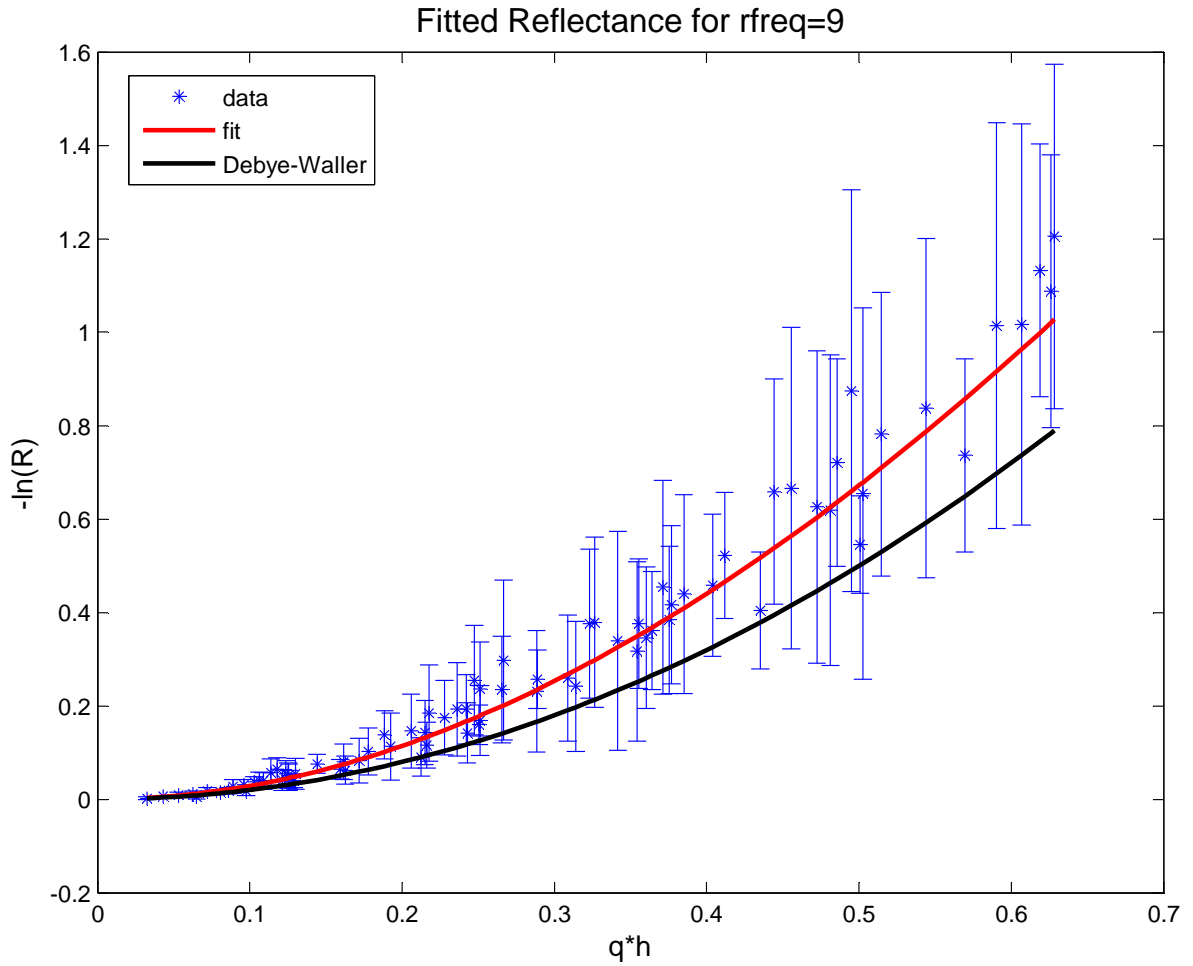


Figure 3.9 Fit for Roughness Frequency of 9. A cubic polynomial fit to reflectance data for a roughness frequency of 9.

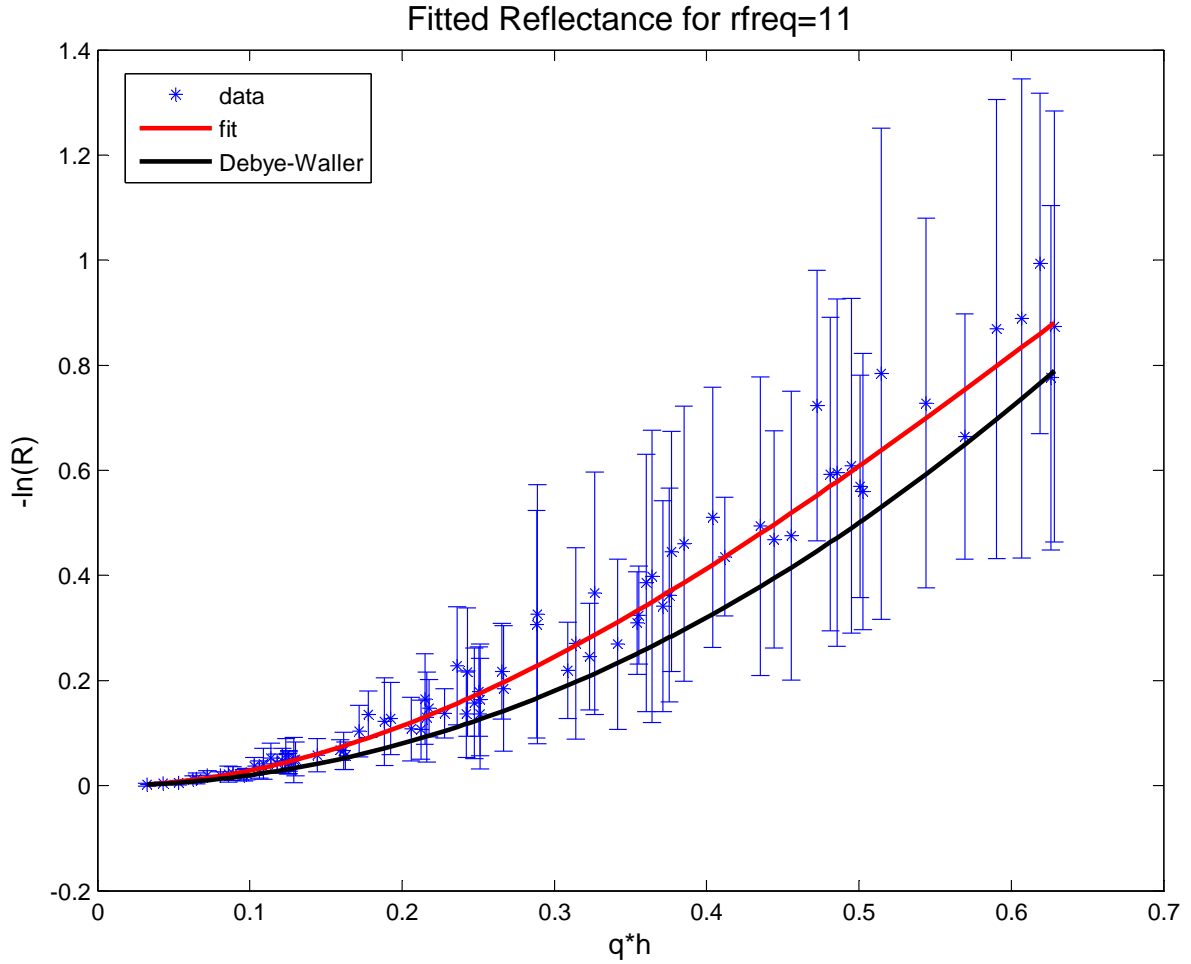


Figure 3.10 Fit for Roughness Frequency of 11. A cubic polynomial fit to reflectance data for a roughness frequency of 11.

