

Emergent Properties of Plasmonic Systems in the Weak to Strong Coupling Regimes:

Author: Aaron Harold Rose

Persistent link: <http://hdl.handle.net/2345/bc-ir:108654>

This work is posted on [eScholarship@BC](#),
Boston College University Libraries.

Boston College Electronic Thesis or Dissertation, 2019

Copyright is held by the author. This work is licensed under a Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0>).

EMERGENT PROPERTIES OF PLASMONIC
SYSTEMS IN THE WEAK TO STRONG
COUPLING REGIMES

Aaron Harold Rose

A dissertation
submitted to the Faculty of
the department of Physics
in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

Boston College
Morrissey College of Arts and Sciences
Graduate School

November 2019

EMERGENT PROPERTIES OF PLASMONIC SYSTEMS IN THE WEAK TO STRONG COUPLING REGIMES

Aaron Harold Rose

Thesis Supervisor: Professor Michael J. Naughton, Ph.D.

Abstract

In this dissertation I present studies of plasmonic interactions in different coupling regimes, from zero to strong coupling and approaching ultrastrong coupling. Different physics are manifest in each regime, with different possible applications. The first project uses finite element electromagnetic simulations to model plasmonic waveguides that couple near field light into the far-field for sub-diffraction limited microscopy. $\lambda/32$ resolution is shown by minimizing coupling between adjacent waveguiding nanowires, with minimal attenuation over a few microns. The next two projects, by contrast, seek to maximize coupling between plasmons and excitons into the strong coupling regime where the optoelectronic properties are modified and quantum coherent phenomena may be observed. Strong exciton–plasmon coupling in MoS₂ is shown experimentally at room temperature and found to be a general phenomenon in other semiconducting transition metal dichalcogenides using transfer matrix modeling. A semiclassical oscillator model is fit to the experimental data to discover coherent hybridization between the ground and first excited states of MoS₂. Enhanced coupling is found at the third excitonic transition, approaching the ultrastrong coupling regime where exotic properties are predicted to emerge, such as ground state virtual photons. Our strong coupling studies motivate further studies of the TMDCs as a platform for coherent quantum physics with possible applications in quantum computing and cryptography.

Acknowledgments

There are many people who were involved in my PhD and I would like to express my gratitude toward all of them. My wife deserves the greatest recognition for her endless support. Amy, thank you for your patience. Thank you for caring for the kids when I was away. Thank you for your encouragement. To my boys, Sam and William, thank you for letting Daddy finish his "project."

To my advisor, Mike Naughton, thank you for supporting me through the many projects and a move to Colorado. You gave me the opportunity to learn a great deal and build a foundation for a career as a scientist.

Thank you Jao van de Lagemaat for giving me the chance to finish my PhD at NREL and learn a new set of skills. Your generosity made my move to Colorado painless.

Many thanks to my other mentors at BC. Professor Kempa for teaching me about plasmonics and entertaining my questions and ideas. Dr. Mike Burns for knowing something about everything and always being available for advice. Professor Juan Merlo for his insight and discussions on plasmonics and for his friendship. Steve Shepard for teaching me much of what I know about clean room techniques.

I was lucky to have great lab mates in the Naughton Lab at BC. When I first arrived, Drs. Tim Kirkpatrick, Binod Rizal, and Fan Ye, and Jeff Naughton provided invaluable mentorship. Drs. Yitzi Calm, Nathan Nesbitt, Luke D'Imperio, Michelle Archibald, and Amy Valera were great colleagues as were Chaobin Yang, Victoria Gabriele, and Mark Schiller. Also, Drs. Shirley Wu and Jiantao Kong were great to work with on theory. I had the pleasure of mentoring and working with many undergraduates including Abdullah Rashed, Robert Hatem, Michelle Solomon, and Laura Simko.

I have also had great colleagues at NREL. Thank you to Dr. Andrew Ferguson for finding me lab space when I first arrived. Dr. Elisa Miller-Link has been a great career mentor. Drs. Hanyu Zhang, Jeremy Dunklin, Sanjini Nanayakkara, Abdoulaye Djire, Zhaodong Li, Yaxin Zhai, Xihan Chen, and Elisabeth McClure have been wonderful

colleagues.

Throughout my PhD I have received funding from a variety of entities, for which I am grateful. Thank you to the W.M. Keck Foundation and the DOE Office of Basic Energy Sciences, Solar Photochemistry Program at NREL for funding me to do research full time. Further, the BC Department of Physics funded me as a teaching assistant for several semesters, which was a rewarding experience.

Now, I must express the following views of the U.S. Government: This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by DOE Office of Science, Office of Basic Energy Sciences, Division of Chemical Sciences, Geosciences, and Biosciences, Solar Photochemistry Program. The views expressed in this dissertation do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting this dissertation for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

Finally, I would like to thank my parents, Valerie and Keith Rose for always having steak ready when I visit.

Contents

Abstract	iii
Acknowledgments	v
List of Figures	xxv
List of Tables	xxvi
1 Introduction	1
1.1 Background and motivation	1
1.2 Plasmonics	2
1.2.1 Properties	2
1.2.2 Dispersion relation	3
1.2.3 Drude response	3
1.2.4 Dispersion plot with Drude response	4
1.2.5 Dispersion plot with experimental response	6
1.2.6 Plasmon length scales	8
1.2.7 Dispersion in thin films	11
1.2.8 Plasmon excitation methods	12
1.3 Theory of quantum emitter–optical cavity coupling	19
1.4 Plasmonic interactions from weak to strong coupling and beyond	25
1.4.1 Zero coupling	25
1.4.2 Weak coupling	26
1.4.3 Strong coupling	27
1.4.4 Ultrastrong coupling	29
1.5 Summary	29
2 Methods	31

2.1	Fabrication	31
2.1.1	Physical vapor deposition	31
2.1.2	Chemical vapor deposition	32
2.1.3	Transfer of quasi two-dimensional films from growth substrate to alternate substrate	33
2.1.4	Substrate cleaning	34
2.2	Experimental	36
2.2.1	Angle-resolved reflectivity in the Kretschmann-Raether config- uration and experimental dispersion	36
2.2.2	Photoluminescence	38
2.2.3	Raman microscopy	39
2.2.4	Atomic force microscopy	40
2.2.5	Profilometry	40
2.2.6	Optical microscopy	41
2.2.7	Ellispometry	41
2.3	Theory	43
2.3.1	Electromagnetic finite element modeling	43
2.3.2	Transfer matrix modeling of optical properties of stacks of thin films	43
2.3.3	Semiclassical oscillator model of strong coupling	44
3	Nanoscope based on nanowaveguides	47
3.1	Abstract	47
3.2	Background	47
3.3	Results and discussion	52
3.4	Conclusions	57
4	Plasmon-mediated coherent superposition of discrete excitons un- der strong exciton–plasmon coupling in few-layer MoS₂ at room temperature	59
4.1	Abstract	59

4.2	Background	60
4.3	Fabrication and material characterization	62
4.4	Data collection, analysis, and discussion	68
4.5	Application to other systems	80
4.6	Conclusions	82
5	Enhanced strong coupling of band-nested excitons in few-layer TMDCs	85
5.1	Abstract	85
5.2	Background	86
5.2.1	Strong coupling and the 2D TMDCs	86
5.2.2	The C-exciton of 2D TMDCs	87
5.2.3	Strong coupling at the C-exciton	91
5.3	Fabrication and data collection	91
5.4	Analysis in k -space	92
5.4.1	Experimental and simulated exciton–plasmon dispersion . . .	92
5.4.2	Coupled oscillator model fit	94
5.4.3	Coupled mode Hopfield coefficients	99
5.5	Analysis in angle-space	101
5.5.1	Experimental and simulated exciton–plasmon dispersion . . .	101
5.5.2	Coupled oscillator model fit	103
5.5.3	Coupled mode Hopfield coefficients	106
5.5.4	Conclusion	108
6	Conclusions	109
	Bibliography	111
A	List of Publications	121
B	Supplementary figures for Chapter 3	123
C	Supplementary strong coupling data	129

