IMD

Version 5.0

Installation Guide & User's Manual

27 November 2013

David L. Windt

davidwindt@gmail.com

Reflective X-ray Optics

Preface

This document is the place to go for help with IMD. It is meant to be a combination Installation Manual (Chapter 1), Quick-Start Guide (Chapter 2) and Tutorial/Reference Manual (Chapter 3 and everything that follows), with a number of examples presented along the way to illustrate IMD's capabilities.

If you are new to IMD I encourage you to read this entire document, or at least those sections on the topics that interest you. In fact, even if you've used previous versions of IMD and already know what you're doing, you might consider reading the whole thing anyway, as this version of IMD contains a number of new features and capabilities, some of which are complex and non-intuitive, such as aperiodic multilayers, genetic algorithms for fitting and optimization, IMD*Multiplot, and so forth.

I'd like to think that every IMD user will take my suggestion and read this document from beginning to end … but of course I realize that's completely unrealistic. Who wants to read (200+ page) software manuals?!? Not me! Nevertheless (and in all seriousness), my recommendation is that, at the very least, everyone should read Chapters 1 and 2, to get IMD installed and running properly. After that, if you're eager to jump right in and start using IMD, and you just don't want to read this document cover to cover, then by all means please proceed. You'll find a number of example files in the '…/`imd/examples`' directory – these correspond to the examples presented in this document (plus some extras), but you can certainly explore them on your own to see how IMD works for yourself. And if you get lost, this document will be here for you. Also, please check out the IMD & TOPO Google group (https://groups.google.com/forum/#!forum/imd_topo) for discussion, and announcements of future releases.

A note about notation

Throughout this manual computer file names are always written in `this font`. When files are written with their full path (or sometimes even with a partial path, as in the very next sentence), I'll use notation applicable to unix (i.e., linux, Mac OS X, and Solaris) operating systems, using a forward slash (/) to delineate subdirectory names. For example: "…`/imd/examples/EUV Telescope`", which means the `EUV Telescope` subdirectory in the `examples` subdirectory in the `imd` installation directory; the leading "…/" means "wherever the `imd` installation directory is installed on your computer". Of course Microsoft Windows platforms use the backward slash (\), as in "…`\imd\examples\EUV Telescope`", so if you're using a Windows computer, please substitute backward slashes for forward slashes when you see a file name with a forward slash in this document.

Acknowledgements

This version of IMD represents more than a decade's worth of improvements and enhancements since the last released version (4.1). The development of this version of IMD has been made possible in large part by continual support from NASA, since 1999, for high-energy astrophysics technology development, most recently under NASA grant number NNX13AC54G. To all the NASA people involved, and the proposal review panels over the years: I can't thank you enough for your support! I am also indebted to all the IMD users who have provided to me valuable feedback, suggestions and optical constants. Many of IMD's new capabilities are the result of your input, and I'm extremely grateful for that. And finally, thanks to all the people who did the hard work of developing the theories and algorithms on which IMD is based, and to those people who've measured the optical constants and atomic scattering factors that are essential to IMD's operation.

IMD

Version 5.0

Installation Guide & User's Manual

Contents

1 Installing IMD 9
1.1 System Requirements 9
1.1.1 Using IMD as an IDL application 9
1.1.2 Using IMD as an XOP extension 9
1.2 Installing IMD as an IDL application 10
1.3 Installing IMD as an XOP extension 13
1.4 Customizing the IMD installation: `imd_cfg.pro` 15
1.4.1 Non-standard installation location 15
1.4.2 Other customizable settings 15
1.5 Common Installation Problems and Solutions 16
2 Using IMD 17
2.1 Overview: What is IMD and what can it do? 17
2.2 Starting IMD 19
2.3 Some (Relatively) Simple Examples, Before We Get Into All The Details 20
2.3.1 Example: Transmittance of a Zr Filter 21
2.3.2 Example: Reflectance of a Si/Mo Multilayer 24
2.3.3 Example: Displaying Filter Transmittance and Multilayer Reflectance in the Same Plot 27
2.3.4 Example: Computing and Displaying Net Efficiency of an Instrument 30
3 Specifying Materials: Optical Constants & Atomic Scattering Factors 33
3.1 The IMD Optical Constants Database 33
3.2 User-Contributed Optical Constants 36
3.3 Atomic Scattering Factors 37

3.4 User-Contributed Atomic Scattering Factors 38
3.5 Neutrons 39
4 Defining the Film Stack 40
4.1 Ambient 41
4.2 Substrate 42
4.3 Layers 43
4.4 Periodic Multilayers 44
4.5 Depth-Graded Multilayers 45
4.6 Aperiodic Multilayers 47
4.6.1 Methods for Populating an Aperiodic Thickness List (Part 1/3): 48
4.7 Aperiodic Thickness List Files 49
4.7.1.1 Methods for Populating an Aperiodic Thickness List (Part 2/3): 49
4.8 Nested Multilayers 52
4.9 Layer-Data Files 52
4.10 Structure Diagram and Layer Thickness Profile Plot 54
4.11 Surface and Interface Imperfections 57
4.11.1 Modified Fresnel Coefficients 58
4.11.1.1 Interface Profile Functions 58
4.11.1.2 Fresnel Coefficient Modification Options 61
4.11.2 Power-Spectral-Density Functions 62
4.11.3 Using PSD Parameters to Determine Interface Widths 65
4.11.4 Depth-Graded PSD Parameters 66
4.11.4.1 Depth-Graded Correlation Length, $\xi_{\parallel}$ 66
4.11.4.2 Depth-Graded Interface Width, σ 68
4.11.5 Interface Width Profile Plot 69
4.11.6 Graded Interfaces 70
4.12 Coupled Parameters 75
4.12.1 Example: Depth-Graded Si/W Multilayer 75
4.12.2 Example: Graded-Γ Al/Zr Multilayer 81
5 Specifying Variables 84
5.1 Dependent Variables 84

5.1.1 Specular Optical Functions and Field Intensities 84
5.1.1.1 User-Defined Specular Optical Functions 84
5.1.2 Non-Specular Reflected Intensity 85
5.2 Independent Variables 86
5.2.1 Incidence Angle and Wavelength/Energy 86
5.2.2 Polarization Parameters 88
5.2.2.1 Computation of 'Average-Polarization' Optical Functions 89
5.2.3 Defining Other Parameters as Independent Variables 90
5.2.3.1 Example: X-ray Transmittance of Aluminum versus Thickness 90
5.2.4 Independent Variables Specific to Field Intensity Computations 93
5.2.5 Independent Variables Specific to Non-Specular Reflected Intensity Computations 94
5.2.5.1 Example: Non-Specular Scattering – Rocking Scan 96
5.2.5.2 Example: Non-Specular Scattering – Detector Scan 97
5.2.5.3 Example: Non-Specular Scattering – Offset Scan 98
5.2.5.4 Scattering Angle q-Space Diagrams 98
6 Visualizing Computations: IMDXPLOT 100
6.1 Example: Si/Mo Multilayer Reflectance vs. Γ 100
6.2 IMDXPLOT: Legends and Labels 103
6.3 IMDXPLOT: Independent Variables Tab 104
6.4 IMDXPLOT: Fixed Curves 106
6.5 IMDXPLOT: Axes Tab 107
6.6 IMDXPLOT: Dependent Variables Tab 108
6.6.1.1 IMDXPLOT: Statistics 109
6.6.1.2 IMDXPLOT: Regions-Of-Interest 110
6.7 Printing Graphics from IMDXPLOT 111
6.8 "What happens in IMDXPLOT stays in IMDXPLOT!" 112
6.9 Example: Si/Mo Multilayer Field Intensity 112
6.10 Example: Si/Mo ML Normal-Incidence Non-Specular Reflected Intensity 114
7 Saving and Exporting Computation Results 120
7.1 Saving to a .imd File 120
7.2 Saving to a .imd File from IMDXPLOT 120

7.3 Exporting Results to Plain Text Files 120
7.4 Exporting Results to IDL Save Files 121
8 Measured Data, Curve-Fitting and Optimization 123
8.1 Measured Data 123
8.1.1 Importing Measured Data Files: X_m, Y_m and SIGY_m 123
8.1.2 Importing Measured Data Files in IMDXPLOT 127
8.2 Modeling Measured Data 127
8.2.1 Example: Modeling XRR Measurements of a Mo/Y Multilayer 127
8.2.2 Example: Modeling Soft X-ray Reflectance Measurements of a C/CoCr Multilayer 128
8.3 Optimization Target Profiles: X_o, Y_o and W_o 130
8.3.1 Importing Optimization Target Profile Files 131
8.3.2 Building Optimization Target Profiles from Scratch 135
8.4 IMDXPLOT: Measured Data and Optimization Target Profile 137
8.4.1 IMDXPLOT: ROI's for Measured Data and Optimization Target Profile 137
8.5 Specifying Fitting and Optimization Parameters 137
8.5.1 Specifying Fitting Parameters 137
8.5.2 Specifying Optimization Parameters 141
8.6 Algorithms for Curve-Fitting and Optimization 144
8.6.1 χ^2 Minimization: Marquardt Algorithm (CURVEFIT) 146
8.6.2 χ^2 Minimization: Levenberg-Markquardt Algorithm (MPFIT/MINPACK-1) 148
8.6.3 Genetic Algorithms 149
8.6.4 FOM Minimization: Genetic Algorithm (Binda) 150
8.6.5 FOM Minimization: Differential Evolution Algorithm (Björck) 153
8.7 Performing the Fit or Optimization 157
8.7.1 Example: Fitting XRR Data for a B_4C/Mo Bilayer (Part 1) 158
8.7.2 The Fitting (or Optimization) Results Window 163
8.7.3 Example: Fitting XRR Data for a B_4C/Mo Bilayer (Part 2) 164
8.7.4 Example: Fitting XRR Data for a B_4C/Mo Bilayer (Part 3) 167
8.7.5 Example: Fitting XRR Data for a Mo/Y Double-Multilayer 170
8.7.6 Example: Optimizing an Aperiodic Multilayer to a Triangular Profile 176
8.7.6.1 Methods for Populating an Aperiodic Thickness List (Part 3/3): 177

8.7.7 Example: Optimizing an Aperiodic Multilayer for Flat Response at Normal Incidence 183
8.8 Parameter Estimation: Confidence Intervals 189
8.8.1 Example: Determining Optical Constants from R-θ Measurements 191
8.8.1.1 IMDXPLOT: Confidence Intervals 193
9 IMD*Multiplot and IMD*Efficiency 196
9.1 IMD*Efficiency 196
9.2 IMD*Multiplot 198
9.2.1 IMD*Multiplot: Using IMD Files Containing Measured Data 199
9.2.1.1 Example: IMD*Multiplot – Comparing the Performance of Three Multilayers 199
9.2.2 IMD*Multiplot: Visualizing Measured Data Sets 204
9.2.2.1 Example: IMD*Multiplot – Using Measured Data to Map Coating Uniformity 204
9.2.2.2 Example: IMD*Multiplot – Measured Performance over Time 215
9.2.2.3 IMD*Multiplot: Using "other" data 219
10 IMD Preferences 221
10.1 The IMD Preferences Window 221
10.2 Importing and Exporting IMD Preferences (.ipr) Files 222
10.3 Automatic IMD Preferences Import at Startup: IMD_PREFS_FILE 222
10.4 IMD and IMD Preferences Auto-Load Files: `imd_autoload.imd` and `imd_autoload.ipr` 223
11 IMD for IDL Programmers 224
11.1 IMD Common Block Variables 224
11.1.1 Dependent and Independent Variable Names 224
11.2 Computational Routines 225
11.2.1 FRESNEL 225
11.2.2 NS_BA 230
11.2.3 NS_DWBA 233
11.3 Optical Constants Routines 235
11.3.1 IMD_F1F2TONK 235
11.3.2 IMD_NK 237
11.4 Data-Importing Routines 237
11.4.1 EROM 238

11.4.2 RD_REFLECT 240
11.4.3 RD_ALS 241
11.4.4 RD_SPEC 242
12 Performance and Troubleshooting 246
12.1 Frequently Asked Questions 246
12.1.1 Q: I can't get IMD to work/start. 246
12.1.2 Q: I'm getting error messages about running out of memory. 246
12.1.3 Q: I'm getting reflectance (or whatever) values that can't be right. 247
12.1.4 Q: I can't get IMD to fit my data! / How do I get IMD to fit my data? 247
12.1.5 Q: The Marquardt or Levenberg-Marquardt algorithms seem to 'converge' way too soon, without finding a good solution to the fitting/optimization problem. 247
12.1.6 Q: Can I get a copy of the IMD source code? 247
12.1.7 Q: Please? I promise I won't show it to anyone 247
12.1.8 Q: Will/can you include feature X in a future release of IMD? 247
12.1.9 Q: Will IMD work with the free IDL Virtual Machine? 247
12.2 Reporting Bugs 247
13 Version History 249
14 References 250

1 Installing IMD

1.1 System Requirements

IMD is written in the IDL scientific programming language, and will run on any platform supported by IDL. Little or no IDL expertise is required to use IMD.

There are two ways to run IMD: (1) using a licensed copy of IDL, and (2) using the free XOP package, which includes an embedded, licensed copy of IDL.

1.1.1 Using IMD as an IDL application

IMD can be run using IDL version 6.3 or higher. IDL is available for purchase from Exelis Visual Information Solutions:

http://www.exelisvis.com/ProductsServices/IDL.aspx

NOTE: The full version of IDL is required to run IMD.

IMD will NOT run using the free IDL Virtual Machine.

1.1.2 Using IMD as an XOP extension

IMD can be run as an 'extension' to the free XOP software package, available at ESRF:

http://www.esrf.eu/Instrumentation/software/data-analysis/xop2.3

1.2 Installing IMD as an IDL application

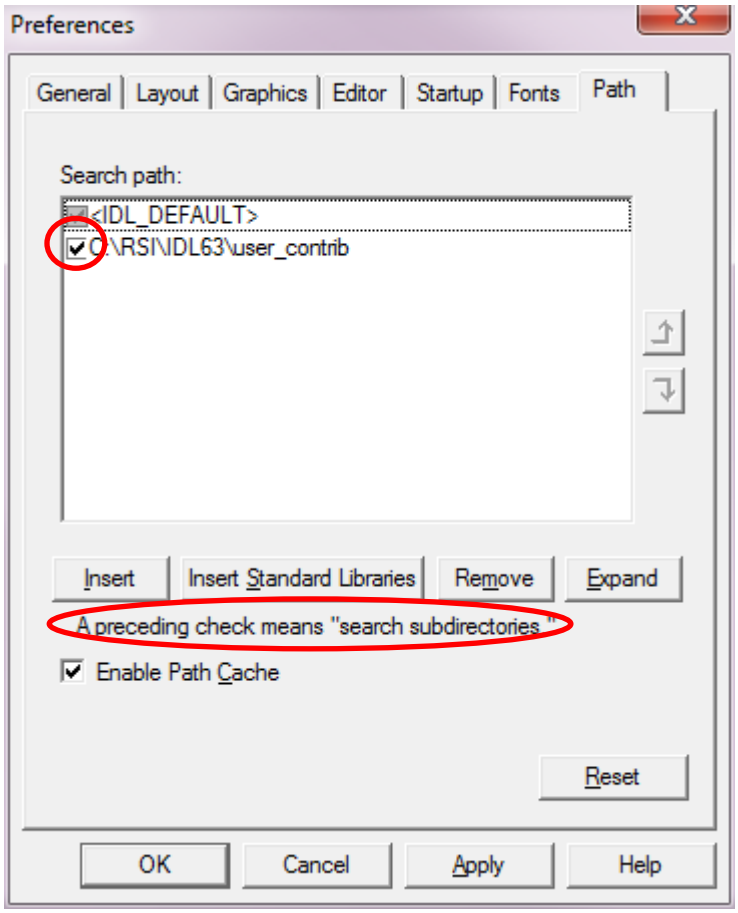
If you will use IMD as an XOP extension, then you can skip this section and proceed to §1.3.

Once you have IDL installed and running on your computer, follow these steps to install and run IMD:

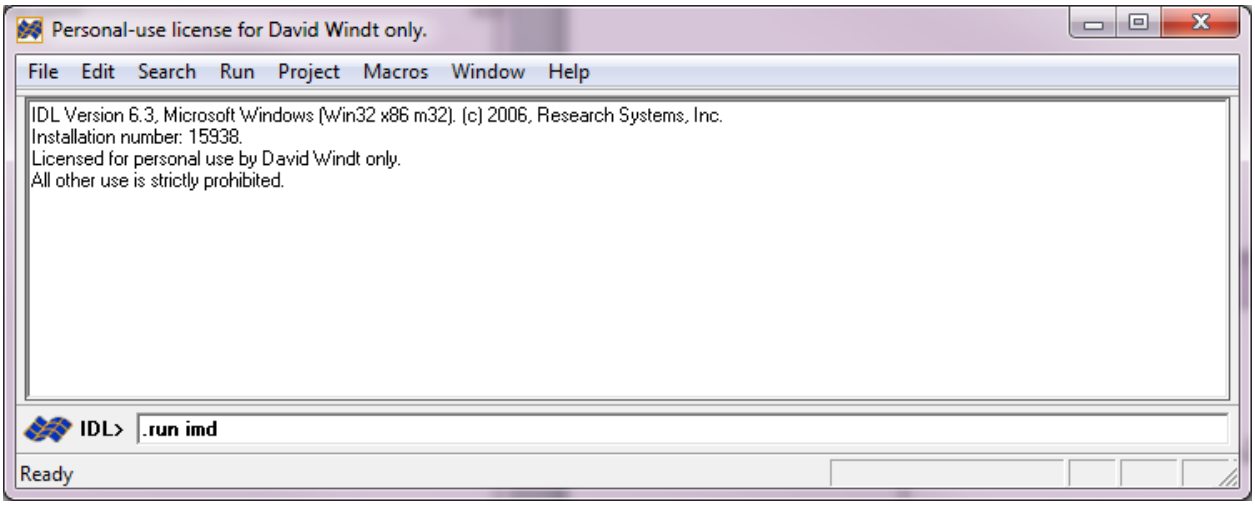
1. Create a directory (folder) called `user_contrib` in the main IDL installation directory.

   NOTE: You can install IMD in a different directory if you so choose. But if you do not install IMD in the `user_contrib` directory as described here, then you **must** edit the `imd_cfg.pro` configuration file to specify the actual installation location. See §1.4.1 for further details.

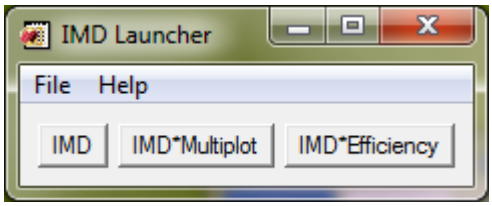
2. Download the IMD installation file `imd_xxx.tar.gz` from http://www.rxollc.com/idl/ (`xxx`=version number; for example, `imd_5.02.tar.gz`)
3. Unpack the `imd_xxx.tar.gz` file containing the `imd` installation directory.
   - For Windows: use a program like Winzip to open `imd_xxx.tar.gz`
   - For Linux/Mac OS X/Solaris: use the gunzip command:
     `$ gunzip imd_xxx.tar.gz`
     followed by the tar command:
     `$ tar –xvf imd_xxx.tar`
4. Move the `imd` installation directory to the `user_contrib` directory created in step 1. (i.e., to …`/idl/user_contrib/imd`)
5. Start IDL and add the new `user_contrib` directory just created in step 1 to your IDL search path. Consult the IDL documentation for help configuring the IDL_PATH environment variable. If setting IDL_PATH manually, be sure to include a "+" sign so that the subdirectories under `user_contrib` will be included (e.g., IDL_PATH=+/usr/local/rsi/idl/user_contrib:…) If you're using IDL's Preferences GUI to set the search path, be sure to check the 'search subdirectories' box:



6. After your IDL search path has been redefined in step 5, type `.run imd` at the IDL prompt to start IMD:



7. The IMD splash-screen will appear. Click 'OK' to start both IMD and IMD Launcher. From IMD Launcher you can re-run IMD, and run IMD*Multiplot and IMD*Efficiency as well:



8. You're up and running!

Now please read the rest of IMD.pdf (through Chapter 2, at the very least) so that you get the most out of IMD!

Also, check out the IMD & TOPO Google group (https://groups.google.com/forum/#!forum/imd_topo) for discussion, and announcements of future releases.

1.3 Installing IMD as an XOP extension

If you will NOT use IMD as an XOP extension, then you can skip this section and proceed to §1.4.

Once you have XOP installed and running on your computer, follow these steps to install and run IMD:

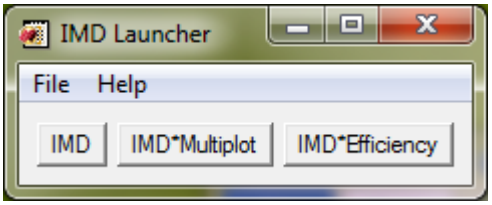
1. Download the IMD installation file `imd_xxx.tar.gz` from http://www.rxollc.com/idl/ (`xxx`=version number; for example, `imd_5.02.tar.gz`)
2. Unpack the `imd_xxx.tar.gz` file containing the `imd` installation directory.
   - For Windows: use a program like Winzip to open `imd_xxx.tar.gz`
   - For Linux/Mac OS X/Solaris: use the gunzip command:
     `$ gunzip imd_xxx.tar.gz`
     followed by the tar command:
     `$ tar –xvf imd_xxx.tar`
3. Move the `imd` installation directory to the `extensions` directory in the XOP installation directory (e.g., to …`/xop2.3/extensions/imd`).
4. *For Windows only:*
   - copy the file `C:\xop2.3\extensions\imd\extras\imd4xop.bat`
     to `C:\xop2.3\imd4xop.bat`
     and
   - copy the file `C:\xop2.3\extensions\imd\extras\imd4xop.vbs`
     to `C:\xop2.3\imd4xop.vbs`

   The name of the XOP installation directory depends on the version of XOP that you have installed, and on where you have installed it on your computer. For this step we've assumed that XOP V2.3 is installed in the default location `C:\xop2.3`. *However, if you have a different version of XOP, or a different installation location, please substitute the correct name and path for your XOP installation directory above.*

   ***Note**: if you don't use the default installation location (*`C:\xop2.3`*), you MUST edit* `imd4xop.bat` *and specify the correct path to* `xop.bat`.

   You might also want to make a shortcut on your Desktop to one or both of these files, so you can start IMD by double-clicking that shortcut. Use `imd4xop.vbs` *if it works on your system, otherwise use* `imd4xop.bat`.
5. Start IMD:
   - For Windows: Double-click on `imd4xop.vbs` (or `imd4xop.bat` if necessary.) Or, start XOP and then use the XOP→Load extension… menu option to select imd4xop.
   - For Linux/Mac OS X/Solaris: At the command line, type: `xop imd4xop`

6. The IMD splash-screen will appear. Click 'OK' to start both IMD and IMD Launcher. From IMD Launcher you can re-run IMD, and run IMD*Multiplot and IMD*Efficiency as well.



7. You're up and running!

Now please read the rest of IMD.pdf (through Chapter 2, at the very least) so that you get the most out of IMD!

Also, check out the IMD & TOPO Google group (https://groups.google.com/forum/#!forum/imd_topo) for discussion, and announcements of future releases.

1.4 Customizing the IMD installation: `imd_cfg.pro`

1.4.1 Non-standard installation location

If you will use IMD with a licensed copy of IDL, and you choose not to install the `imd` installation directory in the default location, i.e., the `user_contrib` directory in the main IDL directory as described in §1.2, then you **must** edit the IMD configuration file `imd_cfg.pro`, which is located in the `imd` installation directory you have selected: specifically, using any plain-text editor, change the value of the `imd_home` variable defined in this file so that it refers to the actual location where you've installed the `imd` directory on your computer.

EXAMPLE: You are using a linux system and you have installed IMD in your home directory, e.g., `/users/joe/imd.`

Edit `imd_cfg.pro` and replace the line

```
if (!version.os_family eq 'unix') then imd_home=!dir+'/user_contrib/imd/'
```

with

```
if (!version.os_family eq 'unix') then imd_home='/users/joe/imd/'
```

1.4.2 Other customizable settings

There are several other operating-system- and site-specific settings that you can modify in the `imd_cfg.pro` configuration file, including those that control the fonts and special characters used by the IMD graphical user interface (GUI), default atomic scattering factors and optical constants directories, and allocation of memory when computing large arrays, to avoid IDL 'out of memory' errors. Please follow the instructions contained within the `imd_cfg.pro` file to make such modifications.

If you are using IMD on a unix platform, you can also use the `Xdefaults` configuration file included in the `imd/extras` directory to set some IMD-specific X-windows preferences. One method to do this is to copy the contents of `imd/extras/Xdefaults` into your personal .Xdefaults file, or into the IDL Xdefaults file, `!dir/resource/X11/lib/X11/app-defaults/Idl`, where `!dir` is the IDL installation directory on your computer.

1.5 Common Installation Problems and Solutions

Problem: IDL can't find IMD:

```
IDL> .run imd
Error opening file. File: imd
```

Solution: You must add the IMD installation directory to your IDL search path, as described in §1.2.

Problem: You see errors like these:

"There are no valid optical constants files…"
"There specified atomic scattering factors directory – xxxx – does not contain any '.ff' files…"
"imd_cfg.pro" is invalid. Please correct the contents of this file."

Solution: You must edit the `imd_cfg.pro` configuration file, as described in §1.4. [i]

Problem: The IMD windows don't look right, and/or don't look much like the examples in this document.

Solution: The fonts used by IMD, which are specified in the `imd_cfg.pro` configuration file, might not be available on your computer. Edit the `imd_cfg.pro` file, as described in §1.4, and specify valid fonts that are actually installed on your computer.[i]

Problem: Special characters, such as 'Å', 'μ', 'δ', etc., don't display correctly in the IMD windows.

Solution: The special character codes for your particular installation, which are specified in the `imd_cfg.pro` configuration file, must be the correct codes for the fonts used by IMD, which are also specified in that file. Edit the `imd_cfg.pro` file, as described in §1.4, and specify valid special character codes for the fonts that you have specified.[i]

[i] If your operating system is not specifically listed in the `imd_cfg.pro` configuration file, you'll need to manually add the necessary entries for optical constants and atomic scattering factors directories, fonts, etc, using as a template the lines in that file that contain "YOUR OS": replace "YOUR OS" with the value of the IDL system variable !version.os as defined on your computer.

2 Using IMD

2.1 Overview: What is IMD and what can it do?

IMD is a user-friendly software package that can be used to calculate both specular and non-specular (diffuse) optical functions (e.g., reflectance, transmittance, scattering, etc.) of an arbitrary multilayer structure, i.e., a film stack comprising any number of layers, of any thickness, and of any material.[1] IMD can be used for modeling, instrument design, and data analysis. With IMD it's easy to model the optical performance of complex film stacks, such as periodic multilayers, depth-graded multilayers (where the layer thicknesses are defined analytically), and aperiodic multilayer structures (where the layer thicknesses are defined numerically). The effects of surface and interface imperfections (e.g., roughness) can be modeled using a variety of methods. Optical functions can be computed as a function of wavelength and/or incidence angle, as well as a function of almost any of the parameters defining the film stack, the interface and surface imperfections within the film stack and at the substrate, and the polarization properties of the incident beam and detector. Parameter estimation (including confidence-interval generation) from fits to measured data, as well as layer-thickness optimization to a target optical profile that you specify, can be performed using any of four algorithms: two gradient-expansion algorithms, and two genetic algorithms.

Included with IMD is a database of optical constants for over 150 materials, spanning the X-ray region to the infrared. It's easy to add your own optical constants as well, if necessary, or to create new optical constants in the EUV/X-ray region for any compound, using atomic scattering factors for 92 elements that have been compiled from databases made available by the Center for X-ray Optics (CXRO) and the Lawrence Livermore National Laboratory (LLNL).[2,3] User-contributed optical constants for a variety of additional materials are included in the IMD distribution as well, along with (unpolarized) neutron atomic scattering factors[4] and neutron optical constants that can be used to model the performance of thin-film coatings for neutron optics.

Specular optical functions – reflectance, transmittance, absorbtance, field intensities, phase shifts, and ellipsometric psi and delta functions – are computed in IMD using an algorithm based on recursive application of the Fresnel equations,[5] modified to approximate the effects of interfacial roughness and/or diffuseness.[6,7] Non-specular reflected intensities can be computed using either a dynamical Born approximation vector theory,[8,9] or the so-called 'Distorted-Wave Born Approximation' formalism,[10,11,12,13] a scalar theory which is nonetheless valid below the critical angle of total external reflection in the X-ray region. For both specular and non-specular computations, a stochastic model of film growth and erosion[14] can be used to account for the evolution of interfacial roughness through the film stack. Alternatively, a more conventional roughness model[15] can be used, with the option of defining depth-graded roughness and/or depth-graded correlation length parameters.

A variety of visualization tools are included with IMD. The IMDXPLOT application enables visualization of 1D or 2D 'slices' through multi-dimensional optical functions computed with IMD. With IMDXPLOT you can interactively vary any of the independent variables used in the computation to see the resulting effect on the optical functions in real time. IMDXPLOT allows you to easily overlay multiple optical functions (resulting from the same computation) on a single plot, and to include a variety of labels and legends. You can also overlay your own measured data in order to compare interactively your measurements to the calculations.

IMD also includes two additional visualization tools: IMD*Multiplot, which is a versatile, multi-purpose application that can be used to compare and analyze any number of separate IMD computations of specular optical functions and/or measured data sets, and IMD*Efficiency, which can be used to model the net throughput of an optical instrument comprising any number of optical surfaces in series, whose specular optical properties have been modeled in IMD.

2.2 Starting IMD

Follow the instructions in Chapter 1 to install IMD on your computer. To start IMD, follow these steps, as appropriate for your installation:

Operating System	IDL or XOP	Starting IMD

Any	IDL	*Start IDL and then type*
		`IDL> .run imd`
		Once you have executed '.run imd' once, you can open IMD again during the same IDL session by typing 'imd' at the prompt:
		`IMD> imd`
		You can also re-start IMD Launcher by typing 'imd_launcher' at the prompt:
		`IMD> imd_launcher`

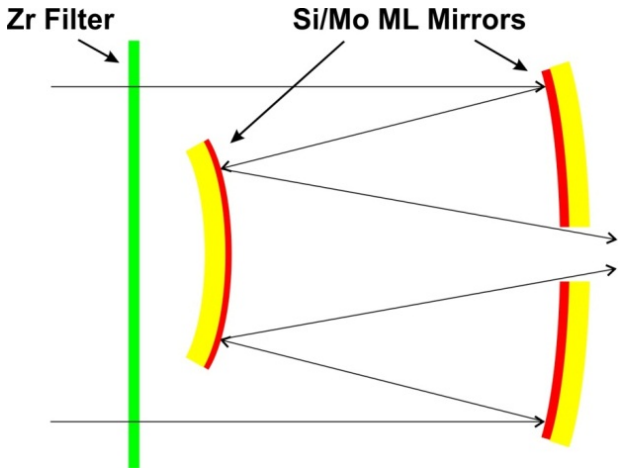
Linux/Mac OS X/ Solaris	XOP	*At the command line, type*
		`$ xop imd4xop`
		
		Or, type
		`$ xop`
		and then
		XOP→Load extension…
		and then select imd4xop *from the list of installed extensions, and then click* 'Accept'.

Windows	XOP	*Double-click*[ii] `C:\xop2.3\imd4xop.vbs` *If that doesn't work* Double-click `C:\xop2.3\imd4xop.bat` *If neither method works, edit* `imd4xop.bat` *and check that the specified path to* `xop.bat` *is correct for your installation.* *Or, start XOP (as per the XOP instructions) and then* XOP→Load extension… *and then select* imd4xop *from the list of installed extensions, and then click* 'Accept'.

[ii] Replace `C:\xop2.3` with your actual XOP installation directory.

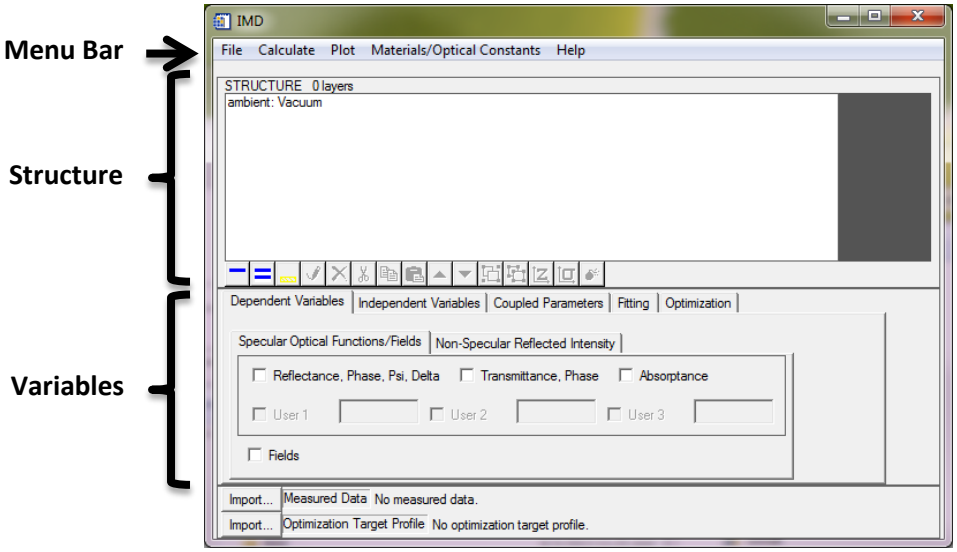
2.3 Some (Relatively) Simple Examples, Before We Get Into All The Details

Starting in Chapter 3 we'll see in great detail how IMD works and how to use it. But before getting into all that, let's just dive right in with some simple examples that illustrate some of IMD's capabilities for modeling specular optical functions. Specifically, we'll model the performance of an EUV telescope comprising a thin Zr entrance filter, and a two-reflection telescope who's mirrors are coated with periodic Si/Mo multilayers tuned to λ=130Å at normal incidence. The optical system looks like this:



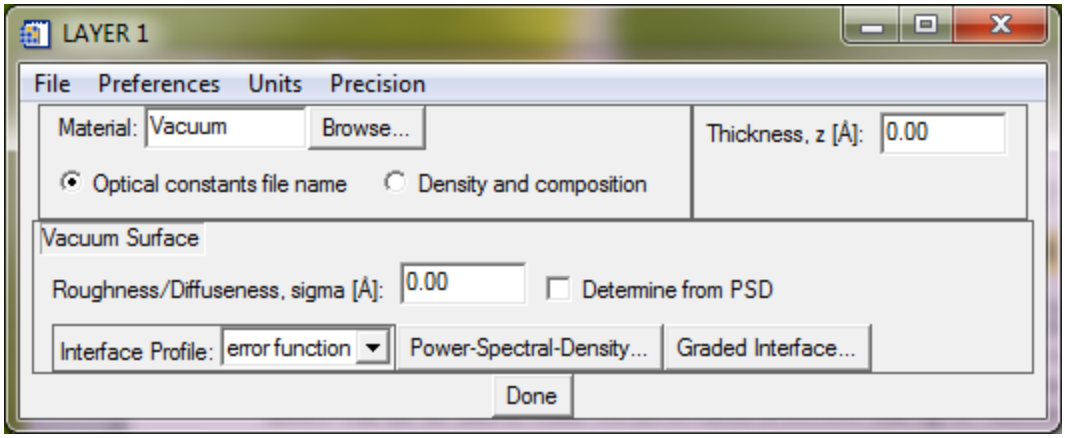
We'll first model the performance of each component – filter and multilayer mirror coating – and then we'll compute the efficiency of the whole instrument.

To begin, let's have a look at the main IMD window as it appears at startup, as shown in the next figure. The Menu Bar, the Structure area, and the Variables area are labeled. We'll refer to these regions of the main IMD window as we proceed with our first examples in this chapter, and throughout the rest of this document as well.

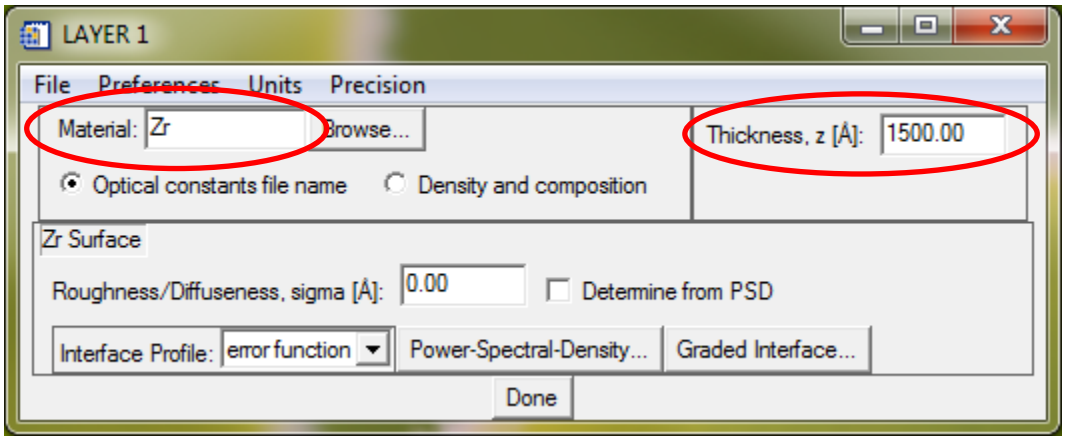


2.3.1 Example: Transmittance of a Zr Filter

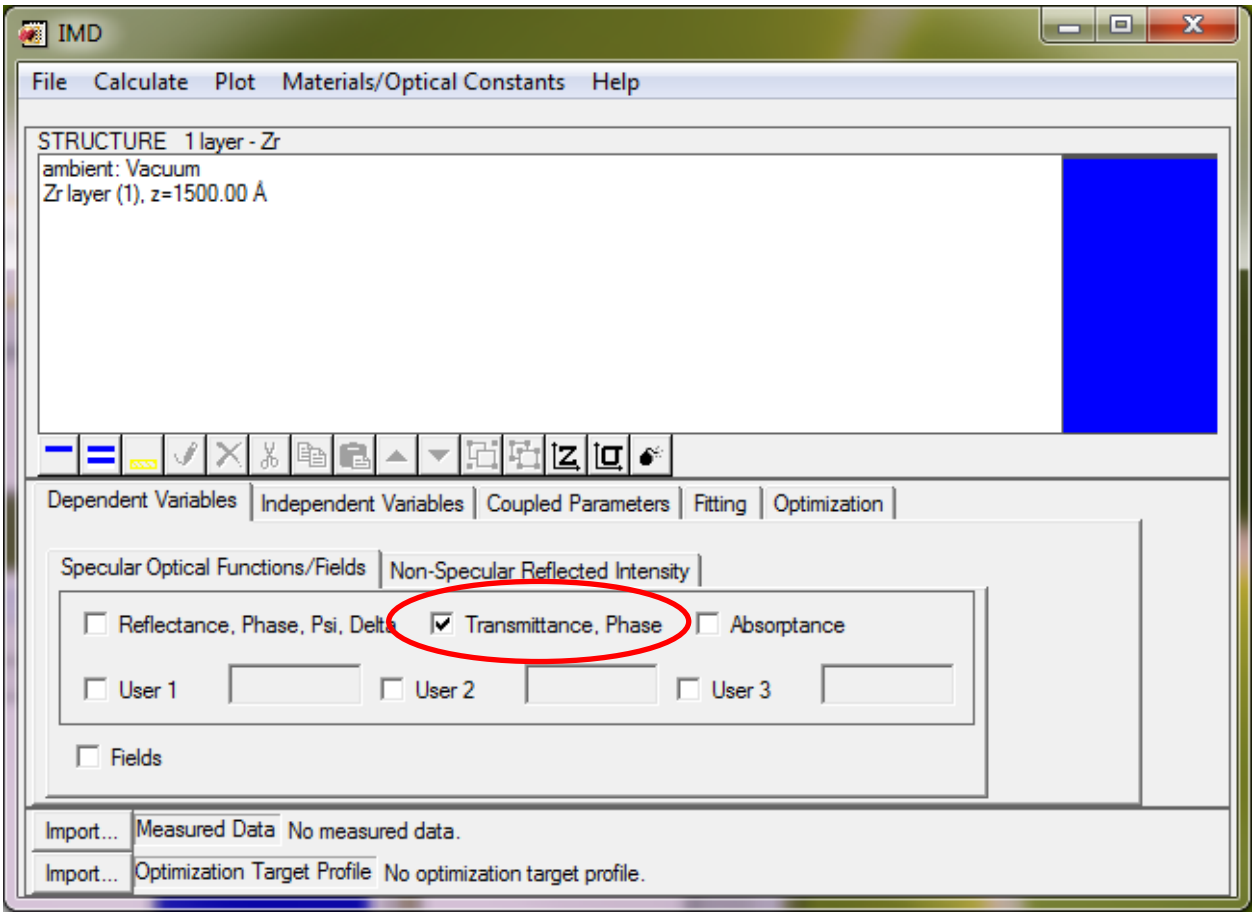
The first example will show how to compute the transmittance of a 1500-Å-thick Zr filter, from 100 Å to 150 Å wavelength, at normal incidence. We start by adding a "layer", representing the free-standing filter, by clicking on the "Add Layer" button at the bottom of the Structure area (i.e., the left-most button, with the horizontal blue bar "layer" icon that looks like this:). The corresponding Layer window appears:



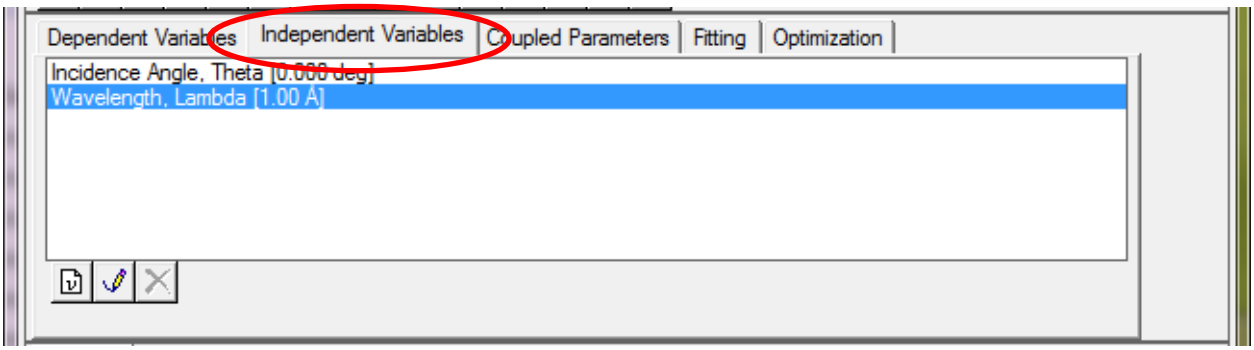
For Material and Thickness, we'll enter "Zr" and 1500.00 in place of the default values of "Vacuum" and 0.00, respectively:



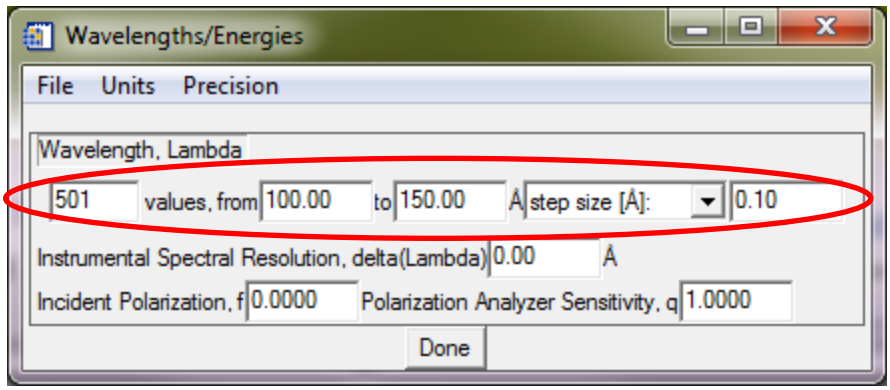
Next, as we want to compute the Transmittance of this filter, we check the box labeled "Transmittance, Phase" in the Variables area of the main IMD window:



Next, we click the "Independent Variables" tab on the main IMD window. That tab looks like this at startup:

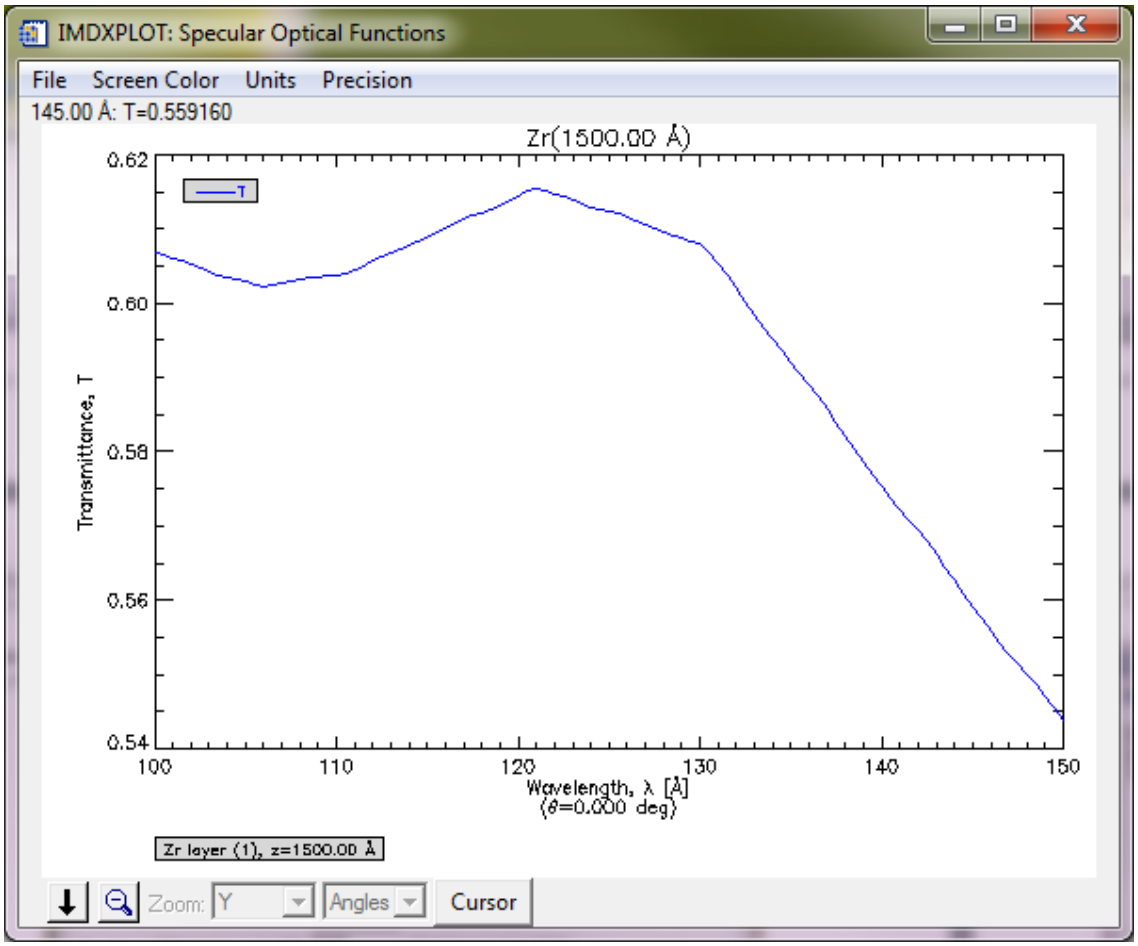


We double-click the second item in the list on the Independent Variables tab, "Wavelength, Lambda", so that we can specify the desired range of wavelengths to be used for the computation. We'll specify 501 wavelength values, from 100.00 Å to 150.00 Å, with a resulting step size of 0.1 Å:



The default value for incidence angle is 0°, which is what we want, so we don't have to make any changes to the Angles independent variable that's listed first on the Independent Variables tab.

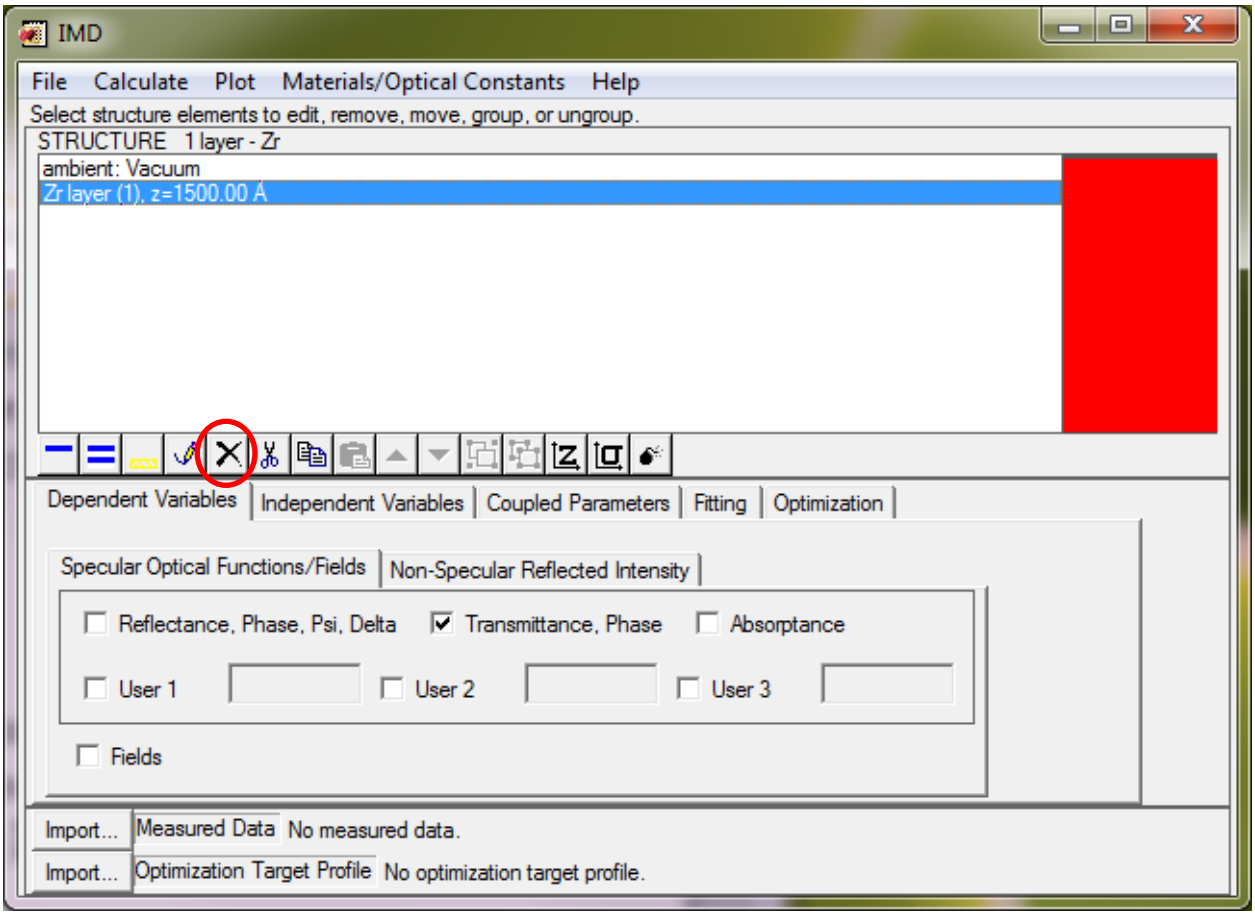
We're now ready for the calculation, so we select "Calculate→Specular Optical Functions/Fields" from the menu bar on the main IMD window. After the calculation is complete, the results appear in an IMDXPLOT window:



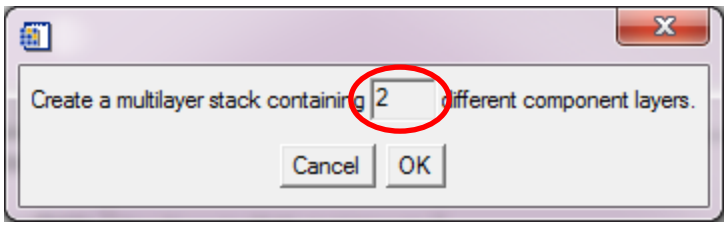
We'll need this result later, to compute the efficiency of the entire instrument, so we'll save it to a file called "`Zr_filter.imd`", by selecting "File→Save As…" from either the IMD menu bar or the IMDXPLOT menu bar. (A copy of this file can be found in the directory called "EUV `Telescope`" in the "`imd/examples`" directory.)

2.3.2 Example: Reflectance of a Si/Mo Multilayer

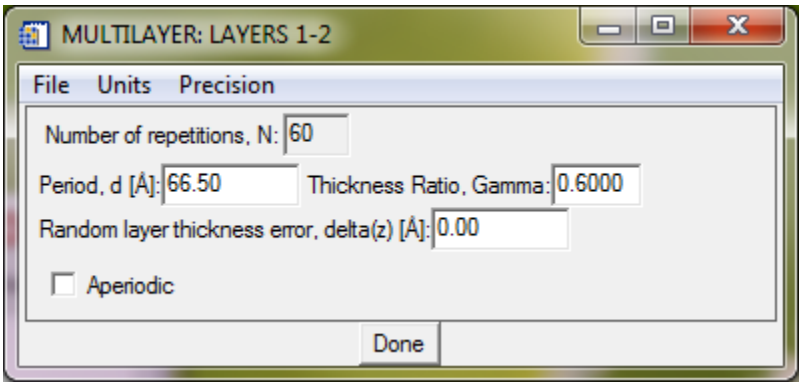
The next calculation to perform is the reflectance of the Si/Mo multilayer coatings used on the two telescope mirrors. We'll first delete the Zr layer just defined: select that layer in the Structure list in the main IMD window and then click on the Remove button (the one with this icon: ✕):



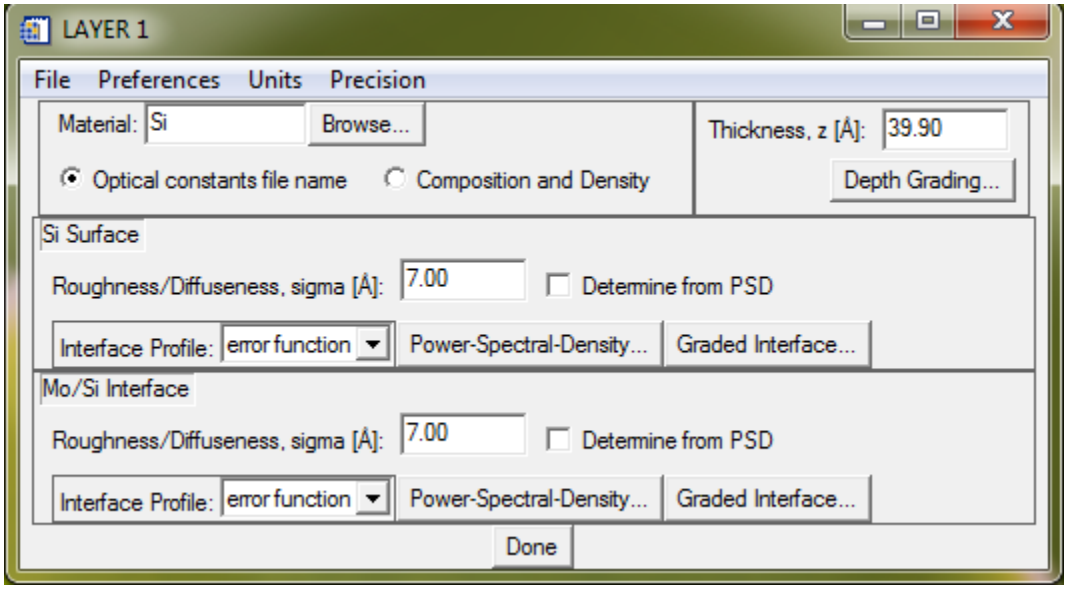
Next we'll add a multilayer by clicking the "Add Multilayer" button (second button from the left, with two blue and two white horizontal bars: 〓); Our Si/Mo multilayer contains two different "component" layers – Si and Mo – so we enter "2" in the pop-up that appears:

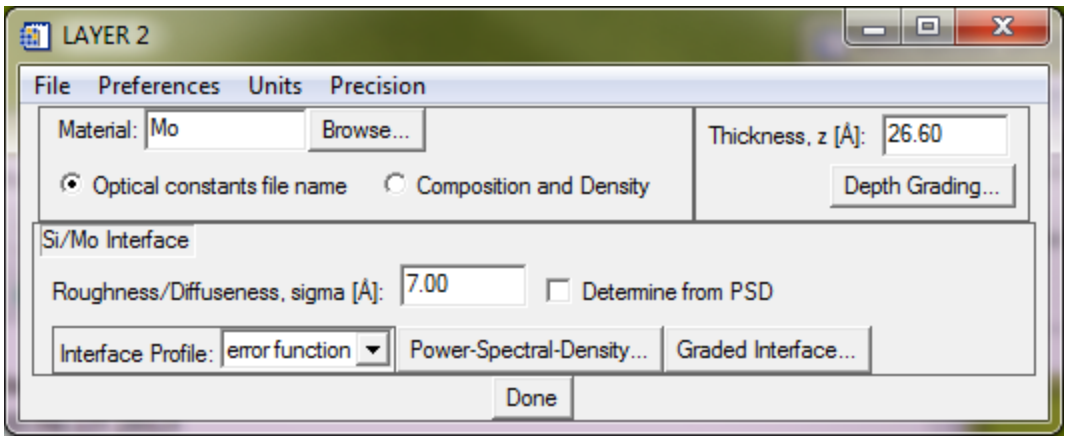


After clicking "OK", two new layers will be added to the structure and grouped together to form a multilayer. In the Multilayer window that appears next, we specify N=60 repetitions, a multilayer period of d=66.50 Å, and a layer thickness ratio (defined later) of Γ=0.6000:

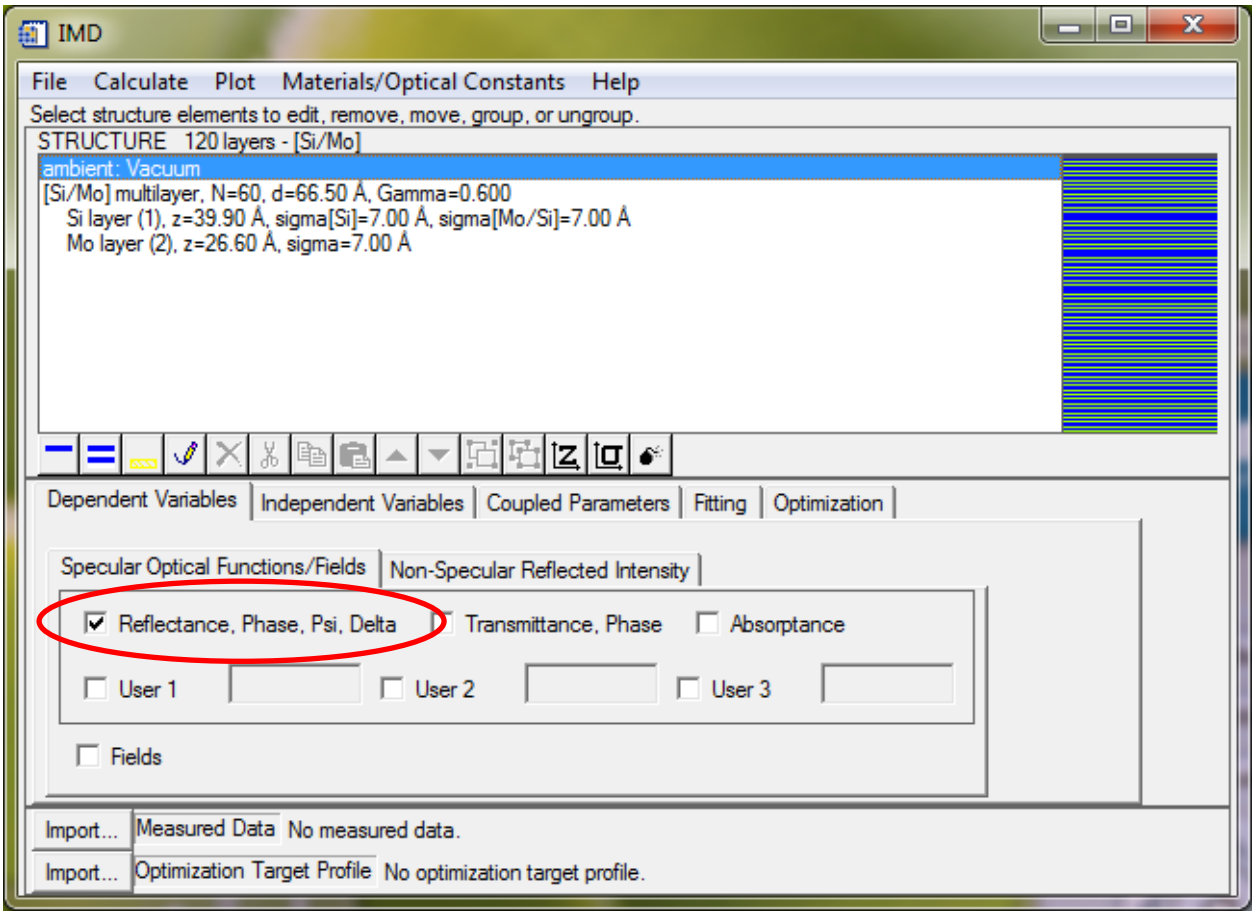


In the two Layer windows, we then specify the materials Si and Mo, respectively, and to account for interface imperfections that occur in Si/Mo multilayers, we specify σ=7.00 Å interface widths in all the boxes labeled "Roughness/Diffuseness, sigma":





Finally, we'll uncheck the Transmittance box in the Dependent Variables section of the main IMD window that we needed for the Zr filter calculation, and we'll check the box labeled "Reflectance, Phase, Psi, Delta":



We're now ready to make the calculation of multilayer reflectance: we again select from the menu bar "Calculate→Specular Optical Functions/Fields", and the results appear in a new IMDXPLOT window:

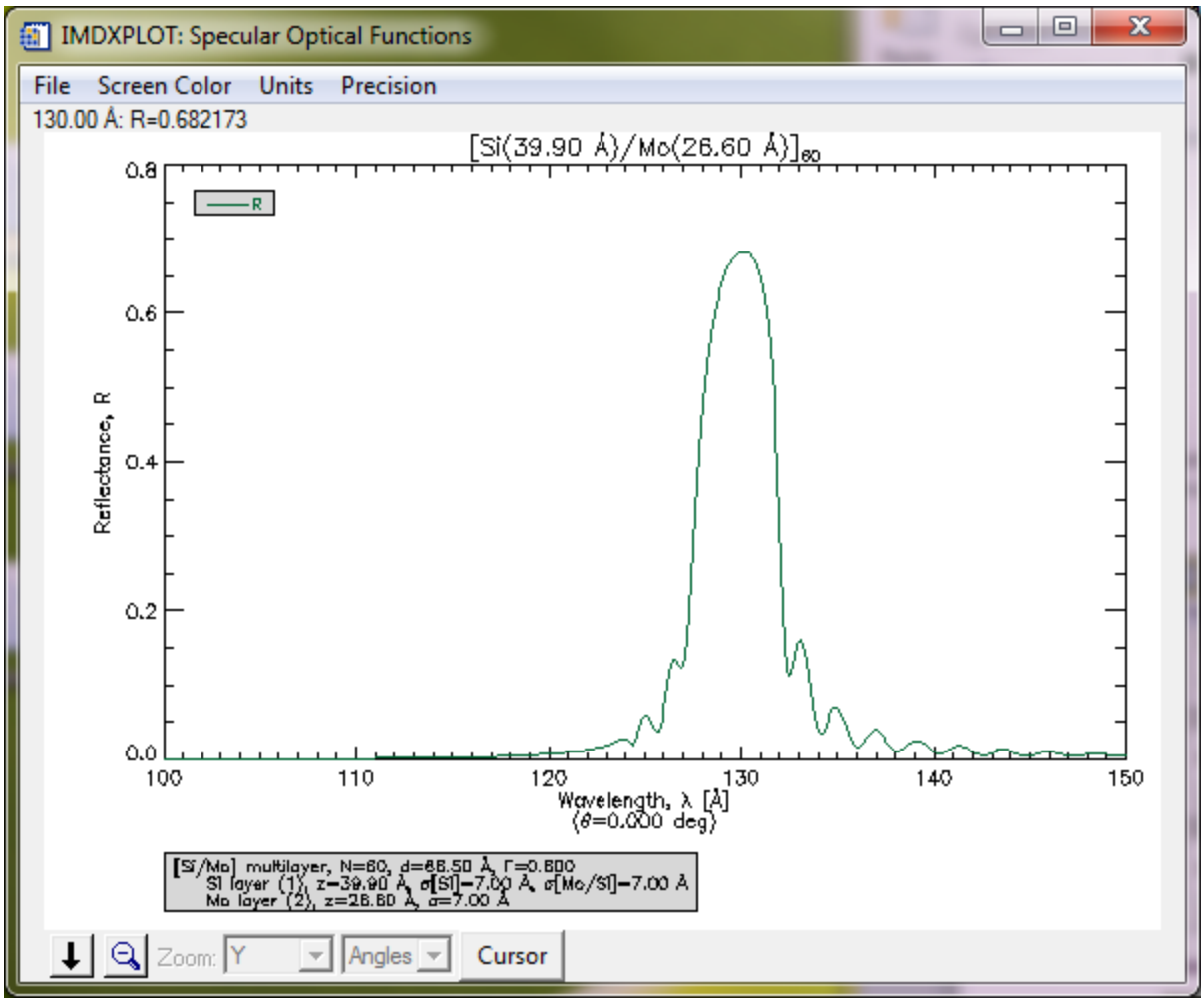


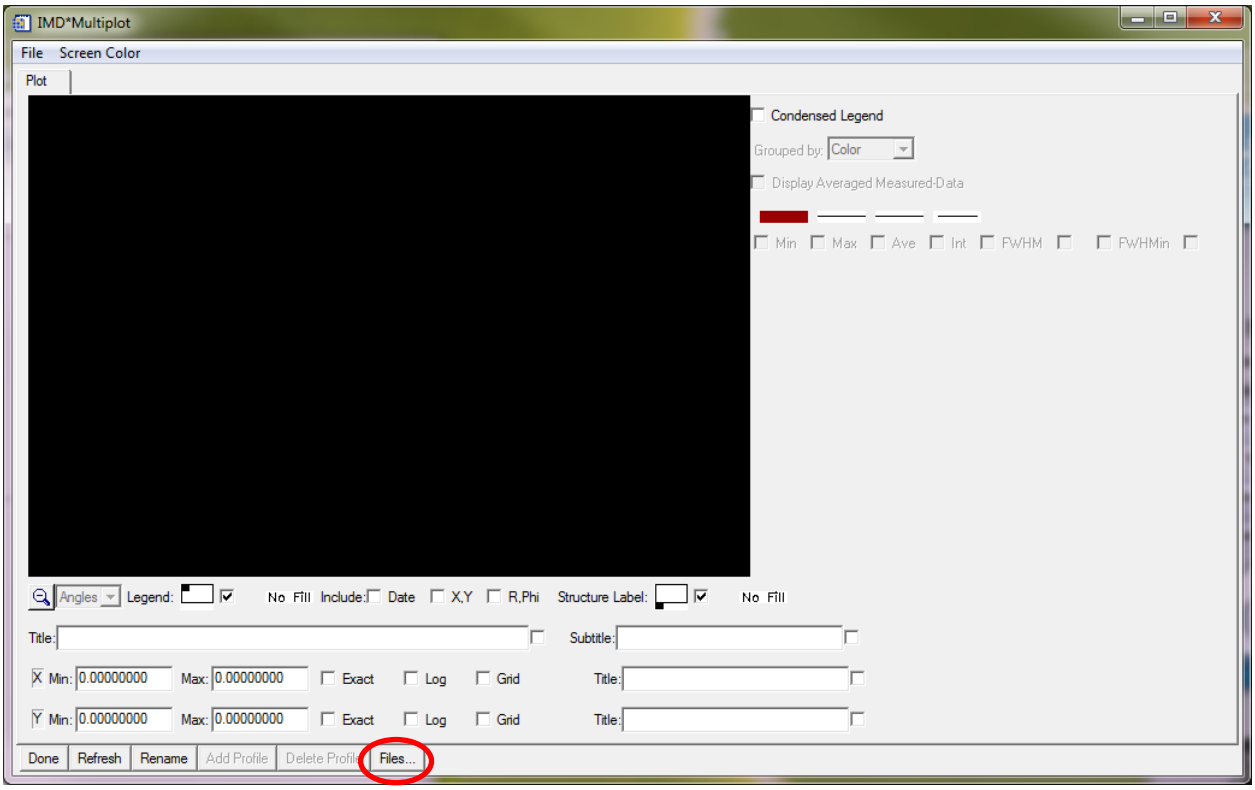
Figure 1. IMDXPLOT window displaying calculated reflectance vs. wavelength.

We'll save these results too (again using "File→Save As…"), as "`Si_Mo_ML.imd`". (A copy of this file also can be found in the directory called "EUV `Telescope`" in the "`imd/examples`" directory.)

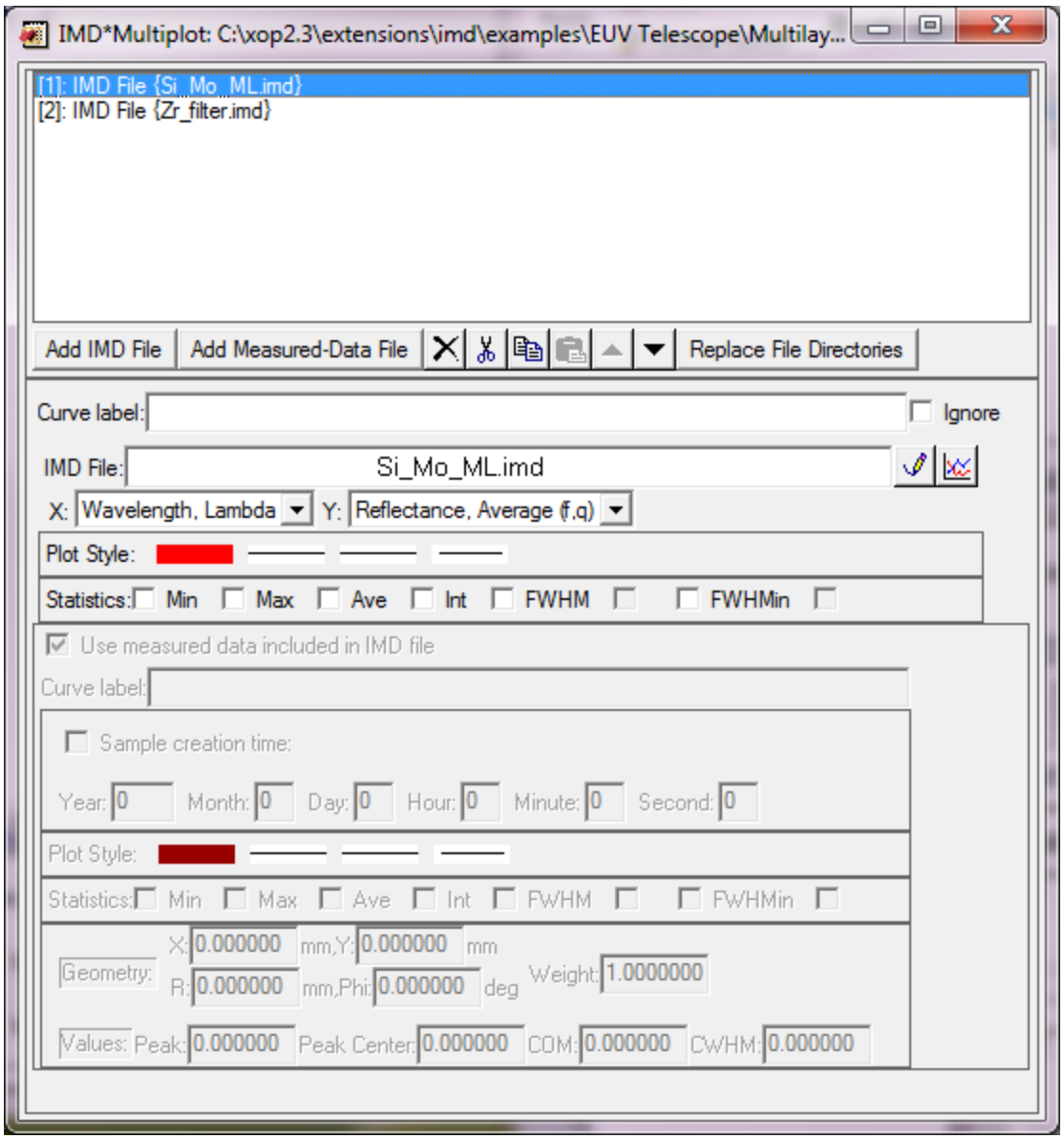
2.3.3 Example: Displaying Filter Transmittance and Multilayer Reflectance in the Same Plot

To recap: at this point we've calculated the transmittance of a Zr filter and the reflectance of a Si/Mo multilayer, and have saved both results as `.imd` files. Now we want to put it all together. First, we'll display both the filter transmittance and the multilayer reflectance in the same plot, using IMD*Multiplot.

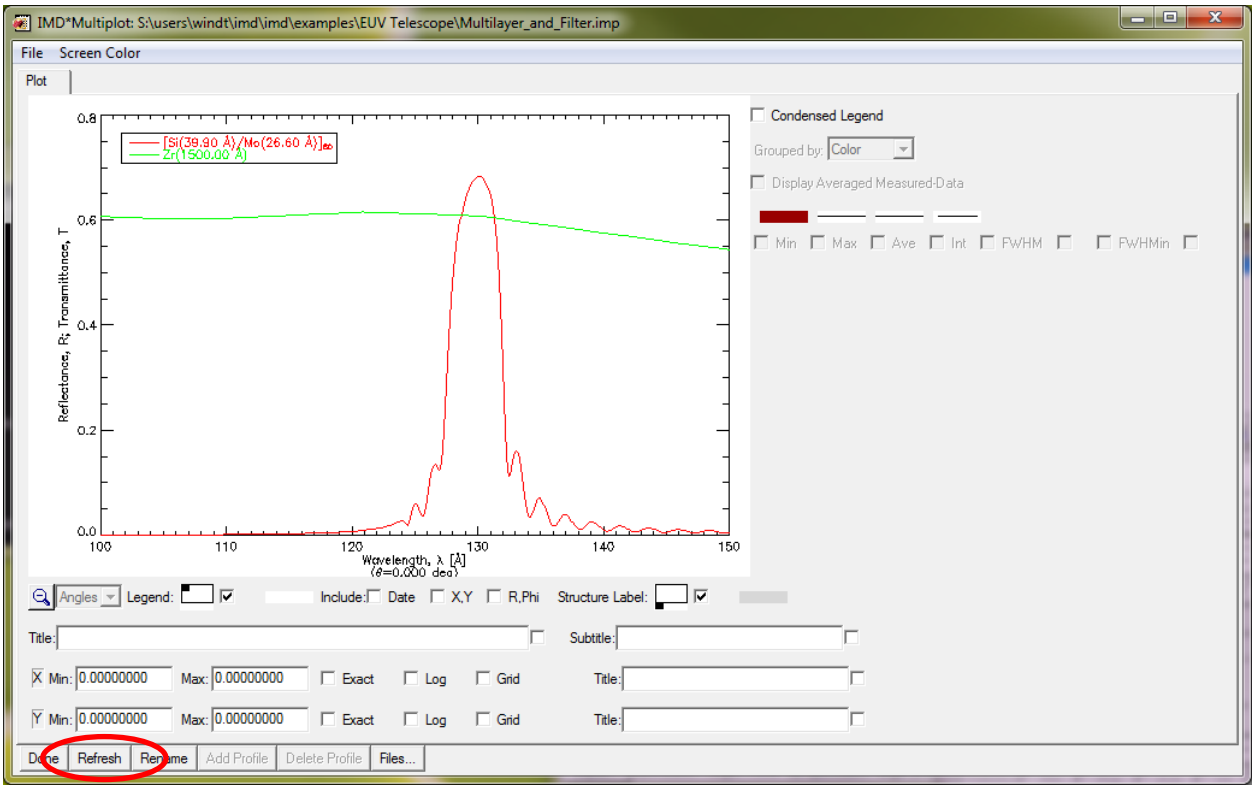
Click the "IMD*Multiplot" button in IMD Launcher, to open a new instance of IMD*Multiplot (or type `imdmultiplot` at the IMD prompt, if you are running IMD as an IDL application). A new IMD*Multiplot window will appear:



Click on the little "Files…" button at the bottom of the IMD*Multiplot window (circled in red in the figure above) to open the "IMD*Multiplot – Files" window associated with this instance of IMD*Multiplot. Then, in the "IMD*Multiplot – Files" window that appears (shown in the next figure), we'll click the button labeled "Add IMD File", twice – this will make room for the two IMD files we just saved and want to display, "`Zr_filter.imd`" and "`Si_Mo_ML.imd`". We'll select the first file in the list in the "IMD*Multiplot – Files" window and enter "`Si_Mo_ML.imd`" in the space labeled "IMD File:". Then we'll select the second file in the list and enter "`Zr_filter.imd`":



With the two files added to the list in "IMD*Multiplot - Files" we'll now return to the main IMD*Multiplot window and click the button labeled "Refresh" to display the results:



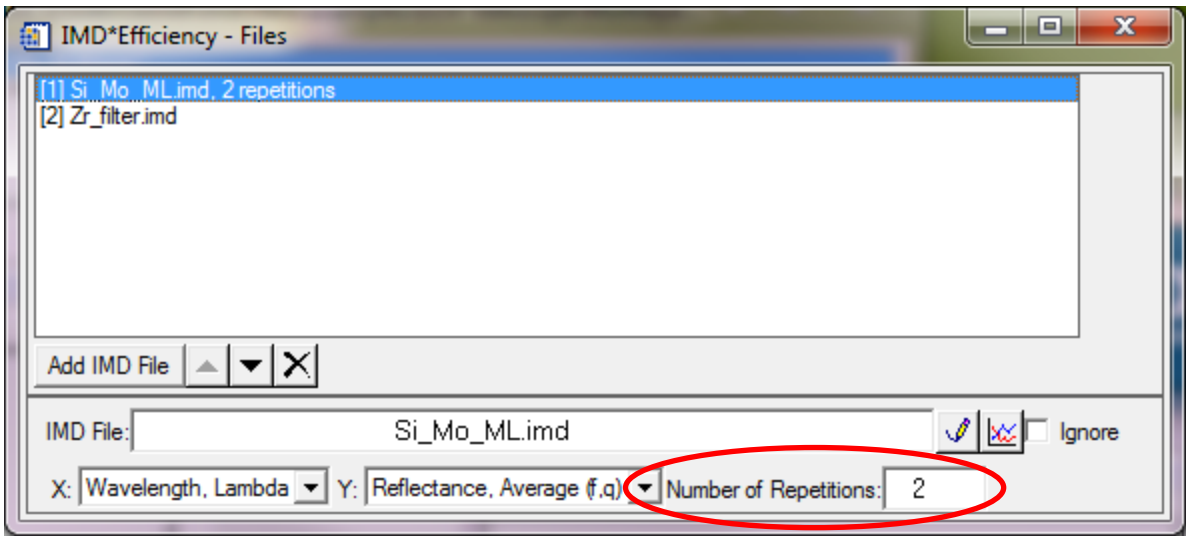
We'll save this IMD*Multiplot result as a ".imp" file: "`Multilayer_and_Filter.imp`". (A copy of this file can be found in the directory called "EUV `Telescope`" in the "`imd/examples`" directory.)

> **Note**: To open this file successfully, **be sure to set "`imd/examples/EUV Telescope`" as your working directory in IMD**, using "File→Change Working Directory…" from the IMD or IMD Launcher menu bars, or using the IDL 'cd' command at the IMD> prompt.

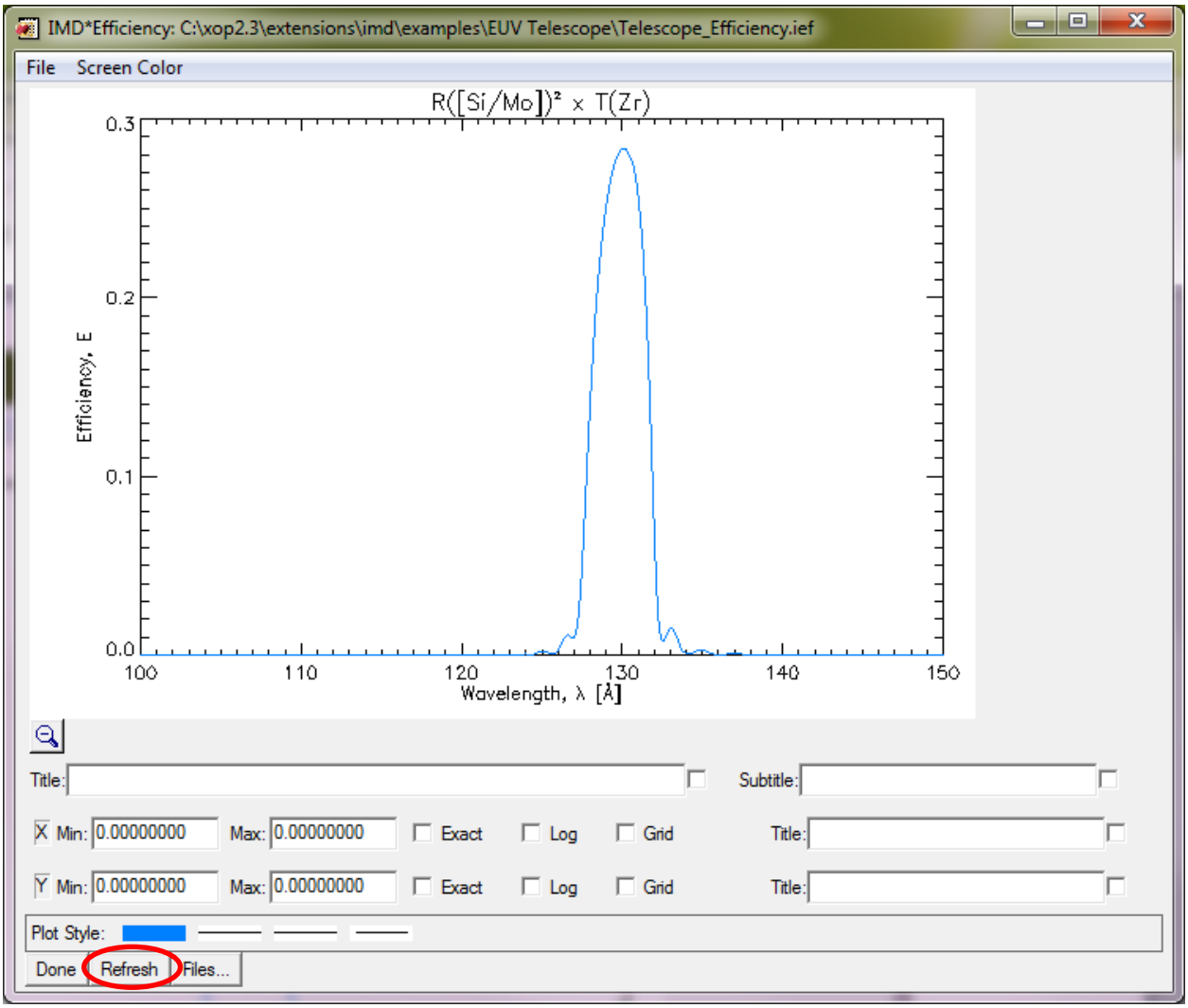
2.3.4 Example: Computing and Displaying Net Efficiency of an Instrument

The final example in this section shows how to compute the efficiency vs. wavelength of the whole instrument whose components we've just modeled. There is one filter and two mirror reflections in series, so we'll need to compute the product of three terms: $T(\lambda)_{filter}$ x $R(\lambda)_{multilayer}$ x $R(\lambda)_{multilayer}$.

Click the "IMD*Efficiency" button in IMD Launcher to open a new instance of IMD*Efficiency. (Or, if you are running IMD as an IDL application, type "imdefficiency" at the IMD prompt.) Once the IMD*Efficiency window appears, click the button labeled "Files…" to open the "IMD*Efficiency – Files" window associated with this instance of IMD*Efficiency. Add the IMD files we saved previously corresponding to the Zr filter and the Si/Mo multilayer, just as we did with IMD*Multiplot in the previous section. As we want to multiply by the Si/Mo multilayer reflectance twice, we'll specify 2 repetitions for this file, and 1 repetition for the Zr filter file:



Finally, we return to the IMD*Efficiency main window and click "Refresh" to display the results:



You will find IMD (`.imd`), IMD*Multiplot (`.imp`) and IMD*Efficiency (`.ief`) files corresponding to all the examples just discussed in the "`imd/examples/EUV Telescope`" directory. Note, however, that the current IDL 'working directory' must be set to this directory in order for the IMD*Multiplot and IMD*Efficiency files to open correctly. (You can use "File→Change Working Directory…" from the IMD or IMD Launcher menu bars.)

3 Specifying Materials: Optical Constants & Atomic Scattering Factors

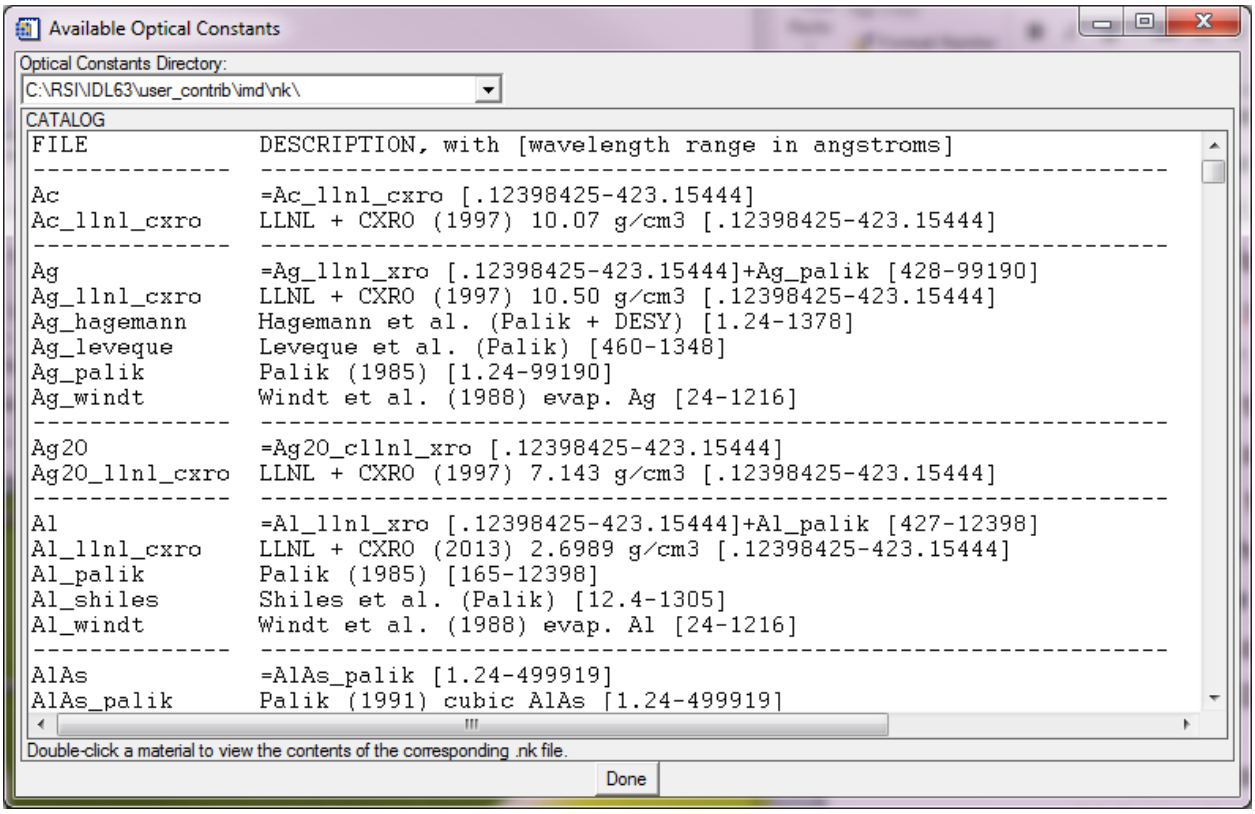
We'll now describe in detail how IMD works and how to use it, starting in this section with a discussion of the optical properties of materials as it relates to calculations of specular and non-specular optical functions made with IMD.

IMD is based on classical electromagnetic theory, where the optical properties of a material are described by its complex index of refraction **n**=n+*i*k: the real part, n, is the Refractive Index, and the imaginary part, k, is the Extinction Coefficient. Both n and k are functions of wavelength. IMD uses tabulated values of optical constants taken from the literature; in the EUV/X-ray range, IMD also can use optical constants computed from tabulated values of atomic scattering factors. However the accuracy of these tabulated values is not guaranteed in all cases. In other words, please keep in mind that **a computation made with IMD is only as good as the optical constants used to describe the materials being considered.**

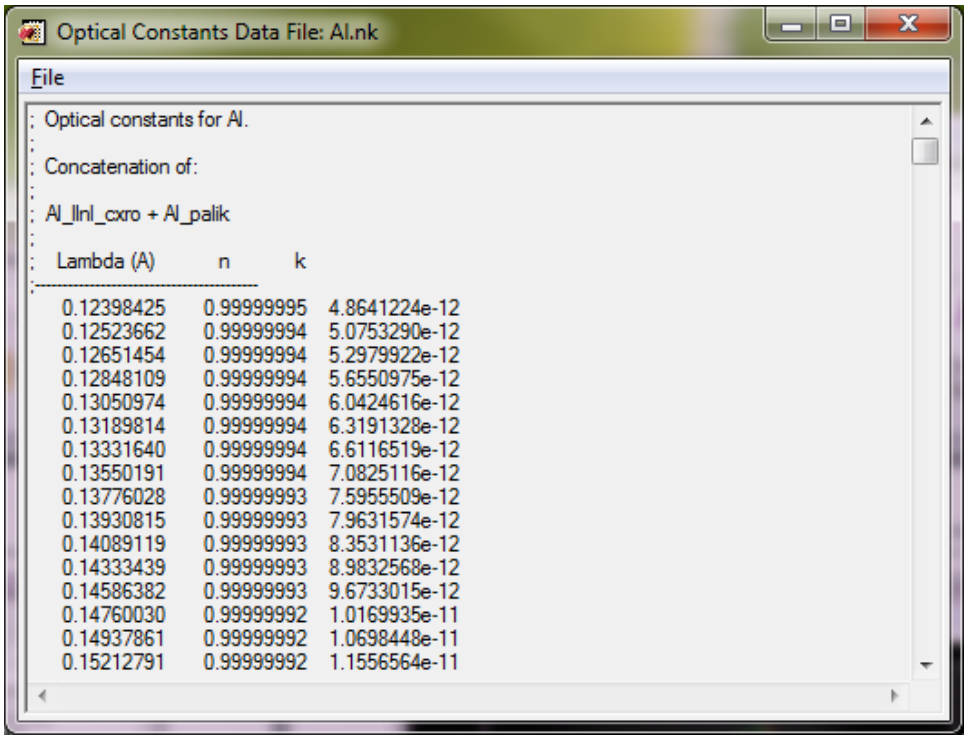
3.1 The IMD Optical Constants Database

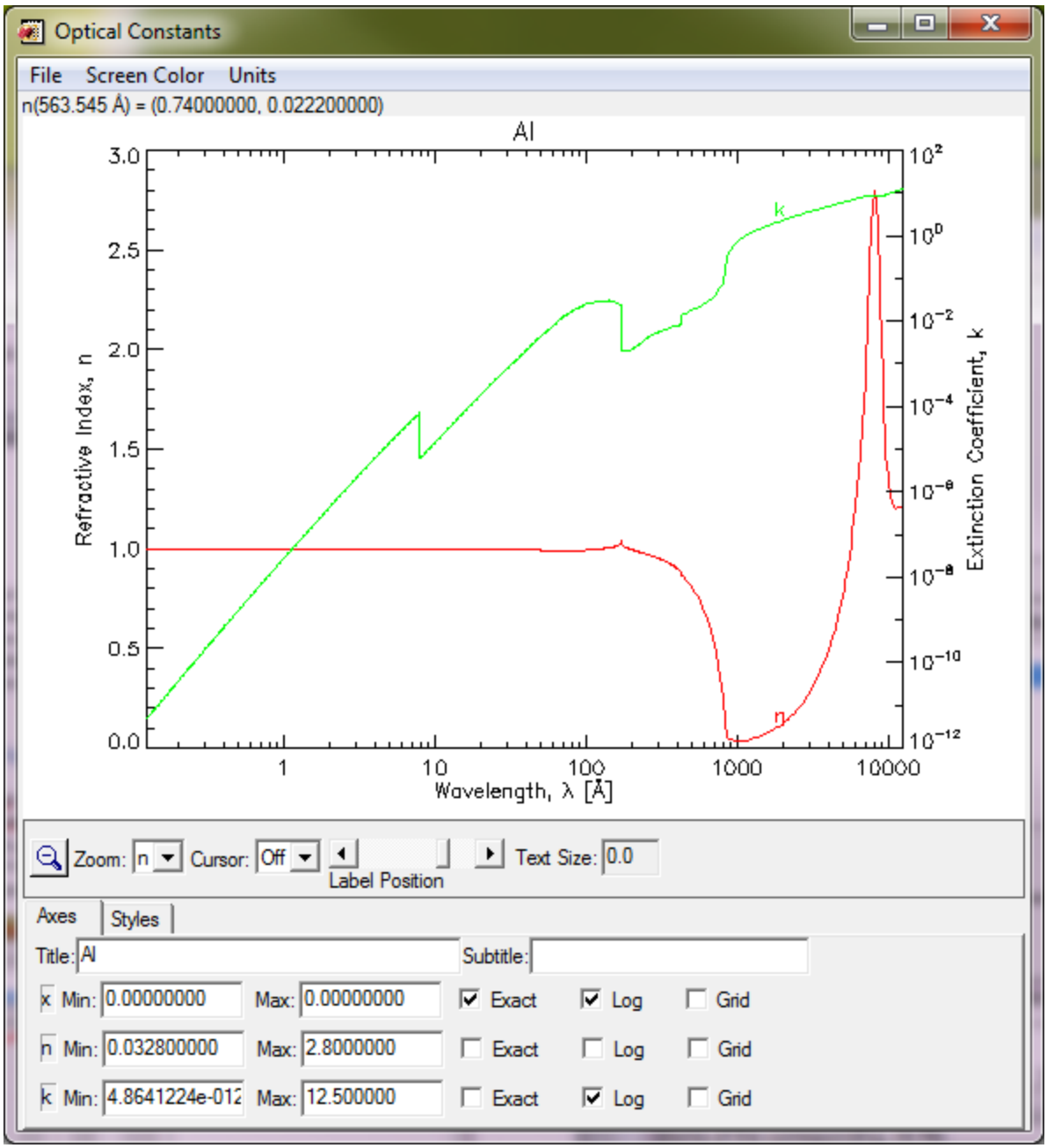
The default optical constants database included with IMD contains optical constants for over 150 materials, spanning the infrared to the X-ray range. This database is located in the directory called 'nk', in the imd installation directory, and comprises individual "nk" files designated by a '.nk' extension: these are simply plain text files containing three columns of data: wavelength [in angstroms, increasing monotonically], n, and k. Comment lines (beginning with a semicolon) may precede the data. Each nk file corresponds to one material, although there is usually more than one nk file for any given material. The naming convention used for nk files is: `Material_reference.nk`. For example, the file `Si_palik.nk` corresponds to the optical constants for silicon, taken from the optical constant handbook by Palik;[16] the file `Si_llnl_cxro.nk` corresponds to other data for silicon, in this case data in the X-ray region compiled from the LLNL and CXRO atomic scattering factors as described below in §3.3. Source reference information is usually included as a comment at the beginning of the nk file. In addition to these source-specific files, for every material included in the `nk` directory there are "compilation" nk files that cover as wide a range in wavelength as possible. The default compilation files include no reference in the file name. For example, the file `Si.nk` is a compilation of the data from `Si_llnl_cxro.nk` and `Si_palik.nk`, and spans the range from 0.124 Å to 3.33×10^6 Å. Other nk files present in the default database may be compilations as well.

A catalog of the optical constants files that are contained in the default database is contained in the file called `AAACATALOG.TXT`, which is also located in the `nk` directory. This file is displayed in the "Available Optical Constants" window that appears when you select "Materials/Optical Constants→Browse Optical Constants Database…" from the IMD menu bar:



You can scroll through the list of available nk files, and double-click on any file to view the contents of that file. For example, here's what you'll see when you double-click on "Al":

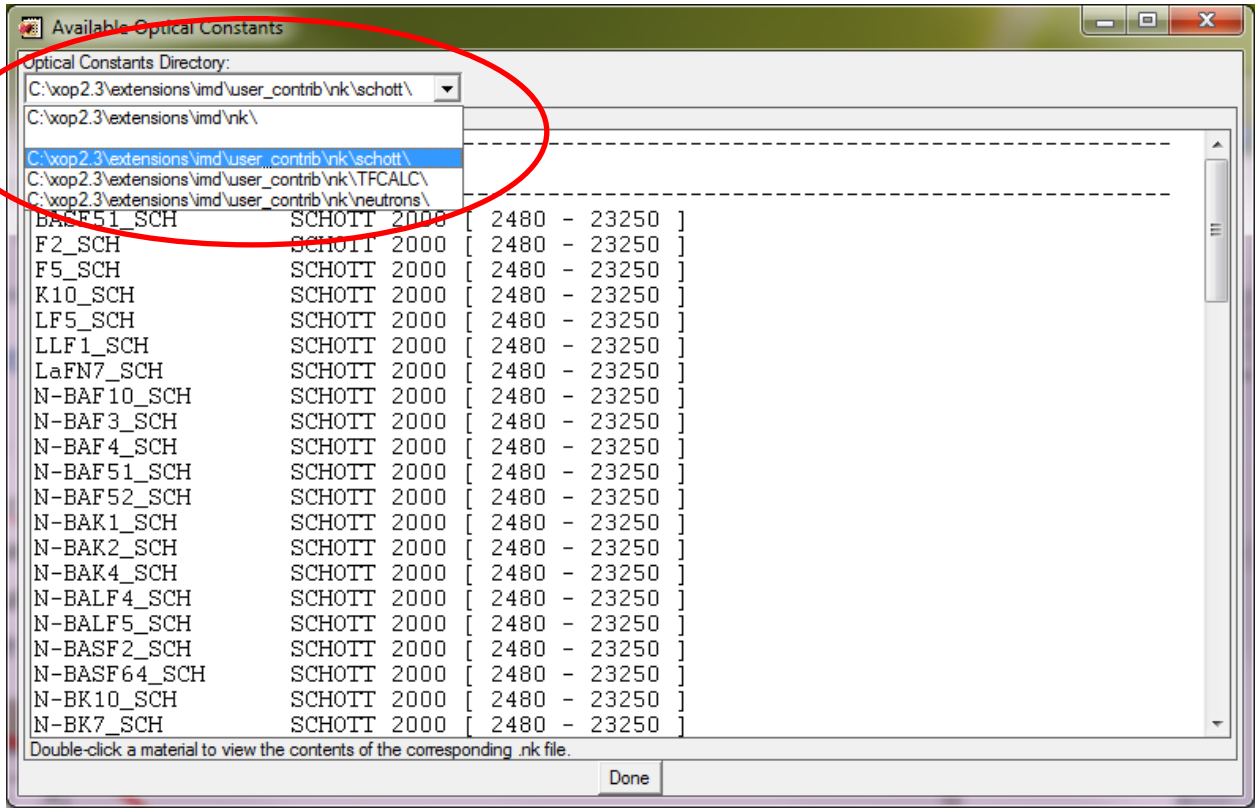




That is, the contents of the selected file are displayed as text in one window, and graphically in another window. As you can see in the last figure above, there are a number of controls in the window that can be used to adjust the appearance of the plot: axis titles, data ranges, grid lines, plots styles, wavelength units, and so forth. I won't explain these options in detail here - my hope is that it's intuitively obvious, so please explore these options on your own.

3.2 User-Contributed Optical Constants

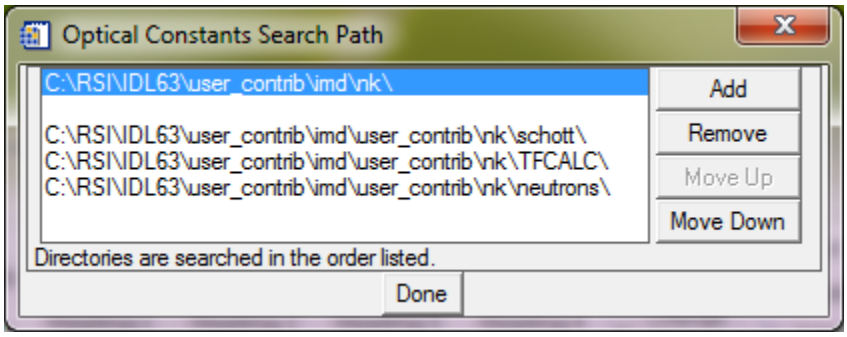
The directory '`user_contrib`' in the `imd` installation directory contains a second directory called `nk`, and within that second `user_contrib/nk` directory are additional collections of nk files that have been contributed by IMD users. You can view the contents of the individual nk catalog files (i.e., the `AAACATALOG.TXT` files) that are located in each of these additional optical constants databases using the drop-list at the top of the "Available Optical Constants" window:



You can also use your own optical constants for IMD computations. To use your own optical constants, follow these two steps:

1) Create one or more nk files: each file must be a plain text file consisting of three columns of data – wavelength [in angstroms, monotonically increasing], n, and k. You can include comments, which must be preceded by semicolons in the first column, and must be located at the beginning of the file; there can be no comments or blank lines anywhere else in the file. (Make sure there are no blank lines at the end of the file.) You should also follow the file-naming convention described in the previous section; that way, when creating plots, IMD can display the reference portion of the file name as a subscript. Also, any numbers in the material name will be displayed as subscripts on any plots produced by IMD, e.g., a material that uses optical constants in `SiO2.nk` will be displayed as "SiO_2" on the plot.

2) Tell IMD where to find the file you've created. This is done by selecting "Materials/Optical Constants→Optical Constants Search Path" from the IMD menu bar:



From this window you can add and subtract directories to the IMD optical constants search path using the buttons on the right; you can also adjust the order of directories searched: IMD will search the directories in the order listed, and will use the first file found that matches the material name you specify (as described in Chapter 4). Please note, however, that any changes to the optical constants search path made following this method will be temporary, and will be lost after you exit IMD. For permanent changes to the optical constants search path you must edit the IMD configuration file (`imd_cfg.pro`) discussed above in §1.4.2.

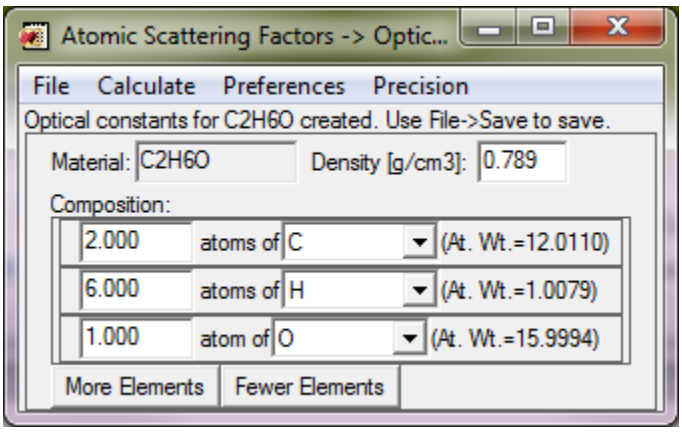
If you have created your own optical constants directories for use with IMD, you can also create a catalog file called `AAACATALOG.TXT` to reside in each additional nk directory you create. By doing so you will be able to use the "Materials/Optical Constants→Browse Optical Constants Database" option to view the contents of your files, as explained above. Follow the same format that's used in the `AAACATALOG.TXT` file in the default `nk` directory.

3.3 Atomic Scattering Factors

The IMD installation includes a directory called `f1f2` that contains X-ray atomic scattering factor data for 92 elements (Z=1–92). The atomic scattering factors f1 & f2 can be used to compute optical constants n & k for elemental or compound materials of known density, as described in reference [2]. The "`.ff`" atomic scattering factor files included with IMD in the `f1f2` directory are plain text files containing three columns of data: energy [in eV, monotonically increasing], f1, and f2. Each file is a concatenation of high-energy X-ray data obtained from LLNL[3] extending from 30 keV to 100 keV, and low-energy data from CXRO[2] extending from 30 keV to 10eV. (Note, however, that optical constants can only be computed at energies where both f1 and f2 are valid; for most materials, the resulting low-energy limit is ~30 eV, though optical constants for some materials can be computed to ~20 eV.)