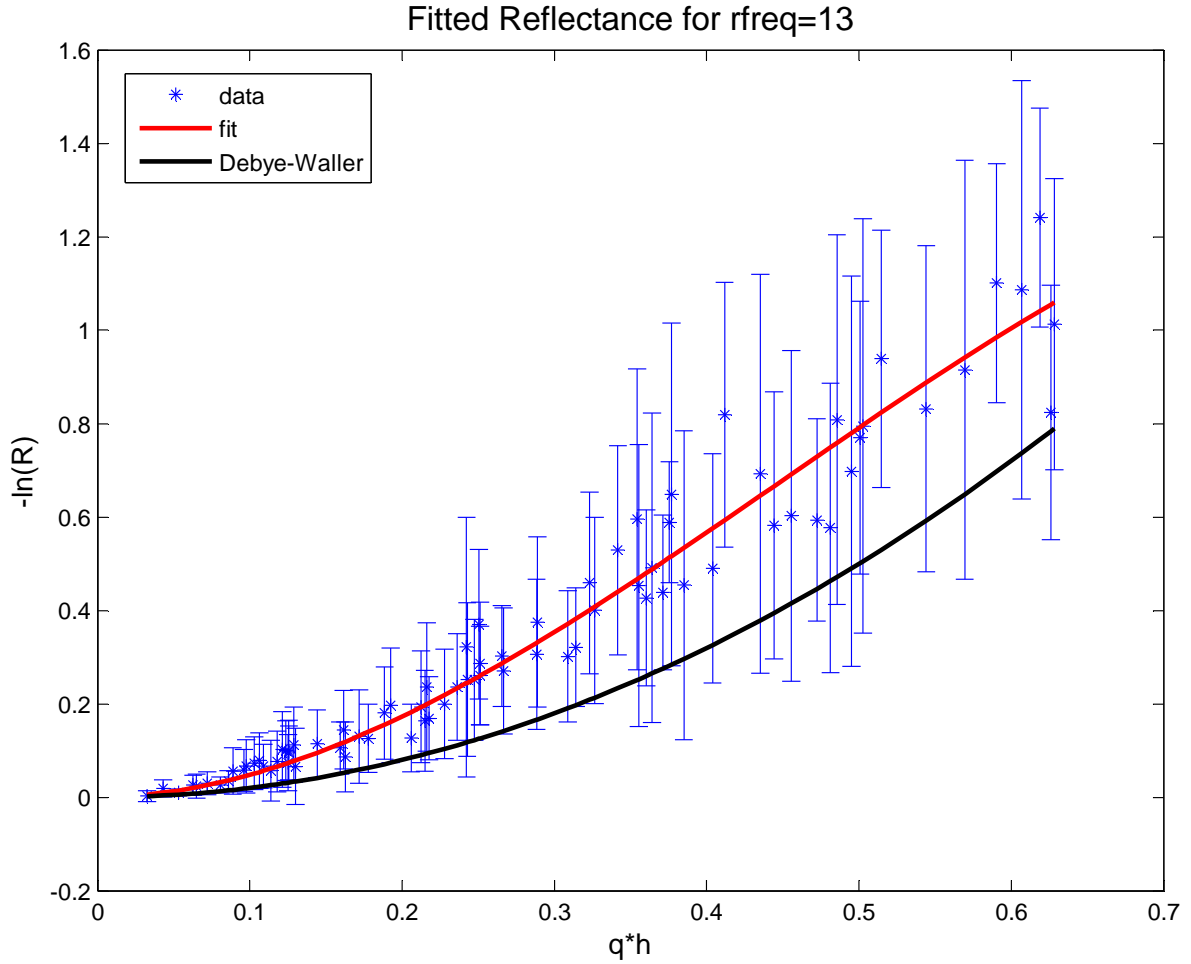


Figure 3.11 Fit for Roughness Frequency of 13. A cubic polynomial fit to reflectance data for a roughness frequency of 13.