C.1	Reproducibility in MoS ₂ -Ag	129
C.2	Strong coupling in MoS ₂ -Au	134
C.3	Strong coupling in aqueous electrolyte	135
D	Embedded Metal Nanopatterns for Light Management in Ultra-thin Solar Absorbers	139
D.1	Background	139
D.2	Amorphous silicon	142
D.3	Cadmium telluride	146
E	Near-infrared plasmonic nanostructures	151
E.1	Nanostructure	152
E.2	Results	153
F	MATLAB code for transfer matrix model	161
F.1	Dependencies	161
F.2	Code	163
F.2.1	ExampleUsage	163
F.2.2	AvsAngle	165
F.2.3	plotDispersion	167
F.2.4	thinfilmRTA	171
F.2.5	jreftran_rt	175
F.3	Refractive Indices	178
G	MATLAB code Strong coupling analysis of Chapter 5	179

List of Figures

1-1	Surface plasmon polariton dispersion at the interface between bulk silver (Drude model) and vacuum. k_x is the wavevector parallel to the surface and the direction of plasmon propagation. The upper branch minimum is at the bulk plasma energy (dotted gray line) while the lower branch approaches monopole surface plasmon energy (dot-dashed gray line). The upper branch is radiative and does not support a bound plasmon. The lower branch lies to the right of the light line (dashed gray line) and is therefore a non-radiative, bound mode.	5
1-2	Surface plasmon polariton dispersion at the interface between bulk silver and vacuum, using experimental values of the silver dielectric function. The gap between the plasmon branches in Figure 1-1 is reduced and replaced with a real solution with negative group velocity. At higher k_x , the plasmon becomes more confined, exhibiting shorter effective wavelength.	7
1-3	Skin depth of plasmon into Ag and SiO ₂ at the interface of semi-infinite Ag and SiO ₂ . Ag refractive index from Johnson and Christy [6]. SiO ₂ modeled with constant refractive index $n = 1.45$	9
1-4	Propagation length of surface plasmon at the interface of semi-infinite Ag and SiO ₂ . Ag refractive index from Johnson and Christy [6]. SiO ₂ modeled with constant refractive index $n = 1.45$	10

1-5 Dispersion of plasmon of thin film Ag sandwiched between semi-infinite layers of SiO₂. Dispersion is numerically calculated using experimental optical properties. Symmetric thin films exhibit symmetric (L-) and antisymmetric (L+) modes, shown in the left and right panels, respectively. For films ~50 nm and greater, the two modes overlap, while they deviate at lower thicknesses. Reprinted figure with permission from Dionne, J. A., Sweatlock, L. A., and Atwater, H. A. *Phys. Rev. B* 72, 075404 (2005). Copyright 2005 by the American Physical Society. 12

1-6 Dispersion of plasmon in 50 nm thin film Ag on semi-infinite fused silica (glass), in air. Experimental refractive indices are used in transfer matrix modeling. [13] P-polarized plane waves are excited from the glass side. Absorptivity A , is calculated as $A = 1 - R$, where R is reflectivity. Transmission is assumed zero due to total internal reflection conditions. White dashed line indicates the light line in air. Black dotted line is light line in fused silica. 16

1-7 Dispersion of two interfacial plasmon modes in 50 nm thin film Ag. As in Figure 1-6, Ag is on semi-infinite fused silica (low index glass, refractive index $n = 1.46$), in air. However, in this model, light is shined through a semi-infinite slab of high index glass (SLAH-79, refractive index $n \simeq 2$) which is 100 nm above the Ag surface, so that air fills the gap. The high k -components of the evanescent field at the surface of the high index glass may couple to modes at the fused silica / Ag and Ag / air interfaces, allowing both modes to be seen in the dispersion. White dashed line indicates the light line in air. White dotted line is light line in low index glass. Black dotted line is light line in high index glass. Modeling is done similar to that in Figure 1-6. 18

1-8 Imaginary part of eq 1.7 showing effect of coupling strength on line width of coupled quantum emitter and optical mode. Under strong coupling conditions, the line widths collapse to one value as the modes become completely hybridized. $F_P = 1$ denotes coupling strength corresponding to a Purcell Factor of 1. $g = 1$ shows the weak-to-strong coupling crossover in this scenario. The line near $g = 1.5$ shows where the Rabi splitting is larger than the average line width and thus approximately where strong coupling starts to be observable and long-living. 20

1-9 Effect of coupling strength on spectrum. a) Real part of eq 1.7 showing effect of coupling strength on mode energy. Under strong coupling conditions, the modes fully hybridize and a gap opens up the size of the Rabi energy. Near the critical point, the observed Rabi splitting may be smaller than the $2g$. At higher coupling strength, $\Omega \rightarrow 2g$. b) Spectral shapes of sum of two Lorentz oscillators with line widths and separation dictated by Figures 1-8 and 1-10, at coupling strengths indicated with red Roman numerals. The Purcell effect modifies the line width of the single mode prior to onset of strong coupling (i, ii). At strong coupling threshold, the hybrid modes are not resolvable (iii). Strong coupling effects compete with other decay mechanisms near the point when the coupling strength equals the average line width (iv), and Rabi splitting is resolvable. 22

1-10 Effect of emitter:cavity linewidth ratio ($\kappa : \gamma$) on coupling behavior. The Purcell effect (orange line indicates $F_P = 1$) onset is earliest for small ratios. The coupling strength required to enter the strong coupling regime goes to zero as the ratio goes to 1 (green line) while the condition to resolve strong coupling (green line) gradually increases. 24

2-1	Plasmon excitation and angle-resolved reflectivity using the Kretschmann-Raether method: The plasmon is excited via p-polarized white light incident on the plasmonic thin film through the backside of the glass prism. Angle-resolved reflection is collected by varying θ_e and measuring the intensity of the reflected light relative. The excited plasmon decays evanescently away from the surface of the prism.	37
3-1	Schematic of imaging with proposed near field optical magnifier comprised of nanowaveguides (white lines) supported by dielectric filler (blue). The subwavelength-size “object” is placed at the input ends of the nanowaveguides array (having waveguide separation $d < \lambda$). When illuminated, the image of the object is propagated along the waveguides and re-emitted as a magnified image at the output ends of the waveguides (with separation $d > \lambda$). Traditional optics are used to capture the magnified image.	49
3-2	SPP propagation along 18.6 μm long silver nanowire. (a) Sketch of optical excitation; I is input and D is distal end of the wire. (b) Microscopic image—the bright spot to the left is the focused exciting light. The arrow indicates light scattered from the distal wire end. (c) SNOM image—the image area corresponds to the white box in (b). (d) 2 μm long cross-cut along the chain dotted line in (c). Optical excitation is with 785 nm laser. Reprinted figure and caption with permission from Ditlbacher, H., et al. Phys. Rev. Lett. 95, 257403 (2005). Copyright 2005 by the American Physical Society.	51
3-3	X and z components of the electric field along silver nanowires 40 nm in diameter and 5 μm in length, excited with light of vacuum wavelength (a) $\lambda = 800$ nm and (b) 500 nm. SPP modes are clearly visible in each case, but with case (b) showing high attenuation over the length of the wire.	52

3-4 X and z components of the electric field along nanocoaxes $5 \mu\text{m}$ in length, excited with light of vacuum wavelength (a) $\lambda = 800 \text{ nm}$ and (b) 500 nm . In both cases, the coaxes have a 40 nm diameter inner silver core, a 20 nm thick vacuum annulus, and a 10 nm thick silver shield. The nanocoax modes appear to be a combination of classical coaxial cable TEM modes and SPP modes. Case (a) shows little attenuation while case (b) shows clear attenuation, albeit less than for the nanowire in Figure 3-3b. 53