Atomic scattering factors can be used directly for computations in IMD, in place of optical constants files, by specifying a material's composition and density, as we'll see in Chapter 4. Alternatively, IMD also includes a stand-alone program – called IMD_F1F2TONK – that provides a user-friendly interface to

the internal IMD function[iii] that computes optical constants from atomic scattering factors and atomic weights, using the material density that you specify. You can use this program to generate X-ray optical constants that can then be saved to an nk file for future use. (The nk files named `*_llnl_cxro.nk` in the default IMD database were in fact all generated using the IMD_F1F2TONK program.) The IMD_F1F2TONK procedure can be run by selecting "Materials/Optical Constants→Create New Optical Constants from Atomic Scattering Factors…" from the IMD menu bar; it also can be run independently of IMD, by typing `imd_f1f2tonk` at the IMD prompt. Here's how it looks after computing the optical constants of ethanol:



Once you specify the constituent elements, their relative concentrations, and the material density, using the controls on the window shown above, select "Calculate→Optical Constants" from the menu bar of this window, and then use "File→Save…" to save the results to an nk file.

3.4 User-Contributed Atomic Scattering Factors

You can use your own atomic scattering factors files in IMD in place of the default .ff files located in the `f1f2` directory. To do so, follow these steps: (1) create a new directory (for example, `…/imd/user_contrib/f1f2/my_f1f2_files`), (2) add whatever .ff files you like to this directory, and (3) use the "Materials/Optical Constants→Atomic Scattering Factors Directory…" menu option to select the new atomic scattering factors directory that you have created containing your .ff files. IMD will check for .ff files in the directory that you specify; at least one must be present. If the check is successful, any materials subsequently specified by composition and density (§4.1) in IMD will use the atomic scattering factors in the directory that you specify to compute optical constants, as will the IMD_F1F2TONK program.

If you do create new .ff files, it's important to note that the .ff file names that you choose will be used as the *element names* in IMD. For example, IMD requires that the file "`C.ff`" corresponds to the atomic

[iii] The IDL *function* included with IMD, which you can use in your own IDL programs, is also called IMD_F1F2TONK, and is described in §11.3.1.

scattering factors for the element carbon. The element name derived from the .ff file name is used for labeling windows and graphics, and, most importantly, for deriving the atomic weight of that element, which is needed for computation of optical constants.[iv] If the .ff file name that you create does not correspond to one of the 92 elements that are recognized by the IDL ATOMIC_WEIGHT function that IMD uses by default (i.e., Z=1 to 92), then you **must** provide an atomic weight value for the element in the first line of the .ff file, using the syntax "; atwt=xxx.xxx"; the first line of the file must also begin with a semicolon. For example, if you want to create an atomic scattering factors file for the isotope ^{14}C, you might call the file "`14C.ff`"; since the ATOMIC_WEIGHT function will not recognize "14C" as an element, the first line of that file should look like this:

```
; atwt=14.003241
```

Additional comment lines (also beginning with a semicolon) can be included after the first line in the .ff file. The atomic scattering factor data in the .ff file must comprise three columns: energy [in eV, monotonically increasing], f1, and f2.

3.5 Neutrons

Also included with IMD are atomic scattering factors for (unpolarized) neutrons, provided by M. Sanchez del Rio and L. Alianelli. These data are located in the `user_contrib/f1f2/neutrons` directory in the `imd` installation directory, and they can be used to model neutron optical properties when specifying materials by composition and density (described in Chapter 4). These data also can be used with IMD_F1F2TONK to generate neutron nk files.

To use the neutron atomic scattering factors in place of the default X-ray atomic scattering factors, you must change the atomic scattering factors working directory as follows: select from the menu bar on the main IMD window "Materials/Optical Constants→Atomic Scattering Factors Directory…", and then select the neutron atomic scattering factors directory …`/imd/user_contrib/f1f2/neutrons`. **Please see the `README_IMD.TXT` file in the `user_contrib/f1f2/neutrons` directory for further details regarding neutron atomic scattering factors and neutron computations in IMD.**

Along with the neutron atomic scattering factors files just described are neutron optical constants (nk) files, which were generated from the neutron atomic scattering factors and also provided by del Rio and Alianelli. These nk files are located in `imd/user_contrib/nk/neutrons`, and they can be used by IMD so long as the `user_contrib/nk/neutrons` directory is included in your optical constants search path (as described above in §3.2) so that the correct nk files are used when specifying materials by reference to optical constants files.

[iv] Atomic weights are computed in IMD using the 'atomic_weight' routine, an IDL function that is included in the 'windt' IDL library, which you can download from http://www.rxollc.com/idl.

4 Defining the Film Stack

IMD computes optical functions for the case of a plane wave incident on a film stack comprising any number of layers, with or without a semi-infinite substrate; the 'ambient' refers to the semi-infinite region above the film stack, as shown in Figure 2.

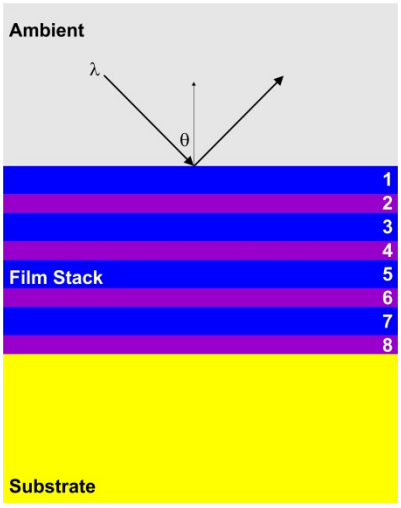


Figure 2. A film stack on a substrate, with a plane wave of wavelength λ incident at an angle θ. In this particular example, the film stack comprises a periodic multilayer having 4 bilayers, or 8 layers in total, numbered (from the top) 1 through 8.

In this chapter we'll see how to define the ambient, the film stack, and the substrate, and how to specify materials, layer thicknesses, and interface properties, all using IMD's graphical user interface (GUI). We'll be using the buttons located at the bottom of the Structure area on the main IMD window. These buttons, as well as the structure diagram showing a picture of the structure as currently defined, are labeled in Figure 3.

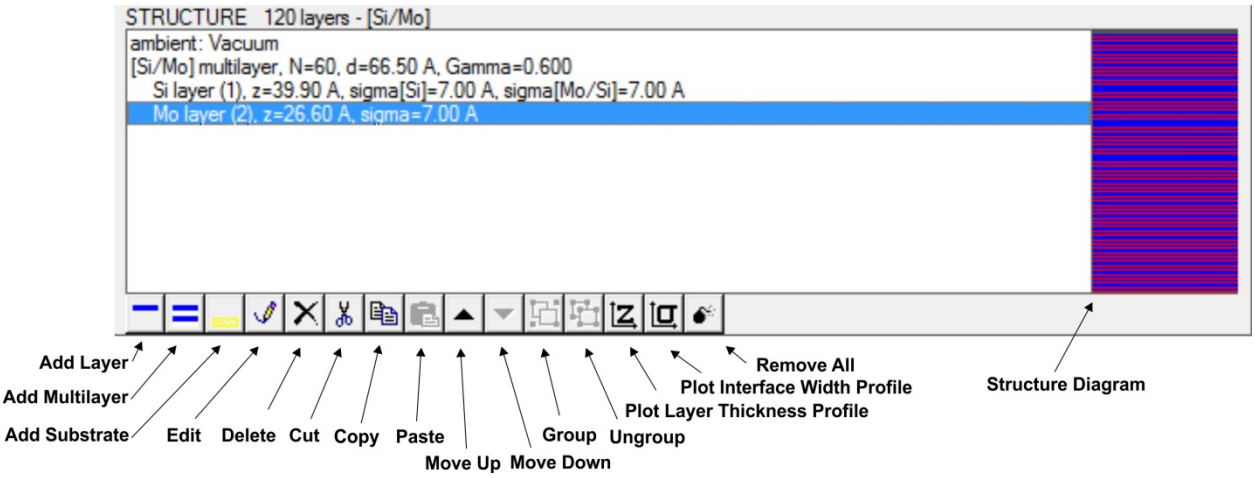
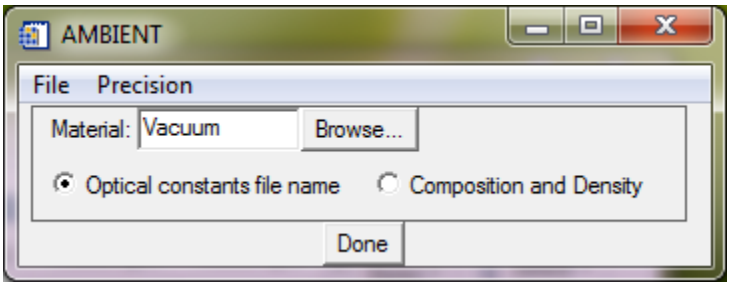


Figure 3. The Structure area of the main IMD window.

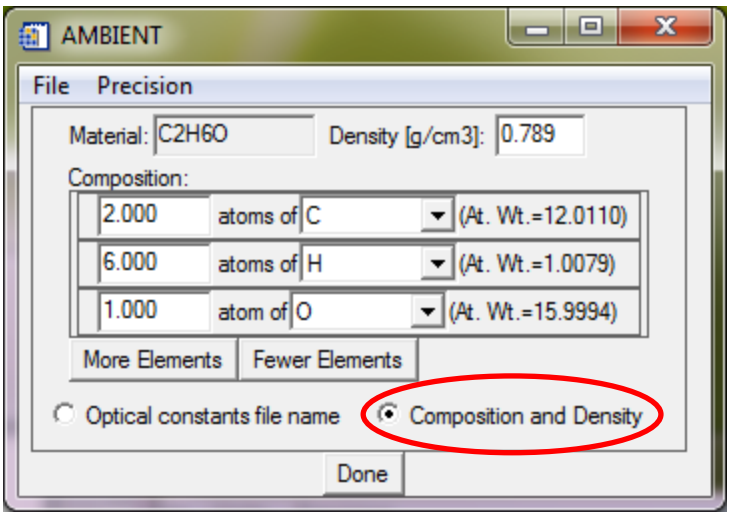
4.1 Ambient

The ambient material used for most calculations is 'Vacuum', which means that the optical constants have the value (n,k)=(1.,0.) for all wavelengths. However, there may be some cases where other ambient materials are desired. To change the ambient material, select the Ambient in the structure list, which is always the first element shown, and then click the Edit button. (Or just double-click the Ambient list item.) This will open the Ambient window:

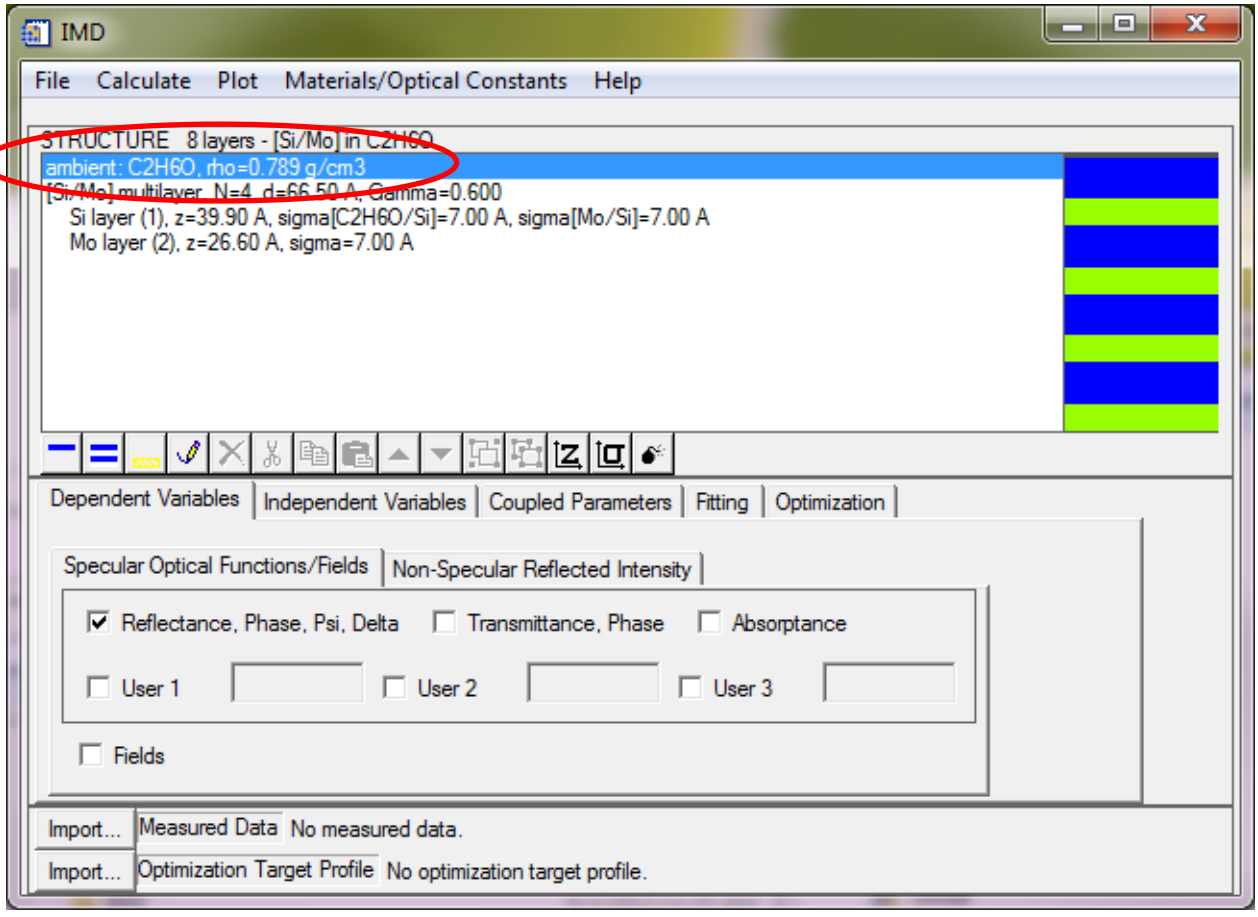


In place of "Vacuum", type the name of the material you want to use. The name you enter must correspond to the name of an nk file in the IMD optical constants database(s) discussed in the previous chapter, but without the `.nk` extension. For example, to model the performance of a mirror operating in water, in place of "Vacuum" type "H2O"; IMD will use the optical constants contained in the first file found named `H2O.nk` as it looks through the optical constants search path explained in the last chapter. There is a file called `H2O.nk` in the default IMD `nk` directory, in fact, and that file happens to be a concatenation of optical constants computed from atomic scattering factors, which are contained in the file `H2O_llnl_cxro.nk`, and the optical constants from `H2O_palik.nk`. If you want to use optical constants contained in a particular file, specify the full name of the file (but always without the `.nk` extension), for example, `H2O_palik`. Alternatively, click the "Browse…" button to the right of the material text-box to choose an nk file interactively.

For computations made in the EUV/X-ray spectral range, you can also use atomic scattering factors directly for the computation: click on the radio button labeled "Composition and Density", and then create whatever material you wish to model. The next figure shows what the Ambient window looks like after specifying ethanol (C_2H_6O) as the ambient material:



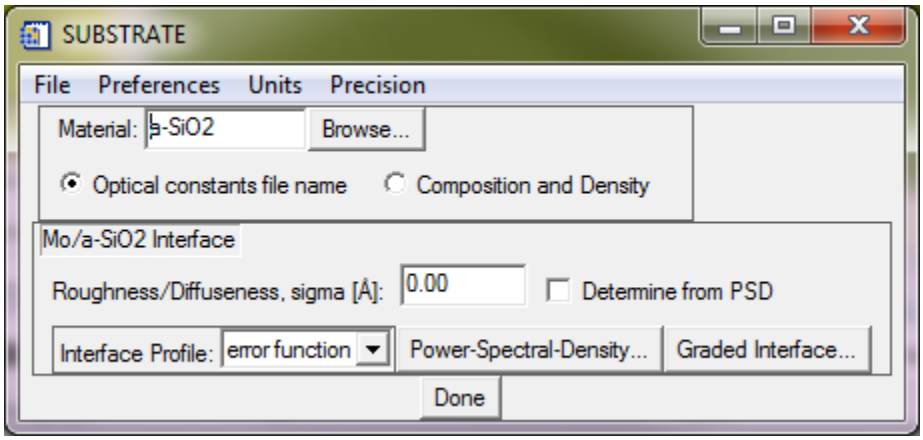
Changes made to the ambient material are also shown in the structure list:



4.2 Substrate

To include a substrate in the structure to be modeled, click the "Add Substrate" button: on the main

IMD window. The Substrate window will appear. Here's how it looks after we've typed "a-SiO2" in place of the default "Vacuum", in order to model a glass substrate:



You can also define the surface properties of the substrate (i.e., roughness, PSD, etc.) We'll explain all those details later, in §4.11.

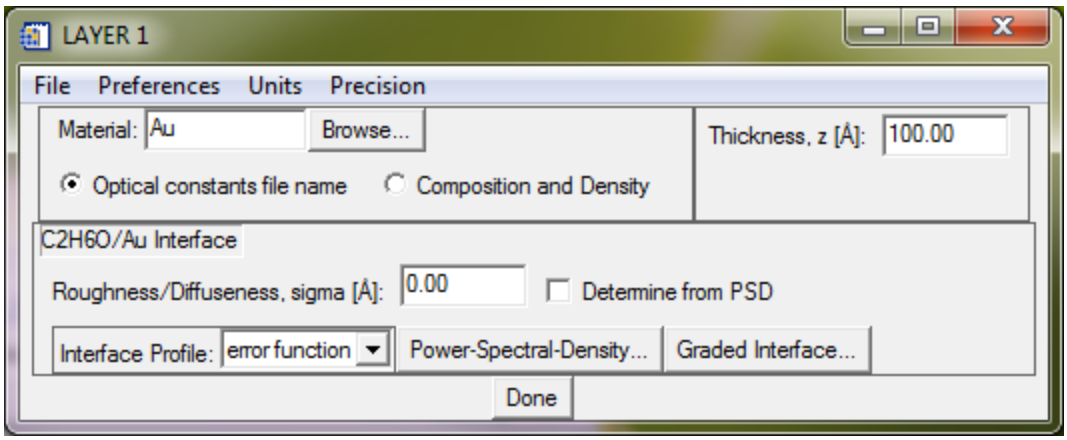
To delete the substrate, select it in the structure list, then click the delete button:  .

> **Note:** If no substrate is defined, the material used for the semi-infinite region below the film stack is the same as the material specified for the ambient region above the film stack.

> **Tip**: To specify the roughness of the backside of the last layer in a free-standing film stack, add a substrate, specify the material as "Vacuum" (or whatever material you want to use for the ambient region below the film stack), and enter the desired roughness and/or PSD parameters (explained in §4.11).

4.3 Layers

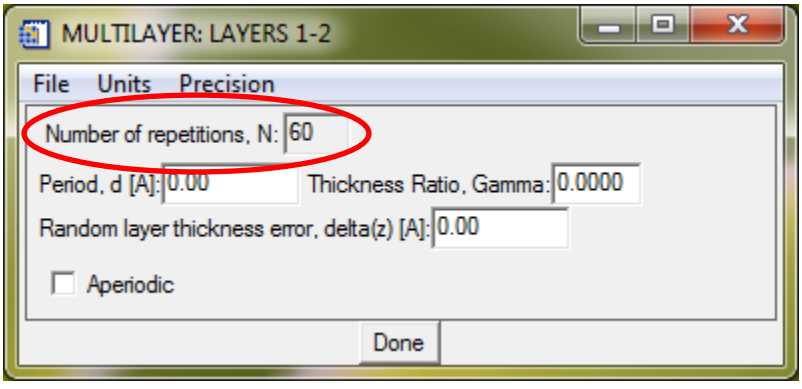
Click the Add Layer button  on the main IMD window to add a layer to the structure you want to model. This will open a corresponding Layer window. Here's how a Layer window looks after specifying gold (Au) as the layer material, and 100 Å as the layer thickness, z:



To add more layers, keep clicking the Add Layer button. Or, you can use the Copy and Paste buttons to replicate existing layers in the stack. When there is more than one layer defined, use the Move Up and Move Down buttons to adjust the order of layers within the film stack. Use the Units menu item on any Layer window to display layer thicknesses (and other length parameters as well) in either Å, nm, or μm. The Precision menu can be used to control the number of digits displayed to the right of the decimal point for the layer thickness (and interface width) that you specify.

4.4 Periodic Multilayers

Use the Add Multilayer button on the main IMD window to create a repeating group of layers containing two or more component layers (as we did in the example presented in §2.3.2). Alternatively, layers that are already defined can be grouped and ungrouped using the Group and Ungroup buttons. Once a layer group (i.e., a multilayer) has been created, open the corresponding Multilayer window in the structure list (by double-clicking) to specify the number of repetitions, N:

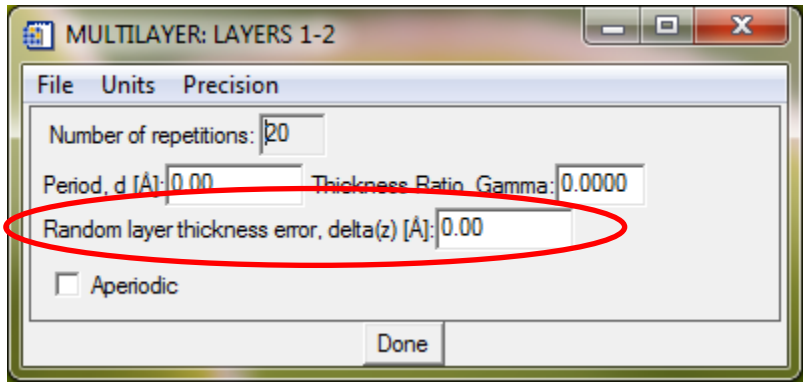


If the layer group comprises only two component layers (e.g., layer A and layer B) then you can specify directly in the Multilayer window the multilayer period, d, which is equal to the thickness of one bilayer, i.e., $d=z_A+z_B$, and the layer thickness ratio, Γ, which is defined in IMD as the thickness of the top layer of the bilayer, z_A, divided by the multilayer period, d:

$$\Gamma \equiv z_A/(z_A+z_B) = z_A/d$$

In the case of layer groups comprising more than two component layers, Γ is ill-defined and cannot be specified. Similarly, the multilayer period d cannot be specified directly in the Multilayer window when there are more than two component layers in the group; you must instead specify the individual layer thicknesses in the corresponding Layer windows.

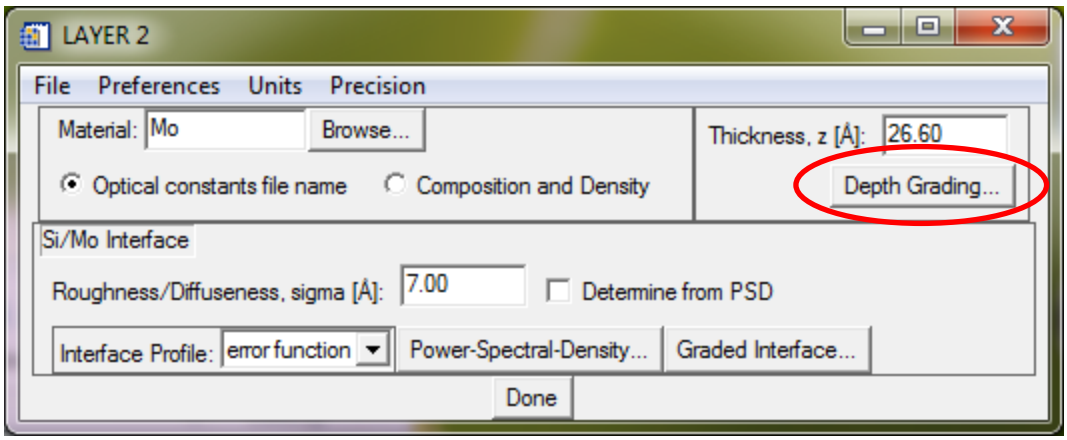
Random layer thickness errors δz can be specified in the Multilayer window as well:



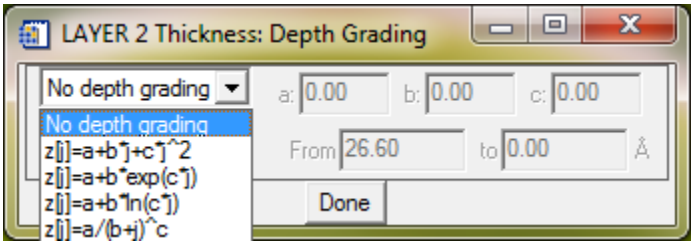
If a non-zero δz value is specified, then the thickness of each layer in that multilayer will be modified by a random amount when the computation is performed. This can be used to model, for example, layer thickness errors that might occur during film deposition. IMD uses IDL's RANDOMN function to simulate random layer thickness errors: the RANDOMN function produces normally-distributed (Gaussian), floating-point, pseudo-random numbers with a mean of zero and standard deviation of one. Thus, the thickness of layer j, z_j, will be modified as $z_j'=z_j+\delta z$*RANDOMN(seed) when IMD performs the computation using non-zero values of δz. ("seed" is an internal IMD variable that ensures that random numbers are not inadvertently repeated.)

4.5 Depth-Graded Multilayers

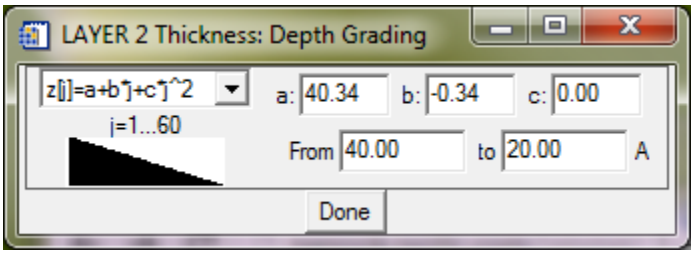
Once a layer group has been defined, the thickness of any of the component layers within the group can be made to vary analytically with depth in the film stack. Open the Layer window for the layer whose thickness is to be varied with depth, and click on the button labeled "Depth Grading…":



This will open another window where you can choose from one of four different analytical depth-grading functions z(j) from a drop-list – parabolic, exponential, logarithmic, or power-law:



For example, here's how to specify a linear variation in thickness (using the parabolic grading function) for one component layer within a layer group containing N=60 repetitions; in this case the layer thickness varies linearly (c=0) from 40 Å at the top of the film stack (j=1) to 20 Å at the bottom of the film stack (j=60):



The thickness of any or all of the layers within a layer group can be independently depth-graded following the procedure just outlined. If you want to grade every layer in a layer group following the same analytic function, you can of course specify identical depth-grading functions and proportional depth-grading parameters for each layer. However, a much easier method that accomplishes the same thing is to utilize Coupled Parameters. We'll explain in §4.12 how to use Coupled Parameters to simplify the design in IMD of such depth-graded multilayers, with specific depth-graded multilayer examples in §4.12.1 and §4.12.2.

4.6 Aperiodic Multilayers

An aperiodic multilayer in IMD is defined as a layer group containing N repetitions of two or more component layers, where the thickness of each individual layer in the stack is specified numerically, not analytically. To create an aperiodic multilayer, first create a layer group of two or more component layers, and then click the box labeled "Aperiodic" in the Multilayer window:

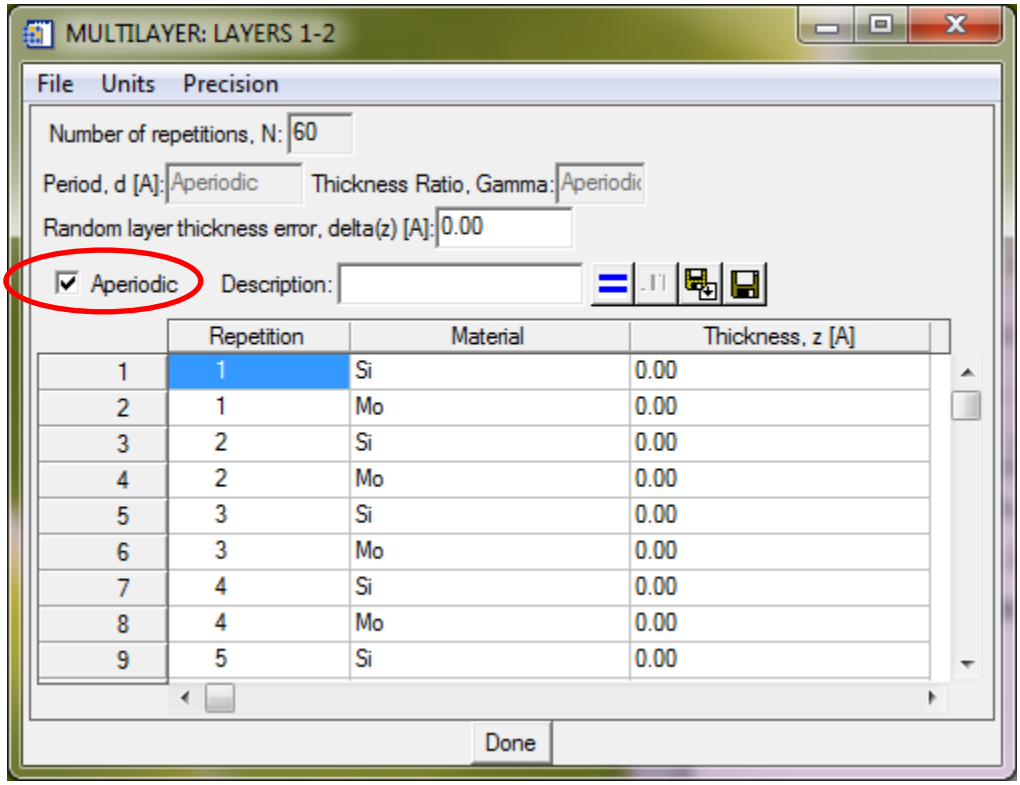


Figure 4. A Multilayer window when the Aperiodic box is checked, displaying a table of layer thicknesses.

A table will appear in the Multilayer window that displays, from left to right, the index number of the layers comprising this layer group (i.e., always starting with "1", regardless of what other layers may be defined on top of the layer group), the repetition number of each component layer in the group, the component layer material, and the thickness of each individual layer, as shown in the figure above. Only the individual layer thickness values can be changed directly in this table, however. The layer materials listed in the table correspond to the materials specified for each component layer; these materials can only be changed in the component Layer windows. (Every instance of each component layer in the aperiodic layer group will also have the same interface parameters, as defined in the component Layer windows and discussed in §4.11.) Similarly, the index number and repetition number for each layer listed in the table are determined by the number of repetitions N for the multilayer, and by the number of component layers that comprise the group. You can change N, and also change, add or delete component layers and their material designations, even after the aperiodic multilayer is created when the "Aperiodic" box is first checked. You can also enter whatever text you want in the box labeled

"Description:" – that text will appear in the structure list on the IMD main window, and in the structure labels that are included in IMD graphics, as will be described in Chapter 6.

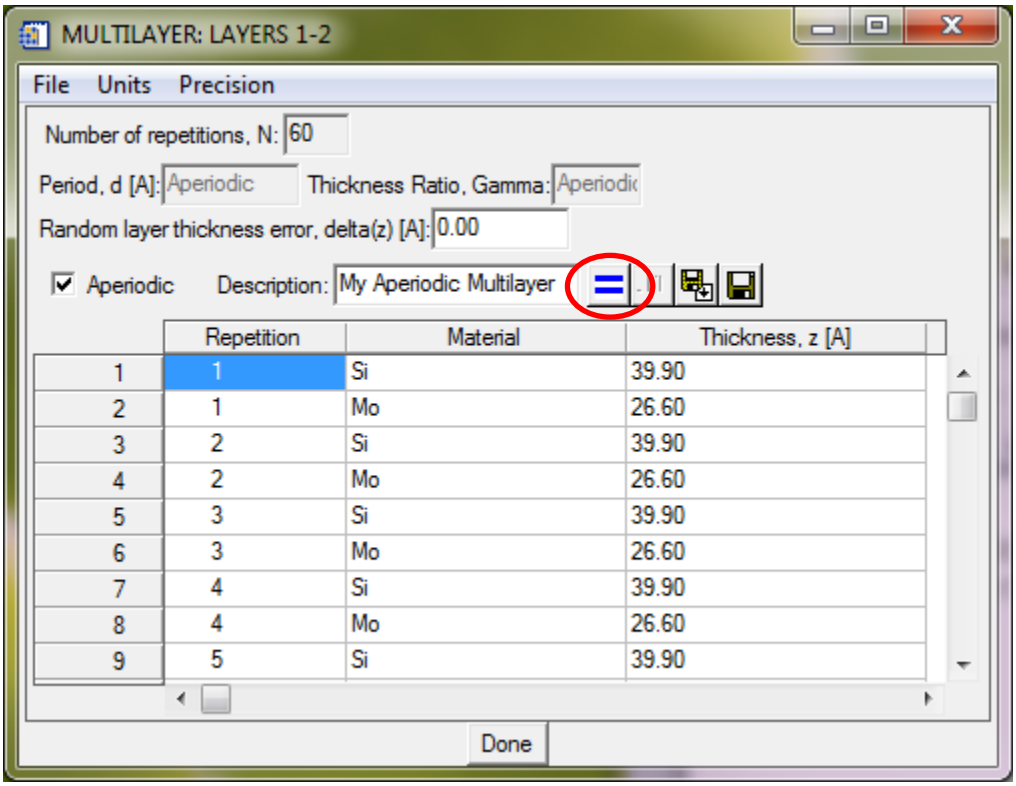
The initial value of each layer thickness in the aperiodic multilayer is zero, as can be seen in Figure 4 above. There are several ways to populate the column of layer thicknesses with non-zero values. We'll describe two methods here, a third method in the next section, and a fourth method in §8.7.6.

4.6.1 Methods for Populating an Aperiodic Thickness List (Part 1/3):

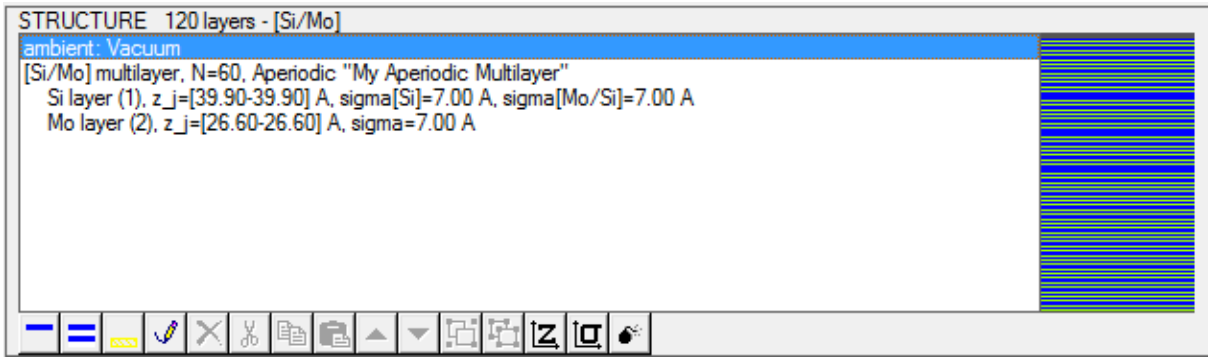
Method 1: The thickness of each layer in an aperiodic multilayer can be specified directly, by manually entering thickness values in the table.

Method 2: If you've already defined the layer group with non-zero layer thicknesses, then you can click the button with the multilayer icon – the one that looks like this: – in the Multilayer window to populate the column of layer thicknesses with the thickness values that were defined prior to checking the Aperiodic button.

For example, suppose you first define a periodic Si/Mo multilayer with N=60, d=66.5 Å, and Γ=0.600, as in the example presented above in §2.3.2. You then check the Aperiodic box in the Multilayer window, and then click the Multilayer button in the same window. Here's what you'll end up with:



And the structure area will look like this (note that I've also entered a Description – "My Aperiodic Multilayer" – in the Multilayer window for this aperiodic multilayer):



Of course this is really still the same periodic structure as before, since all the Si layers so far have the same thickness of 39.90 Å, and all the Mo layers have the same thickness of 26.60 Å. Nevertheless, this second method of populating the aperiodic thickness list with non-zero values is often useful as an initial first step when performing layer-thickness optimizations, as we'll see in Chapter 8. (The starting layer-group structure can be depth-graded too, not just periodic.) The fourth method for generating an initial population of aperiodic layer thicknesses, which is described in §8.7.6 and is based on the formalism developed by Kozhevnikov et al.,[17] is also useful as an initial first step when performing layer-thickness optimizations, as we'll see later.

4.7 Aperiodic Thickness List Files

4.7.1.1 Methods for Populating an Aperiodic Thickness List (Part 2/3):

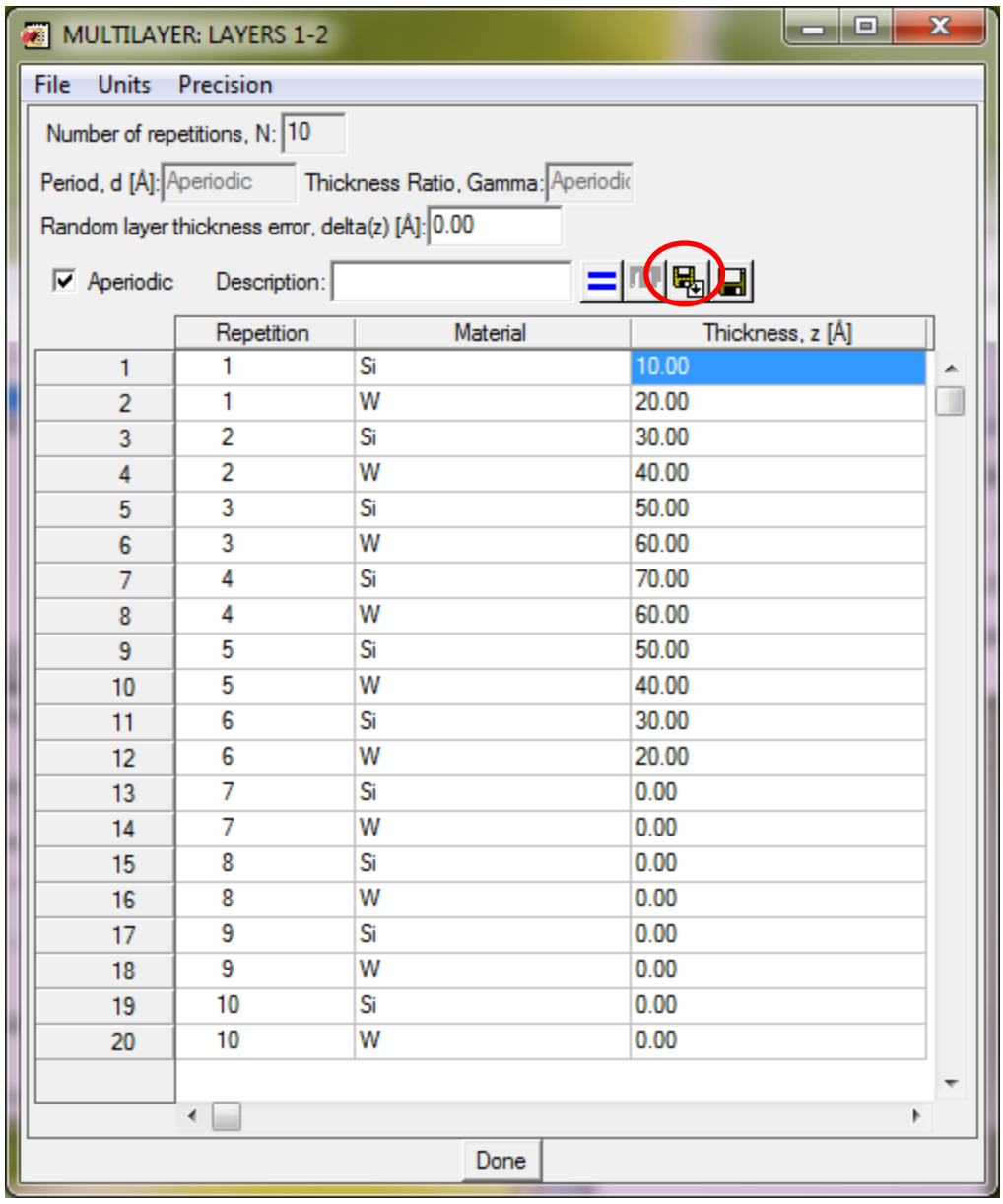
Method 3: The third method for populating the column of aperiodic thicknesses in a Multilayer window when the Aperiodic box is checked is to import a plain text file containing a list of those thicknesses. An IMD "Aperiodic Thickness List" file must contain three columns of ***tab-separated*** data, one line for each layer: index, material, and thickness (in Å), corresponding to the last three columns visible in the Multilayer window above. Some example Aperiodic Thickness List files (with instructions) are included in the `examples/Aperiodic Thickness List` directory in the `imd` installation directory.

Note: while all three columns of data must be present in the file, the first two columns of data (index and material) are not actually used by IMD – the repetition number and material name cannot be changed in the Multilayer window, as explained above in §4.6, and that still applies when importing an Aperiodic Thickness List file. The first two columns in the file are used only as place-holders.

To import an Aperiodic Layer Thickness file into the Multilayer window, click the "Import Thickness List" button – it looks like this:  – and then select the file you want to import. If you have defined fewer

layers in the aperiodic layer group than are listed in the Aperiodic Layer Thickness file, the extra lines in the file will be ignored. Similarly, if you have defined more layers in the aperiodic layer group than are listed in the Aperiodic Layer Thickness file, the thicknesses of the extra layers will not be changed as a result of the import.

Let's look at an example. Suppose we've defined an aperiodic multilayer comprising 10 repetitions of two component layers, Si and W. The aperiodic layer group therefore contains 10x2=20 individual layers. Next, we import an Aperiodic Thickness List file for which only 12 layers are defined (i.e., 6 repetitions of Si/W), as in the example file called `sample_aperiodic_thickness_list.txt` located in the `examples/Aperiodic Thickness List` directory. After importing that file, the Multilayer window looks like this (after resizing to show the entire table):



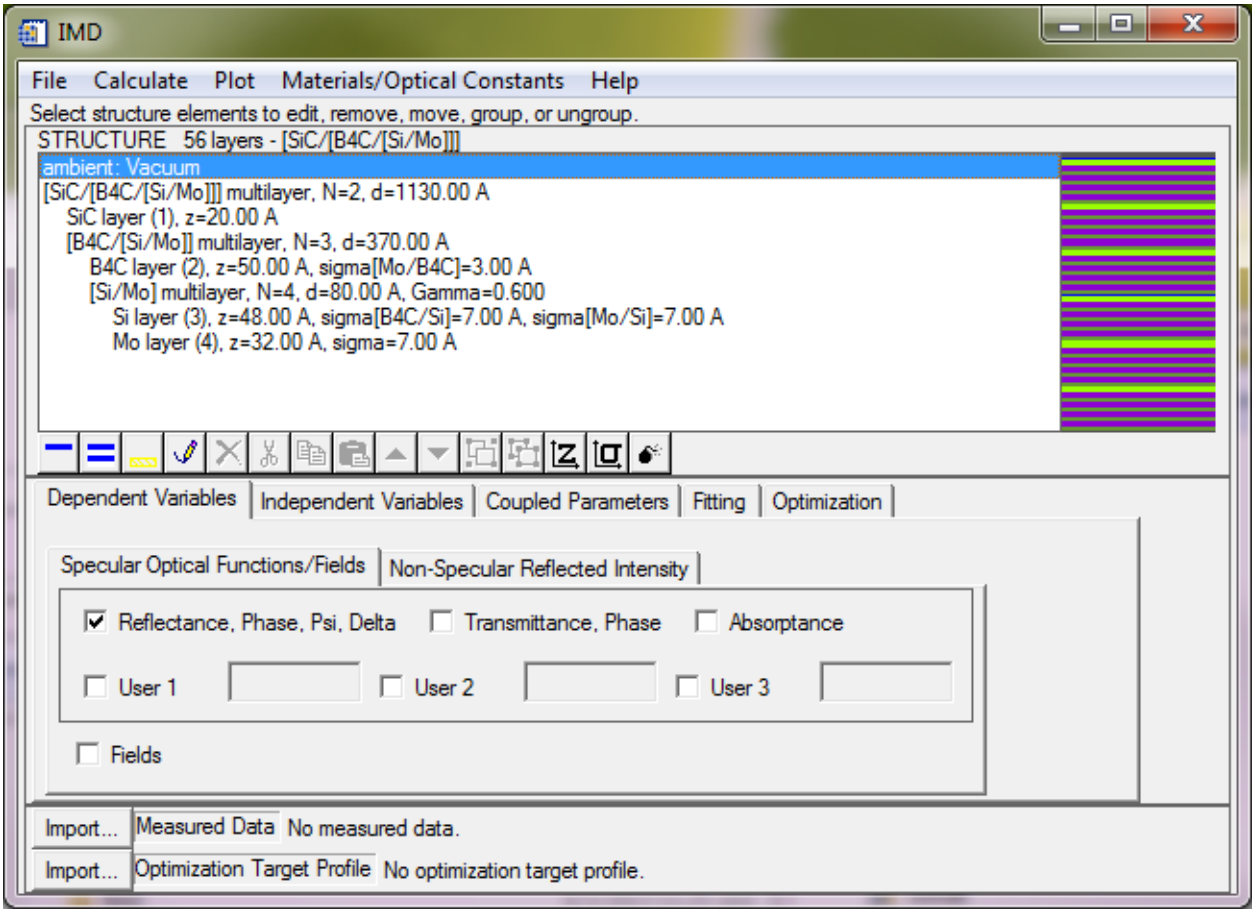
	Repetition	Material	Thickness, z [Å]
1	1	Si	10.00
2	1	W	20.00
3	2	Si	30.00
4	2	W	40.00
5	3	Si	50.00
6	3	W	60.00
7	4	Si	70.00
8	4	W	60.00
9	5	Si	50.00
10	5	W	40.00
11	6	Si	30.00
12	6	W	20.00
13	7	Si	0.00
14	7	W	0.00
15	8	Si	0.00
16	8	W	0.00
17	9	Si	0.00
18	9	W	0.00
19	10	Si	0.00
20	10	W	0.00

Only repetitions 1 through 6 contain non-zero layer thickness values; the remain 4 repetitions are unchanged from their initial values of zero.

Once an aperiodic layer group has been populated with non-zero thicknesses, you might also want to export that list of layer thicknesses to an Aperiodic Thickness List file. To do so, click on the "Export Thickness List" button (next to the "Import" button), which looks like this: . Exported Aperiodic Thickness List data is written in the correct format so that it can be re-imported later.

4.8 Nested Multilayers

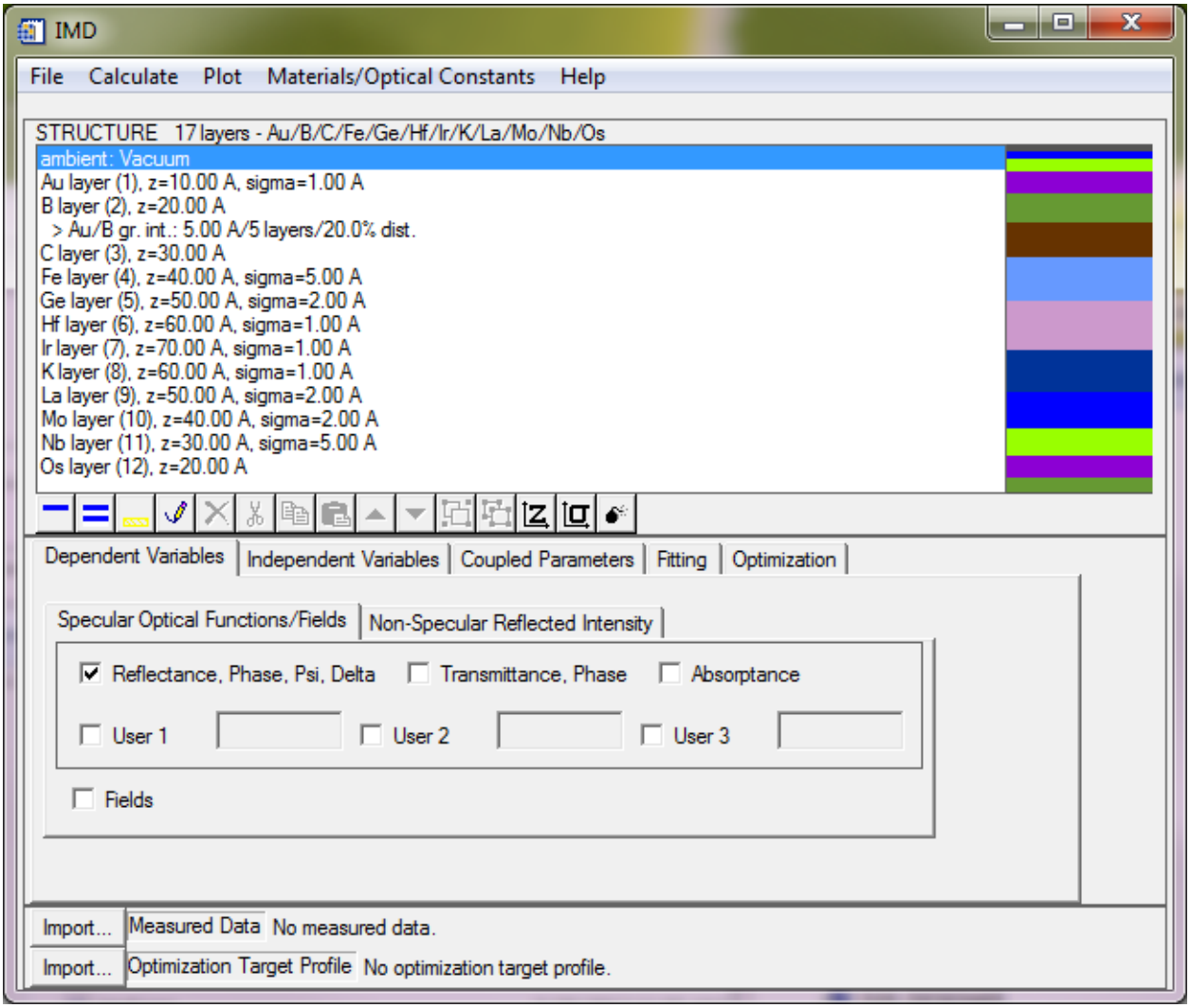
You can use the Group ( ) button located at the bottom of the Structure area on the main IMD window (Figure 3) to form not just groups of individual layers (i.e., multilayers), as we've been discussing in the last few sections, but also layer groups that themselves contain layer groups, nested multilayers, in other words. Here's a crazy example:



4.9 Layer-Data Files

There may be certain occasions where it is cumbersome to use the Add Layer and/or Add Multilayer buttons repeatedly to define the film stack. For example, you might want to model a film stack containing a large number of unique component layers, rather than some repeating sequence of just a few component layers as we've discussed so far in the case of periodic, depth-graded and aperiodic multilayers. In such circumstances it may be more convenient to create a text file in which you specify the layer materials, layer thicknesses and interface properties (discussed in §4.11) for each layer, and then import that text file into IMD. If you choose to use this method, you must first create the text file, and then use "File→Import Layer Data…" from the IMD menu bar.

An IMD "Layer Data" file must be a plain text file containing eight columns of ***tab-separated*** data, one line for each layer: index, material, thickness (in Å), roughness (in Å), interface profile function, graded interface layers, graded interface width, and graded interface distribution. (The five interface parameters will be explained in §4.11.) An example Layer Data file (with instructions) is included in the `examples/Layer Data File` directory in the `imd` installation directory. Loading that example file (`sample_layer_data_file.txt`) into IMD results in the (ridiculous) structure shown here:



You can also use "File→Export Layer Data…" to create Layer Data files once you have a defined film stack in place.

> **Tip**: If you're modeling an aperiodic multilayer, it's far better to use the techniques discussed above in §4.6 rather than to create a Layer Data file with each layer specified as a unique component layer. First, creating a structure with a large number of component layers consumes a large amount of computer resources, which may result in poor performance. Second, if you define a large number of component layers whose thicknesses you intend to adjust by layer-thickness optimization (Chapter 8), you will have to specify each individual layer thickness as an optimization parameter; in contrast, the layers in an aperiodic multilayer (§4.6) can be varied collectively using just one optimization parameter to represent each component layer.

4.10 Structure Diagram and Layer Thickness Profile Plot

The structure diagram located to the right of the structure list on the main IMD window (Figure 3) displays a diagram of the currently defined structure: each component layer is displayed with its own color, and layer thicknesses are drawn to relative scale. Layer colors can be changed in the IMD Preferences window: choose "File→Preferences…" (Layer colors can also be changed in the Layer Thickness Profile and Interface Width Profile plots described below.) There are 8 component layer colors available; if your structure contains more than 8 component layers, the colors will be reused. (i.e., component layer #9 will use layer color #1, etc.)

You can print the structure diagram, or save it to a file so that it can be imported into other documents. For example, you can print to a PostScript (.ps) file, which can then be converted to any number of other graphics file formats using freely available programs (e.g., GhostScript, ImageMagick, etc.), or imported into vector graphics programs (e.g., Corel Draw, Adobe Illustrator, etc.) for conversion and/or further manipulation.

To print the structure diagram, choose "File→Print Structure Diagram…" from the IMD menu bar, or right-click on the structure diagram to bring up the same menu option. To illustrate, here's a better quality picture of the structure diagram for that crazy nested multilayer we discussed in §4.8:

Figure 5. This figure was created by printing the Structure Diagram to a PostScript file, then converting to JPEG (using Corel Draw) in order to import into the Word document used to create the PDF you're currently reading.

You can also view a plot of layer thickness vs. layer index for the currently defined structure. Click the "Plot Layer Thickness Profile" button ( ) located under the structure list in the main IMD window. A new window will appear that looks like this:

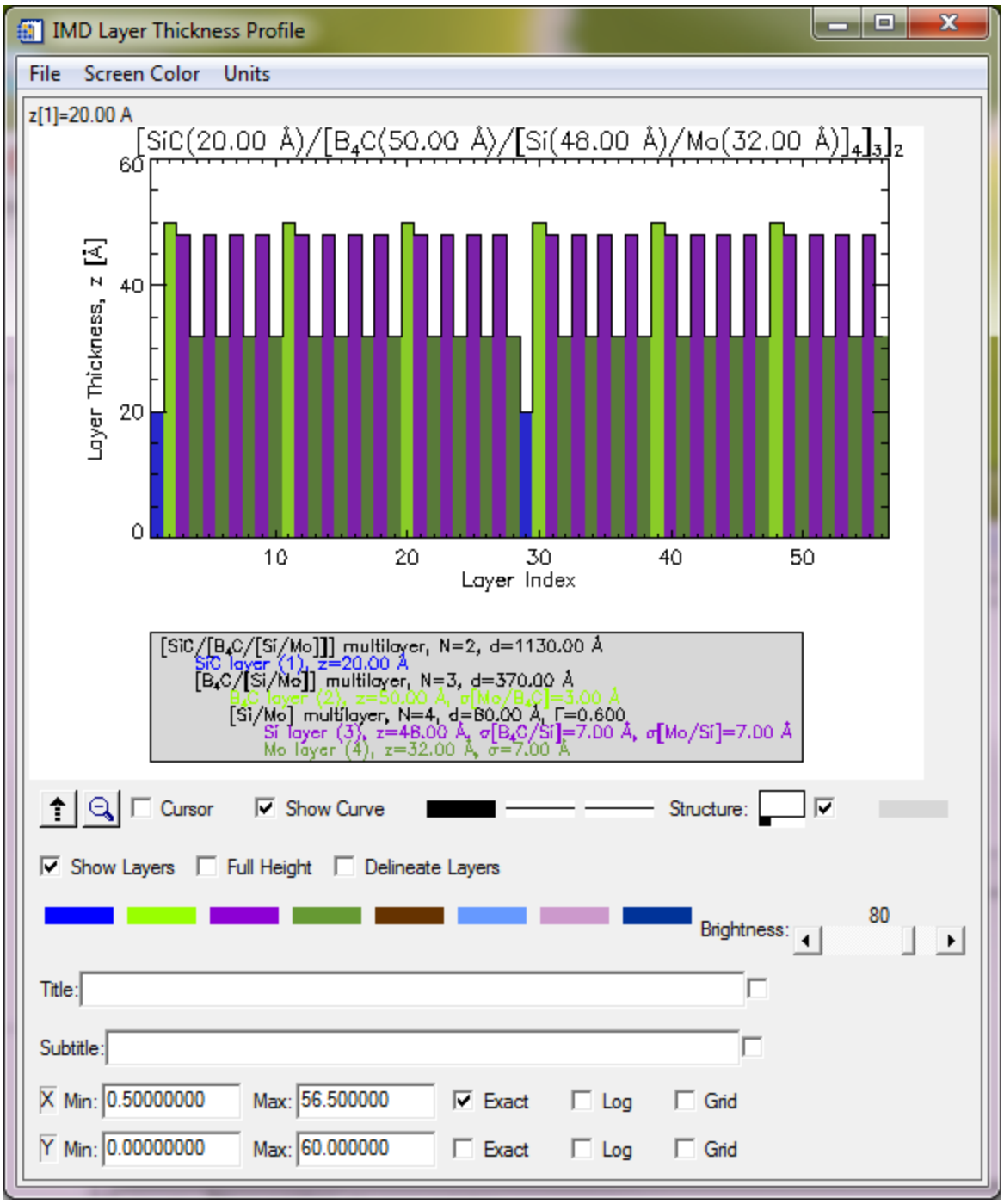


Figure 6. The Layer Thickness Profile window.

By default, each layer component is displayed in this plot (and in the structure label below the plot) with its own color – the same colors as those used in the structure diagram just discussed; these layer colors can be changed directly in the Layer Thickness Profile window shown above, as well as via the IMD Preferences window mentioned previously (i.e., "File→Preferences…"). There are also a variety of other

options to control the appearance of the Layer Thickness Profile plot, as can be seen in Figure 6, and the plot can be printed from the menu bar as well, to create high-quality graphics for presentation, such as this:

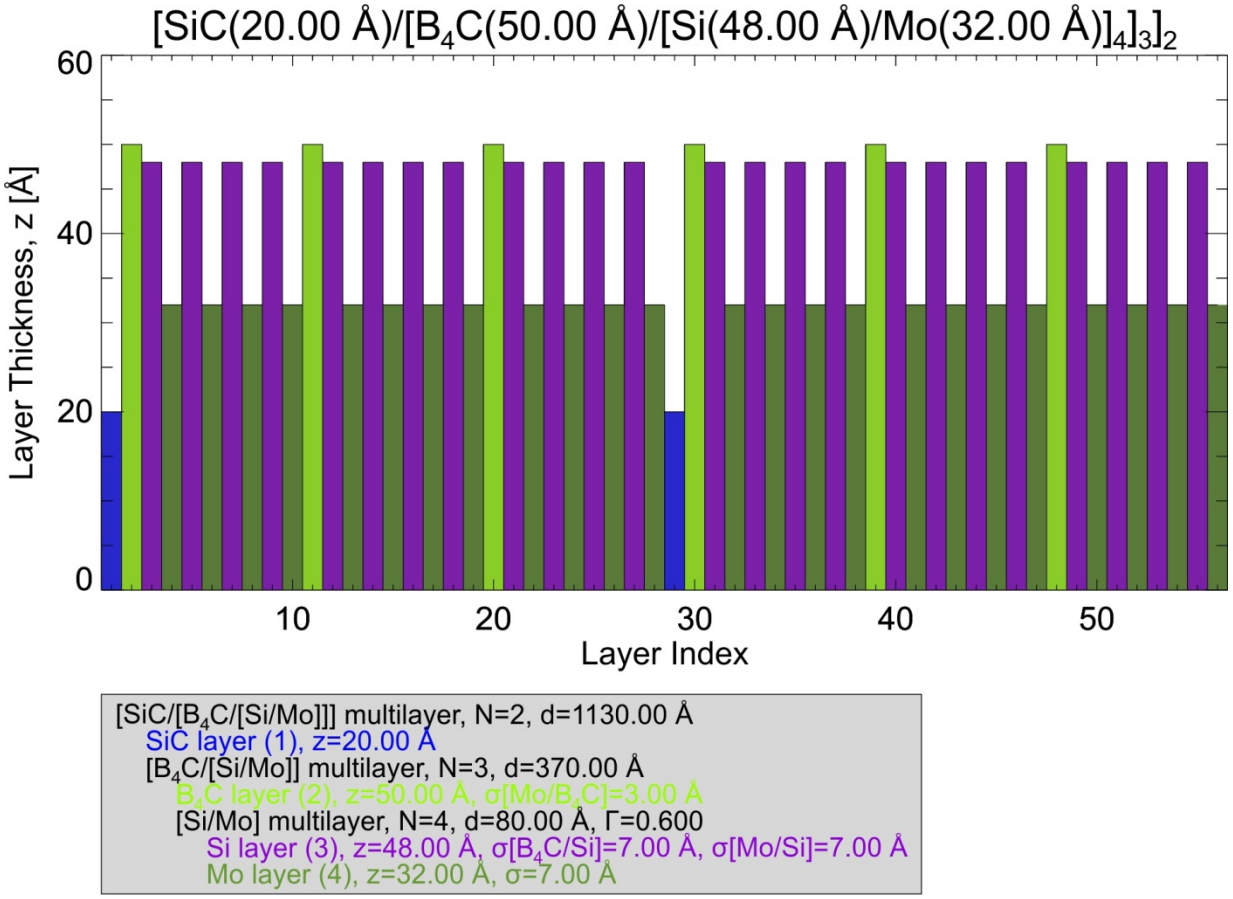


Figure 7. This figure was created by printing ("File→Print…") from the Layer Thickness Profile window (Figure 6) to a PostScript file, then converting that file to JPEG (using Corel Draw) for import into this Word document.

> **Tip:** The Layer Thickness Profile window can remain open as the film stack is changed, or during a fit or optimization (described in Chapter 8), in order to better visualize the film stack as it evolves.

4.11 Surface and Interface Imperfections

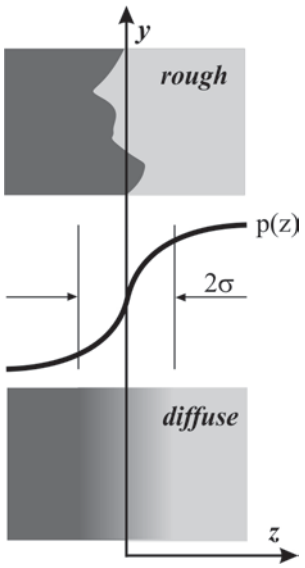
Surface roughness and interface imperfections – interfacial roughness and diffuseness – have the effect of reducing the specular reflectance of a surface or an interface, either by scattering light into non-specular directions in the case of roughness, or by increasing the transmittance of the interface in the case of interfacial diffuseness. Including surface and interface imperfections in your model is absolutely essential for generating accurate results, as in practice, no film stack ever has perfect interfaces.

There are several options available in IMD to account for such imperfections when computing specular and non-specular optical functions, as we now describe.

4.11.1 Modified Fresnel Coefficients

The method used by IMD to compute specular optical functions, explained in more detail in reference [1], makes use of the Fresnel reflection and transmission coefficients computed at each interface in the film stack. To account for the effects of interface imperfections on specular optical functions in IMD, the Fresnel coefficients at each interface in the film stack are modified, in accord with the interface parameters you specify, following the formalism described in reference [6].

In the case of specular optical functions, the effects of interfacial roughness and interfacial diffuseness are indistinguishable: roughness will scatter light into non-specular directions, while diffuseness will increase the transmittance of the interface, yet in the end, both types of imperfections ultimately reduce the specular reflectance of the interface. Consequently, the interface, which is taken to lie in the X-Y plane, can be characterized simply in terms of an 'Interface Profile Function', p(z), and an 'Interface Width', σ, as illustrated schematically here:



In the case of a purely rough interface, the value of the interface width σ is equal to the rms interfacial roughness, σ_r. In the case of a purely diffuse interface, σ refers to the interfacial diffuseness, σ_d. In the general case of an interface that is both rough and diffuse, $\sigma = \sqrt{\sigma_r^2 + \sigma_d^2}$.

4.11.1.1 Interface Profile Functions

The interface profile function, p(z), is defined as the normalized, average value along the z direction of the dielectric function, $\epsilon(\mathbf{x})$ (which is related to the complex index of refraction as $\epsilon = (n + ik)^2$):

$$p(z) = \frac{\iint \epsilon(\mathbf{x})dxdy}{(\epsilon_i - \epsilon_j)\iint dxdy}, \quad \text{where } \epsilon(\mathbf{x}) = \begin{cases} \epsilon_i, & \mathbf{z} \to +\infty \\ \epsilon_j, & \mathbf{z} \to -\infty \end{cases}$$

The resultant loss in reflectance due to interface imperfections is approximated in IMD by multiplying the Fresnel reflection and transmission coefficients at each interface by the function $\tilde{w}(s)$, which is the Fourier transform of w(z)=dp(z)/dz for that interface. Five analytical interface profile functions are available in IMD. These functions, and their corresponding $\tilde{w}(s)$ functions, are summarized in Table 1:

Table 1. Interface profile functions available in IMD.

Interface Profile Function	**p(z)**	$\tilde{w}(s)$
Error Function	$\frac{1}{\sqrt{\pi}}\int_{-\infty}^{z} e^{-t^2/2\sigma^2}dt$	$e^{-s^2\sigma^2/2}$
Exponential	$\begin{cases} \frac{1}{2}e^{\frac{\sqrt{2}z}{\sigma}}, & z \le 0 \\ 1-\frac{1}{2}e^{\frac{\sqrt{2}z}{\sigma}}, & z > 0 \end{cases}$	$\frac{1}{1+s^2\sigma^2/2}$
Linear	$\begin{cases} 0, & z < -\sqrt{3}\sigma \\ \frac{1}{2}+\frac{z}{2\sqrt{3}\sigma}, & \lvert z\rvert \le \sqrt{3}\sigma \\ 1, & z > \sqrt{3}\sigma \end{cases}$	$\frac{\sin(\sqrt{3}\sigma s)}{\sqrt{3}\sigma s}$
Sinusoidal	$\begin{cases} 0, & z < -a\sigma \\ \frac{1}{2}+\frac{1}{2}\sin\left(\frac{\pi z}{2a\sigma}\right), & \lvert z\rvert \le a\sigma \\ 1, & z > a\sigma \end{cases}$ $a = \pi/\sqrt{\pi^2-8}$	$\frac{\pi}{4}\left(\frac{\sin(a\sigma s-\pi/2)}{a\sigma s-\pi/2}+\frac{\sin(a\sigma s+\pi/2)}{a\sigma s+\pi/2}\right)$
Step	$\frac{1}{2}[\delta(z+\sigma)+\delta(z-\sigma)]$	$\cos(\sigma s)$

The interface profile function p(z) and interface width σ can be specified for each unique surface and interface in the structure. The interface profile function is selected from the drop-list in each Layer or

Substrate window (the default interface profile function used by IMD is the Error Function), while the interface width value is entered in the box labeled "Roughness/Diffuseness, sigma:", e.g., as highlighted in this Layer window from our Si/Mo multilayer example of §2.3.2:

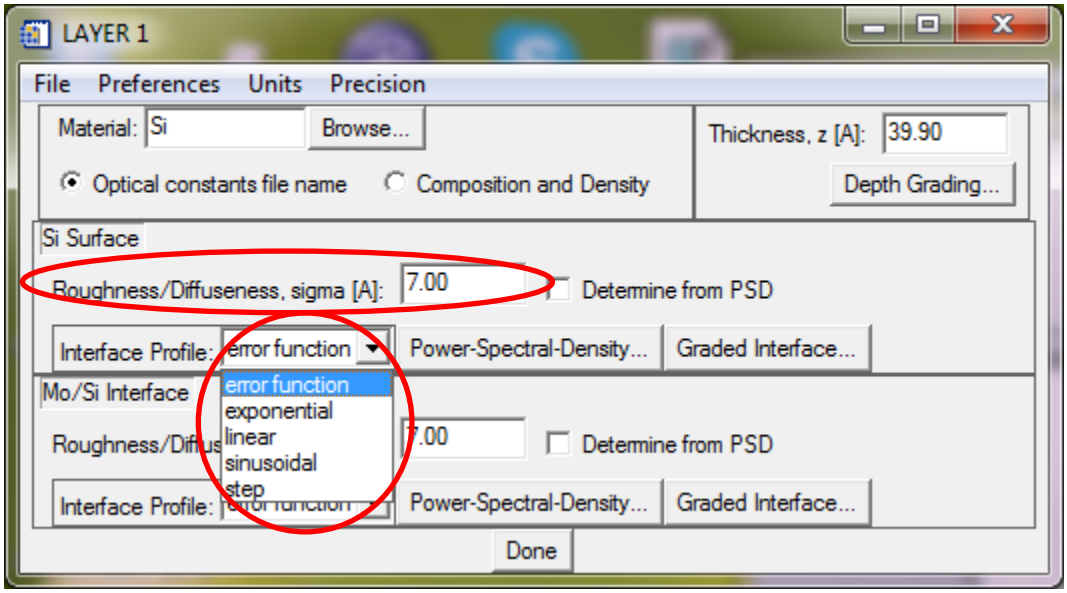


Figure 8. The Layer window for the top-most layer in a layer-group (multilayer).

By default, the interface profile functions that you specify are not displayed in the structure list or in the structure labels included in IMD graphics. To display interface profile functions, use "File→Preferences…" to open the IMD preferences window (§10), navigate to the Interface Profile Function Display tab, and then uncheck the box labeled "Suppress display of interface profile function labels":

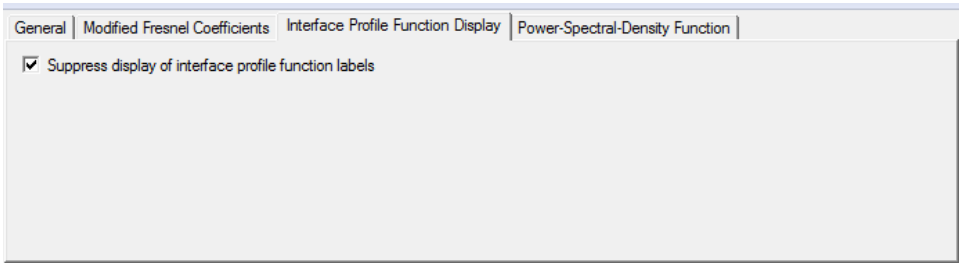


Figure 9. Interface Profile Function Display tab of the IMD Preferences window.

Note, by the way, that the Si layer in this example – which is the top layer of the Si/Mo bilayer we've been discussing – represents two unique interfaces: the Si surface of the top-most layer, and the Mo-on-Si interfaces of the underlying Si layers; the Mo layer of each Si/Mo bilayer has only a Si-on-Mo interface, as illustrated in Figure 11. The Layer window for the Si layer shown above thus provides for two interface profile functions and two interface widths (corresponding to the top-most Si surface and

the underlying Mo/Si interfaces), while the Mo Layer window (Figure 10) includes only one interface profile function and one interface width associated with the Si/Mo interfaces.

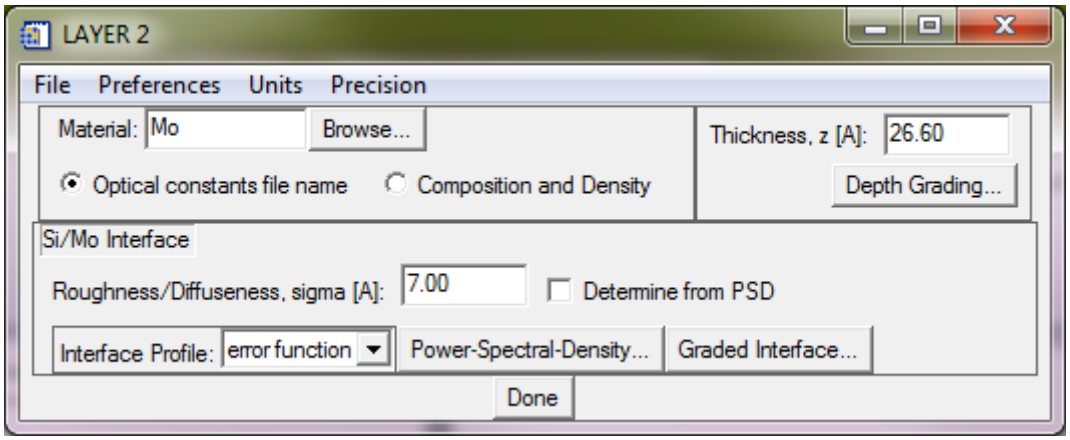


Figure 10. The Layer window for the second component layer in a layer-group.

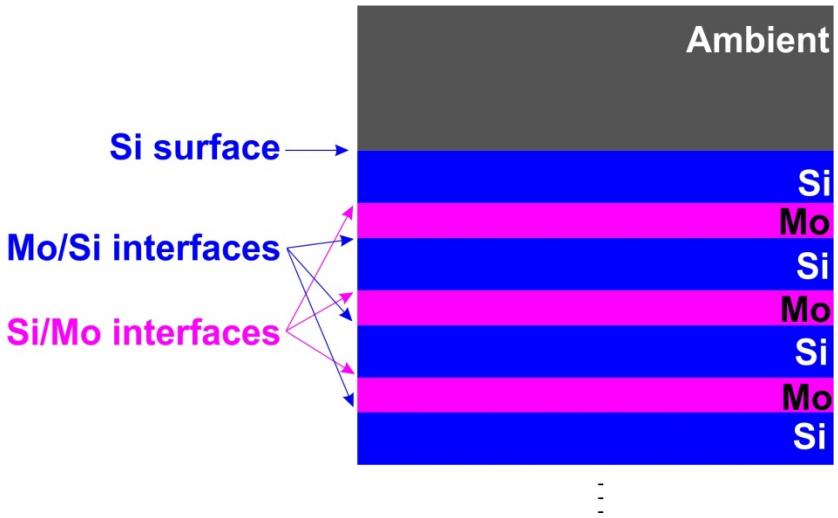


Figure 11. Surfaces and interfaces in a two-component multilayer, in this case comprising Si/Mo bilayers. There are two types of interfaces associated with the top component layer in a layer group (the Si layers, in this example) that can be specified in the corresponding Layer window; the other component layers have only one type of interface.

4.11.1.2 Fresnel Coefficient Modification Options

The Modified Fresnel Coefficient tab on the IMD Preferences window ("File→Preferences…", §10) allows you to specify precisely how you wish to modify the Fresnel coefficients that are used to compute specular optical functions, specular electric fields, and the electric fields that are in turn used to compute non-specular optical functions. This tab is shown in Figure 12:

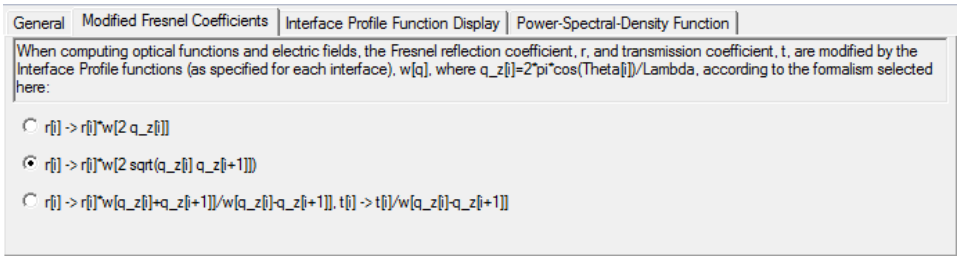


Figure 12. The Modified Fresnel Coefficients tab located on the IMD Preferences window.

There are three options available for Fresnel coefficient modification in IMD:

- In the first option listed in Figure 12, the Fresnel reflection coefficient at the bottom interface[v] of layer i is multiplied by $w(2q_z(i))$, where $q_z(i)=2\pi\cos\theta(i)/\lambda$, $\theta(i)$ is the propagation angle in the layer (determined from the incidence angle through Snell's law,) and λ is the wavelength. The function w corresponds to the interface profile function you specify for the surface or interface, as in Table 1.

- In the second option listed (the default), the reflection coefficient is multiplied by $w(2\sqrt{q_z(i)\times q_z(i+1)})$ (the so-called Névot-Croce factor), in accord with the formalism described in reference [7], which is valid below the critical angle of total external reflection in the X-ray region.

- In the third option listed, both the reflection and the transmission coefficients are modified as indicated in Figure 12, in accord with the formalism described in reference [8].

4.11.2 Power-Spectral-Density Functions

The Power-Spectral-Density (PSD) function, which characterizes the distribution of surface or interface roughness as a function of spatial frequency, is used in IMD for calculations of non-specular optical functions (which are described in §5.1.2). As will be explained below in §4.11.3, the PSD specification for a surface or interface can also be used to derive the interface widths (i.e., σ values) associated with that surface or interface, which are used in specular optical functions computations as described above in §4.11.1.

PSD parameters can be specified for each unique surface and interface in IMD. However, the choice of the PSD functional form is a global setting that applies to all surfaces and interfaces used in the model. Two PSD functional forms are available in IMD, as shown in the "Power-Spectral-Density Function" tab on the IMD Preferences ("File→Preferences…") window (§10):

[v] To be clear: in this document we generally count layers from j=1…n, starting from the top of a film stack containing n layers. But for this discussion of Fresnel coefficient modification only, the index i has a different meaning: here i = j-1 = 0…n-1. Thus, when i=0, the 'bottom interface of layer 0' is actually just the top surface of the film stack (or the surface of the substrate, if no film stack is present).

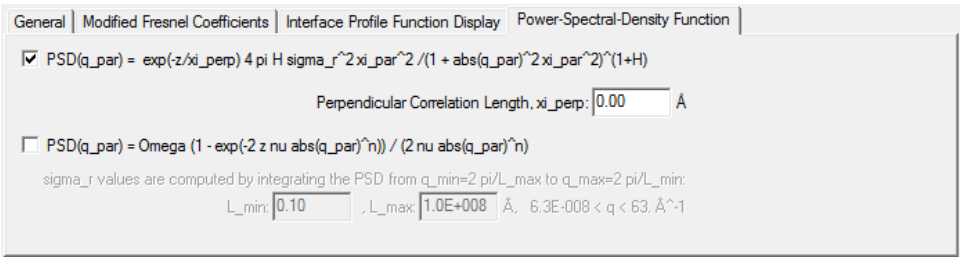


Figure 13. The Power-Spectral-Density tab on the IMD Preferences window.

In the case of the first PSD function shown in Figure 13, which is described in reference [15], each surface or interface is characterized by three parameters: the interfacial roughness, σ_r, the in-plane (parallel) correlation length, $\xi_{\parallel}$, and the so-called jaggedness factor, H. In addition, a (global) vertical or perpendicular correlation length parameter, $\xi_{\perp}$, can be specified on the IMD Preferences PSD tab as well, in order to account for correlated roughness in a film stack. Following the formalism described in reference [13], the vertical correlation function, which describes the correlation between two interfaces i and j, is given by:

$$c_{ji}^{\perp} = \exp\left(-\sum_{n=j_<}^{j_>-1} z_n / \xi_{\perp}\right)$$

where $j_<$=min(j,i), $j_>$=max(j,i) and z_n is the thickness of layer n. The value of the perpendicular correlation length parameter, $\xi_{\perp}$, that is used for computation of non-specular optical functions is specified on the IMD Preferences "Power-Spectral-Density Function" tab shown in Figure 13.

Shown in Figure 14 is the Power-Spectral-Density Function window associated with a layer that is part of a layer-group, when the $\sigma_r/\xi_{\parallel}$/H PSD model just described is selected in the "Power-Spectral-Density Function" tab on the IMD Preferences window. A window like that shown in Figure 14 appears when you click the "Power-Spectral-Density..." button located on a Layer or Substrate window:

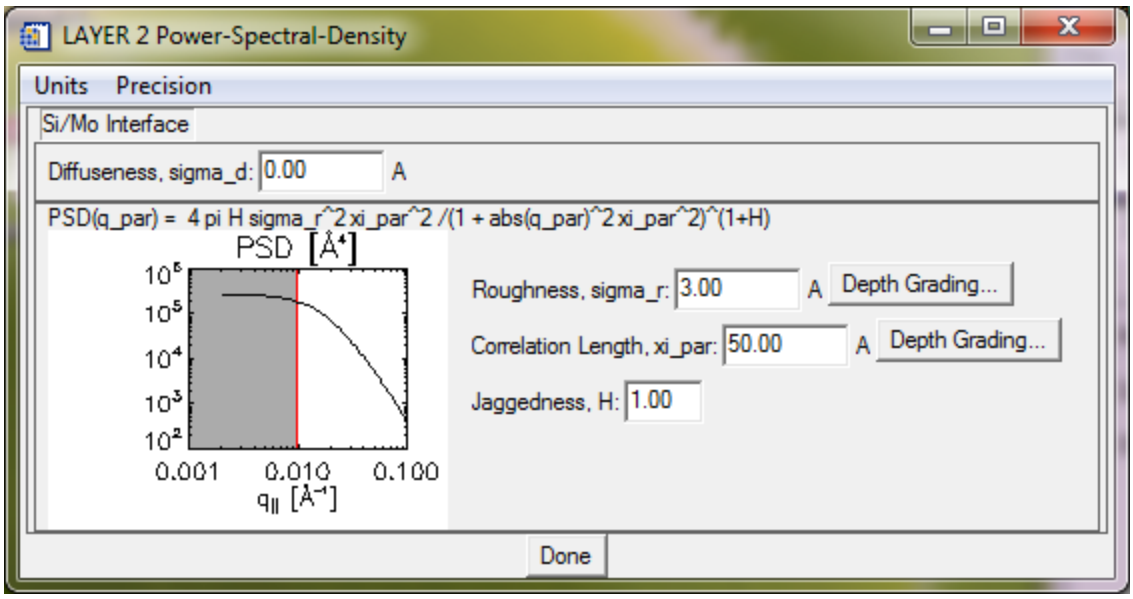


Figure 14. The Power-Spectral-Density window associated with a layer that is part of a layer-group, when the $\sigma_r/\xi_\parallel/H$ PSD model is selected.

A plot of the PSD function for the PSD parameters specified is displayed as a small graph in the PSD window (e.g., Figure 14), over a range of transverse momentum values, $q_\parallel$, that is at least as large as the range of $q_\parallel$ values associated with the incidence and scattering angles that you define for a non-specular optical function computation (described in §5.1.2); the actual range of $q_\parallel$ values that will be used in the computation is indicated by the gray region in the graph.

The second PSD function available in IMD, also shown in Figure 13, is based on the stochastic model of thin film growth and erosion described in reference [14]. In this case, the (intrinsic) PSD at each interface is characterized by the three parameters: Ω, the volume element growth parameter, ν, the relaxation coefficient, and the exponent n; the intrinsic PSD also depends on the layer thickness z. (In the case of the substrate PSD, z does not refer to a real thickness, but is just another parameter that you must specify.) Note that, in accord with this PSD model, which is aiming to capture the evolution of roughness during film growth, the total PSD at a given interface is equal to the intrinsic PSD at that interface (as specified by the $\Omega/\nu/n/z$ parameters for that interface) plus the contribution of the PSD's of all the underlying interfaces; the amount of roughness that is replicated from layer to layer depends on the PSD parameters for each of the interfaces involved.

Figure 15 shows the PSD window for a layer when the $\Omega/\nu/n$ PSD model is selected in the "Power-Spectral-Density Function" tab on the IMD Preferences window:

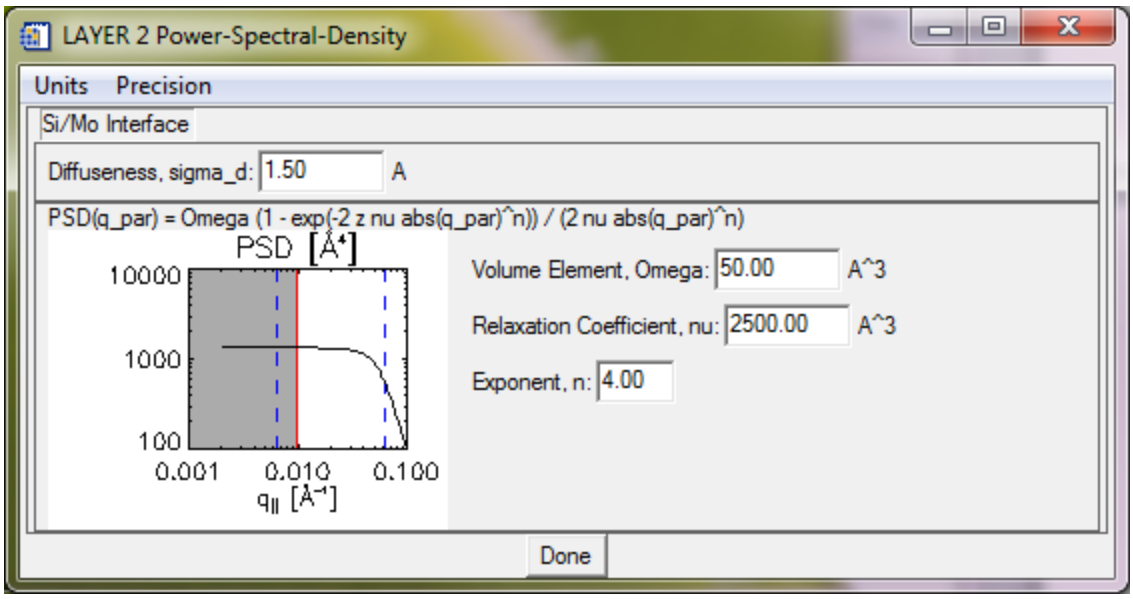


Figure 15. The Power-Spectral-Density window associated with a layer that is part of a layer-group, when the Ω/ν/n PSD model is selected.

As can be seen in Figure 14 and in Figure 15, in addition to specifying the three PSD parameter values for each surface or interface in the model, you can also specify a value of the interface diffuseness, σ_d. If you select to determine interface widths from PSD parameters, as described in the next section, then the value of σ_d that you specify will be used (in conjunction with the other interface profile parameters you specify for that surface or interface) to compute the modified Fresnel coefficients used to calculate the electric fields needed for non-specular reflected intensity calculations. (See references [9–12].)

4.11.3 Using PSD Parameters to Determine Interface Widths

If you have checked the "Determine from PSD" box on a Layer or Substrate window (e.g., Figure 10), then the interface width, σ, for that surface or interface will be determined from $\sigma = \sqrt{\sigma_r^2 + \sigma_d^2}$, and the interface width (σ) so derived will be used to compute modified Fresnel coefficients as described in §4.11.1. The σ_d value is specified directly in the PSD window. The value of σ_r used for the calculation of σ depends on the PSD model you have selected:

- When the $\sigma_r/\xi_\parallel$/H PSD model is selected, the σ_r value used to compute the total σ value is just equal to the value of σ_r that you specify directly on the PSD window.

- When the Ω/ν/n PSD model is selected, σ_r is determined by integrating the PSD function[18] over the range of spatial frequencies *f* that you specify:

$$\sigma_r^2 = 2\pi \int_{f_{min}}^{f_{max}} \mathrm{PSD}(f)\, f\, df$$

where f_{min}=1/L_{max}, f_{max}=1/L_{min}. Global values for L_{min} and L_{max} must be specified in the "Power-Spectral-Density Function" tab on the IMD Preferences window, as shown in Figure 16:

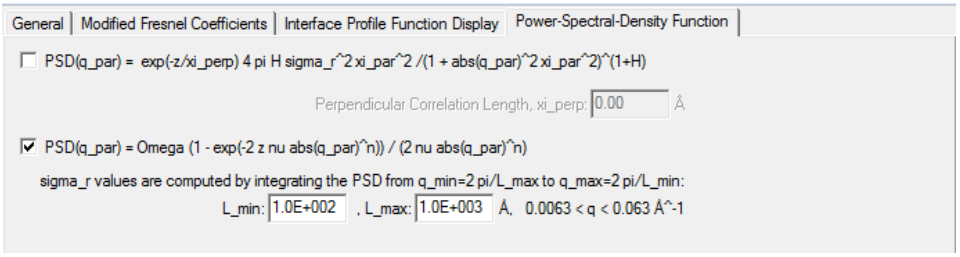


Figure 16. The Power-Spectral-Density tab on the IMD Preferences window, with the Ω/ν/n PSD model selected. The values of L_{min} and L_{max} define the range of spatial frequencies over which the PSD function is integrated to compute the value of σ_r when the "Determine from PSD" box is checked in any Layer or Substrate window.

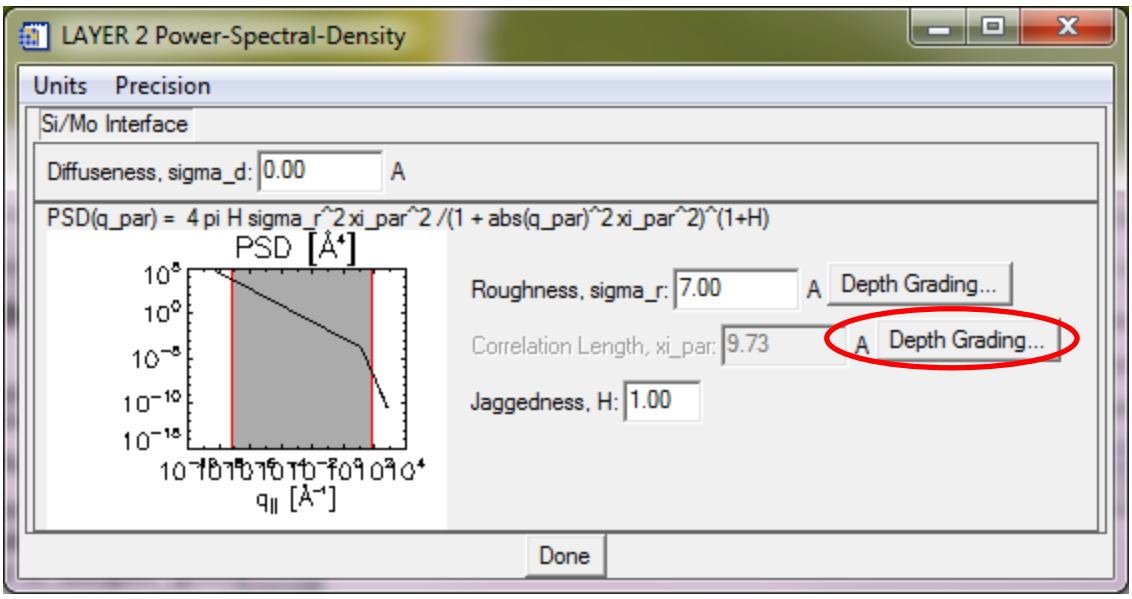
The f_{min} and f_{max} values corresponding to the L_{min} and L_{max} values entered (as in Figure 16) are displayed as dashed blue vertical lines in the PSD plot (e.g., Figure 15), provided that the Ω/ν/n PSD model is selected.

> **Tip:** If you have measured 2D surface profile data for the surface you want to model, you can use in IMD the PSD parameters determined from fitting the radially-averaged PSD using the TOPO software package (or other software). Be sure that there is sufficient overlap between the spatial frequencies sampled by the surface topography measurement used to derived best-fit PSD parameters and the spatial frequencies that impact the IMD computation you are performing.

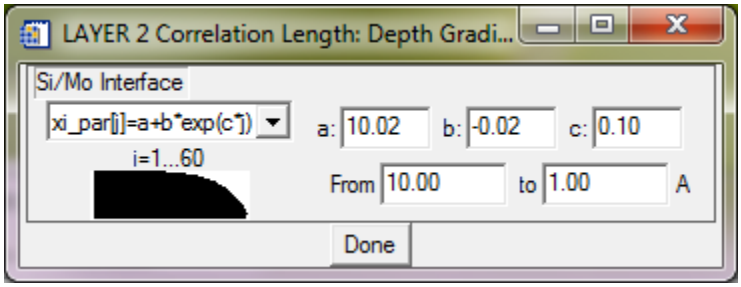
4.11.4 Depth-Graded PSD Parameters

4.11.4.1 Depth-Graded Correlation Length, $\xi_\parallel$

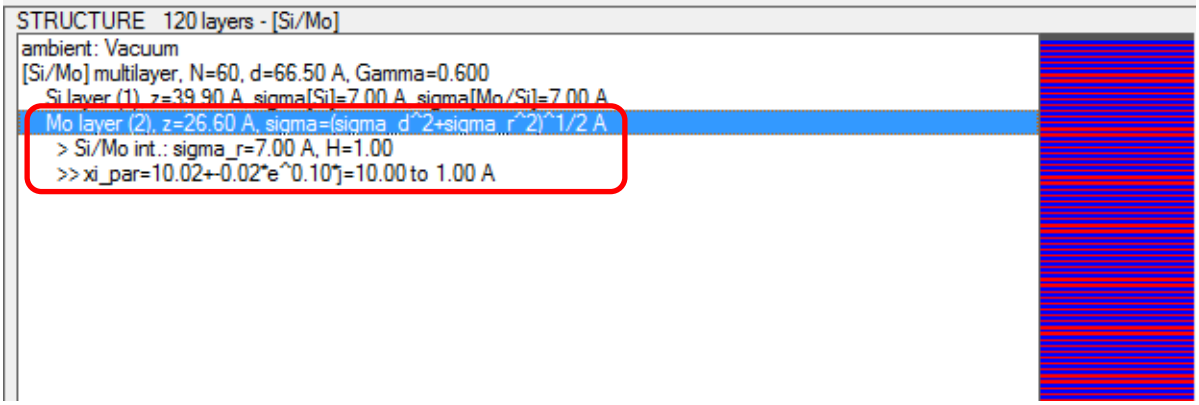
The PSD parameters specified for each interface in a film stack are used in computations of non-specular optical functions. In the case of the $\sigma_r/\xi_\parallel$/H PSD model, the correlation length, $\xi_\parallel$, of a component layer's interface can be made to vary with depth in the film if that layer is part of a layer-group. Click the "Depth Grading…" button next to the box labeled "Correlation length, xi_par" on that layer's PSD window to vary $\xi_\parallel$ with depth:



Enter the desired grading parameters in the new window that pops up when you click the "Depth Grading…" button. As in the case of depth-graded layer thicknesses (§4.5), you can choose from the same four analytic depth-grading functions. In the example below, we have made the $\xi_{\parallel}$ values for the 60 Si-on-Mo interfaces in the Si/Mo multilayer example of §2.3.2 vary exponentially from $\xi_{\parallel}$=1 Å at the bottom of the stack to $\xi_{\parallel}$=10 Å at the top of the stack:



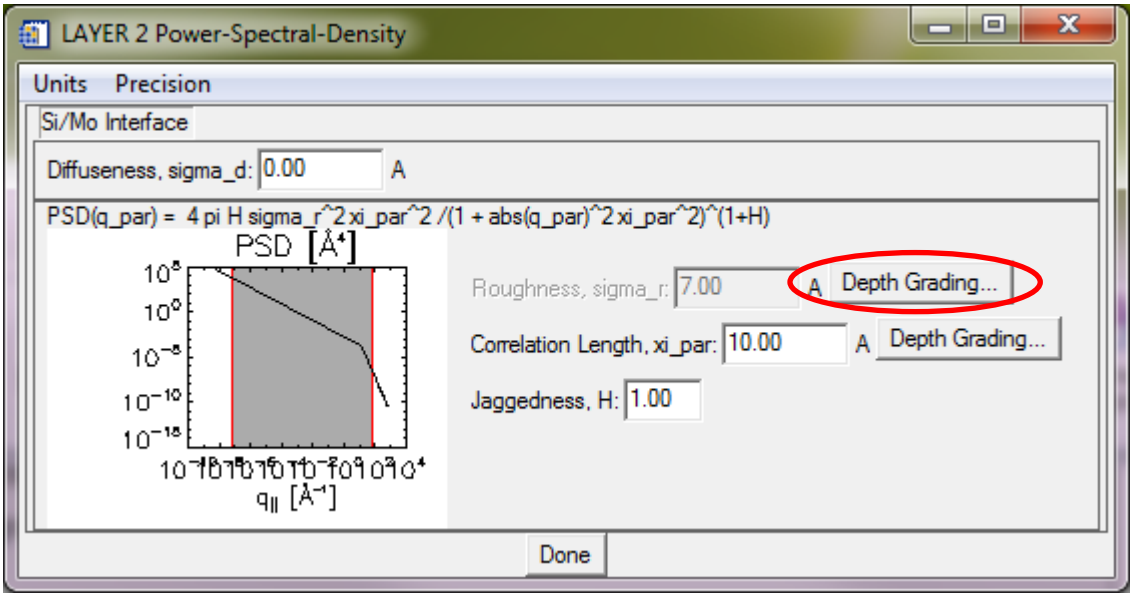
The resulting structure description in the main IMD window looks like this:



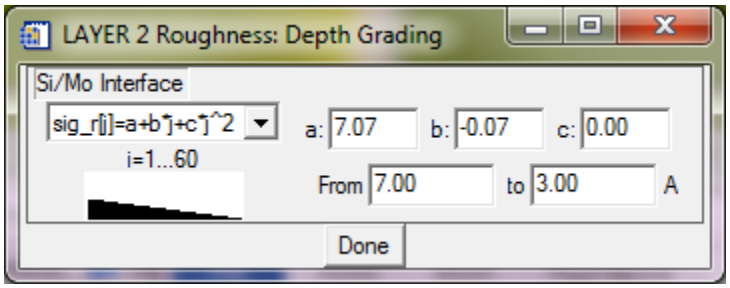
4.11.4.2 Depth-Graded Interface Width, σ

You can also vary the interface width, σ, with film depth in the case of a layer that is part of a layer-group. The variation of σ with film depth (i.e., layer index) depends on the PSD model being used:

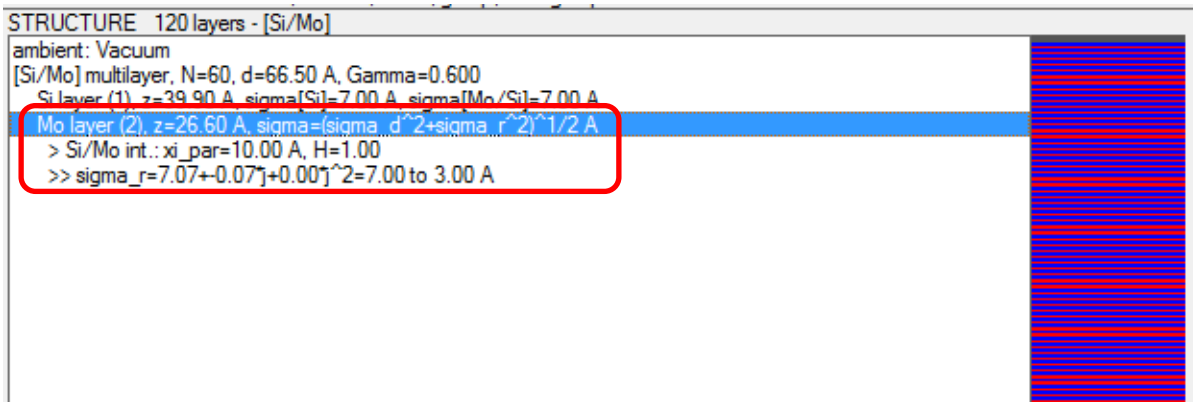
- When the $\sigma_r/\xi_\parallel$/H PSD model is selected, σ_r can be made to vary with depth by direct specification of the associated depth-grading parameters. Click the "Depth Grading…" button next to the box labeled "Roughness, sigma_r" on the layer's PSD window:



Enter the desired grading parameters in the new window that pops up when you click the "Depth Grading…" button. As in the case of depth-graded layer thicknesses (§4.5) and depth-graded $\xi_\parallel$, you can choose from four analytic depth-grading functions. In the example below, we have made the Si-on-Mo interfaces in the Si/Mo multilayer example of §2.3.2 vary linearly from σ_r=3 Å at the bottom of the stack to σ_r=7 Å at the top of the stack:



The resulting structure description in the main IMD window looks like this:



- When the Ω/ν/n PSD model is selected, the σ_r value for each interface in the stack is determined by integrating the PSD for that interface, as explained in §4.11.3. Intrinsic to the Ω/ν/n PSD model is the evolution of the PSD (and thus the interfacial roughness) as the film is grown from bottom to top, as explained in reference [14]. Thus, the variation in σ_r with depth in the film is intrinsic to the model, and determined at each interface by the Ω/ν/n values specified for that interface, along with the PSD values for all the other interfaces that lie below it in the stack.

4.11.5 Interface Width Profile Plot

To display a plot of interface width, σ, as a function of layer index, click the Plot Interface Width button on the main IMD window, which looks like this: . A new Interface Width Profile window will appear: it looks just like the Layer Thickness Profile window shown in Figure 6, except in this case the dependent variable being plotted is the interface width σ.

An example Interface Width plot is shown in Figure 17: we've taken that same Si/Mo multilayer we've been discussing, checked the "Determine from PSD" box for the Mo layer, and using the Ω/ν/n PSD model, set the PSD parameters for the Si-on-Mo interface to Ω=50 Å^3, ν=10 Å^3, and n=4 (we've kept the Si interfaces constant for now):

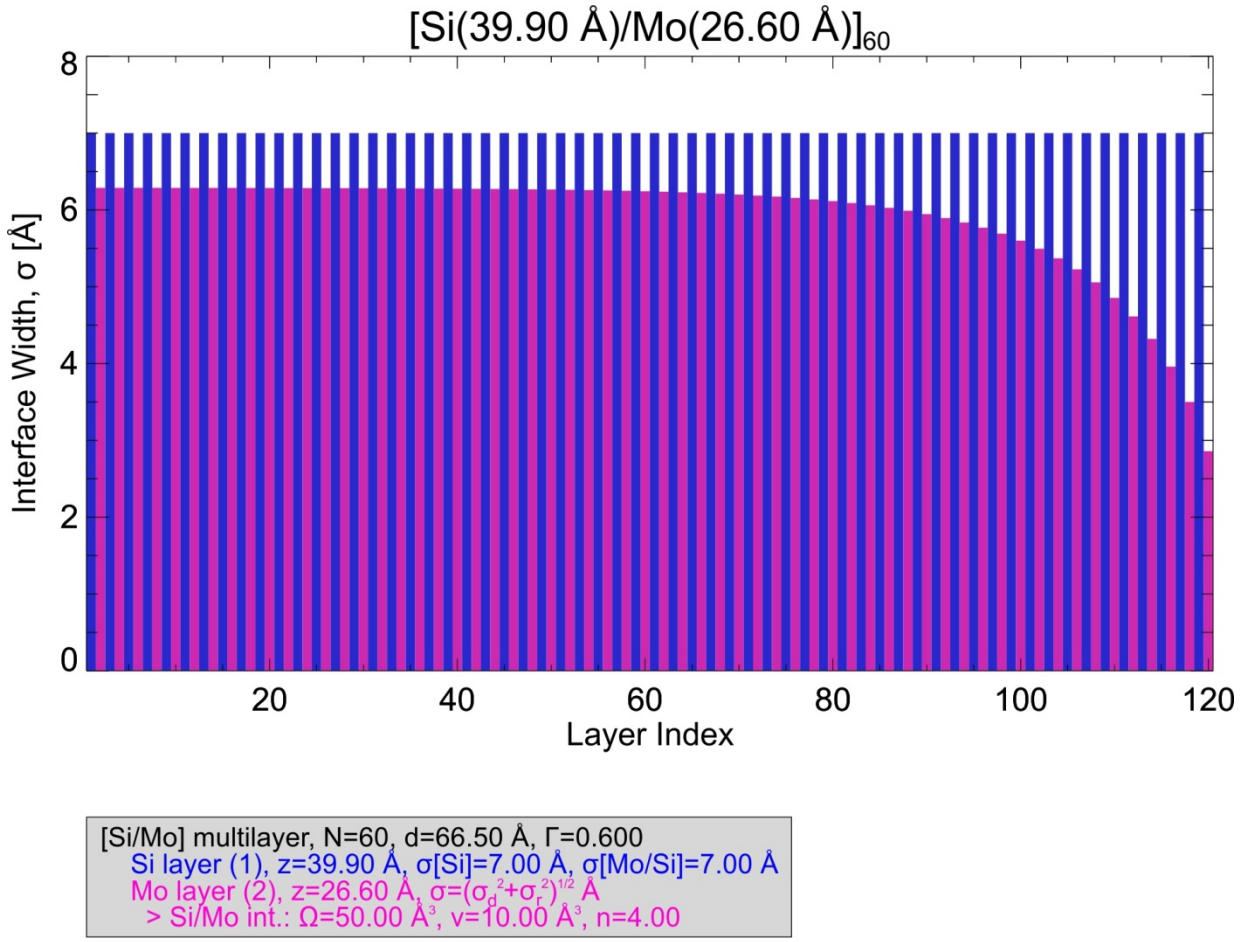


Figure 17. Plot of Interface Width, σ, vs. Layer Index for a periodic Si/Mo multilayer where the Si-on-Mo interfaces vary with depth in the film as determined from the PSD parameters specified in the case of the Ω/ν/n PSD model.

4.11.6 Graded Interfaces

We've already discussed one method available in IMD to approximate the effects of interfacial roughness and interfacial diffuseness that uses the modified Fresnel coefficient formalism, as described above in §4.11.1. Another method that is available in IMD to approximate the effects of interfacial diffuseness in particular is the use of so-called 'graded interfaces'. As shown schematically in Figure 18, when graded interfaces are specified at the interface between layer j-1 and layer j, the formerly abrupt interface is replaced by one or more "interlayers"; the optical constants of the interlayers are *interpolated* from the optical constants of the pure materials on either side of the interface. The number of interlayers, their thickness, and their relative position to the original interface, are determined by parameters that you specify.