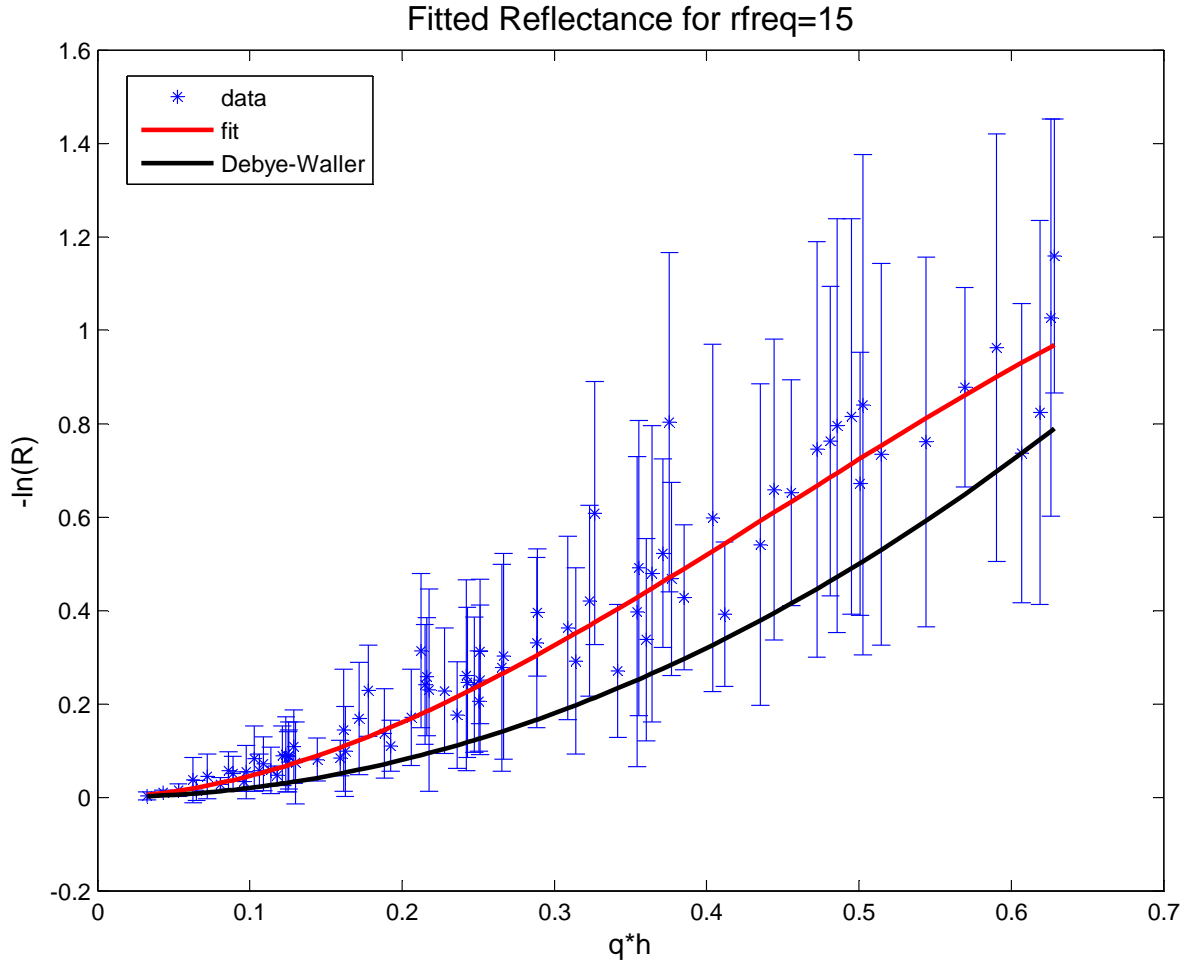


Figure 3.12 Fit for Roughness Frequency of 15. A cubic polynomial fit to reflectance data for a roughness frequency of 15.

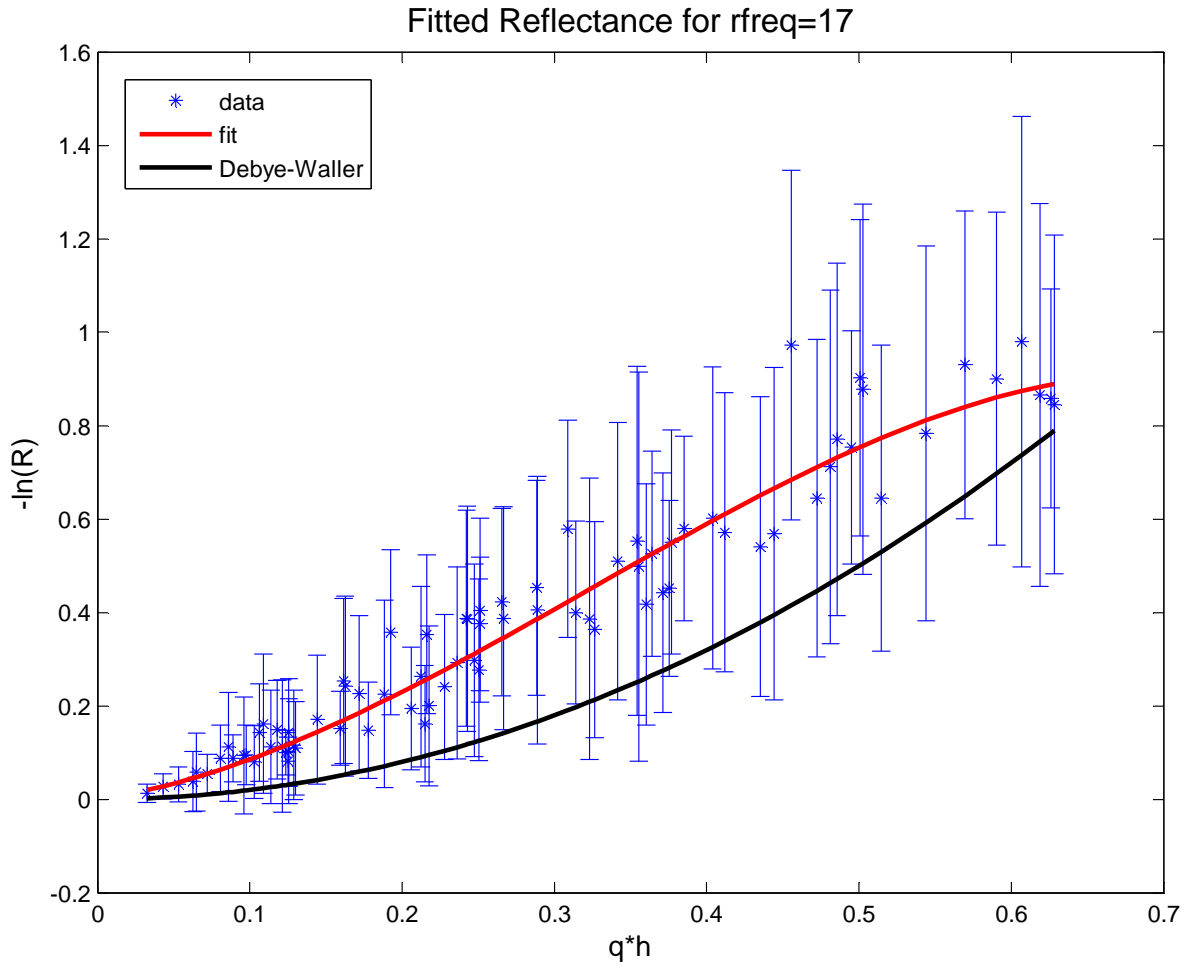


Figure 3.13 Fit for Roughness Frequency of 17. A cubic polynomial fit to reflectance data for a roughness frequency of 17.