3-5 X and z components of the electric field along nanocoaxes $5 \mu\text{m}$ in length, excited with light of vacuum wavelength (a) $\lambda = 800 \text{ nm}$ and (b) 500 nm . In both cases, the coaxes have a 40 nm diameter inner silver core, a 20 nm thick vacuum annulus, and a 10 nm thick silver shield. The nanocoax modes appear to be a combination of classical coaxial cable TEM modes and SPP modes. Case (a) shows little attenuation while case (b) shows clear attenuation, albeit less than for the nanowire in Figure 3-3b. 54

3-6 X and z components of the electric field in a two nanowire NFOM. The wires are 40 nm in diameter, $2 \mu\text{m}$ in length, and made of silver. Their close ends are separated by 80 nm , center-to-center, while their distal ends are separated by $1 \mu\text{m}$, center-to-center. They are excited by $\lambda = 800 \text{ nm}$ vacuum wavelength light, with the aperture positioned adjacent to the center of the top wire ((a) and (c)) or halfway between the two wires ((b) and (d)). Excitation of the top wire results in SPP modes highly localized to that wire alone, while excitation between the wires yields symmetric, low amplitude SPP modes on both wires. . . 56

3-7	Calculated normalized transmitted electric field intensity along the converging wires of Figure 3-6, as the excitation aperture is moved along the x -axis, from halfway between both wires ($x=0$) to above and below both wires ($x=\pm 100$ nm). This intensity is found by averaging the normalized magnitude of the time-averaged Poynting vector in a cylinder 60 nm in radius surrounding the distal half of the wires. The inset shows the geometry.	57
4-1	Plasmon excitation and angle-resolved reflectance using the Kretschmann-Raether method: The plasmon is excited via p-polarized white light incident on the FL-MoS ₂ -Ag thin film through the backside of the glass prism. Angle-resolved reflectance is collected by varying θ_e and measuring the intensity of the reflected light relative to a thick (~200 nm) silver reference.	63
4-2	(a) Optical microscope view of AFM tip over region scanned in panel b. (b) AFM topography of MoS ₂ on Ag. The two regions refer to one transfer and two transfers of FL-MoS ₂	64
4-3	(a) PL signal at two regions of FL-MoS ₂ on sapphire growth wafer. The sharp signal provides evidence of few-layer thickness. (b) Raman signal of two regions.	65
4-4	Complex refractive index of FL-MoS ₂ derived from ellipsometry. . . .	67
4-5	Select reflectance spectra: Reflectance is measured via the Kretschmann-Raether method shown in Figure 4-1. The vertical gray lines mark positions of the uncoupled A- and B-excitons. At $\theta_e = 45^\circ$ the plasmon is resonant at 760 nm and the exciton positions are slightly blue-shifted. In the vicinity of the exciton energies ($\theta_e = 52^\circ$), the plasmon completely hybridizes with the excitons. At $\theta_e = 60^\circ$ the plasmon is resonant at 560 nm and the exciton positions are slightly red-shifted.	69

4-6	Dispersion of FL-MoS ₂ -Ag: (a) Experimental energy versus transverse wavevector dispersion obtained from angle-resolved reflectance. Rabi splitting of 81 meV and 93 meV is evident at the A- and B-exciton energies of 1.86 eV and 2.01 eV, respectively. (b) Modeled dispersion of the sample measured in panel a, calculated using the transfer matrix method. See main text for calculation of $k_{//}$ and other quantities.	71
4-7	(a) Experimental dispersion of Ag control. The plasmon asymptotes at a higher energy without the MoS ₂ than with it. (b) Transfer matrix model dispersion of MoS ₂ -Ag with the lossy part of the MoS ₂ refractive index set to zero. This method approximates the behavior of the plasmon absent coupling.	74
4-8	Semiclassical coupled oscillator model fit: Black data are peaks of the three hybrid modes taken from the experimental dispersion. Red data are peaks from the modeled uncoupled plasmon. Orange lines are the coupled harmonic oscillator fit to the black data. The gray lines labeled E_A and E_B are the uncoupled exciton positions, as fit by the model. Anticrossing of the three hybrid modes is a clear indication of a strongly coupled system. The modelled Rabi splittings are 65 meV and 72 meV at the A- and B-exciton energy positions, respectively.	77
4-9	(a) Mixing coefficients for each of the three hybrid mode dispersion branches, as a function of transverse wavevector. The coefficients are obtained from the coupled harmonic oscillator model. (b) Mixing coefficients of the middle branch showing overlap of A- and B-excitons, implying coherent quantum superposition of the two modes, mediated by strong coupling to the plasmon.	79
4-10	Generalized strong plasmonic coupling in 2D TMDC systems. A transfer matrix model is used and the thickness of each film is tuned to give the largest splitting. Rabi splitting is evident for each system except those with Al, where interpretation is more difficult. The color bar is absorptivity.	81

5-1	Absorptivity, A , of MoS ₂ film on 40 nm Ag on glass. Reflectance of film was taken and A calculated as $A = 1 - R$, where transmission is assumed to be negligible.	88
5-2	Band structure and density of states of MoS ₂ , predicted from density functional theory. The arrows labeled A and B point out transitions between nested bands that lead to peaks in the absorption spectrum (and do not refer to the A- or B-excitons). Reprinted with permission from ref [114]. Copyright 2013 by the American Physical Society. . .	89
5-3	Experimental (a) and simulated (b) dispersion in k -space. (a) Stack is NBK7, 40 nm, ~10 nm FL-MoS ₂ . Little variance is seen at the A- and B-exciton positions near 1.8 and 2.05 eV. Avoided crossing is seen at the C-exciton, 2.57 eV. (b) Transfer matrix method simulation of the stack in (a). The model matches the experimental data well, with similar features and magnitude. The gap region is less pronounced and cut off near 3.5 eV less sharp. Absorptivity is calculated as $1 - R$, where R is reflectance.	93
5-4	Coupled oscillator model fit to data. Black data points are from experiment. Red line is simulated uncoupled plasmon used in the model. Orange lines are coupled oscillator model fit to the data. Gray lines are model fit of A-, B-, and C-exciton energy positions.	95
5-5	Transfer matrix model dispersion of bare plasmon with "lossless" MoS ₂ . Same as Figure 5-3b, but with the imaginary part of the complex refractive index of MoS ₂ set to zero. This method of approximating the bare plasmon is not Kramers-Kronig consistent and results in a poor fit of eq 5.1 to the data.	96
5-6	Coupled oscillator model fit, near C-exciton. Fit value of Rabi splitting is $\Omega = 244$ meV or 9.5% of splitting energy, which fits the data well. .	98

5-7	Hopfield/weighting coefficients of coupled modes in terms of uncoupled modes, taken from semiclassical oscillator model fit to experimental data. Branch 1 is the lowest energy branch seen in the dispersion (Figure 5-4) while Branch 4 is the highest branch. Coupling between excitons is marginal.	100
5-8	Experimental (a) and simulated (b) dispersion in angle-space. (a) Stack is NBK7, 40 nm, ~10 nm FL-MoS ₂ . Little variance is seen at the A- and B-exciton positions near 1.8 and 2.05 eV. Avoided crossing is seen at the C-exciton, 2.57 eV. (b) Transfer matrix method simulation of the stack in (a). The model matches the experimental data well, with similar features and magnitude. The gap region is less pronounced and cut off near 3.5 eV less sharp, while the shape of the simulated data compressed slightly in angle-space. Absorptivity is calculated as $1 - R$, where R is reflectance.	102
5-9	Coupled oscillator model fit to data in angle-space. Black data points are from experiment. Red line is simulated uncoupled plasmon used in the model. Orange lines are coupled oscillator model fit to the data. Gray lines are model fit of A-, B-, and C-exciton energy positions. No Rabi splitting is predicted at the A- and B-excitons.	103
5-10	Coupled oscillator model fit, near C-exciton, in angle-space. Fit value of Rabi splitting is $\Omega = 423$ meV. Fit is poor in angle-space.	104
5-11	Rabi splitting extracted from experimental data at bare exciton-plasmon crossing point, in angle-space. Angle space shows massive Rabi splitting of 607 meV or 23.6 % of bare exciton (using fit value of C-exciton, 2.57 eV), while k-space shows 9.5 %. However, the more fundamental characterization of system is in k-space.	105