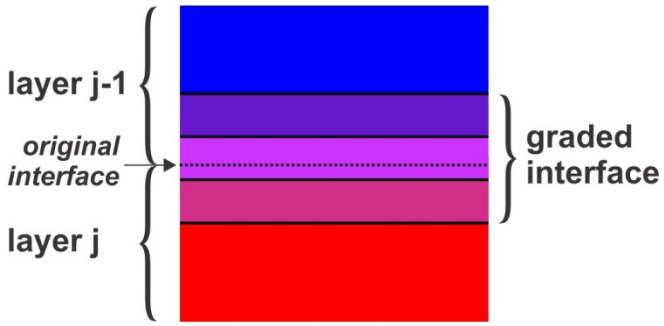


Figure 18. The graded interface concept. In this case the abrupt interface between layer j-1 and layer j is replaced by a graded interface comprising three layers.

A graded interface in IMD is described be three parameters: N_g, w_g, and X_g. N_g is the number of inter-layers comprising the graded interface, and w_g is the total width of the graded interface; as illustrated in Figure 19, each of the N_g inter-layers has width w_g/N_g. The third parameter, X_g, is the distribution factor, which determines the location of the graded interface relative to the original abrupt interface. That is, when a graded interface is specified between two layers, the total thickness of the two layers including the graded interface does not change – the thickness of each layer is reduced in order to keep the total thickness of the bilayer constant. The relative reduction in thickness of each of the two layers is determined by X_g, which can range in value from 0 to 100%. When a graded interface is specified for layer j, with X_g=100% the thickness of layer j is reduced by w_g – i.e., 100% of the graded interface is contained in layer j; when X_g=0%, the thickness of the layer j-1 (which resides above the graded interface) is reduced by w_g – i.e., 0% of the graded interface is contained in layer j; and when X_g is somewhere between 0 and 100%, then the thicknesses of both layer j and layer j-1 are reduced accordingly.

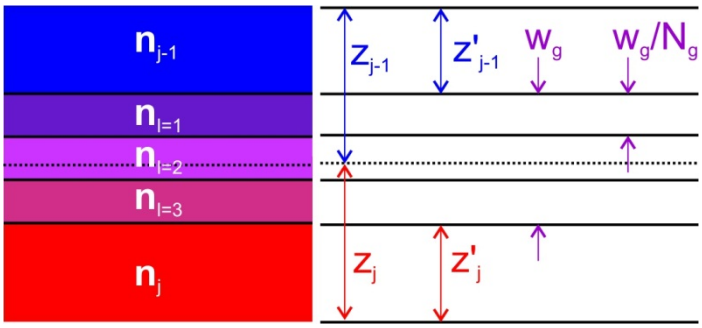


Figure 19. Graded interface parameters. In this example, N_g=3.

If the original thickness of layer j is z_j, and the original thickness of layer j-1 is z_{j-1}, then after adding a graded interface, these layer thickness will be reduced according to:

$$z'_{j-1} = z_{j-1} - w_g(1 - X_g/100.)$$

and

$$z_j' = z_j - w_g X_g/100.$$

with $0\% \le X_g \le 100\%$.

The optical constants $\mathbf{n}_l = n_l + ik_l$ of each of the N_g inter-layers that comprise the graded interface are determined by simple interpolation from the optical constants of layer j, $\mathbf{n}_j$, and the optical constants of layer j-1, $\mathbf{n}_{j-1}$:

$$n_l = \frac{(N_g + 1 - l) \times n_{j-1} + l \times n_j}{(N_g + 1)}$$

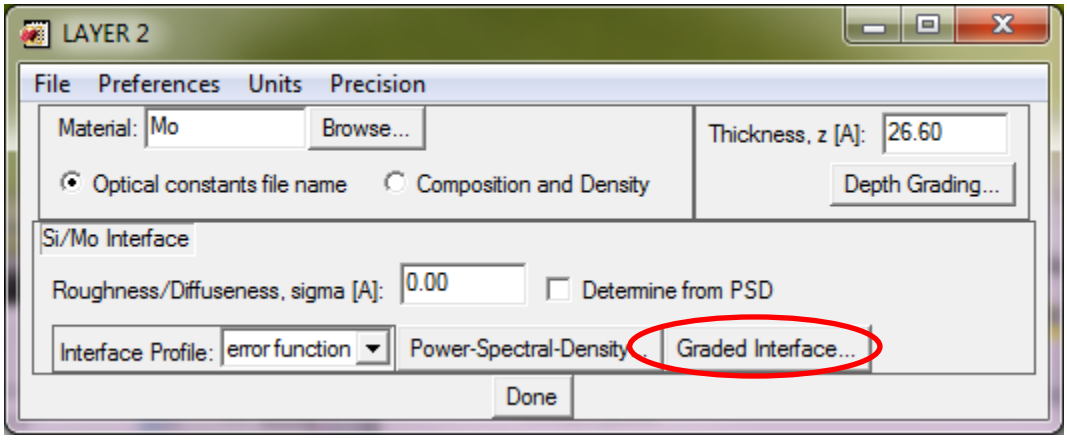
and

$$k_l = \frac{(N_g + 1 - l) \times k_{j-1} + l \times k}{(N_g + 1)}$$

with l=1,…N_g.

> **Note**: The validity of the Graded Interface method for computing the optical constants of the interlayers in a diffuse interface is untested (to say the least). Use at your own risk!

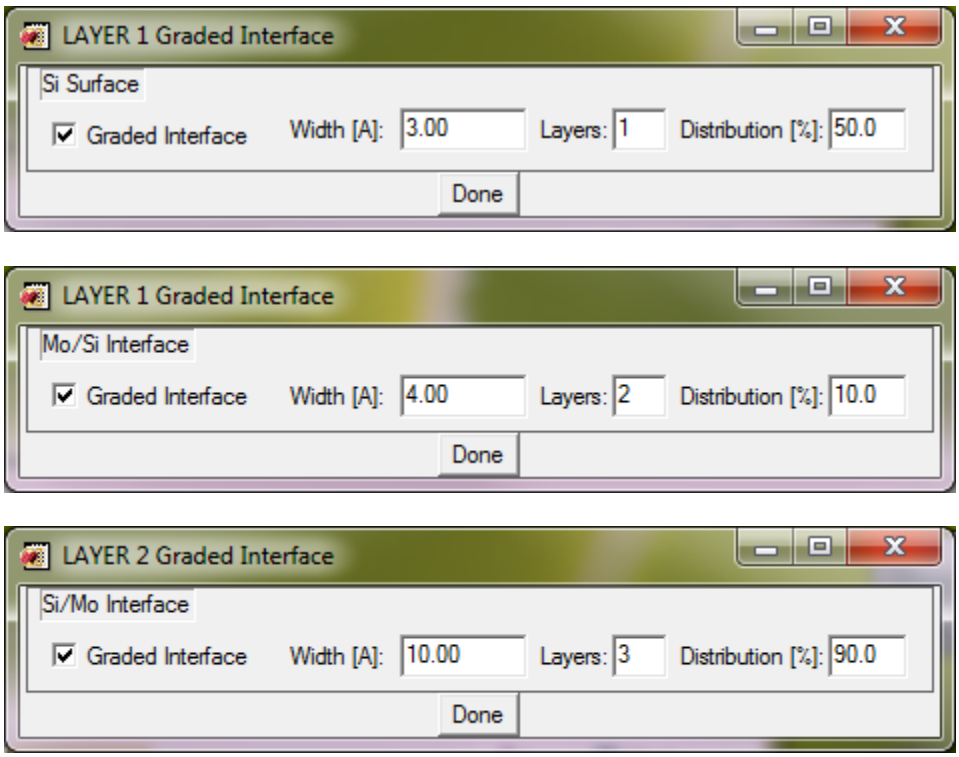
To add a Graded Interface, click the "Graded Interface…" button in the associated Layer window:



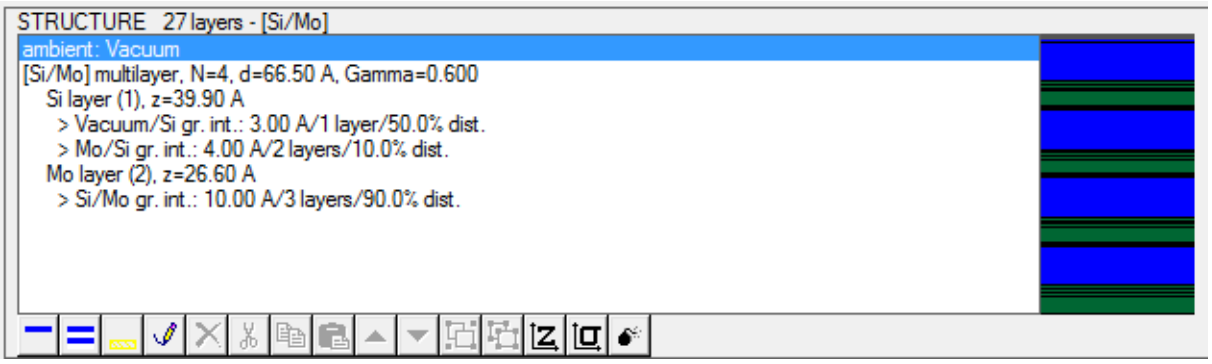
Then, check the "Graded Interface" box in the pop-up window that appears, and enter values for w_g ("Width"), N_g ("Layers") and X_g ("Distribution").

In the example below we've added graded interfaces in a periodic Si/Mo multilayer at each of the three types of interfaces:[vi] at the Vacuum-Si interface we've specified w_g=3 Å, N_g=1, and X_g=50%; at the Si-Mo interface w_g=4 Å, N_g=2 and X_g=10%; and at the Mo-Si, w_g=10 Å, N_g=3, and X_g=90%:

[vi] These graded interface parameters are for illustration purposes only, and are not meant to represent what's actually going on at the interfaces in a Si/Mo multilayer.



The graded interface parameters are listed in the Structure area, and when any graded interfaces are present, the structure diagram is drawn with black lines to delineate the interfaces between layers, including the interlayers. Note that including graded interfaces increases the total number of layers from 8 to 27 in this case, as the interlayers are now counted as distinct layers in the film stack when computing optical functions:



A better view of the structure diagram for this example, with and without graded interfaces, is shown in Figure 20; a plot of Layer Thickness vs. Layer Index is shown in Figure 21.

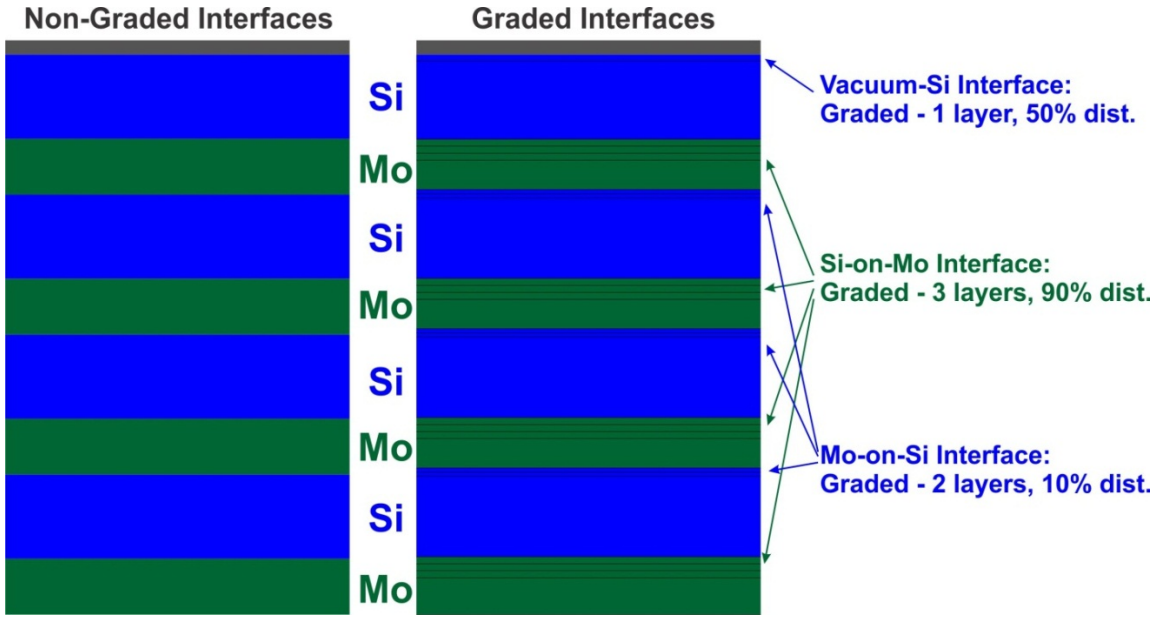


Figure 20. Structure diagram for a Si/Mo multilayer (N=4) before (left) and after (right) adding graded interfaces.

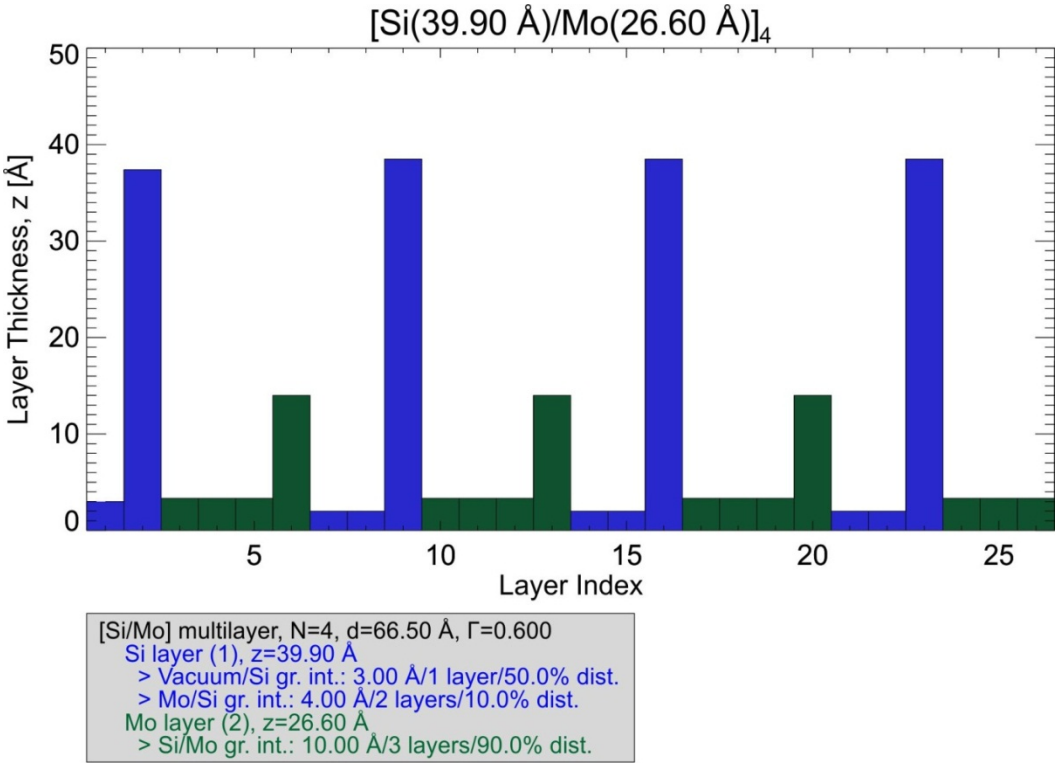


Figure 21. Layer Thickness Profile plot for the Si/Mo multilayer in this example after adding graded interfaces.

Note: When graded interfaces are specified for a given interface, any interface widths and/or PSD parameters that are also specified for that interface are applied to the top interface of each of the interlayers that comprise the graded interface.

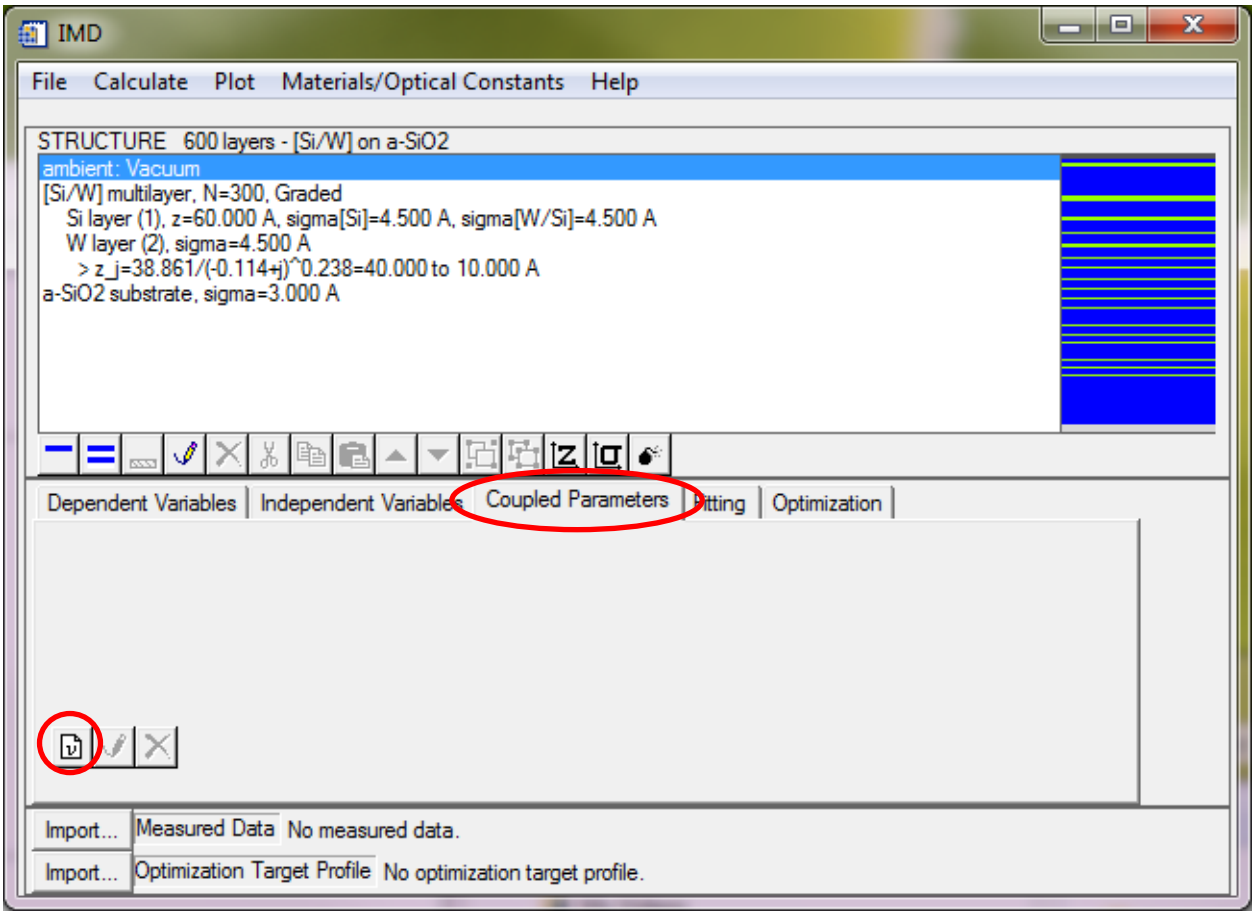
4.12 Coupled Parameters

We've seen in the preceding sections of this chapter how to specify all the parameters that define the structure being modeled in IMD, including optical constants, layer thicknesses, and interface and PSD parameters. In some situations you may want to constrain some of these parameters so that they maintain fixed relationships with other parameters of the same type in the model. For example, we've seen in §4.5 how to create a multilayer where one of the component layer thicknesses varies analytically with depth in the film. But typically *all* the component layers are varied with depth in order to achieve the desired performance in such depth-graded multilayers. One way to do this in IMD, and to maintain a fixed relationship between two parameters of the same type in general, is to use so-called Coupled Parameters. We'll illustrate the Coupled Parameters concept with two specific examples. The `.imd` files for these two examples are located in the `examples/Coupled Parameters` directory, if you want to follow along.

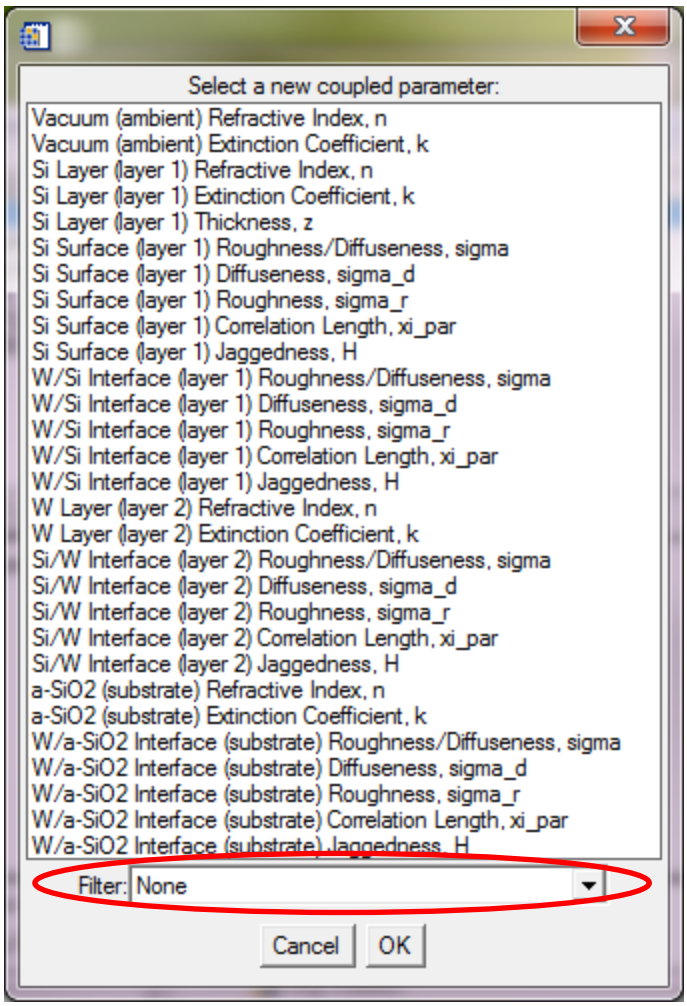
4.12.1 Example: Depth-Graded Si/W Multilayer

Our first example illustrating the use of Coupled Parameters involves a depth-graded Si/W multilayer meant to provide high reflectance over a broad range of X-ray energies at grazing incidence. The figure below shows the main IMD window after we've created a Si/W multilayer having N=300 repetitions, where we've varied the thickness of the W layer with depth in the film stack, from 40 Å at the top to 10 Å at the bottom, with a power-law distribution of layer thickness following the procedures explained in preceding sections. But the Si layer thickness so far remains fixed at 60 Å in this multilayer. We'll now create a Coupled Parameter to maintain a constant ratio between the Si and W layers that form each bilayer in the multilayer, so that both layers will vary with depth in the film following the same power-law functional form.

We first navigate to the Coupled Parameters tab on the main IMD window, by clicking on that tab. We then add a new Coupled Parameter by clicking on the "Add Coupled Parameter" button near the bottom of the tab, which looks like this –  :

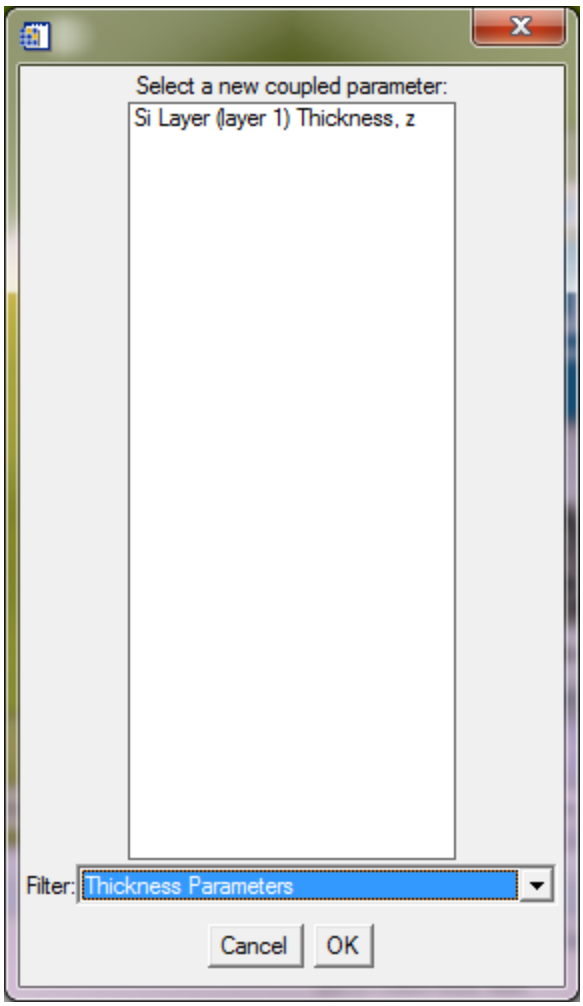


After clicking the "Add Coupled Parameter" button, we're presented with a new window showing the list of available parameters that can be coupled:

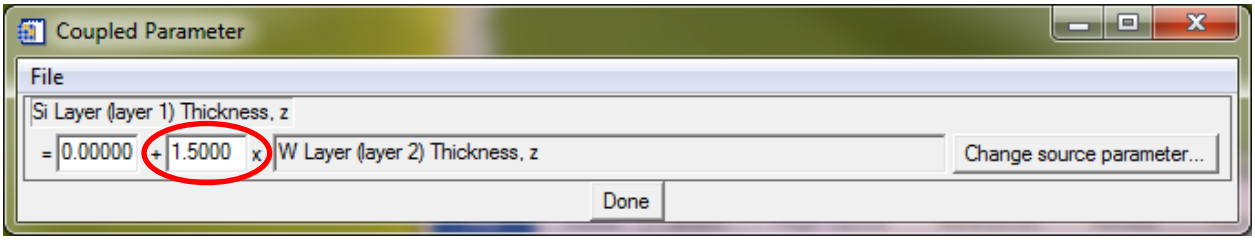


The specific parameters available for coupling at any given moment depends on how the structure is currently defined, on what variables we've already defined (as will be explained Chapter 5), and on any fitting or optimization parameters (Chapter 8) that may already be defined as well. In any case, the list of available parameters is often quite long, but you can use the "Filter" drop-list at the bottom of the Coupled Parameter selection window shown above to limit the choices by "type", as desired, e.g., optical constants, PSD parameters, layer thicknesses, etc.

In this specific example, we're coupling two layer thicknesses, so we select "Thickness Parameters" from the Filter drop-list above to filter the selection, which leaves us but with one choice – the Si layer thickness (since the W layer thickness is already defined to be depth-graded, it is unavailable to be coupled):



We select "Si Layer (layer 1) Thickness, z" from the filtered list shown above. The parameter selected from the list can be coupled only with another available "source" parameter of the same type. For example, thicknesses can only be coupled to other thicknesses, interface widths to other interface widths, and so forth. But in this case, as there is only one possible source parameter – the W layer thickness – the parameter that we've selected from the list (Si layer thickness) is automatically coupled to that sole source parameter. (If more than one eligible source parameter is available, you'll be offered a choice.) The resulting Coupled Parameter window appears, displaying the relationship between the Si layer thickness and the W layer thickness (source parameter):



The coupled parameter will be added to the list of coupled parameters that are shown on the Coupled Parameters tab. Coupled Parameters windows for any coupled parameters listed can be re-opened by double-clicking. You can change the offset and scale coefficients, and you can use the button labeled "Change source parameter" to designate a different source parameter than originally selected.

> **Note**: Once a parameter is selected as a Coupled Parameter, it cannot be selected as an Independent Variable (Chapter 5) or as a Fitting or Optimization Parameter (Chapter 8), and vice versa. However, *source* parameters used in Coupled Parameters *can* be selected as Independent Variables or as Fitting/Optimization Parameters.

For this example, we want to maintain a constant layer thickness ratio of Γ=0.6 throughout the depth-graded multilayer. As $\Gamma \equiv z_{Si}/(z_{Si}+z_W)$, we want the fixed relationship between z_{Si} and z_W to be $z_{Si}=\Gamma/(1-\Gamma)$ x z_W = 1.5 x z_W, and so we've entered a value of 1.5 in the corresponding box above. As a result, the Si layer is now depth-graded in proportion to the W layer, as can be seen in the Structure list shown in Figure 22, and in the Layer Thickness Profile shown in Figure 23. The grazing incidence reflectance vs energy of this depth-graded multilayer example is shown in Figure 24.

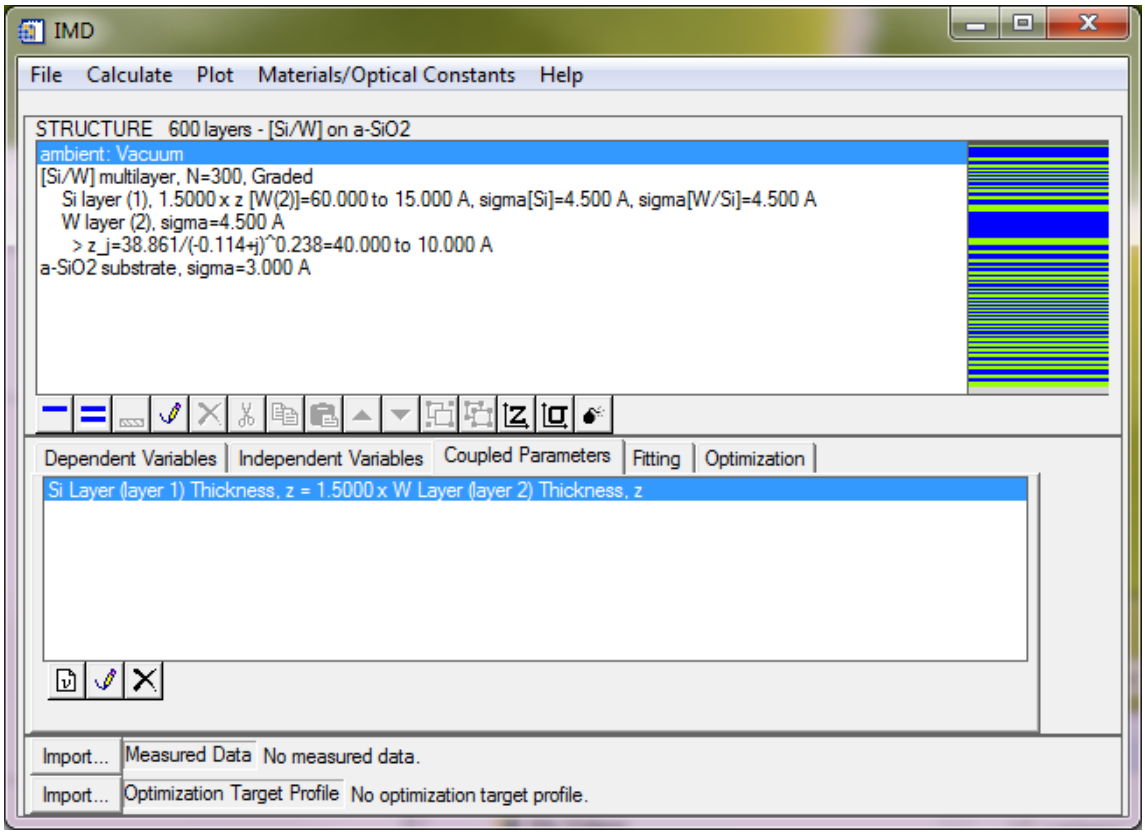


Figure 22. Main IMD window after defining the Coupled Parameter above that fixes the relative thickness of the Si layers in proportion to the W layer thicknesses. (Note that because of the large number of layers in this structure, the structure diagram on the right is distorted due to aliasing.)

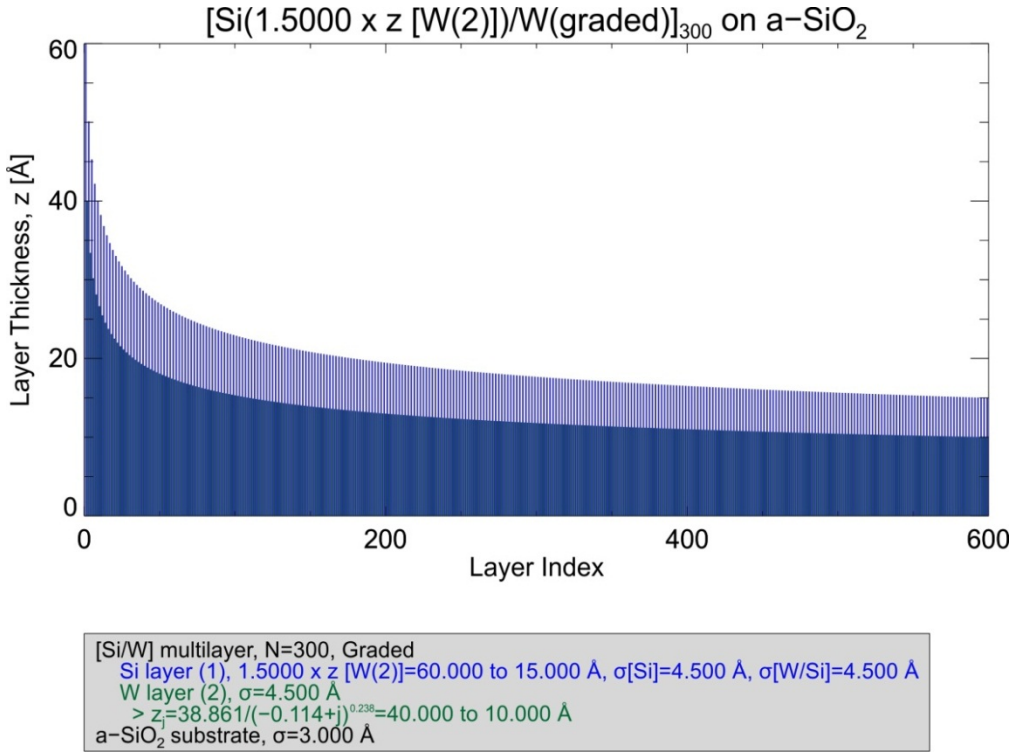


Figure 23. Layer Thickness Profile for the depth-graded Si/W multilayer.

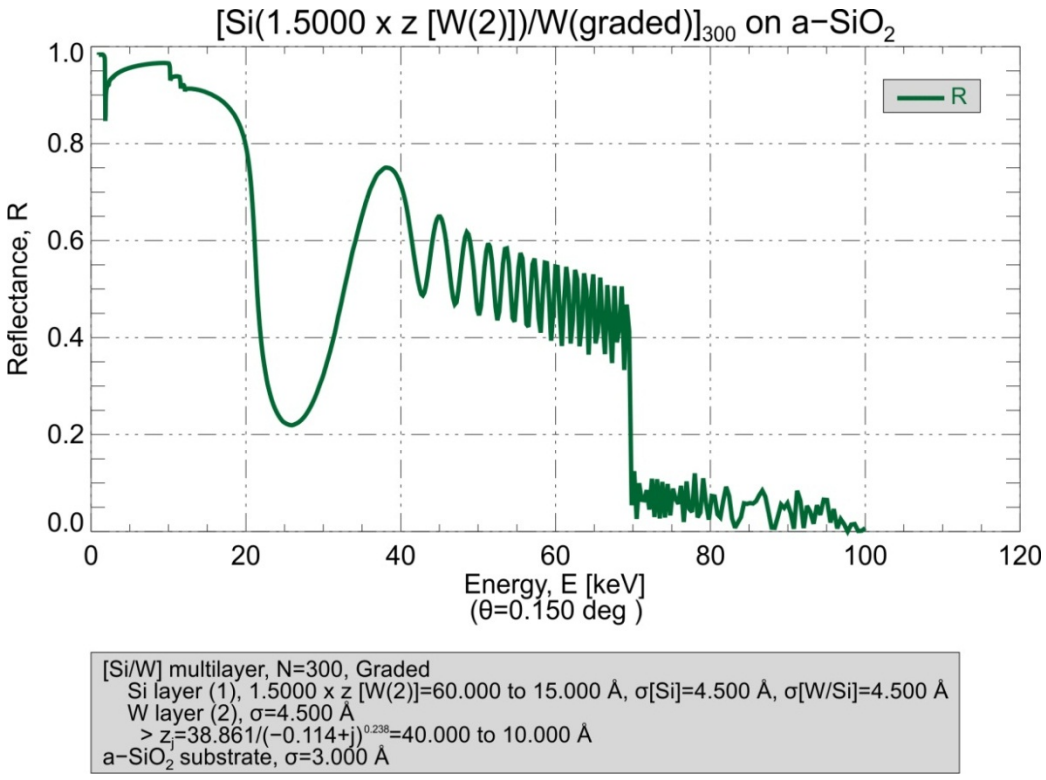
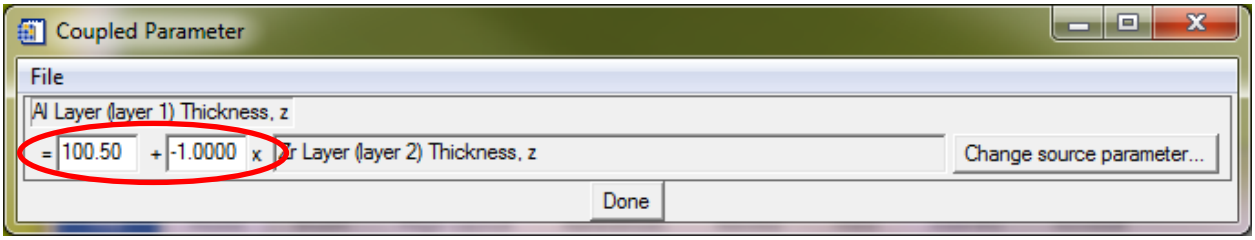


Figure 24. Reflectance vs. Energy of the depth-graded Si/W multilayer at θ=0.15° grazing incidence.

4.12.2 Example: Graded-Γ Al/Zr Multilayer

Our second example illustrating the use of Coupled Parameters involves an Al/Zr multilayer designed for high reflectance near normal incidence in the EUV. In this case we want to vary the layer thickness ratio (Γ) of each bilayer in the stack while maintaining a constant multilayer period of d=100.5 Å for each bilayer, in order that the peak reflectance at 5° incidence occurs at λ=195 Å wavelength. (The so-called "Graded-Γ" design is used to achieve a more narrow spectral band-pass, and to suppress the reflectance of this multilayer for wavelengths shorter than λ=195 Å.) We'll therefore use both a depth-graded layer thickness and a coupled parameter so that the layer thickness ratio Γ varies linearly with depth in the film while the period remains constant.

We first create the periodic Al/Zr multilayer, in this case with N=60 bilayers. We then specify the Zr layer thickness to vary linearly with thickness, from z_{Zr}=20 Å at the top of the film stack, to z_{Zr}=50 Å at the bottom, using the methods described in §4.5. We next add a Coupled Parameter that fixes the Al thickness to be equal to z_{Al}=100.5 Å – z_{Zr}; that way the combination of z_{Al}+z_{Zr} will always equal 100.5 Å:



The resulting Layer Thickness Profile and Structure Diagram are shown in Figure 25, while the EUV reflectance of this film is shown in Figure 26, along with the response of a conventional periodic Al/Zr multilayer with constant d and Γ values.

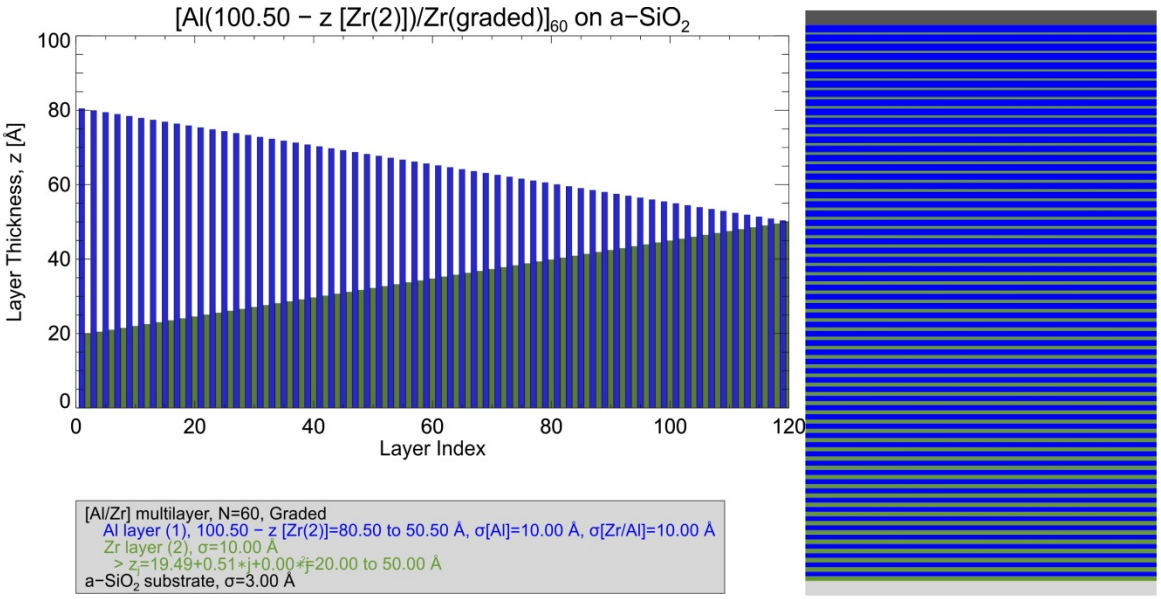


Figure 25. Layer Thickness Profile (left) and Structure Diagram (right) for the Graded-Γ Al/Zr multilayer.

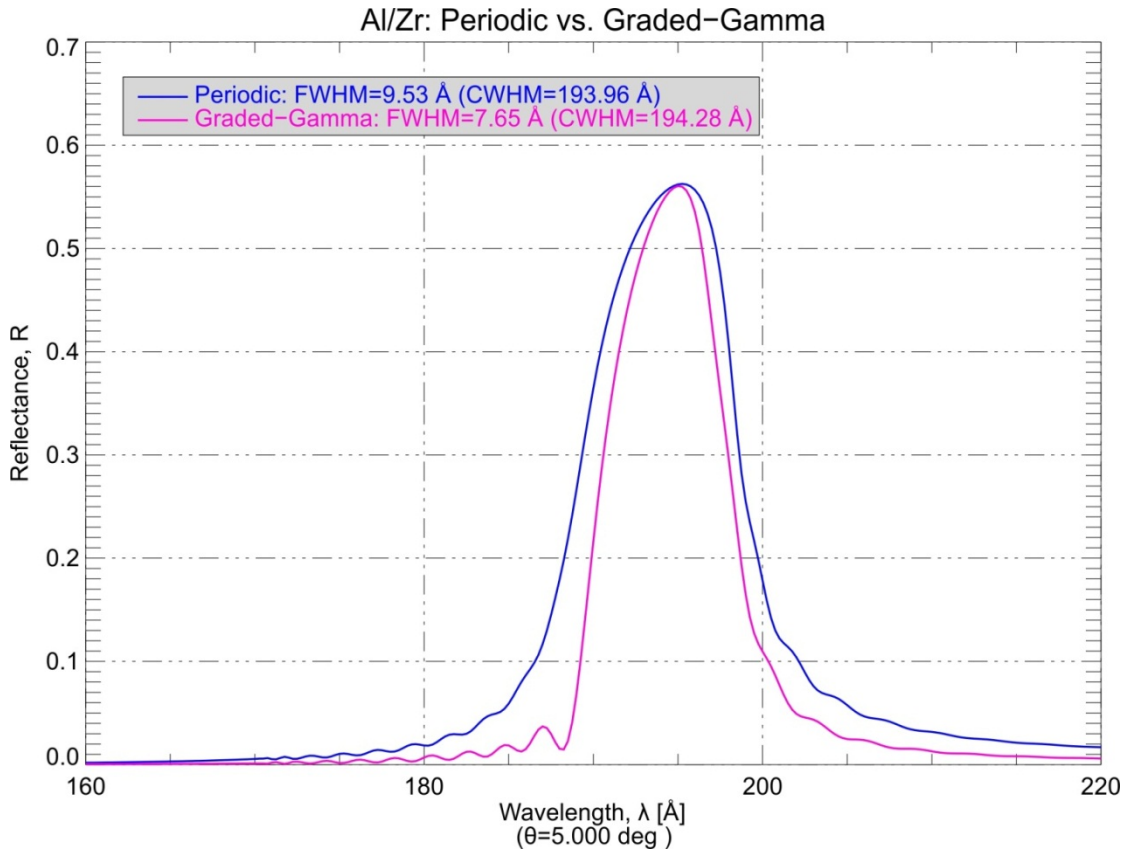


Figure 26. Comparison of normal incidence reflectance of periodic vs. graded-Γ Al/Zr multilayers.

In addition to the two examples presented in this chapter, Coupled Parameters also can be used in a variety of other ways. As we'll see in the chapters ahead, Coupled Parameters can be particularly useful when structure parameters are designated as Independent Variables, Fit Parameters or Optimization Parameters.

5 Specifying Variables

Before you can perform a computation in IMD, in addition to defining the structure to be modeled, as we've just discussed in the last chapter, you also need to specify which optical functions you want to compute, and as a function of which independent variables. The focus of this chapter is on the specification of these dependent and independent variables.

5.1 Dependent Variables

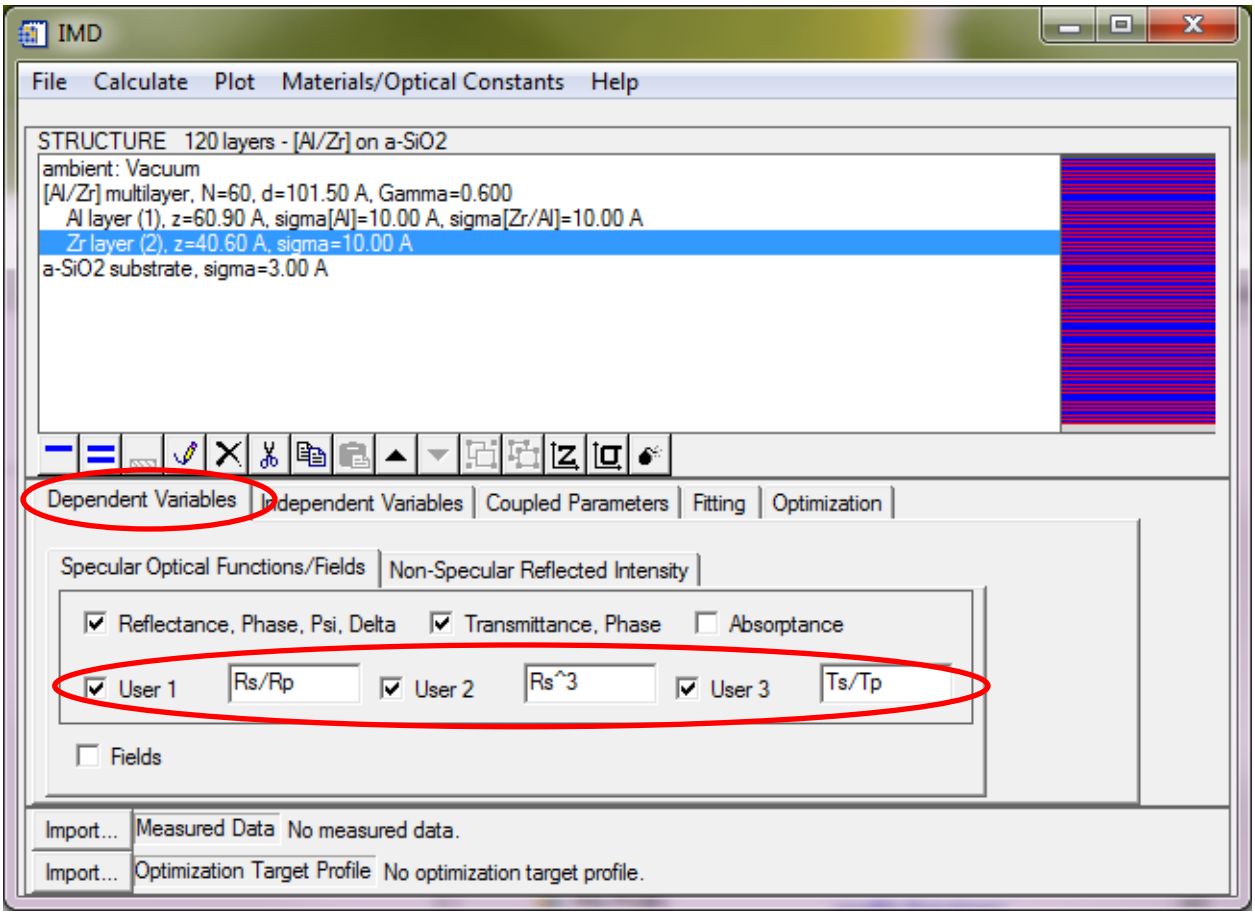
Dependent variables in IMD are simply the optical functions that you want to compute. Check the appropriate boxes on the Dependent Variables tab on the main IMD window to compute specular optical functions and field intensities, shown on the sub-tab labeled Specular Optical Functions/Fields, or non-specular reflected intensity, shown on the sub-tab with that label.

5.1.1 Specular Optical Functions and Field Intensities

For each specular optical function you select on the Dependent Variables tab, IMD will calculate the values of that function for pure s and pure p polarization, as well as the average value of the function for the specified polarization parameters f and q, which are described in §5.2. For instance, if you select reflectance as a dependent variable, when you actually perform the calculation the program will compute the reflectance functions for pure s and p polarization, Rs, and Rp, in addition to the average reflectance function R(f,q). When you compute Reflectance and/or Transmittance, the program will also compute the phases of the reflected and/or transmitted waves for pure s and pure p polarizations; the ellipsometric psi and delta functions will be computed as well when Reflectance is selected as a dependent variable. Once the calculation is performed, any or all of these variables can be displayed, as we'll see in the next chapter.

5.1.1.1 User-Defined Specular Optical Functions

Once you have selected specular optical functions as dependent variables, you can also choose to calculate up to three 'user-defined' optical functions. These functions are completely arbitrary combinations of whatever optical functions you're interested in. For example, to compute the ratio of reflectance for pure s to pure p polarization, you can define a function equal to Rs/Rp, as shown in the next example, where we've also specified two other user functions, Rs^3 and Ts/Tp:



User-defined optical functions can be computed from any of the specular optical functions you select, and can also use any other valid IDL functions. However, you must always (a) use valid IDL syntax, and (b) use the correct variable names of any IMD optical functions you reference – variable names used in IMD are explained in §11.1.1. Also, the user-defined functions that you specify must have the same dimensionality as all the other optical functions resulting from that calculation. For example, the user function "total(Rs)", which is a scalar (i.e., single-valued), is invalid when Rs is multi-valued.

5.1.2 Non-Specular Reflected Intensity

The non-specular (or diffuse, or scattered) reflected intensity from a film stack can be computed in IMD using either the dynamical Born approximation (BA) vector theory described in reference [8], and/or the 'Distorted-Wave Born Approximation' (DWBA) formalism described in references [10–13]. You are highly encouraged to consult the references just cited in order to appreciate the range of validity of each of these theories.

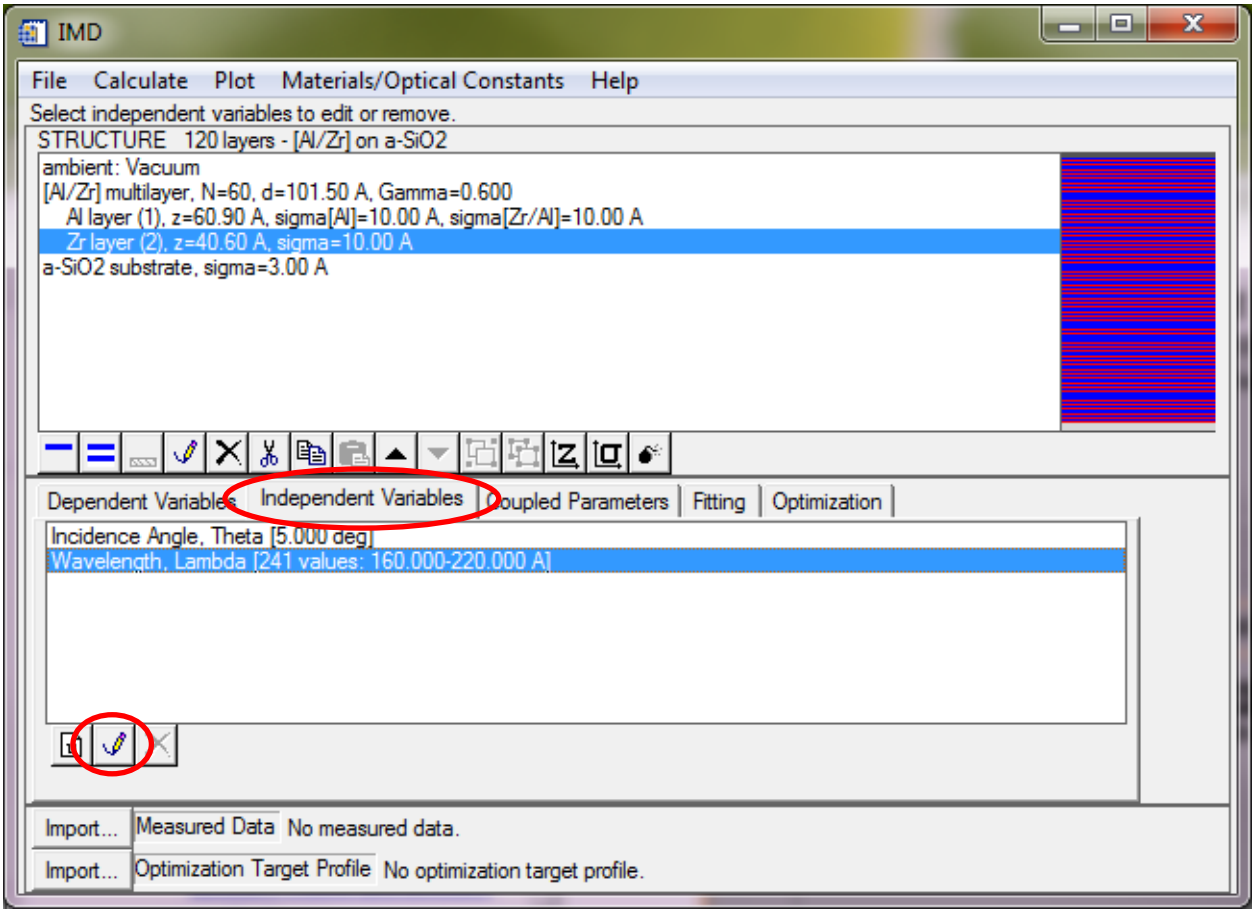
To compute non-specular reflected intensity, select either or both non-specular functions on the Dependent Variables tab, and be sure to specify PSD parameters for each interface in the structure, as explained in §4.11.2.

Note: IMD uses the 'small roughness' approximation when computing non-specular reflected intensities, which is to say that the amount of light scattered from any interface is proportional to the power-spectral-density function of that interface.

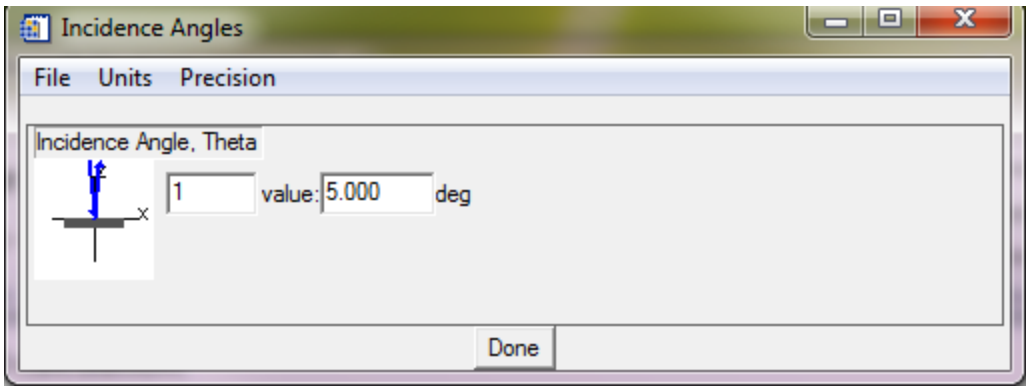
5.2 Independent Variables

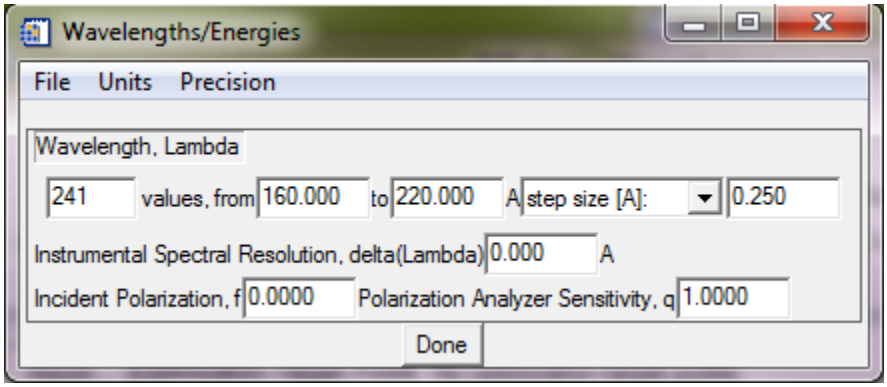
5.2.1 Incidence Angle and Wavelength/Energy

The Independent Variables tab on the main IMD window displays a list of currently defined independent variables. The first two independent variables listed are always Incidence Angle and Wavelength (or Energy, if energy units are selected), both of which must be specified for any IMD calculation:



To specify values for an independent variable, select that variable in the list, and click the "Edit" button that looks like this –  (or just double-click the list item); the corresponding window will appear where you can configure the variable. Here are the Incidence Angle and Wavelength windows associated with the previous figure:

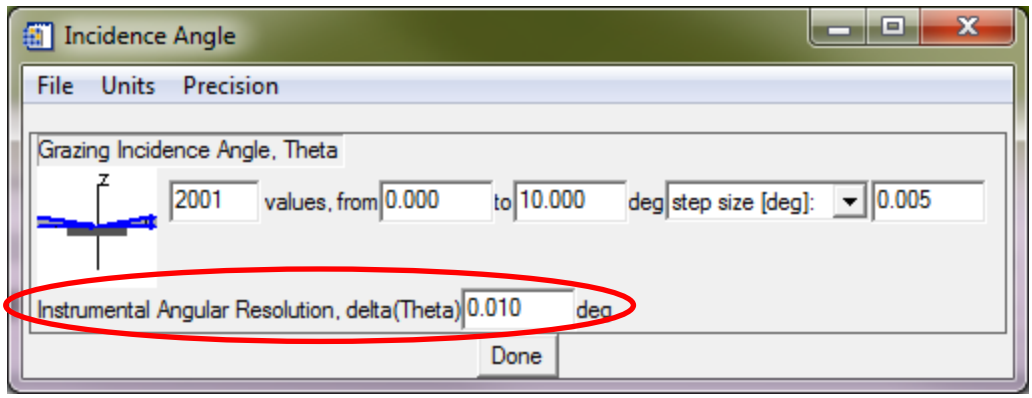




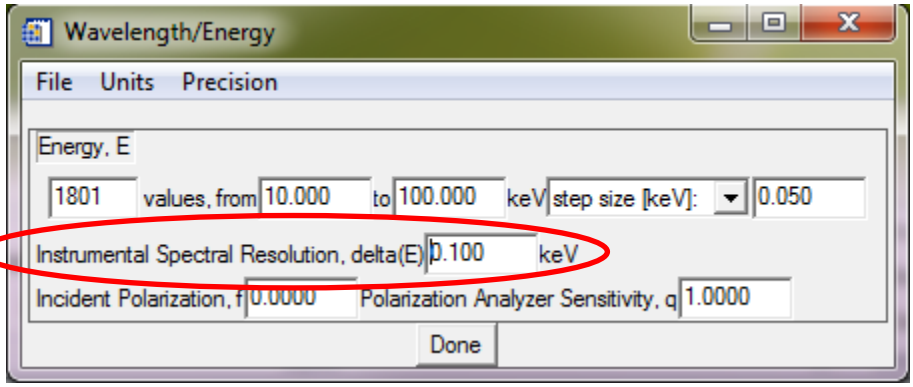
In the figures above we've specified a single incidence angle of 5° (from normal), and 241 wavelength values, from 160 Å to 220 Å, which results in a fixed step size of 0.25 Å. (You can also choose to use logarithmic steps instead of fixed steps for any independent variable, using the drop-list.)

Use the Units menu items to display angles in degrees, minutes, seconds or mrad, relative to either normal or grazing incidence. Similarly, wavelengths can be displayed in Å, nm, μm or cm^{-1}, or as energies with units of eV or keV, where photon energy E in eV is related to photon wavelength λ in Å by E[eV]=12398.425/λ[Å]. When computing neutron optical functions (i.e., using neutron optical constants), you can select meV energy units as well: neutron energy E in meV is related to neutron wavelength λ in Å by E [meV]=81.807/λ[Å].

If three or more angles are defined, you can also specify the Instrumental Angular Resolution, in order to simulate finite instrumental resolution. For example, if you specify an Instrumental Angular Resolution of 0.005°, then the optical functions you calculate will be convolved with a 0.005°-wide Gaussian function. Here's an example: suppose the angular resolution of an X-ray reflectometer is 0.010°. When computing X-ray reflectance vs. graze angle to model the XRR data obtained from that reflectometer, you can define the Angle independent variable like this:



Likewise, you can specify Instrumental Spectral Resolution when defining 3 or more wavelengths or energies. For example, here's how you might define the Energy variable for modeling the X-ray reflectance vs. energy of a multilayer where the detector has 100 eV resolution:



Note: Specifying instrumental resolution has no effect on the Electric Field Intensity calculations.

5.2.2 Polarization Parameters

As you can see in the Wavelength/Energy independent variable window shown in the last figure above, it's also possible to specify in IMD values for the Incident Beam Polarization, f, and the Polarization Analyzer Sensitivity, q. The Incident Beam Polarization, f, is defined as follows:

$$f = \frac{I_s - I_p}{I_s + I_p}$$

where I_s and I_p are the incident beam intensities for pure s and p polarization, respectively. So, for a beam that is 100% s-polarized, f=1, while for pure p polarization, f=-1. For unpolarized radiation, f=0.

> **Note**: Beam polarization is sometimes specified by the quantity $p=I_s/(I_s+I_p)$. In that case, the value of f is given by f=2p-1.

The Polarization Analyzer Sensitivity, q, refers to the detector system used to measure the beam intensity, and is defined as the sensitivity of the detector to s-polarized radiation divided by its sensitivity to p-polarized radiation. Specifying a value of q other than 1 could be used to simulate, for example, the reflectance you would measure using a detector that (for whatever reason) was more or less sensitive to s polarization than to p polarization.

5.2.2.1 Computation of 'Average-Polarization' Optical Functions

For specular optical functions, the f and q values you specify are used to compute the 'average' specular optical functions. For example, the average reflectance R is given by:

$$\mathrm{R} = \mathrm{R_s} \times [\mathrm{q} \times (1 + \mathrm{f})] + \mathrm{R_p} \times \left(\frac{1 - \mathrm{f}}{\mathrm{f} \times (\mathrm{q} - 1) + (\mathrm{q} + 1)}\right)$$

The average transmittance, absorptance and field intensities are given by analogous expressions.

For non-specular reflected intensity calculations using the Born approximation, since there is coupling between s and p polarizations (in general), things are way more complicated:

The s- and p-polarized components of the non-specular reflected intensity, $\frac{d\mathrm{I}}{\mathrm{I_0} d\Omega_\mathrm{s}}$, and $\frac{d\mathrm{I}}{\mathrm{I_0} d\Omega_\mathrm{p}}$, respectively, are given by:

$$\frac{d\mathrm{I}}{\mathrm{I_0} d\Omega_\mathrm{s}} = \frac{1}{2} \times \left[\frac{d\mathrm{I}}{\mathrm{I_0} d\Omega_\mathrm{ss}} \times (1 + f) + \frac{d\mathrm{I}}{\mathrm{I_0} d\Omega_\mathrm{ps}} \times (1 - \mathrm{f})\right]$$

and

$$\frac{d\mathrm{I}}{\mathrm{I_0} d\Omega_\mathrm{p}} = \frac{1}{2} \times \left[\frac{d\mathrm{I}}{\mathrm{I_0} d\Omega_\mathrm{sp}} \times (1 + f) + \frac{d\mathrm{I}}{\mathrm{I_0} d\Omega_\mathrm{pp}} \times (1 - \mathrm{f})\right]$$

where

$\frac{d\mathrm{I}}{\mathrm{I_0} d\Omega_\mathrm{ss}}$ = non-specular reflected intensity component having s polarization, resulting from the fraction of the incident beam having s polarization,

$\frac{d\mathrm{I}}{\mathrm{I_0} d\Omega_\mathrm{sp}}$ = non-specular reflected intensity component having s polarization, resulting from the fraction of the incident beam having p polarization,

$\frac{d\mathrm{I}}{\mathrm{I_0} d\Omega_\mathrm{ps}}$ = non-specular reflected intensity component having p polarization, resulting from the fraction of the incident beam having s polarization,

and

$\frac{dI}{I_0 d\Omega_{pp}}$ = non-specular reflected intensity component having p polarization, resulting from the fraction of the incident beam having p polarization.

Finally, the average non-specular reflected intensity, i.e., as measured with a detector having polarization sensitivity, q (as defined above,) is then given by:

$$\frac{dI}{I_0 d\Omega_a} = \frac{1}{(q+1)\times\left(1+\left|\frac{q-1}{q+1}\right|\right)} \times \left[\frac{dI}{I_0 d\Omega_s} \times q + \frac{dI}{I_0 d\Omega_p}\right]$$

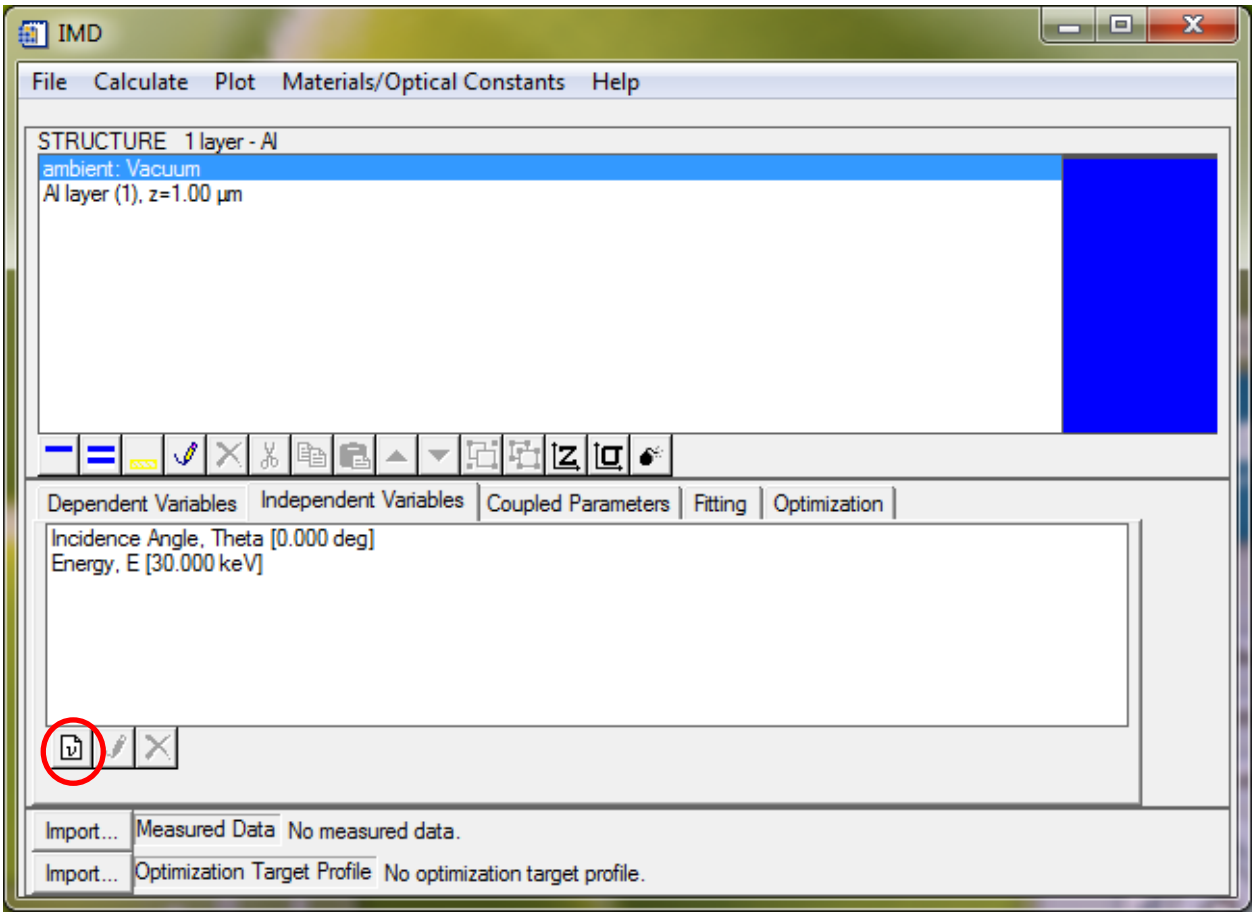
Thus, the values of the s and p components of the scattered intensity depend on the specified value of f, while the 'average' scattered intensity depends on both f and q.

5.2.3 Defining Other Parameters as Independent Variables

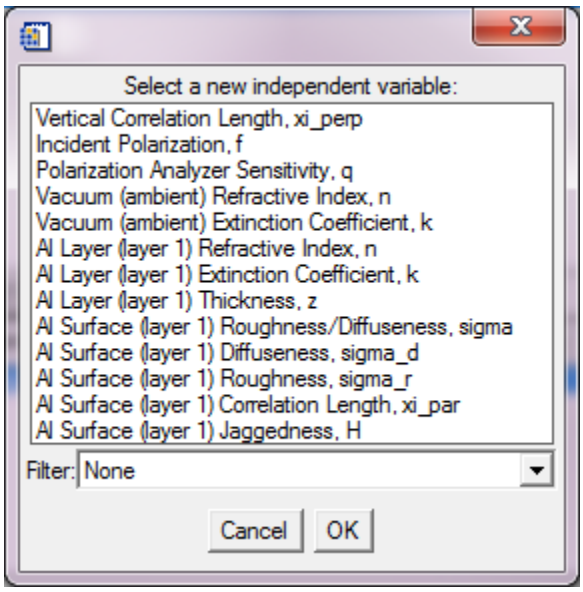
All IMD calculations require values for Angle and Wavelength/Energy; each can be specified as either single- or multi-valued independent variables. As we'll see in §5.2.4, computations of Field Intensity require a third independent variable – Depth – while non-specular reflected intensity calculations require specification of two additional independent variables related to the scattering angles, as we'll see in §5.2.5. But along with these mandatory independent variables, in IMD almost any of the parameters that define the structure (Chapter 4), as well as the instrumental resolution and polarization parameters discussed above, can be designated as independent variables too. Up to 8 independent variables (including all the mandatory independent variables) can be specified simultaneously. In this section we'll see how to define these additional independent variables, and we'll do so with the help of another example.

5.2.3.1 Example: X-ray Transmittance of Aluminum versus Thickness

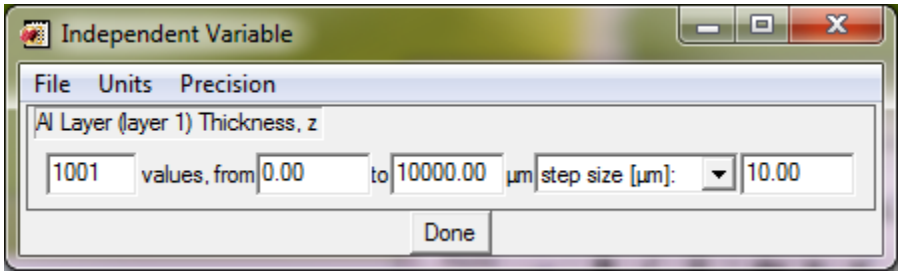
For this example we'll look at the transmittance of Al as a function of thickness, at a photon energy of E=30 keV. We'll start by defining the structure as a single layer of aluminum, of thickness z=1 μm. (See the file `examples/Independent Variables/Al.imd` to follow along.) We've set the incidence angle to be 0° from normal, and the photon energy to be E=30 keV, as shown in the next figure:



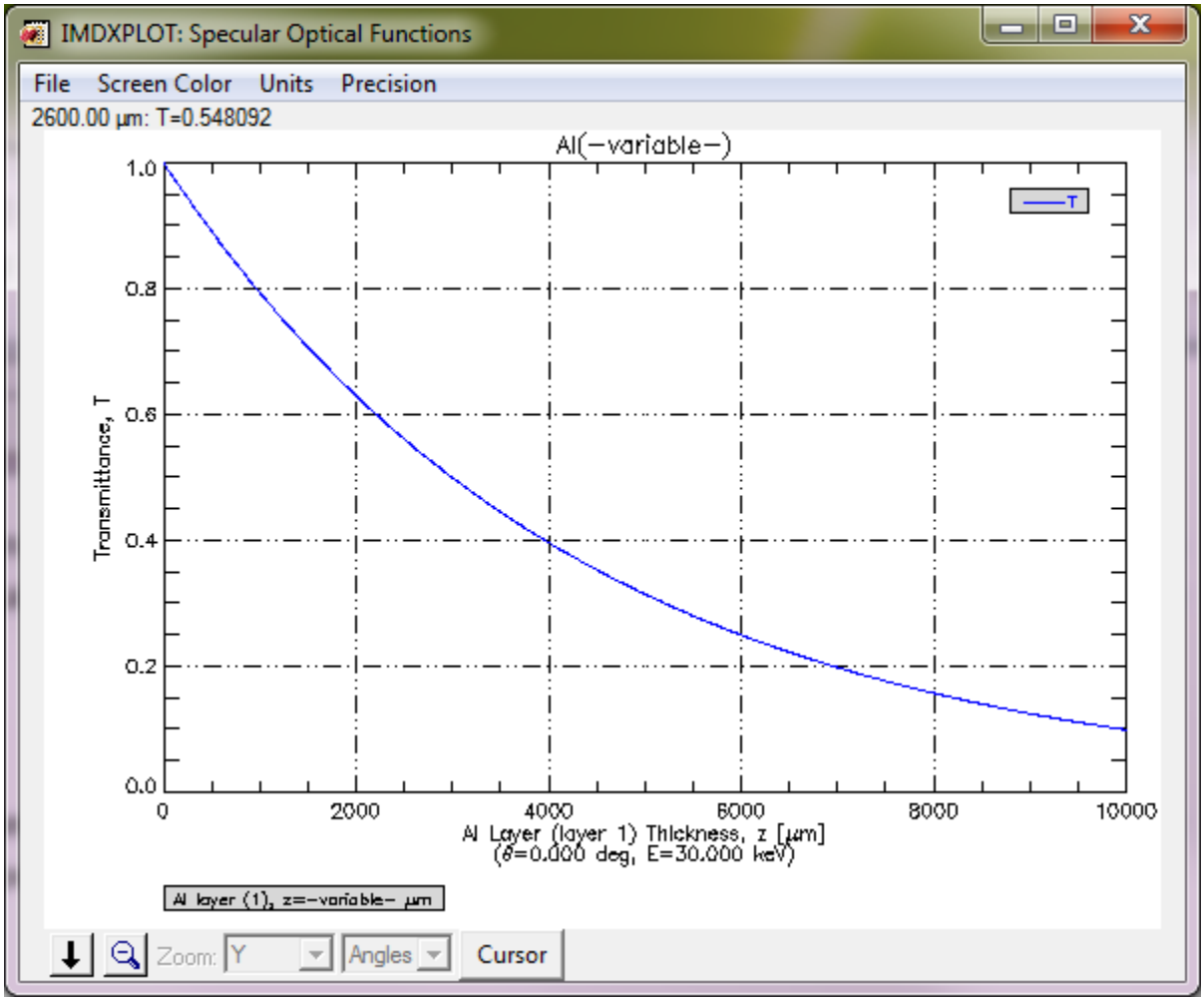
We'll now add a new Independent Variable by selecting the "Add Independent Variable" button at the bottom of the Independent Variables tab shown above, which looks like this:  . We're presented with a new window displaying a list of available parameters that can be selected as independent variables:



We select "Al Layer (layer 1) Thickness, z" from the list, which bring up a new Independent Variables window for this parameter. We specify 1001 values for this parameter, from z=0 μm to z=10,000 μm, in 10 μm steps:



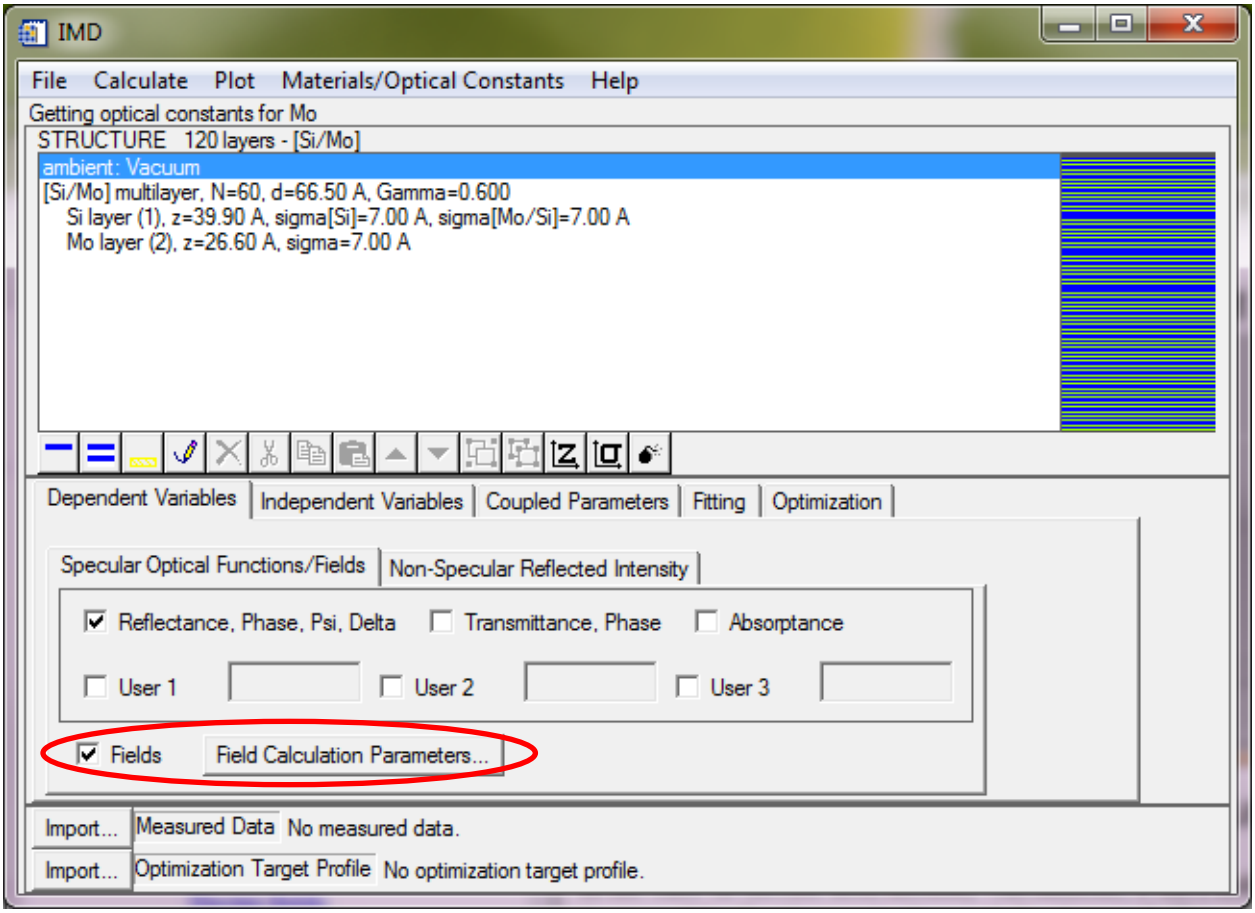
After checking the "Transmittance, Phase" box on the Dependent Variables tab, and then "Calculate→Specular Optical Functions/Fields" from the menu bar on the main IMD window, we're presented with the result:



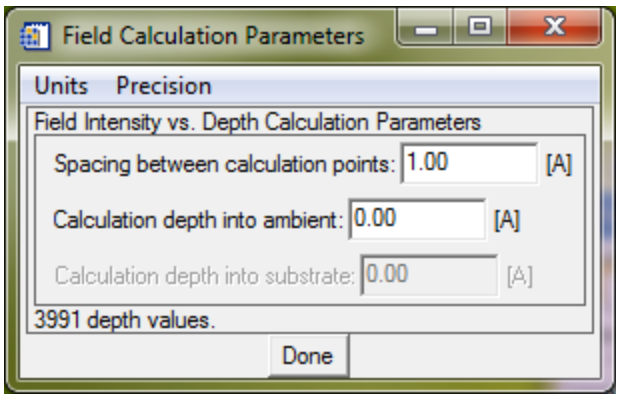
In the next chapter we'll see how to visualize more complicated computations that use more than one multi-valued independent variable.

5.2.4 Independent Variables Specific to Field Intensity Computations

When computing Field Intensity vs. Depth in the film stack, you must specify the parameters that determine how the Depth variable is defined, i.e., the spacing between Depth values, and how far the Depth variable extends into the ambient and the substrate. The Depth variable is never listed on the Independent Variables tab; instead it appears automatically when you check the "Fields" box on the Dependent Variables tab, or when you click the button labeled "Field Calculation Parameters" on that same tab:



The "Field Calculation Parameters" window is where you tell IMD how to construct the Depth variable, z: you specify the step size (i.e., "Spacing between calculation points" – default is 1 Å), and you can also extend the z variable into the ambient and/or into the substrate (if a substrate is present) to see how the fields behave in those regions. The number of depth values that results, i.e., corresponding to the variables you specify, is indicated near the bottom of the window:



5.2.5 Independent Variables Specific to Non-Specular Reflected Intensity Computations

When you select on the Dependent Variables tab of the main IMD window either the BA or the DWBA model to compute non-specular reflected intensity (described above in §5.1.2), two additional Independent Variables are automatically created and added to the list on the Independent Variables tab. These two variables – Scattering Angle and Scattering Plane Azimuthal Angle – allow for specification of the scattering angle(s) θ_{out}, and the scattering plane azimuthal angle(s), ϕ, used to compute non-specular reflected intensity. These two angles define the direction of the scattered wave vector(s), **k'**, in polar coordinates, as illustrated in Figure 27:

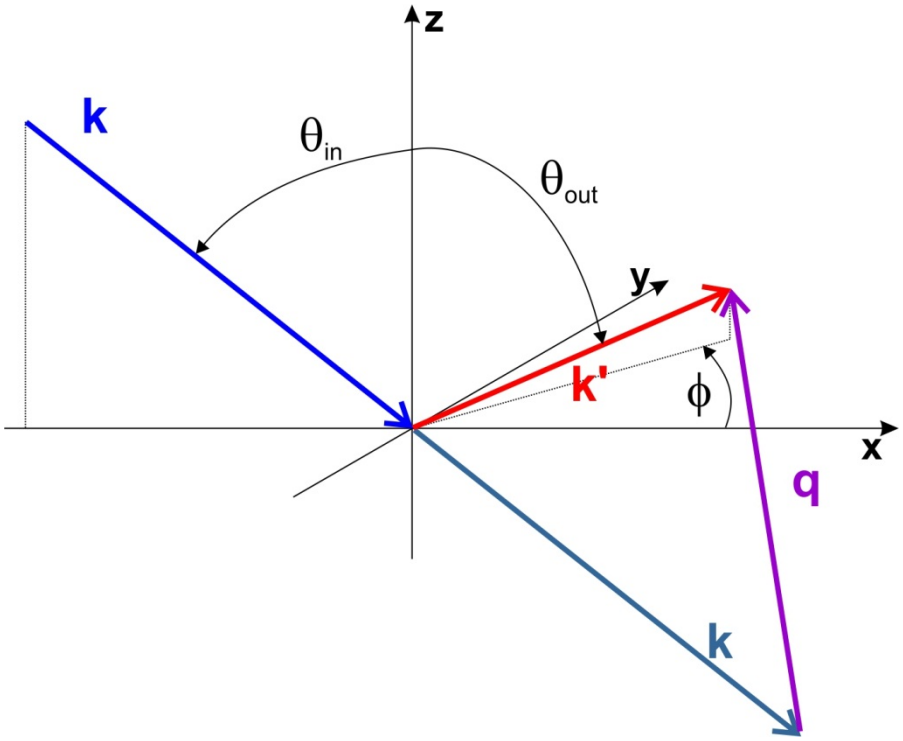
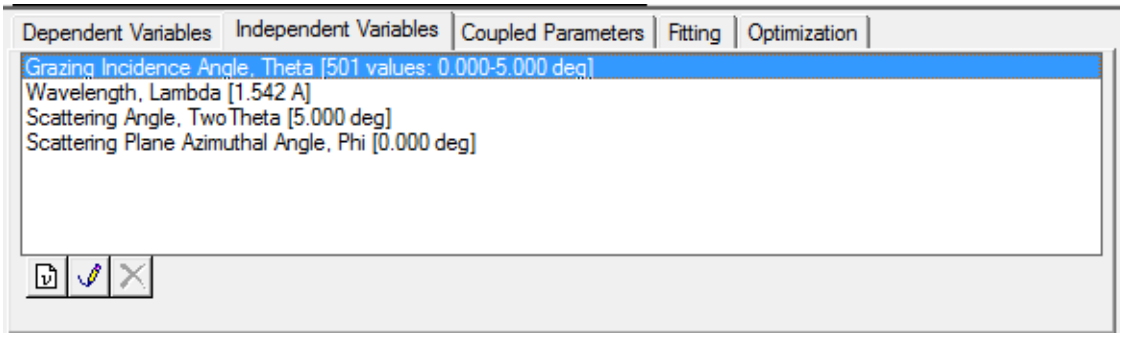


Figure 27. Scattering geometry used for non-specular reflected intensity computations. The momentum transfer vector q is equal to the difference between the scattered wave vector k' and the incident wave vector k.

The two non-specular scattering variables just discussed are always listed after the Incidence Angle and Wavelength/Energy variables on the Independent Variables tab:



Non-specular reflected intensity computations can be performed in IMD for any of three different scattering geometries, or 'scans', as summarized in Table 2:

Table 2. Non-specular scattering geometries available in IMD.

Scan Type	Brief Description
Rocking Scan	Detector fixed, sample rotated
Detector Scan	Sample fixed, detector rotated
Offset Scan	Sample and detector rotated synchronously, with the detector offset from the specular direction by a fixed amount.

The three types of non-specular scattering scans listed in Table 2 – Rocking Scan, Detector Scan, and Offset Scan – are commonly used for grazing incidence X-ray scattering measurements of thin films, but in principle, they can be used for any wavelengths and over any range of incidence angles:

- A **Rocking Scan** refers to the measurement of non-specular reflected intensity where the detector is held fixed relative to the incident beam, and the sample is rotated, or 'rocked', with respect to the incident beam. In this case, both the incidence angle θ_{in} and the scattering angle θ_{out} vary synchronously.

- A **Detector Scan** refers to the case where the sample is fixed relative to the incident beam, and the detector is varied. Thus, θ_{in} is constant while θ_{out} is varied.

- In an **Offset Scan**, the sample and detector are both rotated synchronously, as in an X-ray reflectance measurement in the so-called "θ-2θ" geometry, but the detector is offset from the specular direction by a constant offset angle, i.e., θ_{out}-θ_{in} = constant.

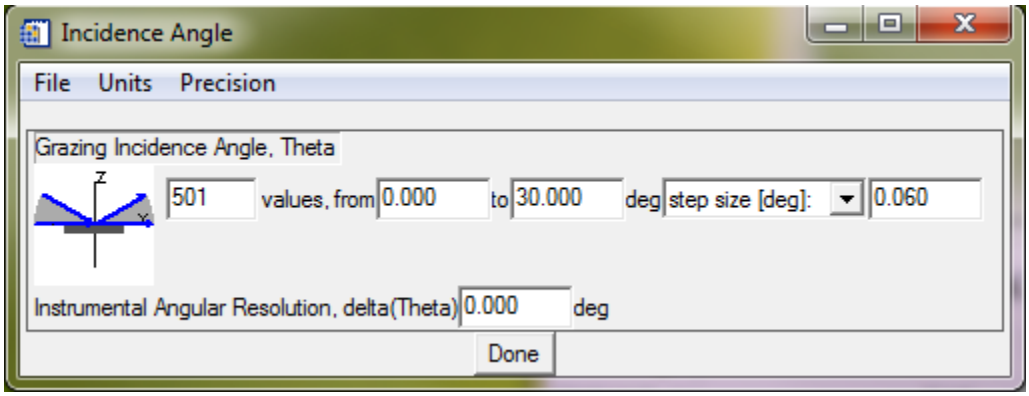
The way that the incidence angles and scattering angles are specified in IMD depends on both the type of scan you have selected in the Scattering Angle window, as we'll see below, and whether you choose to specify grazing incidence angles or normal incidence angles. The various possibilities are summarized in Table 3, and examples of each type of scan follow:

Table 3. The variables specified in the Incidence Angle and Scattering Angle independent variable windows depend on the scan type selected, and on the choice of Normal vs. Grazing Incidence angles.

Scan Type	Angles	Angle specified in IMD Incidence Angle window	Angle specified in IMD Scattering Angle window
Rocking Scan	Normal Incidence	$\theta=\theta_{in}$	$\theta_{in}+\theta_{out}$
Rocking Scan	Grazing Incidence	$\theta=90°-\theta_{in}$	$"2\theta"=180°-\theta_{in}-\theta_{out}$
Detector Scan	Normal Incidence	$\theta=\theta_{in}$	$\theta_{in}+\theta_{out}$
Detector Scan	Grazing Incidence	$\theta=90°-\theta_{in}$	$"2\theta"=180°-\theta_{in}-\theta_{out}$
Offset Scan	Normal Incidence	$\theta=\theta_{in}$	$\Delta("2\theta")$
Offset Scan	Grazing Incidence	$\theta=90°-\theta_{in}$	$\Delta("2\theta")$

5.2.5.1 Example: Non-Specular Scattering – Rocking Scan

Here's how to configure the Incidence and Scattering angles for a Rocking scan where the incidence angle varies from θ = 0 to 30° (grazing), with the scattering angle fixed so that "2θ" = 30°. (We'll work in the plane of incidence, so we'll keep ϕ=0°.) Here are the Incidence Angle and Scattering Angle windows for that geometry:



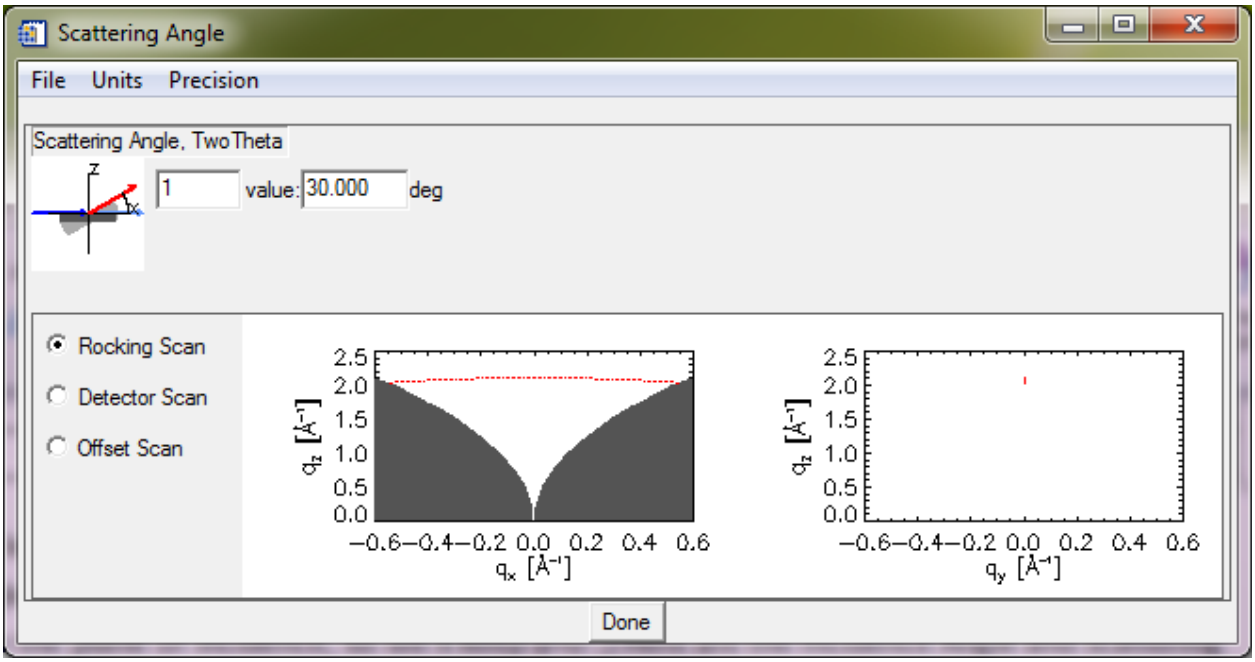
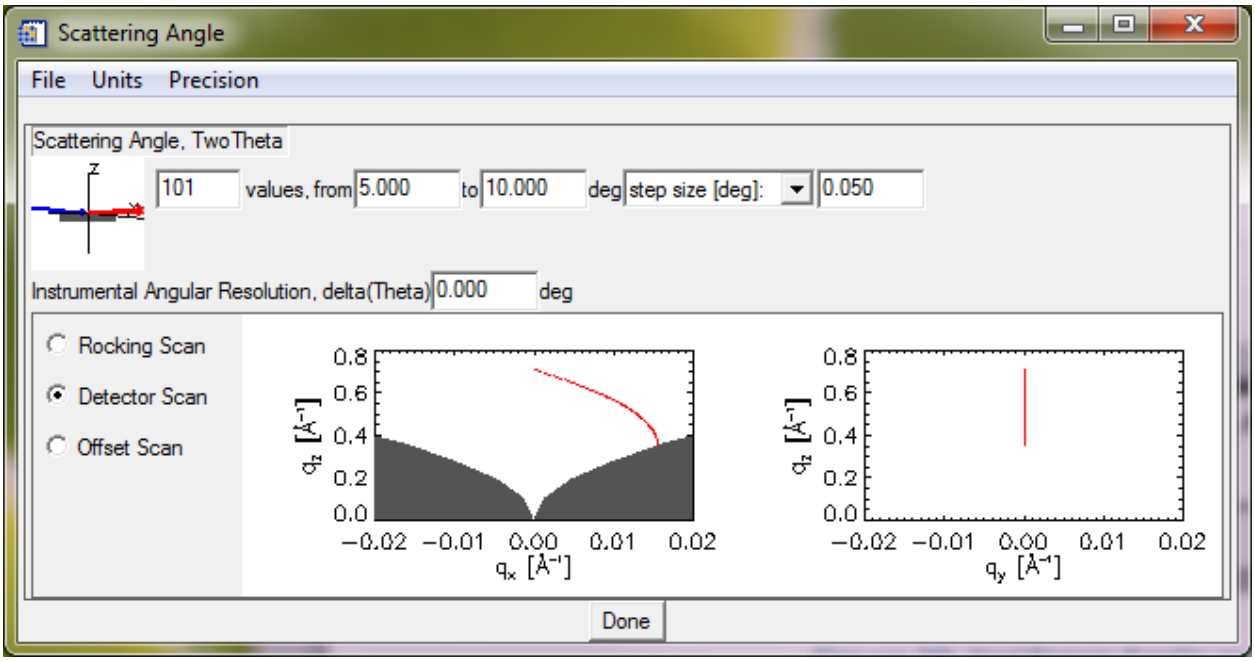


Figure 28. Incidence Angle and Scattering Angle windows showing settings for a Rocking Scan.

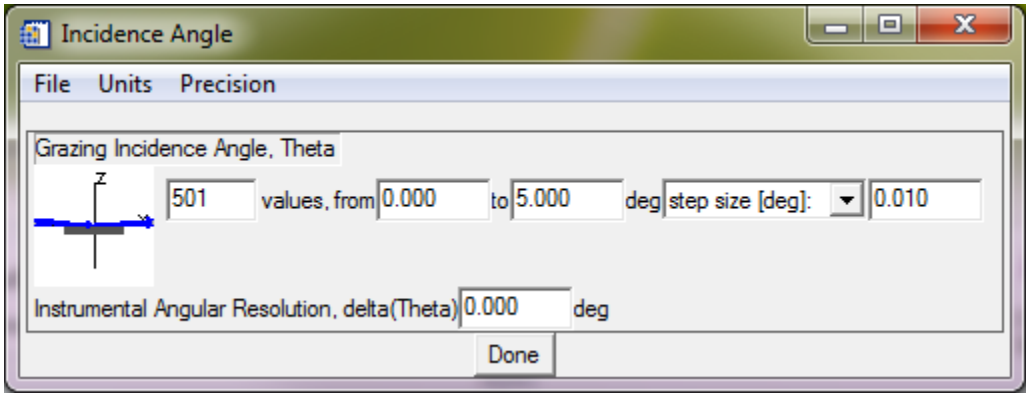
5.2.5.2 Example: Non-Specular Scattering – Detector Scan

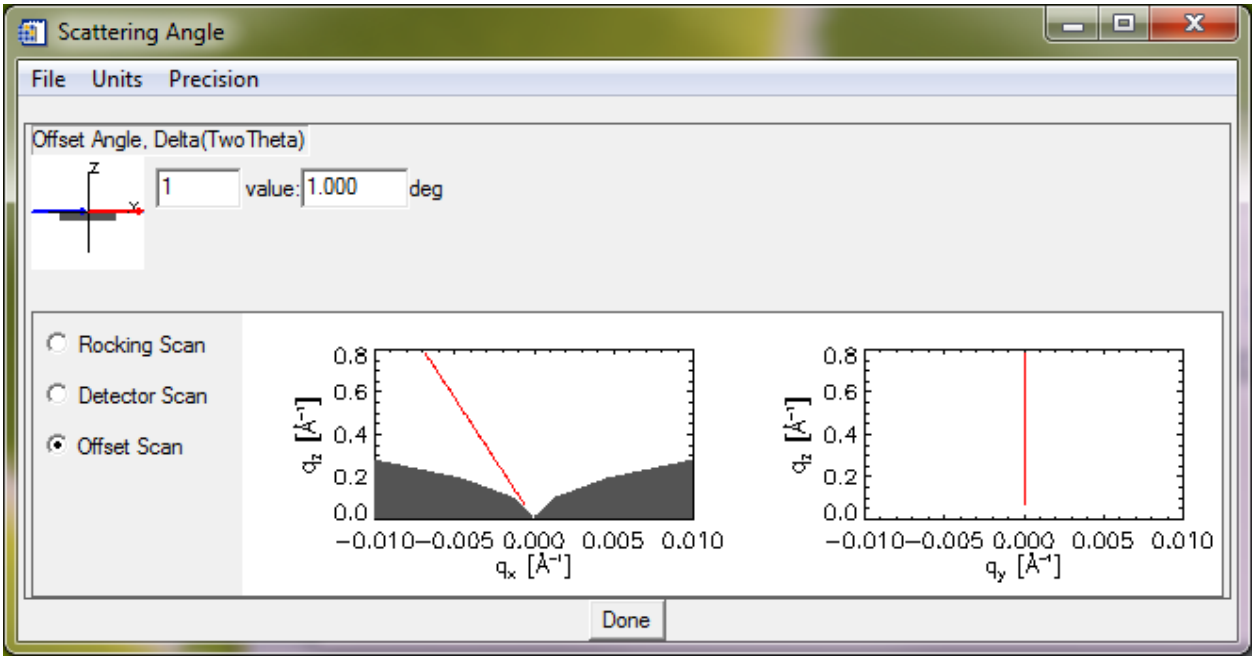
Here is the Scattering Angle window configured for a Detector Scan. The Incidence Angle is fixed at 5° grazing, and the Scattering Angle varies from "2θ"=5° to 10° grazing:



5.2.5.3 Example: Non-Specular Scattering – Offset Scan

For this Offset Scan example, we've set the Incidence Angle to vary from 0 to 5° grazing, and have set the Offset Angle to Δ("2θ")=1°:





5.2.5.4 Scattering Angle q-Space Diagrams

You'll notice that, in addition to the handy little diagrams illustrating the specified range of incidence and scattering angles visible at left in both the Incident Angle and Scattering Angle windows (e.g., Figure 28), there are also plots in the Scattering Angle window of the components of the momentum transfer vector, q, (i.e., q_z vs q_x, and q_z vs q_y,): the scan trajectories in q-space (i.e., momentum space) for the angles and wavelengths specified are shown in red; the grey regions in these plots indicate inaccessible regions of q-space, i.e., regions where either θ_{in} or θ_{out} are greater than 90°.

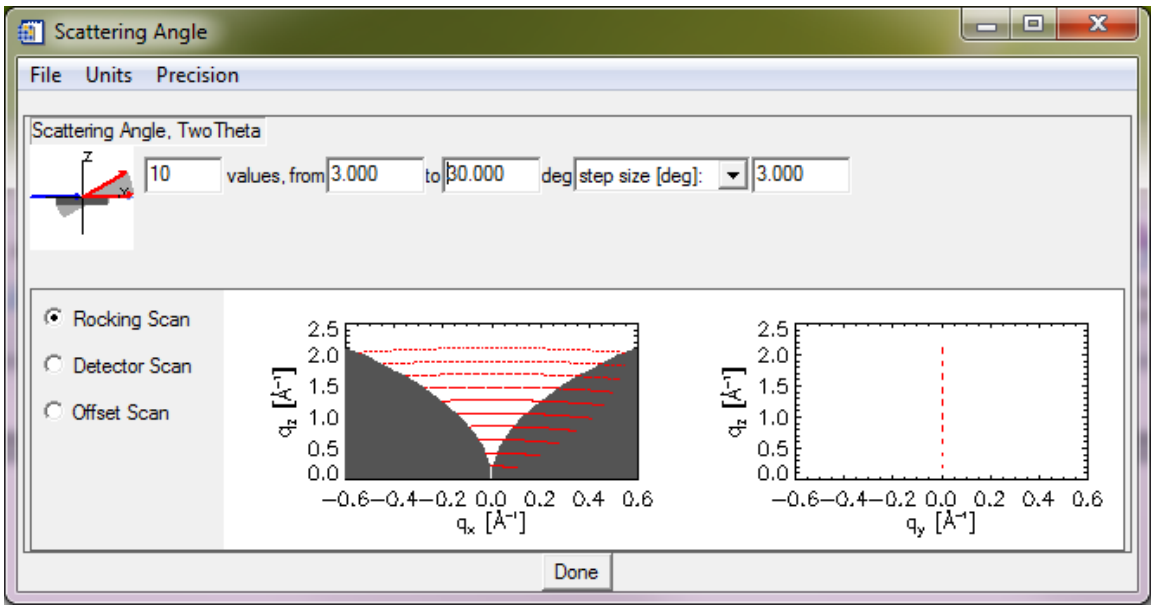
The momentum transfer vector q is shown in Figure 27; the relationship between q and the incident and scattering angles and wave vectors are as follows:

$$\mathbf{q} \equiv \mathbf{k}' - \mathbf{k} = \frac{2\pi}{\lambda}[(\sin\theta_{out}\cos\phi - \sin\theta_{in})\hat{\mathbf{x}} + \sin\theta_{out}\sin\phi\,\hat{\mathbf{y}} + (\cos\theta_{out} - \cos\theta_{in})\hat{\mathbf{z}}]$$

$$q_{\parallel} \equiv \sqrt{q_x^2 + q_y^2} = \frac{2\pi}{\lambda}\sqrt{\sin^2\theta_{out} + \sin^2\theta_{in} - 2\sin\theta_{out}\sin\theta_{in}}$$

$$q_{\perp} = q_z = \frac{2\pi}{\lambda}(\cos\theta_{out} + \cos\theta_{in})$$

Although each of the three non-specular scan types refers to a one-dimensional scan, you can specify both multiple incidence angles and multiple scattering angles for any type of scan, in order to compute non-specular reflected intensity as a function of both angles, so as to cover a larger region of q-space. For example, in the case of a Rocking Scan at grazing incidence, you can compute the non-specular reflected intensity for a range of incidence angles θ, and for a single value of "2θ" (as in Figure 28, for example), or, you might want to compute the scattered intensity for both a range of θ values and a range of "2θ" values. In the latter case, the scattered intensity will be computed as a series of Rocking Scans, i.e., the scattered intensity will be computed as a function of θ, for every value of "2θ" you have specified:



Note: Because of the fact that the momentum transfer vector q depends on the wavelength, the incidence angle, and the two scattering angles, and because it is possible in IMD to vary any or all of these four independent variables simultaneously, it is not possible to specify q-vectors directly. However, the results of your computations can be displayed as plots in momentum space, as we'll see in the next chapter.

6 Visualizing Computations: IMDXPLOT

In this chapter we'll explore IMDXPLOT, which is the program used to visualize the results of IMD computations. We've already seen in the preceding chapters how to define the structure to be modeled (Chapters 3 & 4), and how to specify dependent and independent variables (Chapter 5). Once the structure and variables are set, use the "Calculate" menu on the main IMD window to actually perform the computation: e.g., "Calculate→Specular Optical Functions/Fields", or "Calculate→Non-Specular Reflected Intensity". Once the computation is completed, IMD will display the results graphically in a new IMDXPLOT window (as we've already seen in the examples presented so far). IMDXPLOT allows you to customize the plot to your liking: you control exactly what functions are plotted and over what range of values, and you control the type of plot used and its appearance, including the display of labels and legends, axis titles, colors, symbols and so forth.