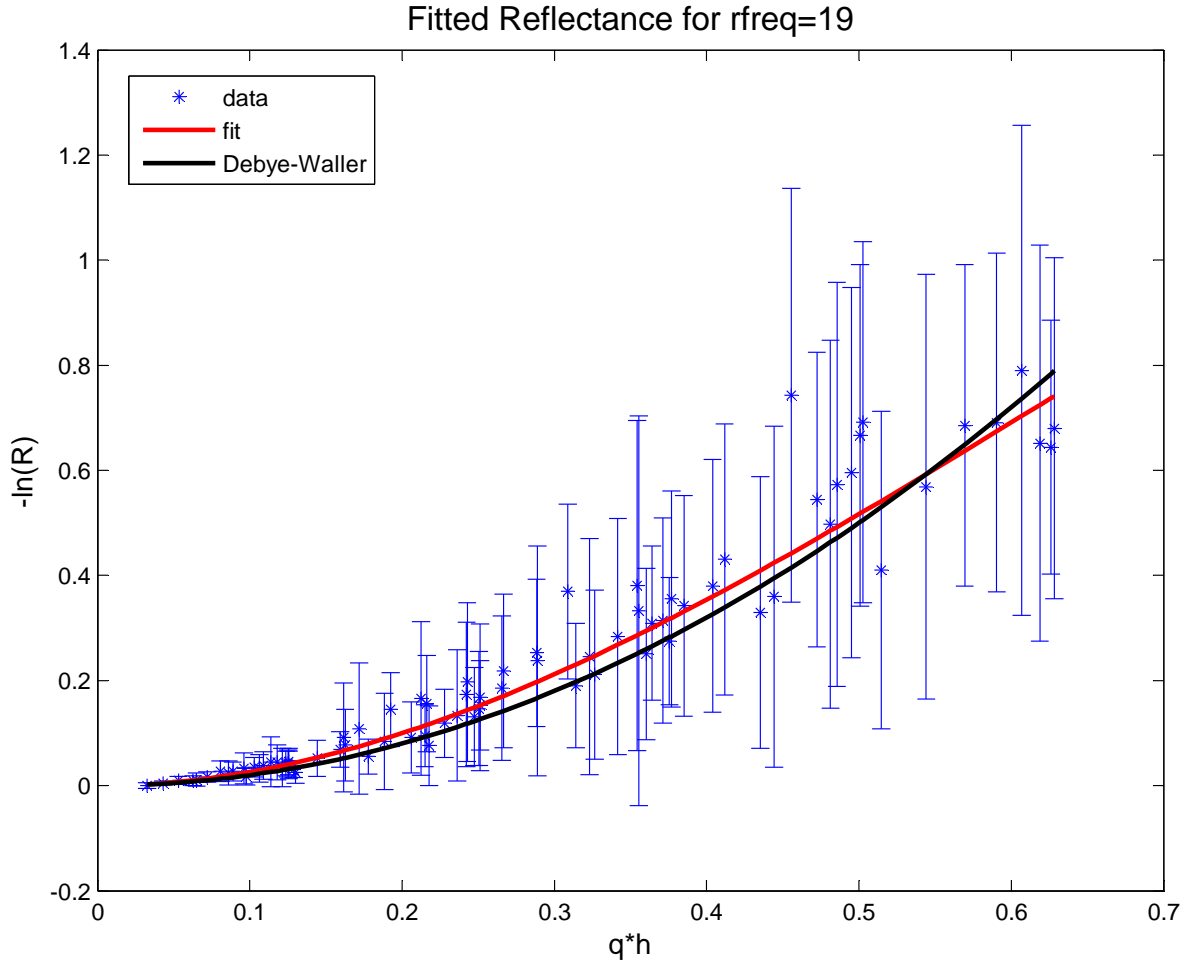


Figure 3.14 Fit for Roughness Frequency of 19. A cubic polynomial fit to reflectance data for a roughness frequency of 19.

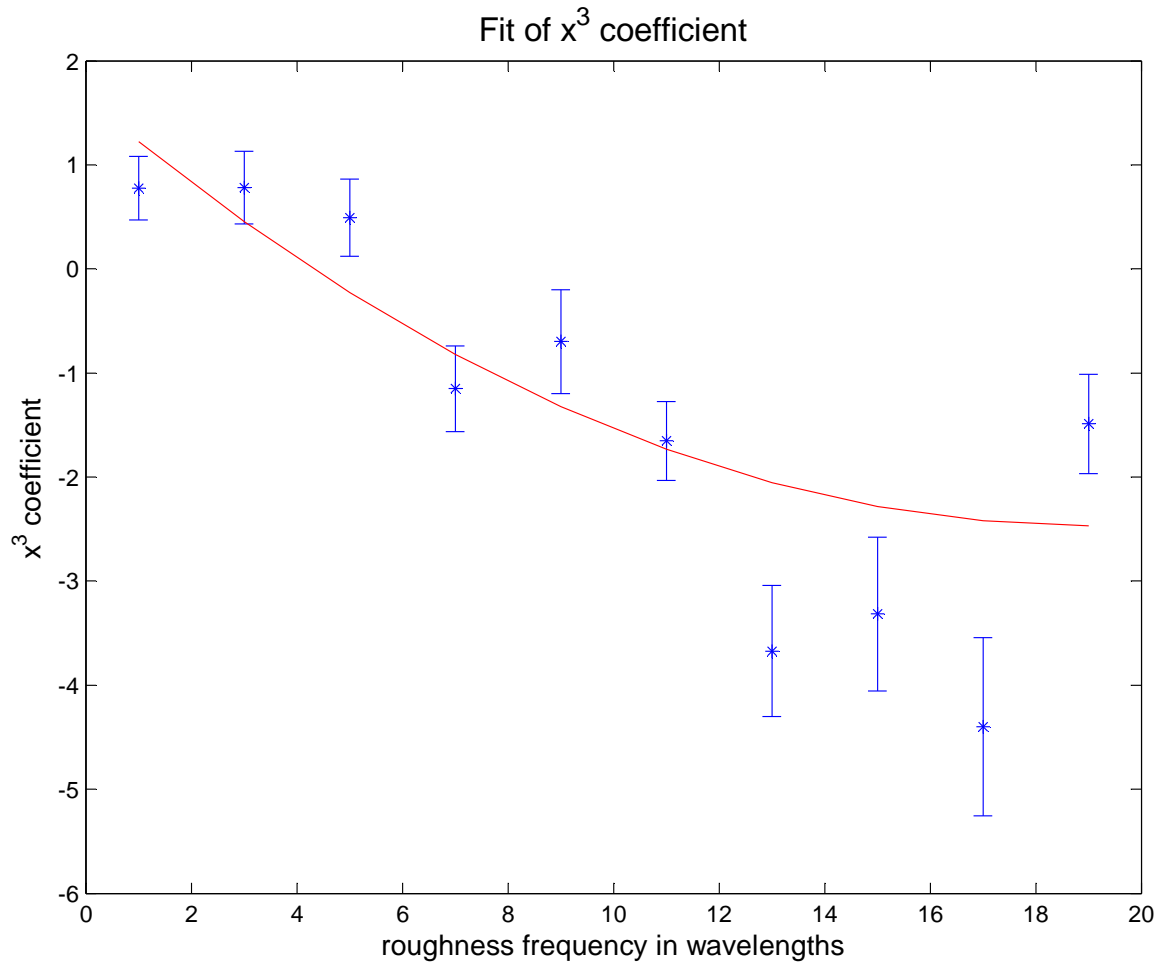


Figure 3.15 Fit of the Cubic Coefficient. This is the fit of the cubic coefficient from the cubic polynomial that resulted from fitting the $-\ln(R)$.