5-12 Hopfield/weighting coefficients of coupled modes in terms of uncoupled modes, taken from semiclassical oscillator model fit to experimental data, in angle-space. Branch 1 is the lowest energy branch seen in the dispersion (Figure 5-4) while Branch 4 is the highest branch. Coupling between excitons is marginal. Asymmetry of curves is due to angle-space representation. 107

B-1 Absolute value of the in-plane magnitude of the electric field for a silver nanowire 5 μm in length and 40 nm in diameter for (a) a smooth wire (control case), (b) a 40 nm diameter wire tapered to 80 nm at the distal end, (c) a 40 nm diameter wire with an 80 nm diameter silver nanosphere embedded halfway along the length of the wire, and (d) a 40 nm diameter wire with a series of 80 nm diameter silver spheres embedded along its full length, spaced 160 nm center-to-center. The excitation vacuum wavelength is 800 nm. The medium surrounding the wires is vacuum. 123

B-2 Calculated normalized transmitted electric field intensity along converging 40 nm (upper) and 60 nm (lower) diameter wires as the excitation aperture is moved along the x-axis, from halfway between both wires ($x=0$) to above and below both wires ($x=\pm 100$ nm). This intensity is found by averaging the normalized magnitude of the time-averaged Poynting vector in a cylinder 160 nm in radius surrounding the distal half of the wires. The inset shows the geometry. 125

B-3	Absolute value of the in-plane magnitude of the electric field for a silver nanowire 5 μm in length and 40 nm in diameter excited by light of vacuum wavelength (a) $\lambda = 800$ nm, (b) 700 nm, (c) 600 nm, (d) 500 nm, and (e) 400 nm. Note that in this figure, the maximum values of the field of any phase are plotted. In some cases, this helps when comparing various plots. In other cases (i.e. plots in Chapter 3), it is unnecessary to plot the maximum values and plotting instead at a constant phase value helps show the shape of the modes.	126
B-4	Maximum value of the absolute value of the in-plane magnitude of the electric field for a silver nanowire with length of 5 μm and diameter of (a) 20 nm, (b) 40 nm, (c) 60 nm, and (d) 100 nm. The excitation vacuum wavelength is 800 nm.	126
B-5	Absolute value of the in-plane magnitude of the electric field for a (a) silver and (b) gold nanowire 5 μm in length and 40 nm in diameter. The excitation vacuum wavelength is 800 nm.	127
C-1	Strong exciton–plasmon coupling at the C-exciton. The dispersion data taken for this figure are from a distinct region of the same sample used for Chapter 5. Strong coupling of similar magnitude to that shown in Chapter 5 is evident.	131
C-2	Strong exciton–plasmon coupling at the A- and B-excitons. The data are taken from a sample similar, but distinct, to that of Chapter 4. Note that the x -axis is the angle of incidence in degrees, rather than wavevector.	132
C-3	Strong exciton–plasmon coupling at the A- and B-excitons. The data are taken from a sample produced from a soda lime glass prism, 40 nm Ag, and chemically exfoliated MoS ₂ that was airbrushed (sprayed) onto the Ag. Note that the x -axis is the angle of incidence in degrees, rather than wavevector.	133
C-4	134

C-5	Transfer matrix model simulations of strongly coupled MoS ₂ -Ag on different substrates and in different media. Panels a) and c) show the dispersion using substrate with refractive index of 1.5, in air and water, respectively. Panels b) and d) show the dispersion using substrate with refractive index of 2.0, in air and water, respectively. Use of the high index substrate is necessary when performing experiments in aqueous media. Note that the dispersion is shown with angle in degrees on the x -axis.	136
C-6	Strong coupling at A- and B-excitons in ~10 nm MoS ₂ on 40 nm Ag on high index glass (SF11, refractive index of 1.8).	137
C-7	Strong coupling at A- and B-excitons in ~10 nm MoS ₂ on 40 nm Ag on high index glass (SF11, refractive index of 1.8), with the MoS ₂ side submersed in aqueous buffered phosphate solution. The modes are shifted to higher k	138
D-1	Cross section representation of embedding metal nanopatterns (EMN) in solar absorber material. The substrates are glass coated with a transparent conductive oxide (TCO) thin film. A thin film of solar absorber is deposited on the TCO. A metal nanopattern of Ag is formed next, using nanosphere lithography. The nanopattern is embedded by depositing another layer of absorber material. Finally, an opaque Ag back reflector is deposited.	141
D-2	Topography of aSi control film consisting of device without metal nanopattern, measured by atomic force microscopy (AFM) before deposition of back reflector.	142
D-3	Image of aSi control film consisting of device without metal nanopattern, measured by scanning electron microscopy (SEM) before deposition of back reflector.	143

D-4	SEM image of aSi device after nanopatterning, before second deposition of aSi. The array of holes in Ag is clearly visible, with 500 nm pitch. The metal is 20 nm thick.	144
D-5	Absorbance (defined as $1 - \text{transmittance} - \text{reflectance}$) of aSi devices. The brown and red lines shows two nominally identical devices with EMN. The orange line shows a control with embedded metal layer (EML), where a metal thin film is deposited during the same run as the EMN devices, but no nanopatterning is performed. The purple line shows a control with no embedded metal and same overall thickness of aSi. The blue line shows a device with 500 nm aSi, which is comparable to a conventional thin film aSi solar cell.	145
D-6	SEM image of CdTe device after nanopatterning, before second deposition of CdTe. The array of holes in Ag is clearly visible, with 500 nm pitch. The metal is 30 nm thick.	147
D-7	Absorbance of 100 nm CdTe with EMN vs. control without EMN. The x-axis is plotted up to the band gap of CdTe. Total absorbance of the EMN film is 93 % and control film is 86 %, leading to a 8 % absorption enhancement with EMN. At the band edge, the enhancement is 16 %.	148
D-8	Absorbance of 150 nm CdTe with EMN vs. control without EMN. The x-axis is plotted up to the band gap of CdTe. Total absorbance of the EMN film is 94 % and control film is 92 %, leading to a 2 % absorption enhancement with EMN.	149
D-9	Simulated absorbance of 100 nm CdTe devices with and without EMN. The dashed lines show absorption in the CdTe only. The balance of absorption may be due to absorption in the EMN or metal back reflector or FTO. The EMN device has overall absorbance 88 %, while the control has 75 %. Considering absorption in the CdTe, the EMN device absorbs 78 % and the control 66 % of incident light. The data indicate little loss occurs in the Ag EMN.	150

D-10	Power loss density in cross section of simulated CdTe EMN device. 800 nm light is incident from $-z$. Polarization is in y . The presence of resonant modes is visible due to the nanopattern.	150
E-1	SEM of 980 nm crosses device. The stripe down the center of the image is the aSi sample region. The lighter thick lines at the top and bottom are contact pads to the aSi. The lighter line that starts on the right of the image, makes a cross, and then a large square under the aSi is a gate and also serves as a back reflector, helping define the optical modes. The darker region that spans the gate vertically except for the side edges is the nanostructured region. The aSi is sandwiched between the gate on bottom and the nanostructure on top, with 10 nm alumina between each layer. The contact pads directly contact the aSi.	154
E-2	SEM of 980 nm crosses device showing the nanostructures on 10 nm alumina on aSi. A scratch can be seen across the middle and left regions.	155
E-3	SEM of 1550 nm crosses device. Similar to the previous figure, but turned counter-clockwise 90° , with 1550 nm crosses pattern.	156
E-4	SEM of 1550 nm crosses device showing the nanostructures on 10 nm alumina on aSi.	156
E-5	SEM of 1550 nm lines device. Similar to the previous figure, but zoomed out further, with 1550 lines pattern.	157
E-6	SEM of 1550 nm lines device showing the nanostructures on 10 nm alumina on aSi.	157
E-7	Reflectance of 980 nm crosses device. The red line is the simulated device, showing actual resonance as a dip near 1,050 nm. The blue data is from an Ocean Optics (OO) spectrometer. NOTE: OO data for wavelengths longer than 1,100 nm is invalid. The black data is from unpolarized FTIR. A broad resonance near 1,300 nm is picked up by the FTIR and matches the simulated spectrum with a slight shift. . .	158