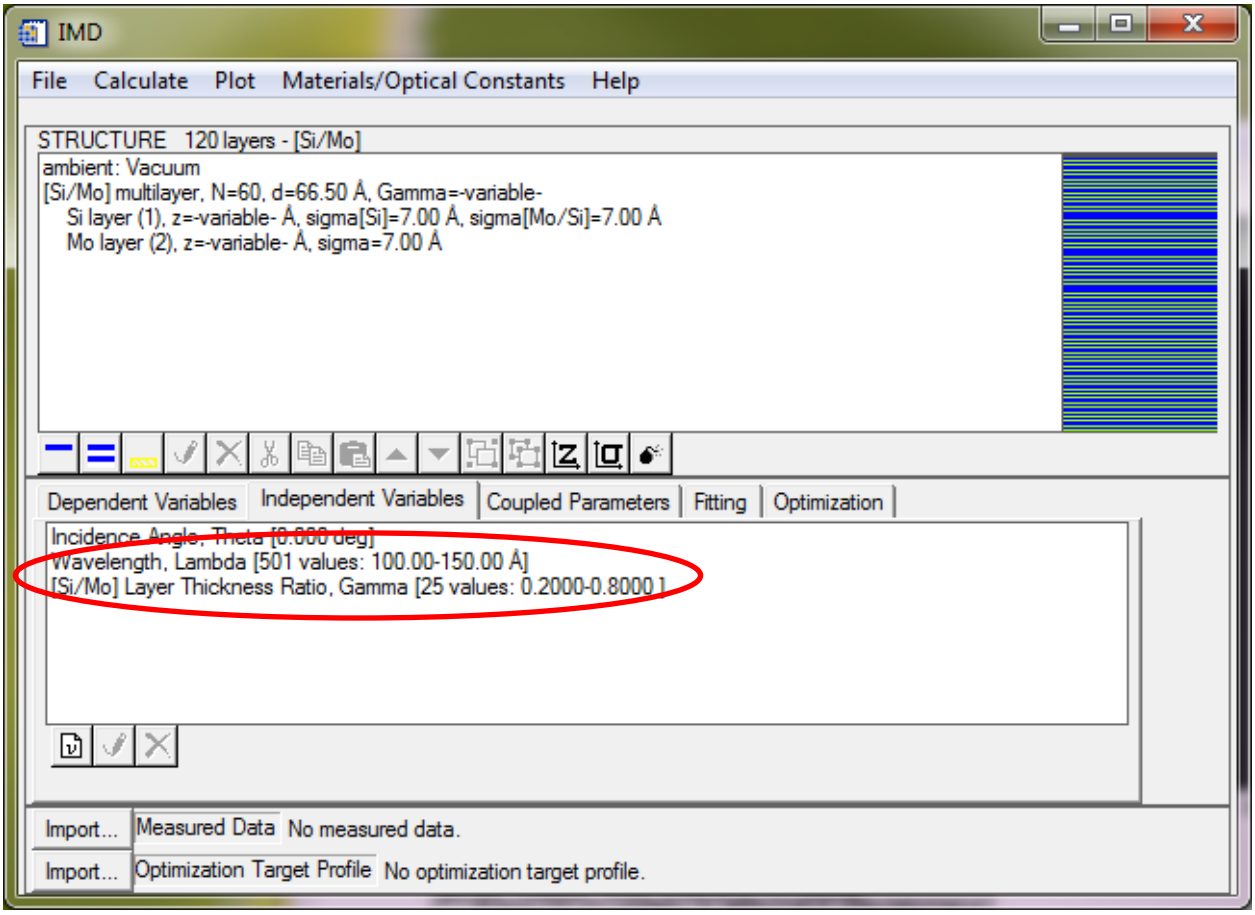
To demonstrate the functionality of IMDXPLOT, we'll go through some example computations of both specular and non-specular optical functions.

6.1 Example: Si/Mo Multilayer Reflectance vs. Γ

This example will illustrate how to visualize the results of a computation when two multi-valued independent variables are defined. In particular, we'll compute the reflectance vs. wavelength of a periodic Si/Mo multilayer as a function of the layer thickness ratio Γ. In other words, our two multi-valued independent variables in this case are Wavelength and Γ. (See the `imd/examples/Periodic Multilayers` directory for the `.imd` files associated with this example.)

We'll start with the same periodic Si/Mo multilayer we used in §2.3.2: N=60 repetitions of Si/Mo bilayers with a period d=66.50 Å. We'll keep the incidence angle at 0°, and we'll leave the Wavelength independent variable as before, with 501 values from λ=100 Å to 150 Å. But instead of keeping the layer thickness ratio fixed at Γ=0.6 as we did earlier, here we'll add Γ as an independent variable, using the techniques explained in §5.2.3. In particular, we'll specify 25 values in the range Γ=0.2 to Γ=0.8. The dependent variables resulting from this computation (e.g., R, Rs, Rp, etc) will therefore each comprise 501 x 25-element arrays once the computation is completed.

After configuring the structure and independent variables as just described, the IMD main window looks like this:



To perform the computation, we select "Calculate→Specular Optical Functions / Fields" from the menu bar. When the computation is complete, a new IMDXPLOT window appears displaying a "3D" view of Reflectance vs. Wavelength vs. Γ, as in Figure 29:

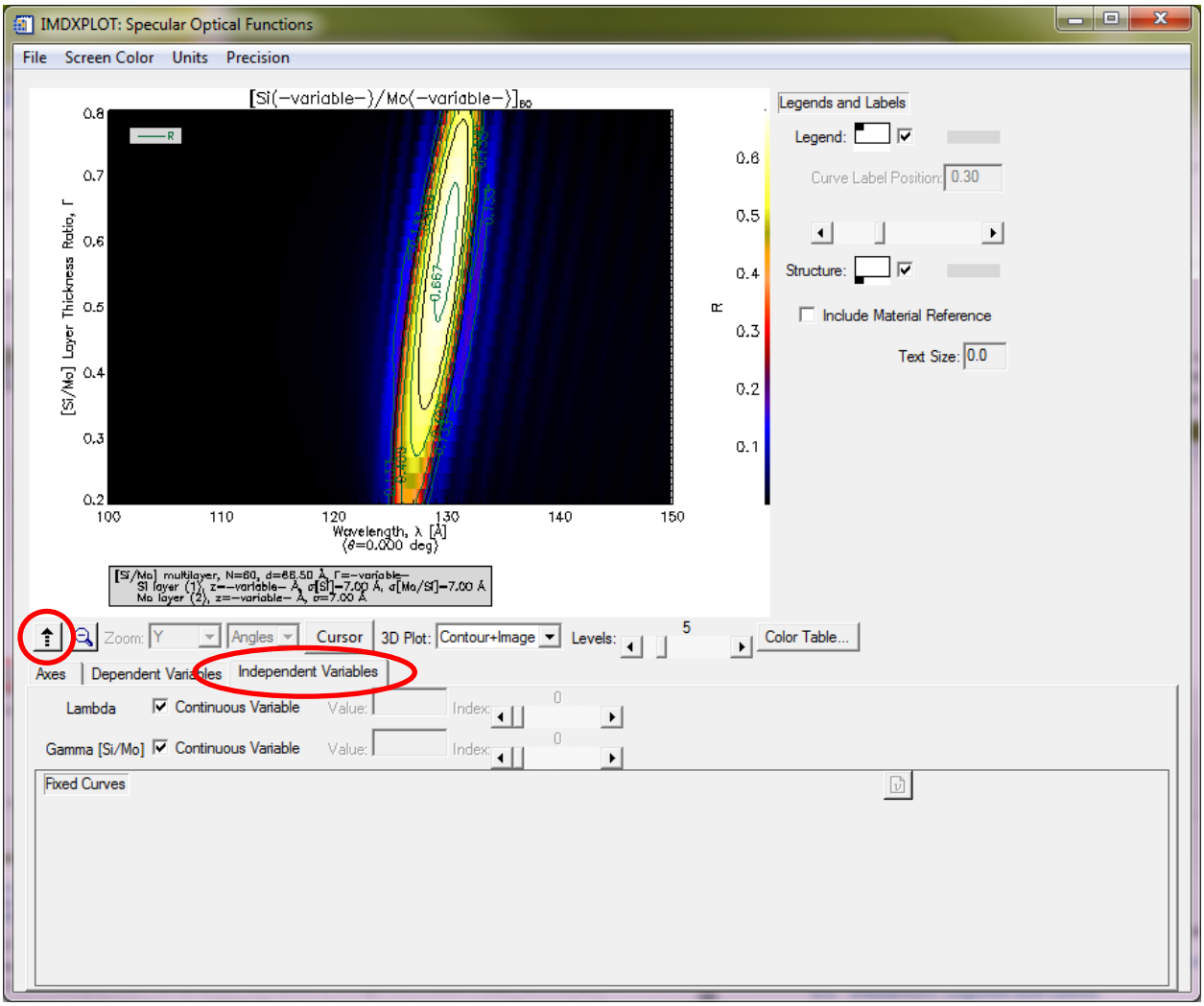


Figure 29. IMDXPLOT window displaying the results of our computation of Reflectance vs. Wavelength vs. Γ. When a computation is performed with more than one multi-valued independent variable, IMDXPLOT opens by default with all the additional controls visible, and with the Independent Variables tab displayed, as in this figure. To hide these controls, use the "Hide Additional Controls" button (circled).

You'll notice that the IMDXPLOT window shown in Figure 29 includes many more components in the area of the window outside of the plot area than in the simple IMDXPLOT window shown in Figure 1 of §2.3.2, for example, where we computed reflectance vs. a single multi-valued independent variable (wavelength in that example). In the IMDXPLOT window shown above there are controls on the right for legends and labels, and tabs on the bottom for "Axes", "Dependent Variables", and "Independent Variables". In fact it's always possible to display or hide these additional components in an IMDXPLOT window: click the "Hide Additional Controls" button on the left just below the plot with the "up" arrow ( ) as in Figure 29 to hide all these controls; click the "Show Additional Controls" button with the "down" arrow ( ) in an IMDXPLOT window (as in Figure 1) to display these controls.

When a computation is performed in IMD with more than one multi-valued independent variable, as we've done here, the IMDXPLOT window will open by default with the Independent Variables tab displayed, and with the optical function displayed as a 3D plot, i.e., vs. two independent variables. The default display for 3D plots is "contour + image", but you can use the "3D Plot" drop-list to select other options. For example, here's the same data displayed as a surface:

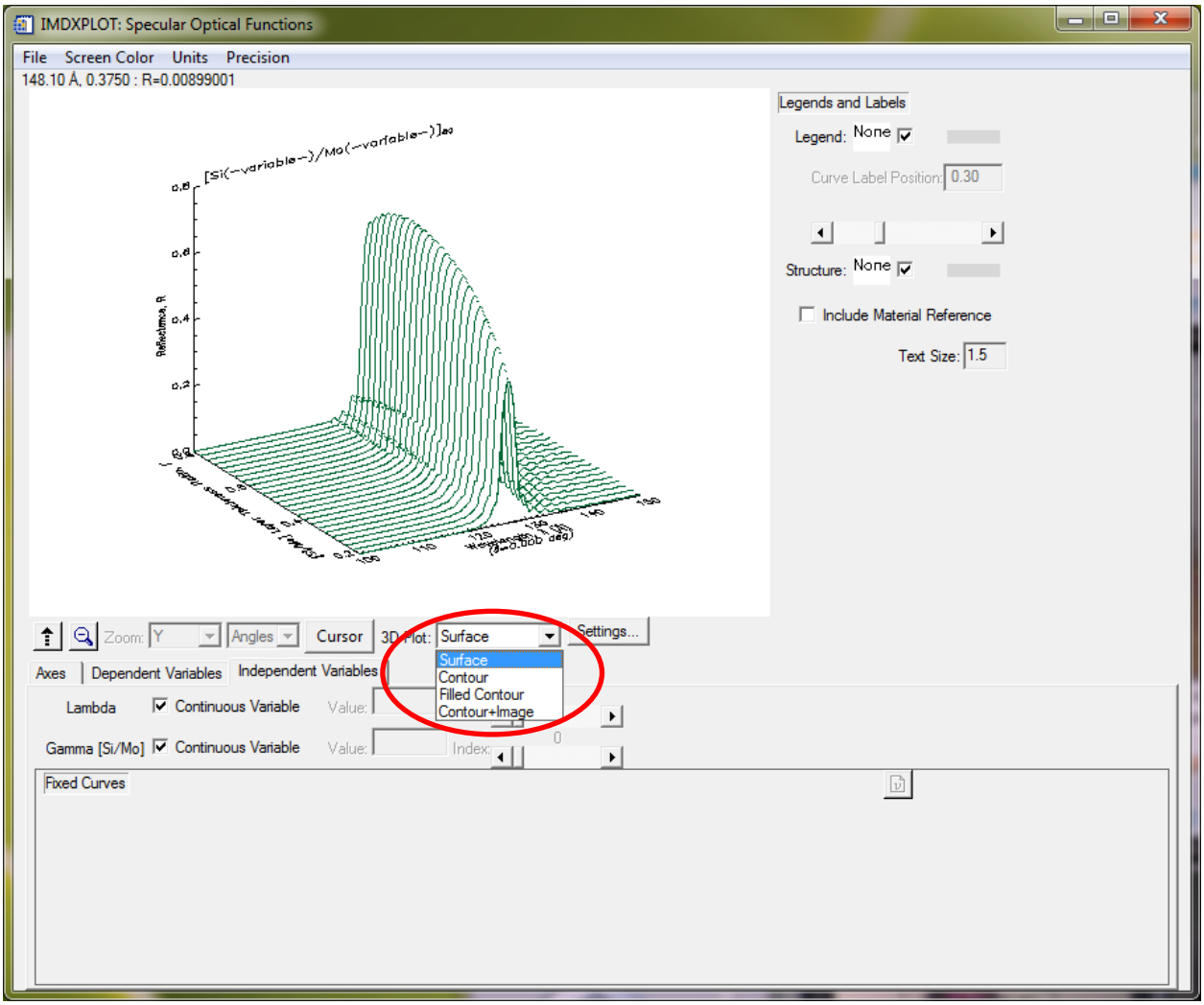


Figure 30. IMDXPLOT window showing the same data as in Figure 29 displayed as a surface.

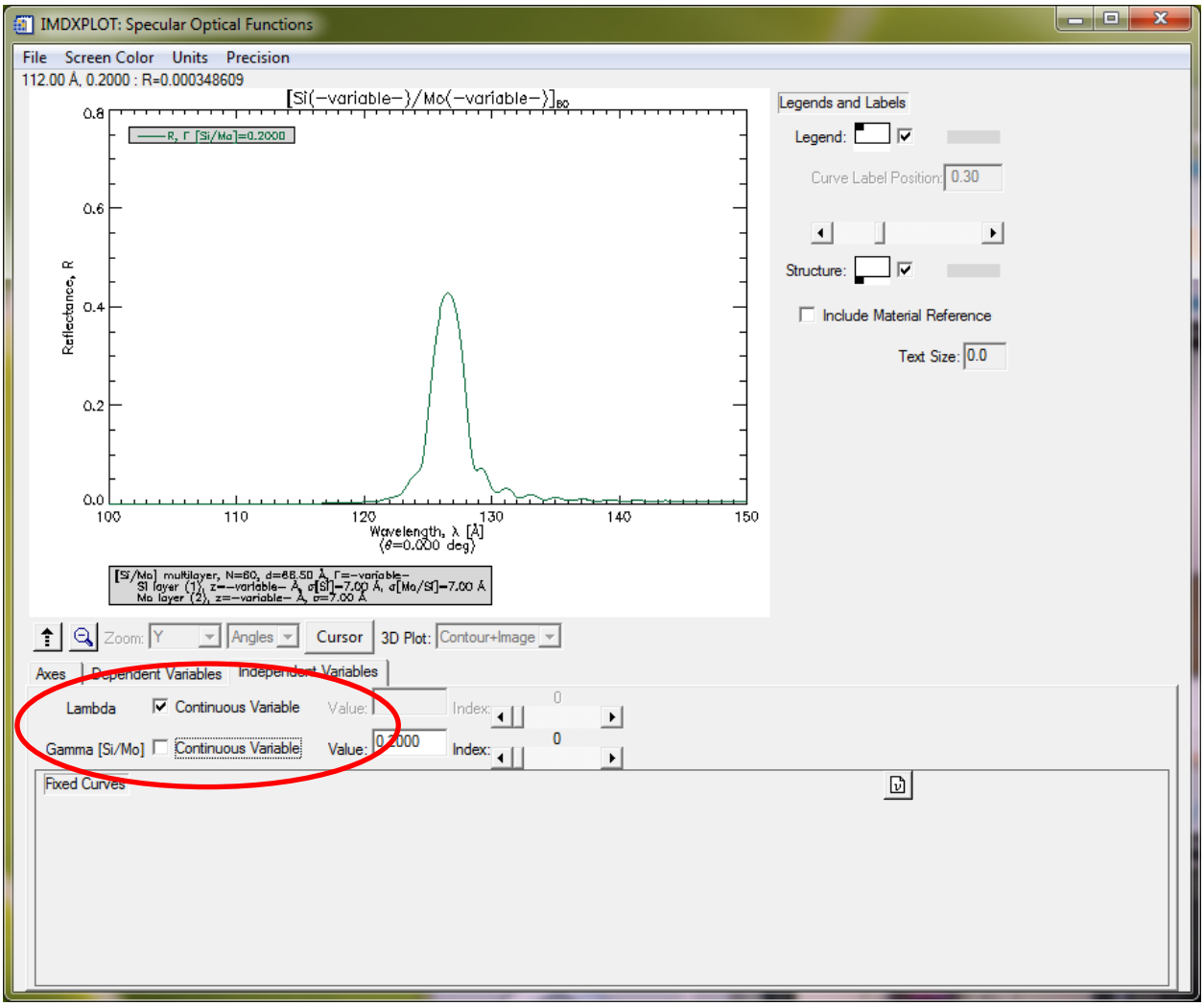
6.2 IMDXPLOT: Legends and Labels

In Figure 30, where we've displayed the results of our computation of reflectance vs. wavelength vs. Γ as a 3D surface, you'll notice that we've also suppressed the display of both the plot legend and the structure diagram that appear in Figure 29. The controls for all such legends and labels are located to the right of the plot area in the IMDXPLOT window. There are menus to control where each legend and label is positioned on the plot, whether or not the legend or label is drawn with a box around it, and if it

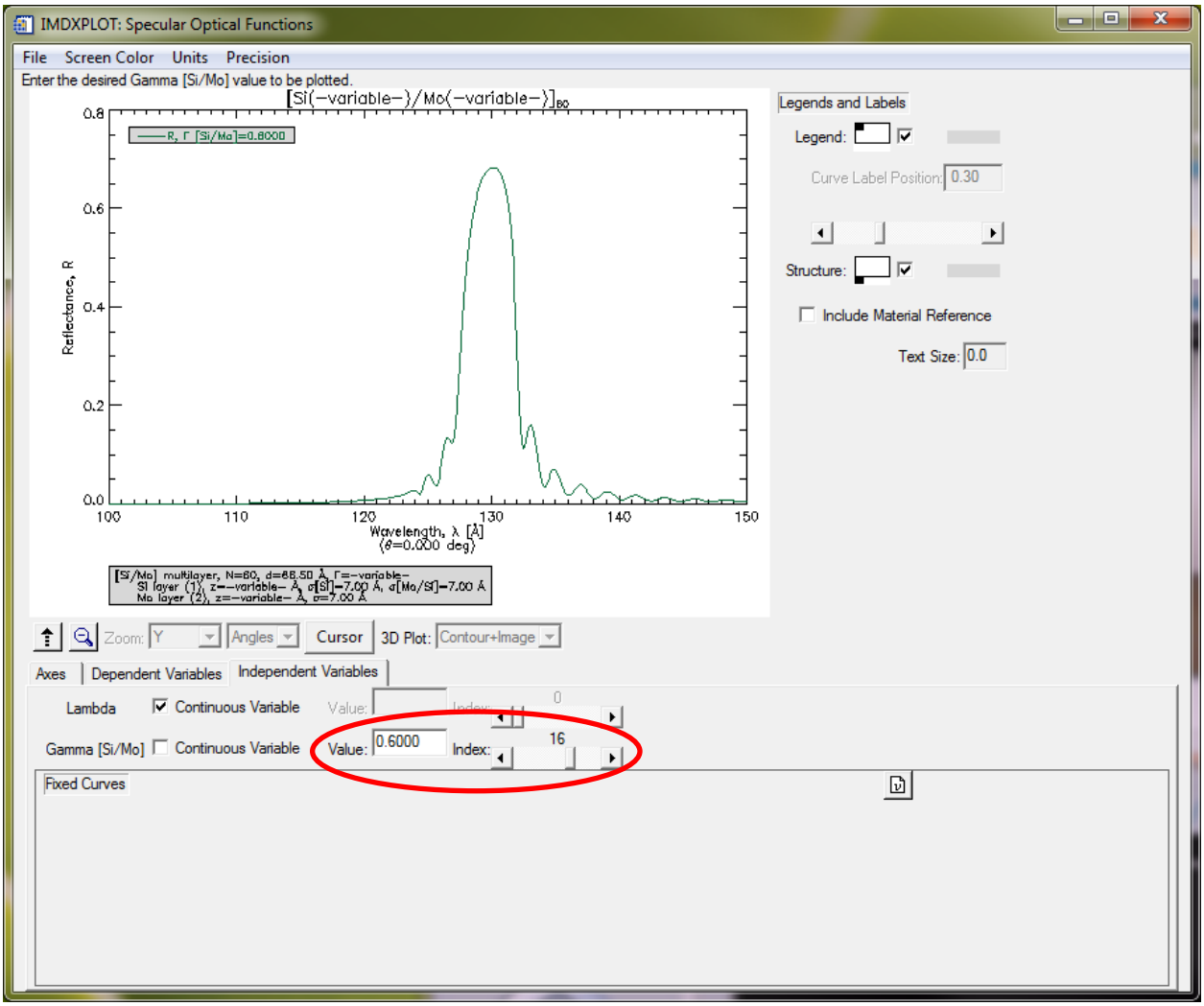
is, the color used to fill the box. You can also adjust the relative size of the text, as we've done in Figure 30 (we've set "Text Size" to 1.5). For computations of field intensity, there will also be controls for display of layers and interfaces when plotting Intensity vs. Depth, as we'll see later. If non-zero instrumental resolution parameters are specified, you can display a label for that too, and when a fit or optimization has been completed, there will be a label to display those details, as we'll see in Chapter 8.

6.3 IMDXPLOT: Independent Variables Tab

It's always possible to display the results of a computation as a function of just one independent variable: when there are multiple multi-valued independent variables (i.e., wavelength and Γ, in this example), you can choose to check either one or two of the boxes labeled "Continuous Variable" associated with those independent variables on the Independent Variables tab of the IMDXPLOT window. If two independent variables are selected as continuous variables (Figure 29, for example), then you'll see a 3D plot of the optical function, as in the above two figures. But if you select only one continuous variable, then you'll see a 2D plot. For example, here's how the IMDXPLOT window from Figure 29 looks after we uncheck "Gamma" as a continuous variable:



The plot now shows Reflectance vs. Wavelength, for a single Γ value – 0.2, in the figure above. You can control which Γ value is displayed using the "Index:" slider for that independent variable, or by entering the desired Γ value into the box labeled "Value:". (*For IDL users*: The number displayed above the slider is the array index; in the above example, we've selected index 0, so we're seeing a plot of R[*,0].) For example, here's the same IMDXPLOT window after we've selected Γ=0.6 (i.e., index=16):



6.4 IMDXPLOT: Fixed Curves

You can simultaneously display several 2D slices of multi-dimensional optical functions on the same plot in IMDXPLOT when there is more than one multi-valued independent variable defined, using so-called "Fixed Curves". When you click the "Add Fixed Curve Button" button ( ) near the bottom right of the IMDXPLOT window (highlighted below), IMDXPLOT will take a snapshot of the curve currently displayed and create a new Fixed Curve that is added to the list of Fixed Curves to be displayed.[vii] Here's our IMDXPLOT window after we've added a few different Γ values to the plot above using Fixed Curves:

[vii] If more than one dependent variable is selected for display on the Dependent Variables tab of IMDXPLOT, when the Add Fixed Curve button is clicked, individual Fixed Curves will be created for each displayed dependent variable.

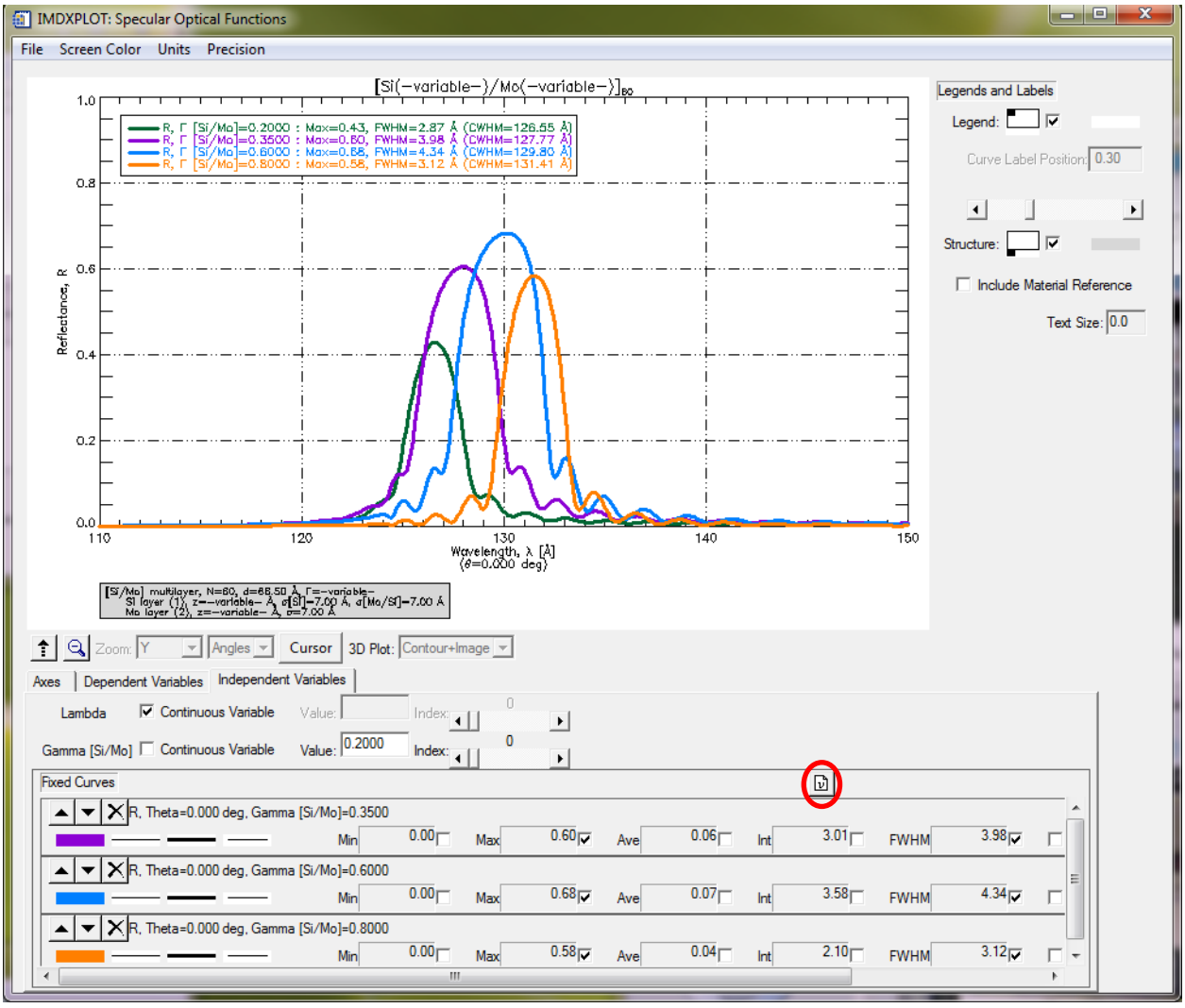
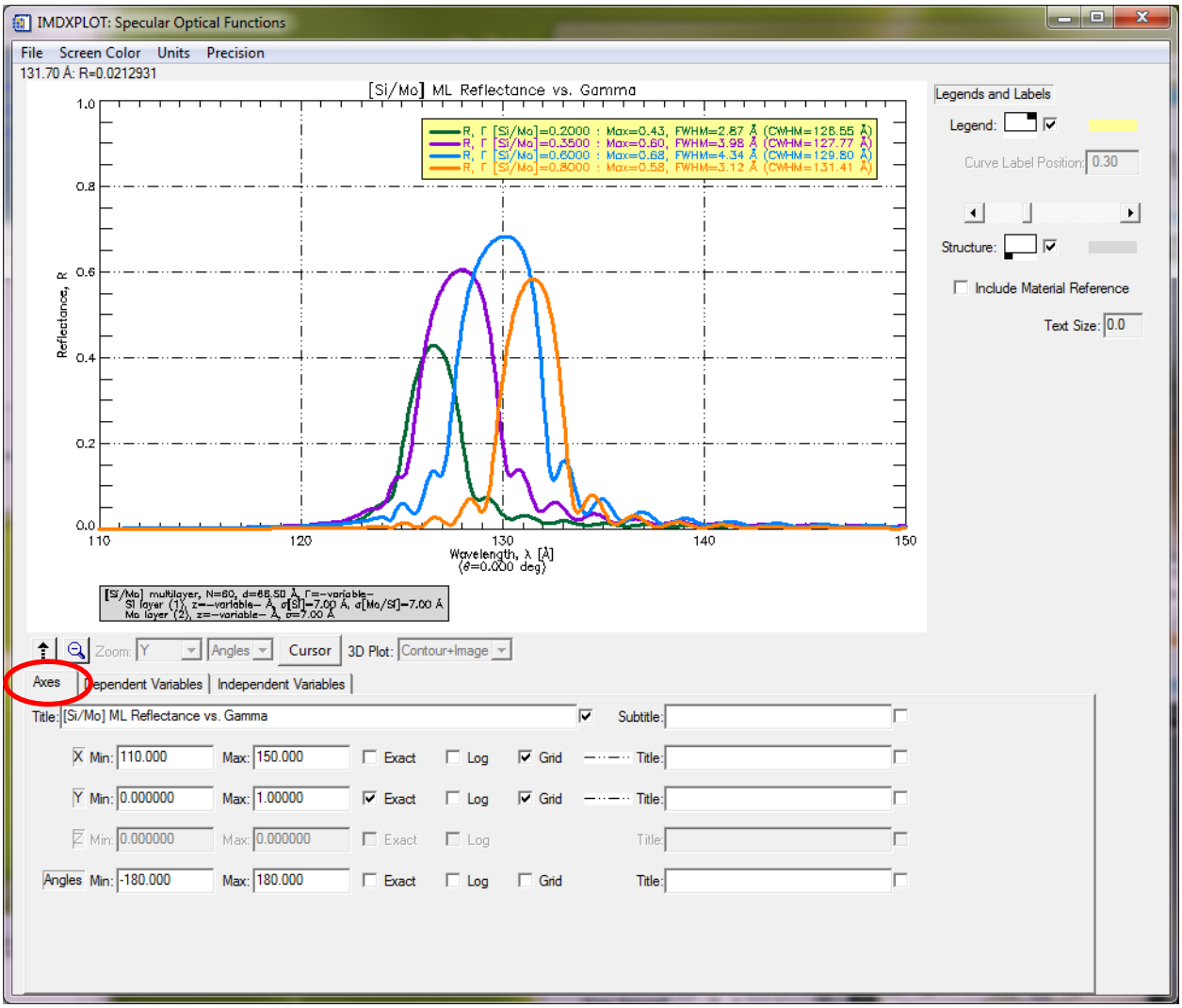


Figure 31. IMDXPLOT showing simultaneous display of several 2D slices of a 3D optical function, using Fixed Curves. The "Add Fixed Curve" button that is used to create a new Fixed Curve is circled in red.

Obviously, we've adjusted the colors of the different curves shown above to make a prettier picture, and we've also selected to display for each curve the maximum reflectance and the FWHM value, which are listed in the plot legend. We've also added grid lines to the plot, and have changed the line thicknesses. Controls for all these settings are located on the Axes and Dependent Variables tabs explained below.

6.5 IMDXPLOT: Axes Tab

The Axes tab on the IMDXPLOT window is where you control the plot axes: data ranges, grid lines, and axis titles. Here's another variation of the IMDXPLOT window from Figure 31, with the Axes tab displayed:



6.6 IMDXPLOT: Dependent Variables Tab

So far, we've been plotting Reflectance as the sole dependent variable of our Si/Mo multilayer example. But the "Dependent Variables" tab on the IMDXPLOT window allows you to (a) select for display any of the optical functions that were computed, and (b) control how those functions appear (i.e., color, line style, symbol, etc.) For example, here's how the IMDXPLOT window of Figure 31 looks after we've selected to suppress display of the average Reflectance, and selected for display both Reflectance and Phase for s-polarization (for Γ=0.6):

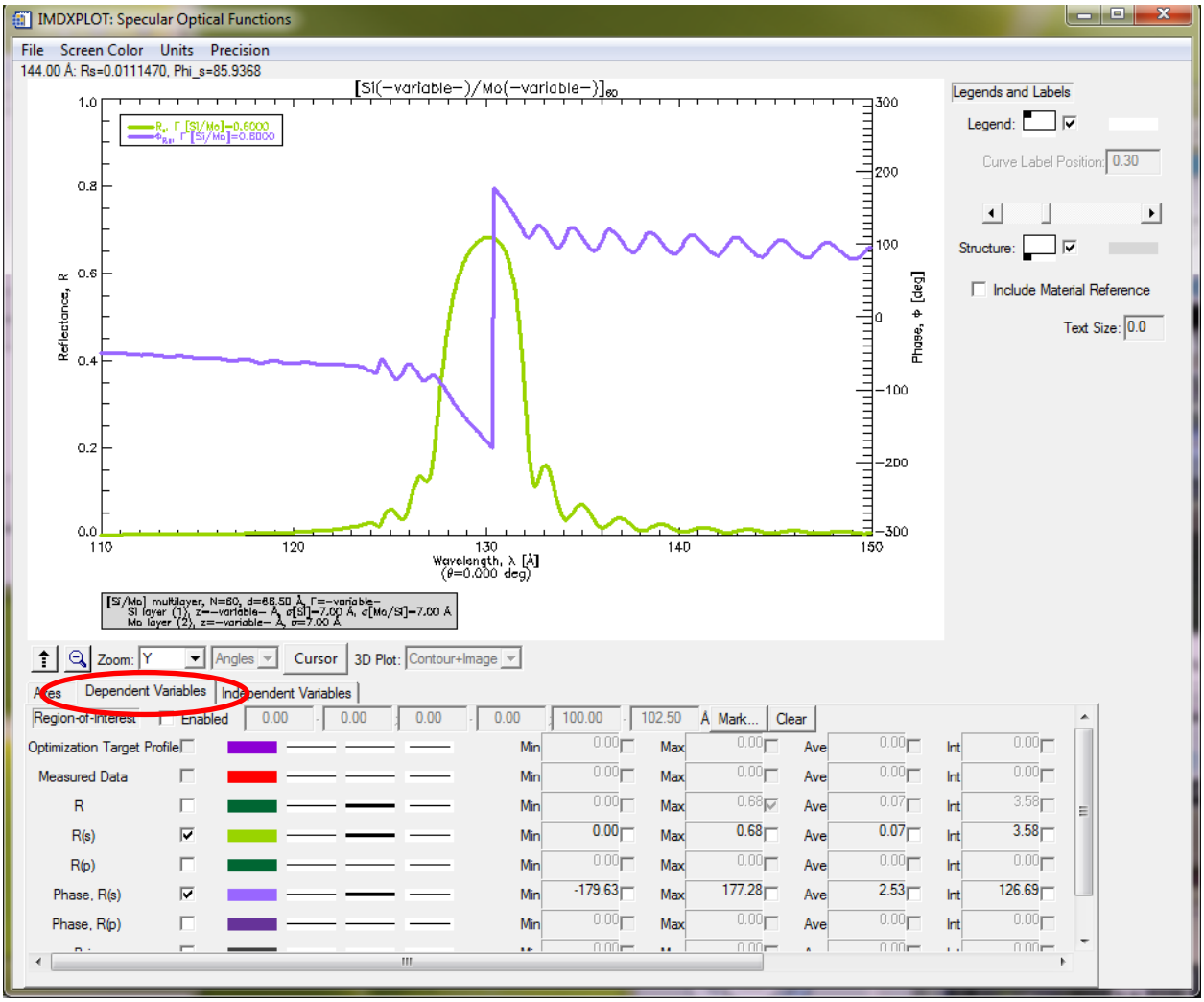


Figure 32. The IMDXPLOT Dependent Variables tab controls what functions are displayed, and how they appear. You can also specify a Region-of-Interest, and select statistics to be included on the plot for any variables being displayed.

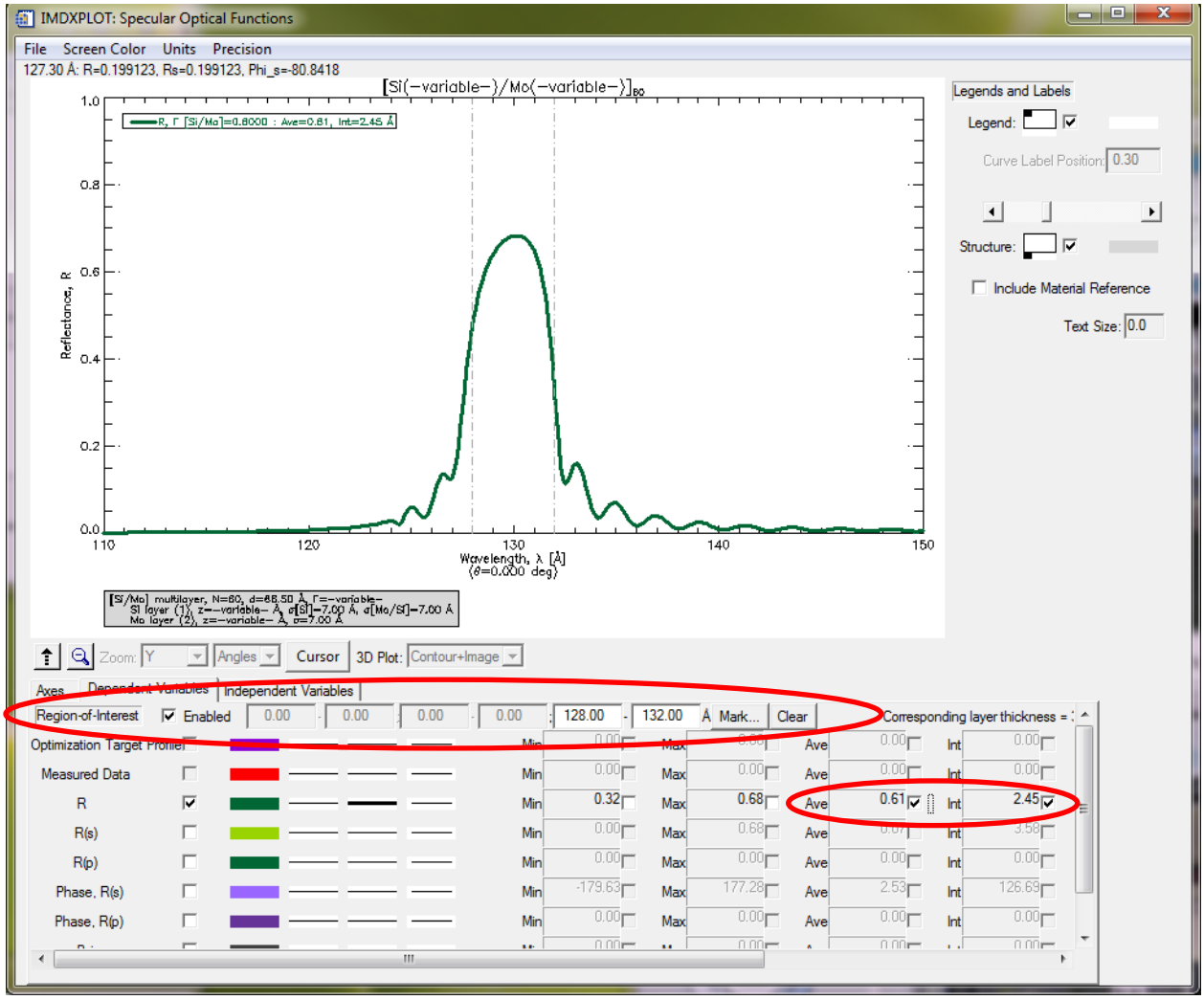
6.6.1.1 IMDXPLOT: Statistics

The Dependent Variables tab also includes a row of 'statistics' values for each optical function computed: Minimum value, Maximum value, Average value, Integral, FWHM (full-width-half-max), and FWHMin (full-width-half-min, for "absorption" features.) Any of these statistics can also be included in the plot, by checking the associated boxes, for any dependent variable selected for display. You can also display statistics for any Fixed Curves that have been defined; those controls are located on the Independent Variables tab in a row with each Fixed Curve you've created.

> **Note**: there are two check-boxes to the right of the FWHM and FWHMin labels in each row of statistics. The second check-box controls whether the FWHM/FWHMin is displayed in the plot legend or on the curve itself.

6.6.1.2 IMDXPLOT: Regions-Of-Interest

The Dependent Variables tab on the IMDXPLOT window allows you to define and display a Region-Of-Interest (ROI). When an ROI is defined – by entering 'start' and 'end' values directly, in the text-boxes described below, or interactively using the mouse after clicking the "Mark…" button – statistics are computed only over the ROI as specified. For example, in the next figure we've started from the IMDXPLOT window shown in Figure 30, have defined the ROI to extend from λ=128 Å to λ=132 Å, and have selected to display the Average and Integral Reflectance values, which are thus computed over that range; the ROI is displayed on the plot as dashed lines:



You'll notice that there are actually three pairs of 'start' & 'end' values (i.e., text-boxes) for the ROI, although the first two pairs are greyed-out in the figure above. What's going on is this: the last pair of ROI start & end values (set to 128 Å and 132 Å in the figure above) apply to the calculated optical functions that have been computed by IMD. In this example, those functions are Reflectance, Phase, Psi, Delta, etc.; the first pair of ROI start & end values are used only for the Optimization Target Profile, if

that's being displayed, while the second pair of ROI start & end values are used only for Measured Data, if that's being displayed. However these variables are not displayed (nor do they exist) in the above examples – in fact, we haven't even explained what those things are yet! But we'll get to that in the Chapter 8, and when we do, we'll further explain the ROI start & end values that apply to those quantities.

6.7 Printing Graphics from IMDXPLOT

Once you've configured IMDXPLOT to display the computation results as you like, you can print a hardcopy, or print the results to a file, using the "File→Print…" menu option in IMDXPLOT. For example, you can print to a PostScript (.ps) file, which can then be converted to any number of other graphics file formats using freely available programs (such as GhostScript, ImageMagick, etc.), or imported into vector graphics programs (such as Corel Draw, Adobe Illustrator, etc.) for conversion and/or further manipulation. To illustrate, here's a figure that was created by importing into CorelDraw a PostScript file that was created by printing from the IMDXPLOT window shown above in Figure 31; the plot title was adjusted manually in Corel, and then the graphics were exported as a JPEG, for import into this Word document:

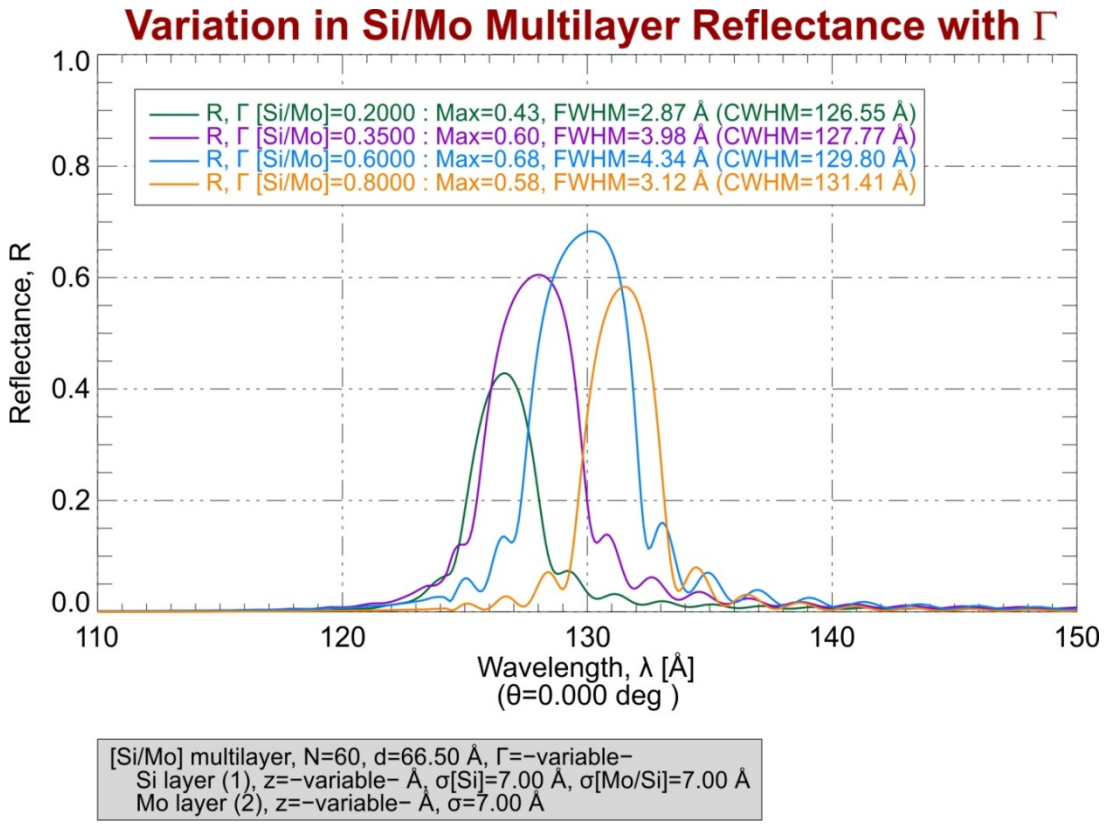


Figure 33. This figure was created from the IMDXPLOT window shown in Figure 31: we used File→Print… from the IMDXPLOT menu bar and saved the plot to a PostScript (.ps) file. The PostScript file was then imported into CorelDraw, the title adjusted, and then exported as a JPEG. The JPEG was imported into the Word document used to create this PDF.

6.8 "What happens in IMDXPLOT stays in IMDXPLOT!"

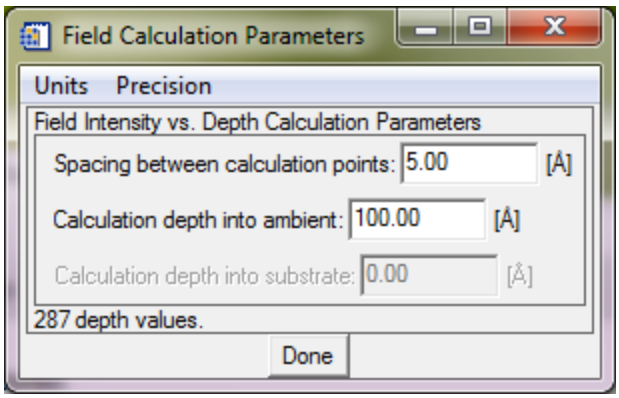
As we've seen, a new instance of IMDXPLOT opens whenever an IMD computation is completed. You can open additional instances of IMDXPLOT showing the same results from the "Plot" menu on the main IMD window after a computation is complete, i.e., using "Plot→Specular Optical Functions…", "Plot→Fields…", etc. It's important to understand that each time a new IMDXPLOT window is created, a private copy of all IMD variables and settings currently defined in the main IMD window are passed to that instance of IMDXPLOT. Any subsequent changes you make to the plot appearance in IMDXPLOT remain local to that instance of IMDXPLOT; if you create a new instance of IMDXPLOT from the main IMD window, any changes to the plot appearance made in the first instance of IMDXPLOT will not be applied. Likewise, any changes you make in the main IMD window have no effect on any existing IMDXPLOT windows.

If you do want changes to the plot appearance made in an IMDXPLOT window to be applied globally, select the IMDXPLOT menu option "File→Send Back to IMD…"; any changes you've made in IMDXPLOT will then be applied to any new instances of IMDXPLOT you subsequently create using the Plot menu from the main IMD window. Keep in mind, however, that if you do use the "File→Send Back to IMD…" menu option from IMDXPLOT, all settings and preferences in the main IMD will be over-written with the data being sent from IMDXPLOT.

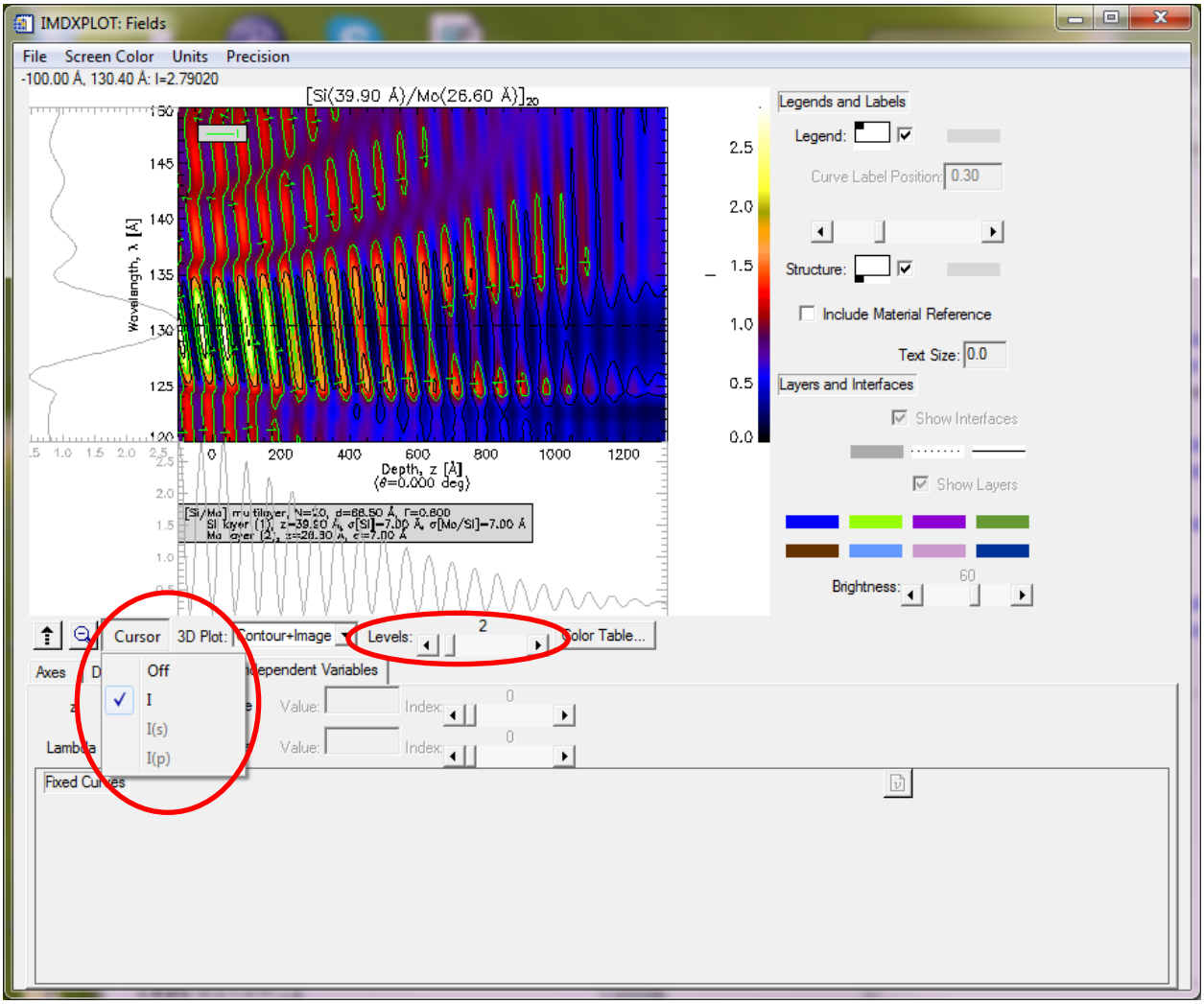
> **Tip**: You're allowed unlimited simultaneous instances of IMDXPLOT, but you can only open a single instance of the main IMD window. If you need to go back and forth between several different IMD computations, keep each computation open in its own IMDXPLOT window and make use of the "File→Send Back To IMD…" menu option in IMDXPLOT to easily switch between computations.

6.9 Example: Si/Mo Multilayer Field Intensity

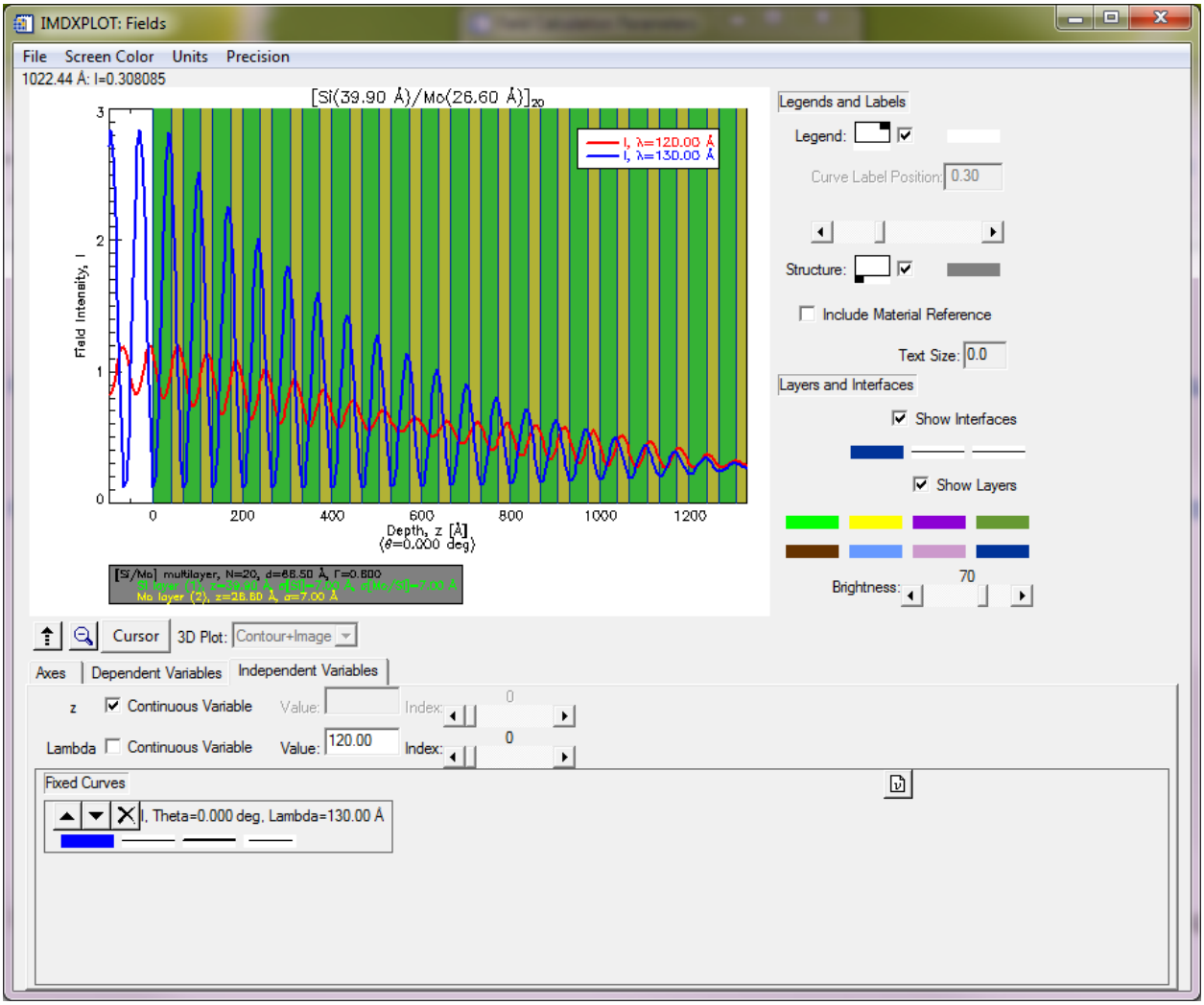
In this example (`imd/examples/Fields/Si_Mo_fields.imd`) we'll compute the electric field intensity as a function of both wavelength and depth in a Si/Mo periodic multilayer. This structure is defined with N=20 repetitions of a Si/Mo bilayer having a period d=66.5 Å, and layer thickness ratio Γ=0.6. We'll specify an incidence angle of θ=0° from normal, and 301 wavelengths from λ=120 to 150 Å. We'll check the box labeled "Fields" on the IMD Dependent Variables tab, and in the Field Calculation Parameters window we'll specify 5 Å spacing between calculation points, with the calculation extending 100 Å into the ambient:



After performing the computation ("File→Calculate Specular Optical Functions/Fields…"), we configure the resulting IMDXPLOT window to display a contour map of Reflectance as a function of both Wavelength and Depth, with 2 contour levels. We also enable the Cursor from the menu, so we can view 1D slices (in gray) as we move the mouse over the plot:



Here's another way to visualize some of these same results: here we're plotting Field Intensity vs Depth for two discrete wavelengths (using Fixed Curves):



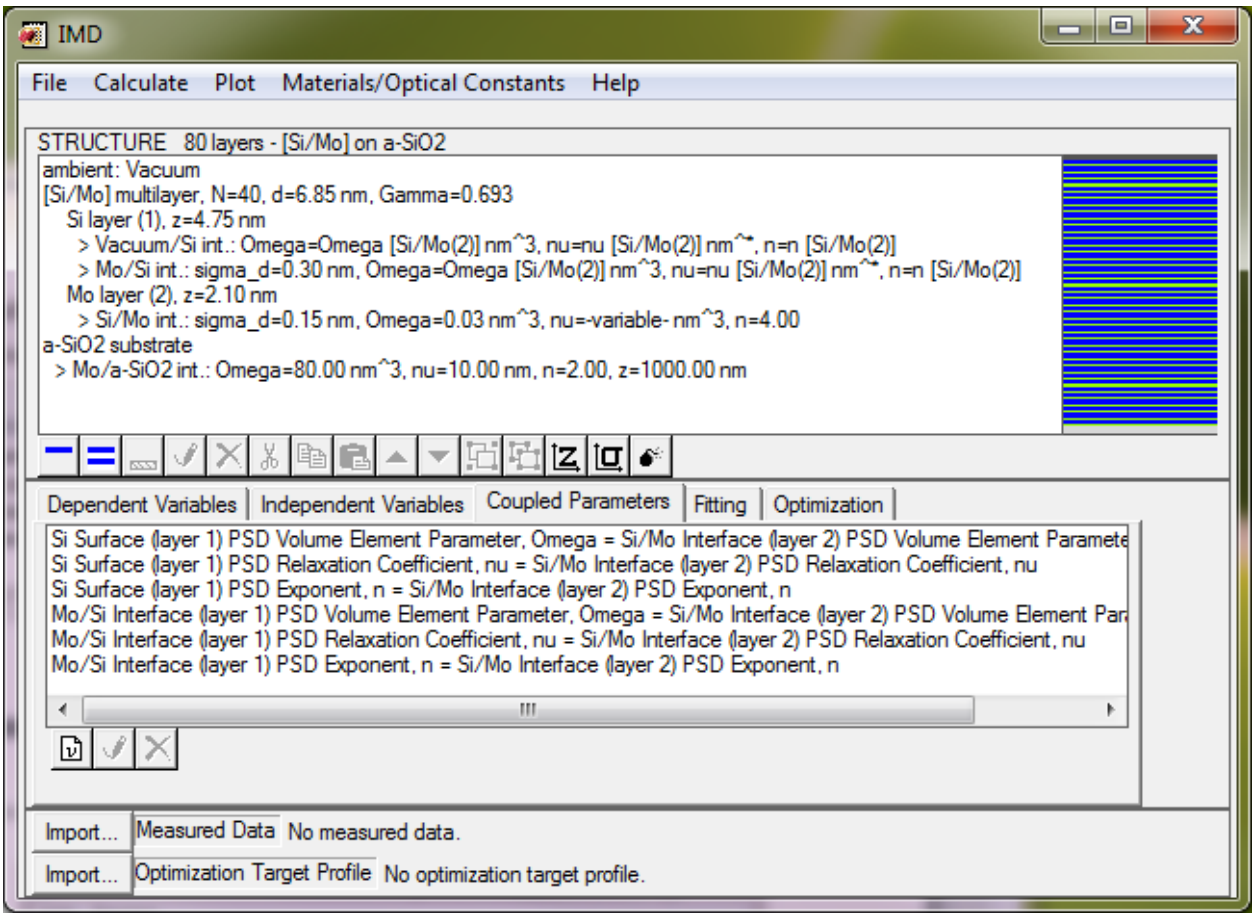
We've used the controls on the right of the IMDXPLOT window above to adjust the display of layers and interfaces, and have adjusted the plot colors, as well as the legend and structure labels.

6.10 Example: Si/Mo ML Normal-Incidence Non-Specular Reflected Intensity

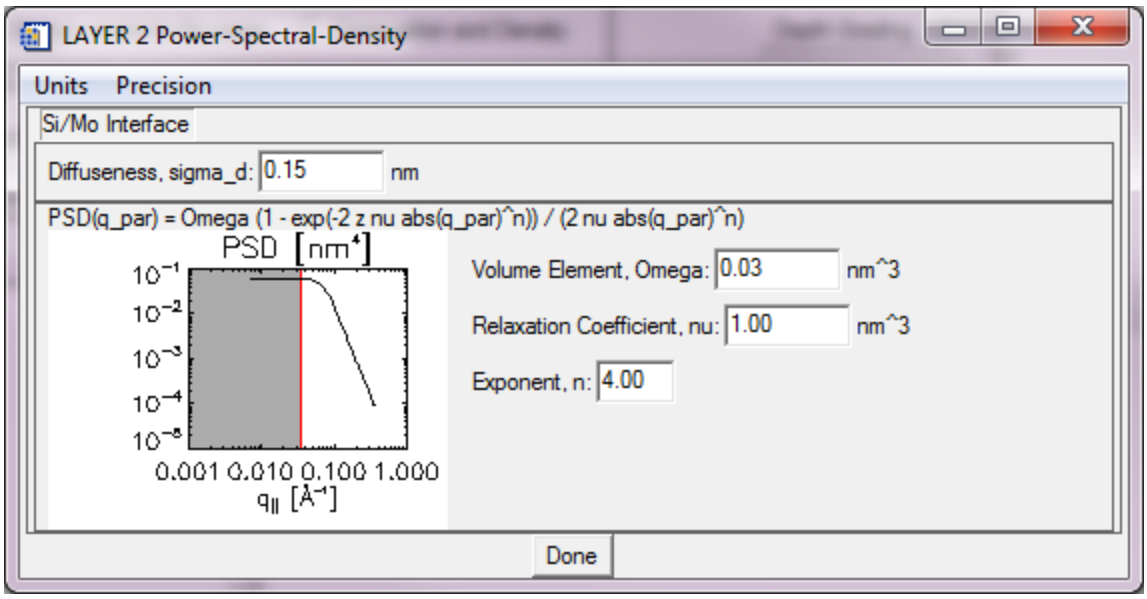
Our last example of this chapter is complicated: we're going to compute the non-specular reflected intensity near normal incidence for a Si/Mo periodic multilayer. We'll specify a Detector Scan (§5.2.5.2) with the Scattering Angle defined as a multi-valued independent variable, and we'll compute the scattered intensity as a function of wavelength. We'll use the Born Approximation formalism described in §5.1.2, and the $\Omega/\nu/n$ PSD model described in §4.11.2. We're going to use six coupled parameters, in order to link the three PSD parameters for the Si surface and the three PSD parameters for the Mo-on-Si interfaces to the three PSD parameters for the Si-on-Mo interface. And finally, we're going to define the

PSD parameter, ν, the relaxation coefficient, as an independent variable: that makes three multi-valued independent variables (i.e., Wavelength, Scattering Angle, and Relaxation Coefficient.) IMD files associated with this example are located in the `imd/examples/Non-Specular Scattering` directory.

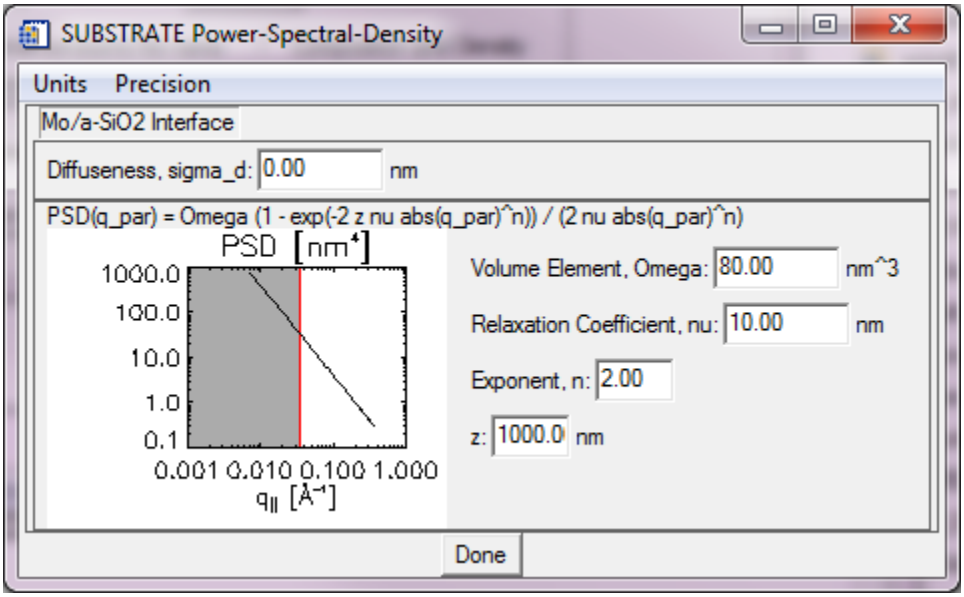
Here's a view of the main IMD window, showing the Coupled Parameters tab listing the six coupled parameters we just described:



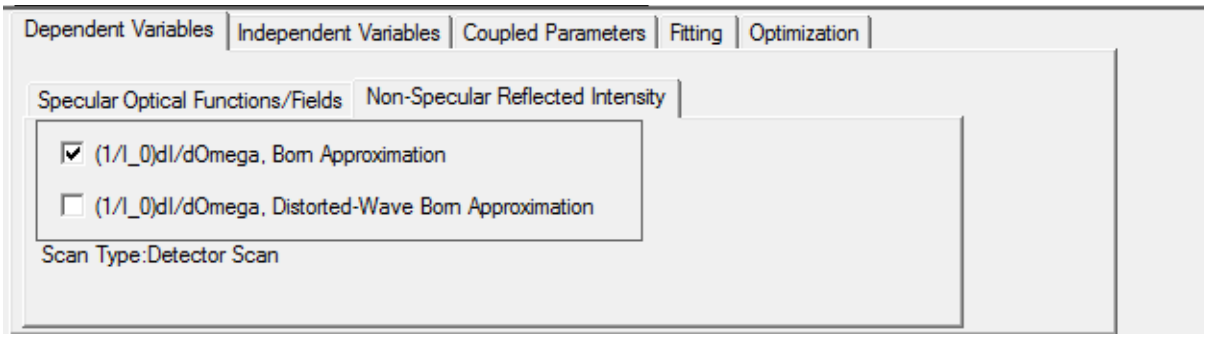
Again, the three PSD parameters describing the Si surface and the three PSD parameters describing the Mo/Si interfaces are all linked (via Coupled Parameters) to the three PSD parameters describing the Si/Mo interfaces. The PSD parameters for the Si/Mo interfaces are configured like this:

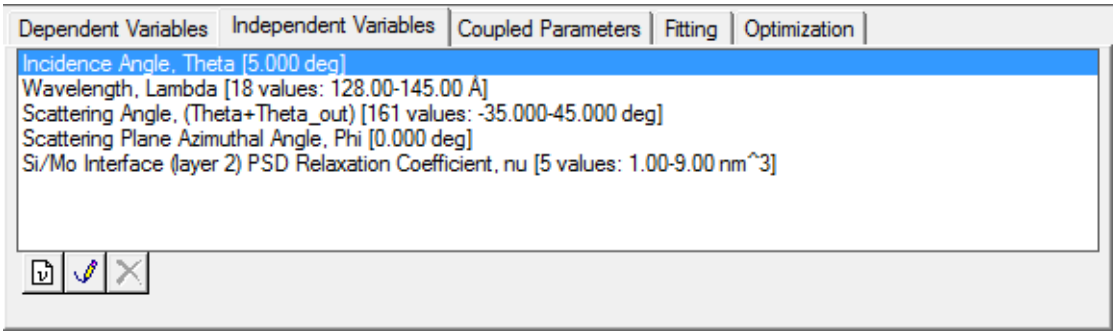


We also need to specify the PSD of the substrate:

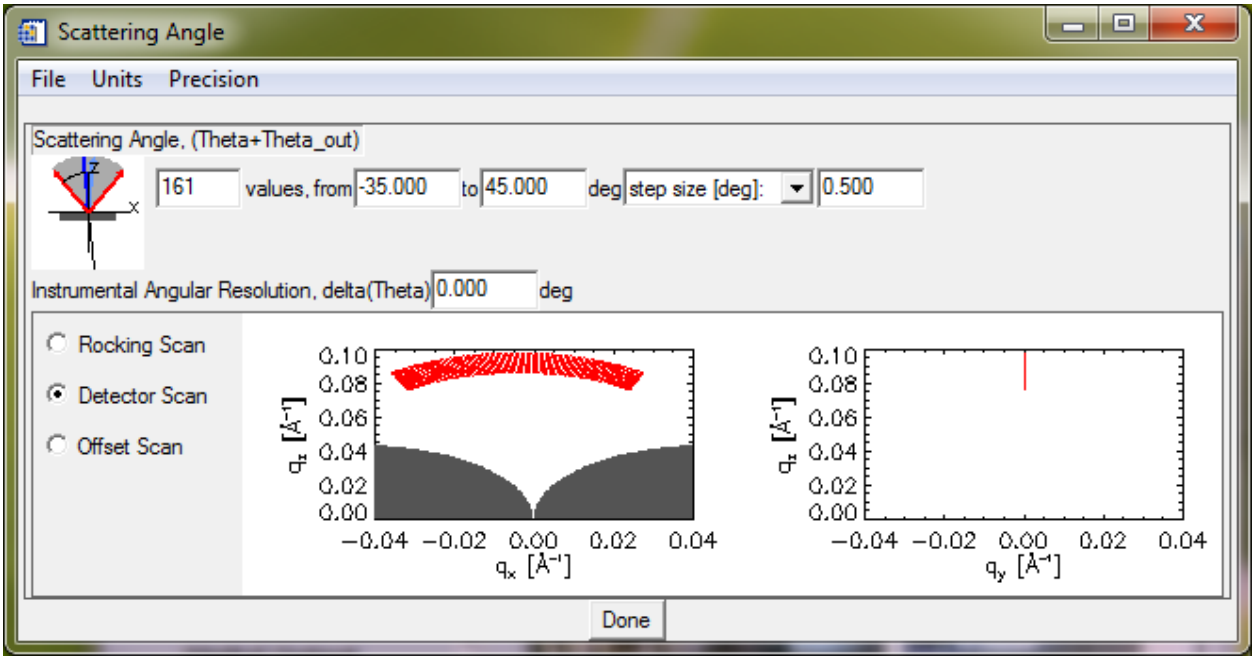


Here are the Dependent Variables and Independent Variables tabs:

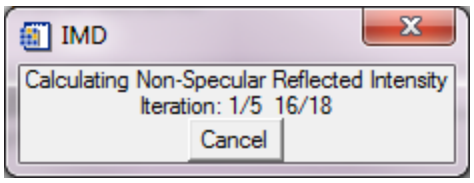




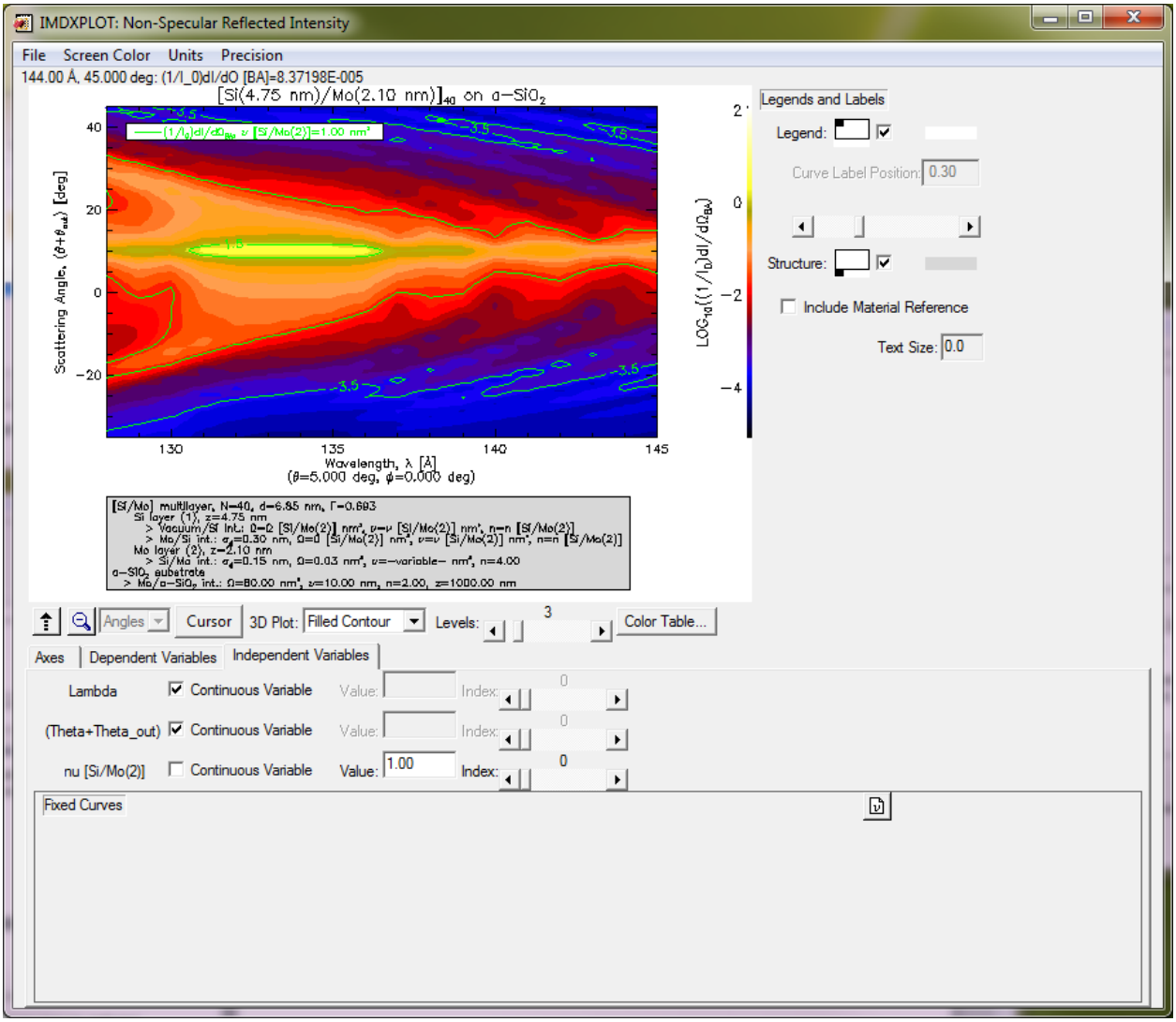
The Scattering Angle Independent Variable window, which shows the region of q-space we'll be exploring in red, looks like this:



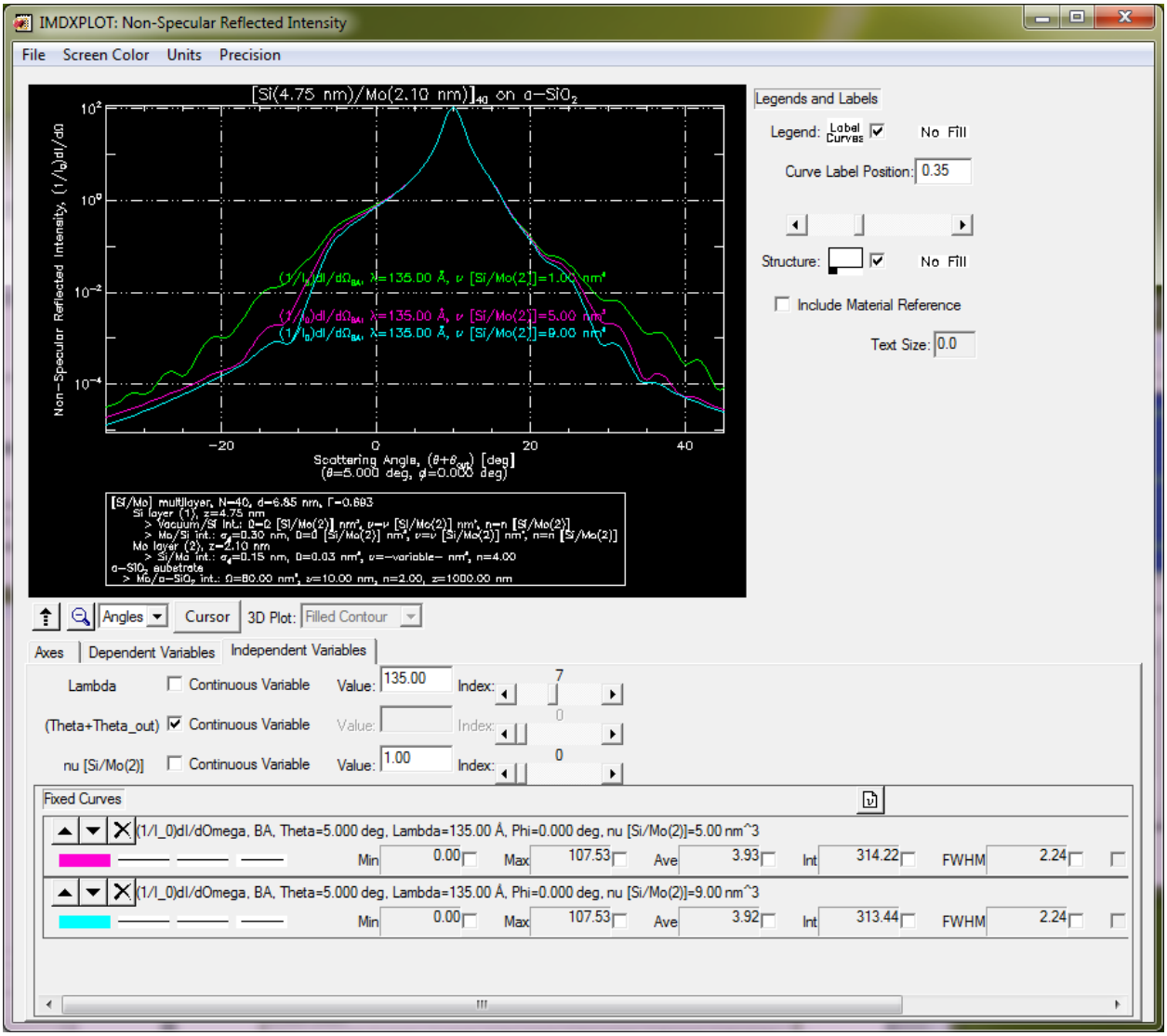
With the model and variables defined, we'll select "Calculate→Non-Specular Reflected Intensity…" from the IMD menu bar, and the calculation begins. This computation will take a while:



When the computation is completed, a new IMDXPLOT window opens. The next figure shows how that IMDXPLOT window looks after we've specified a 3D plot of Reflected Intensity vs. Scattering Angle vs. Wavelength. You can use the Index slider to step through different ν values, to visualize how the distribution and intensity of scattered light changes as a result:



Here's another view of these results, where we're now making a 2D plot of Non-Specular Reflected Intensity vs. Scattering Angle at a wavelength of λ=135 Å, for three different ν values:



In the figure above, we've used the "Screen Color" menu from the IMDXPLOT menu bar to display a black background (instead of the default white background), we've used two Fixed Curves so as to display the results for three different ν values simultaneously, and we've opted to label the three curves directly instead of listing the curve descriptions in a legend, which is the default setting.

7 Saving and Exporting Computation Results

7.1 Saving to a .imd File

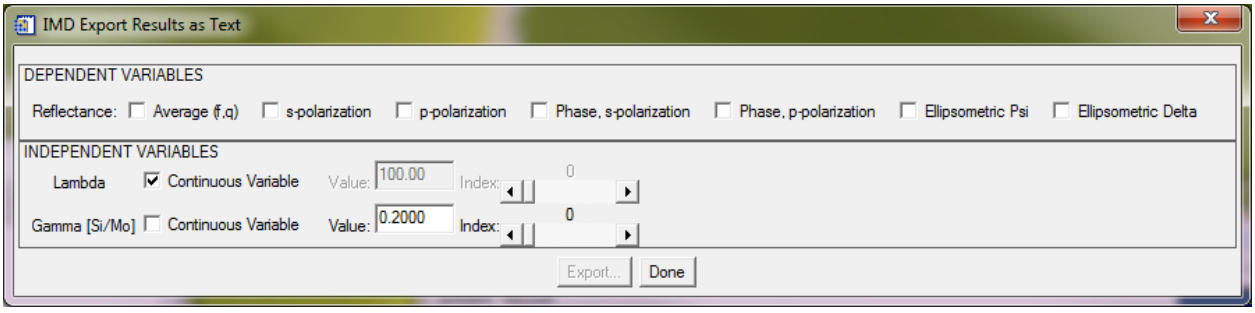
Once an IMD computation is complete, IMDXPLOT opens by default to display the results, as we've just seen in the previous chapter. From IMDXPLOT you can print a hardcopy of the graphics, or print those graphics to a file. To save all the data resulting from a computation to a `.imd` file that can later be re-opened in IMD, use the "File→Save…" menu option from the main IMD window. Use "File→Open…" to re-open previously saved `.imd` files.

7.2 Saving to a .imd File from IMDXPLOT

If you make any changes to the way the computation results are displayed in IMDXPLOT (as in the previous chapter), you can save to a `.imd` file directly from IMDXPLOT (using "File→Save As…" from the IMDXPLOT menu bar) and those changes will be preserved in the `.imd` file you create. So, if you later re-open that `.imd` file in IMD your plot will appear (almost[viii]) exactly the way it did when you originally saved the file from IMDXPLOT.

7.3 Exporting Results to Plain Text Files

You can export one-dimensional slices of the optical functions you compute in IMD to plain text files, using the "File→Export Results As Text…" menu option from the main IMD window. Plain text files can them be imported into other programs you may wish to use. Here's how the IMD Export Results as Text window first looks for the example from §6.1, where we computed Reflectance vs. Wavelength vs. Γ:

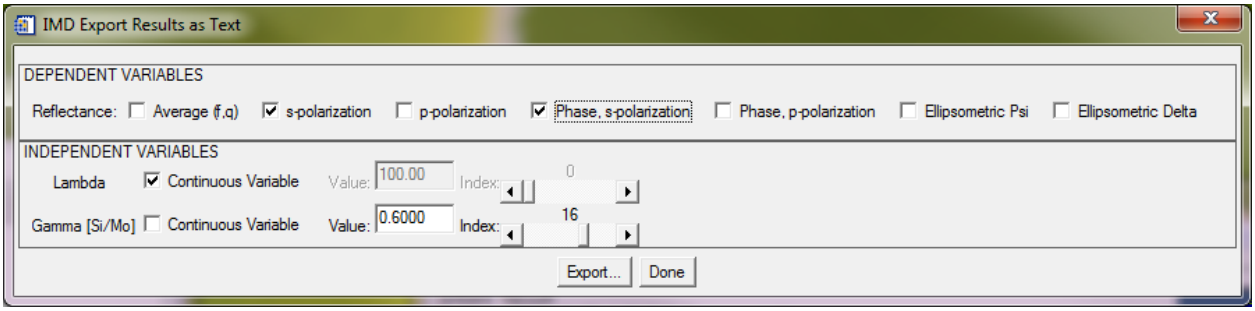


As it's only possible in IMD to export to a text file one-dimensional slices of the optical functions that you compute, you must select one and only one Independent Variable to be continuous; you must then select single-values for any other multi-valued Independent Variables that are defined before clicking the "Export" button. Selection of independent variables on the IMD Export Results as Text window is

[viii] Any changes made to the color table used to display contour plots are not preserved in .imd files.

analogous to independent variable selection on the Independent Variables tab in IMDXPLOT: you can enter values directly, or use the Index slider.

As an example, here's how that same IMD Export Results as Text window looks after we've selected Γ=0.6 (Index=16), and have also checked the dependent variable boxes for Reflectance and Phase, both for s-polarization:



If we now click the "Export…" button on the IMD Export Results as Text window, the selected slice of data will be saved to the file that we specify:

```
;
;  IMD Data
;
;  Saved on Thu Aug 15 14:09:31 2013
;
;  Structure:
;
;  [Si/Mo] multilayer, N=60, d=66.50 A, Gamma=-variable-
;       Si layer (1), z=-variable- A, sigma[Si]=7.00 A, sigma[Mo/Si]=7.00 A
;       Mo layer (2), z=-variable- A, sigma=7.00 A
;
;
;
;  Theta=0.000 deg, Gamma [Si/Mo](Multilayer)=0.6000
;
;  Lambda [A]       R(s)      Phase, R(s) [deg]
       100.00000    0.00032320628       -36.263521
       100.10000    0.00032777393       -36.403341
       100.20000    0.00033221573       -36.557062
       100.30000    0.00033647271       -36.715071
       100.40000    0.00034053940       -36.867794
       100.50000    0.00034442416       -37.016089
       100.60000    0.00034827774       -37.144325
       100.70000    0.00035222108       -37.251689
       100.80000    0.00035637293       -37.342096
       100.90000    0.00036090689       -37.427247
       101.00000    0.00036577321       -37.513217
       101.10000    0.00037094601       -37.609684
       101.20000    0.00037634420       -37.723817
       101.30000    0.00038184863       -37.858591
       101.40000    0.00038733021       -38.012355

       etc
```

7.4 Exporting Results to IDL Save Files

The "File→Export Results to IDL Save File…" option on the IMD menu bar allows you to save independent and dependent variables computed in IMD to a binary IDL Save file (`.sav`).[ix] An IDL save file can be restored during any IDL session using the IDL "RESTORE" command. Exporting to IDL Save

[ix] While `.imd` files are actually IDL Save files, they contain a great deal of additional data for internal use only along with the IMD dependent and independent variables.

files may be useful when you want to use IMD results in other IDL programs, particularly multi-dimensional dependent variables that cannot be saved in a single text file as described in the previous section. If you're running IMD as an IDL application, you can also access IMD common block variables directly from the IDL command line. Please consult Chapter 11 for more details on IMD variable names.

8 Measured Data, Curve-Fitting and Optimization

You may want to import your own measured data into IMD, for modeling and curve-fitting. In this chapter we'll explain how to do all that. We'll also explain how to perform film stack optimization, which is almost the same thing as curve-fitting: in the case of film stack optimization, layer thicknesses are adjusted in order to reach a target profile that you specify. We'll also explain how to import target profiles in this chapter, and how to build a target profile from scratch directly in IMD.

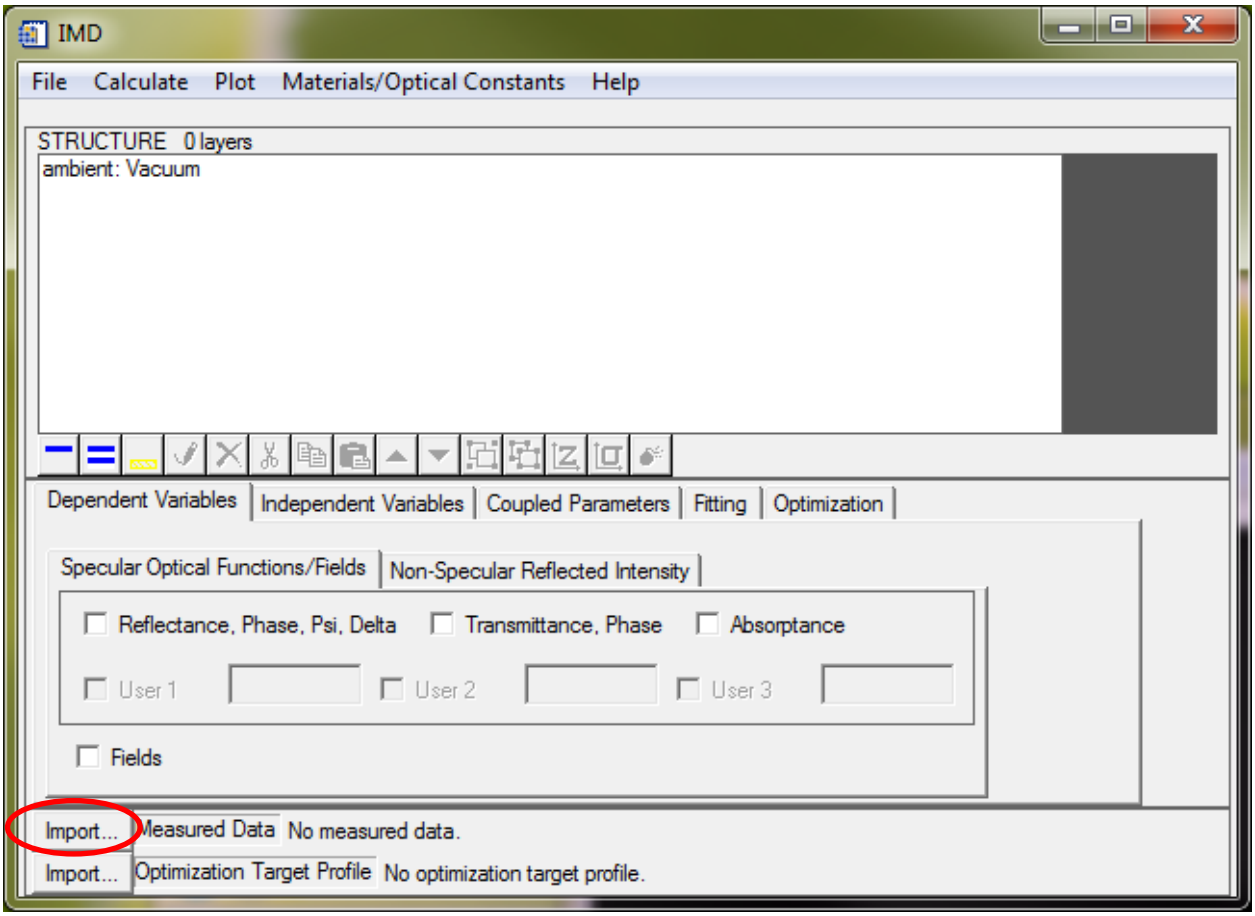
8.1 Measured Data

In IMD, "measured data" means a set of specular or non-specular optical function values that you have measured as a function of some independent variable. For example, measured data might refer to measurements of Reflectance vs. Incidence Angle (e.g., XRR), Transmission vs. Wavelength, Non-Specular Reflected Intensity vs. Scattering Angle, and so forth. In this section we'll explain how to import into IMD data files containing measured data, and we'll illustrate using some examples.

8.1.1 Importing Measured Data Files: X_m, Y_m and SIGY_m

The variable names for measured data that IMD uses internally are X_m for the independent variable, Y_m for the dependent variable, and SIGY_m for the experimental uncertainty in Y_m. That is, the value of the dependent variable is equal to Y_m ± SIGY_m. X_m can correspond to Wavelength (in Å), Incidence Angle (in degrees from normal), or to any other independent variable you may have defined in IMD. The goal of importing measured data into IMD is to properly define X_m and Y_m, and optionally SIGY_m. (There are some limitations on Curve-Fitting if you don't provide SIGY_m values, as we'll explain in §8.5.) Defining X_m, Y_m and SIGY_m is typically done using IMD's built-in file-import capabilities, but if you're running IMD as an IDL application, you can also use whatever techniques you like to define these variables using your own IDL commands and programs. See §11.4 for more details about importing measured data using your own IDL programs.

To import a measured data file using IMD's built-in file-import capabilities, select from the IMD menu bar "File→Import Measured Data…", or click the button labeled "Import…" at the bottom of the IMD window, to the left of the "Measured Data" label:



The IMD Import Measured Data window will appear, as in Figure 34. The Import Measured Data window comprises three main sections: the Graphics area, where measured data will be displayed, the File area, where you specify what files to read and how to read them, and the Variables area, where you specify which Independent and Dependent variables are being imported, how the imported data should be scaled, and if any portion of the imported data should be ignored.

We'll now illustrate the use of the controls on the Import Measured Data window using data files contained in the `imd/examples/Measured Data` directory. You can open the relevant `.imd` files in that directory if you want to follow along. However, please note that these `.imd` files reference the measured data files located in that directory, and the references are specified without the full path names to the files – full path names are operating-system dependent, and also depend on where you've installed IMD on your computer, so there's no way to provide an example file with a full path name that will work with any IMD installation. Therefore, to use these `.imd` files, **be sure to set `imd/examples/Measured Data` as your working directory in IMD**, using "File→Change Working Directory…" from the IMD or IMD Launcher menu bars, or by using the IDL 'cd' command at the IMD> prompt.

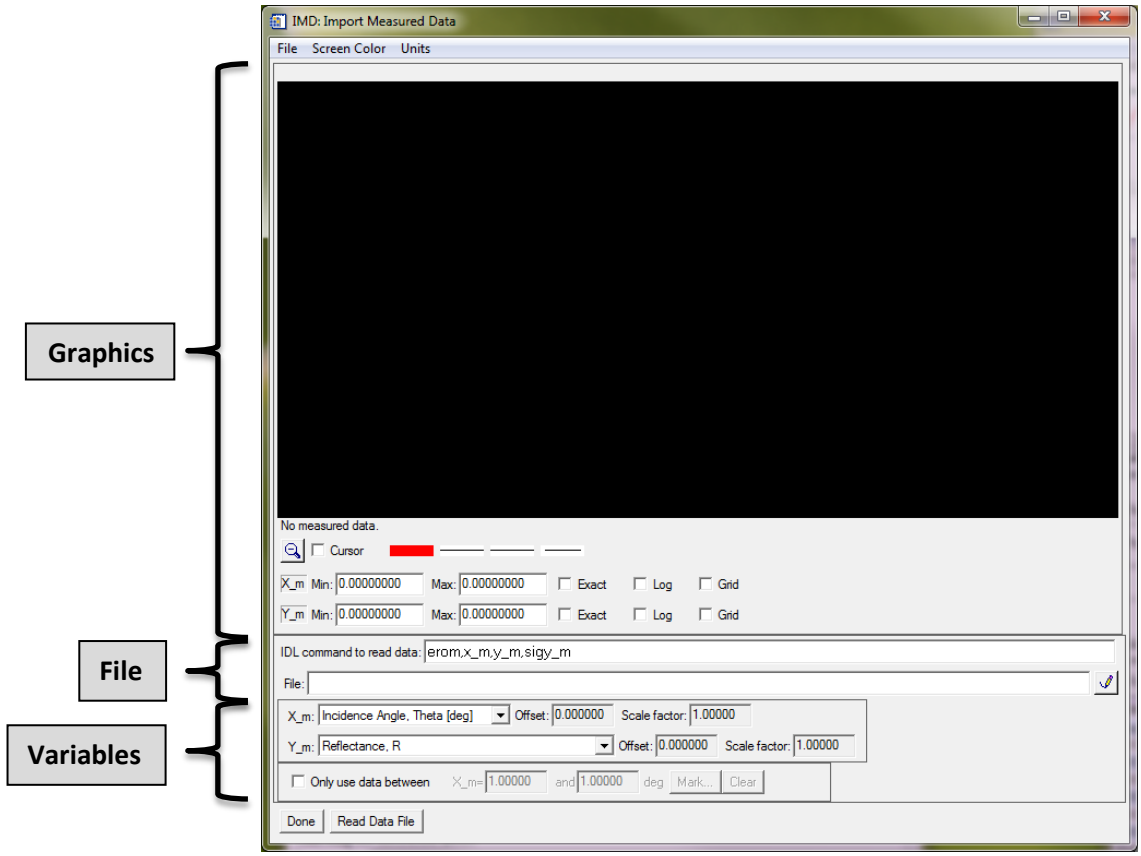
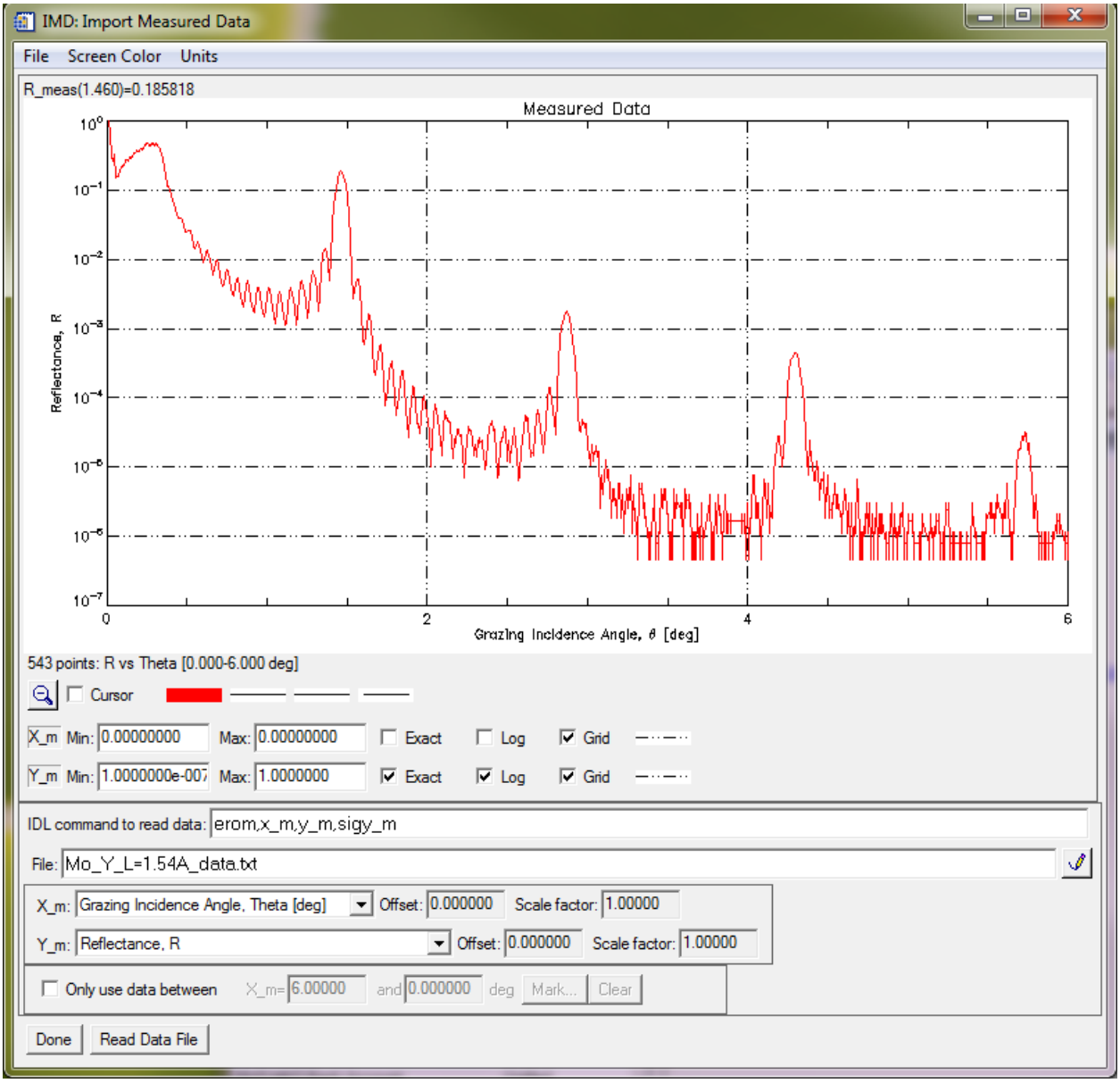


Figure 34. The IMD Import Measured Data window.

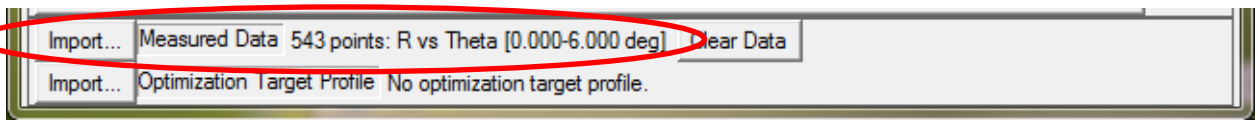
The File area of the Import Measured Data window identified in Figure 34 contains two boxes: the box labeled "IDL command to read data:" is where you type the IDL command used to open your specific measured data file, and the box labeled "File:" is where you type the name of the file to read. You can also select a file interactively, by clicking the button on the right of the "File:" box that looks like this:  .

The default IDL command to open Measured Data files is the "erom" command (§11.4.1), using the syntax "erom,x_m,y_m,sigy_m". This command will open a plain text file containing three columns of data corresponding to the X_m, Y_m, and SIGY_m variables. There are other ways to use the erom command as well, or you can use other commands in place of erom to read your data files: the programs rd_als, rd_reflect, and rd_spec are compiled into IMD, and these programs are explained in Chapter 11; if you are using IMD as an IDL application you can also use your own IDL programs. We'll show another example in the next section that uses the rd_als command, but see §11.4 for all the details on importing measured data files. In any case, for this example we'll just use the default erom command to open a plain text file containing X-ray Reflectance vs. Grazing Incidence Angle data (i.e., XRR data) for a Mo/Y multilayer – this file is called "`Mo_Y_L=1.54A_data.txt`" and is located in the `imd/examples/Measured Data` directory.

The next figure shows the IMD Import Measured Data window after we've imported the Mo/Y XRR data just described, by typing the file name into the "File:" box and then clicking the "Read Data File" button. We've also used the "Units→Angles" menu option to select units of "deg, Grazing", and have then selected "Grazing Incidence Angle, Theta [deg]" for X_m, and "Reflectance, R" for Y_m in the Variables area of the window. Finally, we've adjusted the controls in the Graphics area to display the data with grid lines and a logarithmic Y_m axis:



After successfully importing the data, the bottom of the main IMD window indicates that we've imported 543 data points, extending from θ=0° to 6°:



Use the "Clear Data" button on the main IMD window to delete Measured Data if necessary.

8.1.2 Importing Measured Data Files in IMDXPLOT

You can import measured data directly into an open IMDXPLOT window, using the "File→Import Measured Data…" option from the IMDXPLOT menu bar. This will open a private copy of the Import Measured Data window: any data imported into IMDXPLOT this way will be local to that instance of IMDXPLOT, and will not appear in the main IMD window … unless you later use the "File→Send Back to IMD…" menu option in IMDXPLOT, as explained in §6.8.

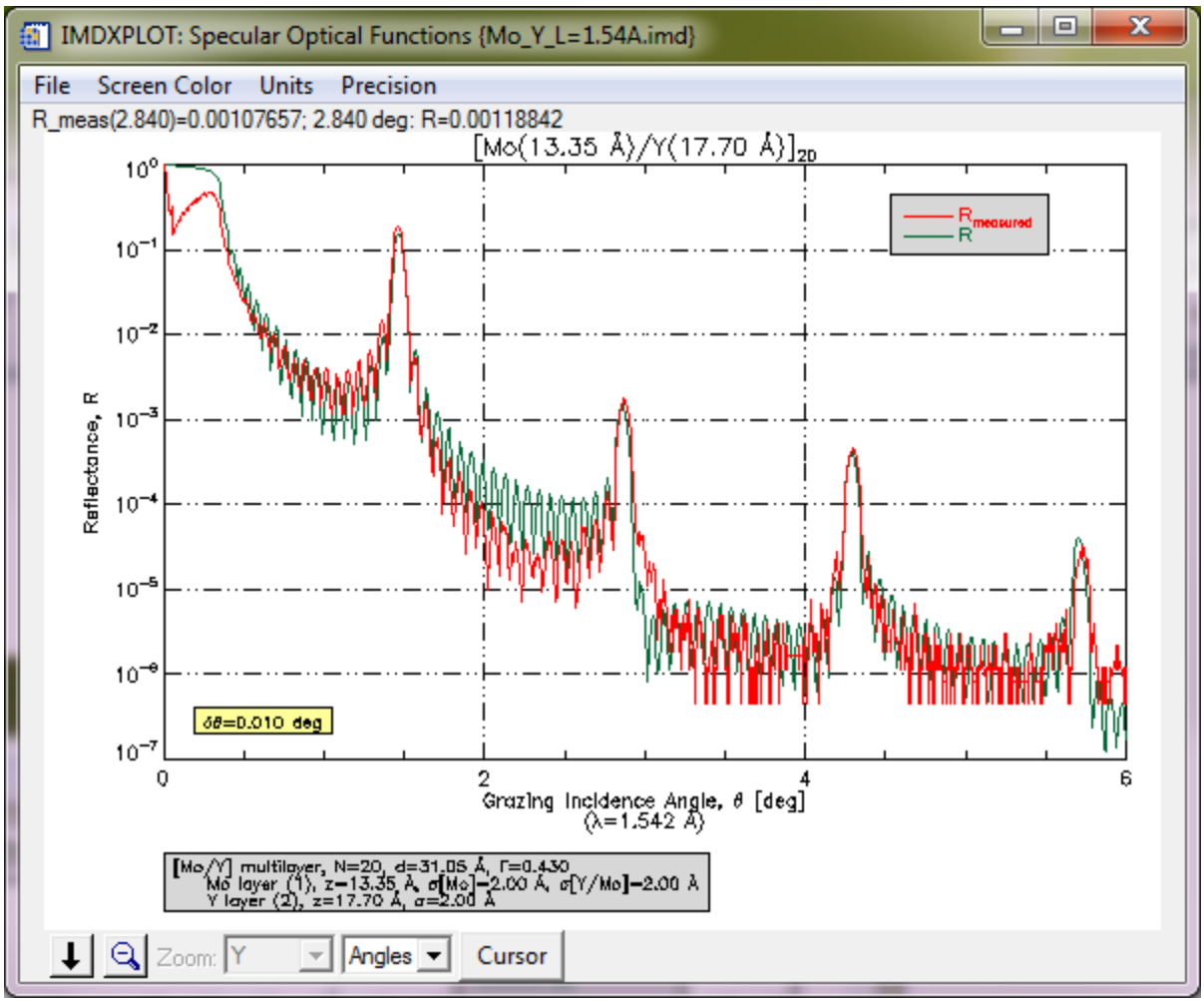
If measured data is already defined in IMDXPLOT, then you can also use the "File→Reload Measured-Data File" menu option in IMDXPLOT – this will cause IMDXPLOT to quietly re-read the measured data file you've specified, without opening the Import Measured Data window again.