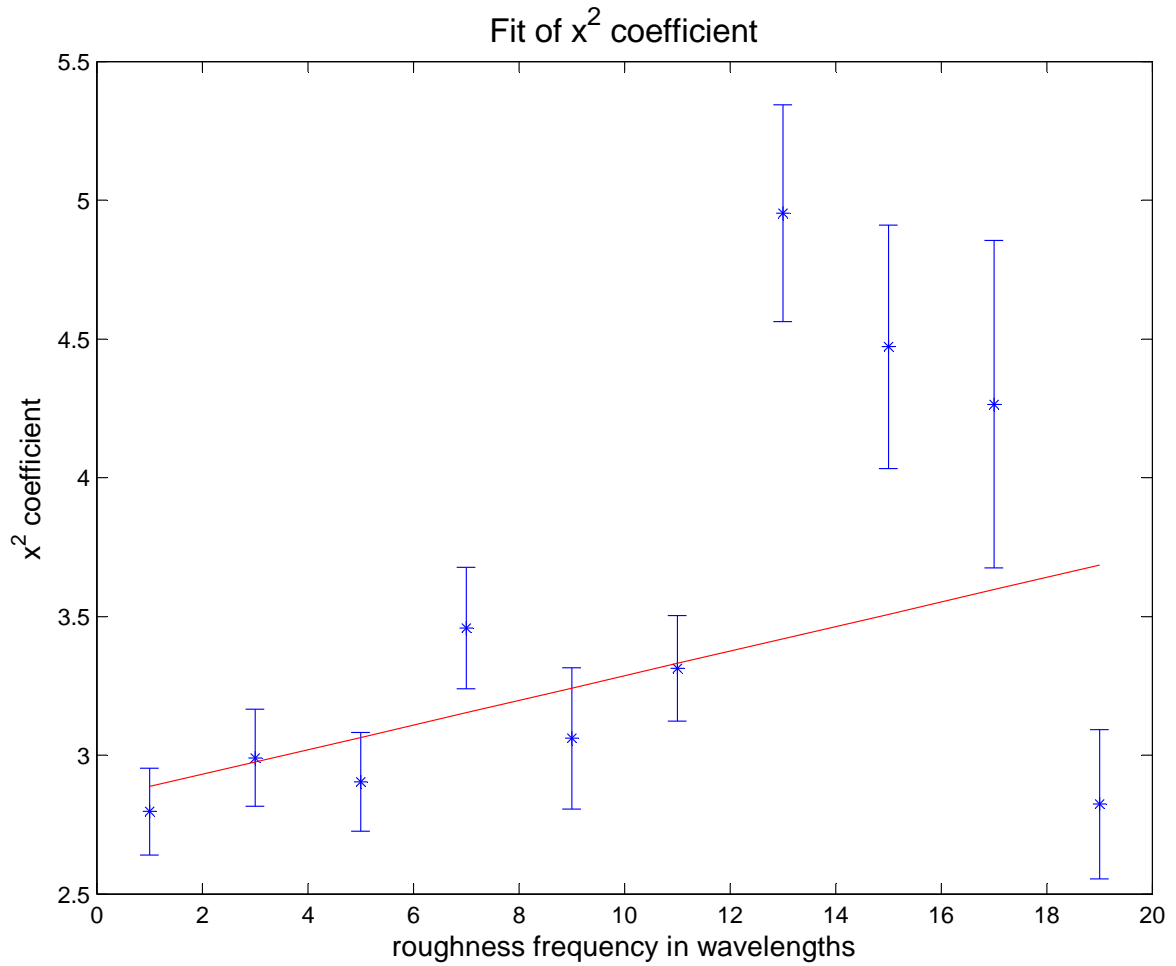


Figure 3.16 Fit of the Quadratic Coefficient. This is the fit of the quadratic coefficient from the cubic polynomial that resulted from fitting the $-\ln(R)$.

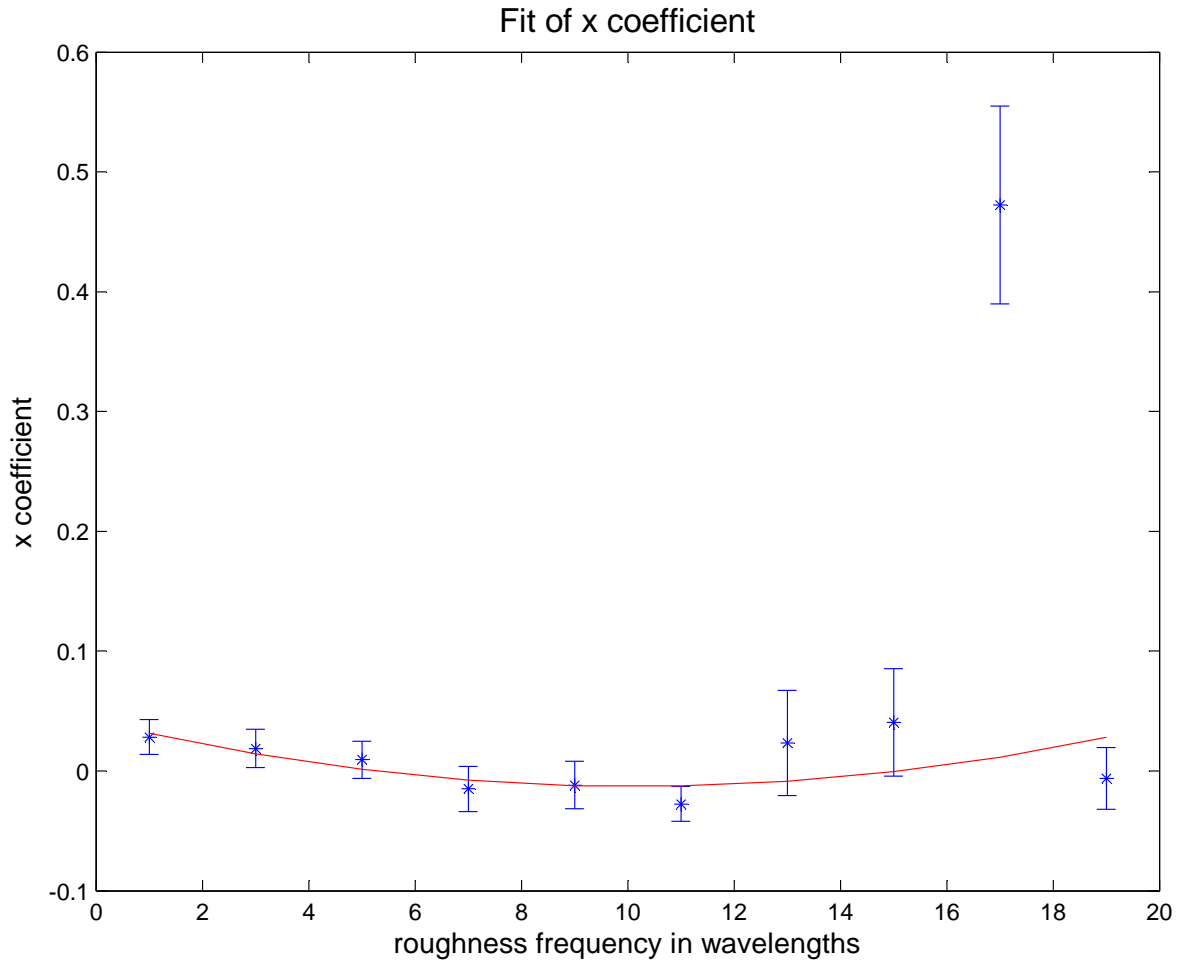


Figure 3.17 Fit of the Linear Coefficient. This is the fit of the linear coefficient from the cubic polynomial that resulted from fitting the $-\ln(R)$.