- E-8 Absorbance of 1550 nm crosses device. The red line is the simulated device, showing actual resonances as peaks near 1,250 nm, 1,500 nm, and 2,00 nm. The blue data is from an Ocean Optics spectrometer. The black data is from unpolarized FTIR. From the FTIR data, peaks appear at 1,200 nm, 1,400 nm, and 1,700 nm, matching the FTIR data well, although shifted. In addition to matching of the peaks (albeit shifted), the dips also match well (again, shifted). 159
- E-9 Reflectance of 1550 nm lines device. The red line is the simulated device, showing actual resonance as dips near 950 nm and 1,350 nm. The black data is from an Ocean Optics (OO) spectrometer. NOTE: OO data for wavelengths longer than 1,100 nm is invalid. The blue data is from unpolarized FTIR, while the green data is from polarized FTIR, tuned to minimize the dip near 1,300 nm. From the FTIR data, the simulated 1,350 nm resonance is shifted slightly to 1,300, but matches shape well. 160

List of Tables

5.1	Best fit parameters from coupled oscillator model for strong coupling to the C-exciton in MoS ₂ . Solved in k -space.	97
5.2	Best fit parameters from coupled oscillator model for strong coupling to the C-exciton in MoS ₂ . Solved in angle-space.	106
E.1	Parameters of fabricated nanostructures.	153
F.1	Transfer matrix model MATLAB scripts' dependencies.	162

Chapter 1

Introduction

1.1 Background and motivation

Surface plasmon polaritons (SPPs) are collective oscillations of free carriers in a material at their interface with a dielectric, driven by coupling to light. One of the unique features of SPPs is that they can concentrate light into subwavelength dimensions. This behavior alone has been exploited for non-diffraction limited optics, ultra-thin film solar cells, surface enhanced Raman scattering, photothermal cancer cell destruction, photocatalysis, enhanced photoluminescence, and strong exciton–plasmon coupling for modification of chemical decay pathways and reaction rates and polariton condensation. In each case, SPPs interact with, or couple to, their environment, transferring energy to it. This coupling, and therefore the different phenomena and applications, can be categorized by its strength on a spectrum from weak to strong coupling, with distinct physics that manifest in the different regimes.

In this dissertation, I will work through three research projects that span the different coupling regimes, characterized by a parameter called the coupling strength, g . In Chapter 3, coupling between quasi-one-dimensional plasmonic nanowires is minimized ($g \rightarrow 0$) to model a technique to achieve sub-diffraction resolution imaging. Chapter 4 analyzes exciton–plasmon strong coupling in quasi-two-dimensional semiconductors. Here, strong coupling occurs when the coupling strength is greater than the decay widths of the excitons and plasmons, $g > \Gamma_{ex}, \Gamma_{SPP}$, and is typically on

the order of a few percent of the photon energy, $g \sim 0.01 E_{\text{photon}}$. In this regime, new states emerge and quantum coherent phenomenon such as polariton condensation can be observed. Finally, in Chapter 5, in a variation on the system of Chapter 4, the coupling strength is increased toward $g \rightarrow 0.1 E_{\text{photon}}$, which is called the ultrastrong coupling regime. Here, the coupling outcompetes other decay mechanisms and quantum coherent effects become efficient enough for potential commercial applications, such as quantum computing.

The next sections of this chapter will review plasmonics and coupling in further detail. Chapter 2 will describe the methods used in Chapters 3–5. Chapter 6 will summarize my findings.

1.2 Plasmonics

1.2.1 Properties

Surface plasmons are collective oscillations of free charge carriers in a material at their interface with a dielectric. Surface plasmons can be compared to ocean waves. The energy of ocean waves is carried by the up and down motion of the water molecules at the ocean surface, while the propagation direction of the wave is parallel to the water surface. In other words, ocean waves are transverse waves. They can be created by applying a displacement force to some of the water molecules, for example by dropping a stone into the water. Just after the stone makes contact with the surface, the water beneath it is displaced downward. A restoring force pushes these molecules back upward, but not before the wave spreads out, causing adjacent molecules to also move downward. In this way, a wave ripples outward from the stone. Surface plasmons can be excited similarly by a displacement force such as from a transient or oscillating electric field, which will drive the charge carriers. As an electromagnetic wave with oscillating electric field, light can be used to excite surface plasmons. When light is coupled into surface plasmons, the resultant quasi particles are called surface plasmon polaritons. [1]

The following subsections follow the formalism of references [2] and [3], where the properties of plasmonics are studied mathematically through the plasmonic dispersion relations. The reader is referred to those references for a full accounting of the problem.

1.2.2 Dispersion relation

Surface plasmons confine light at the interface between a metal (or any medium with free carriers) and a dielectric. If we solve Maxwell's equations at such an interface, assuming semi-infinite media and p-polarized light, we get the wave vector of the confined mode, which propagates along the surface and exponentially decays away from the surface. The dielectric medium is described by the real positive dielectric function ϵ_2 and the metal described by complex $\epsilon_1 = \epsilon'_1 + i\epsilon''_1$. The wave vector of the mode is given by eq 1.1, where the direction of propagation is along x .

$$K_x = \frac{\omega}{c} \left(\frac{\epsilon_1 \epsilon_2}{\epsilon_1 + \epsilon_2} \right)^{\frac{1}{2}} \quad (1.1)$$

For most metals of interest, the lossy component of ϵ_1 is smaller than the real part: $\epsilon''_1 < |\epsilon'_1|$. We then write out the complex wave vector, $K_x = k_x + ik''_x$, as:

$$k_x = \frac{\omega}{c} \left(\frac{\epsilon'_1 \epsilon_2}{\epsilon'_1 + \epsilon_2} \right)^{\frac{1}{2}} \quad (1.2)$$

$$k''_x = \frac{\omega}{c} \left(\frac{\epsilon'_1 \epsilon_2}{\epsilon'_1 + \epsilon_2} \right)^{\frac{3}{2}} \frac{\epsilon''_1}{2(\epsilon_1)^2} \quad (1.3)$$

For most metals in the optical region where surface plasmons are present, $\epsilon'_1 < 0$ and $|\epsilon'_1| > \epsilon_2$, which keeps k_x real. [2]

1.2.3 Drude response

The response of free carriers can be described as a free electron gas, as described by Drude. The Drude form of the dielectric function can be written as shown in eq 1.4, where ω_p is the bulk plasma frequency. [3]

$$\epsilon = 1 - \frac{\omega_p^2}{\omega(\omega + i\gamma)} \quad (1.4)$$

Assuming losses, γ , are small compared to the frequency ω and taking the real part, we can simplify the Drude model to:

$$\epsilon = 1 - \left(\frac{\omega_p}{\omega}\right)^2 \quad (1.5)$$

1.2.4 Dispersion plot with Drude response

If we plug eq 1.5 into 1.2 for ϵ'_1 , we can plot the dispersion of mode energy versus mode wavevector, shown in Figure 1-1. Here we use $\omega_p = 1.35 \times 10^{16}$ for silver [4] and take $\epsilon_2 = 1$ for air.