If you're using IMD to model measured data as it's being acquired in real time by your computer, you can also use the "File→Watch Measured-Data File" menu option in IMDXPLOT: when enabled, IMDXPLOT will repeatedly re-read the measured data file specified, with a time interval between reads equal to the value that you specify from the "File→Poll Time…" menu option (default=10 seconds.)

8.2 Modeling Measured Data

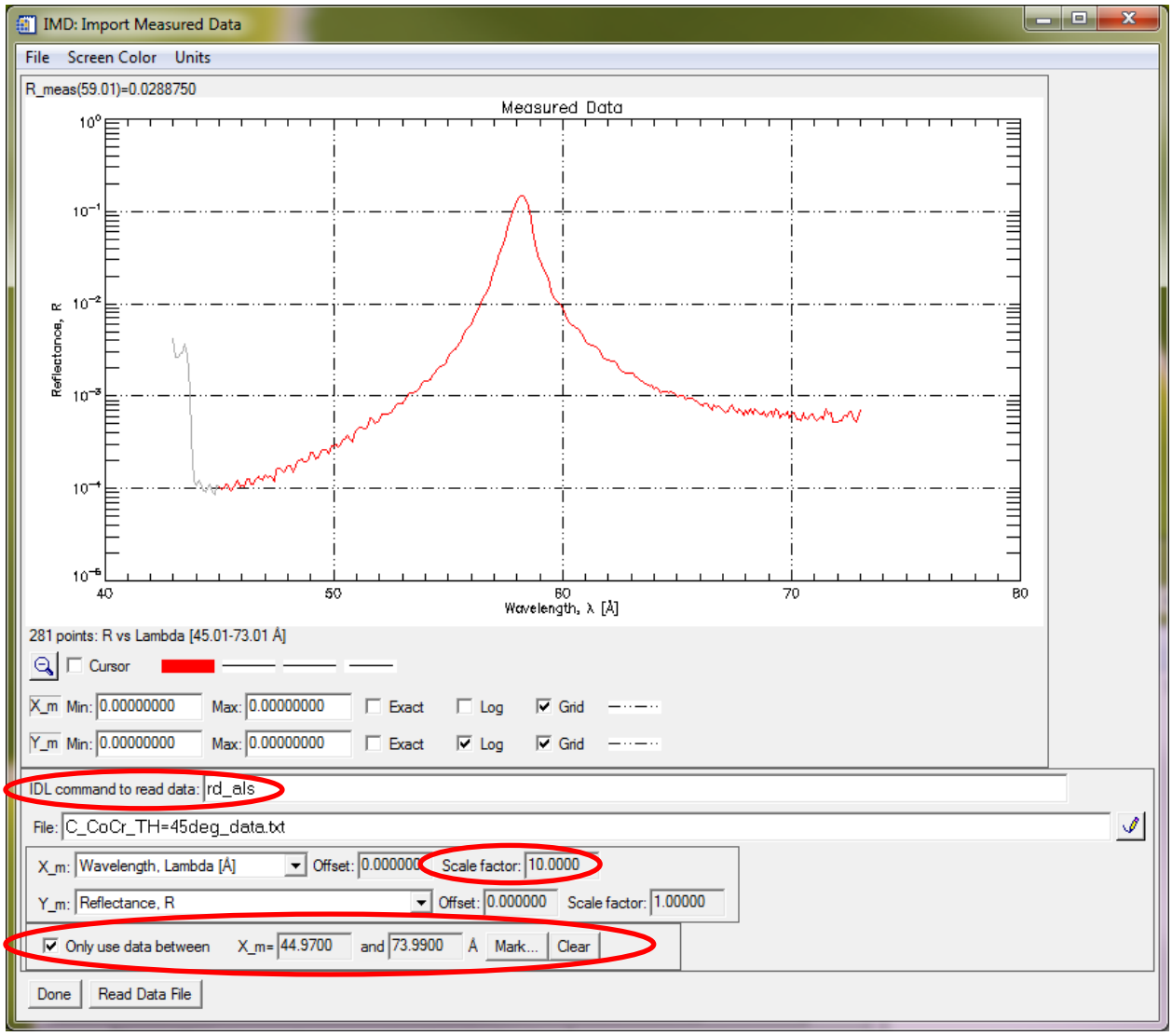
8.2.1 Example: Modeling XRR Measurements of a Mo/Y Multilayer

Once you have imported measured data into IMD, it can be plotted along with any computations you may perform, by checking the box for "Measured Data" in the Dependent Variables tab of IMDXPLOT. For example, here's the Mo/Y XRR data from the previous section displayed in an IMDXPLOT window along with calculated Reflectance vs. Grazing Incidence Angle data. For this calculation (which can be found in the file "`imd/examples/Measured Data/Mo_Y_L=1.54A.imd`") our structure comprises 20 repetitions of a Mo/Y bilayer: we've adjusted the multilayer period d, layer thickness ratio Γ, and interface widths σ by trial and error until we get a reasonably good (but certainly not perfect) "fit" to the measured data. We've also entered a value for instrumental angular resolution (§5.2.1) of $\delta\theta$=0.010°, corresponding to the resolution of the X-ray reflectometer used to make the measurements:



8.2.2 Example: Modeling Soft X-ray Reflectance Measurements of a C/CoCr Multilayer

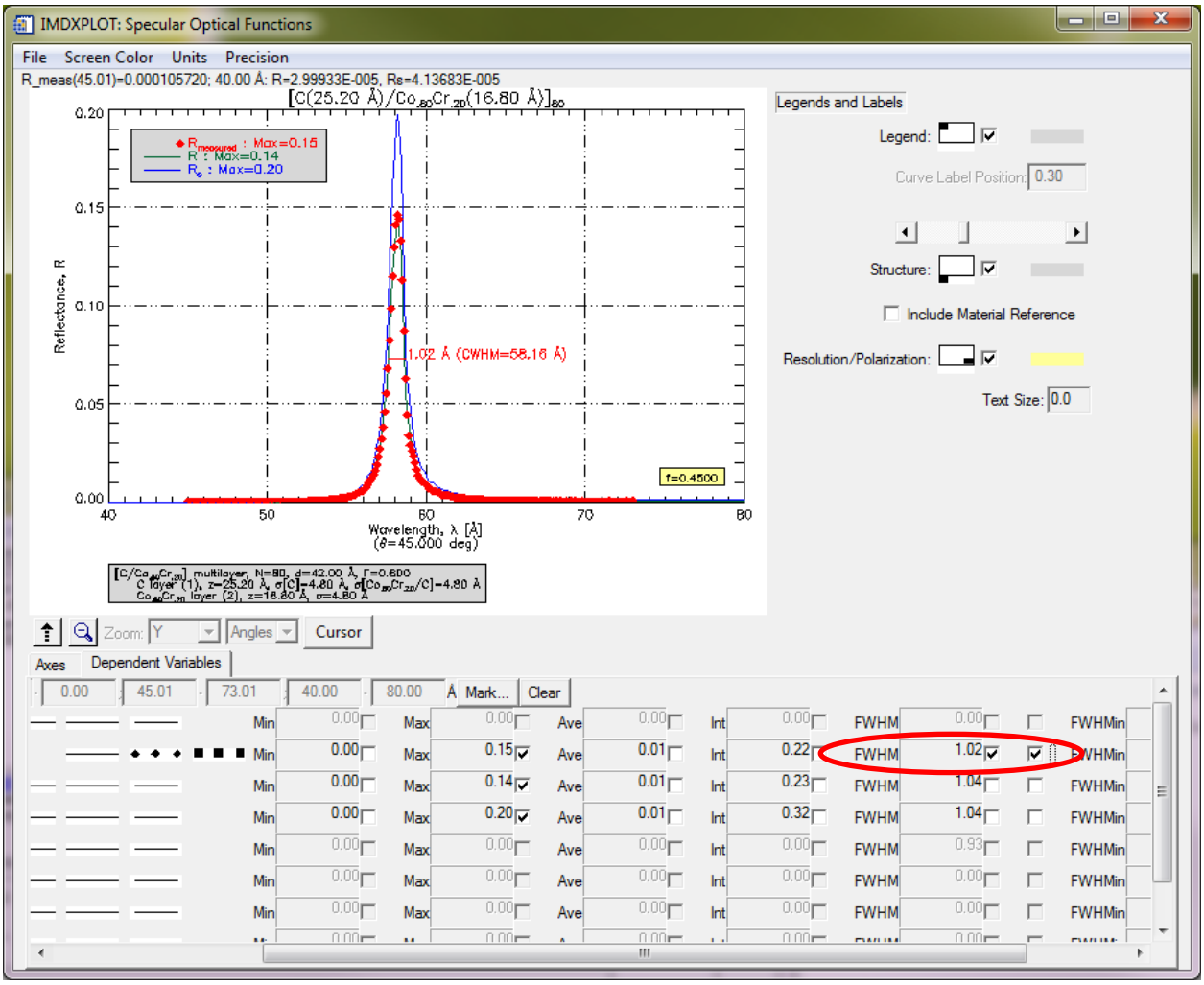
The file "`C_CoCr_TH=45deg_data.imd`" in the `imd/examples/Measured Data` directory contains data imported from the file "`C_CoCr_TH=45deg_data.txt`" in the same directory. That data file contains Reflectance-vs-Wavelength data for a periodic $C/Co_{.80}Cr_{.20}$ multilayer, measured in the soft X-ray range at 45° incidence at the ALS synchrotron, Beamline 6.3.2 (courtesy of E. Gullikson). As the data file in this case is a 'standard' file from this ALS beamline, we can use the "rd_als" (§11.4.3) routine to read the file:



As can be seen in the figure above, we've checked the box labeled "Only use data between" in the Variables area of the Import Measured Data window, in order to ignore the bad data below λ~44Å (i.e., below the C K-edge, where the incident beam intensity was highly attenuated due to the C filter being used for the measurement, resulting in inaccurate data.); the data outside the range specified is displayed in grey in the plot. We've also entered a scale factor of 10.0 for the X_m values, as this particular ALS data file contains wavelength values specified in units of nm, while **X_m always must be defined in IMD in units of Å when X_m corresponds to wavelengths, energies, or any length parameters, and in units of degrees from normal when X_m corresponds to Incidence or Scattering Angles.**

The C/$Co_{.80}Cr_{.20}$ multilayer data was modeled using a structure having N=80 repetitions of a C/$Co_{.80}Cr_{.20}$ bilayer with d=42 Å, Γ=0.6, and with interface widths σ=4.8 Å (`C_CoCr_TH=45deg.imd`). As the ALS beam is partially s-polarized, we've entered a value for the incident beam polarization (§5.2.2) of f=0.45

(based on previous measurements of the beam polarization) to correctly simulate the 'average' reflectance actually measured at 45° incidence. In the IMDXPLOT window shown in the next figure, we've displayed the measured data as red diamonds, the average reflectance (e.g., computed for f=0.45) as a green line, and the reflectance for pure s-polarization as a blue line. We've also displayed in the legend "Max" values for all three quantities, and have checked both boxes next to "FWHM" for the measured data, in order to display the FWHM (first checked box) directly on the plot (second checked box) rather than in the legend:



8.3 Optimization Target Profiles: X_o, Y_o and W_o

Film-stack optimization in IMD refers to a numerical method to identify a set of layer thicknesses in a film stack that yield a specific optical function profile. For example, you can use IMD's optimization capabilities to design an aperiodic multilayer having a specific Reflectance-vs-Wavelength or Reflectance-vs-Angle profile. Optimization is analogous to parameter estimation via curve-fitting: curve-fitting in IMD allows you to determine the structural parameters in a film stack (e.g., layer thicknesses,

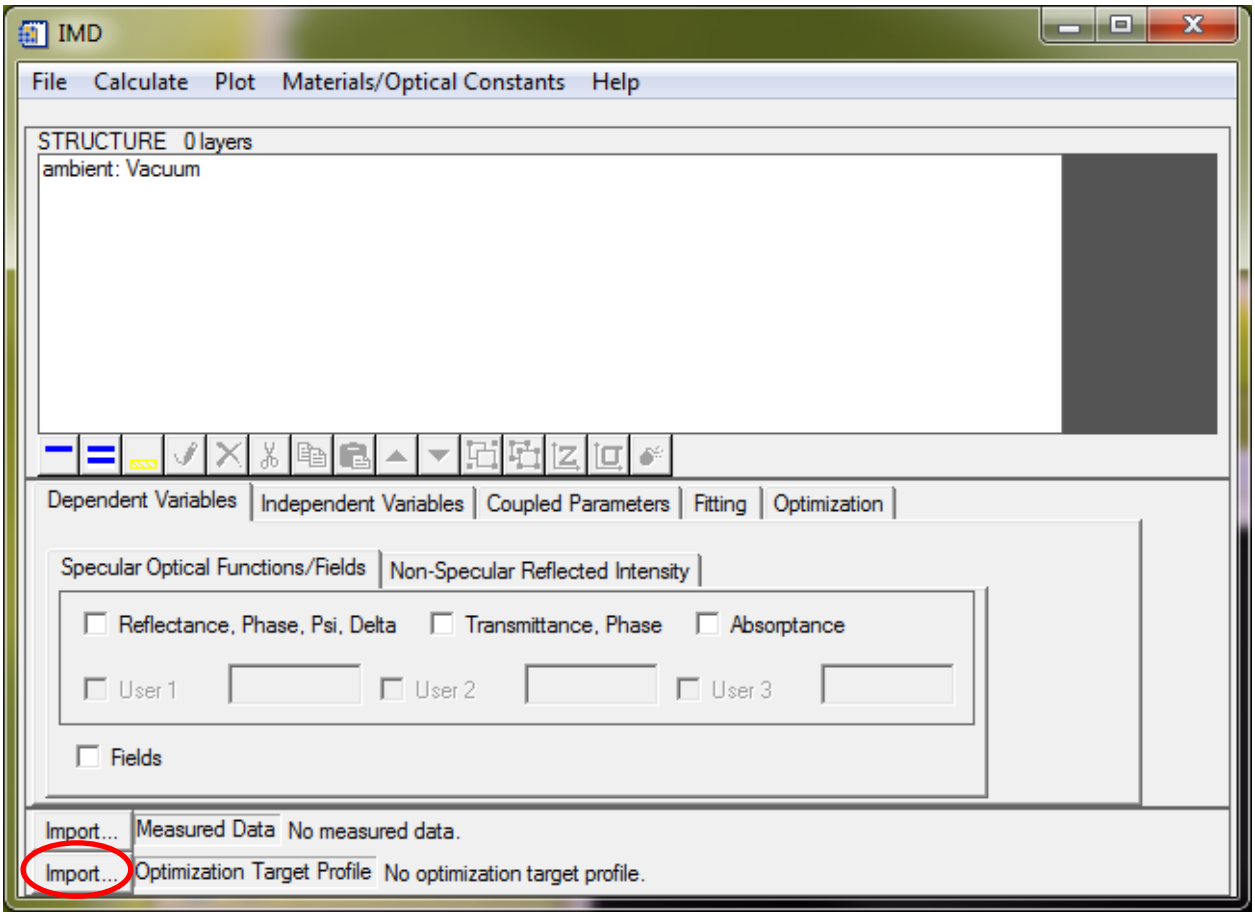
interface widths, etc.) that yield a calculated optical function matching measurements of that same optical function. Curve-fitting can be used with, e.g., XRR data, Reflectance-vs-Wavelength data, and so forth. IMD will adjust the structural parameters that you designate in order to minimize a function that represents a measure of the difference between the computed and measured optical functions – the function being minimized is either the χ^2 function (or more accurately, the S statistic, as explained below,) or a Figure-of-Merit (FOM) function that you define (also explained below), depending on the fitting algorithm selected.

Optimization, like curve-fitting, also adjusts structural parameters (i.e., layer thicknesses) in order to minimize the χ^2 or FOM function. But, while curve-fitting uses Measured Data (X_m, Y_m, SIGY_m) to compute the χ^2 or the FOM function, film-stack optimization uses so-called Optimization Target Profile data – X_o, Y_o, and W_o. The Optimization Target Profile is simply the desired optical function profile that is meant to be achieved: X_o and Y_o are the independent and dependent variable arrays, and W_o is the "weight" array – each element of the dependent variable array Y_o[i] is weighted by W_o[i] when computing the χ^2 statistic or FOM, as we'll explain below. The default W_o value is 1.0, but you can specify W_o as a function of X_o if you want to weight the optimization target profile in some specific way. As in the case of Measured Data, X_o values must be defined internally in units of either Å or degrees from normal.

Defining X_o, Y_o and W_o is typically done using IMD's built-in file-import routines (§8.3.1), or by constructing the target profile using straight line segments directly in IMD, as we'll see in §8.3.2. But if you're running IMD as an IDL application, you can also use your own IDL programs. See §11.4 for more details about importing optimization target profile data using your own IDL programs.

8.3.1 Importing Optimization Target Profile Files

To import an optimization target profile data file using IMD's built-in file-import capabilities, select from the IMD menu bar "File→Import/Create Optimization Target Profile…", or click the button labeled "Import…" at the very bottom of the IMD window, to the left of the "Optimization Target Profile" label:



The IMD Import/Create Optimization Target Profile window will appear, as in Figure 35 below. This window is very similar to the Import Measured Data window shown in Figure 34, and also comprises a Graphics area, where the Optimization Target Profile will be displayed. But the File area is different: you can check either the "Import Target Profile from File" or the "Create Target Profile from Straight Lines" radio buttons. If the "Import Target Profile from File" radio button is selected, then you'll see a Variables area, similar to the analogous area in the Import Measured Data window, where you specify which Independent and Dependent variables are being imported, and how the imported data should be scaled. (We'll get to the "Create Target Profile from Straight Lines" option in the next section.)

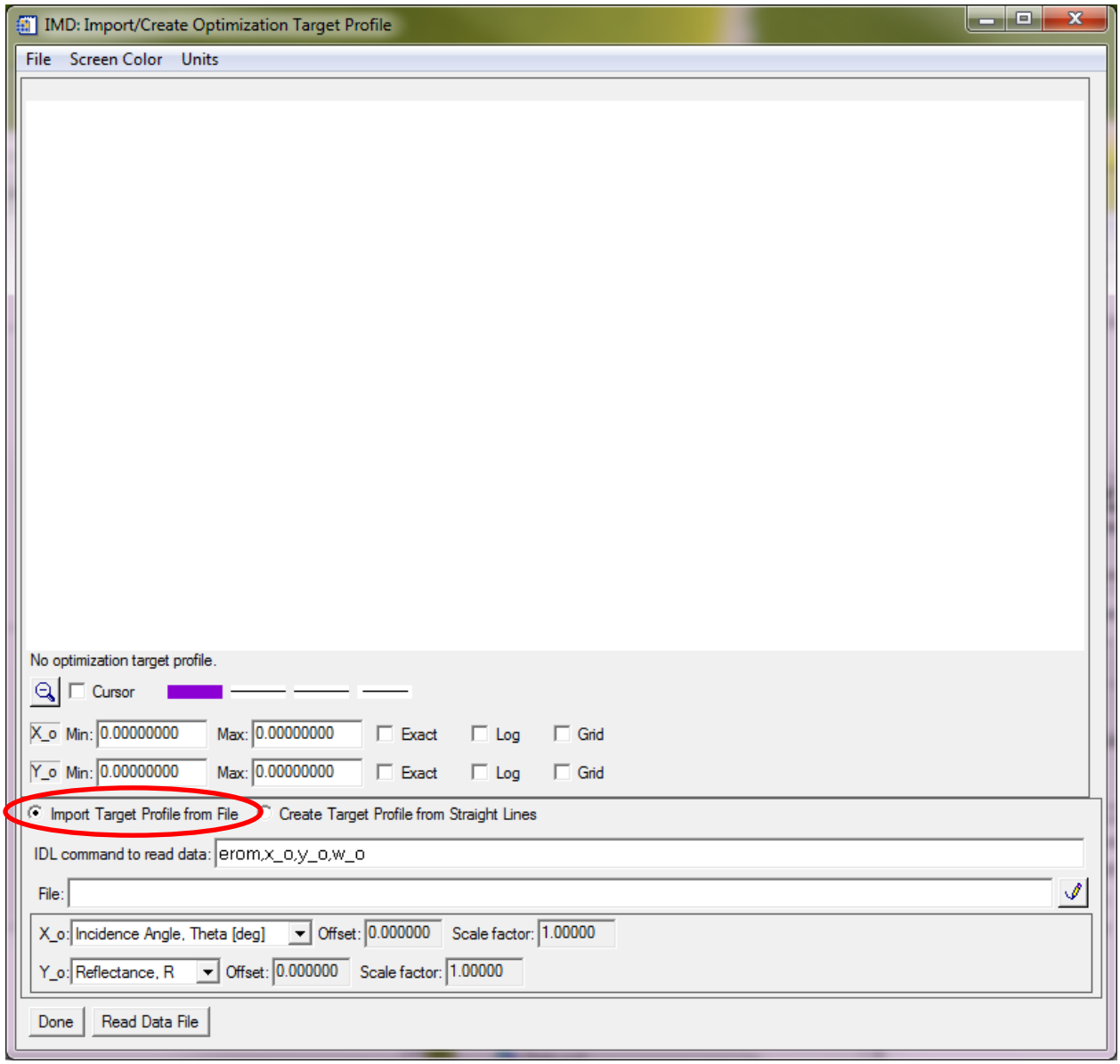
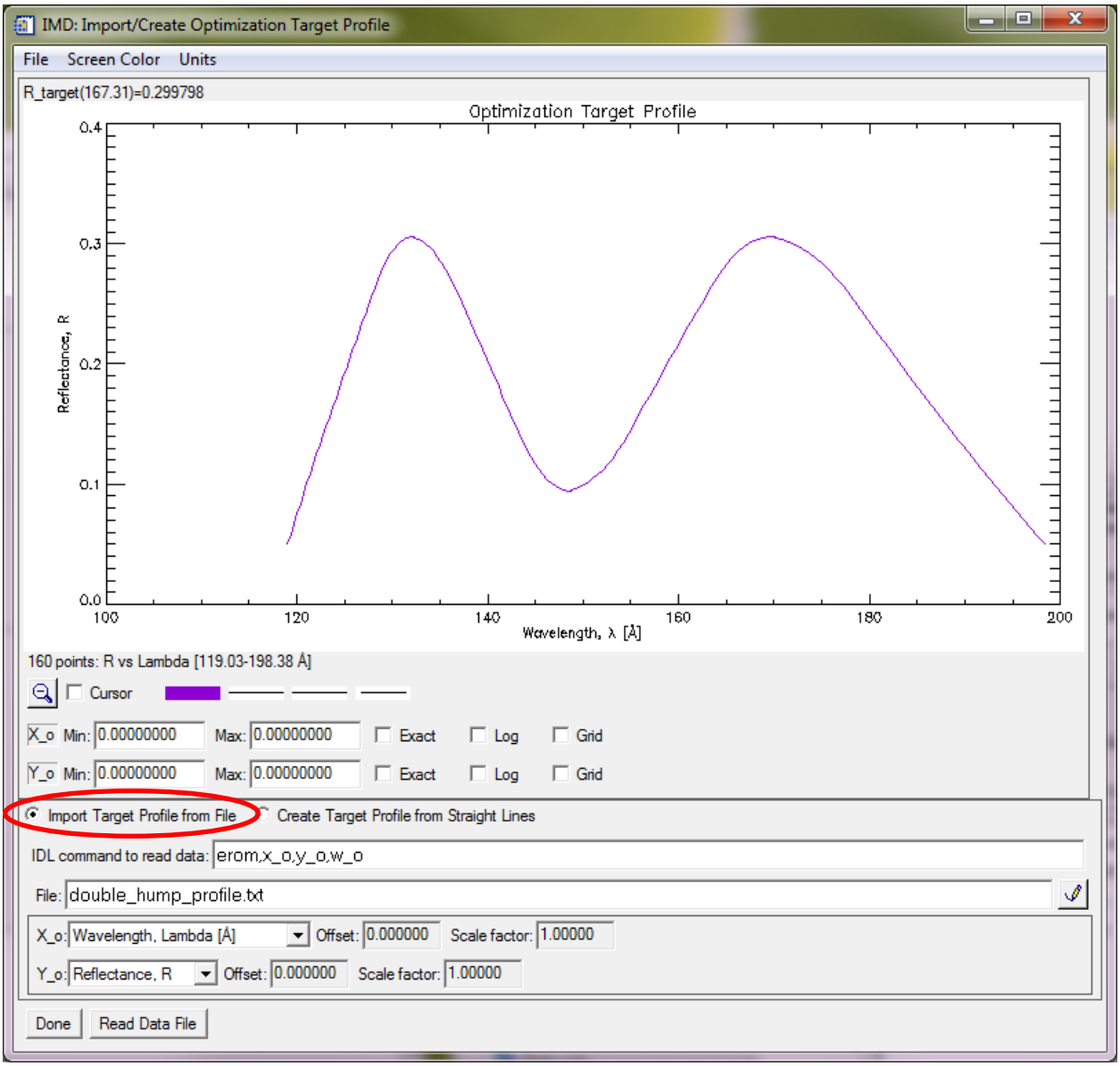


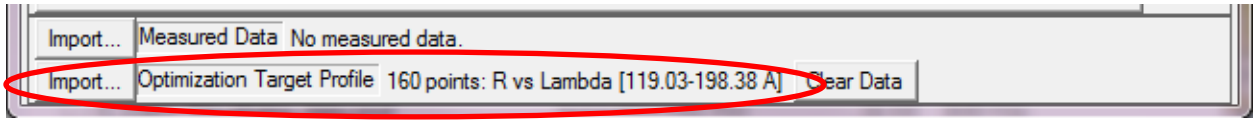
Figure 35. The Import/Create Optimization Target Profile window.

To illustrate, we'll import Optimization Target Profile data from a file. You can open the relevant `.imd` file in the `imd/examples/Optimization` directory to follow along, however, please note again that this `.imd` file references a data file located in that directory, and the reference is specified without the full path names to the file – full path names are operating-system dependent, and also depend on where you've installed IMD on your computer, so there's no way to provide an example file with a full path name that will work with any IMD installation. Therefore, to use this `.imd` file, **be sure to set `imd/examples/Optimization` as your working directory in IMD**, using "File→Change Working Directory…" from the IMD or IMD Launcher menu bars, or using the 'cd' IDL command at the IMD> prompt.

For this example, we'll import the file "double_hump_profile.txt" using the erom command (syntax: erom,x_o,y_o,w_o). Here's the Import/Create Optimization Target Profile window after import, where we've selected "Wavelength, Lambda [Å]" for X_o, and "Reflectance, R" for Y_o:



Once Optimization Target Profile data has been imported (or created), the number of data points, and the type of data imported are displayed at the bottom of the main IMD window:

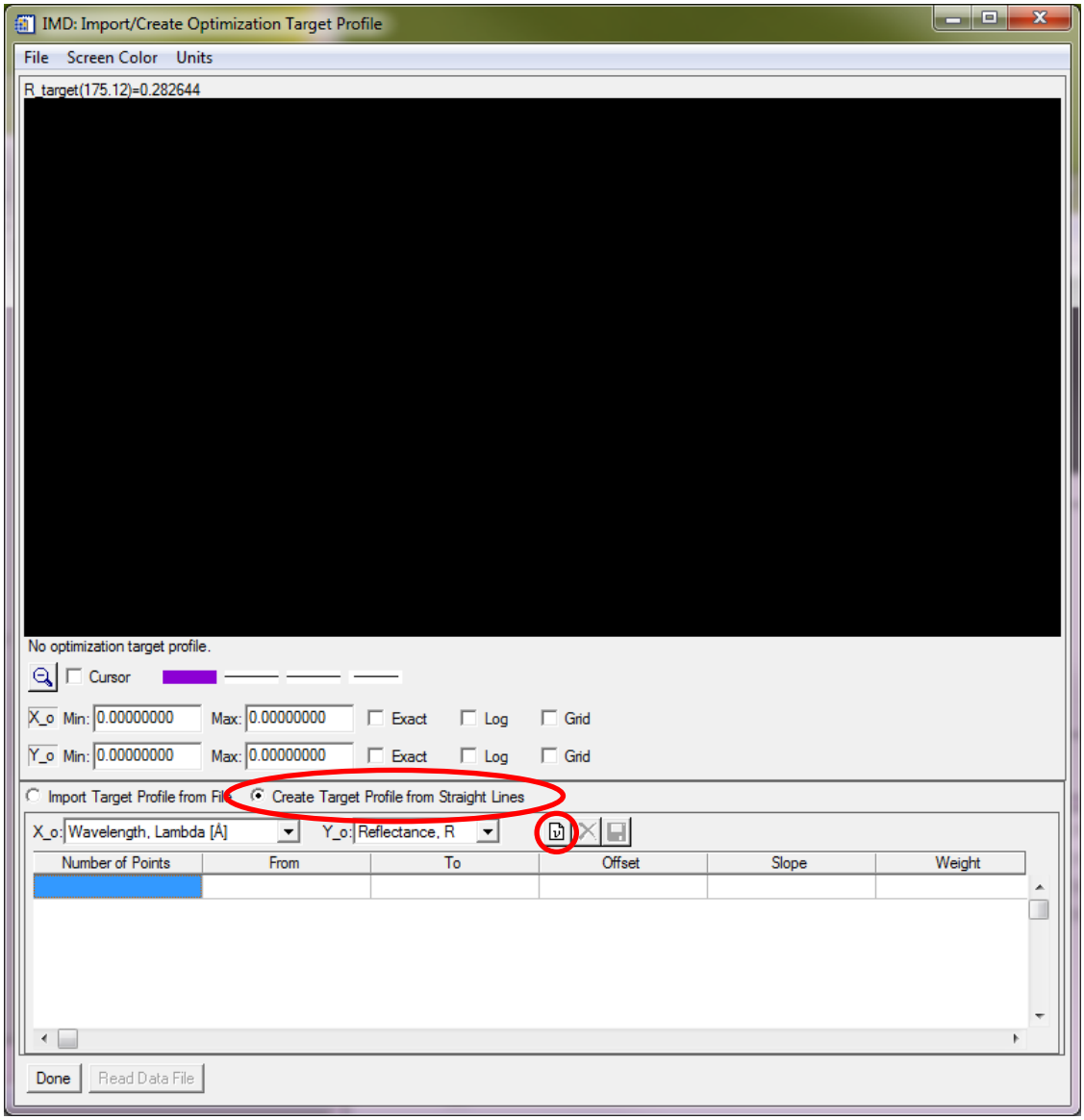


Use the "Clear Data" button to delete Optimization Target Profile data as necessary.

8.3.2 Building Optimization Target Profiles from Scratch

For some film-stack optimizations it may be sufficient to define an Optimization Target Profile from one or more straight-line segments. For example, aperiodic multilayers are often designed to achieve flat response as a function of angle or wavelength; in that case the Optimization Target Profile is nothing more than a constant value over some desired range of angles or wavelengths.

You can construct an Optimization Target Profile from straight line segments directly in IMD's Import/Create Optimization Target Profile window. Click the "Create Target Profile from Straight Lines" radio button, and the window will display a table and some additional controls, as shown in the next figure:



Use the "Add Segment" button ( ) to add new straight line segments one at a time: for each (X_o,Y_o,W_o) segment, enter in the table the Number of Points for that segment, the range for X_o (From and To), the offset and slope, where Y_o=Offset+Slope*X_o over the range of X_o values defined for that segment, and the weights W_o for that segment (default=1.0). The next figure shows the Import/Create Optimization Target Profile window after creating a profile of Reflectance vs. Energy having two triangular peaks:

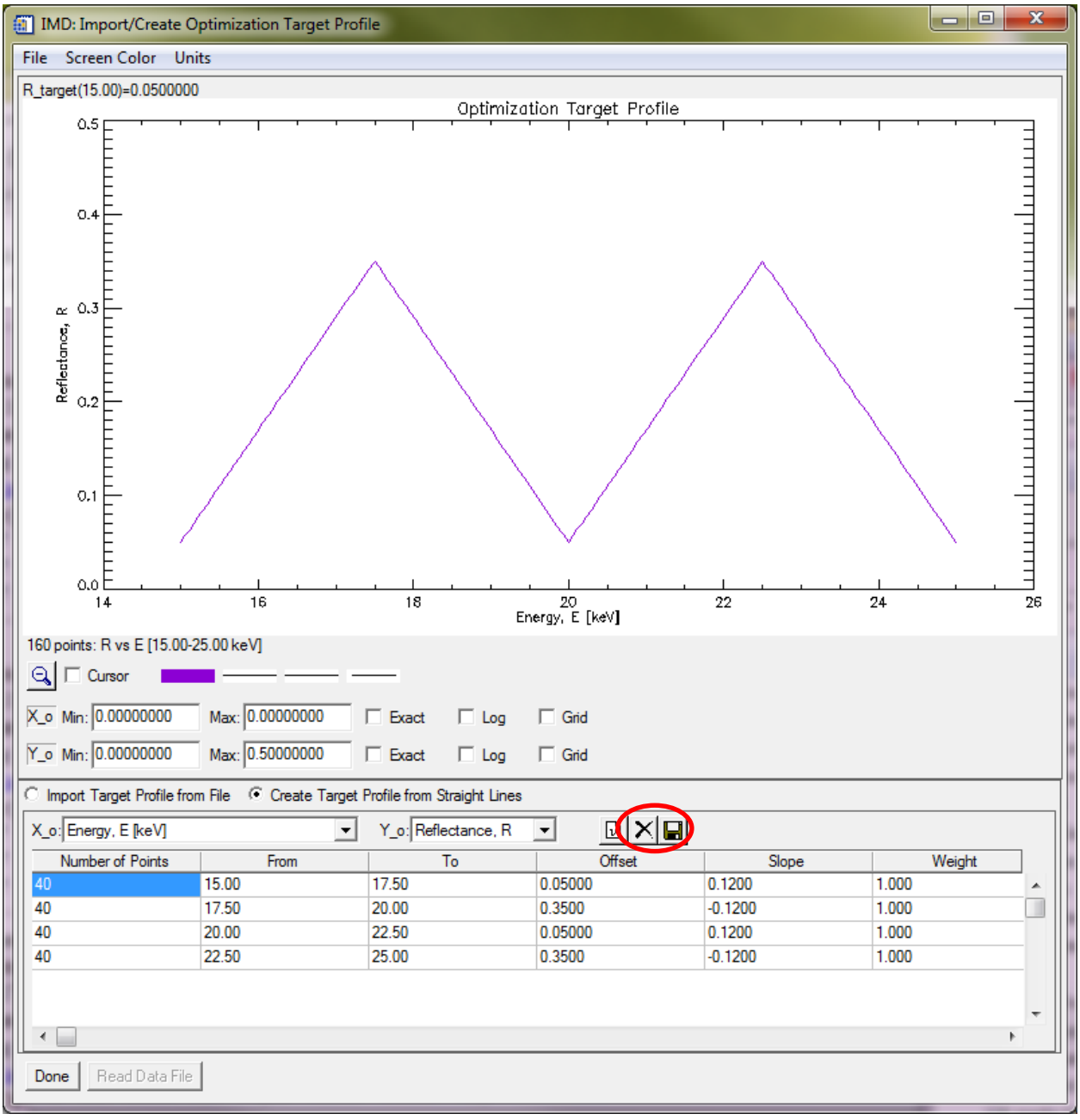


Figure 36. An Optimization Target Profile constructed from straight-line segments.

Segments can be deleted using the Delete Segment button (), and the Optimization Target Profile can be exported to a plain text file using the Export Target Profile button ().

8.4 IMDXPLOT: Measured Data and Optimization Target Profile

Both Measured Data and Optimization Target Profile data can be displayed in plots you create in IMDXPLOT. If Measured Data and/or Optimization Target Profiles are defined in IMDXPLOT, use the controls on the Dependent Variables Tab to control if and how these data are displayed.

8.4.1 IMDXPLOT: ROI's for Measured Data and Optimization Target Profile

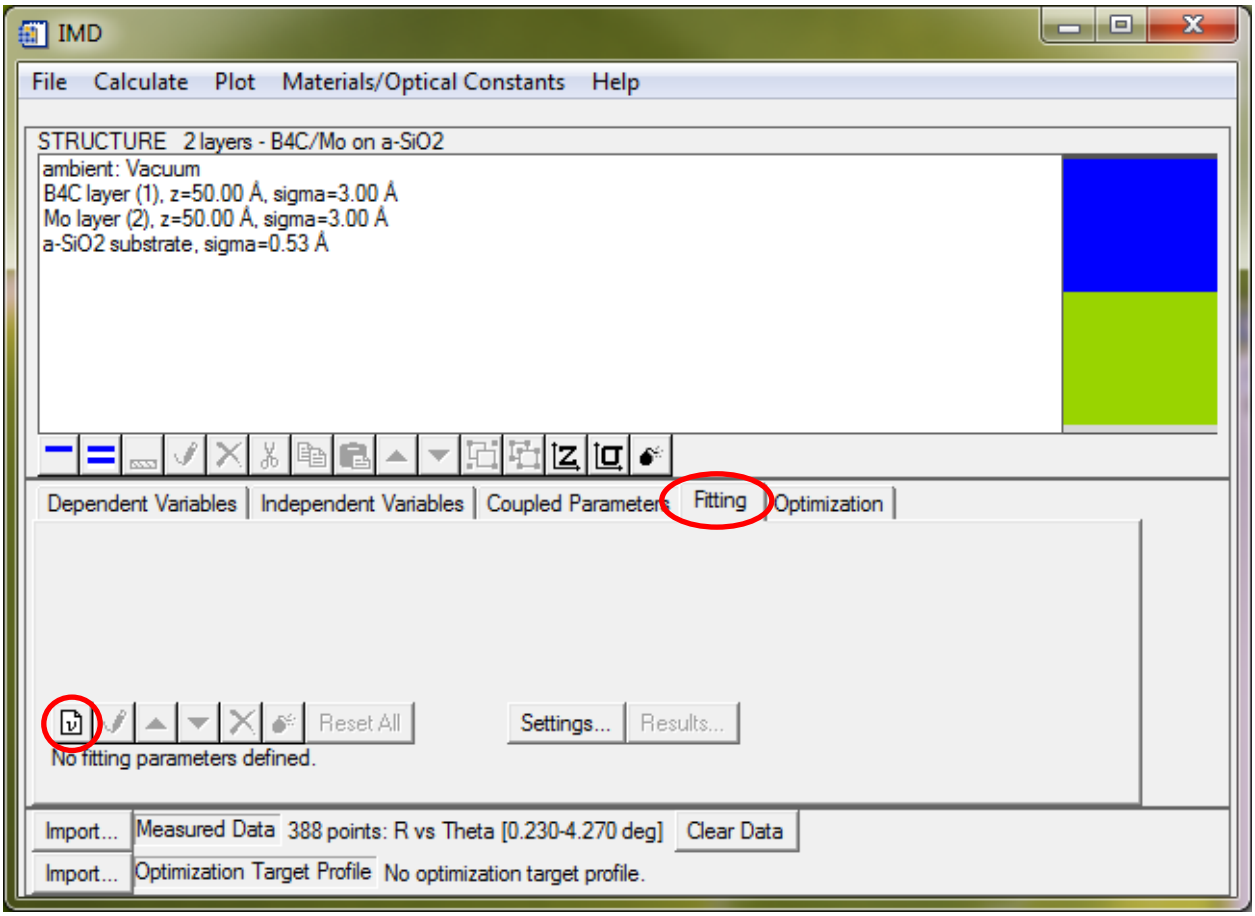
We mentioned in §6.6.1.2 that the ROI settings on the Dependent Variables tab in an IMDXPLOT window include three pairs of 'start' and 'end' values: the first pair is used only for Measured Data, the second pair only for Optimization Target Profile data, and the third pair for computed optical functions. Separate ROI start and end values are needed for Measured Data and Optimization Target Profile Data because their respective independent variables – X_m and X_o – comprise values that may be different, in general, than those of the independent variables used for the computation. You can therefore define whatever ROI start and end values you want – the ROI for Measured Data and/or Optimization Target Profile can overlap with the ROI for computed optical functions, or they can be completely different.

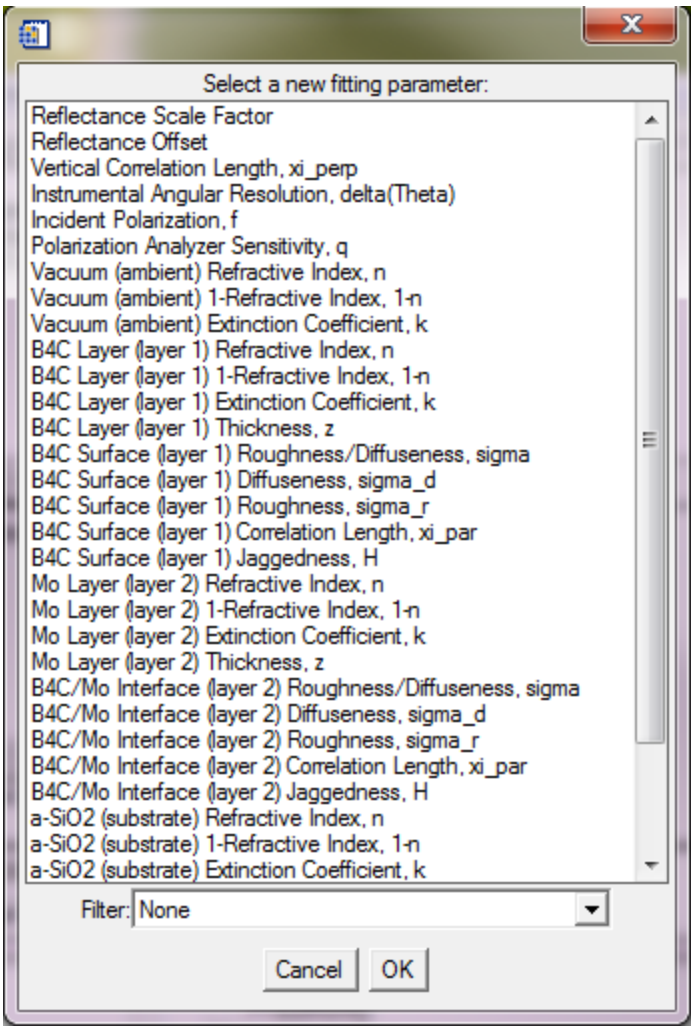
8.5 Specifying Fitting and Optimization Parameters

In order to perform curve-fitting or optimization in IMD, you must specify which parameters should be adjusted during the fit or optimization. We'll explain how to do that in this section: for the case of curve-fitting to measured data, we'll use the example file '`imd/examples/Fitting/B4C_Mo_L=1.54A.imd`'. That file already contains a defined structure – a B_4C/Mo bilayer – and imported XRR data. For optimization, we'll use the example file '`imd/examples/Optimization/Ni_C_AML_0.imd`', which contains optimization target profile data and an aperiodic Ni/C multilayer defined as the structure.

8.5.1 Specifying Fitting Parameters

After opening the file '`imd/examples/Fitting/B4C_Mo_L=1.54A.imd`', we'll add a new fitting parameter by navigating to the "Fitting" tab on the main IMD window, where we click the "Add Fit Parameter" button ():

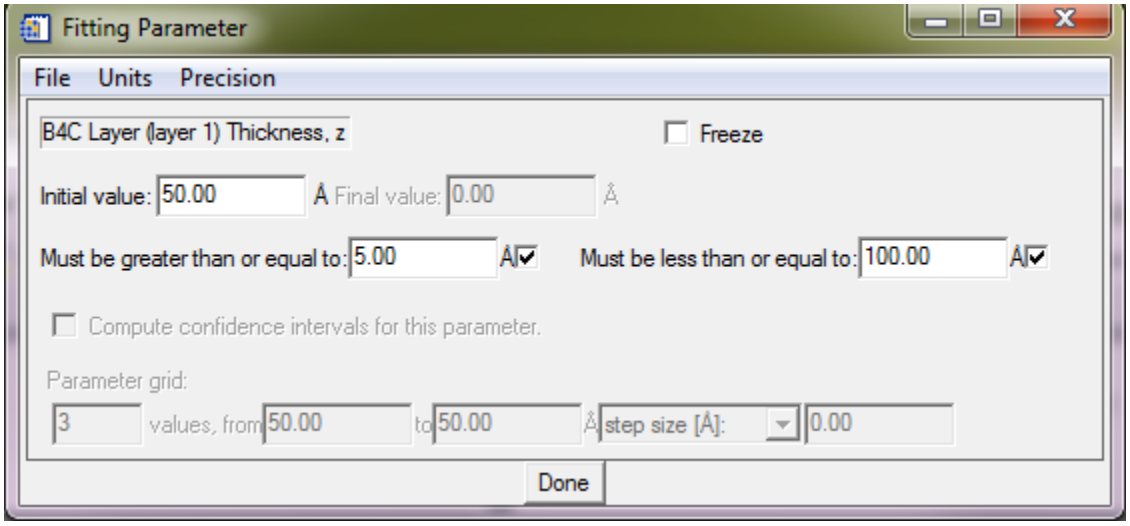

IMD
File Calculate Plot Materials/Optical Constants Help
STRUCTURE 2 layers - B4C/Mo on a-SiO2
ambient: Vacuum
B4C layer (1), z=50.00 Å, sigma=3.00 Å
Mo layer (2), z=50.00 Å, sigma=3.00 Å
a-SiO2 substrate, sigma=0.53 Å
Dependent Variables
Independent Variables
Coupled Parameters
Fitting
Optimization
Reset All
Settings...
Results...
No fitting parameters defined.
Import...
Measured Data
388 points: R vs Theta [0.230-4.270 deg]
Clear Data
Import...
Optimization Target Profile
No optimization target profile.



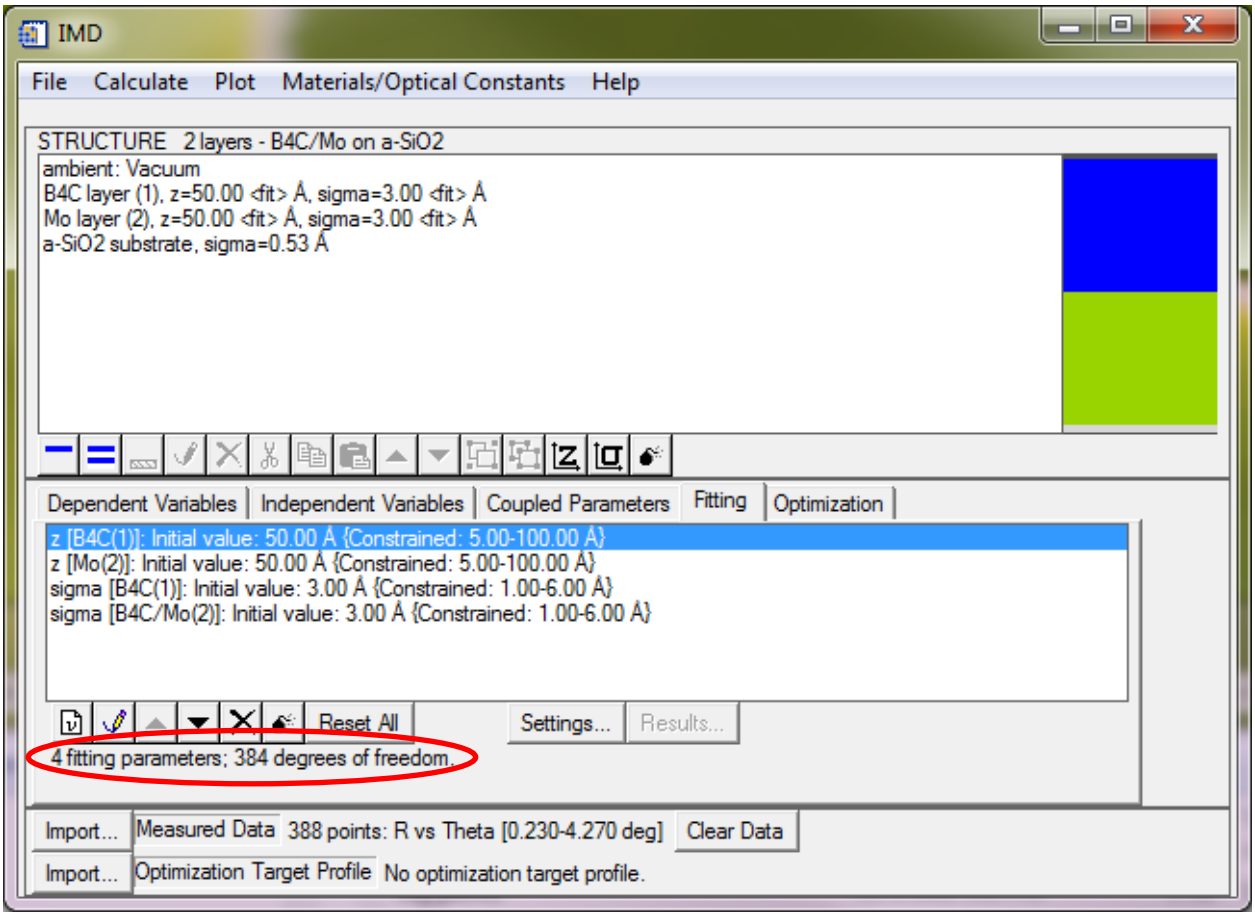
Select the parameters you want to adjust during the fit from the list on the window that appears, shown in the figure above. In addition to all the structural parameters and polarization parameters, two additional parameters are available for selection as fitting parameters: Reflectance Offset and Reflectance Scale Factor. These two parameters can be used to account for unknown systematic errors in your measured data. (If the measured data corresponds to some other dependent variable besides Reflectance, e.g., Transmittance, then Offset and Scale factors for that dependent variable would be listed.) Just as in the case of selection of Coupled Parameters and Independent Variables, the list of parameters available for selection as Fitting Parameters can be filtered using the drop-list at the bottom of the window.

> **Note**: Parameters that are already defined as Coupled Parameters, Independent Variables, or Optimization Parameters cannot be selected as Fitting parameters. However the "source" parameter of a Coupled Parameter can be selected as a fitting parameter.

For this example, where we'll want to determine the B_4C and Mo layer thicknesses and interface widths from fitting the XRR measurement (§8.7.1), we'll select four fitting parameters: B_4C Layer thickness, Mo Layer thickness, B_4C Surface Roughness/Diffuseness, and B_4C/Mo Interface Roughness/Diffuseness. After selecting the first fitting parameter – B_4C Layer thickness – a Fitting Parameter window appears:



On the Fitting Parameters window you can enter an initial value, which is the starting value used in the fit, and optionally specify upper and/or lower bounds, or constraints, on the fitting parameter. (You can check the box labeled "Freeze" if you want to temporarily disable the selected parameter from being adjusted during the fit, without permanently removing that parameter from the list.) In the figure above we've specified an initial value of 50 Å for the B_4C layer thickness, and have constrained this parameter to the range 5 Å < z < 100 Å. We similarly specify the other three fitting parameters, also with constraints. When we're finished defining our four fitting parameters, the IMD main window looks like this:

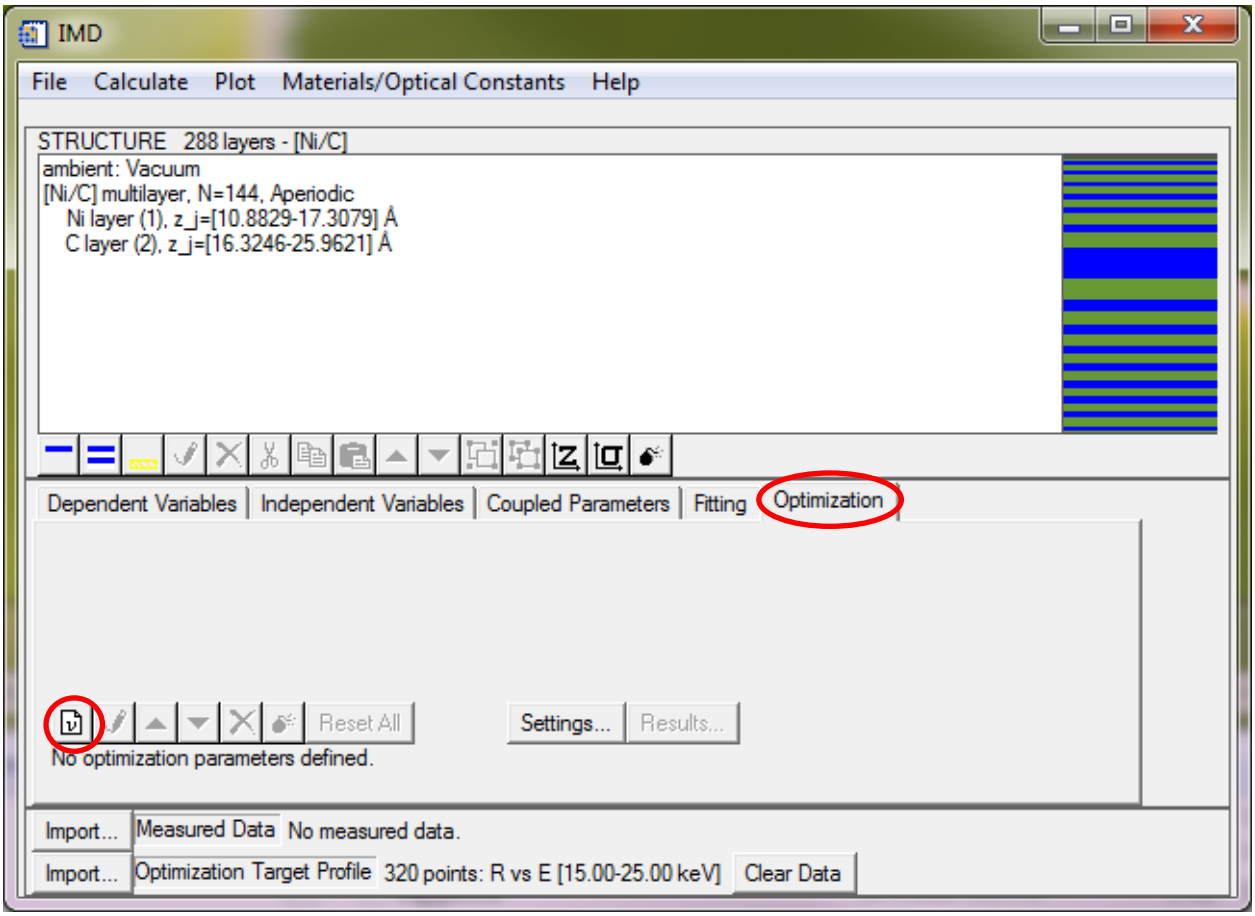


Note that the status area at the bottom of the Fitting tab indicates that we've selected 4 fitting parameters, resulting in 384 degrees of freedom (= 388 data points – 4 fitting parameters.)

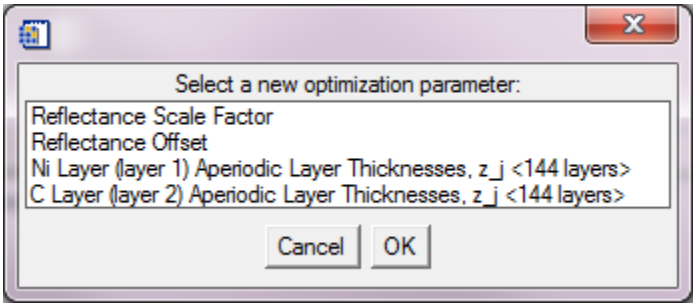
8.5.2 Specifying Optimization Parameters

The structure defined in the example file '`imd/examples/Optimization/Ni_C_AML_0.imd`' is a Ni/C aperiodic multilayer (AML) containing N=144 repetitions, designed for grazing incidence operation (θ=10 mrad) in the X-ray band from E=12 to 30 keV. In §8.7.6 we're going to optimize the Ni and C layer thicknesses in this AML, in order to achieve a triangular reflectance profile like the one shown in Figure 36.

After opening the file '`imd/examples/Optimization/Ni_C_AML_0.imd`', we'll add two new optimization parameters by navigating to the "Optimization" tab on the main IMD window, where we click the "Add Optimization Parameter" button ( ):



The optimization parameter selection window that appears (next figure) shows the list of available optimization parameters. As in the case of fitting parameters described in the previous section, Offset and Scale Factors for the dependent variable can be selected as optimization parameters. But in the case of optimization, the only structural parameters available for selection are layer thicknesses. However, since the structure is defined as an AML, the entire set of Ni layer thicknesses, and the entire set of C layer thicknesses, can each be selected as single parameters – in that case all of the 144 layer thicknesses associated with each selected component layer would be adjusted during the optimization, for a total of 288 adjustable parameters:



For this example we select both the Ni and C layer thicknesses as optimization parameters. As each of these two parameters represent the entire list of 144 Ni thicknesses and the entire list of 144 C thicknesses, respectively, the Initial Value area is no longer a simple box for text entry, but is instead a 'combo-box' where, if you so choose, you can manually enter initial values for each of the 144 aperiodic layers:

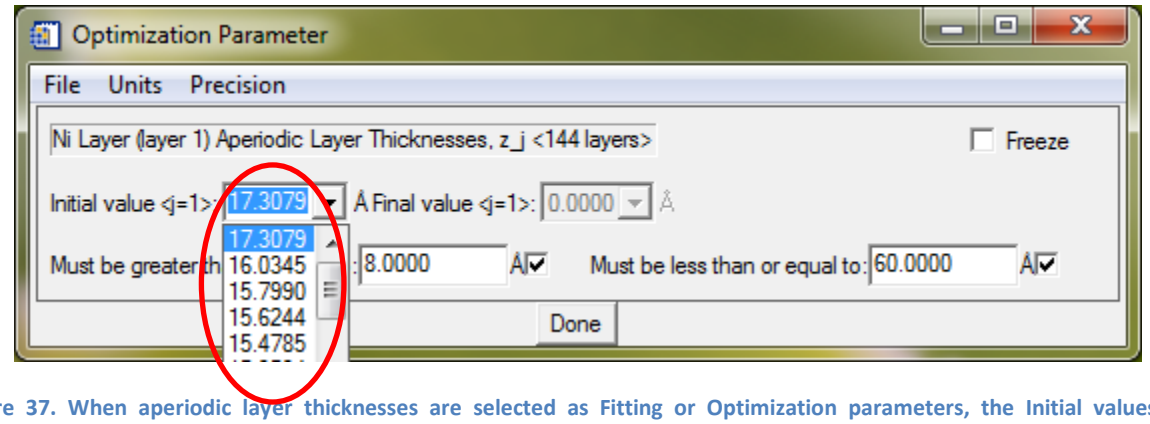
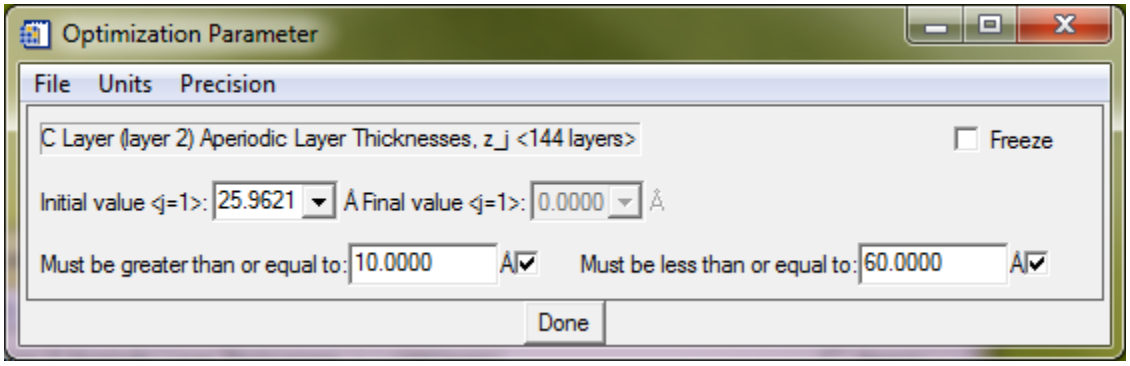
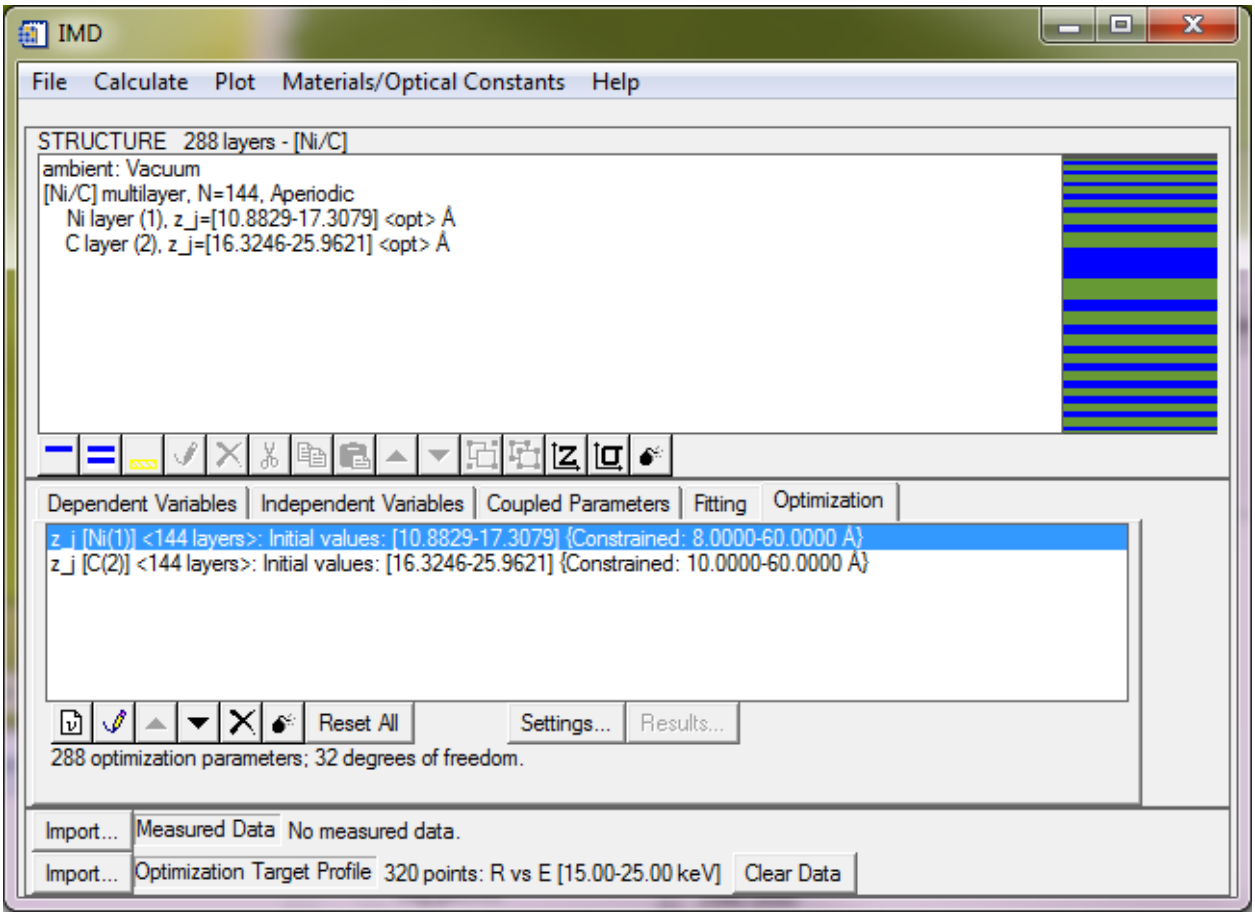


Figure 37. When aperiodic layer thicknesses are selected as Fitting or Optimization parameters, the Initial values are specified using a combo-box, as shown here.

For this example we won't manually enter initial values; we'll just use the default Initial values, which are simply the layer thicknesses that were already defined in the structure (we'll explain in §8.7.6 how we initially defined the aperiodic thicknesses in this example file.) We've also constrained all the Ni thicknesses to the range 8 Å < z < 60 Å. Similarly, the C layer thicknesses are constrained to 10 Å < z < 60 Å:



The main IMD window, after selecting the Ni and C layer thicknesses as optimization parameters, looks like this:



Note: The "Reset Values" menu option from the Fitting Parameter or Optimization Parameter window menu bar causes the Initial value(s) of the corresponding fitting or optimization parameter to be replaced with the currently defined value(s) for that parameter. The "Reset All" button on the Fitting or Optimization tab will reset the initial values of all parameters simultaneously.

Tip: At the completion of a fit or optimization, use the "Reset All" button to set the Initial values of the fitting or optimization parameters to the best-fit values determined from the fit or optimization; that can be helpful when you want to perform a new fit or optimization using those best-fit values as the new starting point values.

8.6 Algorithms for Curve-Fitting and Optimization

The fitting/optimization problem can be stated in a general way as follows: determine values for some fixed number p of adjustable parameters so that the calculated optical function Y(X) most closely fits a

particular set of data Y_{mo}, defined as a function of the independent variable X_{mo}, where X_{mo} takes on i=1,…N_{mo} discrete values. In the case of curve-fitting to measured data in IMD, X_{mo}=X_m and Y_{mo}=Y_m±SIGY_m, while for optimization, X_{mo}=X_o, and Y_{mo}=Y_o.

We've just seen in the preceding sections how to define the X_{mo} and Y_{mo} values required for fitting or optimization, and in §8.5 how to select the p adjustable parameters. In this section, we'll explain the algorithms available in IMD to solve the fitting/optimization problem just stated.

> **Note**: A full description of non-linear curve-fitting concepts is beyond the scope of this document. We'll provide enough information to explain the parameters and settings you'll need to specify in IMD, but you are strongly encouraged to pursue more detail in the references cited here in order to get the most out of IMD's fitting and optimization capabilities.

You can choose from four algorithms in IMD for curve-fitting and optimization. The algorithm selection, and the parameters and settings associated with the selected algorithm, are specified for curve-fitting on the Fitting Settings window, which is opened by clicking the button labeled "Settings" on the Fitting tab of the main IMD window. The analogous parameters and settings are specified for optimization on the Optimization Settings window, which is opened by clicking the button labeled "Settings" on the Optimization tab of the main IMD window:

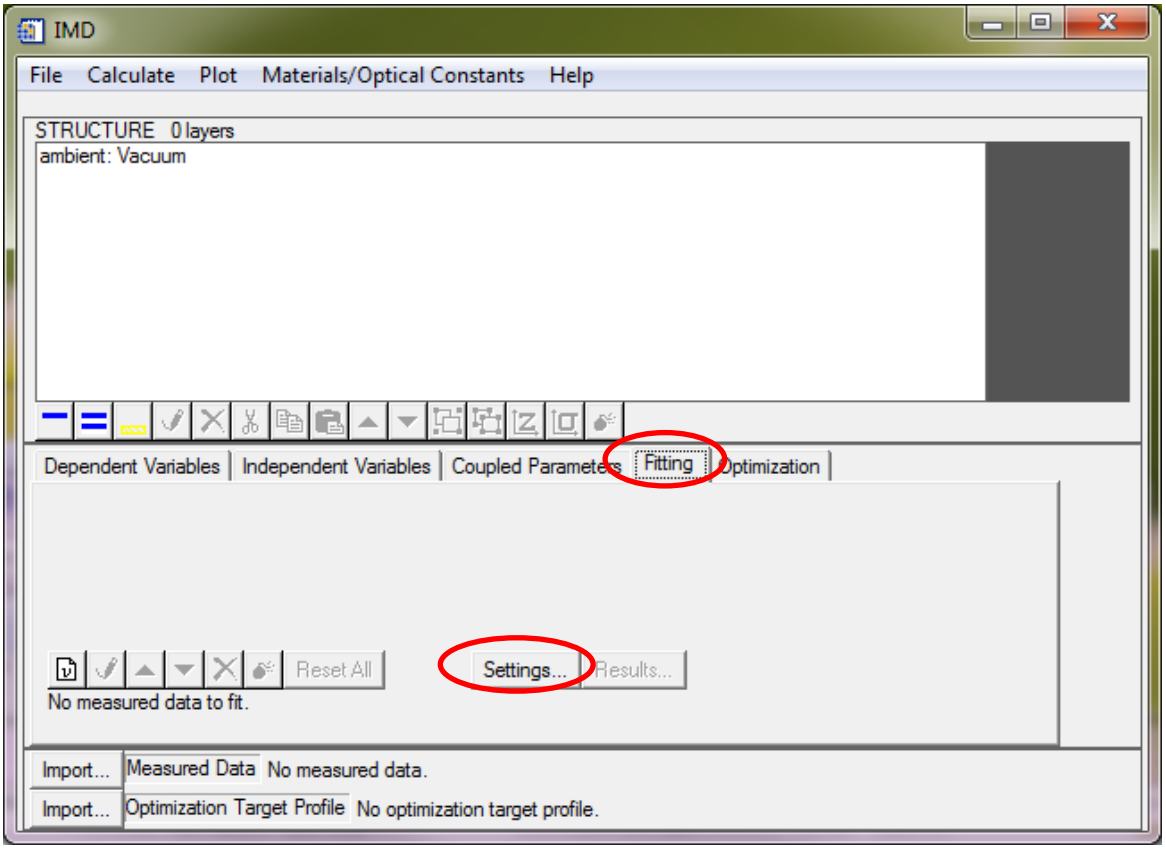


Figure 38. Click the "Settings" button on the "Fitting" tab of the main IMD window to open the Fitting Settings window.

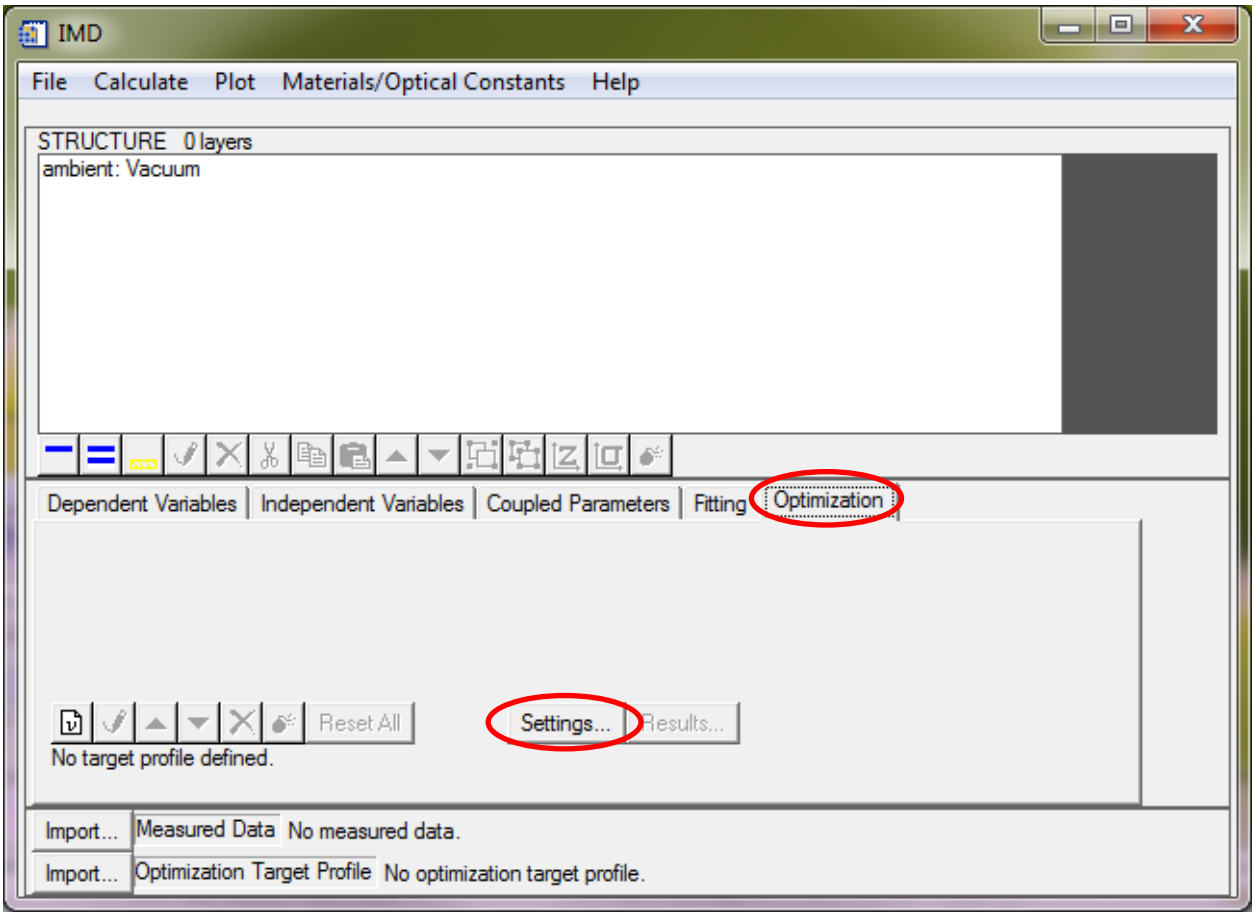


Figure 39. Click the "Settings" button on the "Optimization" tab of the main IMD window to open the Optimization Settings window.

We'll show how the Fitting Settings and Optimization Settings windows appear for each of the four algorithms available in IMD in the subsections that follow.

8.6.1 χ^2 Minimization: Marquardt Algorithm (CURVEFIT)

The first algorithm available in IMD for fitting and optimization is a gradient-expansion algorithm, developed by Marquardt,[19] for determination of non-linear parameters based on the χ^2 test of fit. The IMD implementation of this algorithm is adapted from the IDL CURVEFIT routine, which is in turn based on the methods described in reference [20].

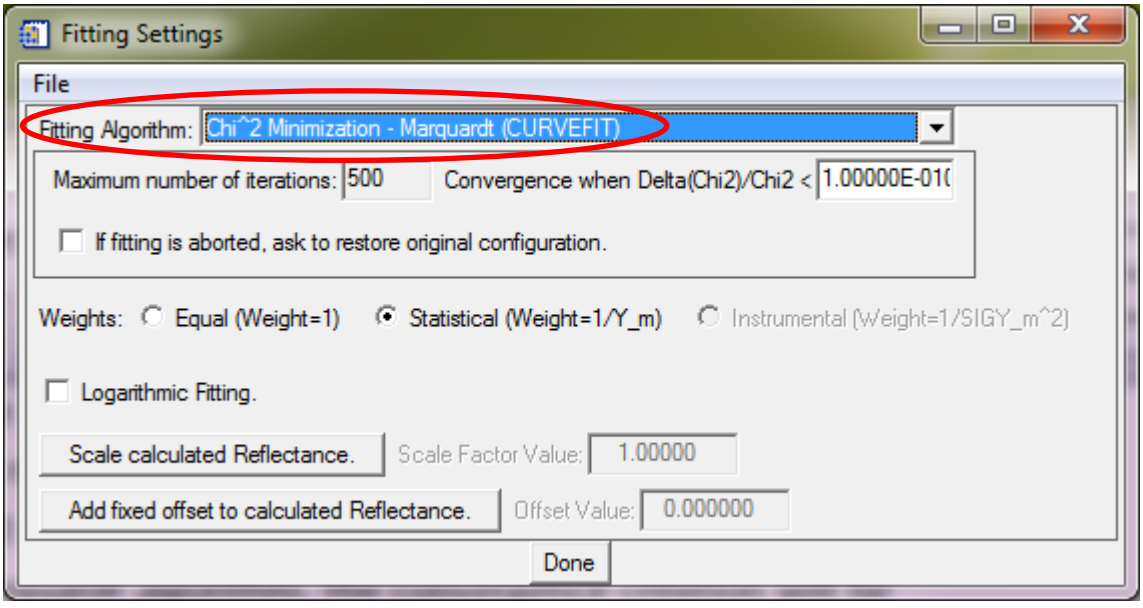
In the Marquardt algorithm, the value of the statistic S is computed after each iteration, using:

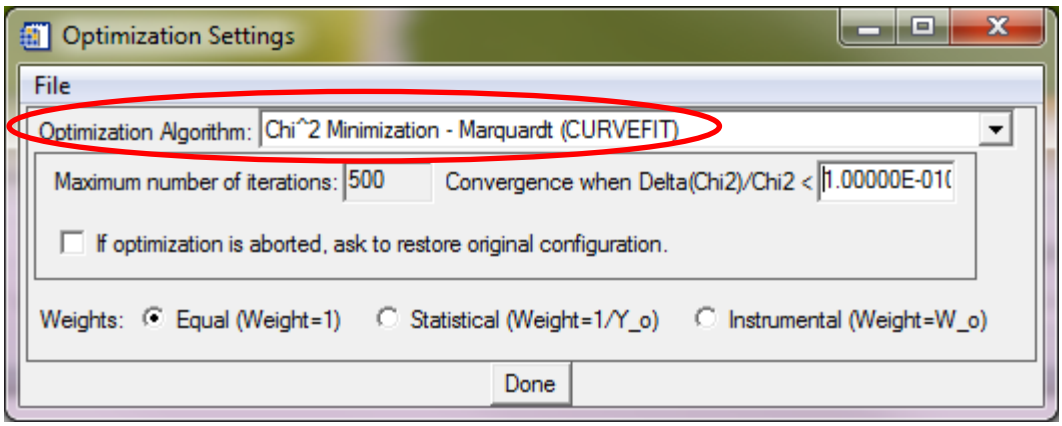
$$S \equiv \sum_{i=1}^{N_{mo}} w[i] \times (Y[i] - Y_{mo}[i])^2 \quad (1)$$

where w[i] are the weighting factors for each point. You can choose for w[i] either (a) Equal weighting, where w[i]=1 for all values of i; (b) Statistical weighting, where w[i]=1/Y_{mo}[i]; or (c) Instrumental

weighting, where w[i]=1/(SIGY_m[i])^2 for curve-fitting, or w[i]=W_o[i] for optimization. In the case of curve-fitting to measured data, if SIGY_m values are not provided (see §8.1.1) then Instrumental weighting will not be available as an option. It is also possible, in the case of curve-fitting to measured data, to specify "Logarithmic Fitting", as we'll see, which means that the numerator in equation (1) is replaced by (ln Y[i] – ln Y_{mo}[i])^2. Furthermore in the case of fitting, you can specify fixed offsets and scaling values to be applied to the optical function being computed, in order to account for known systematic errors in the measured data.

The Fitting Settings and Optimization Settings windows are shown below when the Marquardt algorithm is selected using the drop-lists at the top of each window. You can enter values for the maximum number of iterations, and the convergence criterion. That is, once you start a fit or an optimization, the algorithm will continue to iterate until one of the following conditions are met: (a) the maximum number of iterations (which you specify) has been reached, (b) the convergence criterion (which you specify) has been met, (c) you manually abort the fit or optimization (as we'll explain below in §8.7), or (d) an error was encountered. In the case of the Marquardt algorithm, the convergence criterion will be met when the relative change in the S statistic, $\Delta(\chi^2)/\chi^2$, is less than the value specified, where $\Delta(\chi^2)$ is defined as the difference in the value of the S statistic for the current iteration and that of the previous iteration. In other words, when the relative change in the S statistic defined in equation (1) is less than the value you specify in the Settings window, the algorithm will be considered to have converged. If you plan on stopping the fit or optimization manually, based on your assessment of its progress, enter a large value for the Maximum number of iterations, and a very small value for the Convergence criterion.



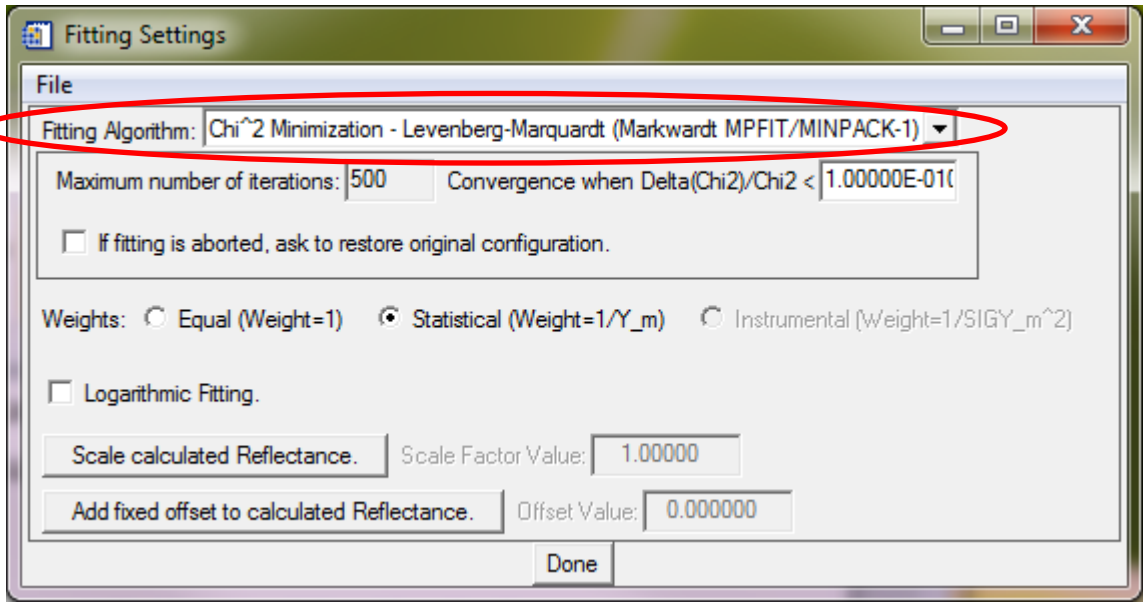


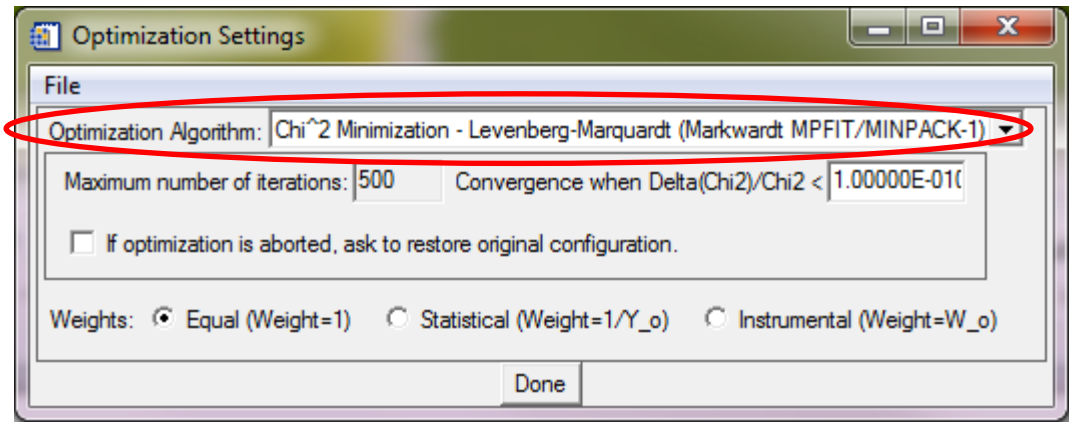
8.6.2 χ^2 Minimization: Levenberg-Markquardt Algorithm (MPFIT/MINPACK-1)

The second algorithm available in IMD is the Levenberg-Marquardt gradient-expansion algorithm; IMD uses the MPFIT program written in IDL by C. Markwardt,[21] which is based on MINPACK-1.[22] From the MPFIT documentation:

> *The Levenberg-Marquardt technique is a particular strategy for iteratively searching for the best fit. This particular implementation is drawn from MINPACK-1 (see NETLIB), and seems to be more robust than routines provided with IDL.*

The Fitting Settings and Optimization Settings windows are shown below when the Levenberg-Marquardt algorithm is selected, using the drop-lists at the top of each window. This algorithm uses the exact same parameters and settings as the Marquardt algorithm described in the previous section.





8.6.3 Genetic Algorithms

Two genetic algorithms are available for fitting and optimization in IMD: the one called Genetic Algorithm (GA), described in §8.6.4, is inspired by the approach described by Binda and Zocchi,[23] while the Differential Evolution (DE) genetic algorithm, described in §8.6.5, is adapted from the approach described by Björck.[24]

Genetic algorithms for numerical optimization are based on the Darwinian principles of genetic inheritance, genetic mutation, natural selection, and survival of the fittest. That is, a starting population of P "individuals" is created, where each individual contains as "genes" the p adjustable parameters whose values are meant to be determined. A figure-of-merit (FOM) value is computed for each individual in the starting population. Those individuals having the best FOM values are retained, and those having the worst FOM values are discarded – survival of the fittest in action. The individuals that are retained are then used to create the next generation of P individuals: each of the p genes in each of the P individuals of the next generation are created by some combination of inheritance and mutation. The process continues until an individual is found whose genes constitute an acceptable solution to the fitting or optimization problem.

Genetic algorithms are generally less sensitive to the choice of initial parameter values, and are less susceptible to the problem of getting stuck at local minima in parameter space than are the gradient-expansion algorithms described in the previous two sections. Nevertheless, genetic algorithms also require that the user specify a number of parameters that control the algorithm's operation, and these parameter should be set with care. To quote Björck, with regard to the DE algorithm of §8.6.5 in particular, but applicable to genetic algorithms more generally:

> *Using DE in a black-box manner bears the risk of obtaining poor performance. To achieve optimal performance of the algorithm, its control parameters may need to be carefully tuned depending on the complexity and difficulty of the problem.*

A complete discussion of genetic algorithm concepts is, again, outside of the scope of this document. We will explain briefly the operation of the GA and DE genetic algorithms as used in IMD, but you are

strongly encouraged to consult references [23] and [24], as well as references cited in those papers, for further detail and recommendations on tuning the parameters that control the algorithm's behavior. In any case, please keep in mind that the default values of the control parameters for the GA and DE algorithms in IMD should not be interpreted as recommended or optimal values for all circumstances.

8.6.4 FOM Minimization: Genetic Algorithm (Binda)

The GA algorithm works as follows. A starting population of P individuals is created, where each individual contains p genes; the genes are the p adjustable parameters you have selected as Fitting Parameters (§8.5.1) or Optimization Parameters (§8.5.2). The values of the p genes in a given individual of the starting population are equal to the Initial values (which you specify) of the corresponding Fit/Optimization parameter, but with a randomly-selected fraction, f_m, of those p genes having values that are modified (mutated) by a random amount, r_m. A figure-of-merit (FOM) function is used to compute the FOM value for each of the P individuals in the starting population, and the population is then sorted by FOM rank. The half of the population having the worst (highest) FOM values is discarded, while the half of the population having the best (lowest) FOM values are retained. The next generation of P individuals is then created by combining the retained (best) half of the starting population with a new population of P/2 individuals having new, randomly-defined genes. The process is repeated until either (a) the maximum number of generations (which you specify) has been reached, (b) the convergence criterion (which you specify) has been met, (c) you manually abort the fit or optimization (as we'll explain below in §8.7), or (d) an error was encountered.

In the case of both the GA and DE algorithms, the FOM function takes the form:

$$\mathrm{FOM} \equiv \frac{\sum_{i=1}^{N_{mo}} w[i] \times |Y[i] - Y_{mo}[i]|^n}{\sum_{i=1}^{N_{mo}} w[i]} \tag{2}$$

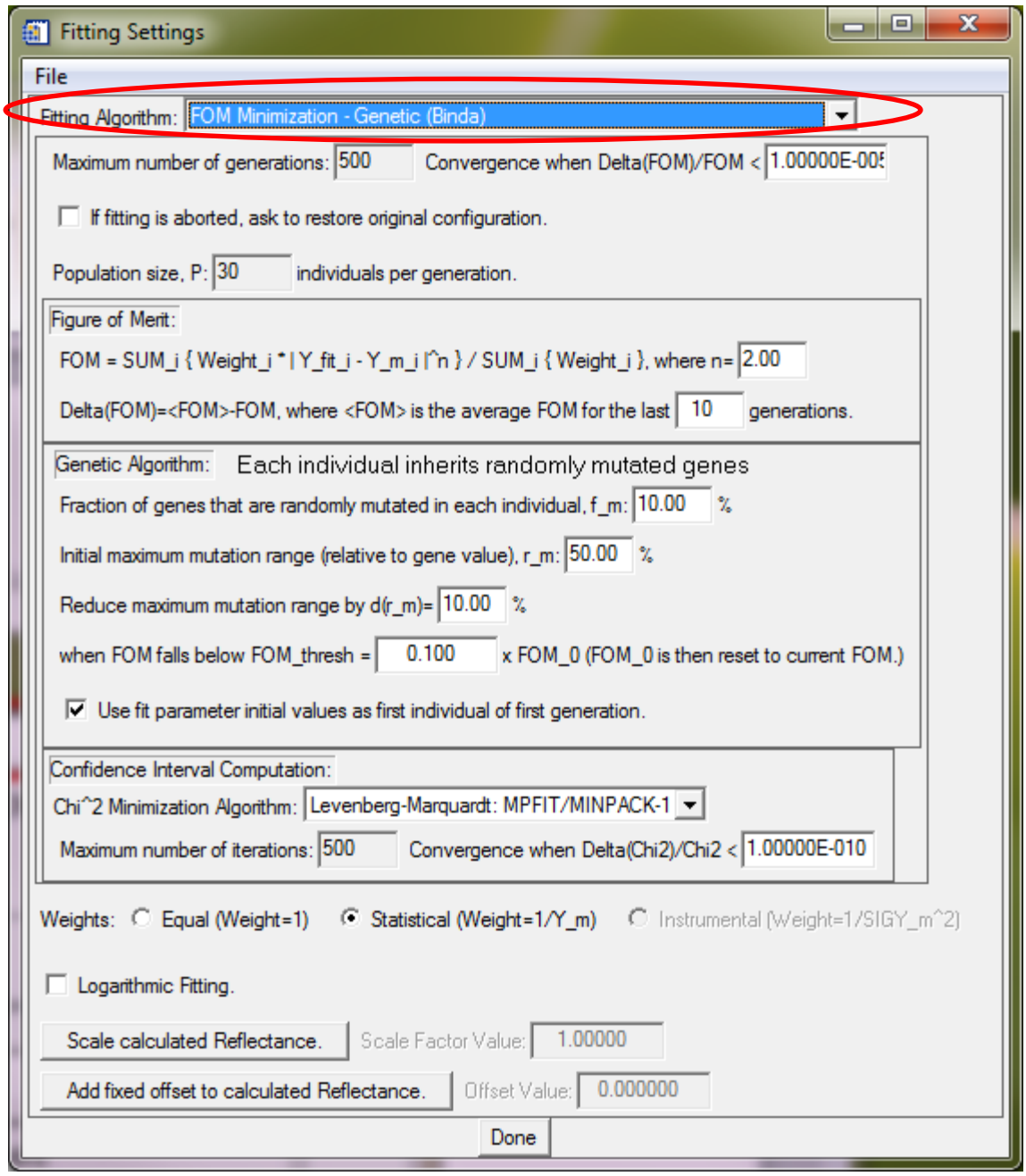
The FOM is similar to the S statistic (equation 1) used in the two gradient-expansion algorithms; the most important difference is that the FOM uses the exponent n, which you specify.

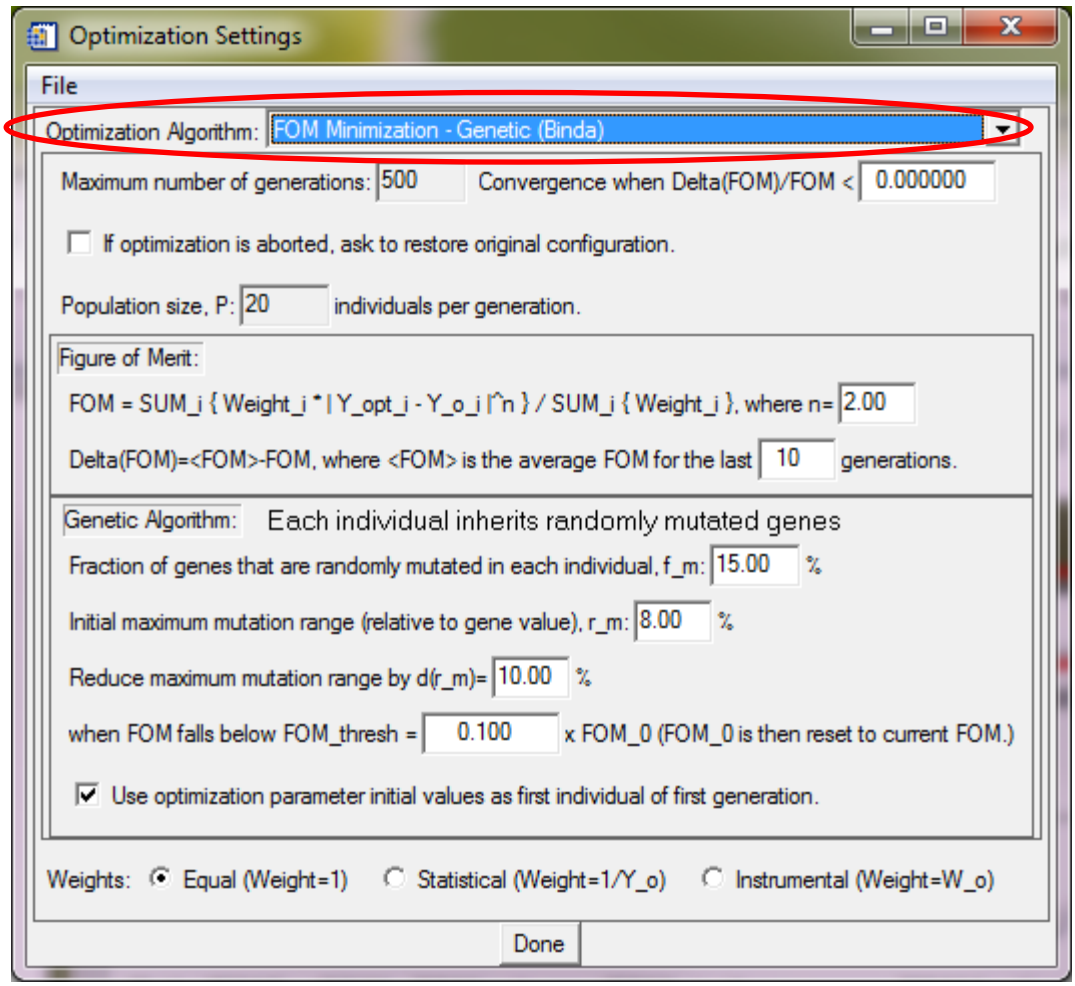
The Fitting Settings and Optimization Settings windows are shown below when the GA algorithm is selected. The Maximum number of generations is analogous to the Maximum number of iterations used for the Marquardt and Levenberg-Marquardt (LM) algorithms. However the Convergence tolerance is somewhat different. As the algorithm proceeds, the value of the FOM will be averaged over the previous X generations, in order to compute the quantity <FOM>. The Convergence tolerance value corresponds to the value of Δ(FOM)/FOM, where Δ(FOM)=<FOM>-FOM, i.e., Δ(FOM)/FOM is the relative change in the FOM using the boxcar average <FOM>, and when Δ(FOM)/FOM falls below the value you specify, the algorithm will have converged. You must therefore also specify the value for X – the number of generations used to compute the boxcar average <FOM> – as well as the exponent n used to compute the FOM itself according to equation (2).

Other parameters that you must specify for the GA algorithm are the population size, P, and the values for f_m and r_m, which control mutation as described above: f_m is the fraction of genes that are randomly mutated in each individual, with allowed values $0 < f_m < 100\%$, and r_m is the mutation range, i.e., the amount that each gene is allowed to vary, specified as a percentage of the genes value, with allowed values $r_m > 0$. Gene values will of course be limited if you specify constraints on any parameter values. The population size P should be set large enough to ensure sufficient genetic diversity in each population; typical values for P are generally several times larger than the number of adjustable parameters p. The f_m and r_m values are critical and should be set with care: they should be large enough to introduce sufficient variability in the population, but not so large that the algorithm never converges.

Finally, you can specify two additional parameters for the GA algorithm, $\delta(r_m)$ and FOM_{thresh}, which control if and how the mutation range, r_m, changes as the algorithm proceeds. That is, it may be advantageous to gradually decrease r_m as the FOM improves, to allow a fine convergence to the best solution. So, specifying a non-zero value for the quantity $\delta(r_m)$ will cause the value of the mutation range, r_m, to decrease by $\delta(r_m)$ per cent when the FOM falls below the value of FOM_{thresh} x FOM_0, where FOM_0 is (initially) the best FOM value from the initial population; whenever r_m is reduced by $\delta(r_m)$ according to this criterion, FOM_0 gets reset to the best FOM value of the current generation. For example, if $\delta(r_m)$ is set to 10% and FOM_{thresh} is set to 0.01, then r_m will be reduced by 10% when the FOM falls to 1% of its initial value, at which point FOM_0 will be reset to the current FOM value (which is equal to or smaller than 1% of the initial FOM determined from the starting population.)

As in the case of the previous two algorithms, you can also specify the Weighting method (equal, statistical, or instrumental), and in the case of Fitting, you can specify Logarithmic fitting, where Y[i] – Y_{mo}[i] in equation (2) is replaced by ln Y[i] – ln Y_{mo}[i], and you can add fixed Scaling and Offset values to account for known systematic errors in the measured data. The GA Fitting Settings window also includes parameters for Confidence Interval Computation, which we'll explain in §8.8.


Fitting Settings
File
Fitting Algorithm: FOM Minimization - Genetic (Binda)
Maximum number of generations: 500
Convergence when Delta(FOM)/FOM < 1.00000E-005
If fitting is aborted, ask to restore original configuration.
Population size, P: 30 individuals per generation.
Figure of Merit:
FOM = SUM_i { Weight_i * | Y_fit_i - Y_m_i |^n } / SUM_i { Weight_i }, where n= 2.00
Delta(FOM)=<FOM>-FOM, where <FOM> is the average FOM for the last 10 generations.
Genetic Algorithm: Each individual inherits randomly mutated genes
Fraction of genes that are randomly mutated in each individual, f_m: 10.00 %
Initial maximum mutation range (relative to gene value), r_m: 50.00 %
Reduce maximum mutation range by d(r_m)= 10.00 %
when FOM falls below FOM_thresh = 0.100 x FOM_0 (FOM_0 is then reset to current FOM.)
Use fit parameter initial values as first individual of first generation.
Confidence Interval Computation:
Chi^2 Minimization Algorithm: Levenberg-Marquardt: MPFIT/MINPACK-1
Maximum number of iterations: 500
Convergence when Delta(Chi2)/Chi2 < 1.00000E-010
Weights: Equal (Weight=1) Statistical (Weight=1/Y_m) Instrumental (Weight=1/SIGY_m^2)
Logarithmic Fitting.
Scale calculated Reflectance.
Scale Factor Value: 1.00000
Add fixed offset to calculated Reflectance.
Offset Value: 0.000000
Done



8.6.5 FOM Minimization: Differential Evolution Algorithm (Björck)

The Differential Evolution (DE) algorithm, which is described by Björck,[24] as implemented in IMD is a somewhat more complicated genetic algorithm, with two sets of populations for each generation: the parent population A={a_0,a_1,…, a_{p-1}} and the trial population B={b_0,b_1,…,b_{p-1}}, each population containing P individuals.[x] The initial parent population, A, is first constructed by assigning to each of the p genes, in each of the P individuals of the population, a random number between the allowed minimum and maximum values – i.e., the parameter constraint values that you specify for each adjustable parameter on the Fitting Parameter or Optimization Parameter windows (§8.5). (If no constraints are specified, the allowed range will be very large, and there may be performance consequences.)

The trial population, B, is formed from the parent population, A, through the application of mutation and recombination operators, as illustrated conceptually in Figure 40. (Figure 40 is identical to Figure 1 of Björck,[24] except we're using different notation in this document.) Each individual a_i in A serves as a

[x] We'll be "borrowing" much of the language used in reference [24] to describe the DE algorithm, although we're using different notation in this document.

parent for one new trial individual b_i in B. After the trial population is created, each of its individuals is compared with its corresponding parent, i.e., a_i is compared with b_i, through the FOM function shown in equation (2). The individual with the lower FOM is selected to move on to the parent population of the next generation and the process is repeated. Thus, only trials that represent an improvement over their respective parents are allowed to propagate; others are discarded and the better fitting parents are kept instead.

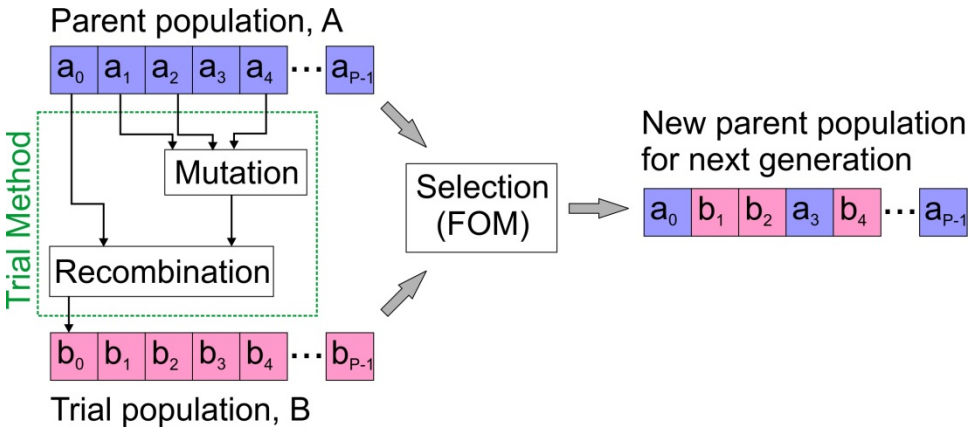


Figure 40. A schematic diagram of the Differential Evolution algorithm over one generation.

The particular process that creates the trial population is called the trial method. The DE trial methods used in IMD are described in detail in reference [24], and will only be described in outline here; we will relate the settings options available on the Fitting Settings and Optimization Settings windows when the DE algorithm is selected, as shown in the two figures below, to the trial method names used in reference [24].

As mentioned above, the trial population is created by a combination of mutation and/or recombination operators. The differential mutation operator represents the difference between two randomly selected parent individuals, having indices r1 and r2. This difference is then scaled by the mutation constant k_m and added to a base vector, $\mathbf{a}_{base}$, to form the mutated individual **m**, where $\mathbf{m}=\mathbf{a}_{base}+k_m(\mathbf{a}_{r1}-\mathbf{a}_{r2})$. Choosing the base vector at random produces mutations with a broad search radius. To make the search more aggressive, the base vector can be chosen to be the best individual so far. These two trial methods are called "rand1bin" and "best1bin", respectively, in Björck. They correspond to the two "Trial Methods" that can be selected from the drop-lists on the Fitting Settings and Optimization Settings windows shown below. The first choice on the drop-list is "Base vector is the best individual of the parent population", which is Björck's "best1bin" trial method. The second choice on the drop-list is "Base vectors are chosen at random from the parent populations", which corresponds to Björck's "rand1bin" trial method. The value for k_m is entered in the box labeled "Mutation constant, k_m:" on the Fitting Settings and Optimization Settings windows.