	quadratic term	linear term	constant term
α	0.01	0.1935	0.7212
β	0	0.0285	1.0116
γ	0.0027	0.202	-0.0017

Table 3.1 Error of Coefficient Fits

The resulting correction factor is

$$\exp(\alpha q^3 h^3 + \beta q^2 h^2 + \gamma q h) \quad (3.5)$$

and

$$\alpha = 0.01131r^2 - 0.4315r + 1.644 \quad (3.6)$$

$$\beta = 0.04435r + 2.842 \quad (3.7)$$

$$\gamma = 0.0005273r^2 - 0.01074r + 0.04171 \quad (3.8)$$

where r is the roughness frequency. Table 3.1 shows the error in the fits of the coefficients. The quadratic and linear terms in γ are consistent with zero, due to the fact that the error is larger than the actual value.

This correction factor shows that the correction factor has cubic, quadratic, and linear terms, unlike Debye-Waller, which only has the quadratic term. It also shows that the correction factor depends on the roughness frequency of the surface, and not just the roughness height.

3.2 Future Research

More surfaces should be used to see what holds for a variety of random surfaces. More roughness frequencies should be used to better see how the correction factor

will change with the roughness frequency. Of particular interest is the deviations in data trends and linear coefficient fits when the roughness frequency is 17. In addition, the Debye-Waller factor uses the RMS roughness height, so where just the roughness heights are used in the fitting, the RMS roughness heights should be computed and used.

This research should be extended to include dielectric multilayers with both s- and p- polarized light. When the dielectrics are involved in the calculations then the Nevot-Croce can also be compared to the results. Nevot-Croce is unusable with the present data because it requires two indices of refraction, and in this case it is a perfect conductor and so there is only one non-zero index of refraction. Also, to remove the diffraction effects of the finite surface, experimenting with a larger surface or a finite beam shape would be helpful. Data that includes these new parameters can be fitted to find a factor that is more exact than the Debye-Waller and other current factors that correct for roughness. Once this factor is found, it can be used to correct reflectance measurements for the roughness of the film. This will help calculations of optical constants to become more accurate.

Bibliography

- [1] “The EUVE Observatory,” <http://heasarc.nasa.gov/docs/euve/euve.html> (2007).
- [2] S. Lunt, “The Use of Genetic Algorithms in Multilayer Mirror Optimization,” Brigham Young University, Honors Thesis (1999).
- [3] *Soft X-rays and Extreme Ultraviolet Radiation: Principles and Applications*, 1st ed. (Cambridge University Press, Cambridge, England, 1999).
- [4] *Physics of Light and Optics*, 1st ed. (Brigham Young University, Provo, UT, 2006).
- [5] D. G. Stearns and E. Gullikson, “Nonspecular scattering from extreme ultraviolet multilayer coatings,” *Physica B* **283**, 84–91 (2000).
- [6] D. K. G. de Boer, “Influence of the roughness profile on the specular reflectivity of x rays and neutrons,” *Physical Review B* **49**, 5817–5820 (1994).
- [7] N. Farnsworth, “Thorium-based mirrors in the Extreme Ultraviolet,” Brigham Young University, Honors Thesis (2005).
- [8] D. G. Stearns, “X-ray scattering from interfacial roughness in multilayer structures,” *Journal of Applied Physics* **71**, 4286–4298 (1992).

- [9] J. E. J. Johnson, “Computationally modeling the effects of surface roughness on soft X-ray multilayer reflectors,” Brigham Young University, Masters Thesis (2006).

- [10] R. S. Turley, “Notes on Matlab Curve Fitting,” Brigham Young University Internal Document (2007).

Appendix A

Matlab Codes

The following are the Matlab codes used to compute the $-\ln(R)$, plot the data, and do the fitting.

```
% fromE.m
%finding R from E
%lnR=zeros(16,5,5);
% sigma=zeros(16,5,5);

j=1;
for h=2:2:10
    i=1;
    for f=10:20:190
        prefix='Randsurf_';
        fname=[prefix num2str(h) '_' num2str(f)]
        load(fname)
        for nr=1:16
            for inpstate=1:10
```

```
%E=mean(Er,3);

F1(nr,inpstate)=max(abs(Er(350:650,nr,inpstate)))
    &./max(abs(Es(350:650,nr)));

R=F1.^2;

%theta=15:5:90;

end

lnR(nr,i,j)=mean(-log(R(nr,:))');
%lnR(nr,i,j)=-log(meanR(nr));
sigma(nr,i,j)=std(-log(R(nr,:))');
w(nr,i,j)=1./(sigma(nr,i,j)).^2;

end

i=i+1;

end

j=j+1;

end

% plots.m

%making plots

close all

%for rfreq=1

theta=15:5:90; q=2*pi*sind(theta);

plot(q*.02,lnR(:,1,1),'bo',q*.04,lnR(:,1,2),'ro',q*.06,lnR(:,1,3),'go',
    &q*.08,lnR(:,1,4),'yo',q*.1,lnR(:,1,5),'mo')

title('rfreq=1') xlabel('q*h') ylabel('-ln(R)')
```

```

lnR1=[lnR(:,1,1) lnR(:,1,2) lnR(:,1,3) lnR(:,1,4) lnR(:,1,5)];
lnR1=reshape(lnR1,1,16*5); q1=[q*.02 q*.04 q*.06 q*.08 q*.1];
q1=reshape(q1,1,16*5); sigma1=[sigma(:,1,1) sigma(:,1,2)
sigma(:,1,3) sigma(:,1,4) sigma(:,1,5)];
sigma1=reshape(sigma1,1,16*5); A1=[q1;lnR1;sigma1];
A1=sortrows(A1'); A1=A1'; lnR1=A1(2,:); q1=A1(1,:); sigma1=A1(3,:);

%for rfreq=3

figure theta=15:5:90; q=2*pi*sind(theta);
plot(q*.02,lnR(:,2,1),'bo',q*.04,lnR(:,2,2),'ro',q*.06,lnR(:,2,3),'go',
&q*.08,lnR(:,2,4),'yo',q*.1,lnR(:,2,5),'mo')
title('rfreq=3') xlabel('q*h') ylabel('-ln(R)')

lnR3=[lnR(:,2,1) lnR(:,2,2) lnR(:,2,3) lnR(:,2,4) lnR(:,2,5)];
lnR3=reshape(lnR3,1,16*5); q3=[q*.02 q*.04 q*.06 q*.08 q*.1];
q31=reshape(q3,1,16*5); sigma3=[sigma(:,2,1) sigma(:,2,2)
sigma(:,2,3) sigma(:,2,4) sigma(:,2,5)];
sigma3=reshape(sigma3,1,16*5); A3=[q3;lnR3;sigma3];
A3=sortrows(A3'); A3=A3'; lnR3=A3(2,:); q3=A3(1,:); sigma3=A3(3,:);

%for rfreq=5