There are a few features of Figure 1-1 to point out. First, the dispersion is split into two branches. The upper branch lies within the light cone, defined by the area to the left of the light line, shown as the dashed line. Incident free photons have a maximum momentum described by $\omega = ckn$, where c is the speed of light, k is the wavevector (proportional to the momentum), and n is the index of refraction of the propagation medium. Thus, photons propagating in the plane of the plasmon have the momentum described by the light line, whereas photons propagating normal the plasmon have no momentum in that direction. There are two consequences to this: 1) the SPP upper branch, lying within the light cone, may couple easily to photons and therefore is a lossy mode that does not support propagating plasmons, 2) the lower branch at higher energies may not interact with light due to momentum mismatch, therefore it is impossible, in theory to couple light to these SPPs. In reality, surface roughness relaxes the momentum constraint and allows for excitation of SPPs on bulk metal films by light. In fact, the strength of the surface plasmon mode, for example from observing dips in reflection measurements, can be a proxy for thin film quality—the rougher the surface, the stronger the SPP modes. Other methods to excite SPPs more controllably are described in the next section.

Other features to note in Figure 1-1 are the energies indicated by the horizontal

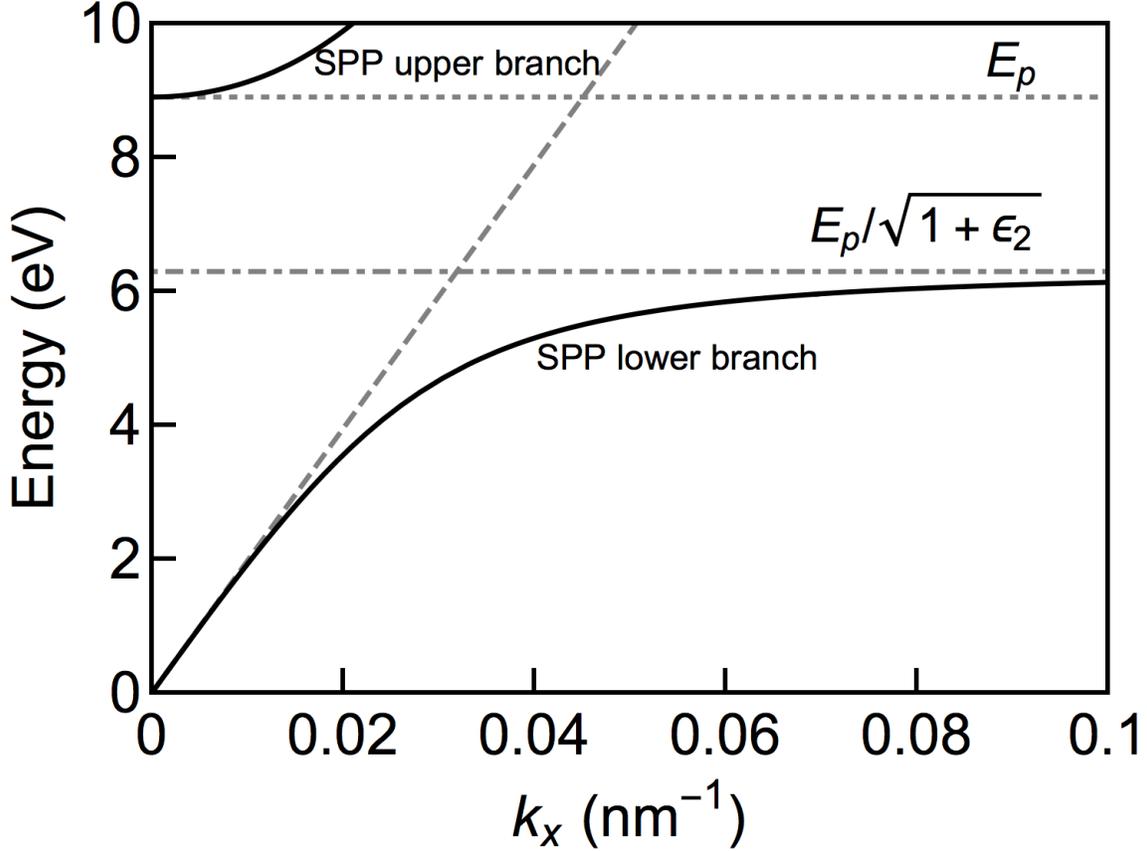


Figure 1-1: Surface plasmon polariton dispersion at the interface between bulk silver (Drude model) and vacuum. k_x is the wavevector parallel to the surface and the direction of plasmon propagation. The upper branch minimum is at the bulk plasma energy (dotted gray line) while the lower branch approaches monopole surface plasmon energy (dot-dashed gray line). The upper branch is radiative and does not support a bound plasmon. The lower branch lies to the right of the light line (dashed gray line) and is therefore a non-radiative, bound mode.

lines. E_p is bulk plasmon energy, typically given in *cgs* units:

$$E_p = \hbar\omega_p = \hbar\sqrt{\frac{4\pi n e^2}{m_0}} \quad (1.6)$$

where n is the free carrier density and m_0 is the electron mass. [1, 2]. The bulk plasma frequency defines the mode where the entire electron (or free carrier, generally) population oscillates longitudinally like a sound wave throughout the bulk. The next line shows the surface plasmon energy. Generally, when plasmonics is discussed in the optics field, we use the words plasmons or surface plasmons and are

really referring to surface plasmon *polaritons*, i.e. the coupled light-plasmon mode. The dot-dashed line shows the uncoupled surface plasmon mode, which is an allowed mode even in the absence of light. [1, 5] Interestingly, the SPP mode asymptotically approaches $E_{SPP} = E_p/\sqrt{1 + \epsilon_2}$, or $1/\sqrt{2}$ for a metal-air interface. Note that the similar resonant condition for a metallic sphere in air is a localized surface plasmon resonance (LSPR) $E_{LSPR} = \sqrt{3}$, which can be found from Mie's theory.

1.2.5 Dispersion plot with experimental response

The Drude model of the permittivity often fits data in the visible poorly To get more accurate estimates of surface plasmon dispersion, we can plug the experimentally determined (e.g. by ellipsometry) dielectric function directly into eq 1.2 and plot the real part $Re(k_x)$. Figure 1-2 shows this using the common Johnson and Christy dataset. [6]