The recombination operation combines the parameters in the current parent vector with the mutant vector. The probability that a mutant parameter will be inserted for the j^{th} parameter in the i^{th} individual is given by the constant k_r. The final trial vector then becomes

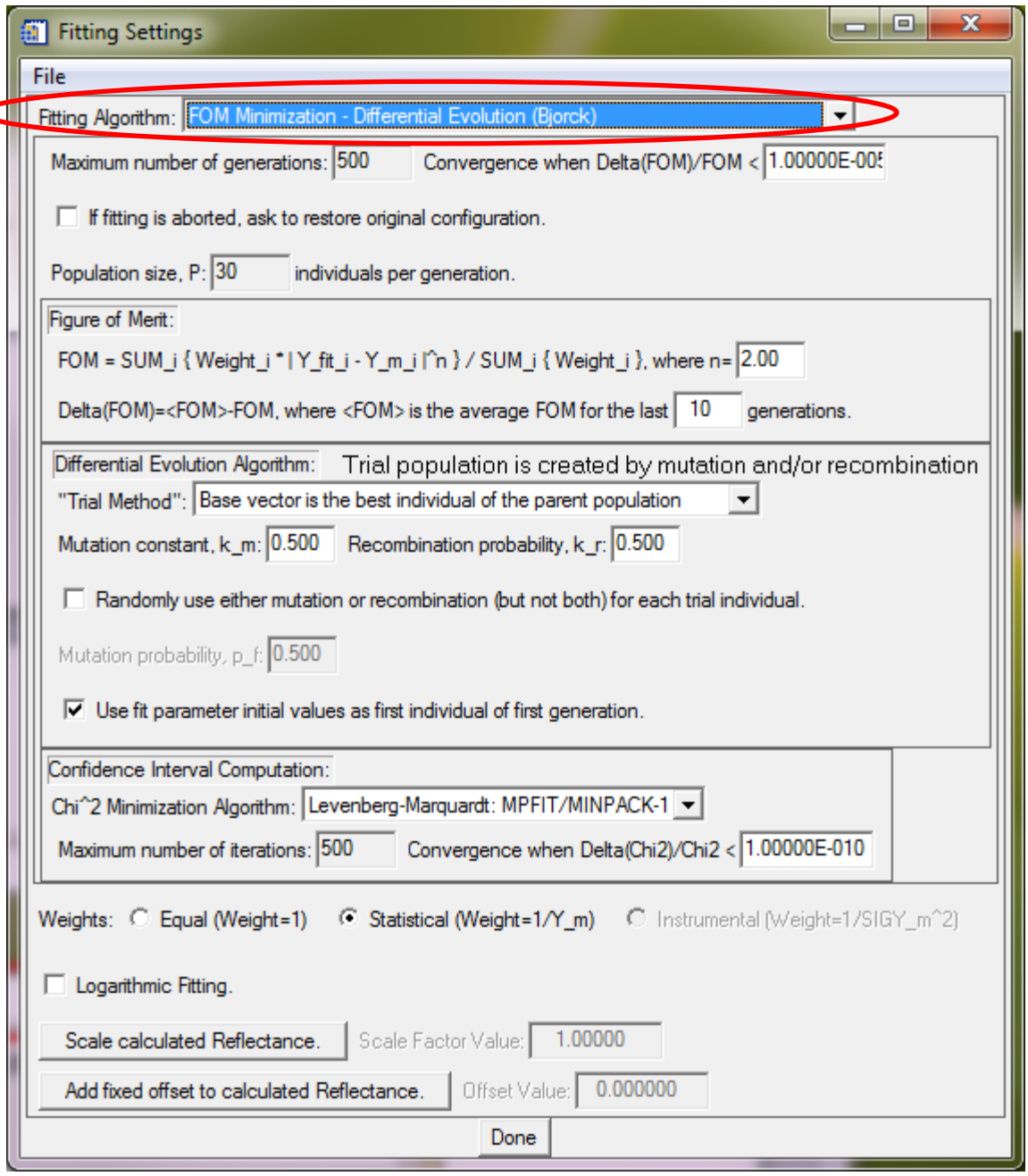
$$\mathbf{b}_{ij} = \begin{cases} \mathbf{m}_{ij} & \text{if random-number} < k_r \\ \mathbf{a}_{ij} & \text{otherwise} \end{cases}$$

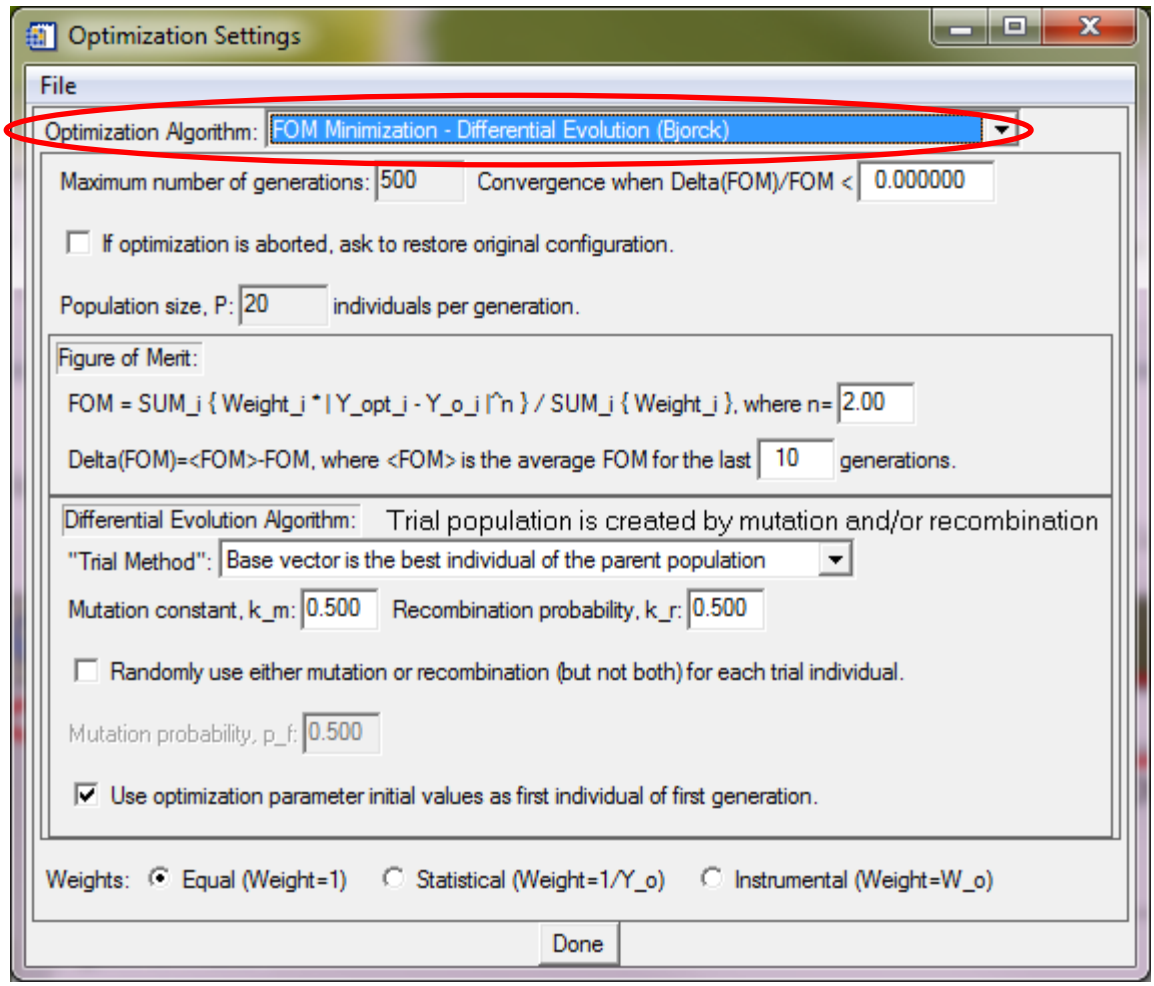
where "random-number" is a randomly-generated number created for each i,j. The value for k_r is entered in the box labeled "Recombination probability, k_r:" on the Fitting Settings and Optimization Settings windows.

Finally, the selected trial method can be modified further (e.g., to avoid the performance loss caused by parameter interdependence) by checking the box labeled "Randomly use either mutation or recombination (but not both) for each trial individual", which causes the use of vector differences for the recombination as well. This selection corresponds to Björck's "rand either/or" method, and results in

$$\mathbf{b}_i = \mathbf{a}_{base} + \begin{cases} k_m(\mathbf{a}_{r1} - \mathbf{a}_{r2}) & \text{if random-number} < p_f \\ k_r(\mathbf{a}_{r1} + \mathbf{a}_{r2} - 2\mathbf{a}_{base}) & \text{otherwise} \end{cases}$$

where p_f is the probability for mutation only, and 'random-number' is a random number generated for each $\mathbf{b}_i$. The value of p_f is specified in the box labeled "Mutation probability, p_f:".


Fitting Settings
File
Fitting Algorithm: FOM Minimization - Differential Evolution (Bjorck)
Maximum number of generations: 500 Convergence when Delta(FOM)/FOM < 1.00000E-005
If fitting is aborted, ask to restore original configuration.
Population size, P: 30 individuals per generation.
Figure of Merit:
FOM = SUM_i { Weight_i * | Y_fit_i - Y_m_i |^n } / SUM_i { Weight_i }, where n= 2.00
Delta(FOM)=<FOM>-FOM, where <FOM> is the average FOM for the last 10 generations.
Differential Evolution Algorithm: Trial population is created by mutation and/or recombination
"Trial Method": Base vector is the best individual of the parent population
Mutation constant, k_m: 0.500 Recombination probability, k_r: 0.500
Randomly use either mutation or recombination (but not both) for each trial individual.
Mutation probability, p_f: 0.500
Use fit parameter initial values as first individual of first generation.
Confidence Interval Computation:
Chi^2 Minimization Algorithm: Levenberg-Marquardt: MPFIT/MINPACK-1
Maximum number of iterations: 500 Convergence when Delta(Chi2)/Chi2 < 1.00000E-010
Weights: Equal (Weight=1) Statistical (Weight=1/Y_m) Instrumental (Weight=1/SIGY_m^2)
Logarithmic Fitting.
Scale calculated Reflectance. Scale Factor Value: 1.00000
Add fixed offset to calculated Reflectance. Offset Value: 0.000000
Done



For the DE algorithm you can specify FOM parameters, exactly as in the GA algorithm described in the previous section, the Weighting method (equal, statistical, or instrumental), and in the case of Fitting, you can specify Logarithmic fitting, where Y[i] $-Y_{mo}[i]$ in equation (2) is replaced by ln Y[i] – ln $Y_{mo}[i]$. You can add fixed Scaling and Offset values to account for known systematic errors in the measured data as well. The DE Fitting Settings window also includes parameters for Confidence Interval Computation, which we'll describe in §8.8.

8.7 Performing the Fit or Optimization

The general procedure for performing a fit or optimization in IMD is as follows:

1. Define the Structure.
2. Import Measured Data for fitting, or Import/Create an Optimization Target Profile for optimization.
3. Define the Dependent and Independent Variables. These variables must correspond to your measured data (fitting) or optimization target profile (optimization) variables, and furthermore

the range of the independent variable should span the same (or a larger) range as the measured data/target profile. If it does not, then only the overlapping portion of the measured data/target profile will be used for the fit/optimization.

4. Specify Fitting Parameters (fitting) or Optimization Parameters (optimization).
5. Select an algorithm and specify algorithm-specific settings.
6. Select "Calculate→Fit to Measured Data" (fitting) or "Calculate→Optimize Specular Optical Function" (optimization).

In the sections that follow we'll demonstrate how to perform a fit or optimization in IMD following the procedures just listed, through the use of some examples.

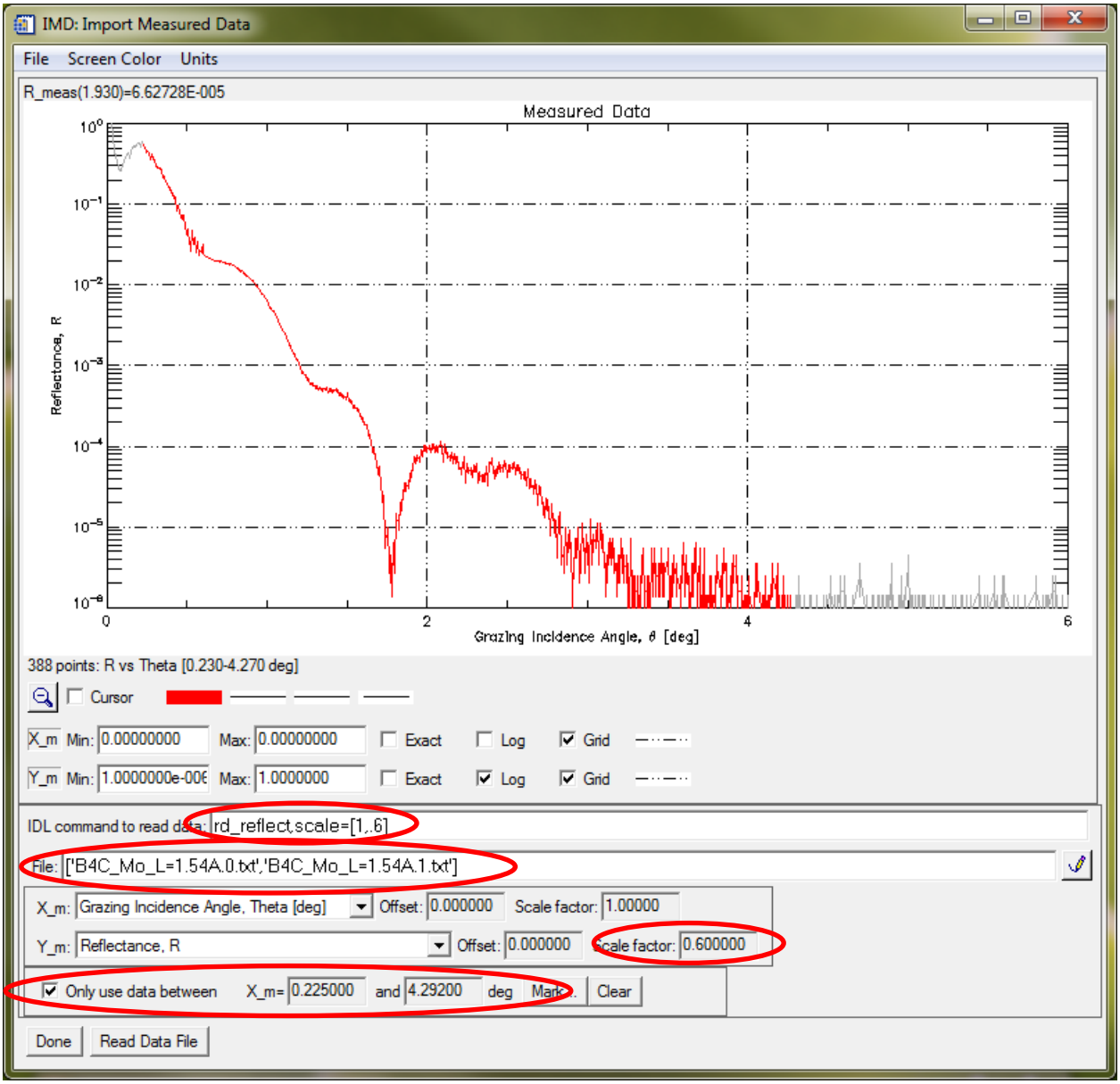
8.7.1 Example: Fitting XRR Data for a B_4C/Mo Bilayer (Part 1)

For our first fitting example we'll endeavor to fit XRR data obtained for a B_4C/Mo bilayer that was deposited onto a polished Si wafer. The X-ray reflectance of this film was measured from θ=0 to 4°, using radiation of wavelength λ=1.54 Å. The target thickness for both the B_4C and Mo layers was z=50 Å, however the deposition rates were not known accurately at the time the film was deposited; actual layer thicknesses determined from fitting the XRR data can be used to calibrate the deposition rates.

As this is the first example of fitting or optimization presented in this document, there will be several new features to explore, as you'll see. We're going to proceed somewhat slowly, therefore, with a few side trips along the way. To follow along with this example, change your working directory in IMD to '`imd/examples/Fitting`', and open the file '`B4C_Mo_L=1.54A.imd`' located in that directory.

The Import Measured Data window containing the measured data is shown in the next figure. First, notice that the box labeled 'Only use data between' is checked, so that we ignore the data below θ=0.225°, where the beam falls off the sample – we don't want to try to fit that data in IMD. We've also put an upper limit on the XRR data (θ=4.292°) to be used, since the data at the larger angles is just noise, and we don't want to try to fit the noise either. Second, this example uses the 'rd_reflect' program (explained fully in §11.4.2) to concatenate two measured data files, `B4C_Mo_L=1.54A.0.txt` and `B4C_Mo_L=1.54A.0.txt`, which correspond, respectively, to measurements made below θ=0.6° using an X-ray attenuating filter to avoid saturating the detector, and measurements made above θ=0.6° without any attenuator. So, in the box labeled 'File:' instead of a simple file name, we've entered a two-element string array (using valid IDL syntax) to specify the two file names just mentioned. However there's a problem with the data in these files: the relative normalization of the two data sets was computed incorrectly when the data was originally acquired, causing a discontinuity at θ=0.6°. To compensate for this error, we've used rd_reflect's SCALE keyword: the SCALE keyword, if used, should be set to an array of the same length as the string array used to specify the two files; each file read by rd_reflect is then scaled by the corresponding value. In this case, we've specified SCALE=[1.,0.6], and so the Y_m values in the first file are scaled by 1.0 and those in the second file are scaled by 0.6. Finally,

we've applied an overall Scale factor of 0.6 to the entire data set (so that the Y_m values in the first file are ultimately scaled by 1.0x0.6=0.6, and those of the second file are scaled by 0.6x0.6=0.36):



The example file '`B4C_Mo_L=1.54A.imd`' already has the correct variables defined (i.e., Reflectance vs. Graze Angle), and the structure is defined using the nominal layer thicknesses. The measured data and the calculated reflectance using these (incorrect) nominal thicknesses are shown in the next figure:

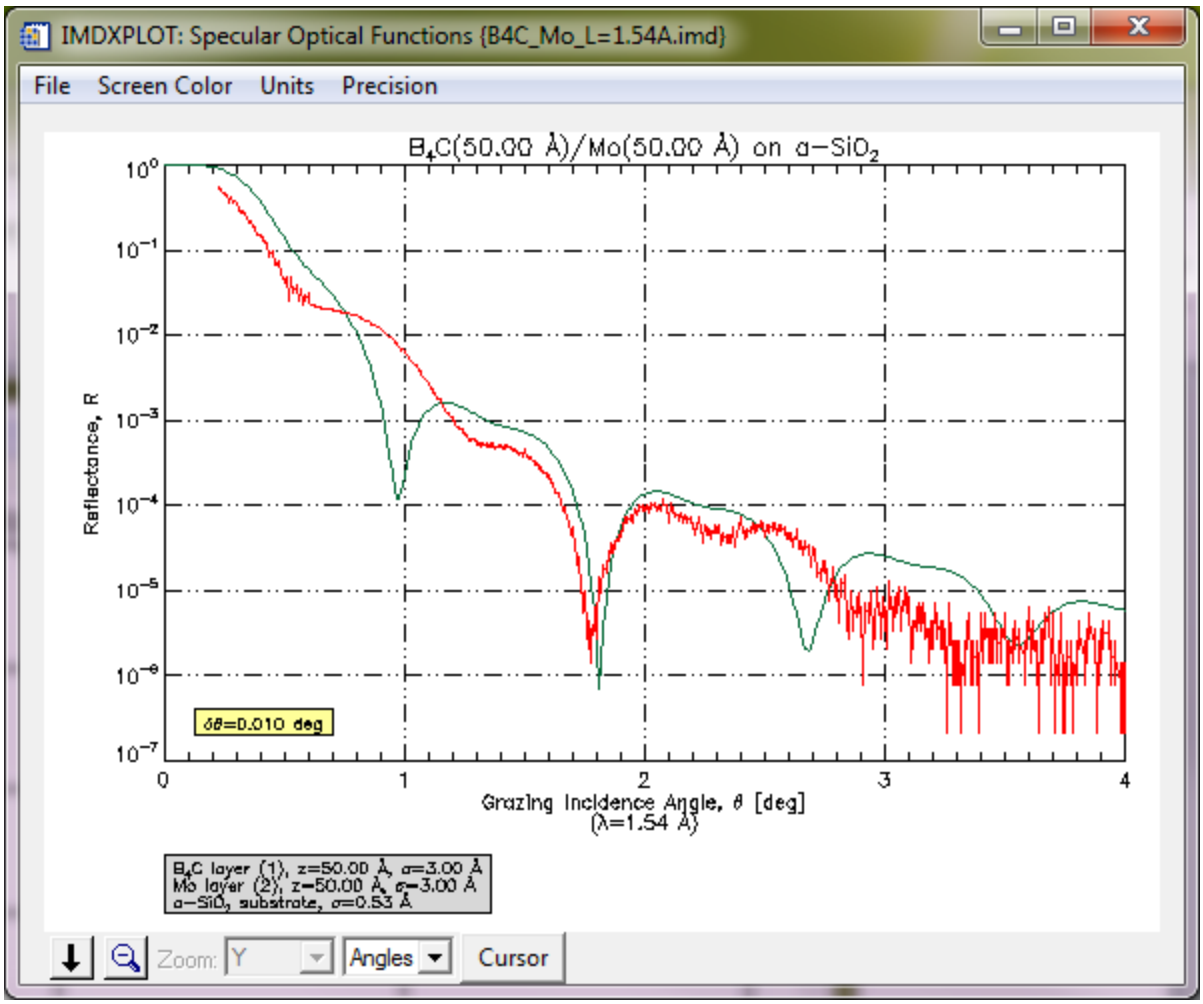
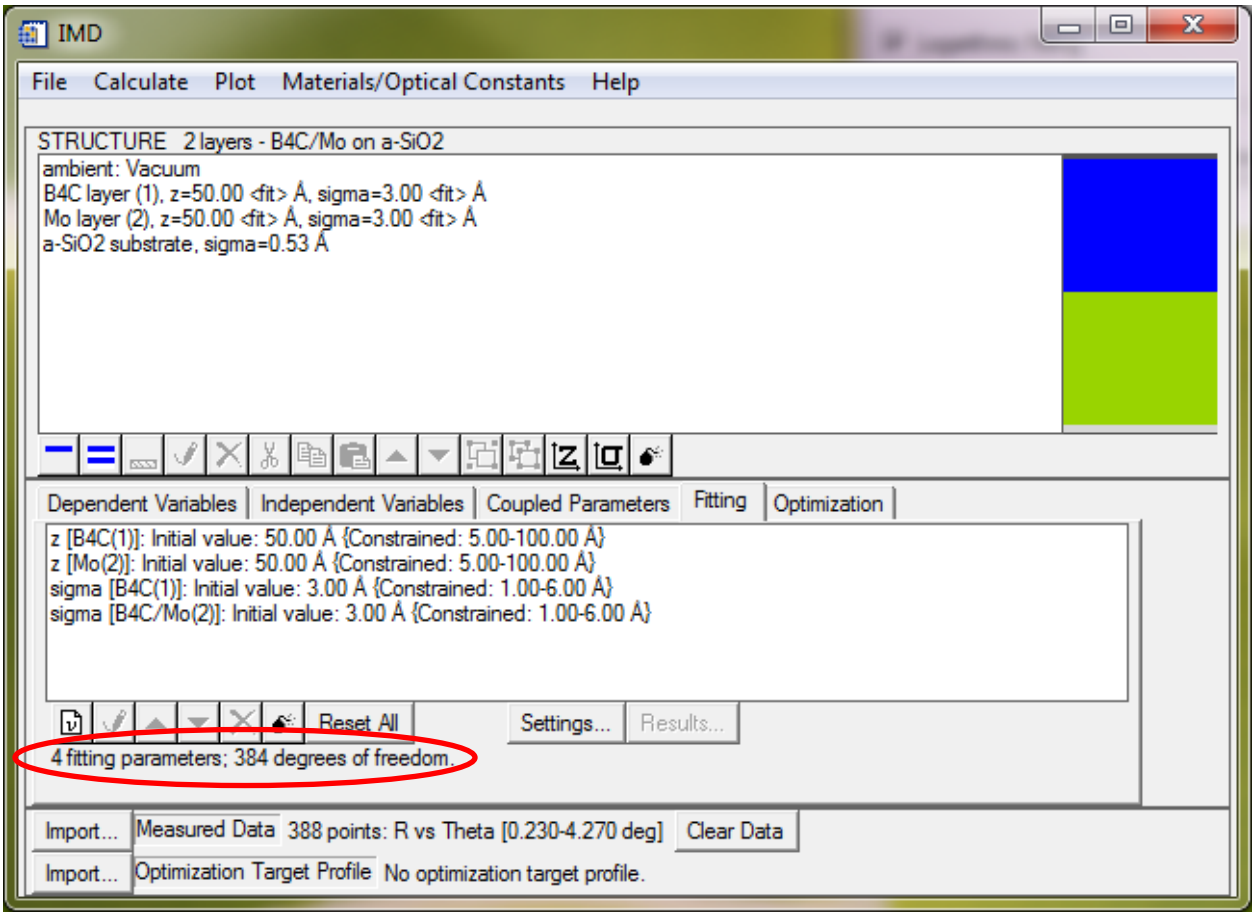
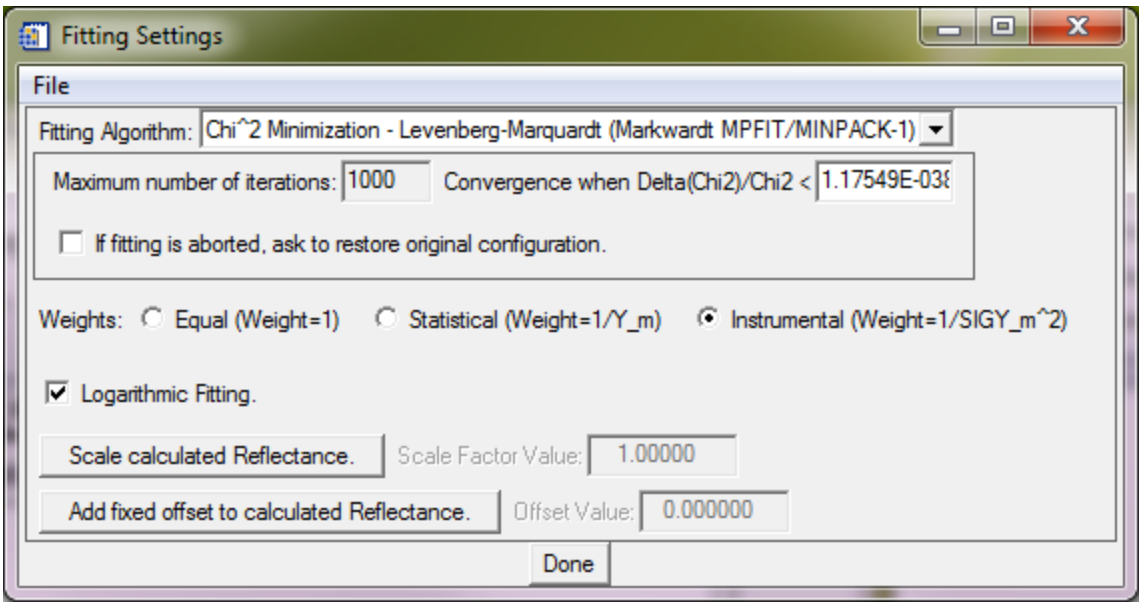


Figure 41. Measured XRR data and calculated reflectance using nominal layer thicknesses. The nominal layer thickness are clearly incorrect.

Next we define four fitting parameters: B_4C layer thickness, Mo layer thickness, B_4C surface roughness and B_4C/Mo interface width. With 388 data points and 4 fitting parameters, we're left with 384 degrees of freedom:



Finally, we select the Levenberg-Marquardt algorithm for fitting, using Instrumental weighting and Logarithmic fitting. Since we'll plan to interrupt the fit manually as we watch it proceed, we've specified 1000 iterations and a tiny Convergence tolerance:



Everything is now configured and so we're ready to perform the fit. (See the example file '`B4C_Mo_L=1.54A_LM_0.imd`', which represents the configuration up to this point.) We select from the IMD menu bar "Calculate→Fit to Measured Data". A new 'Fitting Results' window appears, as shown in the first figure of the next section.

8.7.2 The Fitting (or Optimization) Results Window

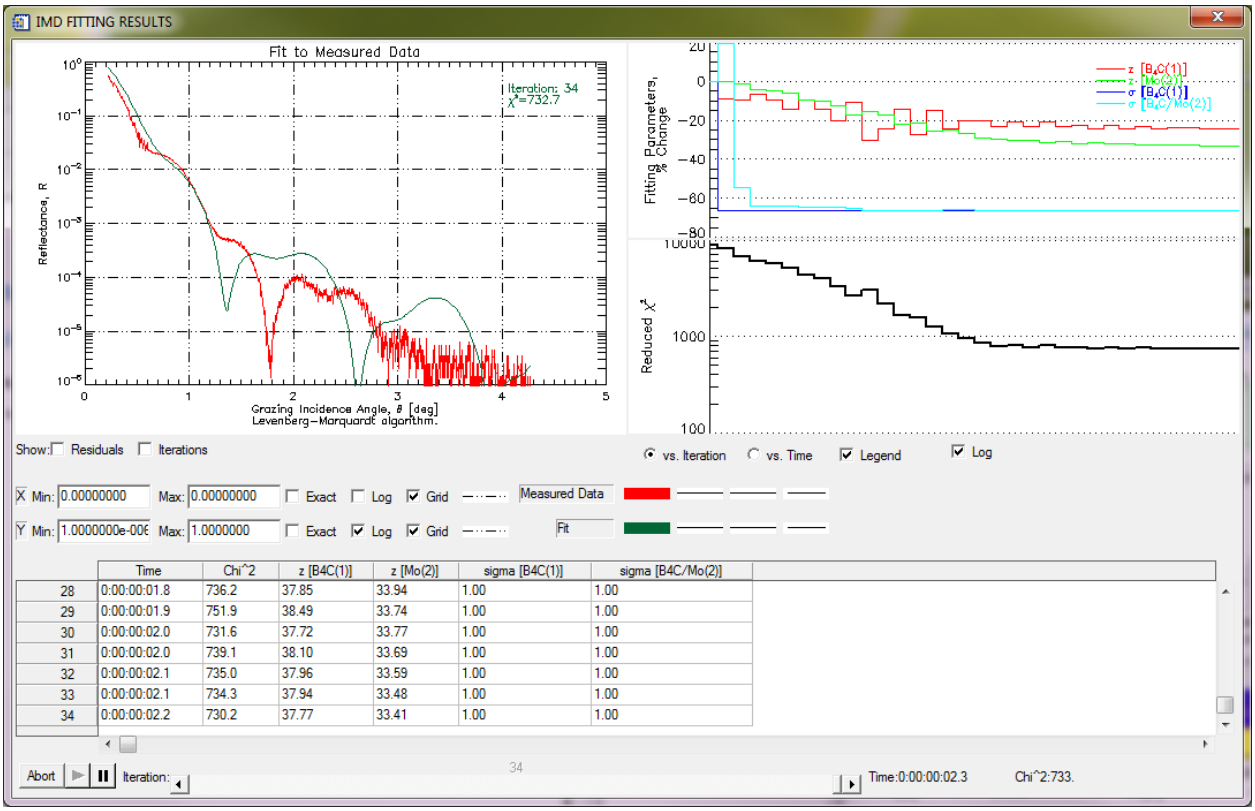


Figure 42. The Fitting Results window as it appears during a fit.

The Fitting Results window shown above displays the fit status as it progresses, and allows for the fit to be paused or aborted. (An equivalent window is used for Optimization Results.) The computed data for the current iteration (or generation, if a genetic algorithm is being used) and the measured data are shown graphically in the large plot on the left. The value of the relative change (in per cent) for each of the fitting parameters is plotted vs. iteration/generation (or vs. time, if the radio button labeled 'vs. Time' is checked), in the plot on the upper right. The quality of fit – Reduced χ^2, FOM, or Δ(FOM)/FOM, depending on fitting algorithm – is plotted on the lower right. Each row of the table comprising the bottom half of the Fitting Results window lists the fitting parameter values for one iteration/generation. You can pause the fit by clicking the pause button at the lower left (⏸): when fitting is paused, you can scroll through iterations/generations completed so far, by using the slider at the bottom of the window, or by selecting rows in the table. Use the play button (▶) to resume fitting. If the algorithm hasn't yet converged, you can abort the fit by clicking the Abort button at the lower left. Aborting a fit simply means that the fitting is stopped at the current iteration, but the results of the fit up that point are preserved. (If you check the box labeled "If fitting is aborted, ask to restore original configuration." on the Fitting Settings window, if a fit is aborted you will be asked if you want to restore the original configuration rather than preserve the results of the fit.)

We'll explain more about the Fitting Results and Optimization Results windows later, but let's first continue with the B_4C/Mo fitting example…

8.7.3 Example: Fitting XRR Data for a B_4C/Mo Bilayer (Part 2)

Continuing with our B_4C/Mo bilayer example of §8.7.1: when allowed to continue, the fitting algorithm does, in fact, converge, after 247 iterations. (See the example file 'B4C_Mo_L=1.54A_LM_1.imd'.) When the fit converges (or when fitting is stopped for any other reason), a new IMDXPLOT window opens displaying the result of the fit. Here's how the fit looks for this example after it has 'converged':

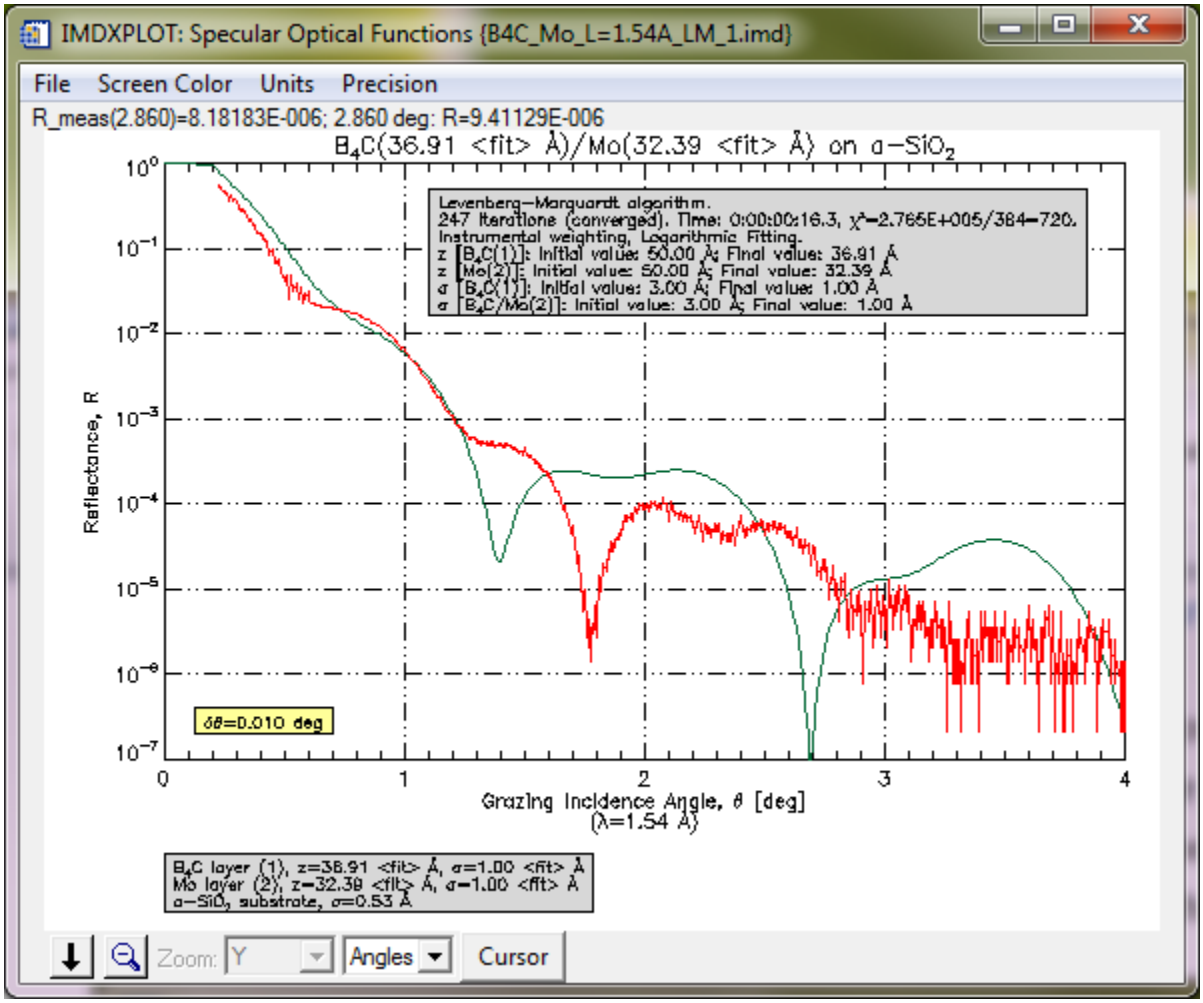
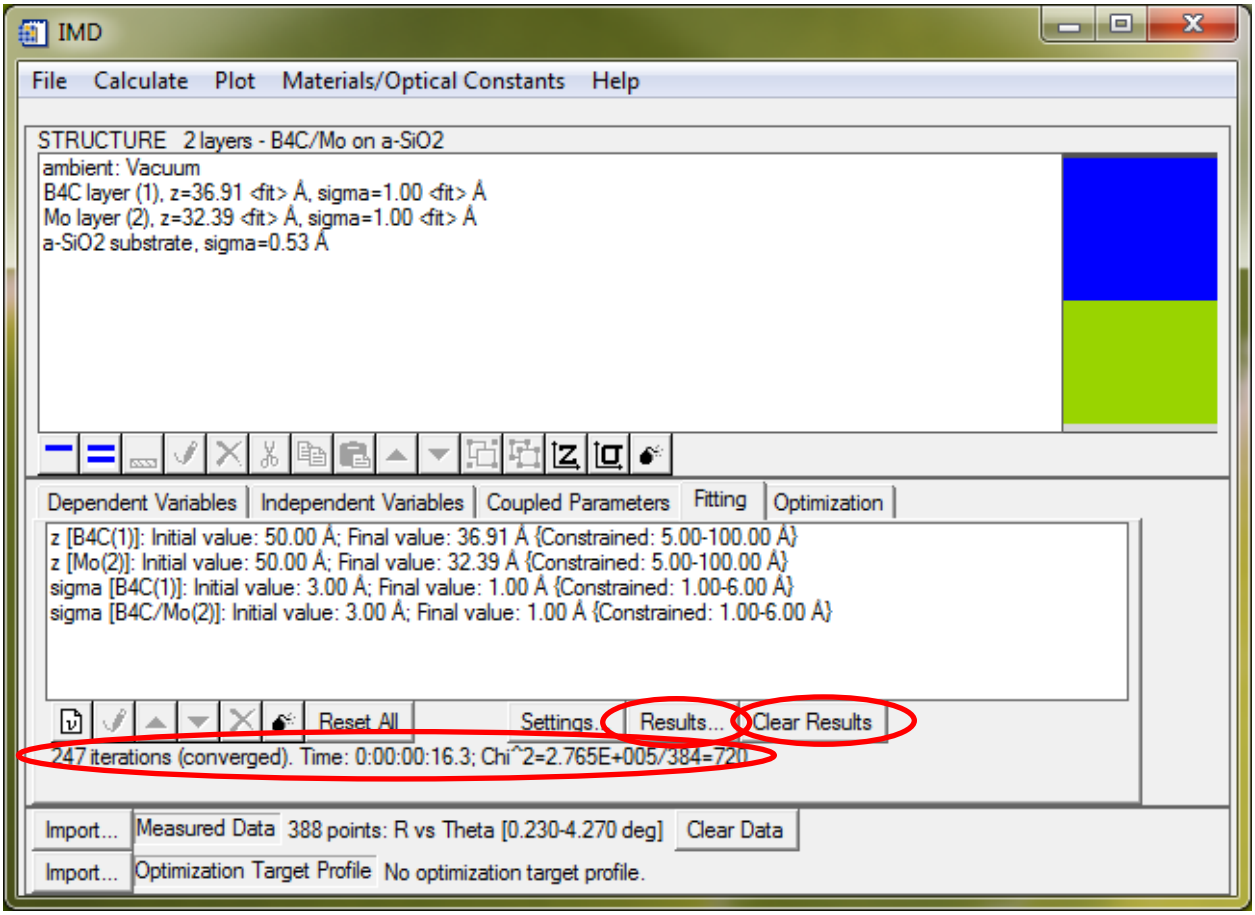


Figure 43. The results of fitting the B_4C/Mo XRR data using the Levenberg-Marquardt algorithm.

Obviously this is not a very good fit, even though the algorithm decided that it converged (because $\Delta(\chi^2)$ approached zero). This particular example is a nice illustration of the main drawback of the Levenberg-Marquardt and Marquardt gradient-expansion algorithms: if the initial values of the adjustable parameters are chosen poorly, i.e., near local minima in parameter space, the algorithm is susceptible to convergence at one of the local minima rather than at the global minimum. We'll see how genetic

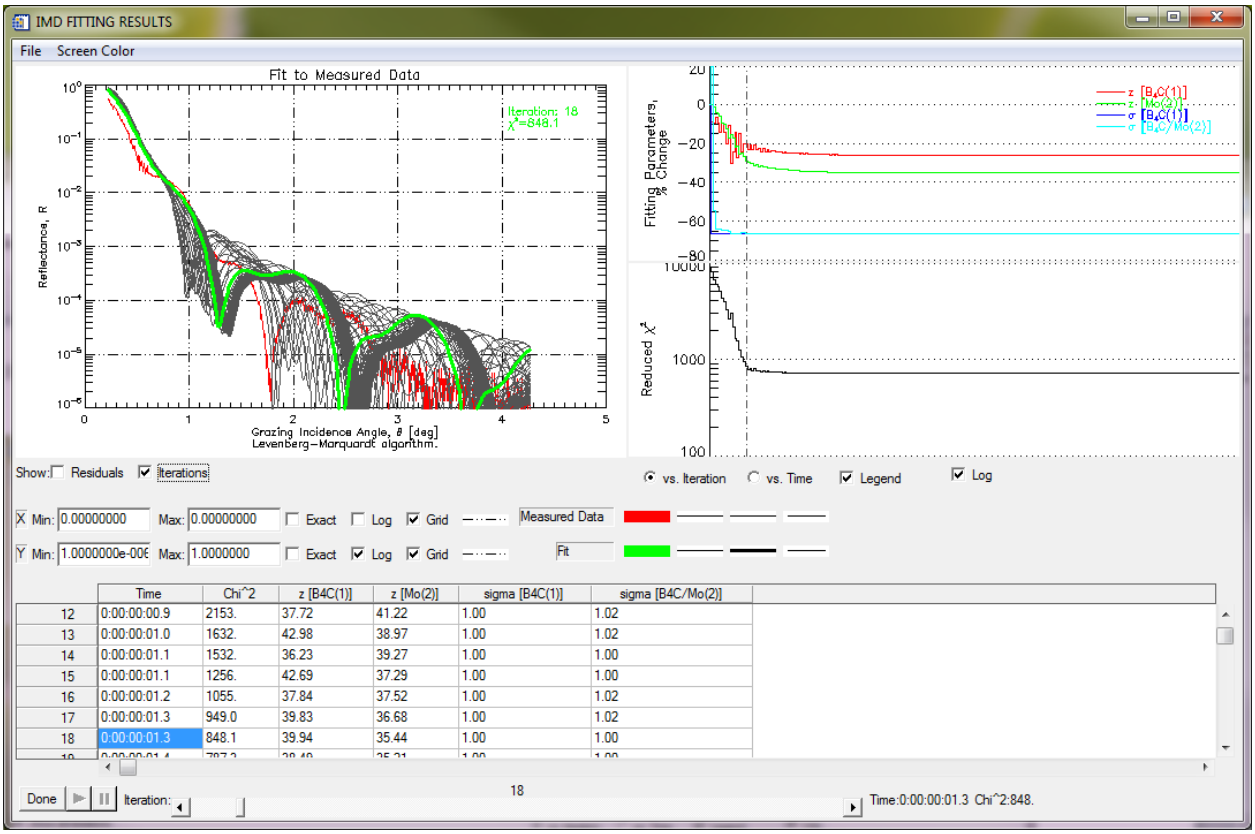
algorithms perform under such circumstances in §8.7.4. But first let's continue with this example, so we can see how a fit works all the way to the end.

Here's a look at the main IMD window after the fitting has been completed:



The status area at the bottom of the Fitting tab now shows the number of iterations (247), whether or not the algorithm converged (it has), the elapsed time (16.3 seconds), the final value of χ^2 (2.765×10^5), the number of degrees of freedom (384), and the Reduced χ^2 ($2.765 \times 10^5/384=720$). The final parameter values are shown in the list of fitting parameters. And two buttons on the Fitting tab are now sensitive: the "Results…" button and the "Clear Results" button, both highlighted above in red.

Clicking the "Results…" button opens up a new Fitting Results window. As the fit is no longer running at this point, there are no Pause and Play buttons – you can just scroll back through iterations/generations using the slider or the table, as before. In the figure below, we've scrolled back to Iteration #18. We've also checked the "Show: iterations" button, whose purpose should be apparent:



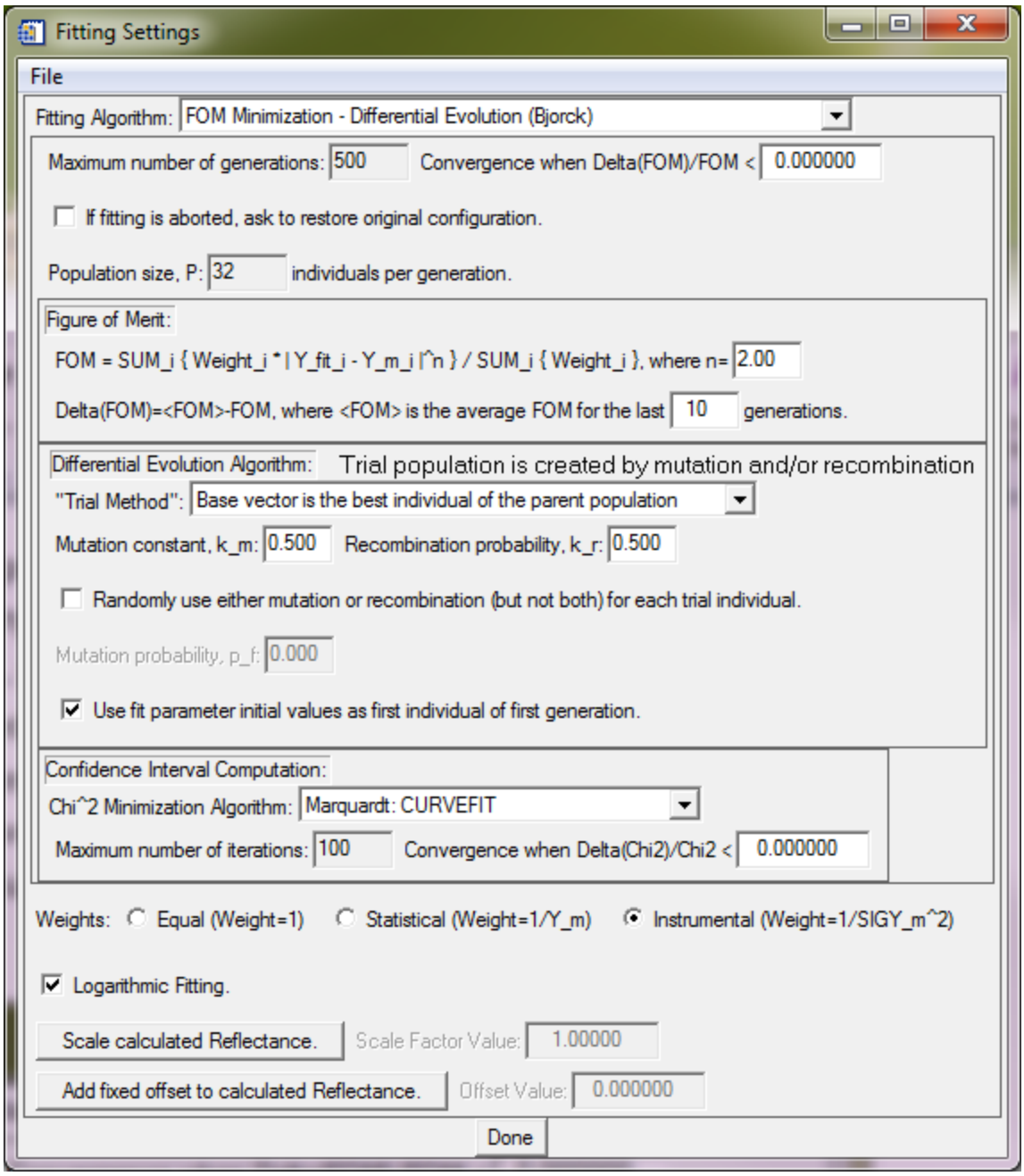
Use the Fitting Results window File menu to (a) print the plot of measured and computed data, or (b) export the fitting parameters vs. iteration/generation to a text file, or (c) export all the fitting results to an IDL save file.

> **Note**: On linux systems that have the '/usr/bin/mpeg_encode' executable installed, you will also have the option from the File menu to "Create Fitting Results Movie": selecting this option will create an mpeg file showing the computed and measured data vs. iteration/generation as an animation.

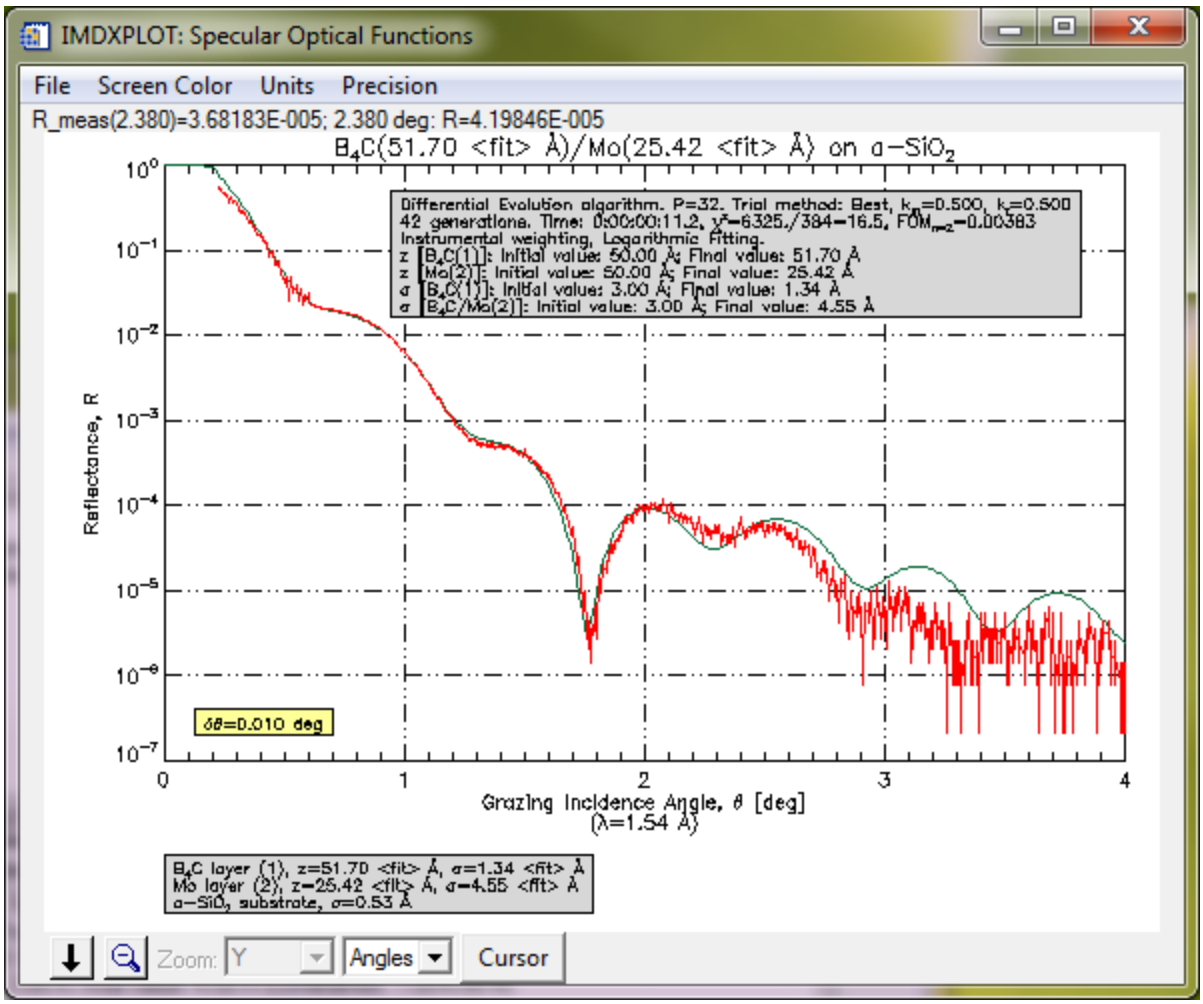
Please keep in mind that the data displayed in the Fitting Results window – and in the analogous Optimization Results window discussed in §8.7.6 – can consume considerable computer memory, especially for a large number of data points, fit/optimization parameters, and/or iterations/generations. If you save to a .imd file after completion of a fit or optimization, the fitting/optimization results data will be included with the .imd file, so that when that .imd file is re-opened you can re-display the fitting/optimization results data whenever you click the "Results…" button on the Fitting or Optimization Tab. However this data will also be included in any IMDXPLOT windows you may open, and because of the large amount of memory consumed by the fitting/optimization results, there may be performance problems as a result of having to pass that much data around. If you no longer need the fitting or optimization results data once the fit or optimization is completed, you can delete it by clicking the "Clear Results…" button on the Fitting or Optimization tab before saving to a .imd file.

8.7.4 Example: Fitting XRR Data for a B_4C/Mo Bilayer (Part 3)

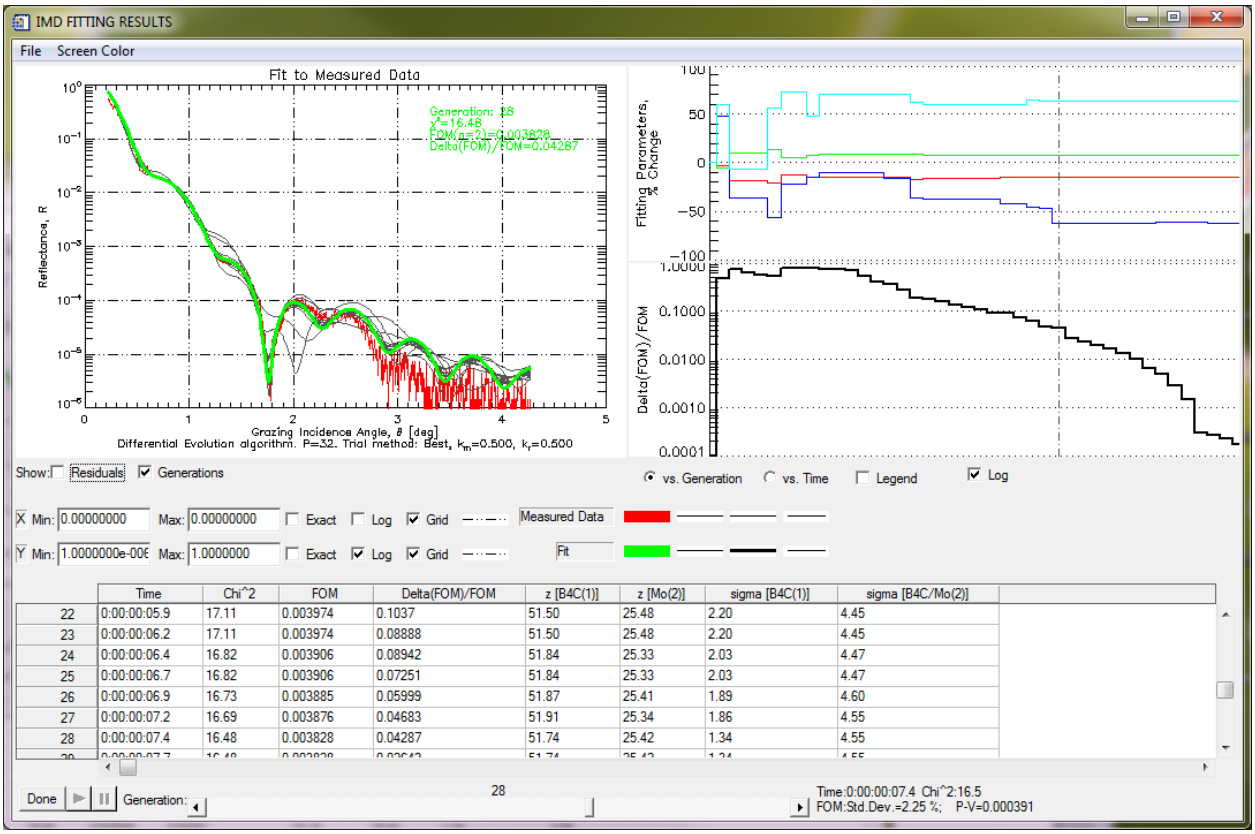
The Levenberg-Marquardt algorithm did a poor job of fitting the XRR data from the previous section, as can be seen in Figure 43. Let's instead try to fit these data using the DE genetic algorithm. In the example file "`B4C_Mo_L=1.54A_DE_0.imd`" we've defined the same structure, variables and fitting parameters as before, but we've selected the Differential Evolution algorithm for fitting, with a population size of P=32 individuals, Convergence criterion of 0 (we'll stop the fit manually), an FOM exponent n=2, with <FOM> computed over the past 10 generations, the "best1bin" trial method, mutation constant k_m=0.5, and recombination probability k_r=0.5. The Fitting Settings window so-configured is shown in the next figure:



We then select from the IMD menu bar "Calculate→Fit to Measured Data". The DE algorithm converges quickly – we observed during the fitting that the parameter values stabilized around generation 28, and the fit itself looked pretty good, so we aborted soon after, at 42 generations:

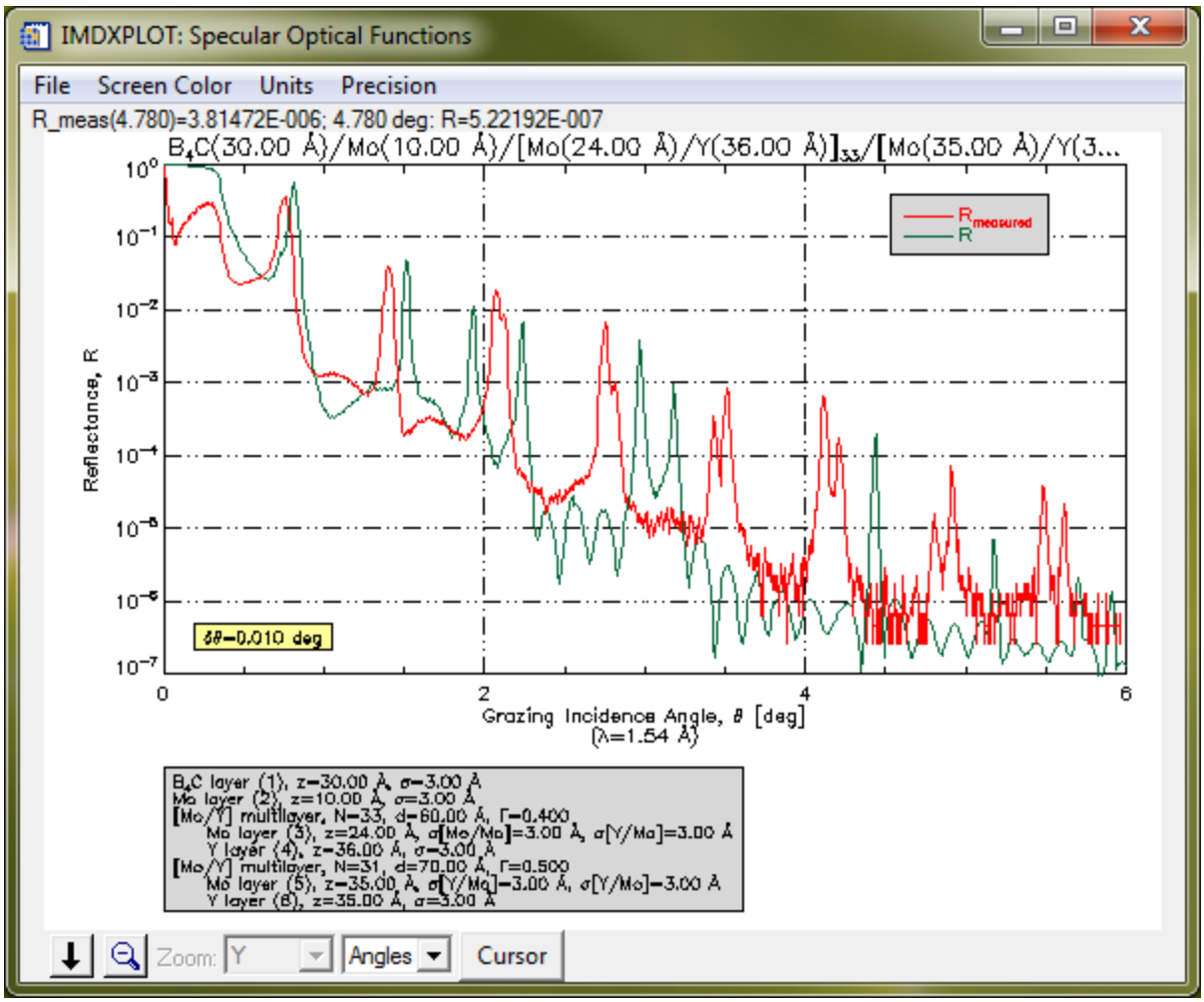


The Fitting Results window in the next figure (displayed by clicking the "Results…" button on the Fitting tab) shows the fit history:

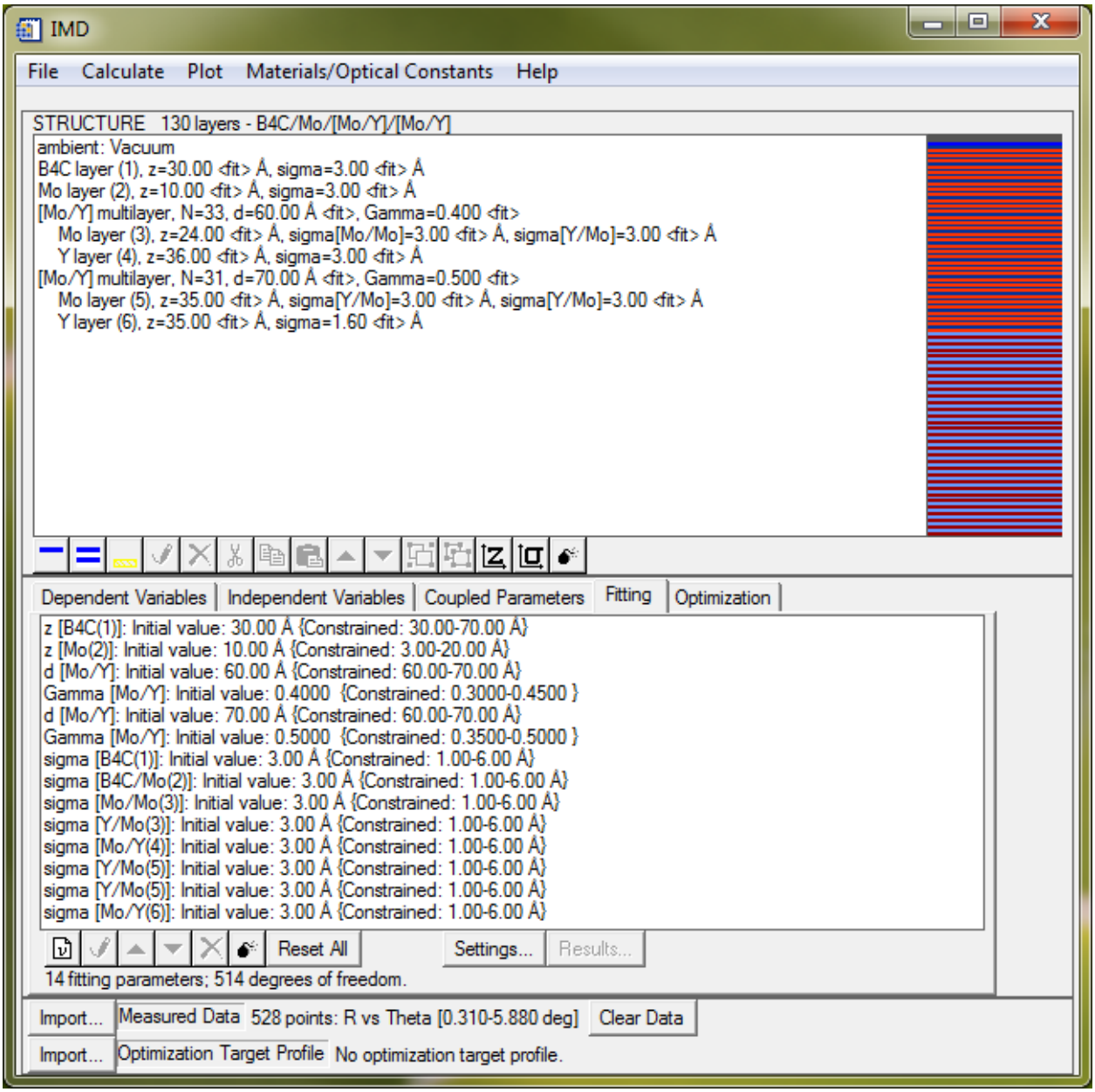


8.7.5 Example: Fitting XRR Data for a Mo/Y Double-Multilayer

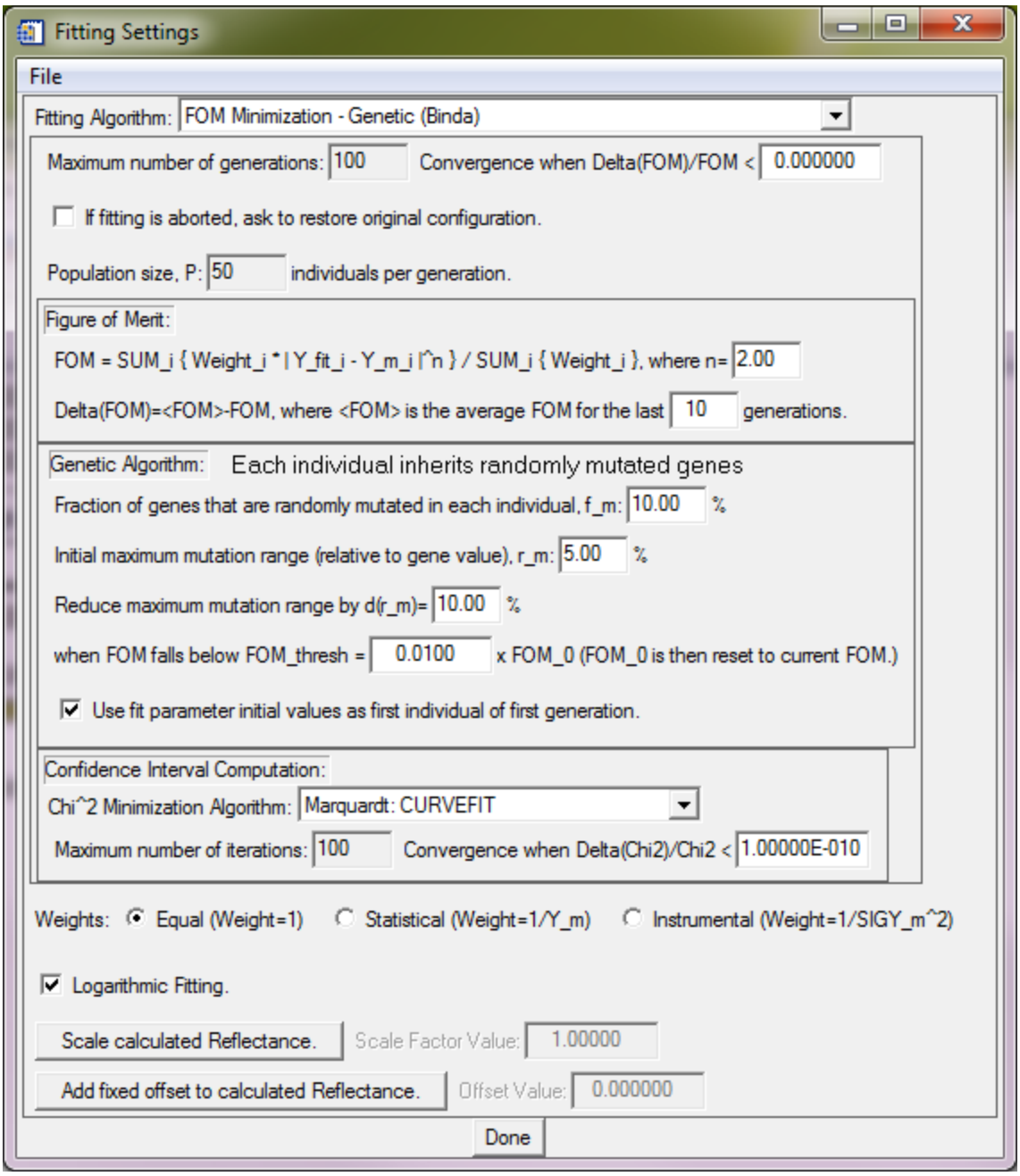
For this fitting example, we'll start with the file '`Mo_Y_double_ML_L=1.54A.imd`' in the '`imd/examples/Fitting`' directory, which contains XRR data measured for a film stack comprising a periodic Mo/Y multilayer having N=31 repetitions, with a second periodic Mo/Y multilayer having N=33 repetitions on top of the first multilayer, and then a final B_4C/Mo capping layer. The measured and calculated reflectance curves, computed using nominal (but erroneous) structure parameters, are shown in the next figure:



Clearly the nominal structure values don't produce a reflectance curve that matches the measured XRR data, and so we've got some fitting to do. As in the file '`Mo_Y_double_ML_L=1.54A_GA_0.imd`', we've selected 14 fitting parameters: the B_4C layer thickness, the top Mo layer thickness, the two multilayer periods (d) and layer thickness ratios (Γ), and all 8 interface widths:



We'll start with the GA algorithm, configured as in the Fitting Settings window shown in the next figure:



After 100 generations, we have a pretty decent fit – see '`Mo_Y_double_ML_L=1.54A_GA_1.imd`'. It's certainly not a perfect fit, but it's probably good enough to extract meaningful layer thicknesses:

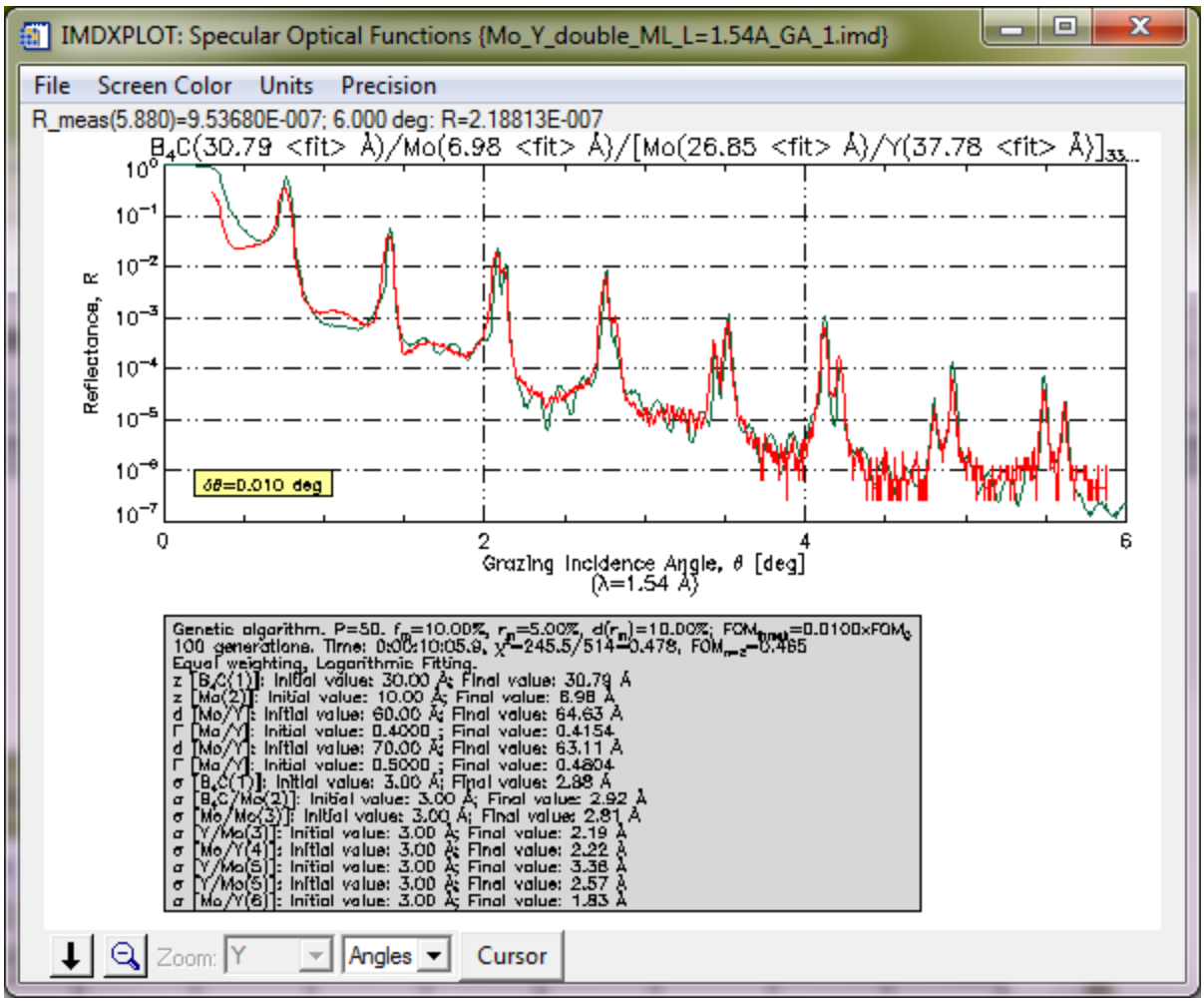
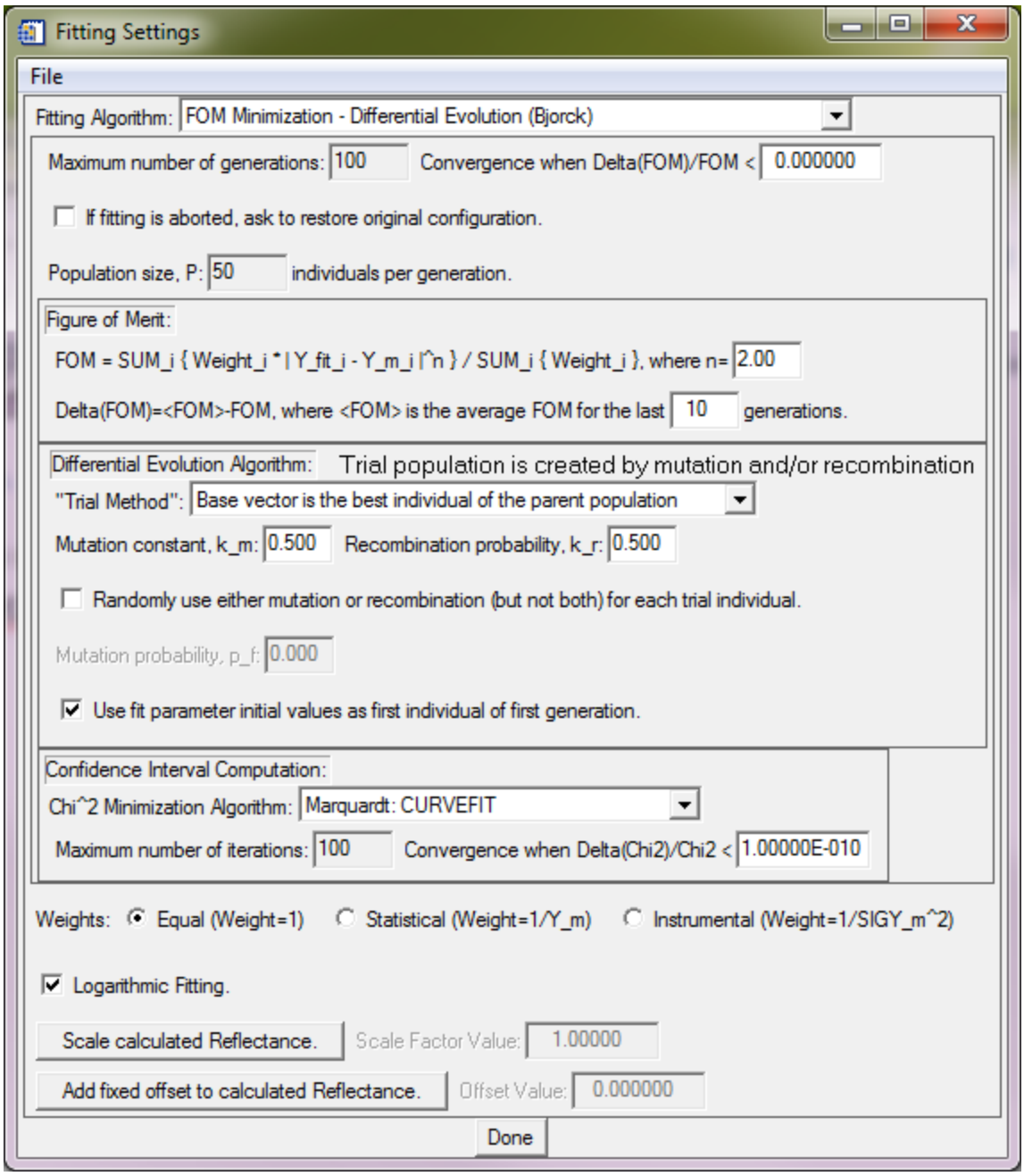


Figure 44. Fitting results using the GA algorithm.

For comparison, let's fit the same data using the DE algorithm. The Fitting Settings window shows the configuration (as in '`Mo_Y_double_ML_L=1.54A_DE_0.imd`'):



In this case the DE algorithm was able to achieve a better FOM after 55 generations than the GA algorithm achieved after 100. Again, not a perfect fit, but nevertheless useful:

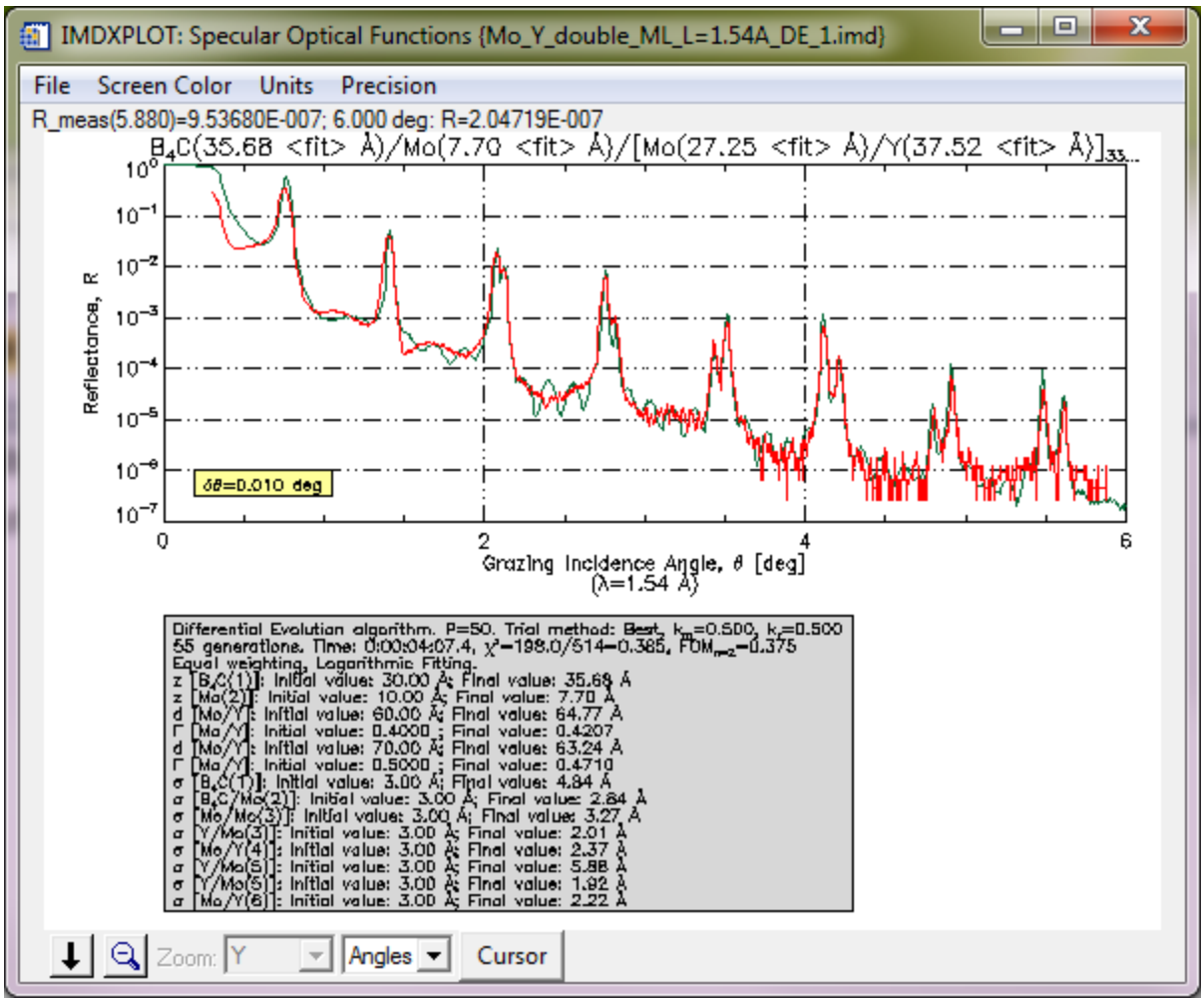


Figure 45. Fitting results using the DE algorithm.

8.7.6 Example: Optimizing an Aperiodic Multilayer to a Triangular Profile

In this section we'll attempt to design an aperiodic Ni/C multilayer (the same one as in §8.5.2) that has a Reflectance-vs-Energy profile that matches the double-triangular Optimization Target Profile shown in Figure 36. In fact this optimization example is very similar to one of the examples presented in Kozhevnikov et al.,[17] specifically the example shown in Figure 11 of that paper, and we'll use this example to demonstrate the fourth method for populating the list of aperiodic thicknesses that we alluded to back in §4.6.

The starting example file is called '`Ni_C_AML_0.imd`', located in the '`imd/examples/Optimization`' directory. The Multilayer window for this aperiodic multilayer is shown in the next figure:

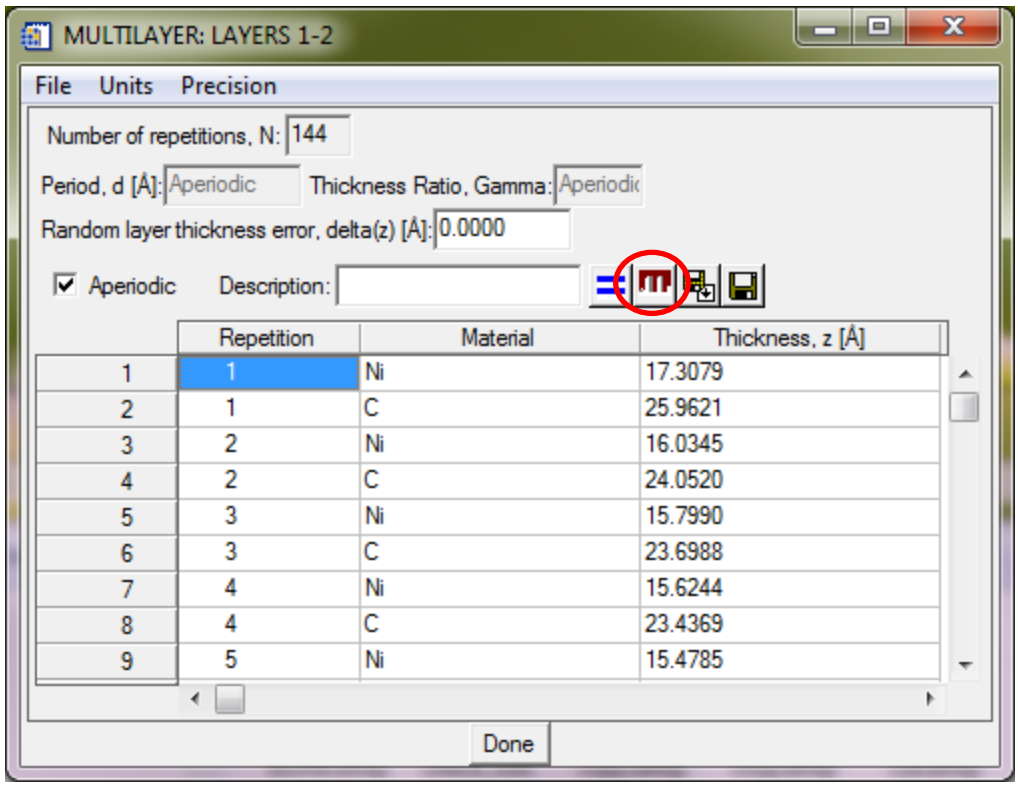


Figure 46. The Multilayer window for an aperiodic multilayer whose thicknesses were determined using the formalism developed by Kozhevnikov et al, described in reference [17], by clicking the button highlighted by the red circle.

As you can see in the figure above, the list of aperiodic layer thickness has been populated already. In this case, we have used the formalism developed by Kozhevnikov et al., described in reference [17], to produce an initial aperiodic thickness list based on the Optimization Target Profile already defined (Figure 36). That is, once the structure, variables and Optimization Target Profile are properly defined, the aperiodic thickness list can be populated using a fourth method:

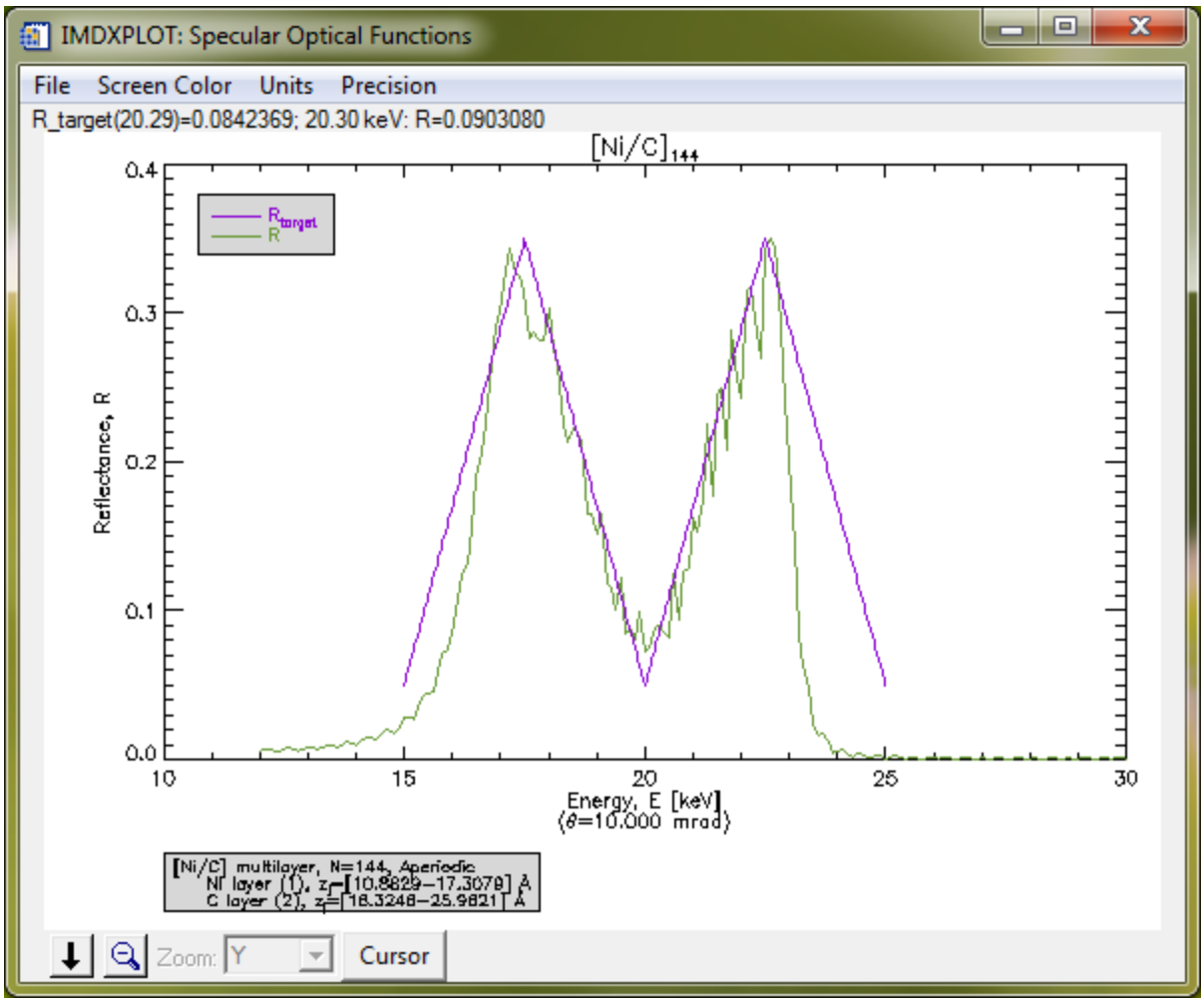
8.7.6.1 Methods for Populating an Aperiodic Thickness List (Part 3/3):

(continued from §4.7**)**

Method 4: The thickness of each layer in the aperiodic multilayer can be computed using the Kozhevnikov formalism of reference [17], by clicking the button with this icon –  – in the Multilayer window, as shown above in Figure 46.[xi]

After using the Kozhevnikov formalism, the calculated reflectance of our Ni/C AML looks like this:

[xi] In case you're wondering: the  icon represents the skyline of the Taj Mahal hotel in Atlantic City, New Jersey, in tribute to Figure 12 of reference [17].



So, with a reasonably good starting point now defined for our optimization of aperiodic layer thicknesses, we'll next add two Optimization parameters – the Ni and C aperiodic layer thicknesses – exactly as we did in §8.5.2. We'll perform the optimization using the Levenberg-Marquardt algorithm. The results of this optimization are contained in the file '`Ni_C_AML_1.imd`': the optimization was aborted after 74 iterations, because the parameter values seem to have stabilized, and the χ^2 value was only improving very slowly. The results of this optimization are displayed in the Optimization Results window shown in the next figure. This window is completely analogous to the Fitting Results window described in §8.7.2.

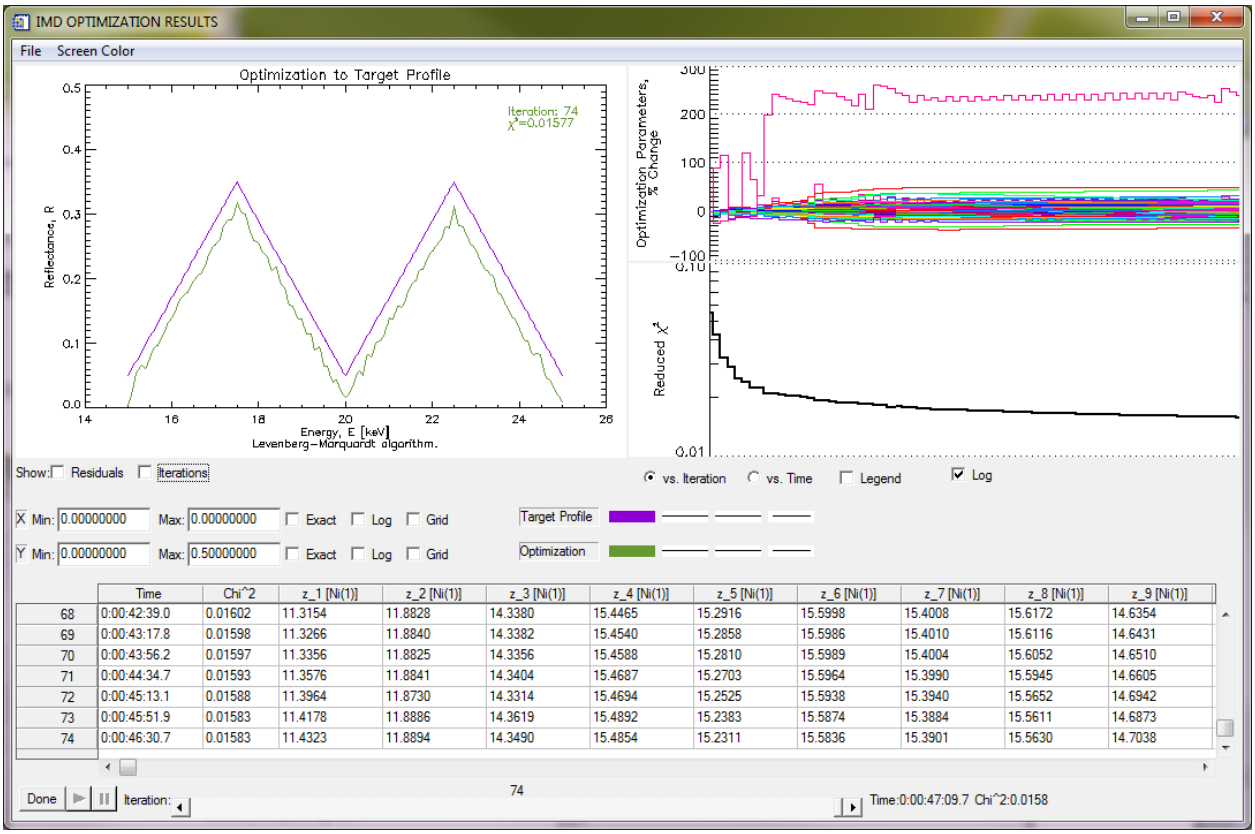
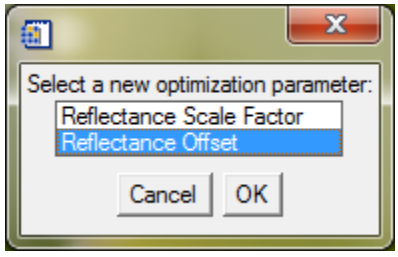
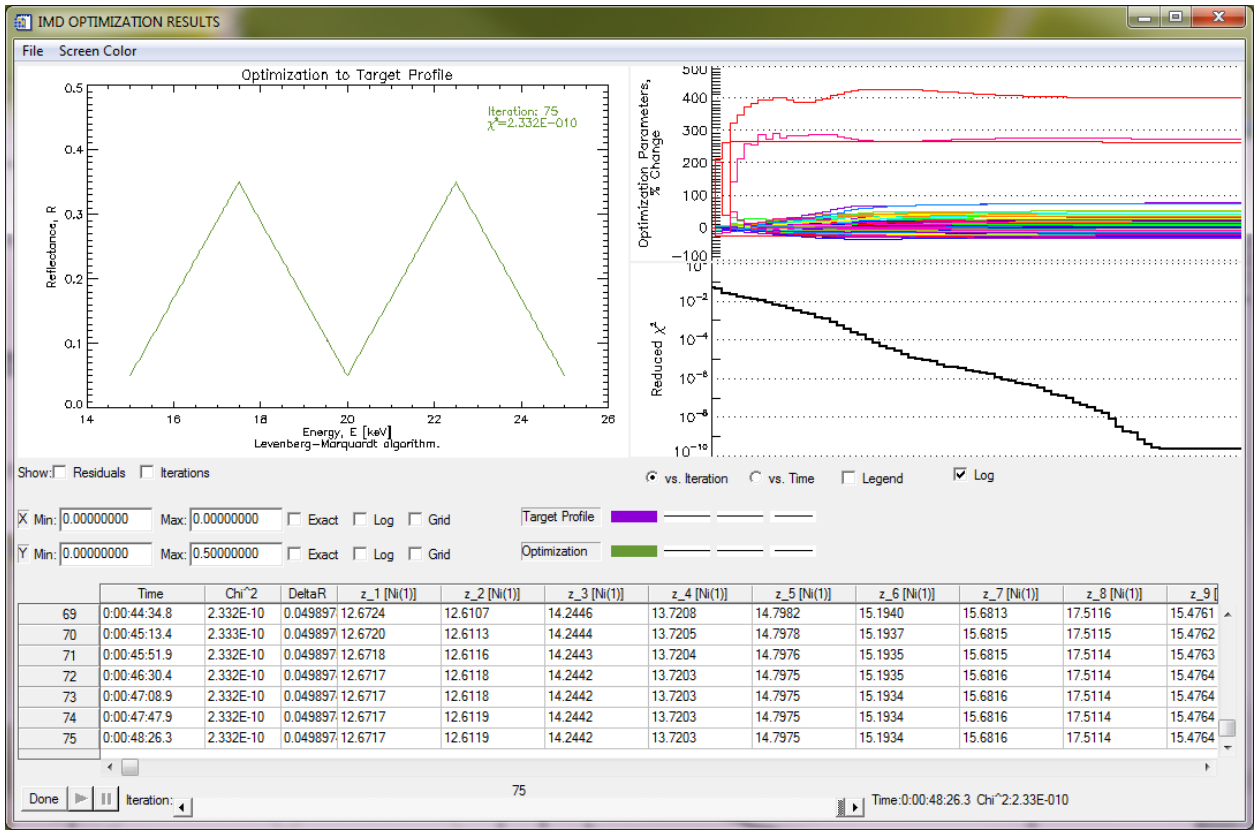


Figure 47. The Optimization Results window.

The resulting reflectance profile shown in Figure 47 is pretty close to the target. But the straight-line portions of the computed reflectance are not perfectly straight – they're a little noisy. In fact we can do better. It turns out that, in retrospect, the target reflectance profile we specified was a little too high, higher than can be achieved using the Ni/C AML as we've defined it. One solution would be too shift the target profile down a bit. Another solution is to add "Reflectance Offset", $\Delta(R)$, as an additional optimization parameter:



With Reflectance Offset specified as a third adjustable parameter, we're able to match the target profile quite well (see '`Ni_C_AML_2.imd`'): after 75 iterations, the reflectance profile is nearly indistinguishable from the target profile, and the χ^2 value is many orders of magnitude smaller than it was after the previous optimization:



	Time	Chi^2	DeltaR	z_1 [Ni(1)]	z_2 [Ni(1)]	z_3 [Ni(1)]	z_4 [Ni(1)]	z_5 [Ni(1)]	z_6 [Ni(1)]	z_7 [Ni(1)]	z_8 [Ni(1)]	z_9 [
69	0:00:44:34.8	2.332E-10	0.049897	12.6724	12.6107	14.2446	13.7208	14.7982	15.1940	15.6813	17.5116	15.4761
70	0:00:45:13.4	2.333E-10	0.049897	12.6720	12.6113	14.2444	13.7205	14.7978	15.1937	15.6815	17.5115	15.4762
71	0:00:45:51.9	2.332E-10	0.049897	12.6718	12.6116	14.2443	13.7204	14.7976	15.1935	15.6815	17.5114	15.4763
72	0:00:46:30.4	2.332E-10	0.049897	12.6717	12.6118	14.2442	13.7203	14.7975	15.1935	15.6816	17.5114	15.4764
73	0:00:47:08.9	2.332E-10	0.049897	12.6717	12.6118	14.2442	13.7203	14.7975	15.1934	15.6816	17.5114	15.4764
74	0:00:47:47.9	2.332E-10	0.049897	12.6717	12.6119	14.2442	13.7203	14.7975	15.1934	15.6816	17.5114	15.4764
75	0:00:48:26.3	2.332E-10	0.049897	12.6717	12.6119	14.2442	13.7203	14.7975	15.1934	15.6816	17.5114	15.4764

It's important to note, however, that the best-fit Reflectance Offset value (Δ(R)=0.049897) has been included in the computed reflectance shown above in the Optimization Results window, and also in the IMDXPLOT window that appears after the optimization is completed:

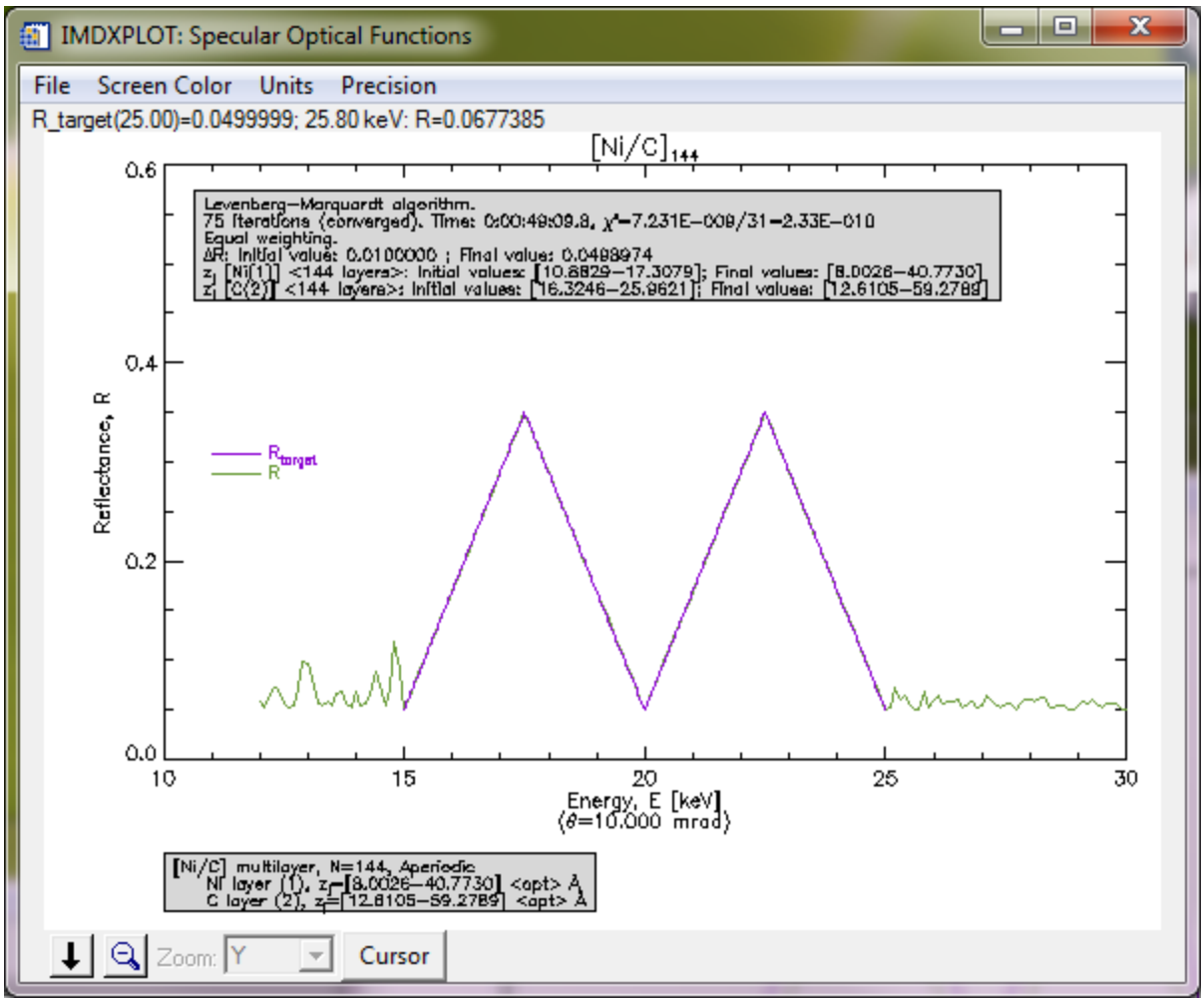


Figure 48. IMDXPLOT window showing the results of the optimization. A value of Δ(R)=0.0498974 is included in the computed reflectance shown here, because Reflectance Offset was included as an optimization parameter.