```

```

figure theta=15:5:90; q=2*pi*sind(theta);
plot(q*.02,lnR(:,3,1),'bo',q*.04,lnR(:,3,2),'ro',q*.06,lnR(:,3,3),'go',
      &q*.08,lnR(:,3,4),'yo',q*.1,lnR(:,3,5),'mo')
title('rfreq=5') xlabel('q*h') ylabel('-ln(R)')

```

```

lnR5=[lnR(:,3,1) lnR(:,3,2) lnR(:,3,3) lnR(:,3,4) lnR(:,3,5)];
lnR5=reshape(lnR5,1,16*5); q5=[q*.02 q*.04 q*.06 q*.08 q*.1];
q5=reshape(q5,1,16*5); sigma5=[sigma(:,3,1) sigma(:,3,2)
sigma(:,3,3) sigma(:,3,4) sigma(:,3,5)];
sigma5=reshape(sigma5,1,16*5); A5=[q5;lnR5;sigma5];
A5=sortrows(A5'); A5=A5'; lnR5=A5(2,:); q5=A5(1,:); sigma5=A5(3,:);

```

```

%for rfreq=7

```

```

figure theta=15:5:90; q=2*pi*sind(theta);
plot(q*.02,lnR(:,4,1),'bo',q*.04,lnR(:,4,2),'ro',q*.06,lnR(:,4,3),'go',
      &q*.08,lnR(:,4,4),'yo',q*.1,lnR(:,4,5),'mo')
title('rfreq=7') xlabel('q*h') ylabel('-ln(R)')

```

```

lnR7=[lnR(:,4,1) lnR(:,4,2) lnR(:,4,3) lnR(:,4,4) lnR(:,4,5)];
lnR7=reshape(lnR7,1,16*5); q7=[q*.02 q*.04 q*.06 q*.08 q*.1];
q7=reshape(q7,1,16*5); sigma7=[sigma(:,4,1) sigma(:,4,2)
sigma(:,4,3) sigma(:,4,4) sigma(:,4,5)];
sigma7=reshape(sigma7,1,16*5); A7=[q7;lnR7;sigma7];

```

```
A7=sortrows(A7'); A7=A7'; lnR7=A7(2,:); q7=A7(1,:); sigma7=A7(3,:);
```

```
%for rfreq=9
```

```
figure theta=15:5:90; q=2*pi*sind(theta);
plot(q*.02,lnR(:,5,1),'bo',q*.04,lnR(:,5,2),'ro',q*.06,lnR(:,5,3),'go',
      &q*.08,lnR(:,5,4),'yo',q*.1,lnR(:,5,5),'mo')
title('rfreq=9') xlabel('q*h') ylabel('-ln(R)')
```

```
lnR9=[lnR(:,5,1) lnR(:,5,2) lnR(:,5,3) lnR(:,5,4) lnR(:,5,5)];
lnR9=reshape(lnR9,1,16*5); q9=[q*.02 q*.04 q*.06 q*.08 q*.1];
q9=reshape(q9,1,16*5); sigma9=[sigma(:,5,1) sigma(:,5,2)
sigma(:,5,3) sigma(:,5,4) sigma(:,5,5)];
sigma9=reshape(sigma9,1,16*5); A9=[q9;lnR9;sigma9];
A9=sortrows(A9'); A9=A9'; lnR9=A9(2,:); q9=A9(1,:); sigma9=A9(3,:);
```

```
%for rfreq=11
```

```
figure theta=15:5:90; q=2*pi*sind(theta);
plot(q*.02,lnR(:,6,1),'bo',q*.04,lnR(:,6,2),'ro',q*.06,lnR(:,6,3),'go',
      &q*.08,lnR(:,6,4),'yo',q*.1,lnR(:,6,5),'mo')
title('rfreq=11') xlabel('q*h') ylabel('-ln(R)')
```

```
lnR11=[lnR(:,6,1) lnR(:,6,2) lnR(:,6,3) lnR(:,6,4) lnR(:,6,5)];
```

```

lnR11=reshape(lnR11,1,16*5); q11=[q*.02 q*.04 q*.06 q*.08 q*.1];
q11=reshape(q11,1,16*5); sigma11=[sigma(:,6,1) sigma(:,6,2)
sigma(:,6,3) sigma(:,6,4) sigma(:,6,5)];
sigma11=reshape(sigma11,1,16*5); A11=[q11;lnR11;sigma11];
A11=sortrows(A11'); A11=A11'; lnR11=A11(2,:); q11=A11(1,:);
sigma11=A11(3,:);

```

```

%for rfreq=13

```

```

figure theta=15:5:90; q=2*pi*sind(theta);
plot(q*.02,lnR(:,7,1),'bo',q*.04,lnR(:,7,2),'ro',q*.06,lnR(:,7,3),'go',
&q*.08,lnR(:,7,4),'yo',q*.1,lnR(:,7,5),'mo')
title('rfreq=13') xlabel('q*h') ylabel('-ln(R)')

```

```

lnR13=[lnR(:,7,1) lnR(:,7,2) lnR(:,7,3) lnR(:,7,4) lnR(:,7,5)];
lnR13=reshape(lnR13,1,16*5); q13=[q*.02 q*.04 q*.06 q*.08 q*.1];
q13=reshape(q13,1,16*5); sigma13=[sigma(:,7,1) sigma(:,7,2)
sigma(:,7,3) sigma(:,7,4) sigma(:,7,5)];
sigma13=reshape(sigma13,1,16*5); A13=[q13;lnR13;sigma13];
A13=sortrows(A13'); A13=A13'; lnR13=A13(2,:); q13=A13(1,:);
sigma13=A13(3,:);

```

```

%for rfreq=15

```

```

figure theta=15:5:90; q=2*pi*sind(theta);
plot(q*.02,lnR(:,8,1),'bo',q*.04,lnR(:,8,2),'ro',q*.06,lnR(:,8,3),'go',
      &q*.08,lnR(:,8,4),'yo',q*.1,lnR(:,8,5),'mo')
title('rfreq=15') xlabel('q*h') ylabel('-ln(R)')

```

```

lnR15=[lnR(:,8,1) lnR(:,8,2) lnR(:,8,3) lnR(:,8,4) lnR(:,8,5)];
lnR15=reshape(lnR15,1,16*5); q15=[q*.02 q*.04 q*.06 q*.08 q*.1];
q15=reshape(q15,1,16*5); sigma15=[sigma(:,8,1) sigma(:,8,2)
sigma(:,8,3) sigma(:,8,4) sigma(:,8,5)];
sigma15=reshape(sigma15,1,16*5); A15=[q15;lnR15;sigma15];
A15=sortrows(A15'); A15=A15'; lnR15=A15(2,:); q15=A15(1,:);
sigma15=A15(3,:);