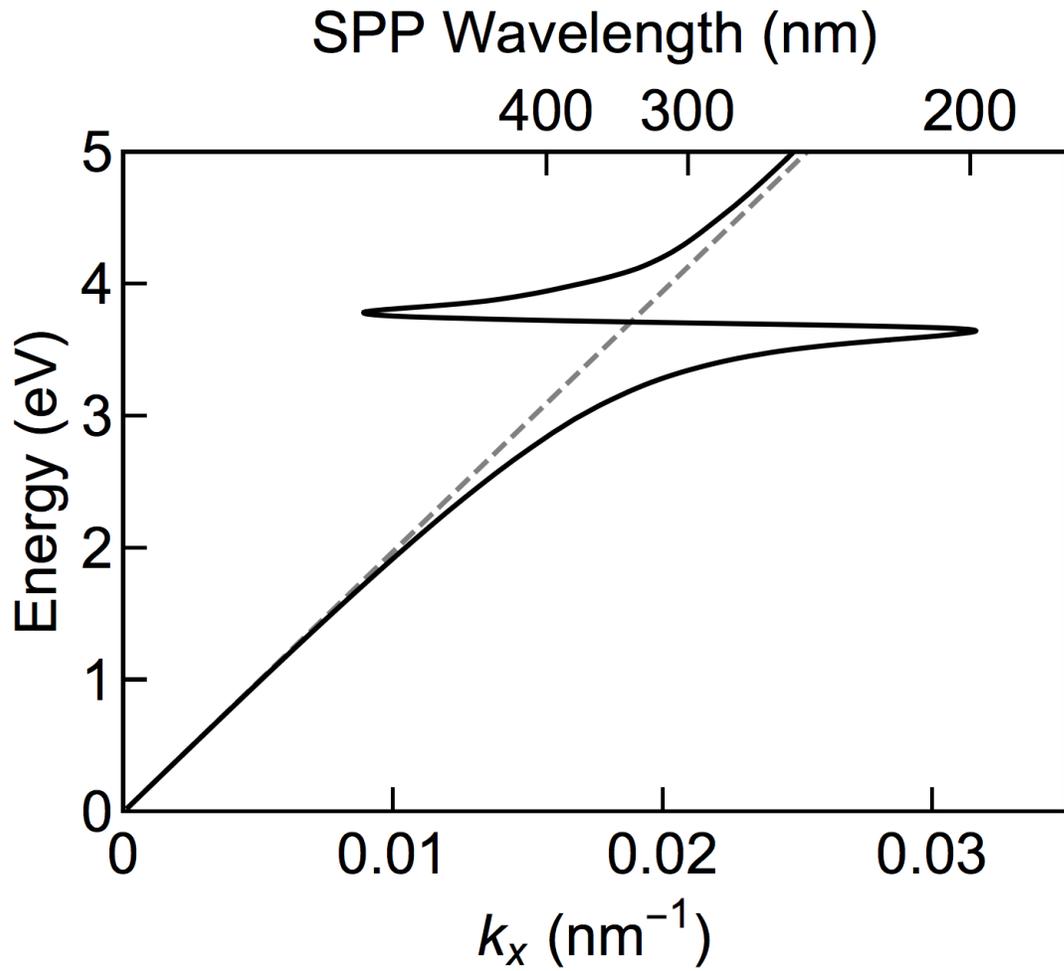


Figure 1-2: Surface plasmon polariton dispersion at the interface between bulk silver and vacuum, using experimental values of the silver dielectric function. The gap between the plasmon branches in Figure 1-1 is reduced and replaced with a real solution with negative group velocity. At higher k_x , the plasmon becomes more confined, exhibiting shorter effective wavelength.

In Figure 1-2, the features are clearly moved to lower energy, with the forbidden band between ~6.2 and 8.9 eV seen in Figure 1-1 replaced by a continuous real solution to k_x . This region has negative group velocity and may be interesting for metamaterials with negative index, and is called the quasibound region. The plasmon is only allowed (or "bound") for finite values of k_x , below the light line, compared to the theoretical asymptotic behavior. The surface plasmon energy at the maximum value of k_x is 3.66 eV.

1.2.6 Plasmon length scales

The spatial extent of surface plasmons is of great interest. Figure 1-2 indicates that the wavelength associated with the surface plasmon, $\lambda_{SPP} = 2\pi/k_x$, is reduced from that of free space. For example, at $k_x = 0.032$, the plasmon energy is 3.66 eV or 340 nm vacuum wavelength, while the $\lambda_{SPP} = 200$ nm, thus the light is squeezed along x . The spatial extent into the metal and dielectric is plotted in Figure 1-3, for silver and silicon dioxide (glass).

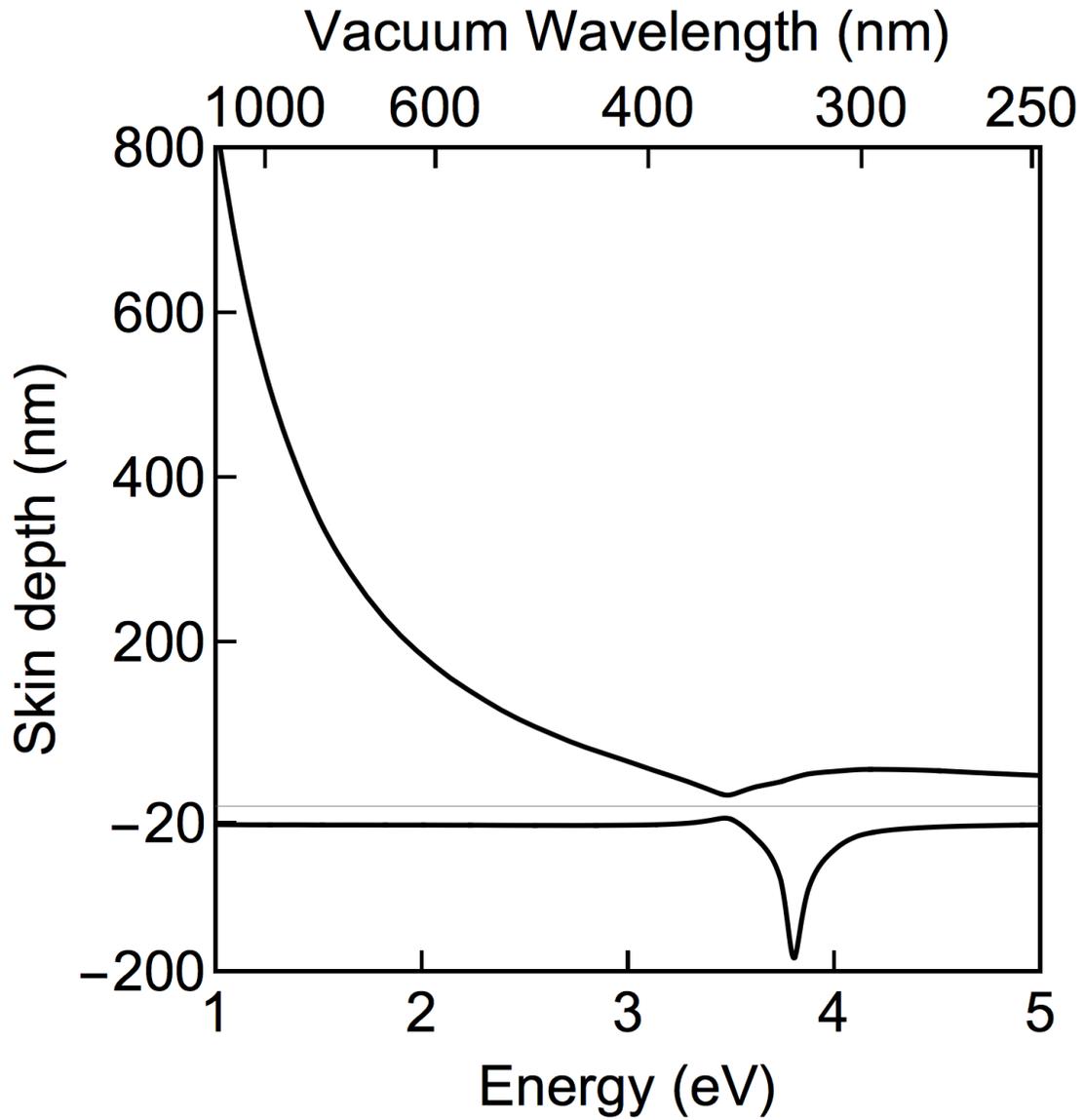


Figure 1-3: Skin depth of plasmon into Ag and SiO₂ at the interface of semi-infinite Ag and SiO₂. Ag refractive index from Johnson and Christy [6]. SiO₂ modeled with constant refractive index $n = 1.45$.

Here, we see that the penetration depth of the exponentially decaying field is only ~ 20 nm in the metal, but varies widely in the dielectric. At resonance of the SP mode, i.e. when the mode is most bound, the skin depth in the dielectric reaches a minimum of 15 nm. At lower energies, the plasmon approaches the light line and becomes more light-like in nature, thus the skin depth in the dielectric approaches the free space value.

Finally, we discuss the propagation lengths of the SPPs at a silver-silicon dioxide interface, plotted in Figure 1-4.