In fact the actual theoretical reflectance resulting from the best-fit layer thicknesses will be lower without the offset included. After completing the optimization, selecting "Calculate→Specular Optical Functions/Fields" from the IMD menu bar will display the actual reflectance, without the Δ(R) offset:

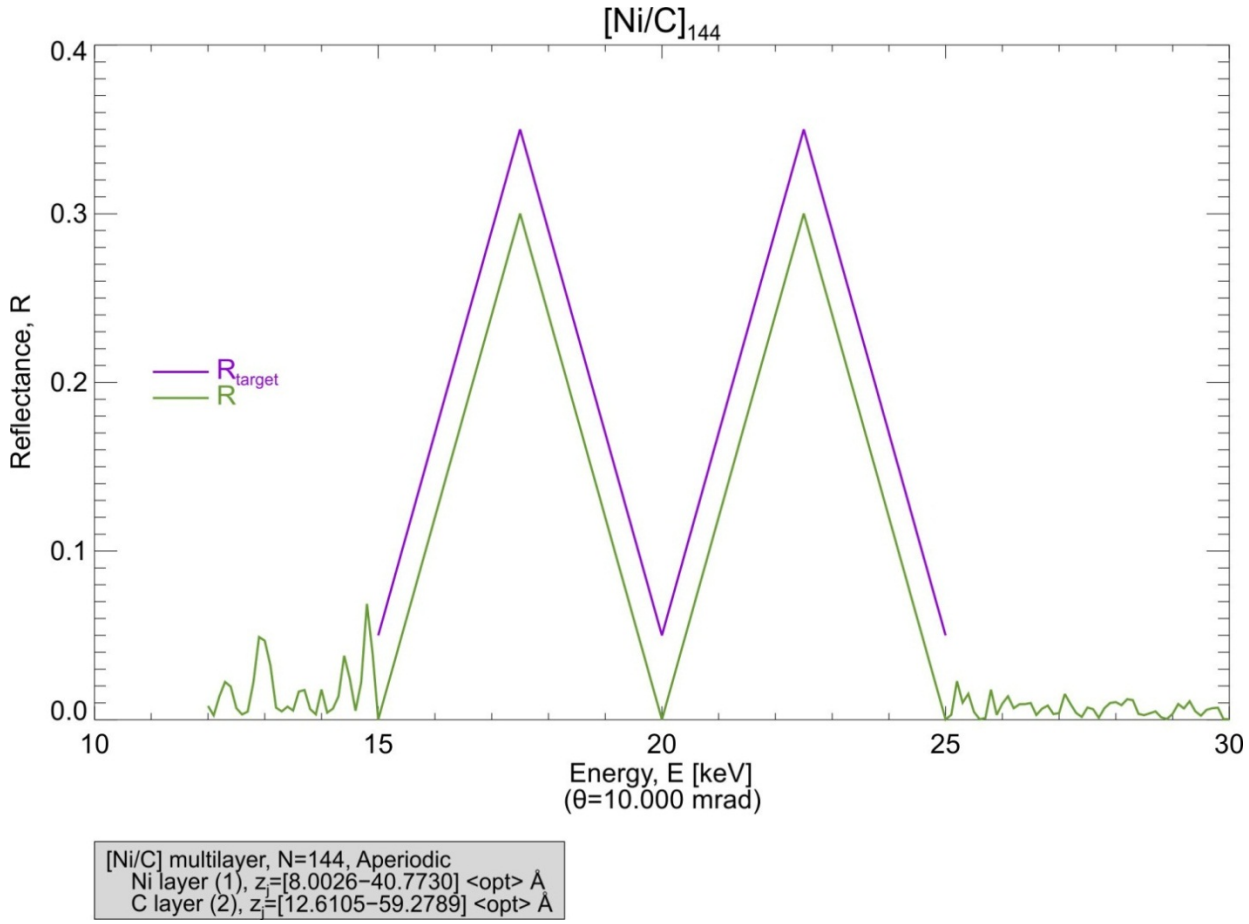
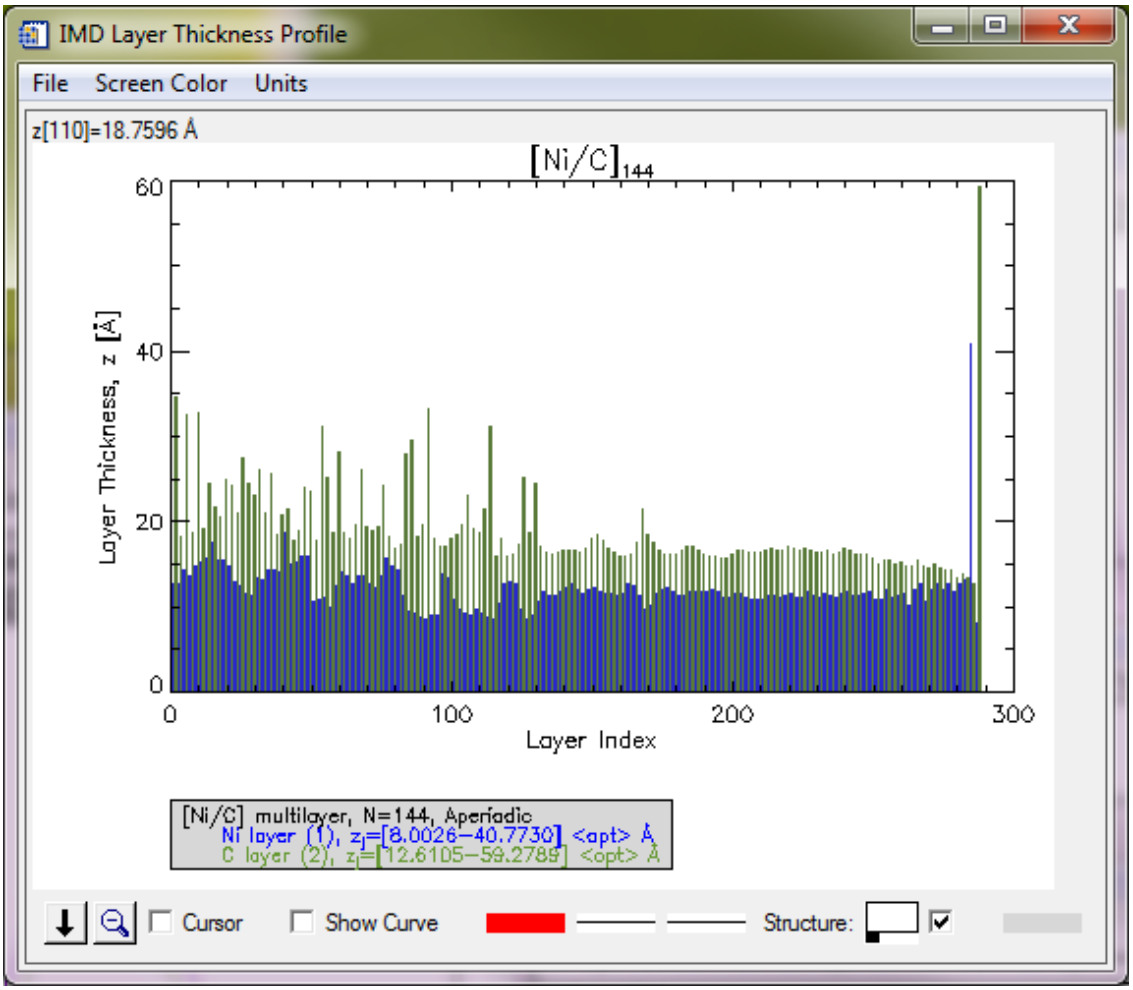


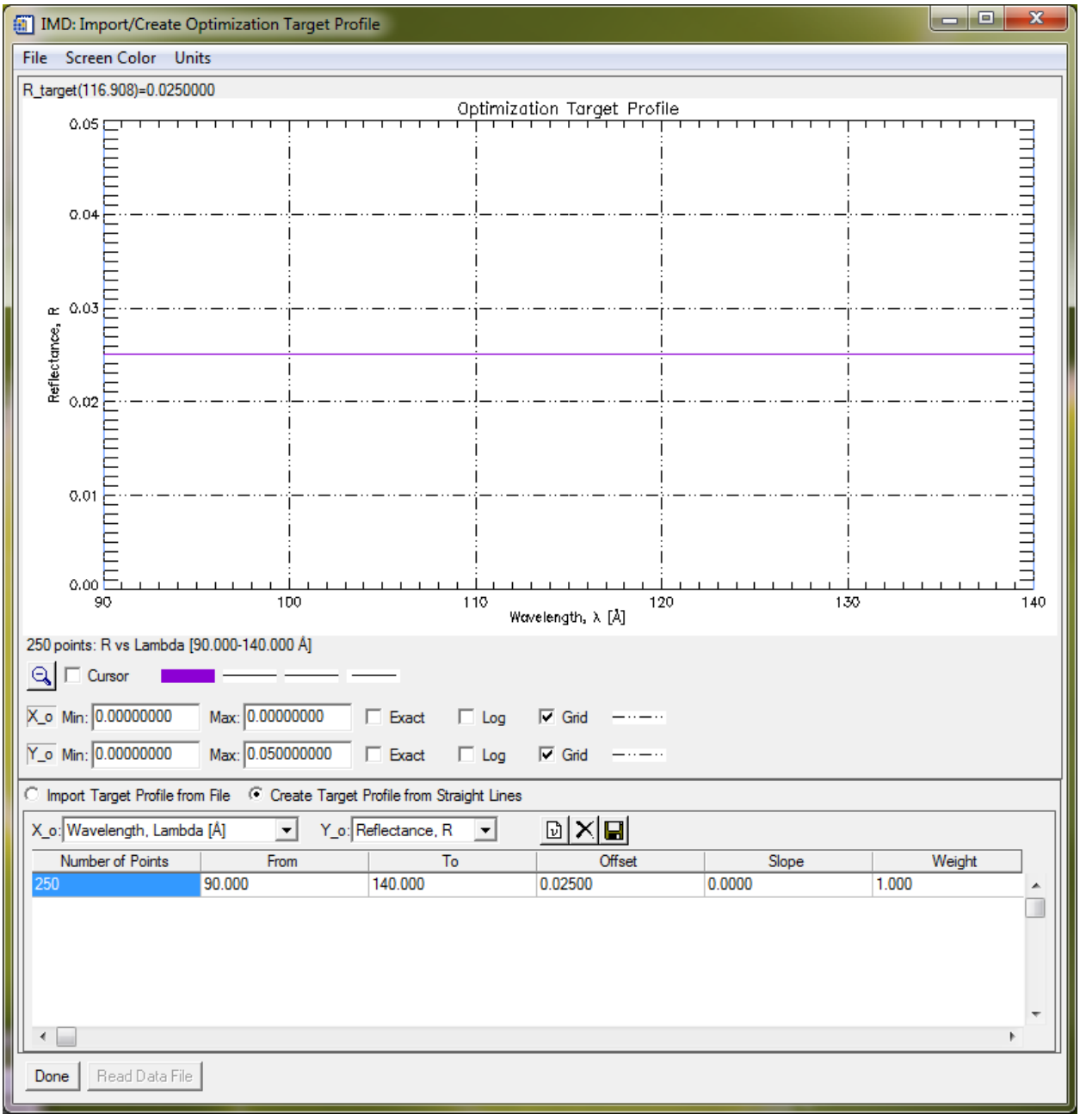
Figure 49. Plot of the *calculated* reflectance of the optimized Ni/C multilayer, using the aperiodic thicknesses determined from optimization. Unlike the plot from the previous figure, the calculated reflectance shown here (computed using "Calculate→Specular Optical Functions/Fields") does not include the Reflectance Offset that was included as an optimization parameter.

The aperiodic layer thicknesses determined from this optimization can be displayed in the Layer Thickness Profile window (§4.10), as shown in the next figure:

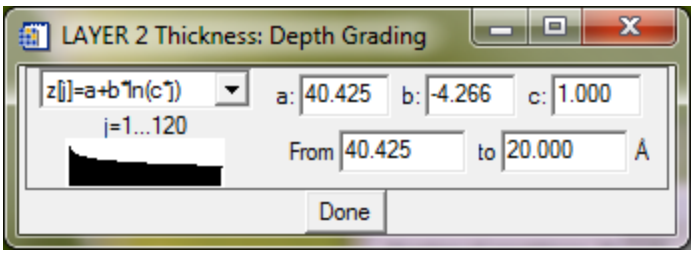


8.7.7 Example: Optimizing an Aperiodic Multilayer for Flat Response at Normal Incidence

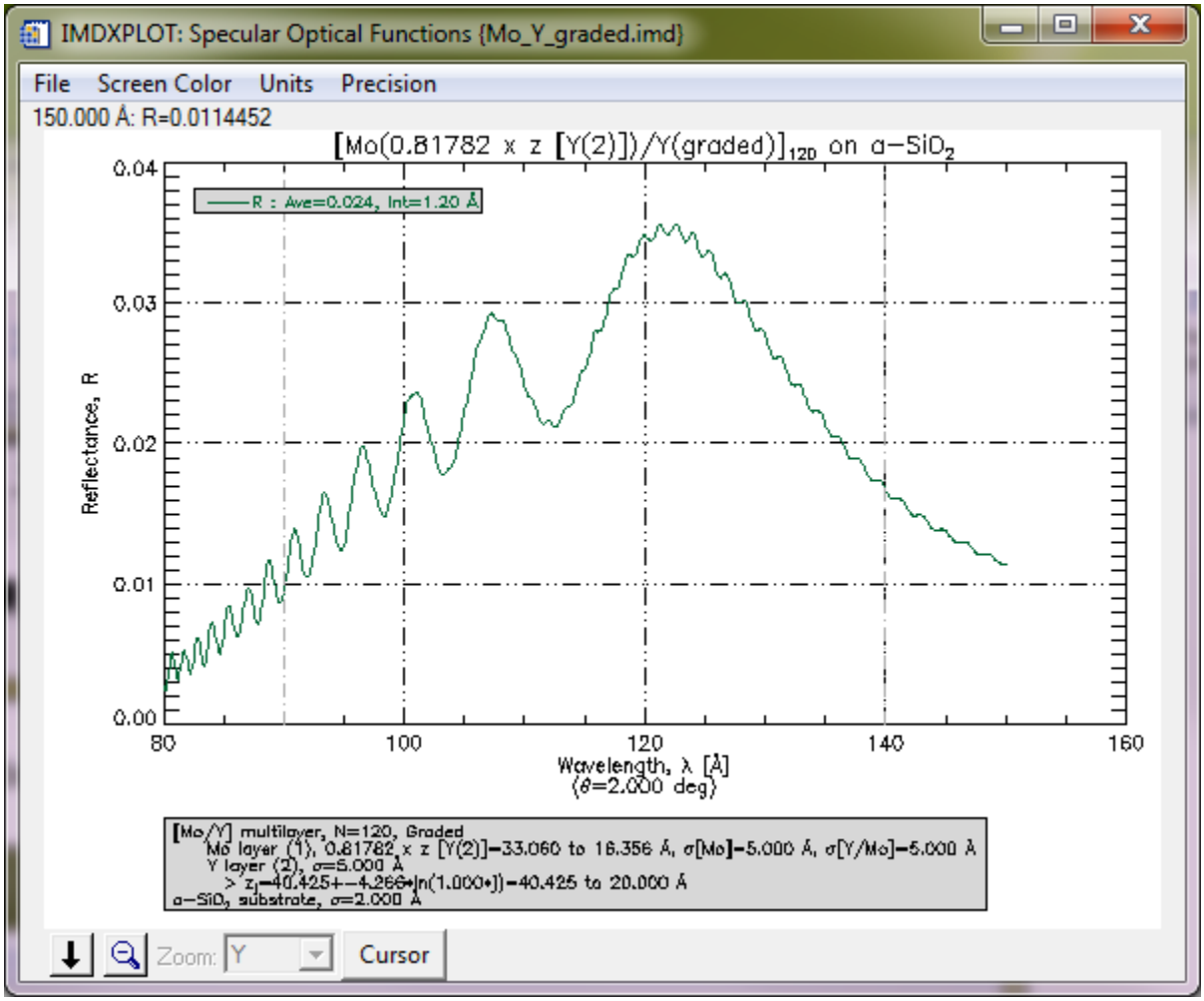
Aperiodic multilayers can be designed to provide high reflectance over a wide range of angles or wavelengths. The next optimization example in this chapter will focus on the optimization of an aperiodic Mo/Y multilayer designed for flat reflectance at normal incidence, over the wavelength range λ=90 – 140 Å. The optimization target profile we'll use is just one straight line segment over that whole wavelength range, and looks like this:



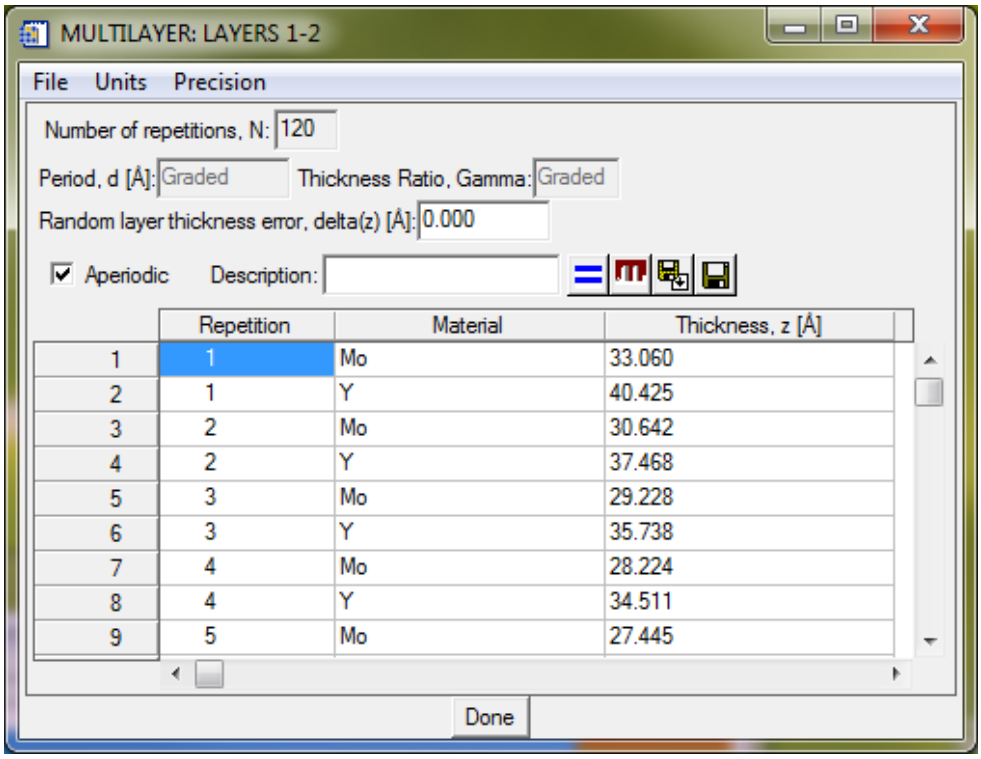
In the file '`imd/examples/Optimization/Mo_Y_graded.imd`', we've created a multilayer containing 120 repetitions of Mo/Y bilayers, with σ=5 Å interface widths. As a starting point for this optimization, we've transformed an initially periodic Mo/Y multilayer into a depth-graded multilayer as follows: we've first added a Coupled Parameter (§4.12) so that the Mo layer thickness is equal to 0.81782 times that of the Y layer thickness (i.e., Γ=0.45); we then specify for the Y layer logarithmic depth-grading like this:



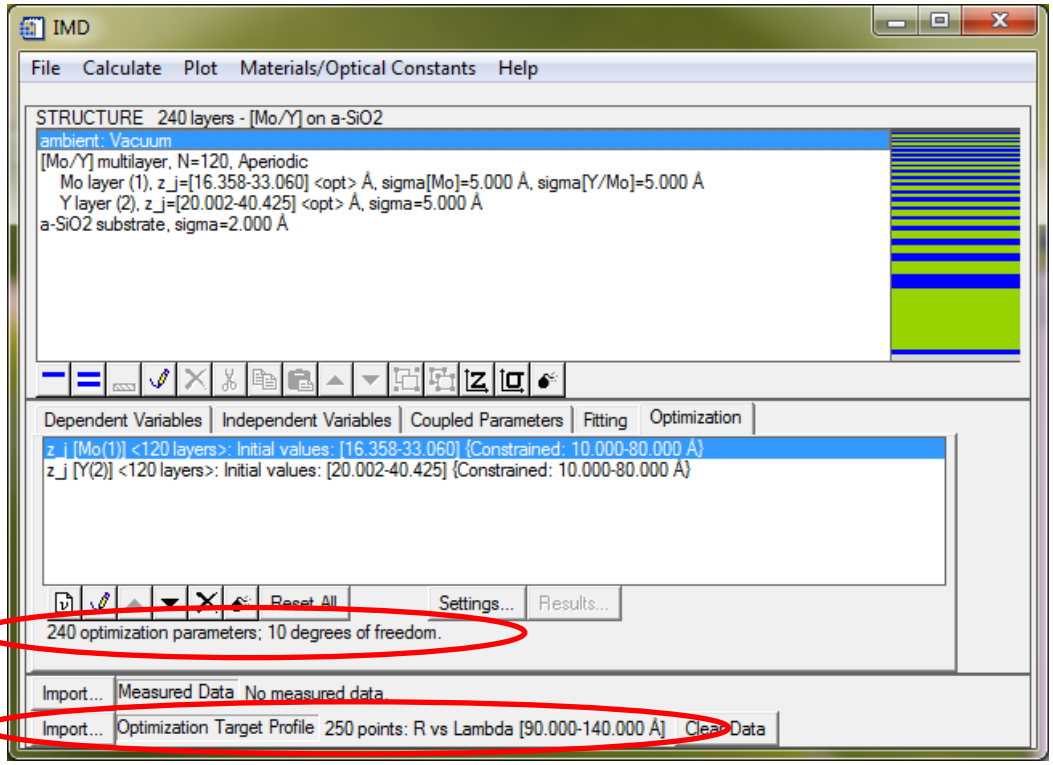
The reflectance vs. wavelength at 2° incidence for this structure looks like this:



The next step is to convert this depth-graded multilayer to an aperiodic multilayer. We open the Multilayer window, check the "Aperiodic" box, and then click the = button on that window to populate the aperiodic layer thickness list with our logarithmically-graded thickness values just generated (as explained in §4.6):

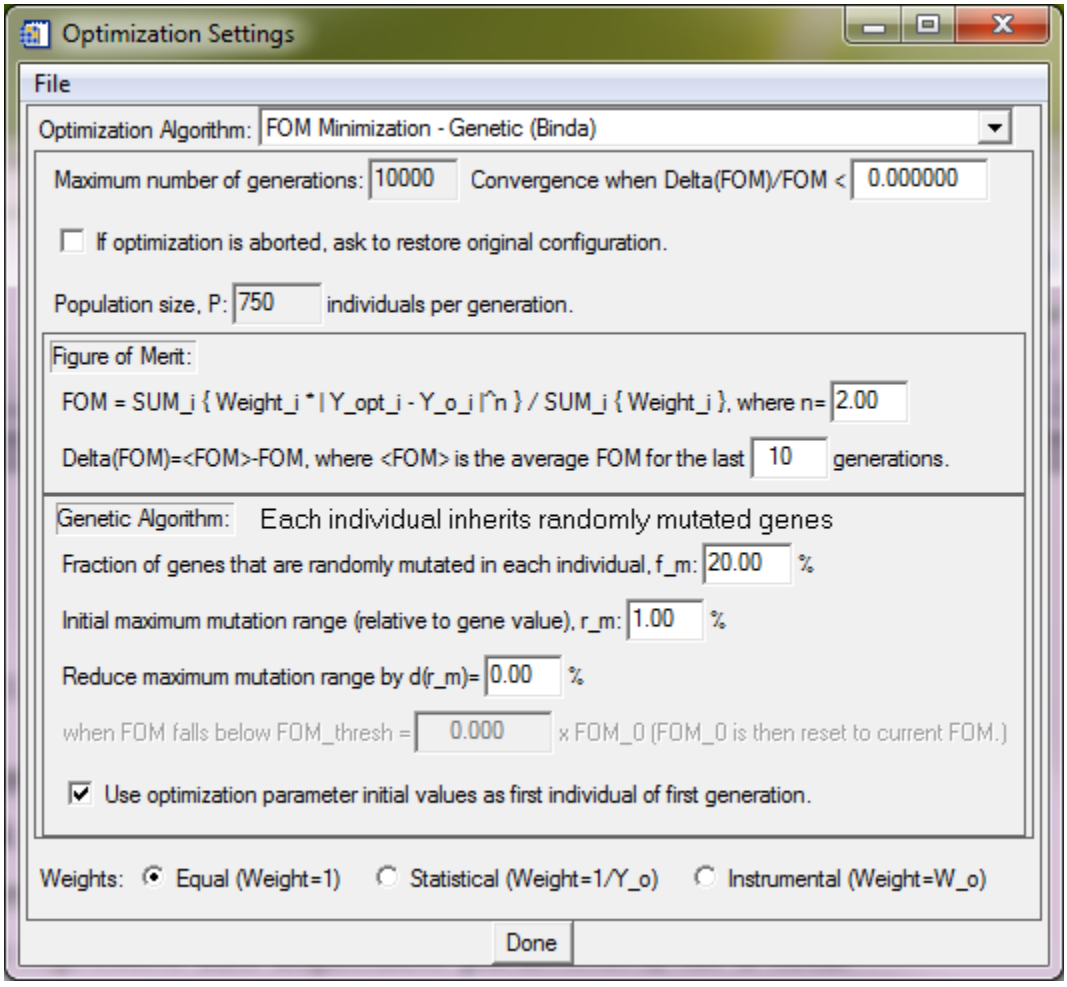


We then remove the coupled parameter linking Mo thickness to Y thickness, and add the Mo and Y aperiodic layer thicknesses as our two optimization parameters:

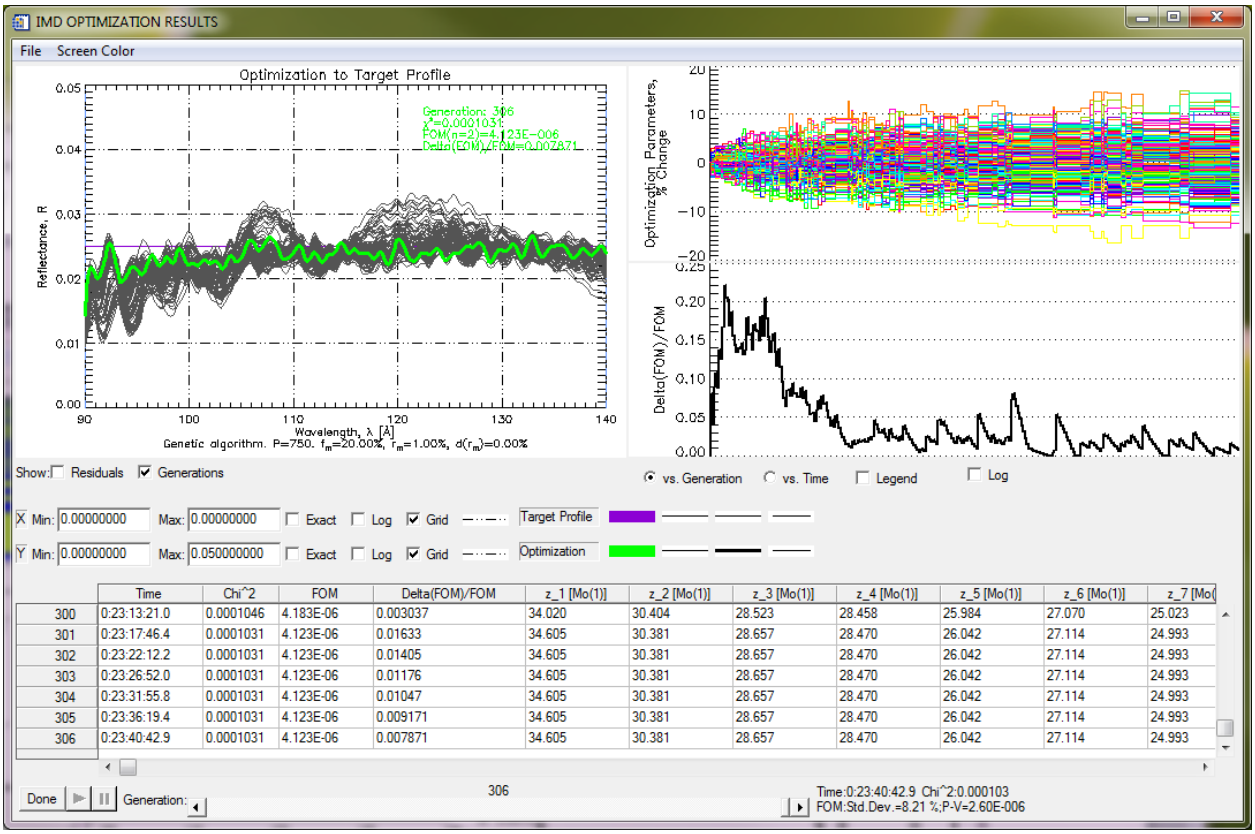


At this point our structure still comprises Mo and Y layer thicknesses that follow the same logarithmic distribution in depth as before. However the film stack is now defined as an aperiodic structure, which means that we can treat each individual layer thickness as an adjustable parameter for optimization. Indeed, the Mo and Y aperiodic layer thicknesses each represent 120 adjustable parameters, for a total of 240 adjustable parameters. Recall that our optimization target profile shown above is just a straight line, and in principle it can be specified with just two points. However, in order to perform the optimization, we need at least 1 degree of freedom; we've therefore created the optimization target profile using 250 points, so that we have 250 data points – 240 optimization parameters = 10 degrees of freedom.

We'll start with the GA algorithm for the optimization, using a population size of 750 individuals, with f_m=20% and r_m=1%, as in the file '`Mo_Y_opt_GA_0.imd`':



After starting the optimization, we aborted it after 306 generations (over 23 hours on my computer). Here's how the optimization looks at that point (see '`Mo_Y_AML_GA_1.imd`'):



We've got a reasonably good reflectance profile, close to the target. But the reflectance vs. wavelength curve is not really as flat as we'd like. So, now let's switch to the Levenberg-Marquardt (LM) algorithm, for "clean-up": before starting the new optimization, we'll click the "Reset All" button on the Optimization Tab of the main IMD window – this will set the best-fit layer thicknesses we determined using the GA algorithm as the Initial values for the new optimization. After 15 iterations using the Levenberg-Marquardt algorithm (see '`Mo_Y_AML_LM_2.imd`'), we've got a flat reflectance curve that closely matches the target profile we were aiming for:

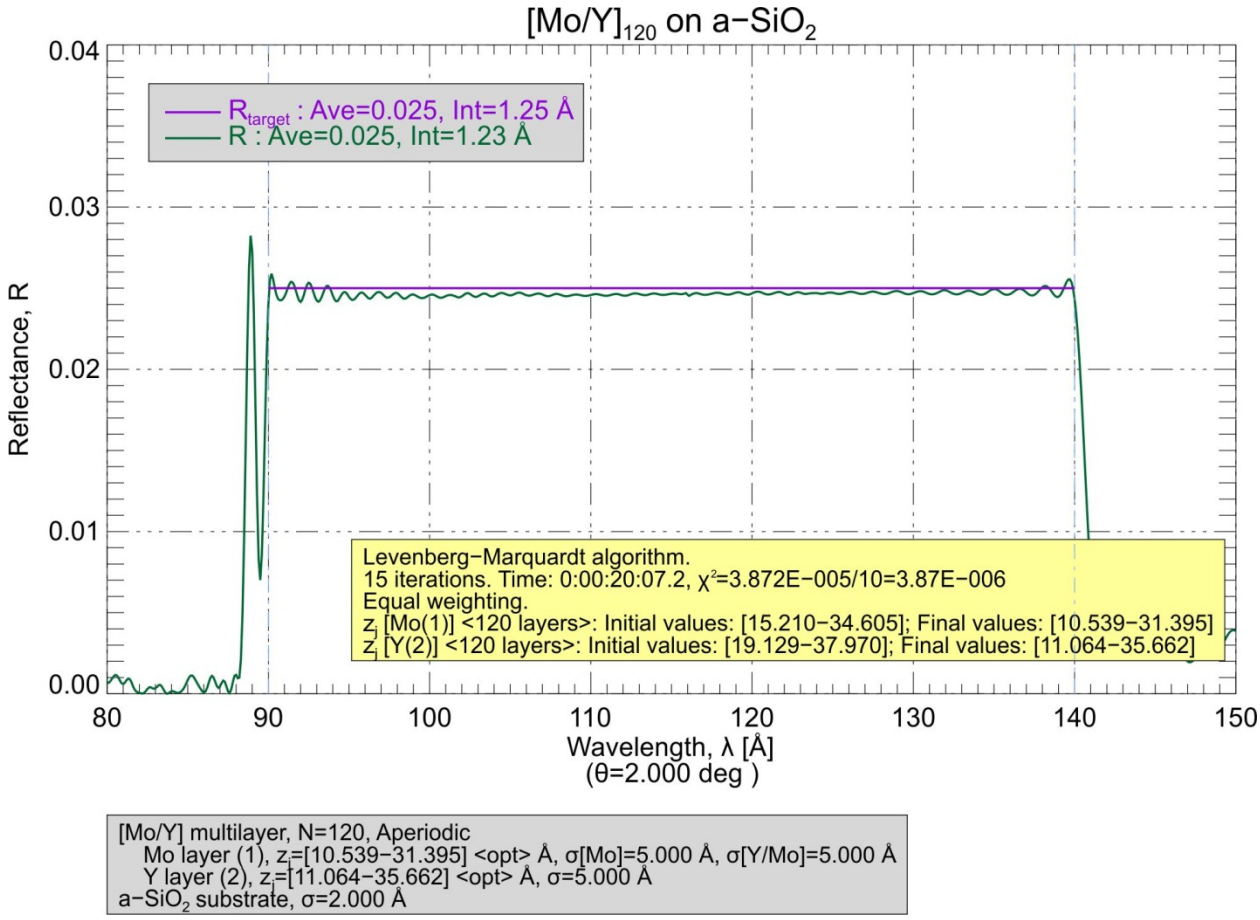


Figure 50. The final result of layer-thickness optimization, first using the GA algorithm, then using the LM algorithm. The average and integrated reflectance values, computed over a Region-of-Interest spanning the range λ=90–140 Å, are indicated in the legend for both the target and actual reflectance profile.

8.8 Parameter Estimation: Confidence Intervals

We presented in §8.7.5 two fits to the same measured data, one fit using the GA algorithm, one using the DE algorithm. You may have noticed that the final parameter values determined using the GA algorithm shown in Figure 44 are somewhat different than those determined using the DE algorithm shown in Figure 45. Actually, the variation in best-fit parameter values is not due to using different algorithms; even multiple fitting runs using the same algorithm will (and should) generally yield slightly different best-fit parameter values.

The variation in best-fit parameter values estimated from successive fittings in those two examples of §8.7.5 underscores an important point with regard to parameter estimation, which was well-stated by Lampton:[25]

> *In parameter estimation, it is the range of parameter values to which a theory is restricted that is the useful result of an experiment. The discrete best-fitting values of the parameters are*

essentially statistical artifacts subject to a variety of correlated random errors originating in the counting statistics of the original data. If an experiment could be repeated without systematic changes, best-fitting parameter values would differ, while their properly derived allowed ranges will overlap.

In this section we'll learn how to compute the "properly derived allowed ranges" associated with best-fit parameters determined from curve-fitting, using the methods described in references [25] and [26]. In IMD these allowed ranges are called multi-dimensional "confidence intervals".

When using the χ^2 test of fit [equation (1)], it can be shown that the minimum value of the S statistic, S_{min}, associated with the best-fit parameter values is distributed as the χ^2 probability function with (N_m-p) degrees of freedom:

(3) $$S_{\min} = \chi^2_{N_m - p}(\alpha)$$

where α is the significance of fit, and N_m is the number of measured data points. That is, if we find, for example, our best fit has an $S_{\min} = \chi^2_{N_m - p}(0.68)$, then we can conclude that there is 68% probability that the model correctly describes the data when the p best-fit parameter values are used. The confidence region, with significance α', is then defined as the p-dimensional region of parameter space for which the value of S is less than or equal to some value S_L, where

(4) $$S_L = S_{\min} + \Delta S(\alpha')$$

and $\Delta S(\alpha')$ is equal to the value of the χ^2 probability function with p degrees of freedom and significance α'; the confidence region so defined would enclose the true values of the p parameters in (1-α') of all experiments.

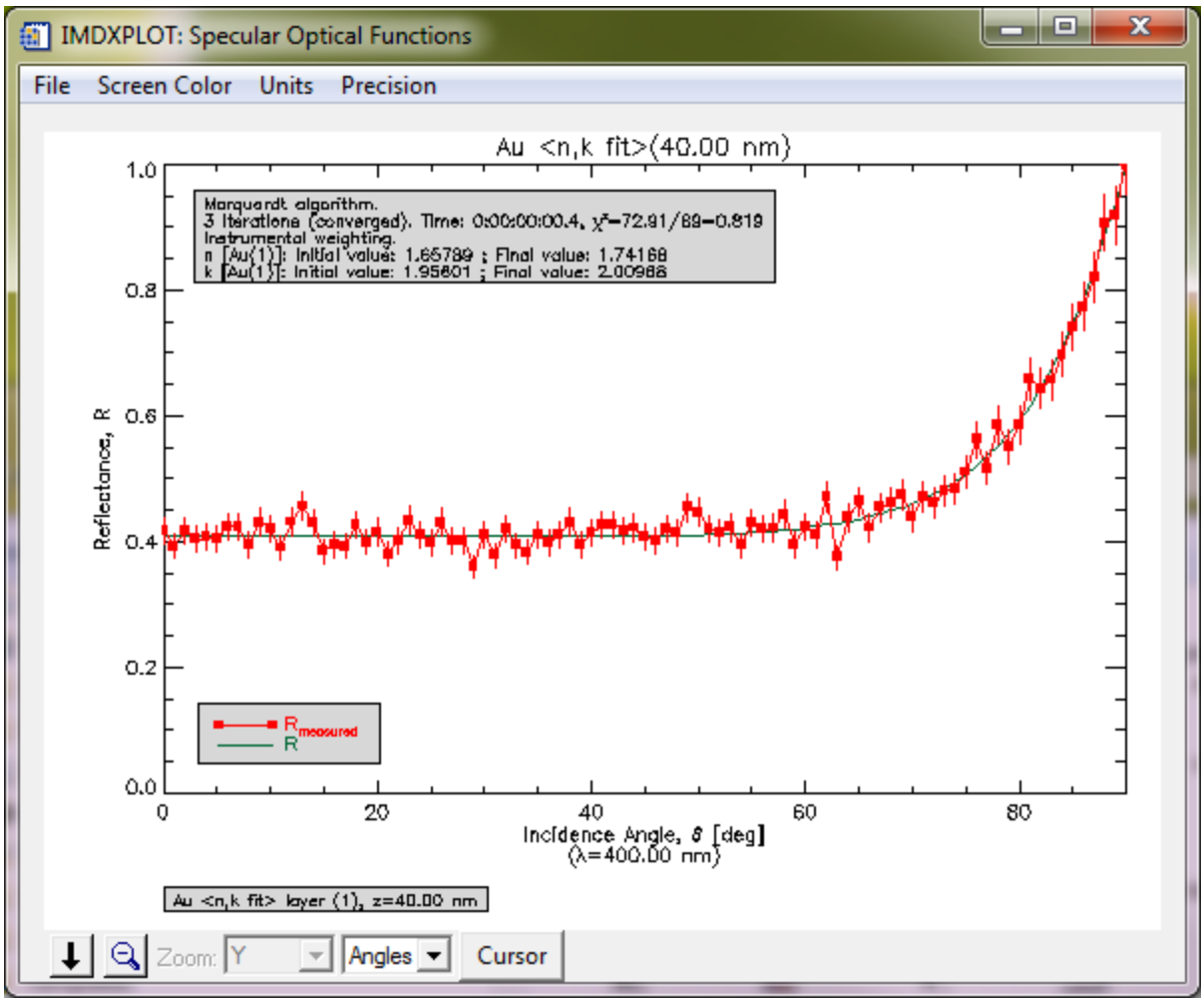
In IMD, a multi-dimensional confidence region can be determined for up to eight fit parameters simultaneously; IMDXPLOT allows you to interactively view 1D or 2D slices of the multi-dimensional confidence region that you compute, i.e., as 2D or 3D graphics. IMD uses a grid-search algorithm to compute confidence regions, and you must specify the extent and resolution of the grid along each the axis of each parameter you specify. At each point on the grid that you define, the value of the statistic S is computed, using one of two methods, depending on the dimensionality of the confidence region being computed relative to the number of adjustable parameters p that have been used for to compute the fit. That is, suppose that the best-fit parameters have been determined for a fit where we've used p adjustable (fit) parameters. We're now interested in computing the confidence region associated with some subset of parameters q, where q≤p. When q=p, so that we're computing a confidence region associated with all the fit parameters, then the value of S at each point on the grid is just computed directly using equation (1). But when q is less than p, so that we're only computing the confidence region for a subset of the adjustable parameters used for the fit, then the value of S at each point on the grid is determined by curve-fitting (using either the Marquardt or the Levenberg-Marquardt algorithm), but with only (p-q) adjustable parameters, i.e., a quantity q of the original p parameters are now held

fixed (to their best-fit values) when the fit is computed at every point on the grid. In that case, the correct value of $\Delta S(\alpha')$ to be used in equation (4) is equal to the χ^2 probability function with q (not p) degrees of freedom and significance α'. The latter option is useful if, for example, your fit parameters comprise a mix of "interesting" and "uninteresting" parameters, and you only want to determine the allowed range on the "interesting" parameters. We'll illustrate how to compute confidence intervals for the first case (p=q) using an example in the next section.

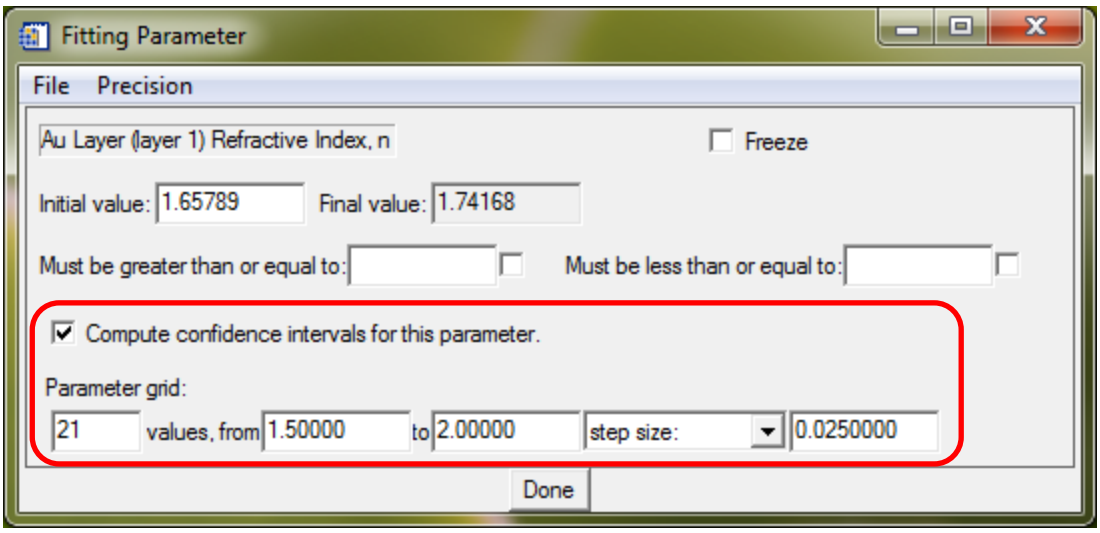
As mentioned above, confidence intervals are computed using either the Marquardt or Levenberg-Marquardt gradient-expansion algorithms. If you have performed a fit using a genetic algorithm, the choice of algorithm used to compute confidence intervals (i.e., Marquardt or Levenberg-Marquardt) is made using the drop-list on the Fitting Settings window, is shown in the figures in §8.6.4 and §8.6.5.

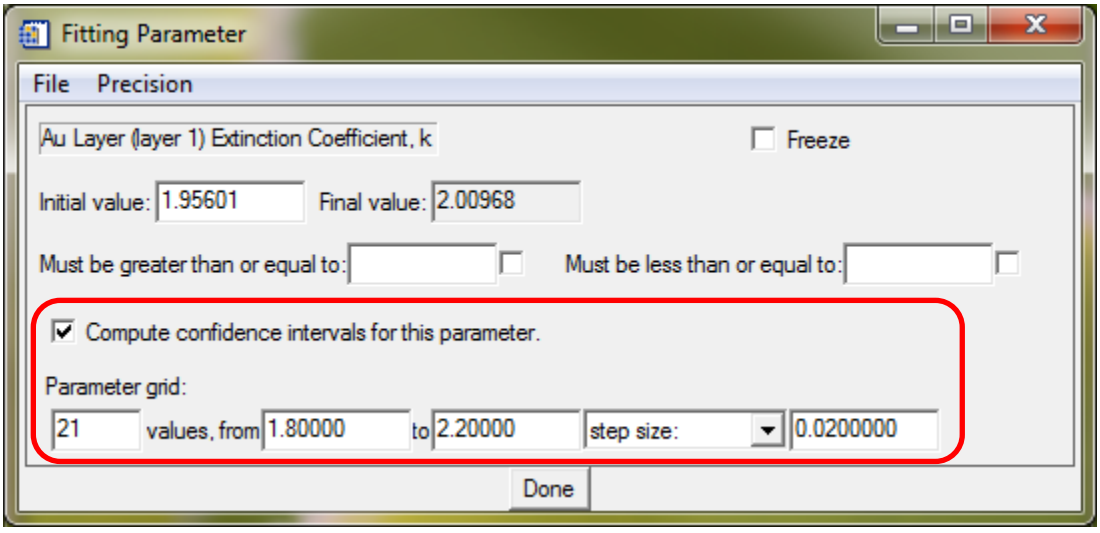
8.8.1 Example: Determining Optical Constants from R-θ Measurements

The example file '`Au_n_k_fit.imd`' in the directory '`imd/examples/Confidence Intervals`' contains the result of a simple fit where we've derived optical constants (n,k) at a fixed wavelength (400 nm) from (fake) measured Reflectance vs. Incidence Angle data for a gold film. The best-fit values of n and k derived from the fit are n=1.74168 and k=2.00968, as shown here:



We'll now compute the confidence region associated with both adjustable parameters, n and k. (So p=q for this example.) We start by specifying the extent and resolution of the parameter space over which we'll compute the S statistic. These values are entered in the Fitting Parameters windows associated with each of the two fit parameters. After opening these windows (from the Fitting tab on the main IMD window), we check the boxes labeled "Compute confidence intervals for this parameter."; we specify 21 grid points for each parameter, with n values ranging from n=1.5 to n=2.0, and k values ranging from k=1.8 to k=2.2. (So the best-fit n,k values we found are roughly in the center of the two-dimensional grid we've defined.) The two Fitting Parameters windows so configured are shown here:





Next, we select from the IMD menu bar "Calculate→Confidence Intervals". The program computes the value of S, using equation (1), at each of the 21x21 points on the grid we've specified.

8.8.1.1 IMDXPLOT: Confidence Intervals

When the confidence interval computation is finished, a new IMDXPLOT window displays the results. In this case (see 'Au_n_k_fit_CI.imd' in 'imd/examples/Confidence Intervals') the results are shown as a contour plot of the χ^2 statistic S as a function of both n and k:

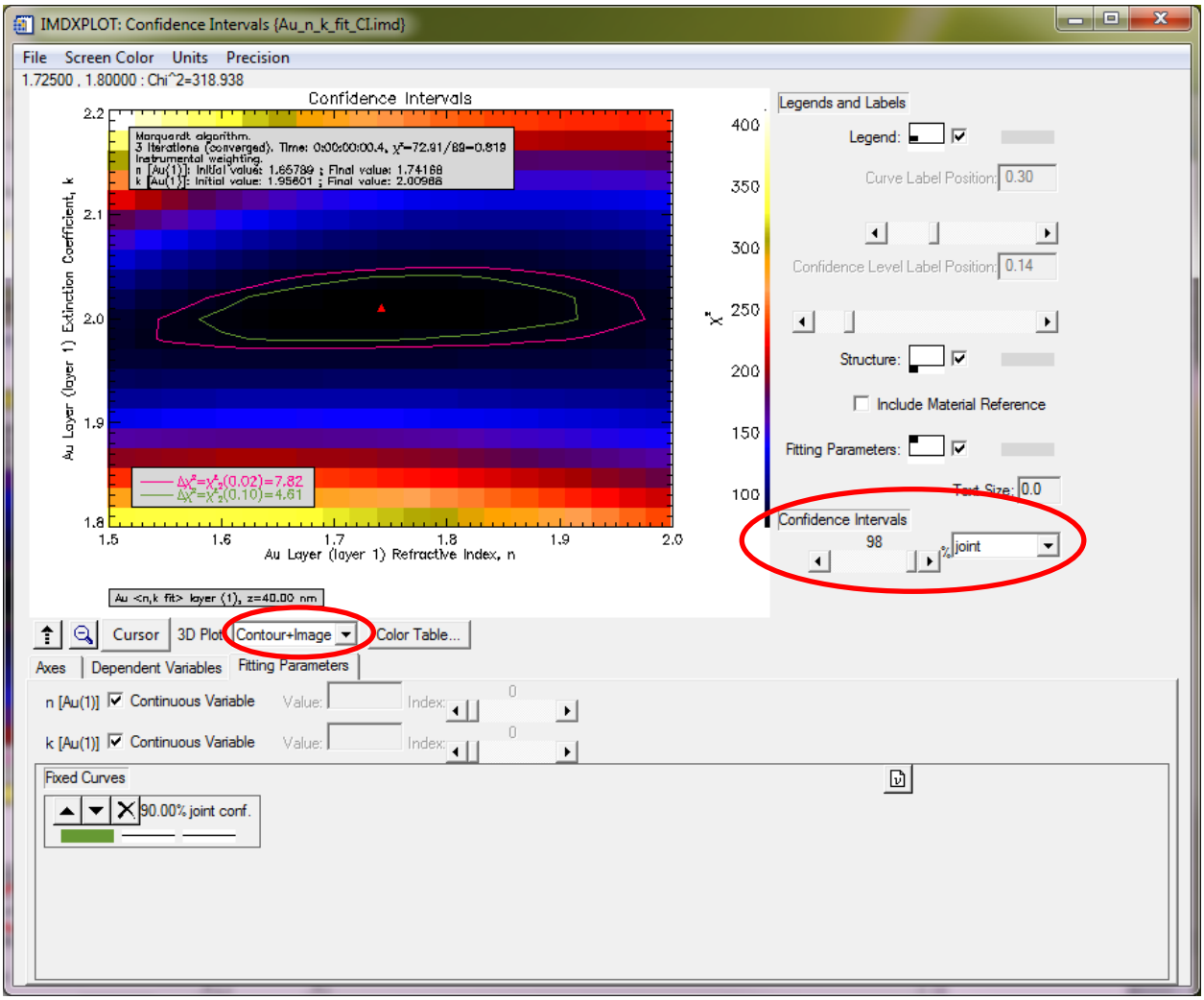
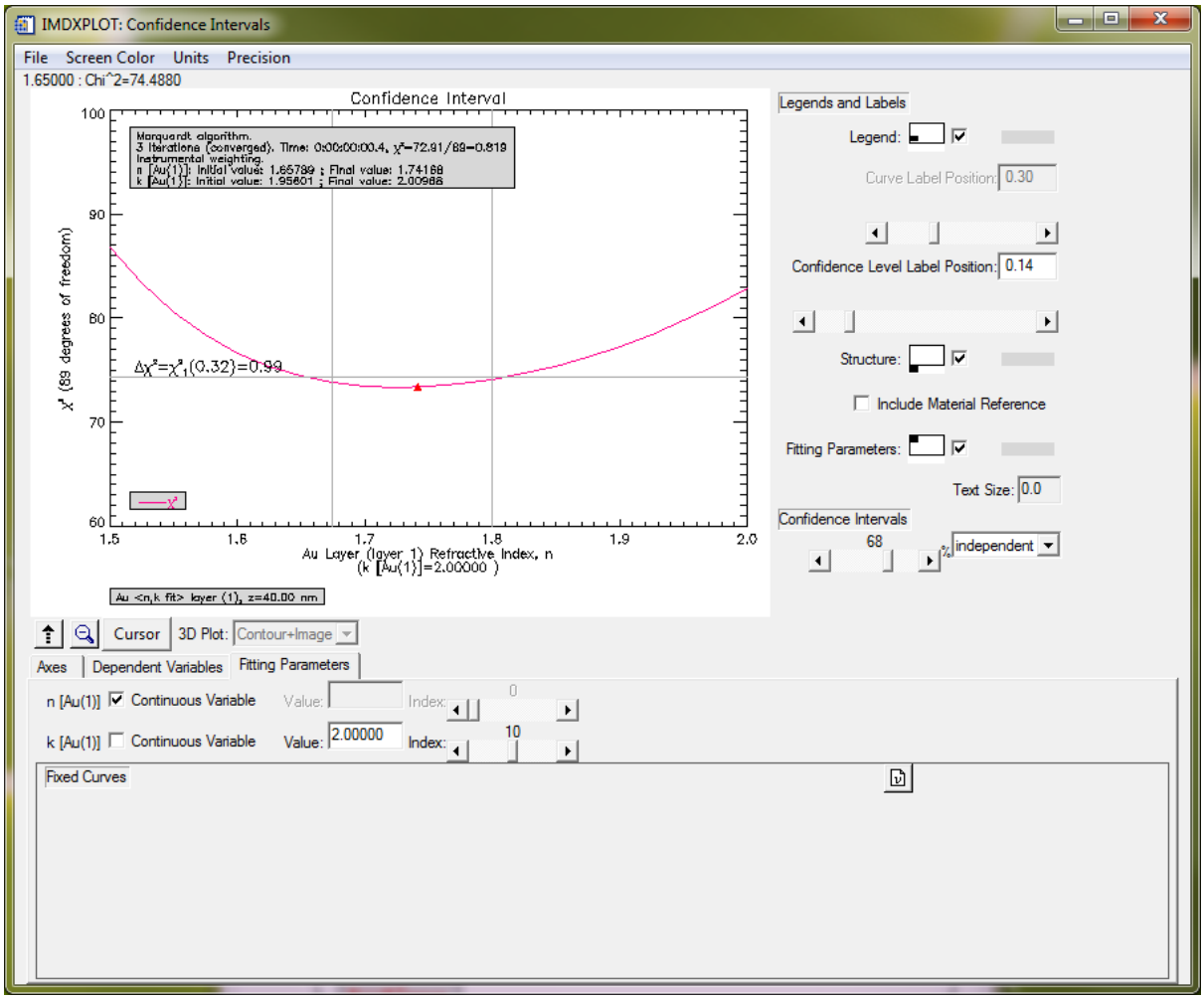


Figure 51. IMDXPLOT window showing confidence intervals.

We've configured the IMDXPLOT window shown in Figure 51 to display two confidence intervals (using Fixed Curves, as described in §6.4), corresponding to the 90% (α=0.10) and 98% (α=0.02) joint confidence intervals for n and k: that is, the two solid contour lines drawn on the plot (Figure 51) in olive and pink enclose the regions in parameter space where the value of S is less than the minimum value determined from the fit (72.91, in this case) – which is indicated on the plot as a red triangular point – plus an amount $\Delta\chi^2$= 4.61 or 7.82, respectively. The meaning of these *joint* confidence contours is as follows: there is a 90% probability that the true values of n and k *both* lie within the olive contour, and a 98% probability that the true values *both* lie within the (bigger) pink contour. You can also display independent (rather than joint) confidence intervals, using the drop-list on the right side of the IMDXPLOT window, which is highlighted above. Please see reference [25] for more details about joint vs. independent confidence intervals.

It's also possible to display 1D slices of parameter space, rather than 2D slices as in the previous figure, by plotting the S statistic as a function of just one of the adjustable (fit) parameters, at discrete values of all the other adjustable parameters used to compute the confidence interval. The selection of fixed parameters is made on the 'Fitting Parameters' tab of IMDXPLOT, in a manner analogous to the way in which Independent Variables are selected for display of optical functions in IMDXPLOT: continuous variables are selected using the check-boxes, and discrete values are selected using the text-boxes and sliders, as illustrated in the next figure.



In the confidence interval plot shown above, we're viewing the χ^2 statistic S computed as a function of the index of refraction, n, with the extinction coefficient fixed at k=2.0. The minimum value of the χ^2 statistic is indicated by the red triangular point; the relative increase in S corresponding to the 68% independent confidence interval for 1 degree of freedom is equal to 0.99, and is shown as the horizontal gray line. The points where the computed S curve crosses that line are indicated by vertical gray lines: the true value of n lies within this region with 68% (1σ) probability.

9 IMD*Multiplot and IMD*Efficiency

IMD*Multiplot and IMD*Efficiency are separate programs included with IMD that can be used to further analyze and visualize IMD simulations. Both programs use data imported from .imd files that you create. IMD*Multiplot can also use measured data contained in the .imd files that you may have saved, and it can read measured data files directly as well. A large variety of analyses and visualizations can be produced by IMD*Multiplot when multiple measured data sets are imported into IMD*Multiplot, as we'll see later in this chapter. But we'll start in the next section with IMD*Efficiency, which is a much simpler program than IMD*Multiplot.

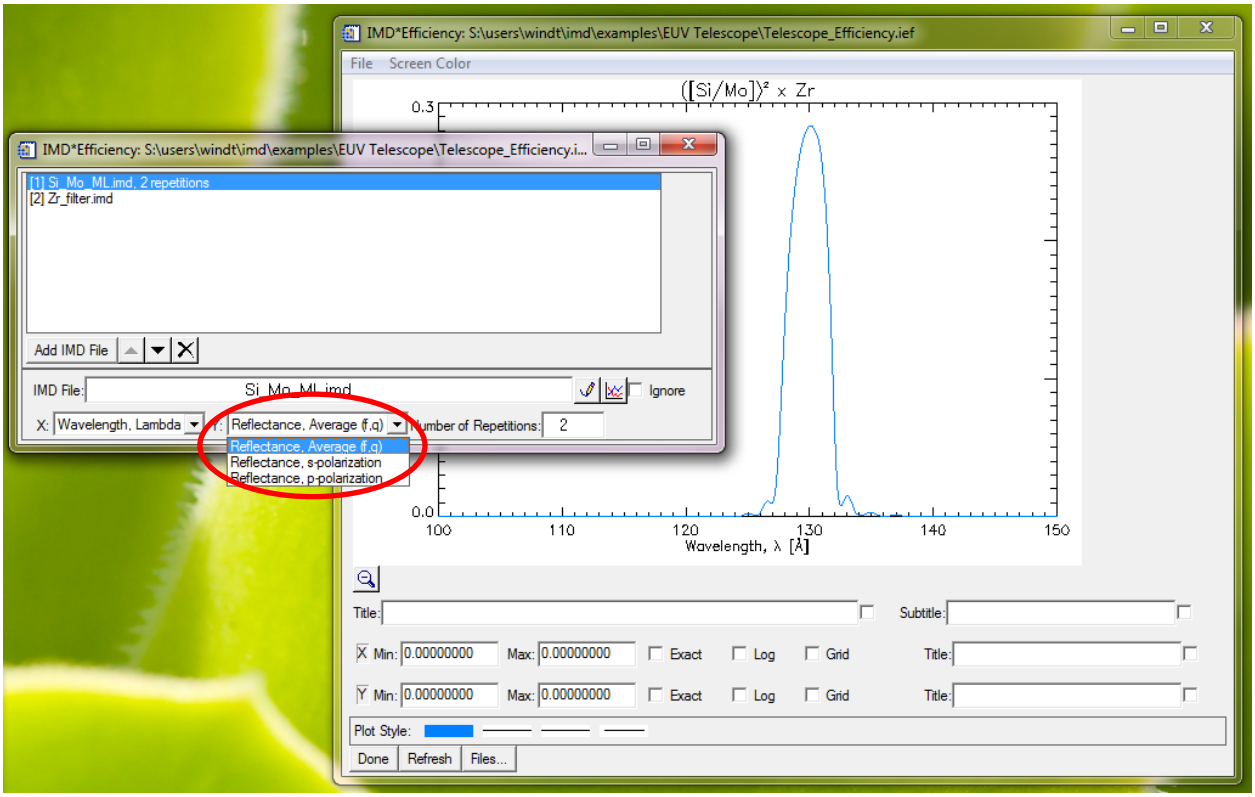
9.1 IMD*Efficiency

The purpose of the IMD*Efficiency program is to combine multiple IMD simulations – each corresponding to the calculated response of an optical surface that is part of a multi-element optical instrument, and each contained in its own .imd file (§7.1) – in order to compute the net efficiency of the instrument. We've already seen an example of how IMD*Efficiency works in §2.3.4: in that example we combined the calculated transmittance vs. wavelength of a Zr filter with the calculated reflectance vs. wavelength of a Si/Mo multilayer, in order to compute the efficiency of an EUV telescope comprising two Si/Mo-coated mirrors and a Zr filter in series. In fact, there's not that much more to explain about how IMD*Efficiency works in light of what we've already seen with that example in Chapter 2. To summarize: Start IMD*Efficiency by clicking the corresponding button in IMD Launcher, or by typing 'imdefficiency' at the IMD prompt if running IMD as an IDL application. Open the "IMD*Efficiency – Files" window by clicking the button labeled "Files…" at the bottom of the IMD*Efficiency window. Add .imd files to the list of files that are displayed in the list. Select each file in turn in the list, and enter the number of repetitions to be used for the selected file; the repetition number represents the number of times incoming light interacts with the optical surface represented by that .imd file. Once all the files are specified, click the "Refresh" button on the main IMD*Efficiency window to display the computed efficiency. You can adjust the plot appearance using the controls on the main IMD*Efficiency window. You can also export the computed efficiency to a plain text file using the "File→Export Efficiency…" menu option.

We've seen in §2.3.4 how to add .imd files directly to the "IMD*Efficiency – Files" window. In addition to that method, you can also use the "File→Send to IMD*Efficiency…" menu options on the IMD and IMDXPLOT menu bars to add .imd files to the list. You must first save IMD computation results to a .imd file before these menu options are available, however.

> **Note**: The first time you use "File→Send to IMD*Efficiency…" from the IMD menu bar, a new instance of IMD*Efficiency will be created. Additional .imd files will be sent to that same instance of IMD*Efficiency (so long as it exists) if you continue to use "File→Send to IMD*Efficiency…" from the IMD menu bar.

Once a .imd file has been added to the list of files in IMD*Efficiency, you can choose exactly which optical function contained in that file is to be used for the efficiency calculation. That is, even if you've computed just one optical function in IMD – Reflectance, for example – the .imd file will contain optical functions for s, p and average polarizations –Rs, Rp and R, respectively (as explained in §5.2.2). You can choose any one of those optical functions for use in IMD*Efficiency, by selecting the file of interest in the list on the "IMD*Efficiency – Files" window, and then using the drop-list, as shown in the next figure.



There are some restrictions on the type of .imd files that are allowed for use in IMD*Efficiency, however: each .imd file must contain the computation of a *specular* optical function calculated as a function of just *one* multi-valued independent variable, either wavelength/energy or incidence angle; all .imd files should use that same multi-valued independent variable. (Though all .imd files need not use the exact same values for the independent variable – interpolation will be used by IMD*Efficiency as necessary.) Example .imd files that can be used with IMD*Efficiency include those containing Reflectance vs. Wavelength/Energy, Reflectance vs. Angle, Transmittance vs. Wavelength/Energy, etc. Also, keep in mind that units cannot be adjusted in IMD*Efficiency: units are defined by the units that were specified in the first .imd file in the list when it was saved. (To be clear: you can change units by opening in IMD the first .imd file in the list, setting units as desired, and then re-saving the .imd file.) You can re-arrange the list of files on the "IMD*Efficiency – Files" window, using the Move File Up and Move File Down buttons (and). The plot icon ( ) next to the file name will send the selected file back to IMD.

There is no limit on the number of instances of IMD*Efficiency that can be opened simultaneously.

9.2 IMD*Multiplot

The IMD*Multiplot program is a multi-purpose application that can be used for a variety of analyses and comparisons of IMD computation results and/or measured data. The visualizations that are possible in IMD*Multiplot grew out of my own research needs, where it is sometimes necessary to extract information from a set of measured-data files that are related to each other in some way. For example, consider the case where a series of reflectance vs wavelength measurements from a single coating sample are obtained periodically over some length of time. IMD*Multiplot allows you to display those multiple measured reflectance vs. wavelength curves on a single plot, in order to more easily identify any changes that may have occurred. IMD*Multiplot also computes from each measured-data file values of Peak Reflectance (or Peak whatever-you-are-plotting), Peak Center (i.e., the wavelength/energy or angle value where the peak value of the optical function occurs), COM (Center-of-Mass, i.e., the point that divides the area under the curve), and CWHM (i.e., the center of the range defined by the FWHM); it's therefore possible to create plots of any of those quantities as a function of time, in order to quantitatively assess coating stability. As another example, when coating mirror substrates with EUV multilayers, the coating uniformity over the surface of the mirror can be extracted from measured reflectance vs. wavelength curves obtained over a grid of points on the mirror surface. IMD*Multiplot allows you to assemble such a collection of measured reflectance curves, and then compute 1D or 2D arrays of Peak Reflectance, Peak Center, etc, as a function of position on the surface (X,Y); line plots or contour plots of these quantities can then be produced in IMD*Multiplot to quantitatively assess coating uniformity.

There are three general scenarios where IMD*Multiplot can be used: First, it can be used to compare multiple IMD computations that have been saved as .imd files. We've seen an example of that use of IMD*Multiplot already, in §2.3.3, where we displayed the transmittance of a Zr filter and the reflectance of a Si/Mo multilayer on the same plot, as a function of wavelength. Second, it can be used to compare multiple IMD files that were saved with both IMD computations and measured data. We'll show how that works using an example, in §9.2.1.1. And third, IMD*Multiplot can be used to compare multiple measured data files, either imported directly in the "IMD*Multiplot – Files" window, or imported via .imd files. That can get a little complicated, as we'll see in §9.2.2. (To be clear: you can mix both .imd files and measured-data files in IMD*Multiplot, without limitations.)

To start IMD*Multiplot, click the corresponding button in IMD Launcher, or type "imdmultiplot" at the IMD prompt if you are using IMD as an IDL application. There is no limit on the number of instances of IMD*Multiplot that can be opened simultaneously.

The remaining sections of this chapter illustrate the use of IMD*Multiplot with various types of .imd and measured-data files.

9.2.1 IMD*Multiplot: Using IMD Files Containing Measured Data

We've already covered in §2.3.3 how IMD*Multiplot works when comparing two .imd files containing only IMD computations, and that extends to situations where you may have more than two such files. The basic procedure is this: open the "IMD*Multiplot – Files" window by clicking the button labeled "Files…" at the bottom of the main IMD*Multiplot window; then add .imd files by clicking the button labeled "Add IMD File"; after all the .imd files have been added to the list, click the "Refresh" button on the main window to display the data.

You can add .imd files directly to the "IMD*Multiplot – Files" window as just described, or you can use the "File→Send to IMD*Multiplot…" menu options on the IMD and IMDXPLOT menu bars. You must first save IMD computation results to a .imd file before these menu options are available, however.

> **Note**: When you first use "File→Send to IMD*Multiplot…" from the IMD menu bar, a new instance of IMD*Multiplot will be created. Additional .imd files will be sent to that same instance of IMD*Multiplot (so long as it exists) if you continue to use "File→Send to IMD*Multiplot…" from the IMD menu bar.

As in the case of IMD*Efficiency, you can select the optical function contained in each .imd file that's to be used by IMD*Multiplot, by selecting that file in the list on the "IMD*Multiplot - Files" window, and then using the drop-list labeled "Y:"; that drop-list is analogous to the IMD*Efficiency optical function drop-list described in §9.1. You can also specify the plot style (i.e., color, line-style, thickness, symbol, and symbol size) to be used for each file, using the controls on the "IMD*Multiplot – Files" window as you select each file. In addition, you can display statistics associated with each .imd file, using the check-boxes on the "IMD*Multiplot – Files" window labeled "Min", "Max", "Ave", "Int", "FWHM", and "FWHMin"; these are the same quantities that can be displayed in IMDXPLOT, as described in §6.6.1.1.

> **Note**: If you have defined a Region-of-Interest in the .imd file, then statistics will be computed over that ROI, just as they are in IMDXPLOT.

The same restrictions on the type of .imd files that are allowed for use in IMD*Efficiency apply to IMD*Multiplot: each .imd file must contain the computation of a *specular* optical function calculated as a function of just *one* multi-valued independent variable, either wavelength/energy or incidence angle, and all files should use that same multi-valued independent variable. Also, units cannot be adjusted in IMD*Multiplot: units are defined by the units that were specified in the first .imd file in the list when it was saved. You can re-arrange the list of files on the "IMD*Multiplot – Files" window, using the Move File Up and Move File Down buttons (and); the file list can also be modified using the Cut and Paste buttons. The plot icon ( ) next to the file name will send the selected file back to IMD.

9.2.1.1 Example: IMD*Multiplot – Comparing the Performance of Three Multilayers

The .imd files added to the list of files in IMD*Multiplot may contain measured data. You can mix .imd

files containing measured data with .imd files that don't contain measured data. And you can choose to ignore either the IMD computation or the measured data in any .imd file that you add to the list. We'll explore how to use .imd files containing measured in this section through an example: we'll compare three .imd files, each containing different measured data and simulations of those respective data.

The IMD*Multiplot file associated with this example is located in "`imd/examples/IMD-Multiplot/C_CoCr_TH=45.0.imp`", along with the three .imd files that have been loaded into that .imp file: "`C_CoCr_d=4.2nm_als.imd`", "`C_CoCr_d=3.8nm_als.imd`", and "`C_CoCr_d=3.5nm_als.imd`". Each of these .imd files in turn contains measured data – Reflectance vs. Wavelength data at 45° incidence in the soft X-ray range, measured at the ALS Beamline 6.3.2 (provided by E. Gullikson) – for three $C/Co_{.8}Cr_{.2}$ periodic multilayers having periods d=4.2 nm, d=3.8 nm, and d=3.5 nm, respectively. The .imd files contain simulations of the measured data, using the measured beam polarization to compute average reflectance. The measured-data files that were imported into the .imd files are called "`C_CoCr_d=4.2nm_als.abs`", "`C_CoCr_d=3.8nm_als.abs`", and "`C_CoCr_d=3.5nm_als.abs`". These data files are imported using the 'rd_als' routine described in §11.4.3, and illustrated previously, using some of the very same data, in §8.2.2.

> **Note**: To use the example files discussed here, it is essential that you change your working directory to "`imd/examples/IMD-Multiplot`". The .imd file names saved in the IMD*Multiplot (.imp) file, and the measured-data file names saved in the .imd files, are all specified without their full paths (in order to ensure that these files work on any platform). You will encounter errors if you attempt to open the .imp file without setting "`imd/examples/IMD-Multiplot`" as your working directory, using "File→Change Working Directory…" from the IMD or IMD Launcher menu bars, or by using the 'cd' command at the IMD prompt.

After starting IMD*Multiplot, using the methods described above, select "File→Open…" from the IMD*Multiplot menu bar and open the IMD*Multiplot file "`C_CoCr_TH=45.0.imp`". The IMD*Multiplot window looks like this after the file loads:

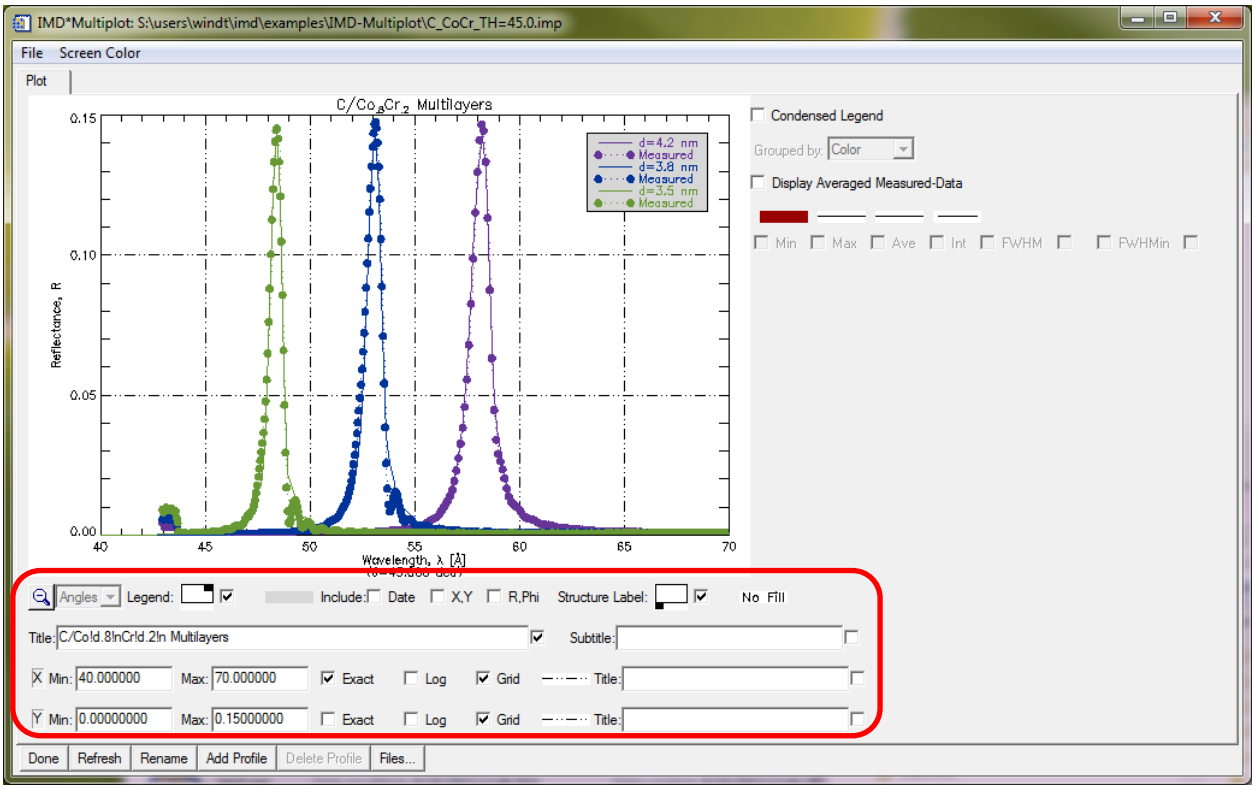


Figure 52. IMD*Multiplot window displaying the contents of the example file "C_CoCr_TH=45.0.imp", which contains three .imd files, each with measured data and an IMD simulation of reflectance vs. wavelength.

We've added grid lines to the plot in Figure 52, set the X and Y axis ranges, adjusted the legend position and color, and added our own title, all done using the controls on the main window. The three simulations are displayed using solid lines, while the measurements are displayed using filled circles connected by dotted lines. These last settings are made on the "IMD*Multiplot – Files" window, which is shown in the next figure.

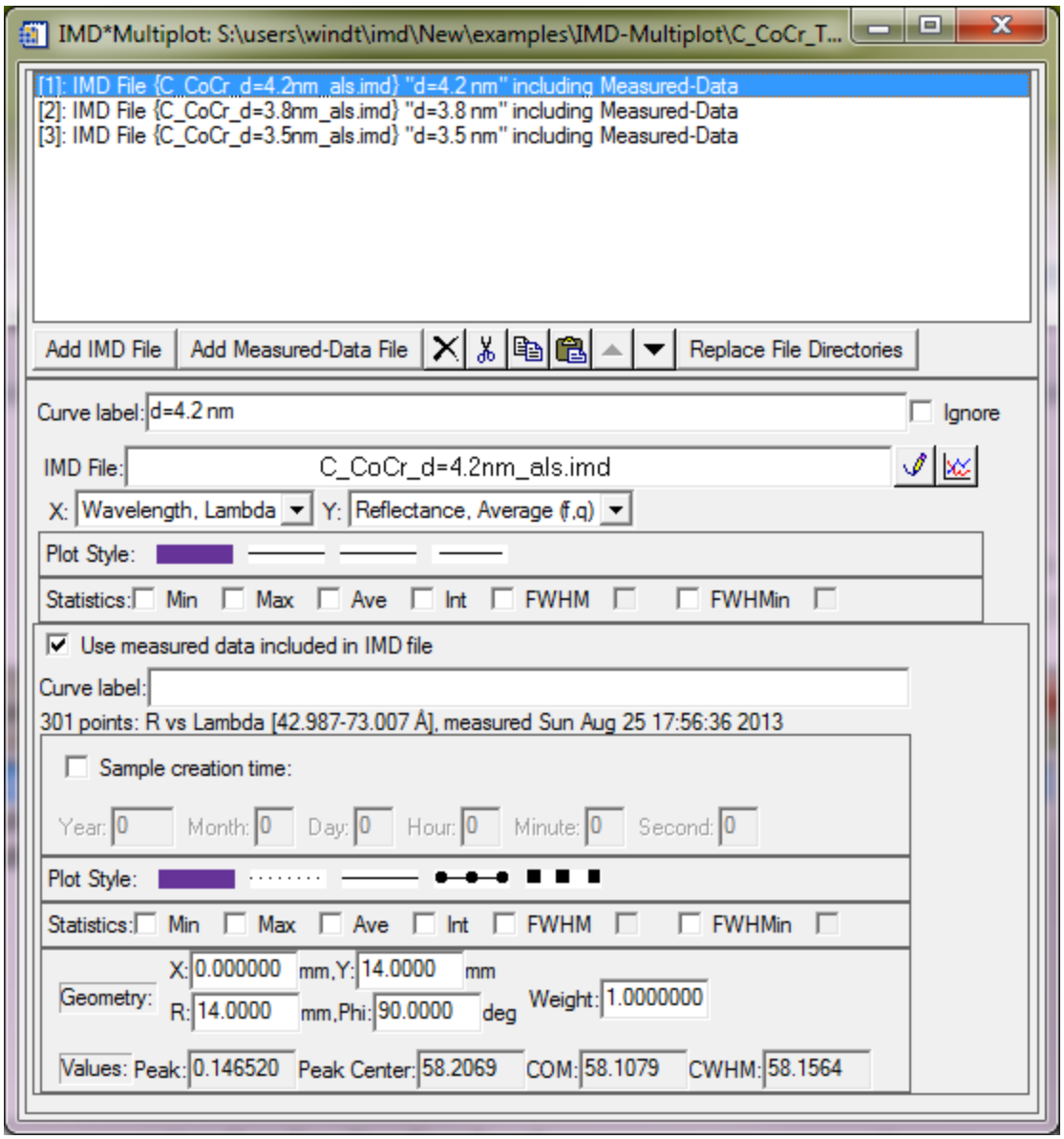


Figure 53. IMD*Multiplot - Files window associated with the example shown in Figure 52.

Let's look at each control shown in Figure 53, starting with the box labeled "Curve label:". Using that box we've added 'd=4.2 nm' as a label in the plot legend (Figure 52) for the file selected in Figure 53, which is the first file in the list. By explicitly specifying a label, we've over-ridden the default label, which is derived from the contents of the IMD file automatically (as in §2.3.3). The box labeled "Ignore" remains unchecked in this example, so this file will be used in the plot. Next down is a text-box to enter the file name, along with a button to choose a file interactively(), and a button to send the specified .imd file back to IMD (). In this example we've used the usual controls on the next row, next to "Plot Style", to set the plot color, line-style, thickness, symbol and symbol size. We haven't checked any of the boxes that would display statistics, which is the next row down.

When an IMD file contains measured data, like the file selected above in Figure 53, there are number of additional controls available on the "IMD*Multiplot – Files" window. Let's now go through those controls, starting with the check-box labeled "Use measured data included in IMD file" and working down. That check-box does just what is says, and can be un-checked if you don't want to display the measured data included in the .imd file (it's initially checked by default if the file contains measured data.) You can override the default 'Measured Data' label used in plot legends by entering text in the box labeled "Curve label:". We'll explain later, in §9.2.2.2, the area with the box labeled "Sample creation time:". Next are the usual controls for plot style – color, line-style, thickness, symbol, and symbol size – in this case specific to the measured data. (We've used the same color for both measured data and the IMD simulation in this particular example, but it's not necessary to do so.) After that are the usual check-boxes to control display of statistics for the measured data contained in the IMD file currently selected in the list. Again, any ROI for measured data that may be defined in the .imd file will be used when computing those statistics. Finally, there's a set of boxes to the right of the label "Geometry:". The boxes labeled "X:", "Y:", "R:", and "Phi:" can be used to enter (X,Y) or (R,Φ) coordinates identifying the location of the measurement on the substrate surface, in cases, for example, where coating uniformity is of interest. (But note that X, Y, R and Φ can represent any quantities whatsoever – they don't have to be actual position coordinates, as we'll see later in §9.2.2.3.) And finally there is one last row labeled "Values:". This row shows the computed values of Peak (=Peak Reflectance, in this case), Peak Center (i.e., the wavelength where Reflectance = Peak Reflectance), COM (Center-of-Mass, i.e., the point that divides the area under the curve), and CWHM (i.e., center of the range defined by the FWHM) computed for the measured data. As we'll see in the next section, any of these four quantities (Peak, Peak Center, COM, and CWHM) can be plotted in IMD*Multiplot as a function of X, Y, R, and/or Φ. In fact we'll see a lot more about that in §9.2.2.1.

The menu bar on the main IMD*Multiplot window includes options for saving to an IMD*Multiplot (.imp) file, and for printing graphics. Here's the printed output from the example shown above in Figure 52, after conversion to a JPEG for import into the Word document used to create this PDF:

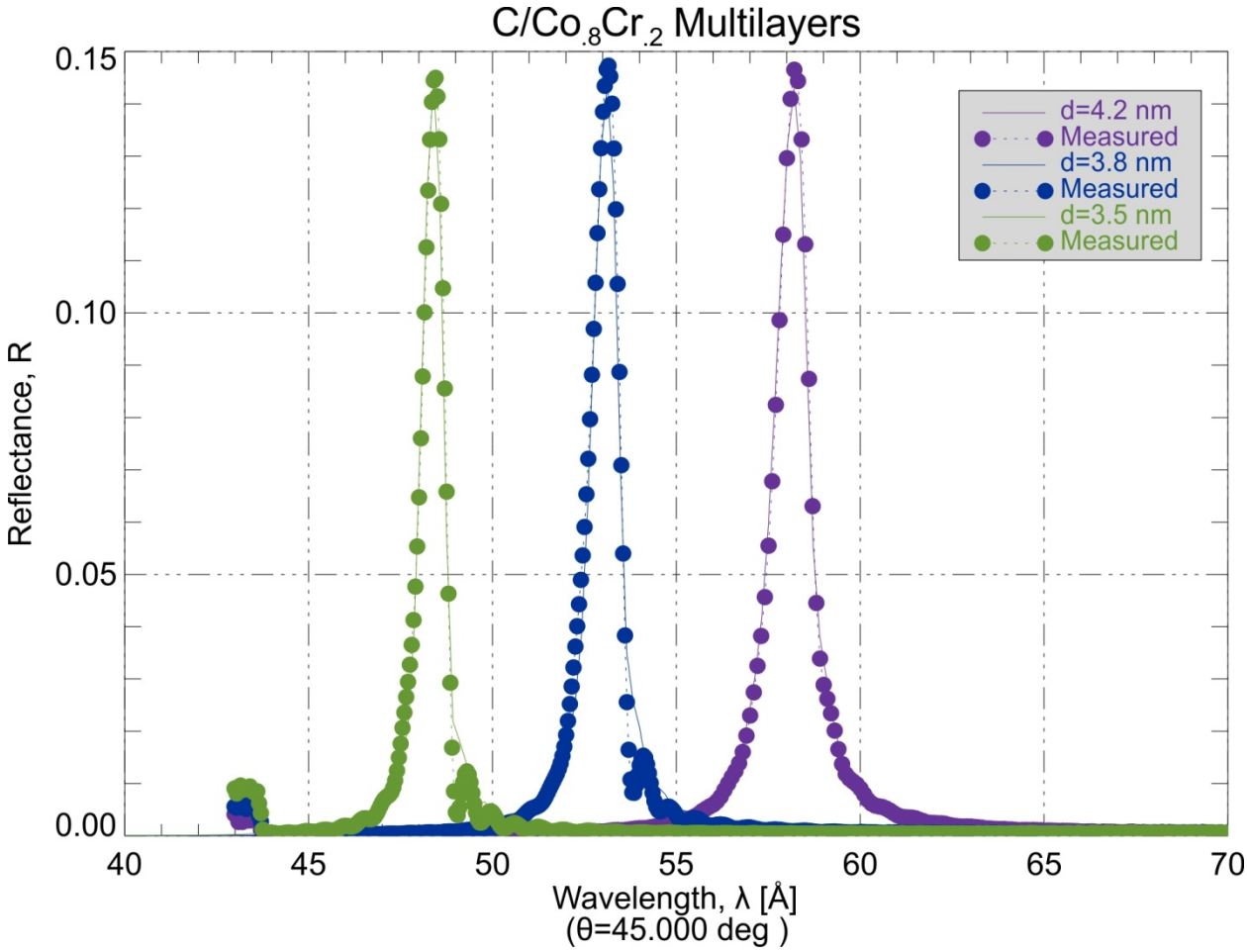


Figure 54. Graphics created using the File→Print… menu option for the IMD*Multiplot example shown in Figure 52. A PostScript output file was created and then converted to JPEG (using CorelDraw) for import into this document.

9.2.2 IMD*Multiplot: Visualizing Measured Data Sets

When multiple measured data sets are included in IMD*Multiplot, many new capabilities for visualization are possible, as we'll see through the two examples presented in this section. The first example shows how to visualize a collection of reflectance-vs-wavelength data measured over a grid of points on a multilayer-coated mirror, in order to map coating uniformity over the mirror surface. The second example demonstrates how to monitor coating temporal stability, using reflectance measurements of one or more coatings made periodically over time. We'll tackle the more complicated example first.

9.2.2.1 Example: IMD*Multiplot – Using Measured Data to Map Coating Uniformity

This example uses files contained in the directory '`imd/examples/IMD-Multiplot - Coating Uniformity`'; please **be sure to use this as your working directory in IMD before attempting to open the IMD*Multiplot file '`Mo_Y_mirror_uniformity.imp`'** located in that directory.

There are four measured-data files in the directory just specified: each file contains reflectance vs wavelength data, obtained near normal incidence, from a figured, Mo/Y-ML-coated mirror (~20 cm in diameter) that was measured at various locations (R,Φ) on the mirror surface. The data were obtained at the ALS Beamline 6.3.2 (provided by R. Soufli/E. Gullikson), at four radii (measured from the mirror center): R=52 mm, R=60 mm, R=76 mm, and R=90 mm. The corresponding data files – `Mo_Y_mirror_r=52mm.dat`, `Mo_Y_mirror_r=60mm.dat`, `Mo_Y_mirror_r=76mm.dat`, and `Mo_Y_mirror_r=90mm.dat` – each contain R(λ) data at several azimuthal positions, Φ. We'll use the 'rd_als' routine (§11.4.3) once again to read these files; however for these ALS files, which contain multiple columns of data, we'll use the syntax "rd_als,x_m_column,y_m_column", where x_m_column is the column index corresponding to the X_m values (i.e., wavelength), and y_m_column is the column index corresponding to the Y_m values: x_m_column is always equal to 0 for these particular files, while y_m_column values will range from 1 to 7, as we'll see.

After starting a new instance of IMD*Multiplot (by typing 'imdmultiplot' at the IMD prompt, or by clicking "IMD*Multiplot" on IMD Launcher), use "File→Open…" from the menu bar to open the file "`Mo_Y_mirror_uniformity.imp`". When this .imp file was created, we had added 30 Measured Data files to the file list, using the "Add Measured-Data File" button on the "IMD*Multiplot – Files" window:

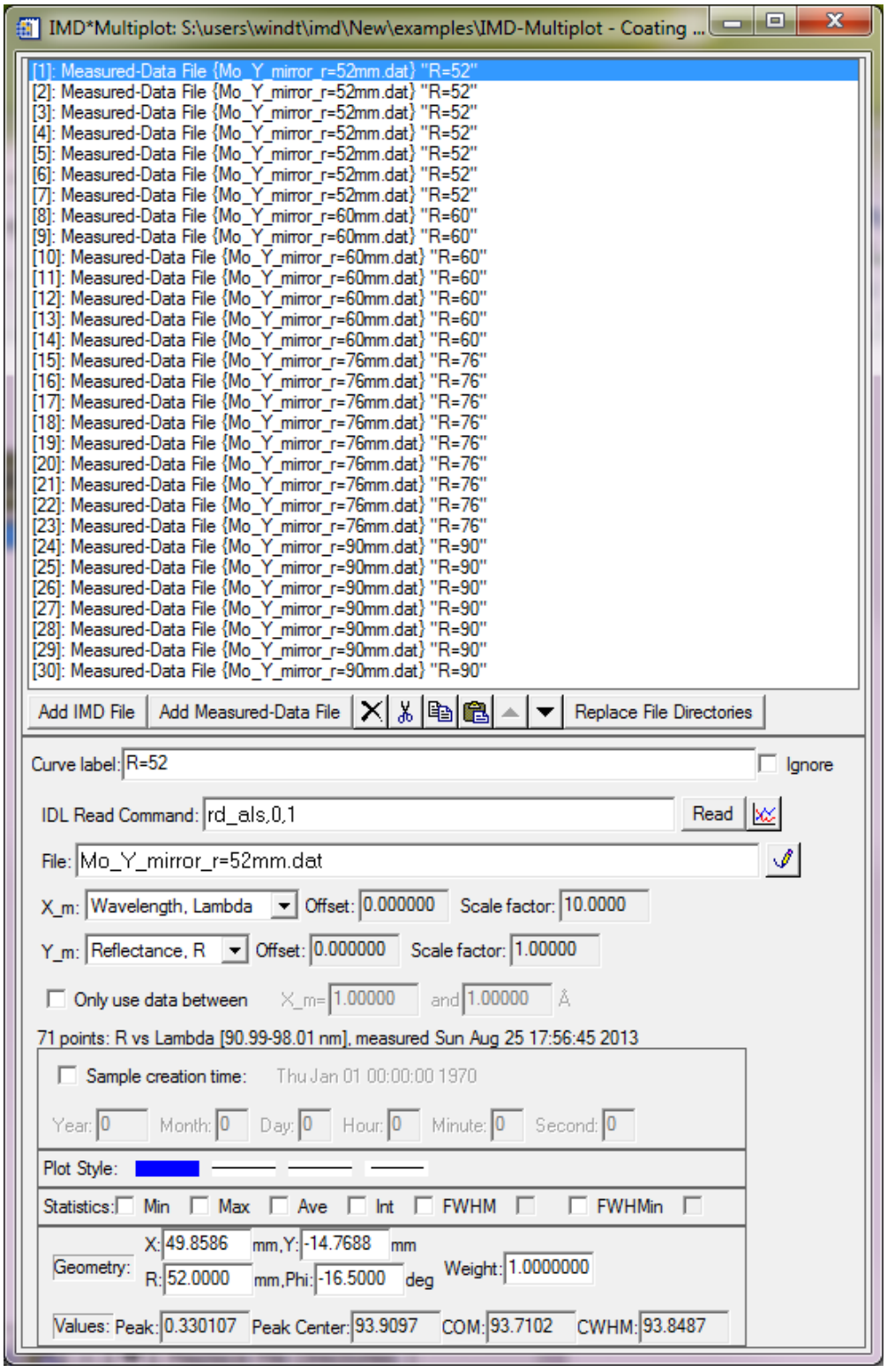


Figure 55. "IMD*Multiplot - Files" window showing the list of 30 measured-data sets included in the example .imp file used this section: "`imd/examples/IMD-Multiplot - Coating Uniformity/Mo_Y_mirror_uniformity.imp`"

In Figure 55 above, the first file in the list is selected. Let's go through each of the settings we've configured for this file, starting with the box labeled "Curve label:". We've entered "R=52" as the label for this first file; in fact we've used the same label for the first 7 files in the list, as they all correspond to measured data obtained at R=52 mm on the mirror surface (and they all use the same .dat file). Each of the first 7 files correspond to different azimuthal locations, however. The next row below the "Curve label:" box shows the IDL command we've used to read the file, '`rd_als,0,1`' in this case, and the line below that shows the data-file name – "`Mo_Y_mirror_r=52mm.dat`": we're reading the first two columns of data in that file for X_m and Y_m, respectively. We've used the two drop-lists in the next two rows to specify that X_m corresponds to Wavelength and Y_m corresponds to Reflectance; and as the X_m values in this file are in units of nm, we've entered a scale factor of 10.0 for X_m to convert the X_m values to units of Å, as is required.

> **Note**: While X_m values must always be *defined* in units of Å or degrees from normal in IMD*Multiplot, in cases such as this, where no .imd files are included in the file list, X_m values will be *displayed* in IMD*Multiplot in whatever units were defined in IMD at the moment we created the original instance of IMD*Multiplot that ended up as this example file. Be sure to configure units as you prefer before creating a new instance of IMD*Multiplot that will contain only measured-data files.

Continuing with the controls shown in Figure 55: The next row down has a check-box labeled "Only use data between:"; this control serves an analogous function as that described in §8.2.2 when importing measured data into IMD: it's function is to limit the range of measured data that will be used by IMD*Multiplot. For this example we'll use the entire range of measured data, which comprises 71 points from λ=90.992 to 98.010 Å, as indicated on the next row down (along with the measurement date, which is simply the file-creation time returned by the operating system.) After that is the "Sample creation time" area – again, we'll explain what that's all about in §9.2.2.2 where we're interested in time-stamps associated with each measured-data file. Next are the usual controls for plot style and display of statistics. And finally, at the bottom of the window, is an area just like the area described earlier in §9.2.1.1 when .imd files containing measured data are used in IMD*Multiplot: this is where you can enter (X,Y) or (R,Φ) coordinates for the measurement – for this first file in the list, we've entered the known measurement coordinates as (R,Φ)=(52 mm, -16.5°). Below the location coordinates are the computed values of Peak, Peak Center, COM, and CWHM for this measured data; these values are explained in §9.2.1.1, and for this particular measured-data file correspond to the values of Peak Reflectance (R_{max}=0.330107), Peak Center Wavelength (λ_{center}=93.9097 Å), COM (λ_{com}=93.7102 Å), and CWHM (λ_{cwhm}=93.8487 Å). There's also a box labeled "Weight:" in this area, with a default value of 1.0; the Weight value is used when computing the average reflectance of an entire set of measured data, as we'll explain below.

The next 6 measured-data files in the list (numbers [2] through [7]) all corresponding to data extracted from the same file "`Mo_Y_Mirror_r=52mm.data`": they correspond to measurements made at other Φ values at the same radius on the substrate, R=52 mm. The "IDL Read Command" was specified

accordingly for each file, as 'rd_als,0,n', where n=2,...7, and the Φ values corresponding to the azimuthal angles used for each measurement were entered in the box labeled "Phi:" in the Geometry area near the bottom of the window as each file was selected. The process was repeated for the remaining files containing data at other radii. The curve labels were specified similarly. We've also used the same plot color for data measured at the same radii, i.e., R=52 mm in blue, R=60 mm in violet, R=76 mm in pink, and R=90 mm in green.

Figure 56 shows the IMD*Multiplot main window when the example file being discussed here is first opened. All 30 reflectance-vs-wavelength curves are displayed on the same plot. We've checked the 'Condensed Legend' box on the right of the plot: using a condensed legend means that all files configured with the same color (in this case) are represented by only one line in the legend. (A condensed legend can also be constructed based on 3 other plot style attributes as well: line-style, thickness, and symbol. The order of labels displayed in a condensed legend corresponds to the selected attribute's numerical values arranged in increasing order.) As we've selected the same plot color for all files having the same radii, the legend includes just one entry each for R=52, R=60, R=76 and R=90. Note that, when a condensed legend is used, the curve label specified for the first file of each condensed group is used for the legend label representing that group.

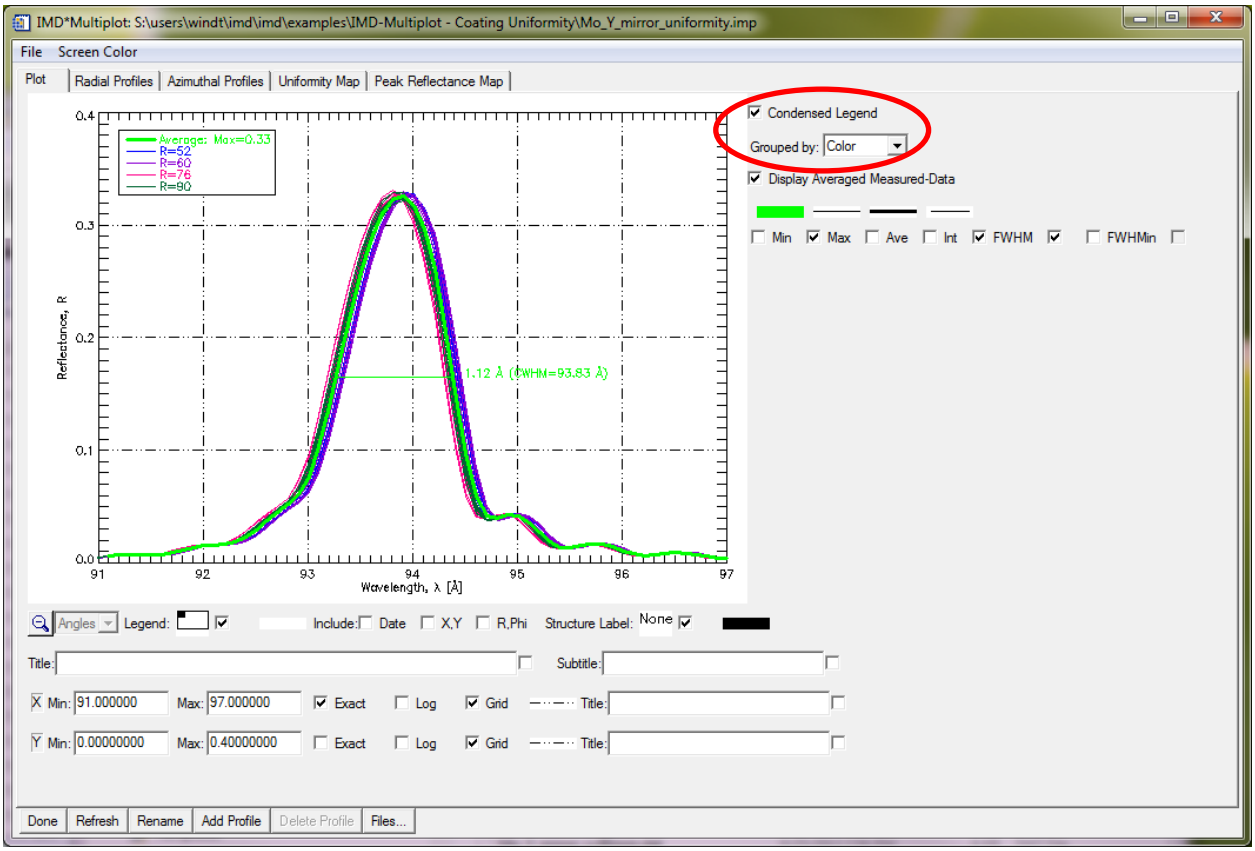
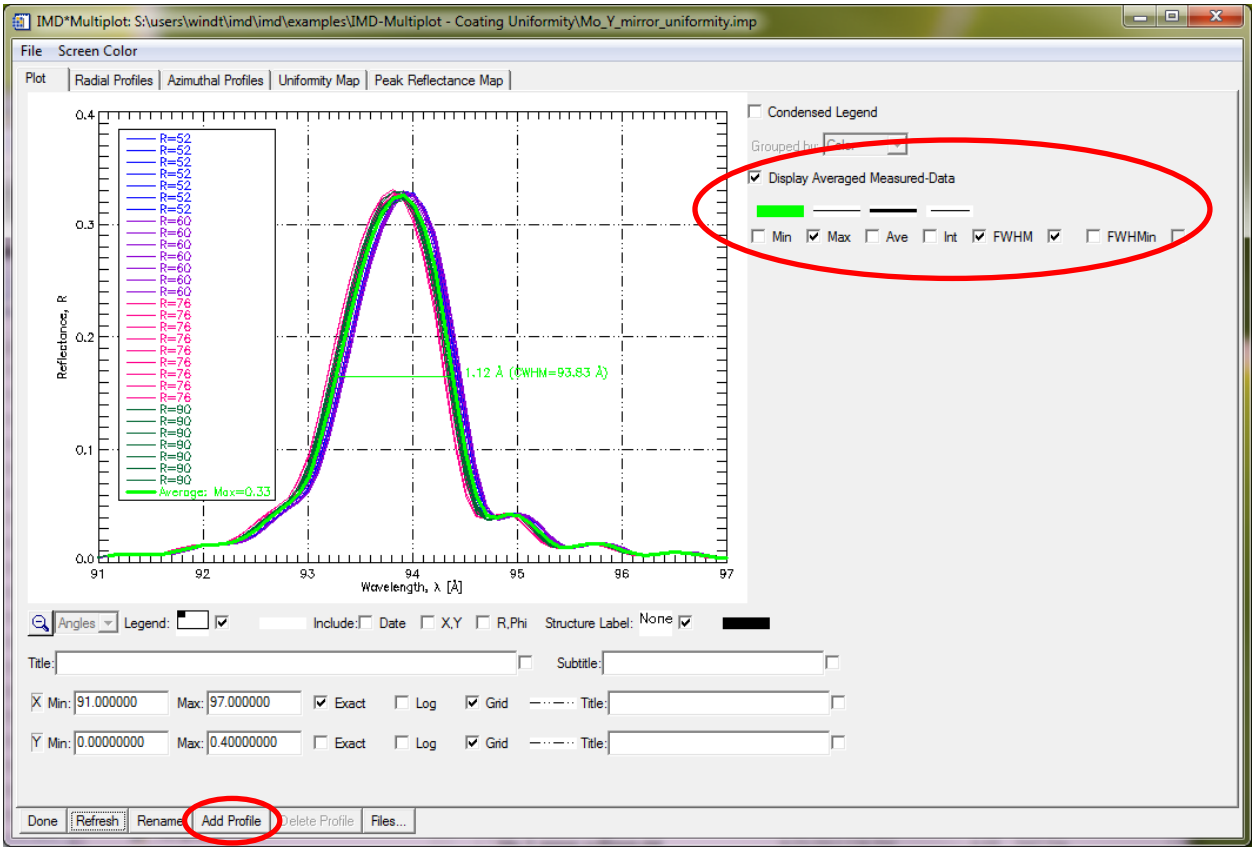


Figure 56. IMD*Multiplot window as it appears when the file "imd/examples/IMD-Multiplot - Coating Uniformity/Mo_Y_mirror_uniformity.imp" is first opened.

When the 'Condensed Legend' box is un-checked, each file will be listed in the legend. Here's how the IMD*Multiplot window looks in that case:



We've checked the box labeled "Display Averaged Measured-Data" in Figure 56, and in the figure above as well. When this box is checked, IMD*Multiplot will compute the average of all the measured data included in the list of files (both directly-read Measured-Data Files and measured data included in any .imd files included in the list.) The independent variable associated with Averaged Measured-Data is determined by the independent variables of all the measured data sets, using only that portion of the independent variable that overlaps for all data sets; the measured data sets therefore need not use the exact same independent variable values, so long as they overlap sufficiently. The display of the Averaged Measured-Data is determined by the plot style controls under the check-box (e.g., for color, line-style, thickness, symbol and symbol size), and statistics for the averaged measured-data can be displayed on the plot as well, as we've done in this example, by using the associated check-boxes. Averaged measured-data, when computed, can also be exported to a plain text file, using the "File→Export Averaged Measured-Data…" menu option.

The "Weight:" box on the "IMD*Multiplot – Files" window mentioned above can be used to weight each measured data file in order to compute a weighted average. That is, if a non-uniform grid of measurement locations is used, then different measured-data files may be representative of a smaller or

larger area on the mirror surface than other files; specifying a weight value unique to each measurement point can be used to compensate accordingly, so that the weighted average accurately represents the average performance of the coating when uniformly illuminated.

The button labeled "Add Profile" (highlighted in the figure above) is used to create additional tabs containing measurement "profiles" on the IMD*Multiplot window, as was done in the case of our example shown in Figure 56. For this example we've created four additional profiles, and have named them (using the button labeled "Rename" next to the "Add Profile" button) Radial Profiles, Azimuthal Profiles, Uniformity Map, and Peak Reflectance Map. We'll now explain the purpose of each of these additional profile tabs, and how they were created.