```

```

%for rfreq=17

```

```

figure theta=15:5:90; q=2*pi*sind(theta);
plot(q*.02,lnR(:,9,1),'bo',q*.04,lnR(:,9,2),'ro',q*.06,lnR(:,9,3),'go',
      &q*.08,lnR(:,9,4),'yo',q*.1,lnR(:,9,5),'mo')
title('rfreq=17') xlabel('q*h') ylabel('-ln(R)')

```

```

lnR17=[lnR(:,9,1) lnR(:,9,2) lnR(:,9,3) lnR(:,9,4) lnR(:,9,5)];
lnR17=reshape(lnR17,1,16*5); q17=[q*.02 q*.04 q*.06 q*.08 q*.1];
q17=reshape(q17,1,16*5); sigma17=[sigma(:,9,1) sigma(:,9,2)
sigma(:,9,3) sigma(:,9,4) sigma(:,9,5)];
sigma17=reshape(sigma17,1,16*5); A17=[q17;lnR17;sigma17];

```

```

A17=sortrows(A17'); A17=A17'; lnR17=A17(2,:); q17=A17(1,:);
sigma17=A17(3,:);

%for rfreq=19

figure theta=15:5:90; q=2*pi*sind(theta);
plot(q*.02,lnR(:,10,1),'bo',q*.04,lnR(:,10,2),'ro',q*.06,lnR(:,10,3),
      &'go',q*.08,lnR(:,10,4),'yo',q*.1,lnR(:,10,5),'mo')
title('rfreq=19') xlabel('q*h') ylabel('-ln(R)')

lnR19=[lnR(:,10,1) lnR(:,10,2) lnR(:,10,3) lnR(:,10,4) lnR(:,10,5)];
lnR19=reshape(lnR19,1,16*5); q19=[q*.02 q*.04 q*.06 q*.08 q*.1];
q19=reshape(q19,1,16*5); sigma19=[sigma(:,10,1) sigma(:,10,2)
sigma(:,10,3) sigma(:,10,4) sigma(:,10,5)];
sigma19=reshape(sigma19,1,16*5); A19=[q19;lnR19;sigma19];
A19=sortrows(A19'); A19=A19'; lnR19=A19(2,:); q19=A19(1,:);
sigma19=A19(3,:);

M=[q1;q3;q5;q7;q9;q11;q13;q15;q17;q19];
N=[lnR1;lnR3;lnR5;lnR7;lnR9;lnR11;lnR13;lnR15;lnR17;lnR19];
L=[sigma1;sigma3;sigma5;sigma7;sigma9;sigma11;sigma13;sigma15;sigma17;sigma19];

% Fit the data
p1=zeros(10,1); p2=zeros(10,1); p3=zeros(10,1); bounds=zeros(5,2,4);
for x=1:10;

```

```
options=fitoptions('poly3');
options.Lower=[-Inf -Inf -Inf 0];
options.Upper=[Inf Inf Inf 0];
options.Weights=1./(L(x,:)).^2;
sfit=fit(M(x,:),N(x,),'poly3',options);
p1(x)=sfit.p1;
p2(x)=sfit.p2;
p3(x)=sfit.p3;
bounds(x,,:) = confint(sfit);
end

% Plot the fitted data
for x=1:10;
    figure
    errorbar(M(x,:),N(x,:),L(x,),'b*')
    hold on
    t=p1(x).*M(x,:).^3+p2(x).*M(x,:).^2+p3(x).*M(x,:);
    plot(M(x,:),t,'r')
    hold on
    DW=2.*M(x,:).^2;
    plot(M(x,:),DW,'k-')
    xlabel('q*h')
    ylabel('-ln(R)')
    title('Fitted Reflectance')
end
```



```
% Compute the sigmas from the fits
% Factor of 4.2 is 2 (for half the difference) and 2.1
% to relate the 95% confidence interval from the Student
% T distribution to sigma
e1=(bounds(:,2,1)-bounds(:,1,1))/4.2;
e2=(bounds(:,2,2)-bounds(:,1,2))/4.2;
e3=(bounds(:,2,3)-bounds(:,1,3))/4.2; e=[e1 e2 e3];

rfreq=1:2:19;

figure

    options=fitoptions('poly2');
    options.Lower=[-Inf -Inf -Inf];
    options.Upper=[Inf Inf Inf];
    options.Weights=1./e1.^2;
[f1,g1]=fit(rfreq',p1,'poly2',options);
pred1=f1.p1*rfreq.^2+f1.p2*rfreq+f1.p3; errorbar(rfreq,p1,e1,'b*')
hold on plot(rfreq,pred1,'r-') xlabel('roughness frequency in
wavelengths') ylabel('x^3 coefficient') title('Fit of x^3
coefficient') bounds1=confint(f1);
errorp1=(bounds1(2,1)-bounds1(1,1))/4.2;
errorq1=(bounds1(2,2)-bounds1(1,2))/4.2;
errorr1=(bounds1(2,3)-bounds1(1,3))/4.2;

figure
```

```
options=fitoptions('poly1');
options.Lower=[-Inf -Inf];
options.Upper=[Inf Inf];
options.Weights=1./e2.^2;
[f2,g2]=fit(rfreq',p2,'poly1',options); pred2=f2.p1*rfreq+f2.p2;
errorbar(rfreq,p2,e2,'b*') hold on plot(rfreq,pred2,'r-')
xlabel('roughness frequency in wavelengths') ylabel('x^2
coefficient') title('Fit of x^2 coefficient') bounds2=confint(f2);
errorp2=(bounds2(2,1)-bounds1(1,1))/4.2;
errorq2=(bounds2(2,2)-bounds1(1,2))/4.2;
```

figure

```
options=fitoptions('poly2');
options.Lower=[-Inf -Inf -Inf];
options.Upper=[Inf Inf Inf];
options.Weights=1./e3.^2;
[f3,g3]=fit(rfreq',p3,'poly2',options);
pred3=f3.p1*rfreq.^2+f3.p2*rfreq+f3.p3; errorbar(rfreq,p3,e3,'b*')
hold on plot(rfreq,pred3,'r-') xlabel('roughness frequency in
wavelengths') ylabel('x coefficient') title('Fit of x coefficient')
bounds3=confint(f3); errorp3=(bounds3(2,1)-bounds1(1,1))/4.2;
errorq3=(bounds3(2,2)-bounds1(1,2))/4.2;
errorr3=(bounds3(2,3)-bounds1(1,3))/4.2;
```

Index

astronomy, 1

Debye-Waller, 6, 8, 14, 29, 30

EUV, 1, 2

Fresnel, 11

Gaussian, 8

index of refraction, 2–4, 14

multilayer, 2, 3, 30

Nevot-Croce, 6, 8

non-specular scattering, 4

patches, 11

photolithography, 1

roughness, 4, 7, 8, 11, 14, 29, 30

standard deviation, 14