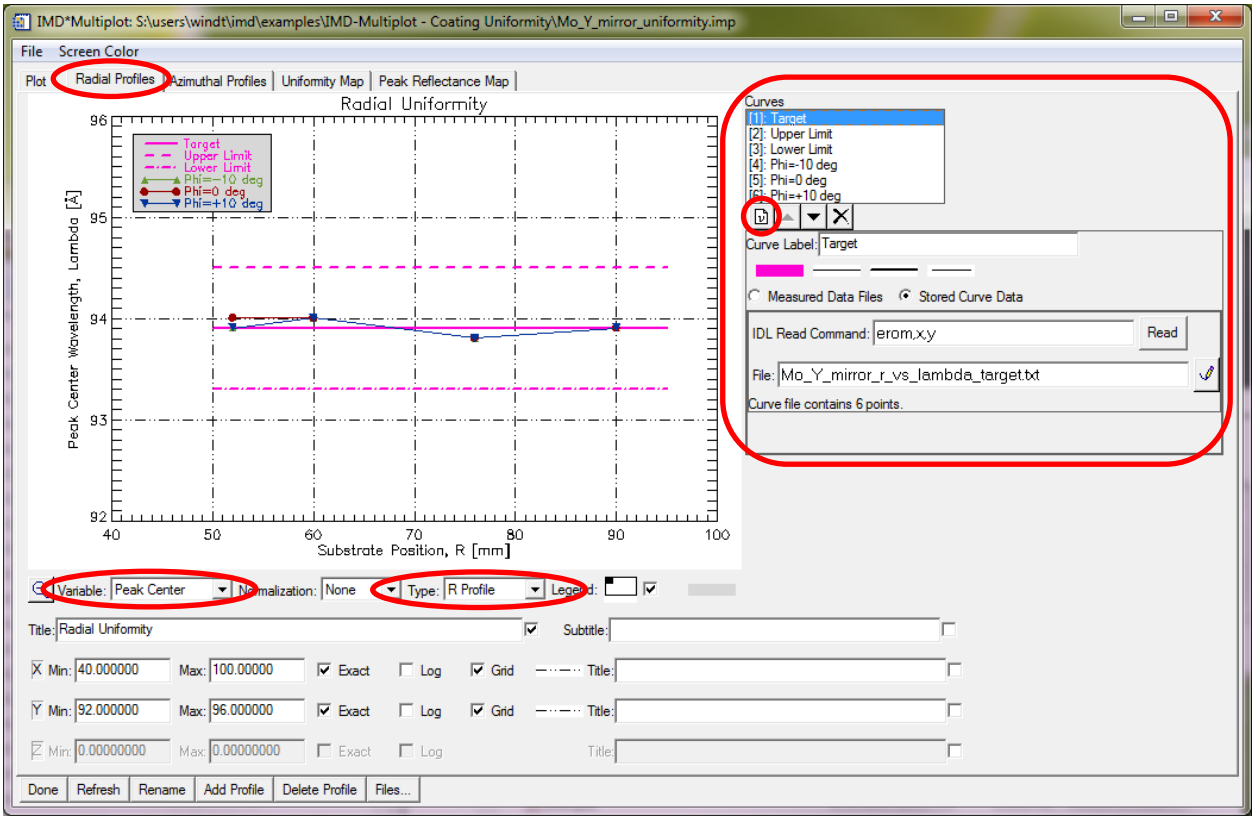
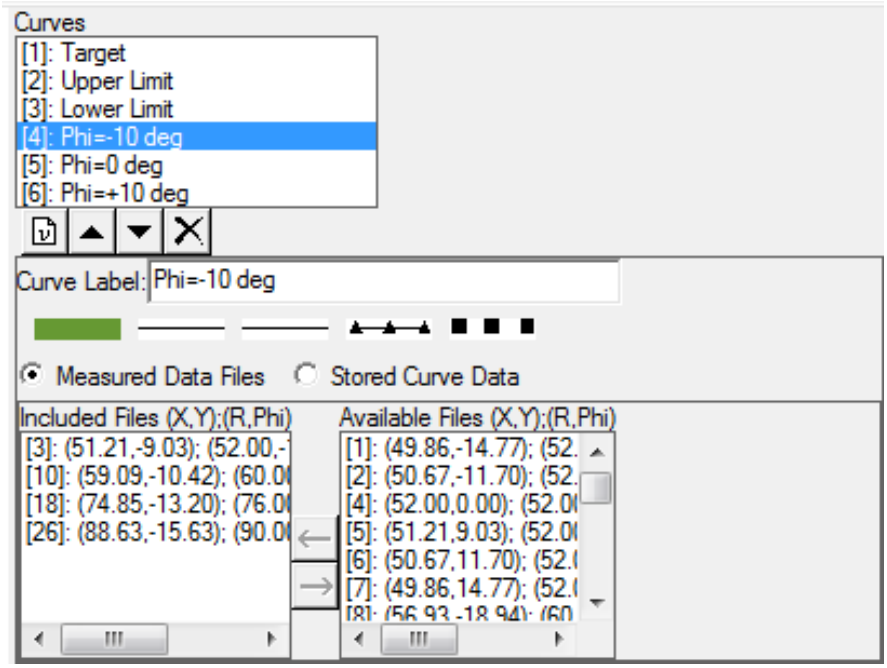


Figure 57. The "Radial Profiles" tab of the IMD*Multiplot window shown in Figure 56.

The Radial Profiles tab shown above in Figure 57 displays a plot of Peak Center Wavelength, λ_{center}, in units of Å, as a function of radial position, R, on the mirror substrate. These particular dependent and independent variables were selected using the drop-lists highlighted in the figure, labeled "Variable:" and "Type:". In this example there are three curves displaying measured data, as indicated in the legend, corresponding to λ_{center} values computed for measurements made as a function of radius at three Φ values, Φ=-10°, Φ=0°, and Φ=+10°. There are also three pink straight lines included in the plot: as indicated in the legend, these correspond to the target value of λ_{center} as a function of radius, and the allowable upper and lower limits on λ_{center} that were specified for this mirror.

The six curves shown in Figure 57 were created using the controls to the right of the plot, highlighted in red. The "New Curve" button ( ) is used to add a new curve to the list. Each of the six curves we've created using the "New Curve" button are listed by number in the "Curves" list at the top (and correspond to the six items in the plot legend), and we've selected the first curve in that list in Figure 57. We've specified "Target" as a label for that first curve to be used in the plot legend, as it corresponds to the target Peak Center wavelength as a function of radius, in this case. We've also checked the radio button labeled "Stored Curve Data" for this first curve in the list, which means that this particular curve is not actually derived from the measured-data files included in this instance of IMD*Multiplot, but instead the data used to generate the curve is read from a separate text file, called "`Mo_Y_mirror_r_vs_lambda_target.txt`", that we had already created: we've specified the IDL command to read that file as "erom,x,y", as the file is a plain-text file containing two columns of data, corresponding to the target values of λ_{center} as a function of radial position on the mirror surface, and so it can be read using erom (see §11.4.1) to define the curve's x and y values. You can enter in this text-box whatever valid IDL commands you like to read Stored Curve Data files and define x and y – the erom routine is just one possible choice. We've used the plot style controls located below the "Curve label:" box to set the color of this curve to pink. The Upper Limit and Lower Limit curves also use Stored Curve Data files, and were similarly configured.

Curve numbers 4, 5 and 6 in the "Curves" list are constructed from the measured data files: when the radio button labeled "Measured Data Files" is checked, as it is for curves 4, 5 and 6, the Curves area looks like this:



In the figure above the Curves area displays two lists, which are used to select the files that are used construct the curve: on the right is a list of all "Available" files, i.e., those files containing measured data

that have not yet been "included" in the selected curve, and on the left is a list of the "Included Files", i.e., those files that have been selected for inclusion in this curve. The files are listed in both cases by number (corresponding to their order in the "IMD*Multiplot - Files" window), along with the (X,Y) and (R,Φ) values that were specified in the Geometry area on the "IMD*Multiplot - Files" window for each file. The left (←) and right(→) arrow buttons located between the two lists are used to move files from the Available Files list to the Included Files list and vice versa.

For the curve selected in the figure above (i.e., curve number 4), we've included file numbers 3, 10, 18 and 26, which all correspond to measurements made at Φ=-10°, at different radii. In other words, curve number 4 comprises four points, corresponding to the values of Peak Center as a function of R, at Φ=-10°; the four points are derived from file numbers 3, 10, 18 and 26. Similarly, for curve number 5, we've included file numbers 4, 11, 19 and 27, corresponding to measurements made at Φ=0°, and for curve number 6 we've included file numbers 5,12, 20 and 28 that were obtained at Φ=+10°. We've specified labels appropriately for each of these curves, and have used the plot style controls to draw each curve in its own color, and with a different symbol, as in Figure 57.

The dependent and independent variables used to construct each curve in a 1D profile are determined by the selections made using the drop-lists on the main IMD*Multiplot window labeled "Variable:" and "Type:". For the example shown in Figure 57 we had selected Peak Center for "Variable:", and "R Profile" for type, hence the curves comprise values of Peak Center as a function of R, where the Peak Center values are computed by IMD*Multiplot and the R values are entered by you. Other "Variable:" and "Type:" selections work the same way.

For example, the "Azimuthal Profiles" tab shown in Figure 58 was created the same way as the "Radial Profiles" tab just described, except that the four curves displaying measured data were constructed by grouping measured-data files obtained at the same radii, but at different Φ values:

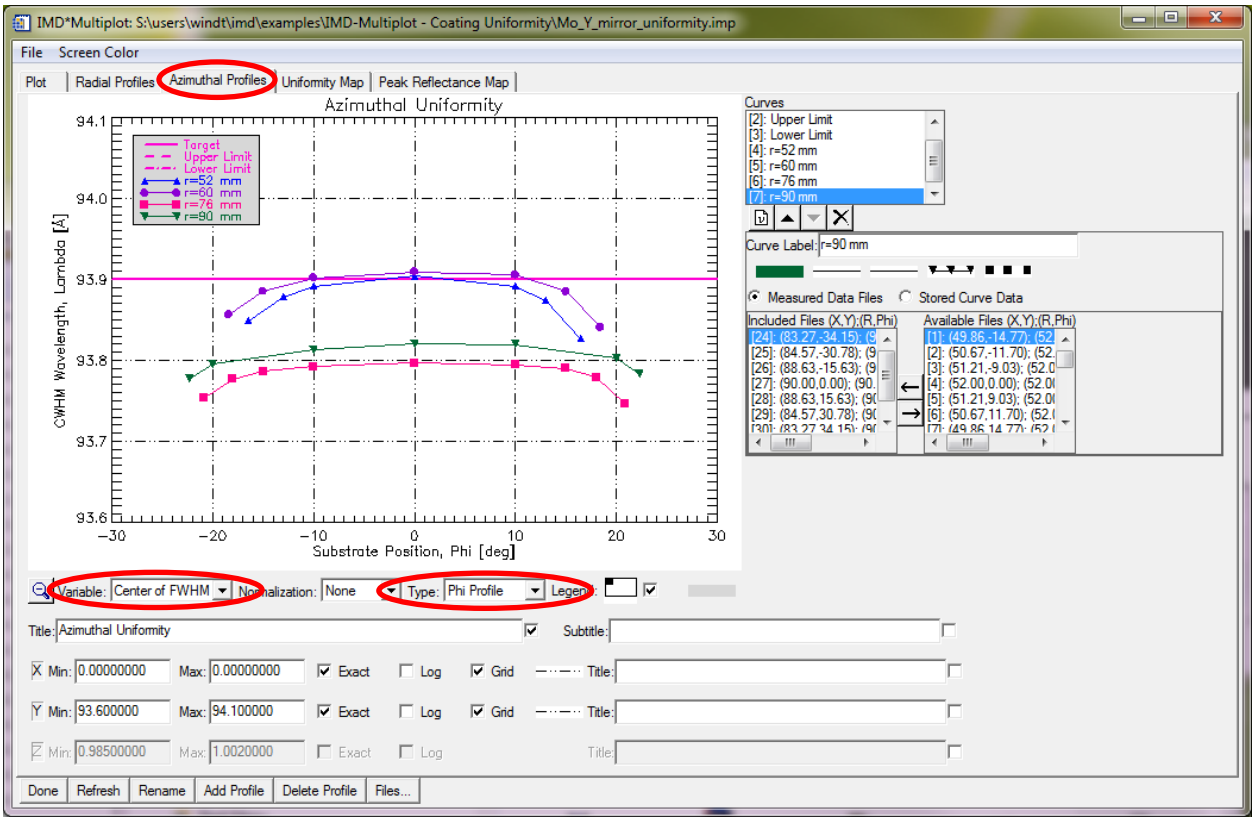


Figure 58. The "Azimuthal Profiles" tab of the IMD*Multiplot window shown in Figure 54.

For the azimuthal uniformity data shown in Figure 58, you'll notice that we've selected "Center of FWHM" (CWHM) as the dependent variable, not Peak Center as in Figure 57, and we've adjusted the vertical axis to zoom in on the measured data curves. The CWHM values provide much better resolution than Peak Center values: the reason for the superior resolution of the CWHM value is that this quantity is computed numerically from the raw data using interpolation, whereas the Peak value determined from measured data simply corresponds to the discrete X_m point where the peak value occurs, and is thus limited by the resolution of the original measured data.

The fourth tab, labeled "Uniformity Map", displays a contour plot of normalized CWHM as a function of position (X,Y), as shown in Figure 59. When "X-Y Contour" is selected using the "Type:" drop-list, all measured data is used (excluding those that are marked to be ignored) – there are no Curves to construct. CWHM was selected as the dependent variable using the "Variable:" drop-list on the main IMD*Multiplot window, and Global Normalization was selected using the "Normalization:" drop-list.

Contour plots display smooth, filled contours of the selected variable as a function of position (X,Y); the contours are computed using bi-linear interpolation between the measurement locations. When a contour plot is selected, additional controls are available to select the color table used draw the filled contours, and to display the measurement points as small green squares, contour lines (not selected in

Figure 59), and a color bar. The "Isotropic" box will cause the X and Y axis of a contour plot to be displayed using a 1:1 aspect ratio.

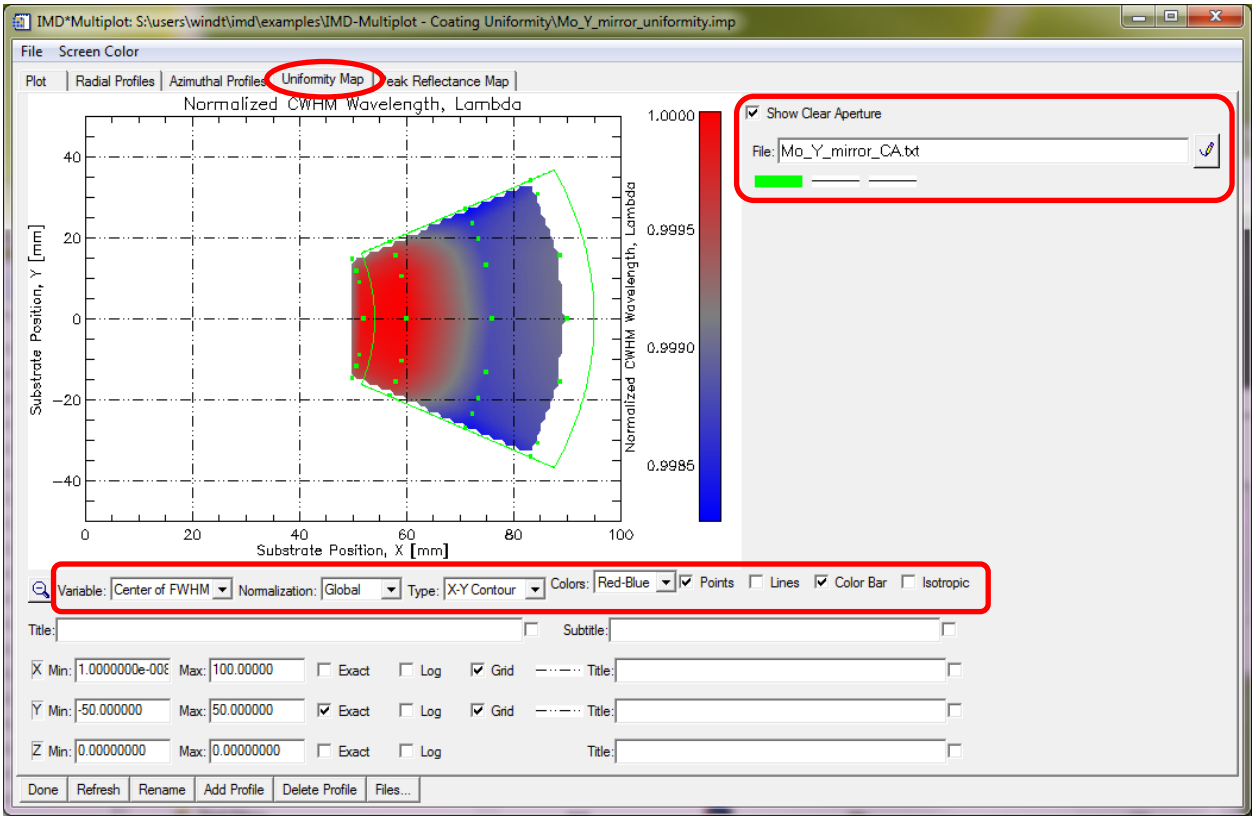


Figure 59. The "Uniformity Map" tab of the IMD*Multiplot window shown in Figure 54.

> **Note**: When a 1D Profile is selected (e.g., as in Figure 57 and Figure 58) two types of normalization options are available: Global and Individual. Global normalization means that all curves will be normalized to 1.0, using the maximum value determined from all curves; Individual normalization means that each curve is normalized to 1.0 separately, using the maximum value determined from that curve.

The coating being analyzed in this particular example was deposited only over a portion of the mirror surface (as can be inferred from the limited spatial extent of the measurements apparent in Figure 59), using masked deposition. (Other multilayer coatings were deposited elsewhere on the mirror surface.) The specified tolerance on multilayer coating uniformity (i.e., the Upper and Lower Limits displayed in Figure 57 and Figure 58) was defined only over the clear aperture (CA). IMD*Multiplot makes it possible to overlay the outline of a CA on a contour plot, as we've done in Figure 59 above, using the controls to the right of the plot: we've checked the box labeled "Show Clear Aperture", and have specified the file "`Mo_Y_mirror_CA.txt`", which contains data corresponding to the outline of the CA. (The IDL program used to generate this particular CA is included in the examples directory as well.) CA files must be readable by the erom routine, described in §11.4.1.

The contour plot shown above reveals how the variation in coating uniformity (proportional to CWHM) varies in two dimensions over the mirror surface. We can also see how the peak reflectance of the coating varies over the mirror surface: the final tab of this IMD*Multiplot example displays such a contour plot, of peak reflectance as a function of position. The (printed) results are shown in Figure 60.

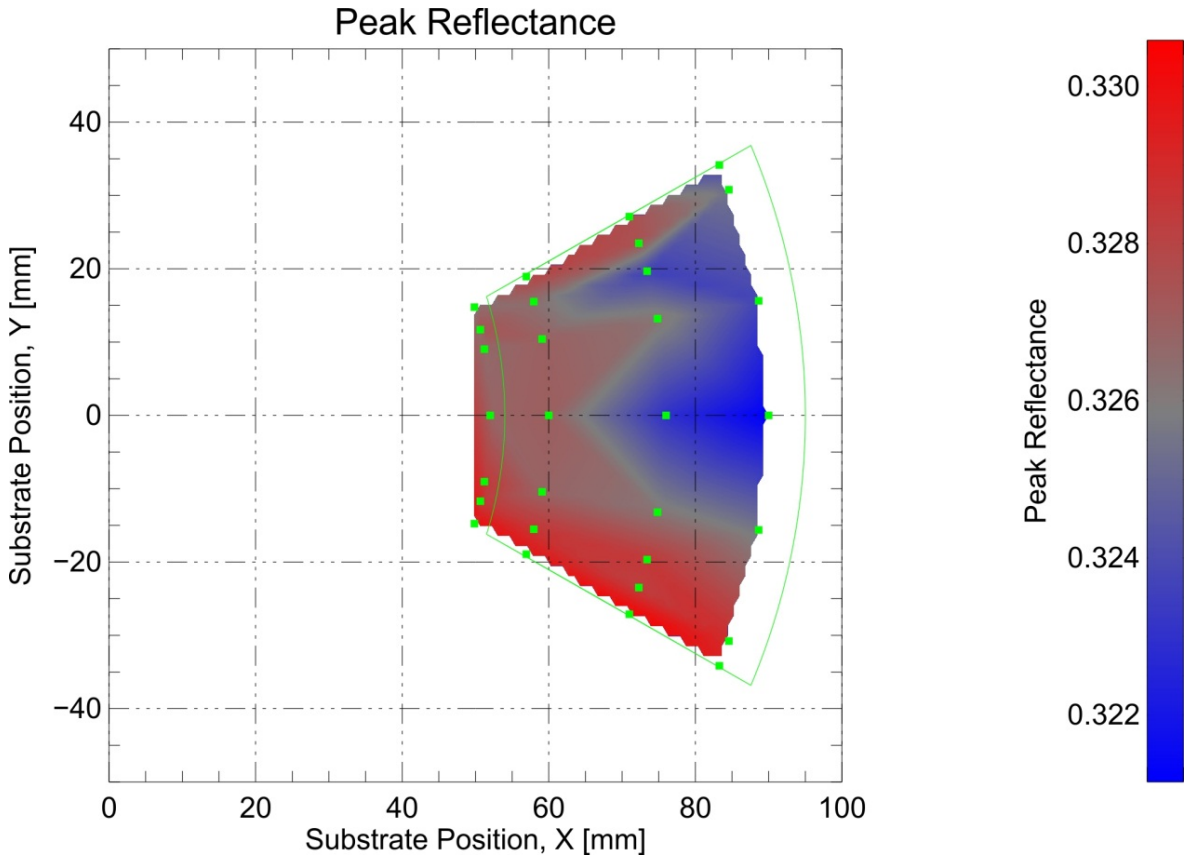


Figure 60. A plot of Peak Reflectance as a function of position on the mirror surface. This figure was generated by printing ("File→Print…") from the "Peak Reflectance Map" tab of the IMD*Multiplot window shown in Figure 54; the PostScript file that was generated was converted to JPEG (using CorelDraw) so that it could be imported into this document.

9.2.2.2 Example: IMD*Multiplot – Measured Performance over Time

Our final example of this chapter (and indeed the final example in this document - finally!) illustrates the use of IMD*Multiplot to visualize the evolution of coating performance over time, using a series of Reflectance vs. Wavelength measurements of (yet another) Mo/Y multilayer made over a period of ~7 months. The example files are located in "`imd/examples/IMD-Multiplot - Temporal Stability`", and the IMD*Multiplot in that directory is called "`Mo_Y_stability.imp`". (Don't forget to set your working directory appropriately before opening the .imp file so that it will find the data files.)

The IMD*Multiplot file "`Mo_Y_stability.imp`" contains 5 measured-data files, each containing measured reflectance vs. wavelength near normal incidence. The first tab in the IMD*Multiplot window is shown in Figure 61.

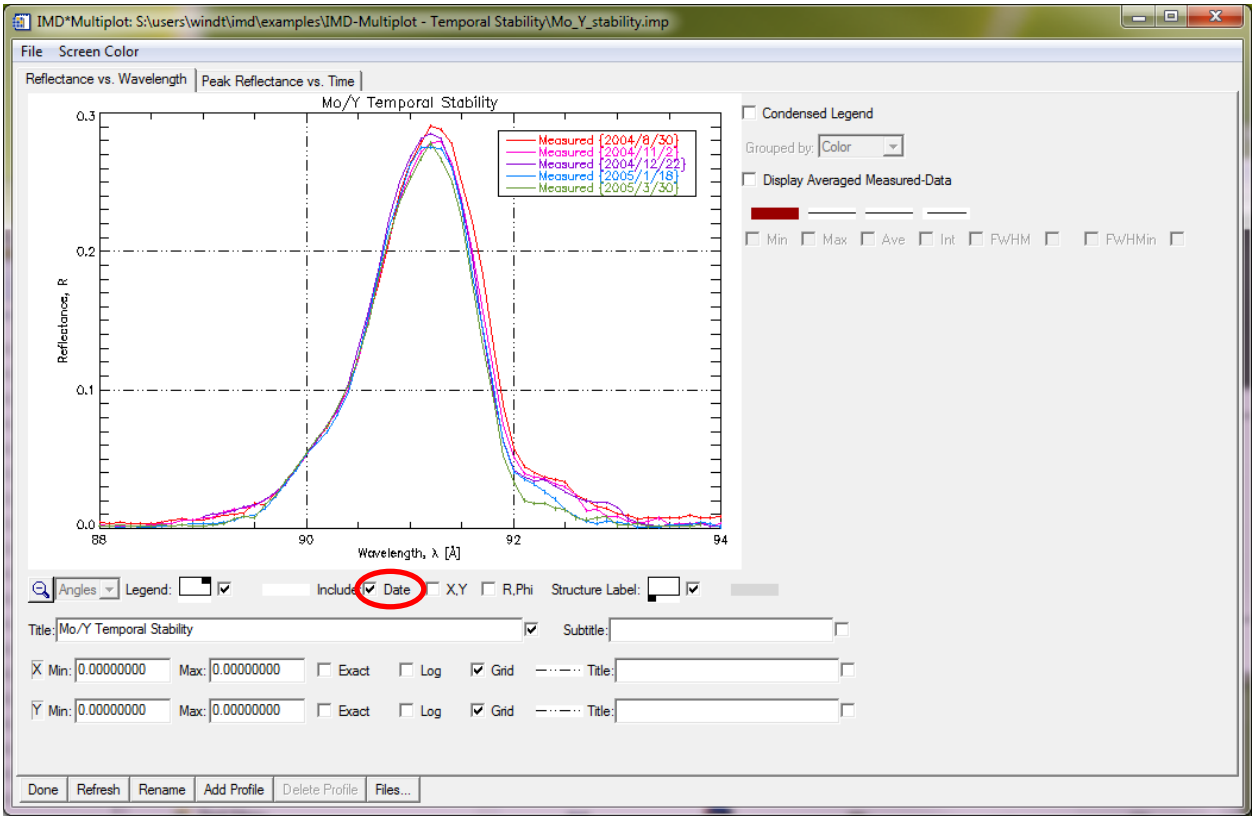
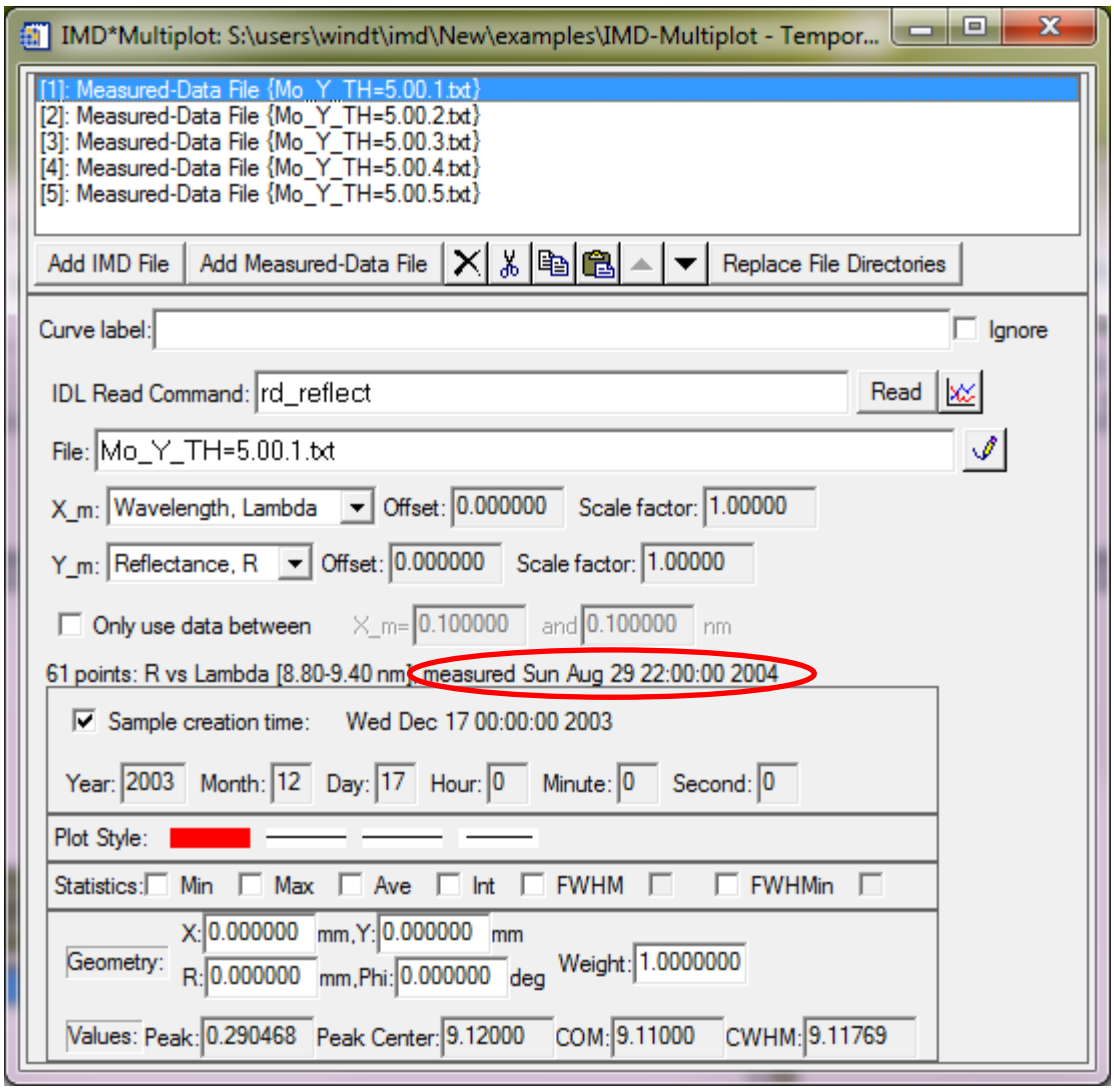


Figure 61. IMD*Multiplot window as it appears when the file "`imd/examples/IMD-Multiplot - Temporal Stability/Mo_Y_stability.imp`" is first opened.

The 5 measured-date files used in Figure 61 were created when the measurements took place, from Aug 2004 to Mar 2005. When IMD*Multiplot reads these files, it defines the date of the measurement as the file creation date that is provided by the operating system; the measurement date so defined is included in the plot legend when the box labeled "Date" is checked, as shown above, and is also listed on the "IMD*Multiplot - Files" window, as in the next figure. (The measurement position you specify can also be included in the legend by checking the box labeled "X,Y" or the box labeled "R,Phi".)



The second tab on the IMD*Multiplot window shown in Figure 61 is labeled "Peak Reflectance vs. Time", and shows a plot of the Normalized Peak Reflectance values derived from the five measured-data files vs. time in days:

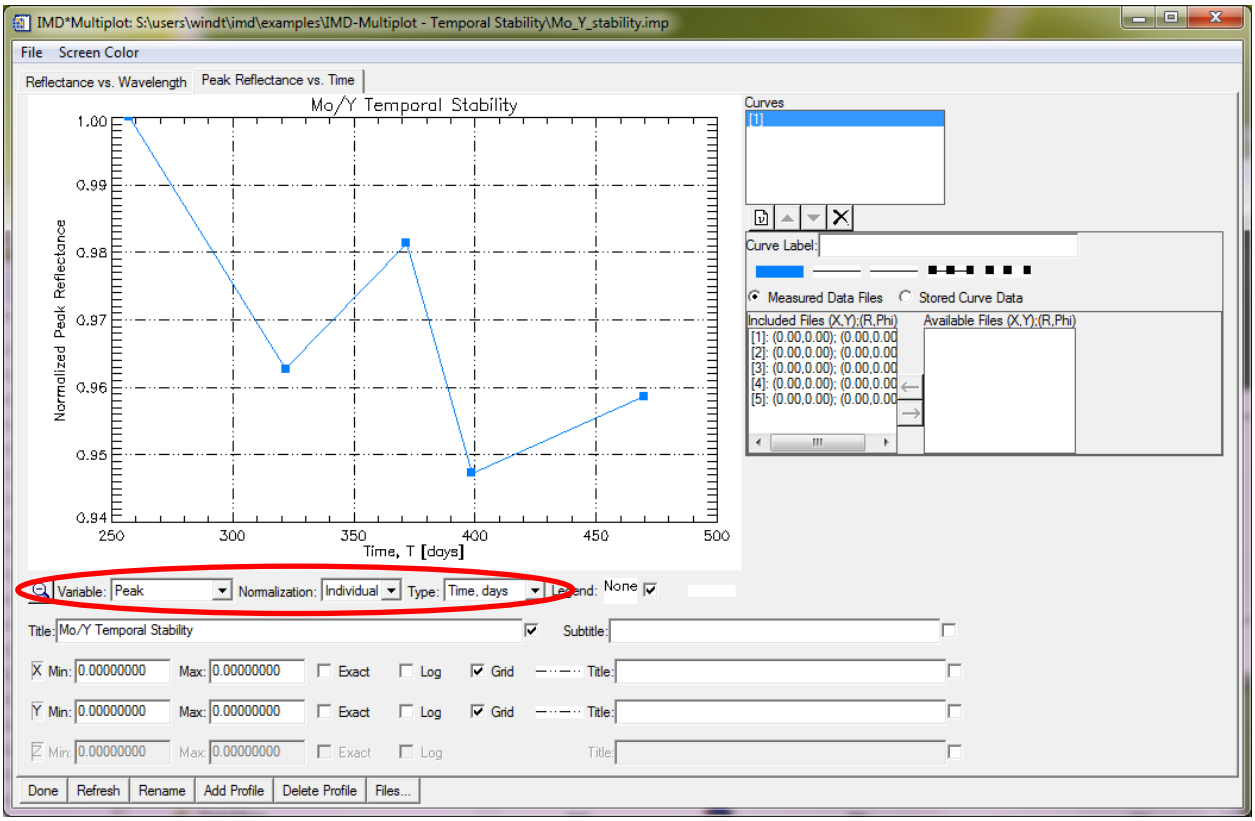
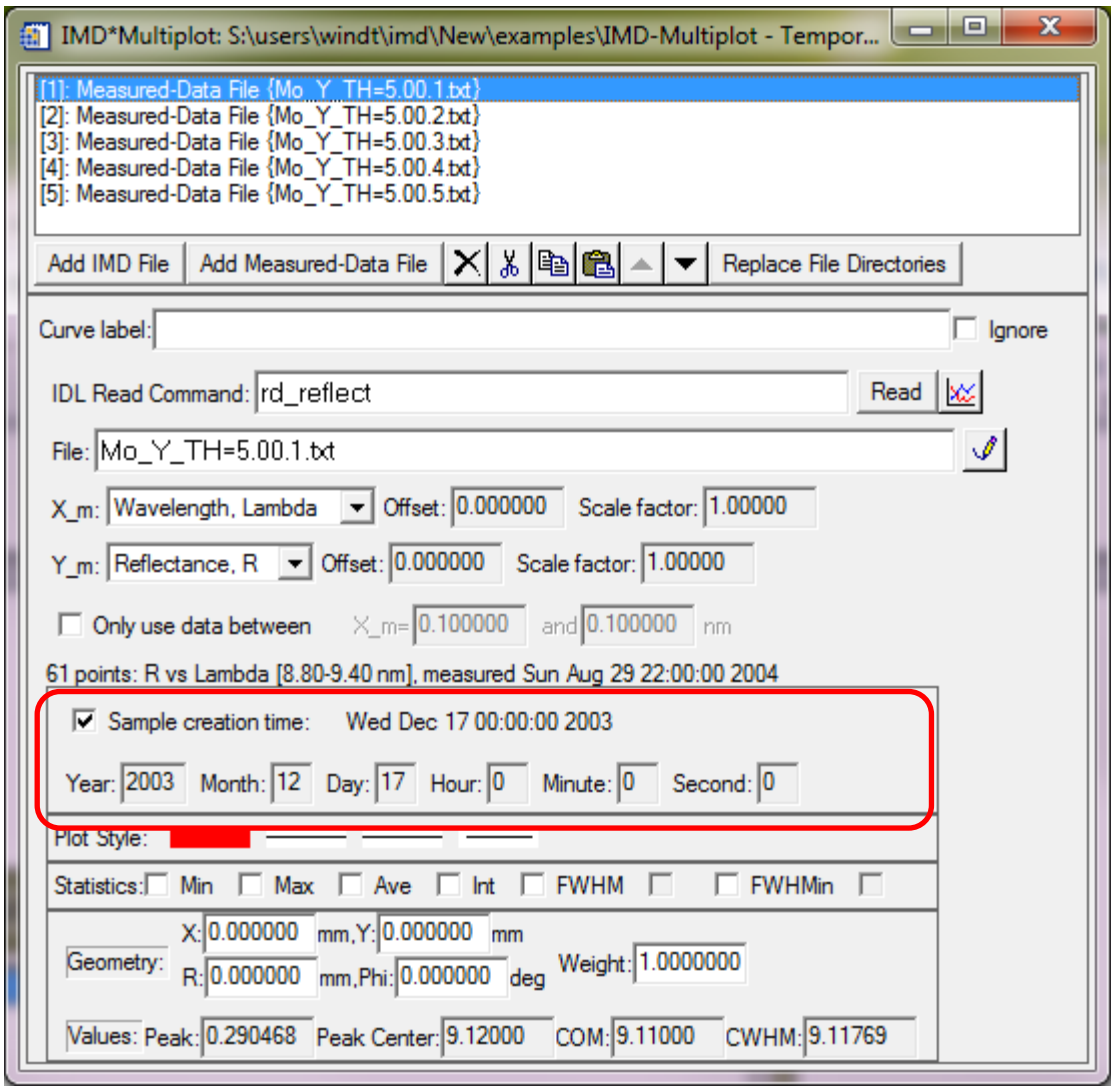


Figure 62. The second tab of the IMD*Multiplot window shown in Figure 61, displaying a plot of normalized peak reflectance vs sample age in days.

As can be seen in Figure 62, a single "Curve" was created for the plot, using the techniques explained in the previous section, and all 5 files are "included" in the Curve. The 5 values used for the independent variable in this plot – Time, in days – were derived from the measurement dates for each of the files, as just explained. (These particular data, which were obtained using a laser-plasma-source reflectometer, have much worse signal-to-noise than the ALS Beamline 6.3.2 data presented elsewhere in this document, and so there is considerable noise in the plot of peak reflectance vs. time shown in Figure 62.) By default, the measurement date of the first file in the list is used as "day 1", and then the relative dates of subsequent files included in the curve are computed from the difference in their measurement dates relative to that first date.

In this particular example, however, the first measured-data file in our list was created several months after the film was originally deposited; as we're interested in the relative change in peak reflectance as a function of the absolute age of the film (for this example, anyway), we've used the "Sample create time" area on the "IMD*Multiplot - Files" window in order to override the default behavior, and have entered the actual date that the film was deposited:



As a result of specifying the actual "sample creation time" as just described, the plot shown in Figure 62 displays time in days starting at a value of T=250 days, as the first measured-data file was created about that long after the sample was created.

9.2.2.3 IMD*Multiplot: Using "other" data

It's worth noting that IMD*Multiplot can be used with files containing any type of one-dimensional data – the data do not have to be "measured", nor do they even have to be optical data. After adding a "measured-data file" to the list, using the "Add Measured-Data File" button on the "IMD*Multiplot - Files" window, you could load, for example, files containing exported efficiency data computed in IMD*Efficiency, to compare the performance of several instrument configurations. Or you might want to load files containing exported averaged reflectance data computed in some other instance of IMD*Multiplot. Here's another possible use of "Measured-Data Files" in IMD*Multiplot: If you want to compare the results of many large IMD computations in IMD*Multiplot, since it's much faster to load

plain text files (i.e., Measured-Data Files) than it is to load .imd files, you might export each IMD result as a plain text file, and then load the plain text files into IMD*Multiplot as Measured-Data files in lieu of the original .imd files.

The Geometry settings for each Measured-Data File in IMD*Multiplot also can be used for other purposes than originally intended. That is, the values for (X, Y) or (R,Φ) that you can enter for each Measured-Data file in IMD*Multiplot are meant to correspond to measurement positions on an optical surface, in order to assess coating uniformity. But these quantities can represent any independent variable values you like. For example, perhaps you have measured a series of reflectance vs. wavelength curves obtained after heating a multilayer to different temperatures, and wish to visualize the evolution of that data as a function of annealing temperature. In that case you can load each measured data file into IMD*Multiplot and then enter the corresponding temperature values as X (say), so that you can create profiles of, e.g., Peak Reflectance or CWHM vs. Temperature (i.e., X). If the measured-data files don't contain IMD-type optical data, you can use the text-entry boxes on the main IMD*Multiplot window to specify whatever axis titles you like, and thus override the default plot titles generated automatically in IMD*Multiplot.

10 IMD Preferences

10.1 The IMD Preferences Window

The IMD Preferences window, shown below in Figure 63, is opened via the "File→Preferences…" menu option on the IMD menu bar. We've already seen the tabs located at the top of the window labeled "Modified Fresnel Coefficients" (§4.11.1.2), "Interface Profile Function Display" (§4.11.1.1), and "Power-Spectral-Density Function" (§4.11.2). Along with those tabs, the "General" tab visible in Figure 63 is used to control whether IMD generates default .imd file names derived from the structure you've defined, whether .imd files are saved automatically after every computation, and whether measured data is included in the .imd files. Note that these three particular settings are persistent, in that they do not get over-ridden when opening .imd files having different values for these settings. The display of alerts can be controlled on the "General" tab as well.

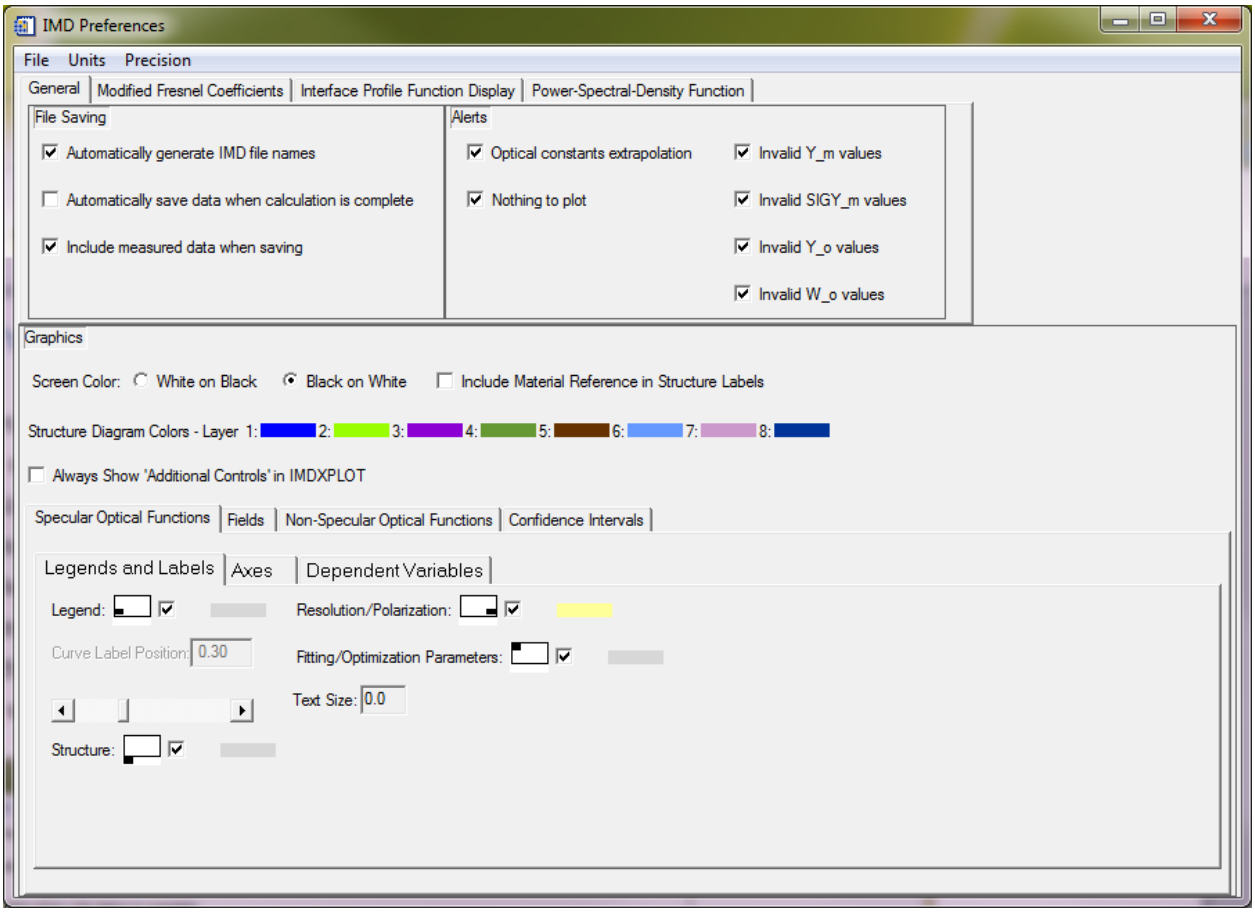


Figure 63. The IMD Preferences window.

The lower half of the IMD Preferences window contains controls for graphics display, in IMDXPLOT and elsewhere, with separate tabs for settings associated with Specular Optical Functions, Fields, Non-

Specular Optical Functions, and Confidence Intervals. Each tab in turn has sub-tabs where controls are segregated for Labels and Legends, Axes, and Dependent Variables.

The menu bar also provides options for setting units and precision of displayed numbers (i.e., by number of places to the right of the decimal point). Equivalent settings for units and precision can also be accessed through many other IMD windows, as we've seen previously.

10.2 Importing and Exporting IMD Preferences (.ipr) Files

The IMD menu bar options "File→Import Preferences and Settings…" and "File→Export Preferences and Settings…" allows you to use saved IMD preferences (.ipr) files containing the (non-persistent) settings values you have specified on the IMD Preferences window described in the previous section, as well as other settings, including those made on the Layer Thickness Profile window (§4.10), the Interface Width Profile window (§4.11.5), the Import Measured Data window (§8.1.1), the Import/Create Optimization Target Profile window (§8.3), the Fitting Settings and Optimization Settings windows (§8.6), and the Optical Constants and Atomic Scattering Factors Search Paths windows (Chapter 3). If you prefer to use settings different than the default settings that are defined when IMD starts fresh, you can use "File→Import Preferences and Settings…" to load a saved .ipr file to restore your preferred (non-persistent) settings that you had previously saved in that .ipr file using "File→Export Preferences and Settings…".

Keep in mind that whenever you open a .imd file in IMD – by using the "File→Open…" menu option on the IMD menu bar, by using the "File→Send Back to IMD" menu option from IMDXPLOT, or by using the "Open File in IMD" buttons () in IMD*Multiplot and IMD*Efficiency, all of your (non-persistent) preferences will be over-written by the settings contained in the .imd file that you open. As a convenience, once you have successfully imported a preferences file into IMD using the "File→Import Preferences and Settings…" menu option, you can use "File→Reload Last Imported Preferences and Settings…" to restore your imported preferences and settings. After you import preferences, you will have to repeat any computations that may have been performed prior to the import, however, as the imported preferences may include settings that affect computations (e.g., Modified Fresnel Coefficients, PSD functions, etc.)

10.3 Automatic IMD Preferences Import at Startup: IMD_PREFS_FILE

If you are using IMD as an IDL application, it is possible to automatically load a previously-saved IMD Preferences file when you first start IMD in a new IDL session. Start IDL as usual, but before starting IMD, define a variable called IMD_PREFS_FILE as a single-element string variable with the full name of the file, including the path. For example:

```
IDL> IMD_PREFS_FILE=!DIR+'/user_contrib/imd/examples/Preferences/sample_prefs_file.ipr'
```

Then start IMD as usual:

```
IDL> .r imd
```

The preferences file you specify will load automatically when IMD starts, and it will be available for "re-load" using the menu option described in the previous section. You can also define IMD_PREFS_FILE automatically in an IDL Startup File. Consult the IDL documentation for help configuring IDL Startup Files.

10.4 IMD and IMD Preferences Auto-Load Files: `imd_autoload.imd` and `imd_autoload.ipr`

When IMD first starts, and whenever the main IMD window is re-opened in the same session (i.e., by clicking the "IMD" button in IMD Launcher, or by typing "imd" at the IMD command prompt, after first exiting the main IMD window), the current working directory will be checked for the presence of any IMD and/or IMD Preferences files named, respectively, "imd_autoload.imd" and "imd_autoload.ipr" (case insensitive). If such "auto-load" files are found, they will be loaded automatically.

> **Note**: If both IMD and IMD Preferences auto-load files are found, the .imd file will be loaded first; the .ipr file will be loaded last, thereby over-writing the preferences and settings contained in the .imd file.

> **Note**: Any auto-load files found will be loaded in addition to, but *after* any startup IMD Preferences file that is loaded via the method described in §10.3; the auto-load files will thus over-write the preferences and settings contained in the startup IMD Preferences file.

11 IMD for IDL Programmers

11.1 IMD Common Block Variables

If you are using IMD as an IDL application, then you will have started IMD at the IDL command line, using ".run imd", as explained in Chapter 2. IMD uses a number of IDL common blocks, which are defined and accessible at the command line when you run `imd.pro` this way. The majority of these common block variables should not be used directly, but you need to know their names in order to avoid inadvertent problems, e.g., by using variables having the same names in your own IDL commands and programs.

IDL common block variables are defined in the file `imd_com.pro` in the imd installation directory. Please consult that file for the complete list of IMD common block variables. In §11.1.1 we describe select IMD common block variables that may be of interest to you directly for your own programs, or that can be used to define user-defined specular optical functions in IMD, as explained in §5.1.1.1.

11.1.1 Dependent and Independent Variable Names

```
;  THETA - Incidence angles, in degrees from normal.
;
;  LAMBDA - Wavelengths, in angstroms.
;
;  IDVAR - A structure containing independent variable arrays.
;          Each tag of IDVAR is an array. i.e., IDVAR.IV0 = Theta,
;          IDVAR.IV1 = Lambda, etc.
;
;  FITPAR - Similar to IDVAR, but for Fit Parameters.
;
;  CHI_2 - Array of Chi^2 values.  Dimensions are given by the
;          number of tags to FITPAR
;
;  NC - Complex array of indices of refraction.  Dimensions are equal to
;       N_ELEMENTS(Z)+2 by N_ELEMENTS(LAMBDA)
;
;  Z - 1D array of layer thicknesses, in angstroms.
;
;  SIGMA - 1D array of interface roughnesses.
;
;  SIGMA_FROM_PSD - 1D array of flags indicating that the SIGMA value
;                   for that interface is to be computed from the PSD.
;
;  INTERFACE - 1D array of interface profile pointers
;
;  F - Polarization factor
;
;  Q - Polarization analyzer sensitivity
;
;  ARES - Angular resolution, in degrees.
;
;  PRES - Spectral resolution, in angstroms.
;
;  R, RS, RP - Reflectance
;
;  PRS, PRP - Reflected phases.
;
;  PSI, DELTA - Ellipsometric Psi and Delta.
;
;  T, TS, TP - Transmittance
```

```
;
;  PTS, PTP - Transmitted phases
;
;  A, AS, AP - Absorptance
;
;  USER1, USER2, USER3 - User-defined functions.
;
;   Note: the dimensions of the optical properties are determined by the
;   number of independent variables.


;  DEPTH - Depth array (for field intensities)
;
;  INT, INTS, INTP - Field intensities.
;
;   Note: the dimensions of the field intensities are determined by the
;   number of independent variables.


;   THETA_IN - Multi-dimensional array of Theta_in values, in degress
;              from normal.
;
;   THETA_OUT - Multi-dimensional array of Theta_out values, in
;               degress from normal.
;
;   PHI - Scattering plane azimuthal angle, in degrees.
;
;   PSD_XI_PERP - Perpendicular correlation length, used with the
;                 sigma_r/xi_par/H PSD model.
;
;   SIGMA_D - 1D array of interface diffuseness values.
;
;   PSD_SIGMA_R - 1D array of interface roughnesses.
;
;   PSD_XI_PAR - 1D array of interface correlation lengths.
;
;   PSD_H - 1D array of interface jaggedness factors.
;
;   PSD_OMEGA - 1D array of interface volume element parameters.
;
;   PSD_NU - 1D array of interface relaxation parameters.
;
;   PSD_N - 1D array of interface PSD exponents.
;
;   DIDOMEGA_DWBA - Non-specular reflected intensity computed from the
;                   DWBA
;
;   DIDOMEGA_BAS - s-polarized component of non-specular reflected
;                  intensity computed using the BA.
;
;   DIDOMEGA_BAP - p-polarized component of non-specular reflected
;                  intensity computed using the BA.
;
;   DIDOMEGA_BAA - polarization-averaged non-specular reflected
;                  intensity computed using the BA.
```

11.2 Computational Routines

This section provides documentation for the core computational routines used by IMD to calculate specular and non-specular optical functions. These routines can be used directly in your own IDL programs, following the instructions provided here.

11.2.1 FRESNEL

```
; NAME:
;
;       FRESNEL
;
; PURPOSE:
;
;       Compute specular optical functions and electric
;       field intensity for a multilayer stack.
;
; CALLING SEQUENCE:
;
;       FRESNEL, THETA,LAMBDA,NC,Z[,SIGMA,INTERFACE,F,Q]
;
; INPUTS:
;
;       THETA - Scalar or 1-D array of incidence angles, in degrees,
;               measured from the normal.
;
;       LAMBDA - Scalar or 1-D array of wavelengths.  Units for LAMBDA
;                are the same as for Z and SIGMA.
;
;       NC - Complex array of optical constants.  The dimensions of NC
;            must be (N_ELEMENTS(Z)+2,N_ELEMENTS(LAMBDA)).
;
;       Z - 1-D array of layer thicknesses. Units for Z are the same
;           as for SIGMA and LAMBDA.
;
;
; OPTIONAL INPUTS:
;
;
;       SIGMA - Scalar, 1D or 3D array of interface widths.  If SIGMA
;               is a scalar, then the same roughness value is applied
;               to each interface.  If SIGMA is a 1-D array, it must
;               have (N_ELEMENTS(Z)+1) elements, corresponding to the
;               number of interfaces in the stack.  If SIGMA is a 3-D
;               array, it must have
;               (N_ELEMENTS(THETA),N_ELEMENTS(LAMBDA),N_ELEMENTS(Z)+1)
;               elements. Units for SIGMA are the same as for LAMBDA
;               and Z.
;
;       INTERFACE - scalar, 1-D array, or 3-D array of interface
;                   roughness profile types.
;
;                   INTERFACE=0 corresponds to an error-function
;                   interface profile;
;
;                   INTERFACE=1 corresponds to an exponential
;                   interface profile;
;
;                   INTERFACE=2 corresponds to a linear interface
;                   profile;
;
;                   INTERFACE=3 corresponds to a sinusoidal interface
;                   profile;
;
;                   INTERFACE=4 corresponds to a step function
;                   interface.

;                   If INTERFACE is a scalar, then the same roughness
;                   profile is applied to each interface.  If
;                   INTERFACE is a 1-D array, it must have
;                   (N_ELEMENTS(Z)+1) elements, corresponding to the
;                   number of interfaces in the stack.  If INTERFACE
;                   is a 3-D array, it must have
;                   (N_ELEMENTS(THETA),N_ELEMENTS(LAMBDA),N_ELEMENTS(Z)+1)
;                   elements.
;
;       F - Incident polarization factor, defined as the ratio
```

```
;           (I(s)-I(p)) / (I(s)+I(p)), where I(s) and I(p) are the
;           incident intensities of s and p polarizations. So for pure
;           s-polarization specify F=+1; for pure p-polarization
;           specify F=-1, or for unpolarized incident radiation,
;           specify F=0.
;
;       Q - Polarization analyzer sensitivity, defined as the
;           sensitivity to s-polarization divided by the sensitivity
;           to p-polarization.  Specifying a value of q other than 1.0
;           could be used to simulate, for example, the reflectance
;           you would measure using a detector that (for whatever
;           reason) was more or less sensitive to s-polarization than
;           to p-polarization.
;
;           NOTE: The values of F and Q that you specify only affect
;           the 'average' optical properties; the optical properties
;           for pure s and pure p polarization are unaffected.
;
; KEYWORD PARAMETERS:
;
;       INPUTS:
;
;              MFC_MODEL - an integer specifying the form of the modified
;                          Fresnel coefficients to be used to account for
;                          interface imperfections.
;
;              MFC_MODEL=0,1, or 2, corresponding (in order) to
;              the options listed on the IMD Preferences window.
;
;              COMPUTE_TA - Set to compute reflectance, transmittance
;                           and absorptance.
;
;              COMPUTE_FIELD - Set to compute fields, reflectance,
;                              transmittance and absorptance.
;
;              Note: if neither COMPUTE_TA nor COMPUTE_FIELD is set,
;              then only reflectance is computed.
;
;
;              ARES - Instrumental angular resolution, in degrees. The
;                     optical properties (R, T, A) vs. angle will be
;                     convolved with a Gaussian of width ARES, in
;                     order to simulate a finite instrumental
;                     resolution.  You must specify three or more
;                     THETA values to use ARES.
;
;
;              WRES - Instrumental spectral resolution, in the same
;                     units as LAMBDA. The optical properties (R, T,
;                     A) vs. wavelength will be convolved with a
;                     Gaussian of width PRES, in order to simulate a
;                     finite instrumental resolution. You must specify
;                     three or more LAMBDA values to use PRES.
;
;       INPUTS FOR FIELD COMPUTATIONS ONLY (i.e., with COMPUTE_FIELD set):
;
;              PT_SPACING - The spacing (in same units as Z) between
;                           field intensity points.  If not specified,
;                           default value of 1 is used.
;
;                           Note that the actual spacing between
;                           points will not be regular, as it is
;                           necessary to compute the field intensities
;                           precisely at the interfaces between
;                           layers.  See DEPTH below.

;
;              AMBIENT_DEPTH - The depth (in same units as Z) into the
;                              ambient material (the material above
```

```
;                           the multilayer) for which the field
;                           intensities are to be calculated.
;                           If not specified, a default value of 0
;                           is used.
;
;
;              SUBSTRATE_DEPTH - The depth (in same units as Z) into
;                                the substrate material (the material
;                                below the multilayer) for which the
;                                field intensities are to be
;                                calculated.  If not specified, a
;                                default value of 0 is used.
;
;       OUTPUTS:
;
;              RS - Reflectance for s-polarization.
;
;              RP - Reflectance for p-polarization.
;
;              RA - Average reflectance, =
;                   (RS*Q*(1+F)+RP*(1-F))/(F*(Q-1)+(Q+1))
;
;              PRS - Phase of reflected wave for s-polarization.
;
;              PRP - Phase of reflected wave for p-polarization.
;
;              CRS - Complex reflected amplitude for s-polarization.
;
;              CRP - Complex reflected amplitude for p-polarization.
;
;              RHO - Ratio of CRP/CRS
;
;              PSI - Ellipsometric Psi function = ATAN(RHO*CONJ(RHO))
;
;              DELTA - Ellipsometric Delta function =
;                      ATAN(IMAGINARY(RHO),DOUBLE(RHO))
;
;              FUNC_USER1,
;
;              FUNC_USER2,
;
;              FUNC_USER3 - Strings specifying the USER1, USER2 and
;                           USER3 functions.  The IDL EXECUTE function
;                           is used to compute these functions, as in:
;                           ee=EXECUTE('user1='+func_user1).  Thus,
;                           these strings must use valid IDL syntax,
;                           and must be valid functions of any of the
;                           variables and functions internal to the
;                           FRESNEL procedure.
;
;              USER1,
;
;              USER2,
;
;              USER3 - The user functions determined by the values of
;                      associated FUNC_USER* strings described above.
;
;
;       ADDITIONAL OUTPUTS when COMPUTE_TA or COMPUTE_FIELD keywords are set:
;
;              TS - Transmittance for s-polarization.
;
;              TP - Transmittance for p-polarization.
;
;              TA - Average transmittance, =
;                   (TS*Q*(1+F)+TP*(1-F))/(F*(Q-1)+(Q+1))
;
;              PTS - Phase of transmitted wave for s-polarization.
;
```

```
;              PTP - Phase of transmitted wave for p-polarization.
;
;              AS - Absorptance for s-polarization.
;
;              AP - Absorptance for p-polarization.
;
;              AA - Average absorptance, =
;                   (AS*Q*(1+F)+AP*(1-F))/(F*(Q-1)+(Q+1))
;
;        The dimensions of RS, RP, RA, PRS, PRP, TS, TP, TA, PTS, PTP,
;        AS, AP, and AA are (N_ELEMENTS(THETA),N_ELEMENTS(LAMBDA).
;
;        ADDITIONAL OUTPUTS WHEN COMPUTE_FIELD IS SET:
;
;              DEPTH - 1-D array of depth values. The number of depth
;                      values is
;
;                      1+FIX(TOTAL(Z)+AMBIENT_DEPTH+SUBSTRATE_DEPTH)/PT_SPACING
;
;                      The depth values are not equally spaced, as it
;                      is necessary to calculate the field intensities
;                      precisely at the interfaces between layers.
;                      If you need an equally spaced DEPTH array,
;                      you can interpolate the DEPTH, and INT values.
;
;             INTS - Field intensity for s-polarization.
;
;                     Dimensions for INTS, INTP, and INTA are
;                     (N_ELEMENTS(DEPTH),N_ELEMENTS(THETA),N_ELEMENTS(LAMBDA))
;
;             INTP - Field intensity for p-polarization.
;
;             INTS - Average field intensity,
;                   =(INTS*Q*(1+F)+INTP*(1-F))/(F*(Q-1)+(Q+1))
;
;             FPS  - Electric field amplitude, positive-going
;                     (i.e., towards the substrate), s-polarization.
;
;             FMS  - Electric field amplitude, negative-going
;                     (i.e., away from the substrate), s-polarization.
;
;             FPPN - Electric field amplitude, positive-going
;                     (i.e., towards the substrate), p-polarization, normal component
;
;             FMPN - Electric field amplitude, negative-going
;                     (i.e., away from the substrate), p-polarization, normal component
;
;             FPPT - Electric field amplitude, positive-going
;                     (i.e., towards the substrate), p-polarization, tangential component
;
;             FMPT - Electric field amplitude, negative-going
;                     (i.e., away from the substrate), p-polarization, tangential component
;
;                     Dimensions for FPS, FMS, FPPN, FMPN, FPPT and FMPT are
;                     (N_ELEMENTS(DEPTH),N_ELEMENTS(THETA),N_ELEMENTS(LAMBDA))
;
;
; COMMON BLOCKS:
;
;       COMMON IMD,IMD
;
; PROCEDURE:
;
;       As the name suggests, this procedure uses the Fresnel
;       equations to compute the optical properties of a multilayer as
;       a function of angle and wavelength.  All computations are
;       performed in double precision.  A straightforward recursion
;       algorithm is used, which is to say that no shortcuts are
;       employed to take advantage of situations in which the
```

```
;       multilayer is purely periodic.  As such, other algorithms
;       might be faster, but this one is more general.
;
;       If the COMPUTE_TA keyword is set, to compute transmittance and
;       absorptance in addition to reflectance, then the procedure is
;       significantly slower, due to the extra computation required.
;       If the COMPUTE_FIELD keyword is set, the computation time is
;       longer still.
;
;       Scattering losses due to interfacial roughness or diffuseness
;       are handled according to the methods described in section 4.11
;       of IMD.pdf.
;
;       Polarization averaging is performed using the function SP2A,
;       usage:
;
;       Result=SP2A(XS,XP,F,Q),
;
;       where F and Q are as defined above, XS and XP are the optical
;       functions for pure s and p polarizations, and Result is the
;       average function. For example, the average reflectance RA is
;       given by:
;
;       RA=(RS*Q*(1+F)+RP*(1-F))/(F*(Q-1)+(Q+1))
;
;       Instrumental resolution is computed by convolution with a
;       Gaussian, using the function INSTRUMENT_RES, usage:
;
;       Result=INSTRUMENT_RES(X,Y,RES,ANGLE=ANGLE,
;                             WAVELENGTH=WAVELENGTH,ENERGY=ENERGY),
;
;       where RES is the width of the Gaussian, and X and Y are the
;       independent and dependent variables (e.g., X=THETA, Y=RA).
;       Set ANGLE to convolve reflectance, transmittance, or
;       absorptance vs. angle, WAVELENGTH to convolve r/t/a vs.
;       wavelength, or ENERGY, to convolve r/t/a vs. energy. In the
;       case of ENERGY, both X and RES must be specified in angstroms,
;       but are converted to eV before doing the convolution.
;
; EXAMPLE:
;
;       Compute the reflectance of a gold film, 500 A thick, vs angle,
;       at a wavelength of 4000 A:
;
;       Define THETA=VECTOR(0.,90.,91), an array of angles from 0 to 90
;       degrees, in 1 degree steps.
;
;       Define LAMBDA=4000.
;
;       Using the optical constants for Au at 4000 A, we define
;       NC=[complex(1.,0.),complex(1.658,1.956),complex(1.,0.)],
;       assuming the Au film is surround by vacuum on either side.
;
;       Define Z=[500.]
;
;       FRESNEL, THETA, LAMBDA, NC, Z, RA=RA
;
;       The value for RA we compute in this example corresponds to
;       unpolarized light.
;
; MODIFICATION HISTORY:
;
;       David L. Windt, Bell Labs, April, 1998
;-
```

11.2.2 NS_BA

```
;+
```

```
; NAME:
;
;       NS_BA
;
; PURPOSE:
;
;       Procedure to compute the non-specular reflected intensity for
;       a multilayer film, using the Born approximation theory
;       developed by D. Stearns, as described in reference [9].
;
; CALLING SEQUENCE:
;
;       NS_BA, THETA_IN,THETA_OUT,LAMBDA,NC,Z [,F,Q]
;
; INPUTS:
;
;       THETA_IN - Scalar or 1-D array of incidence angles, in
;                  degrees, measured from the normal.
;
;       THETA_OUT - Scalar or 1-D array of scattering angles, in
;                   degrees, measured from the normal.
;
;       If THETA_IN is a scalar, then the same value of THETA_IN will
;       be used for every value of THETA_OUT.  Similarly, if THETA_OUT
;       is a scalar, then the same value of THETA_OUT will be used for
;       every value of THETA_IN.  If THETA_IN and THETA_OUT are both
;       1D arrays, then they must have the same number of elements.
;       In any case, the non-specular reflected intensity is computed
;       for pairs of THETA_IN, THETA_OUT values.
;
;       LAMBDA - Wavelength (scalar).
;
;       NC - Complex array of optical constants.  The dimensions of NC
;            must be (N_ELEMENTS(Z)+2)
;
;       Z - 1-D array of layer thicknesses. Units for Z are the same
;           as for SIGMA and LAMBDA.
;
; OPTIONAL INPUTS:
;
;
;       F - Incident polarization factor, defined as the ratio
;           (I(s)-I(p)) / (I(s)+I(p)), where I(s) and I(p) are the
;           incident intensities of s and p polarizations. So for pure
;           s-polarization specify F=+1; for pure p-polarization
;           specify F=-1, or for unpolarized incident radiation,
;           specify F=0.
;
;       Q - Polarization analyzer sensitivity, defined as the
;           sensitivity to s-polarization divided by the sensitivity
;           to p-polarization.  Specifying a value of q other than 1.0
;           could be used to simulate, for example, the reflectance
;           you would measure using a detector that (for whatever
;           reason) was more or less sensitive to s-polarization than
;           to p-polarization.
;
; KEYWORD PARAMETERS:
;
;       INPUTS:
;
;       PHI - Scattering plane azimuthal angle. (Scalar.) Default is 0 deg.
;
;       MFC_MODEL - an integer specifying the form of the modified
;                   Fresnel coefficients to be used to account for
;                   interface imperfections.
;
;                   MFC_MODEL=0,1, or 2, corresponding (in order) to
;                   the options listed on the IMD Preferences window.
;
```

```
;       PSD_MODEL - Set PSD_MODEL=0 to use the sigma_r/xi_par/H PSD model.
;                   Set PSD_MODEL=1 to use the Omega/nu/n PSD model.
;
;
;       If PSD_MODEL=0, then the following three parameters must be
;       supplied:
;
;       SIGMA_R - 1D array of interface roughnesses. N_ELEMENTS(Z)+1
;                 elements.
;
;       XI_PAR - 1D array of interface correlation lengths. N_ELEMENTS(Z)+1
;                 elements.
;
;       H - 1D array of interface jaggedness factors. N_ELEMENTS(Z)+1
;           elements.
;
;       In addition, you can also specify:
;
;       XI_PERP - Perpendicular correlation length (scalar).
;
;
;       If PSD_MODEL=1, then the following three parameters must be
;       supplied:
;
;       OMEGA - 1D array of interface volume
;               elements. N_ELEMENTS(Z)+1 elements.
;
;       NU - 1D array of interface relaxation
;            coefficients. N_ELEMENTS(Z)+1 elements.
;
;       N_ - 1D array of PSD exponents. N_ELEMENTS(Z)+1 elements.
;
;       You can also specify:
;
;       SUBSTRATE_Z - The Z value of the 'substrate', needed to
;                     compute the PSD of the bottom-most interface
;                     when using the Omega/nu/n PSD model.
;
;       Diffuse interfaces can be used when computing the electric
;       field intensities (needed to compute the scattered intensity)
;       by specifying values for SIGMA_D and INTERFACE.  The electric
;       fields will be computed from modified Fresnel coefficients.
;
;              SIGMA_D - 1D array of interface diffuseness factors.
;                        SIGMA_D must have N_ELEMENTS(Z)+1 elements.
;
;
;              INTERFACE - 1D array of interface profiles
;                          functions. INTERFACE must have
;                          N_ELEMENTS(Z)+1 elements.
;
;                          INTERFACE=0 corresponds to an
;                          error-function interface profile;
;
;                          INTERFACE=1 corresponds to an exponential
;                          interface profile;
;
;                          INTERFACE=2 corresponds to a linear
;                          interface profile;
;
;                          INTERFACE=3 corresponds to a sinusoidal
;                          interface profile;
;
;                          INTERFACE=4 corresponds to a step function
;                          interface.
;
;
;
;              ARES - Instrumental angular resolution, in degrees. The
```

```
;                     non-specular reflected intensity vs. angle will
;                     be convolved with a Gaussian of width ARES, in
;                     order to simulate a finite instrumental
;                     resolution.  You must specify three or more
;                     THETA_IN or THETA_OUT values to use ARES.
;
;       OUTPUTS:
;
;              DIDOMEGA_S - s-polarized component of reflected
;                           intensity.
;
;              DIDOMEGA_P - p-polarized component of reflected
;                           intensity.
;
;              DIDOMEGA_A - polarization-averaged reflected intensity.
;
;              See section 5.2.2 for a complete description of how
;              these three functions depend on f and q.
;
; COMMON BLOCKS:
;
;       COMMON IMD,IMD
;
; PROCEDURE:
;
;       Instrumental resolution is computed by convolution with a
;       Gaussian, using the function INSTRUMENT_RES, usage:
;
;       Result=INSTRUMENT_RES(X,Y,RES,/ANGLE)
;
;       where RES is the width of the Gaussian, and X and Y are the
;       independent and dependent variables (e.g., X=THETA,
;       Y=DIDOMEGA_S).  Set ANGLE to convolve didomega vs. angle.
;
; MODIFICATION HISTORY:
;
;       David L. Windt, Bell Labs, April, 1998
;-
```

11.2.3 NS_DWBA

```
;+
; NAME:
;
;       NS_DWBA
;
; PURPOSE:
;
;       Procedure to compute the non-specular reflected intensity for
;       a multilayer film, using the distorted-wave Born approximation
;       theory developed by Sinha/Holy/de Boer, described in
;       references [10, 11, 12, and 13].
;
; CALLING SEQUENCE:
;
;       NS_DWBA, THETA_IN,THETA_OUT,LAMBDA,NC,Z
;
; INPUTS:
;
;       THETA_IN - Scalar or 1-D array of incidence angles, in
;                  degrees, measured from the normal.
;
;       THETA_OUT - Scalar or 1-D array of scattering angles, in
;                   degrees, measured from the normal.
;
;       If THETA_IN is a scalar, then the same value of THETA_IN will
```

```
;       be used for every value of THETA_OUT.  Similarly, if THETA_OUT
;       is a scalar, then the same value of THETA_OUT will be used for
;       every value of THETA_IN.  If THETA_IN and THETA_OUT are both
;       1D arrays, then they must have the same number of elements.
;       In any case, the non-specular reflected intensity is computed
;       for pairs of THETA_IN, THETA_OUT values.
;
;       LAMBDA - Wavelength (scalar).
;
;       NC - Complex array of optical constants.  The dimensions of NC
;            must be (N_ELEMENTS(Z)+2)
;
;       Z - 1-D array of layer thicknesses. Units for Z are the same
;           as for SIGMA and LAMBDA.
;
; KEYWORD PARAMETERS:
;
;       INPUTS:
;
;       PHI - Scattering plane azimuthal angle. (Scalar.) Default is 0.
;
;       MFC_MODEL - an integer specifying the form of the modified
;                   Fresnel coefficients to be used to account for
;                   interface imperfections.
;
;       MFC_MODEL=0,1, or 2, corresponding (in order) to
;       the options listed on the IMD Preferences window.
;
;       PSD_MODEL - Set PSD_MODEL=0 to use the sigma_r/xi_par/H PSD model.
;                   Set PSD_MODEL=1 to use the Omega/nu/n PSD model.
;
;
;       If PSD_MODEL=0, then the following three parameters must be
;       supplied:
;
;       SIGMA_R - 1D array of interface roughnesses. N_ELEMENTS(Z)+1
;                 elements.
;
;       XI_PAR - 1D array of interface correlation lengths. N_ELEMENTS(Z)+1
;                 elements.
;
;       H - 1D array of interface jaggedness factors. N_ELEMENTS(Z)+1
;           elements.
;
;       In addition, you can also specify:
;
;       XI_PERP - Perpendicular correlation length (scalar).
;
;
;       If PSD_MODEL=1, then the following three parameters must be
;       supplied:
;
;       OMEGA - 1D array of interface volume
;               elements. N_ELEMENTS(Z)+1 elements.
;
;       NU - 1D array of interface relaxation
;            coefficients. N_ELEMENTS(Z)+1 elements.
;
;       N_ - 1D array of PSD exponents. N_ELEMENTS(Z)+1 elements.
;
;       You can also specify:
;
;       SUBSTRATE_Z - The Z value of the 'substrate', needed to
;                     compute the PSD of the bottom-most interface
;                     when using the Omega/nu/n PSD model.
;
;       Diffuse interfaces can be used when computing the electric
;       field intensities (needed to compute the scattered intensity)
;       by specifying values for SIGMA_D and INTERFACE. The electric
```

```
;       fields will be computed from modified Fresnel coefficients.
;
;              SIGMA_D - 1D array of interface diffuseness factors.
;                        SIGMA_D must have N_ELEMENTS(Z)+1 elements.
;
;
;              INTERFACE - 1D array of interface profiles
;                          functions. INTERFACE must have
;                          N_ELEMENTS(Z)+1 elements.
;
;                          INTERFACE=0 corresponds to an
;                          error-function interface profile;
;
;                          INTERFACE=1 corresponds to an exponential
;                          interface profile;
;
;                          INTERFACE=2 corresponds to a linear
;                          interface profile;
;
;                          INTERFACE=3 corresponds to a sinusoidal
;                          interface profile;
;
;                          INTERFACE=4 corresponds to a step function
;                          interface.
;
;              ARES - Instrumental angular resolution, in degrees. The
;                     non-specular reflected intensity vs. angle will
;                     be convolved with a Gaussian of width ARES, in
;                     order to simulate a finite instrumental
;                     resolution.  You must specify three or more
;                     THETA_IN or THETA_OUT values to use ARES.
;       OUTPUTS:
;
;              DIDOMEGA - non-specular reflected intensity.
;
; COMMON BLOCKS:
;
;       COMMON IMD,IMD
;
; PROCEDURE:
;
;       Instrumental resolution is computed by convolution with a
;       Gaussian, using the function INSTRUMENT_RES, usage:
;
;       Result=INSTRUMENT_RES(X,Y,RES,/ANGLE)
;
;       where RES is the width of the Gaussian, and X and Y are the
;       independent and dependent variables (e.g., X=THETA,
;       Y=DIDOMEGA_S).  Set ANGLE to convolve didomega vs. angle.
;
; MODIFICATION HISTORY:
;
;       David L. Windt, Bell Labs, April, 1998
;
;-
```

11.3 Optical Constants Routines

11.3.1 IMD_F1F2TONK

```
;+
; NAME:
;
;           IMD_F1F2TONK
;
; PURPOSE:
;
```

```
;           This function computes optical constants (n,k) from
;           atomic scattering factors for a compound specified
;           by its composition and density.
;
; CALLING SEQUENCE:
;
;           Result=IMD_F1F2TONK(DENSITY,NUMBER,ELEMENT[,WAVELENGTH])
;
;
; INPUT PARAMETERS:
;
;           DENSITY - compound density, in gm/cm3.
;
;           NUMBER - scalar or array of relative concentrations. For example,
;                    to compute the optical constants for Al2O3, set
;                    NUMBER=[2.,3.]
;
;           ELEMENT - scalar or array of chemical element symbols. For example,
;                     to compute the optical constants for Al2O3, set
;                     ELEMENT=['Al','O'].
;
;           WAVELENGTH - vector of wavelengths.
;
;           If wavelength is omitted, the energy vectors contained in each atomic
;           scattering factor file, corresponding to the elements specified, will be
;           read; the energy vectors will be combined, and then only the unique values
;           will be retained and sorted to form the wavelength vector used for the
;           the computation. This will ensure that no energy resolution, e.g., near
;           sharp absorption features, will be lost when computed n and k.
;
; OUTPUTS:
;
;       Result - A complex scalar or 1-D array (the same length
;                as WAVELENGTH) containing the optical constants (n,k).
;
; KEYWORD PARAMETERS:
;
;     F1F2 - optional input or output array of atomic scattering factors.
;            Dimensions=[2,N_ELEMENTS(NUMBER),N_ELEMENTS(WAVELENGTH)]; Thus
;            F1F2(0,0,*) are the F1 values for the first element, and
;            F1F2(1,0,*) are the F2 values for the first element, and so on.
;            If F1F2 is dimensioned correctly, then the atomic scattering
;            factors are not read from the files corresponding to the elements
;            of the ELEMENT array, but rather the passed values are used.
;            (So be sure that the F1F2 values are correct!)
;            Conversely, if F1F2 is undefined (or ill-defined) then the
;            returned value of F1F2 will contain the F1 and F2 values read
;            from the atomic scattering factors files.
;
; COMMON BLOCKS:
;
;           IMD
;           IMD_F1F2TONK
;
; RESTRICTIONS:
;
;  The program uses atomic scattering factors compiled from the CXRO
;  and LLNL websites.
;
;  The atomic scattering factor files must be located in the directory
;  specified in the file IMD_CFG.PRO, located in the IMD installation
;  directory.  However, if the files are not found, the user will be
;  prompted to enter the path leading to these files.
;
; PROCEDURE:
;
;  The optical constants are computed according to the principles
;  described in the paper "X-ray interactions: photoabsorption,
;  scattering, transmission, and reflection at E=50-30,000 eV,
```

```
;  Z=1-92", B.L. Henke, E. M. Gullikson, and J. C. Davis, Atomic
;  Data and Nuclear Data Tables, Vol. 54, No. 2, July 1993. (See
;  especially equations (17) and (18).)
;
; MODIFICATION HISTORY:
;
;   David L. Windt, Bell Labs, Nov 1997.
;
;   July 2013: Modified to compute the default wavelength vector so as to capture the
;   intrinsic energy resolution of the atomic scattering factors files.
;-
```

11.3.2 IMD_NK

```
;+
; NAME:
;
;       IMD_NK
;
; PURPOSE:
;
;       Function used to read IMD optical constant files (.nk files).
;
;
; CALLING SEQUENCE:
;
;       Result=IMD_NK(MATERIAL,WAVELENGTH)
;
; INPUTS:
;
;       MATERIAL - A string specifying the name of a valid
;                  optical constant file, without the .nk extension.
;
;       WAVELENGTH - A scalar or 1-D array specifying the wavelength
;                    in angstroms.
;
; OUTPUTS:
;
;       Result - A complex scalar or 1-D array (the same length
;                as WAVELENGTH) containing the optical constants (n,k).
;
; COMMON BLOCKS:
;
;       IMD
;       IMD_WIDGET
;
; PROCEDURE:
;
;       The optical constant file is read, and the data are interpolated
;       (using FINDEX and INTERPOLATE) to get n and k at the specified
;       wavelengths.
;
; MODIFICATION HISTORY:
;
;  David L. Windt, Bell Labs, April 1997
;-
```

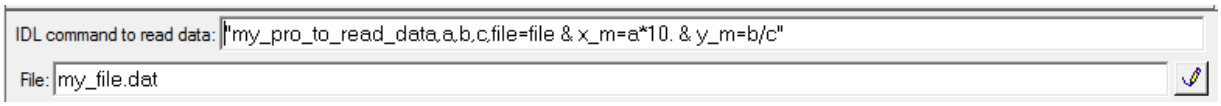
11.4 Data-Importing Routines

The IDL routines described in this section can be used for a variety of purposes in IMD, including importing measured-data files, importing optimization target profile data, and importing stored curve data in IMD*Multiplot. The EROM routine is also used to read Aperiodic Thickness List Files, Layer Data Files, and Clear Aperture files in IMD*Multiplot.

If you are using IMD as an XOP extension, you can use any of the routines described here to import measured data or optimization target profiles into IMD and IMD*Multiplot, as these routines are already compiled into IMD. If you are running IMD as an IDL application, you can also use your own IDL procedures and functions: enter a valid IDL command, using whatever IDL routines you like, in the text-box for "IDL command to read data:" on either the Import Measured Data window or the Import Optimization Target Profile window; you can also use your own IDL routines to read measured-data files in IMD*Multiplot.

If you use your own IDL programs to import measured data or optimization target profile data, keep in mind that the independent variables X_m and X_o must be defined internally in IMD in either angstroms (e.g., for wavelengths/energies or other length variables) or degrees from normal (angles.)

Compound IDL commands (i.e., using the "&" symbol) are allowed when using the GUI interfaces to import data, provided that (a) you surround the entire command string with quotation marks (i.e., " and "), and (b) the first routine in the compound command string that you specify includes as a keyword "FILE=FILE". For example, here's a compound IDL command string to define X_m and Y_m, using some IDL procedure called "my_pro_to_read_data" that reads a file called 'my_file.dat' and somehow defines three quantities: a, b, and c. The values for X_m and Y_m are then computed from a, b, and c as X_m=a*10. and Y_m=b/c:



The procedure used in any compound IDL commands to read data files – "my_pro_to_read_data" in this example – **must** allow the keyword "FILE" to be specified when called; the value passed to the routine using this keyword is the string that you enter in the "File:" text box (i.e., 'my_file.data', in this example.)

11.4.1 EROM

```
;+
; NAME:
;
;      EROM
;
; PURPOSE:
;
;       Read columns of data from a text file.
;
;       This program can be used to read data written by the MORE
;       program.
;
;       The file to be read must be such that if the data are
;       space-separated, then all variables are numeric; String
;       variables are allowed only if the data are separated by tabs,
;       colons, etc.
;
;       The file may contain any number of comment lines - which MUST
;       begin with a semicolon, and MUST be positioned before all data
;       lines.
;
; CALLING SEQUENCE:
```

```
;
;       EROM,V0[,V1,V2,...V9]
;
;       or
;
;       EROM,V=V
;
; KEYWORD PARAMETERS:
;
;   V - Set this keyword to a named variable that will be returned as
;       an array of structures holding the data and the variable names
;       specified in the last comment line.  See RESTRICTIONS below
;       for more details.
;
;   FILE - String specifying the name of a file; if not supplied, the
;          user is queried.
;
;   SKIP - The number of lines at the beginning of the file that
;          should be skipped.
;
;   TAB - Specify /TAB for tab-separated data.  (The default is space-
;         separated data.)  It is only necessary to specify this
;         keyword if the file contains any string data columns.
;
;   SEPARATOR - A string specifying the character separating the data
;               columns.
;
;   COMMENT - Set this keyword to a named variable that will be
;             returned as a string array holding the comment lines
;             included in the file.
;
;   GROUP - GROUP_LEADER keyword passed to DIALOG_PICKFILE if FILE is
;           not specified.
;
;   CANCEL - Set this keyword to a named variable that will be
;            returned to indicate if the user pressed the CANCEL
;            button when prompted for a file to read, if the FILE
;            keyword is not set.
;
; OUTPUTS:
;
;       If the V keyword is not used, then the user must specify the
;       correct number of Vi (V0, V1, etc.) output parameters.
;       There must be as many Vi's specified in the call to EROM as
;       there are columns of data.  The V's are double-precision
;       arrays, unless either the TAB or SEPARATOR keyword is
;       specified in which case they are all string arrays.
;
; RESTRICTIONS:
;
;       If EROM is called with the V keyword, then the columns of data
;       contained in the file are returned as double-precision fields
;       in the returned V structure variable.  Use of the V keyword
;       requires that the data file contain at least one comment line,
;       and the last comment line MUST include the names of the data
;       variables separated by the "|" character.
;
;       For example, to read a file using the V keyword containing
;       three columns of 10 rows of data, then the last comment line
;       in the file must look like this:
;
;       ;  First Variable Name | Second One | Another Variable Name
;
;       Thus the V structure returned by EROM will have the following
;       tag names:
;
;       HELP,/STR,V
;
;       V[0].VALUE     DOUBLE  Array[10]
```

```
;       V[0].NAME      STRING  'First Variable Name'
;       V[1].VALUE     DOUBLE  Array[10]
;       V[1].NAME      STRING  'Second One'
;       V[2].VALUE     DOUBLE  Array[10]
;       V[2].NAME      STRING  'Another Variable Name'
;
; MODIFICATION HISTORY:
;
;       David L. Windt, Bell Labs, March 1990
;
;       January, 1997 - DLW
;         Modified to ignore lines beginning with semicolons, and to
;         accept data separated by tabs, etc.; Removed the NOTITLE and
;         COMMENT keyword; included PICKFILE to prompt for filenames
;         when not specified.
;
;       June, 1997 - DLW
;         Returned numeric variables are now double-precision instead
;         of floating-point.
;
;       DLW, May 2003
;
;         Added V, COMMENTS, GROUP and CANCEL keywords.
;         Replaced call to PICKFILE with call to DIALOG_PICKFILE
;
;       davidwindt@gmail.com
;-
```

11.4.2 RD_REFLECT

```
;+
; NAME:
;
;       RD_REFLECT
;
; PURPOSE:
;
;       Read measured reflectance data files.
;
; CATEGORY:
;
;       IMD
;
; CALLING SEQUENCE:
;
;       RD_REFLECT
;
; KEYWORD PARAMETERS:
;
;       FILE = A scalar or string array of Reflect-compatible file names.
;       If FILE is an array, then the optical data contained in each
;       specified file is concatenated to construct a single vector of
;       X_M, Y_M, and SIGY_M values.
;
;       SMOOTH = if non-zero and the specified file contains raw
;       intensity data, then the normalized intensity is recomputed as
;       Normalized_Intensity = smooth(Intensity,SMOOTH) /
;       smooth(Incident_Intensity,SMOOTH).
;
;       SCALE = if FILE is an array, then SCALE can be used to apply
;       different scaling factors to the data contained in each file.
;       For example, of FILE is a 3 element array, then SCALE would
;       also be a 3 element array containing the values that will be
;       used to scale the Y_M values in each of the 3 files.  (e.g.,
;       SCALE=[1.,1.e-3,1.e-6])
;
;       X_MIN, X_MAX = set these values to exclude any data with X_M
;       less than or equal to X_MIN and/or X_M greater than or equal
```

```
;       to X_MAX
;
; COMMON BLOCKS:
;
;       COMMON IMD
;       COMMON IMD_M_O
;
; SIDE EFFECTS:
;
;       The IMD variables X_M, Y_M, SIGY_M, IMD.X_M and IMD.Y_M are
;       defined.
;
; RESTRICTIONS:
;
;       REFLECT files must be read-able using "EROM,V=V"; see EROM
;       documentation for details. Furthermore, they must follow the
;       following format precisely:
;
;       The "|"-separated header read by EROM,V=V must include one
;       column title that includes the exact phrase 'Wavelength [A]',
;       'Theta', or 'Energy [keV]', corresponding to the independent
;       variable data, and one column title that includes the phrase
;       'Reflectance, R', 'Transmittance, T', 'Non-Specular Reflected
;       Intensity, (1/I_0)dI/dOmega', or 'Non-Specular Transmitted
;       Intensity, (1/I_0)dI/dOmega', corresponding to the dependent
;       variable data. These names must be exactly right! If the
;       independent variable title contains the letters 'Theta', and
;       also the letters 'grazing' (e.g., 'Theta, grazing') then
;       angle data will be read as grazing incidence angles.
;
;       If there is also a column title that includes the dependent
;       variable name included in parenthesis, with a "d" in front,
;       e.g., "d(Reflectance, R)", then that column of data will be
;       used to define SIGY_M.
;
; MODIFICATION HISTORY:
;
;       David L. Windt, Reflective X-ray Optics
;
;       2004
;
;       davidwindt@gmail.com
;-
```

11.4.3 RD_ALS

```
;+
; NAME:
;
;       RD_ALS
;
; PURPOSE:
;
;       Read in optical data created at the ALS Reflectometry and
;       Scattering Beamline 6.3.2.
;
;       The ALS data file should be a plain text file containing 2 or
;       more columns of data; the first line of the file is a header
;       (which is ignored by this program.)
;
; CATEGORY:
;
;       IMD
;
; CALLING SEQUENCE:
;
;       RD_ALS[,X_M_COLUMN,Y_M_COLUMN]
;
```

```
; INPUT PARAMETERS:
;
;       X_M_COLUMN = an integer specifying which data column (starting
;       at 0) contains the IMD X_M data. (This is generally equal to
;       0, i.e., corresponding to the first column.)
;
;       Y_M_COLUMN = an integer specifying which data column contains
;       the IMD Y_M data.
;
;       X_M_COLUMN and Y_M_COLUMN may be omitted, in which case they
;       default to the values 0 and 1, respectively.
;
; KEYWORD PARAMETERS:
;
;       FILE = A scalar or string array of ALS data file names.  If
;       FILE is an array, then the optical data contained in each
;       specified file is concatenated to construct a single vector of
;       X_M, Y_M, and SIGY_M values.
;
;       X_M_SCALE = Scaling factor for X_M values. If the ALS data
;       contains data that is not in the units used internally by IMD,
;       i.e., angstroms and normal incidence in degrees, then you can
;       scale the data in the file when it is read using the X_M_SCALE
;       keyword.
;
;       Y_M_SCALE = Scaling factor for Y_M values. If FILE is an array,
;       then Y_M_SCALE can be used to apply different scaling factors to
;       the data contained in each file.  For example, of FILE is a 3
;       element array, then Y_M_SCALE would also be a 3 element array
;       containing the values that will be used to scale the Y_M
;       values in each of the 3 files.  (e.g., Y_M_SCALE=[1.,1.e-3,1.e-6])
;
;       X_MIN, X_MAX = set these values to exclude any data with X_M
;       less than or equal to X_MIN and/or X_M greater than or equal
;       to X_MAX
;
;       UNCERTAINTY = a numerical constant use to create SIGY_M values
;       from Y_M data: i.e., SIGY_M=Y_M*UNCERTAINTY. Default=0.0
;
; COMMON BLOCKS:
;
;      COMMON IMD
;      COMMON IMD_M_O
;
; SIDE EFFECTS:
;
;       The IMD variables X_M, Y_M, and SIGY_M are defined.
;
; MODIFICATION HISTORY:
;
;      David L. Windt, Reflective X-ray Optics
;
;      February 2013
;
;      davidwindt@gmail.com
;-
```

11.4.4 RD_SPEC

```
;+
; NAME:
;
;       RD_SPEC
;
; PURPOSE:
;
;       Read data files written by the program SPEC.
;
```

```
; CALLING SEQUENCE:
;
;       RD_SPEC[,DETECTOR]
;
; OPTIONAL OUTPUTS:
;
;       DETECTOR - Array of detector counts.
;
; KEYWORD PARAMETERS:
;
;       FILE - A string or string array specifying the file(s) to
;              read.  If more than one file is specified, then the
;              data arrays are concatenated.
;
;       SCAN - A scalar, or an array of integers, specifying which
;              scan(s) to be read; default is the 1st found if not
;              set.  You can concatenate multiple scans from the same
;              file by specifying a scalar value for FILE, and an
;              array for SCAN.
;
;       MONITOR - Array of monitor counts.
;
;       SECONDS - Array of integration times.
;
;       EPOCH - Array of seconds since epoch time.
;
;       H - h-momentum transfer vector.
;
;       K - k-momentum transfer vector.
;
;       L - l-momentum transfer vector.
;
;       THETA - Array of theta values.
;
;       TWOTHETA - Array of two-theta values.
;
;       PHI - Array of phi values.
;
;       CHI - Array of chi values.
;
;       COLUMN_TITLES - A string array that returns the data column
;                       titles in the file.
;
;       NAME_DETECTOR - Set this keyword to the name of the data
;                       column title corresponding to the detector
;                       counts if it is different than the default
;                       value of 'Detector'.
;
;       NAME_V1, NAME_V2, NAME_V3, V1, V2, V3 - If there are data
;                                               columns with column
;                                               titles other than the
;                                               standard ones
;                                               (detector, monitor,
;                                               seconds, epoch, h, k,
;                                               l theta, twotheta,
;                                               phi, and chi) then the
;                                               V1, V2, and V3 values
;                                               can be used in
;                                               conjunction with the
;                                               NAME_V1, NAME_V2, and
;                                               NAME_V3 to extract
;                                               these data.
;
;                   V1, V2, and V3 are the data arrays, while NAME_Vx
;                   are the column titles.  For example, if you have
;                   a SPEC file with the column title 'Ion Chamber',
;                   then set NAME_V1='Ion Chamber'; the values of
;                   'Ion Chamber' will be returned in the V1
;                   variable.
```

```
;
;       SILENT - set to inhibit printing file info.
;
;       SET_IMD - Set to instruct RD_SPEC to convert the data
;                 contained in the file to IMD Measured Data
;                 variables, Reflectance vs. Theta.  It is assumed
;                 that V1=TH in this case.  The IMD angle pointer
;                 (IMD.X_M) will be set to 0 (angles).
;
;       INC_INT - Incident intensity: If SET_IMD is set, then the data
;                 will be normalized to the incident intensity.  If
;                 INC_INT is equal to zero, or undefined, then it will
;                 be assumed that the incidence intensity is equal to
;                 twice the value of the first element of DETECTOR (as
;                 would be the case for reflectance vs angle data
;                 starting from TH=0., for a properly aligned sample.)
;
;       ATTENUATION - An additional attenuation factor used to
;                     normalized the data: If SET_IMD is set, and
;                     ATTENUATION is non-zero, then the computed
;                     reflectance will be given by:
;                     Y_m=DETECTOR/INC_INT/ATTENUATION
;
;                      ATTENUATION can be an array, equal to the
;                      number of elements of FILE, in order to
;                      normalize several data sets at once.
;
;       OFFSET - Often, when measuring the specular reflectance of a
;                film, i.e. in the TH-TTH geometry, it is desirable to
;                also perform an offset scan, i.e., a TH-TTH scan with
;                TTH offset by some small amount, in order to account
;                for diffuse scattering.  If an offset scan has been
;                performed, the diffuse intensity can be subtracted
;                from the specular intensity when computing
;                reflectance (assuming the SET_IMD keyword is set) by
;                setting this variable to an array (having the same
;                number of elements as SCAN) of 0s and 1s, where the
;                1s point to the offset scan data.  Make sure that the
;                ATTENUATION variable corresponding to the offset scan
;                data is set properly.
;
;                For example, suppose three scans were performed for a
;                film: 1. TH=0-1, with a 1000:1 attenuator, 2. TH=1-5
;                with no attenuator, and 3. an offset scan from TH=0-5
;                (with TTH=2*TH+0.3), also with no attenuator.  In
;                this case, the command to read the file and compute
;                the offset-corrected reflectance would be:
;
;                RD_SPEC,/SET_IMD,SCAN=[1,2,3],OFFSET=[0,0,1],ATTEN=[1.,1000.,1000.]
;
; COMMON BLOCKS:
;
;       COMMON IMD
;       COMMON IMD_M_O
;
; SIDE EFFECTS:
;
;       If SET_IMD is set, then the IMD variables X_M, Y_M, SIGY_M,
;       and IMD.X_M are defined. (Set IMD.Y_M manually, according to
;       the type of data: 0 for reflectance, 1 for transmittance, 2
;       for absorptance.)
;
; MODIFICATION HISTORY:
;
;       David L. Windt, Bell Laboratories, June 1996.
;
;       Modified to work with IMD V3, (Added SET_IMD, INC_INT, and
;       ATTENUATION keywords) May 1997
;
```

```
;       Include code to read extra header line (G2) in spec V4.03 files,
;       Aug 1997
;
;       Added OFFSET keyword, December 1997.
;
;       Updated IMD common blocks to IMD version 4.0, May 1998
;
;       Almost completely re-written to handle arbitrary SPEC file formats.
;       The program now figures out which columns of data are which, and
;       returns the appropriate data vectors into keyword variables.
;
;       August 2013: I haven't actually used this program in years
;       now, and have no idea if it really works as advertised anymore
;       on arbitrary SPEC files.
;
;      davidwindt@gmail.com
;-
```

12 Performance and Troubleshooting

This version of IMD was developed almost entirely on a linux platform (CentOS), using IDL 6.3 since it was released in 2007, and previous versions of IDL before that. Some rudimentary testing was performed on Windows 7 while this document was being written and the example files were being created (you may recognize the screen captures as coming from Win7). Windows 7 testing was performed using either IDL 6.3 or XOP 2.3, which in turn uses IDL 7. The linux version was also tested briefly prior to release using IDL 8.2.3, and on Mac OS X minimal testing using XOP 2.3 was performed as well. If you find any performance issues on these or any other platforms, please don't hesitate to let me know.

I was surprised to find, when comparing the performance of IMD using IDL 6.3 on two comparable computers, one running linux and one running Win 7, that the Win 7 computation times were consistently a little faster than the linux computation times, by a small but non-negligible amount. However (!), the performance of IMD's GUI on Win 7 platforms is often rather poor, to say the least. For example, opening the Dependent Variables tab in IMDXPLOT is really slow; opening the Dependent Variables tab in IMD Preferences is ridiculously slow! On the other hand, I find that IMD's GUI is generally fast and very responsive on even my oldest linux computers. These performances differences in IMD's GUI are apparently a manifestation of differences in IDL's "widgets", which are used to create the IMD GUI, and that use different GUI toolkits on different platforms. (Or perhaps there's a bug somewhere!?!) I've done what I can to maximize the GUI performance on Win 7, but it's still not great, I know. Please understand that it's essentially out of my control with regard to how the IDL code is written. I wish IDL's widgets worked better under Windows. You may want to consider a linux computer for IDL/IMD if you find the Windows 7 GUI performance to be as poor as I have!

12.1 Frequently Asked Questions

12.1.1 Q: I can't get IMD to work/start.

A: Check that you've installed IMD exactly as per the instructions. See §1.5

12.1.2 Q: I'm getting error messages about running out of memory.

A: If you create a structure containing many layers, and/or have defined many independent variable values, IMD will need to create large, multi-dimensional arrays that might consume more memory than what is available to IDL. Should IDL run out of memory during a computation, you can try adjusting the value of the SP_MAX_ARRAY_SIZE and/or NS_MAX_ARRAY_SIZE variables (depending on the optical functions you're trying to compute) in the `imd_cfg.pro` configuration file described in §1.4.

12.1.3 Q: I'm getting reflectance (or whatever) values that can't be right.

A: Double-check that the structure and variables that you've defined make sense. For example, when modeling interface imperfections in particular, IMD uses some approximations, as described in §4.11. It's easy to violate the conditions under which those approximations are valid, for example by specifying enormous interface widths. Also, make sure that the optical constants you're using for the calculation are accurate.

12.1.4 Q: I can't get IMD to fit my data! / How do I get IMD to fit my data?

A: I just write the code, dude – I'm no 'fitting wizard', unfortunately. It seems to me to be more of an art than a science. My advice: read up on the subject, and keep trying!

12.1.5 Q: The Marquardt or Levenberg-Marquardt algorithms seem to 'converge' way too soon, without finding a good solution to the fitting/optimization problem.

A: Yep.

12.1.6 Q: Can I get a copy of the IMD source code?

A: Nope.

12.1.7 Q: Please? I promise I won't show it to anyone.

A: Still no. Sorry.

12.1.8 Q: Will/can you include feature X in a future release of IMD?

A: Maybe! Tell me about it.

12.1.9 Q: Will IMD work with the free IDL Virtual Machine?

A: Unfortunately, no. IMD makes extensive use of the IDL "execute" function, which is explicitly excluded from the Virtual Machine.

12.2 Reporting Bugs

While there are almost certainly programming errors in the IMD source code that remain undetected at the time of this release, IMD is supposed to be resilient to such problems. Nevertheless, nobody's perfect. So if you do find a situation where IMD stops working, I'd like to know about it. Please contact me with as much detail as you can provide (exactly what you were doing, exactly what happened, etc.); if you can send a faulty .imd file that would help greatly.

Meanwhile, the simplest work-around for you if IMD stops working, assuming that you're using IMD as an IDL application, is to exit IDL and start over. Or, you can try typing the command "retall" at the IMD prompt and see if IMD comes back to life. If you encounter a fatal error using IMD as an XOP extension, the program will probably just exit completely (perhaps with an error message that you can report to me).

If you've identified a computation result that seems wrong or non-physical, or that doesn't agree with some other calculation you've made somehow, and you think it's due to a programming problem in IMD, again, I'd be very interested to know about it. However, please keep in mind that the core computational routines used by IMD – FRESNEL, NS_BA, and NS_DWBA, described in §11.2 – remain unchanged for the most part since the last release more than a decade ago; as far as I know, there are no known problems associated with those routines. So, I hope you'll please endeavor to be sure that you've found a real programming error before contacting me about it.

davidwindt@gmail.com

Also, check out the IMD & TOPO Google group (https://groups.google.com/forum/#!forum/imd_topo) for discussion, and announcements of future releases.

13 Version History

Version 5.02 – 27-Nov-2013

- Various bug fixes and minor user-interface changes/enhancements.
- Added auto-load capabilities (see §10.4).
- Changes to IMD Preferences window: added alerts controls; added ability to display additional controls in IMDXPLOT by default.
- Changes to Fitting Settings and Optimization Settings windows: added option to ask to restore original configuration on abort; added option to genetic algorithms to not use initial values as genes of first individual of first generation.
- Added optional display of residuals to Fitting Results and Optimization Results windows.
- Updated IMD.pdf.

Version 5.01 – 17-Oct-2013

- Various bug fixes.

Version 5.0 – 7-Oct-2013

14 References

[1] D. L. Windt, 'IMD - Software for modeling the optical properties of multilayer films', Computers in Physics, 12, 360-370 (1998)

[2] B. L. Henke, E. M. Gullikson, and J. C. Davis, 'X-ray interactions: photabsorption, scattering, transmission, and reflection at E-50-30,000 eV, Z=1-92', Atomic Data and Nuclear Tables, 54 (1993). In addition to the data contained therein, the Center for X-Ray Optics (CXRO), Lawrence Berkeley Laboratory, maintains an active database of atomic scattering factors; these data are available at http://www-cxro.lbl.gov, and have been included in IMD courtesy of E. M. Gullikson

[3] High-energy atomic scattering factors were available at one time from the Lawrence Livermore National Laboratory at http://www-phys.llnl.gov/V_Div/scattering/asf.html. That link no long works, apparently, and I'm not sure where these data now reside (other than in the imd/f1f2 directory!)

[4] Neutron atomic scattering factors provided courtesy of M. Sanchez del Rio and L. Alianelli

[5] M. Born and E. Wolf, Principles of Optics, sixth edition, Pergamon Press, Oxford, 1980

[6] D. G. Stearns, 'The scattering of X-rays from non-ideal multilayer structures', J. Appl. Phys. 65, 491-506 (1989)

[7] L. Névot and P. Croce, 'Caracterisation des surface par reflexion rasante de rayons X. Application a l'etude du polissage de quelques verres silicates', Revue. Phys. Appl., 15, 761-779 (1980)

[8] D. G. Stearns, 'X-ray scattering from interfacial roughness in multilayer structures', J. Appl. Phys., 71, 4286-4298 (1992)

[9] D. G. Stearns, D. P. Gaines, D. W. Sweeney, and E. M. Gullikson, 'Nonspecular X-ray scattering in a multilayer-coated imaging system', J. Appl. Phys. 84, 1003-1028 (1998)

[10] S. K. Sinha, E. B. Sirota, S. Garoff, and H. B. Stanley, 'X-ray and neutron scattering from rough surfaces', Phys. Rev. B, 38, 2297-2311 (1988)

[11] V. Holy, J. Kubena, I. Ohlidal, K. Lischka, and W. Plotz, 'X-ray reflection from rough layered systems', Phys. Rev. B, 47, 15896-15903 (1993)

[12] V. Holy and T. Baumbach, 'Nonspecular X-ray reflection from rough multilayers', Phys. Rev. B, 49, 10668-10676 (1994)

[13] D. K. G. de Boer, 'X-ray scattering and x-ray fluorescence from materials with rough interfaces', Phys.Rev. B, 53, 6048-6064 (1996)

[14] D. G. Stearns, 'Stochastic model for thin film growth and erosion', Appl. Phys. Lett. 62, 1745-1747 (1993)

[15] D. K. G. de Boer, 'X-ray reflection and transmission by rough surfaces', Phys. Rev. B, 51, 5297-5305 (1995)

[16] Handbook of Optical Constants of Solids, edited by E. D. Palik, Academic Press, Inc., 1985

[17] I. V. Kozhevnikov, I. N. Bukreeva, E. Ziegler, 'Design of X-ray supermirrors', Nucl. Inst. Meth., A, 460, 424 – 443 (2001)

[18] D. L. Windt, W. K. Waskiewicz and J. E. Griffith, `Surface finish requirements for soft X-ray mirrors', App. Opt., 33, 2025-2031 (1994)

[19] D. W. Marquardt, 'An algorithm for least-squares estimation of nonlinear parameters', J. Soc. Ind. Appl. Math., 11, 431-441 (1963)

[20] P. R. Bevington, Data Reduction and Error Analysis for the Physical Sciences, McGraw-Hill, New York, 1969

[21] C. Markwardt's MPFIT program can be found at http://cow.physics.wisc.edu/~craigm/idl/idl.html

[22] MINPACK-1, J. Moré, FORTRAN code available at http://www.netlib.org; Optimization Software Guide, Jorge Moré and Stephen Wright, SIAM, Frontiers in Applied Mathematics, 14 (1993)

[23] P. D. Binda and F. E. Zocchi, 'Genetic algorithm optimization of X-ray multilayer coatings', SPIE Vol. 5536, 97-108 (2004)

[24] M. Björck, 'Fitting with differential evolution: an introduction and evaluation', J. Appl. Cryst., 44, 1198-1204 (2011)

[25] M. Lampton, B. Margon and S. Bowyer, 'Parameter estimation in X-ray astronomy', Ap. J., 208, 177 (1976)

[26] W. Cash, 'Generation of confidence intervals for model parameters in X-ray astronomy', Astron. & Astrophys., 52, 307 (1976)