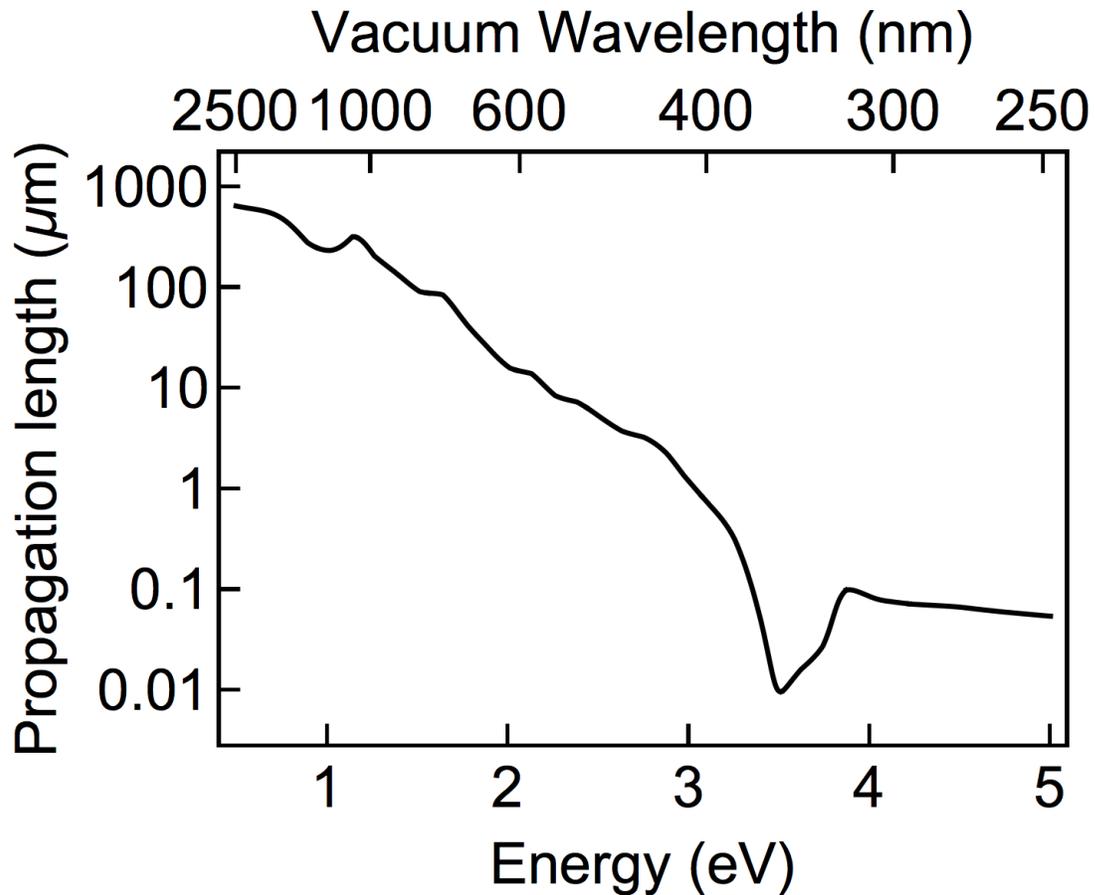


Figure 1-4: Propagation length of surface plasmon at the interface of semi-infinite Ag and SiO_2 . Ag refractive index from Johnson and Christy [6]. SiO_2 modeled with constant refractive index $n = 1.45$.

Figure 1-4 shows that the propagation lengths of SPPs may be only ~ 10 nm at the SP mode energy or microns in the infrared. Propagation length gives a sense of the degree of loss of plasmonics and is hugely important in many applications. For example, the holy grail of photonics is to replace integrated circuits (ICs) based on metal with photonic ones, the idea being that computing based on conventional metal interconnects has reached a point of diminishing returns for running chips faster because losses in the metal become large. Free space light is lossless, hence the desire to process information with photons. However, the wavelength of free space light is large compared to IC dimensions, e.g. the modern transistor is ~ 10 nm long, so plasmonics has been invoked due to its ability shrink light. A study of Figures 1-3 and 1-4 shows that there is trade off between confinement and propagation length.

1.2.7 Dispersion in thin films

We now consider a thin metal film sandwiched by two semi-infinite dielectrics. Such a film will support SPPs at both interfaces, similar to that of the semi-infinite slab case. For thin enough films, however, the SPP field penetration depth into the metal can be larger than thickness of the film, thus the geometry of the system will modify the modes. Figure 1-5 shows the two allowable transverse SPP modes in such a film, an antisymmetric mode (L+) and a symmetric mode (L-). The antisymmetric mode can be thought of as ripples in the electron density in the film such that a peak in electron density on one side of the film means a valley on the other side. The symmetric mode is also called the breathing mode as the carriers at given point in the film move out toward the interface and back again.

Figure 1-5 shows the thin film modes for different thicknesses of Ag sandwiched in SiO_2 . For thicker films, e.g. 50 nm, the two modes lie on top of each other and will likely be experimentally indistinguishable. For thinner films, the symmetric mode resonance energy remains constant while the antisymmetric mode blue shifts with decreasing thickness, increasing the possibility of observing the two modes spectrally.

Note that the solution of Maxwell's equations for the thin film case has no analytic solution and must be solved numerically. [3]

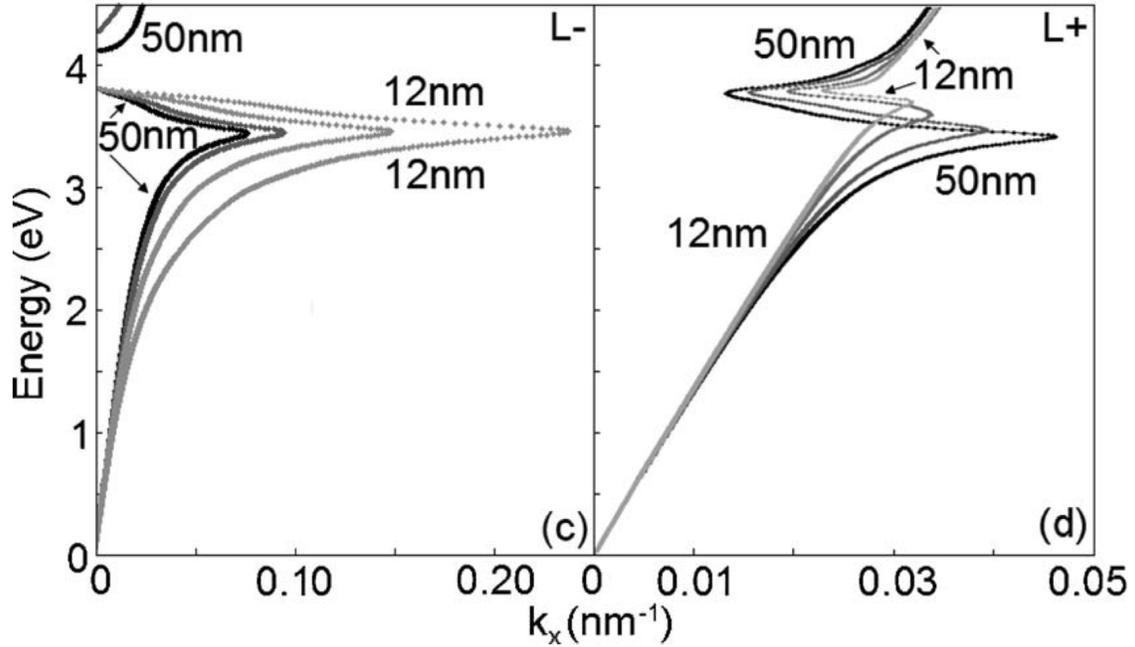


Figure 1-5: Dispersion of plasmon of thin film Ag sandwiched between semi-infinite layers of SiO_2 . Dispersion is numerically calculated using experimental optical properties. Symmetric thin films exhibit symmetric (L^-) and antisymmetric (L^+) modes, shown in the left and right panels, respectively. For films ~ 50 nm and greater, the two modes overlap, while they deviate at lower thicknesses. Reprinted figure with permission from Dionne, J. A., Sweatlock, L. A., and Atwater, H. A. *Phys. Rev. B* 72, 075404 (2005). Copyright 2005 by the American Physical Society.

1.2.8 Plasmon excitation methods

We now turn to the problem coupling incident light to SPPs. As discussed earlier, in theory free space light cannot excite SPPs at bulk metal/bulk dielectric interface due to lack of momentum, shown by the light line lying to the left of the bound SPP branch in Figures 1-1 and 1-2. A method is needed to generate higher k -components that will intersect the plasmon dispersion.

One common way to couple to plasmons is by using nanoparticles with 5-100 nm dimensions. At the smallest dimensions, very high k -components can be generated due to the Heisenberg uncertainty principle $\Delta x \Delta p \geq \hbar/2$, where the wavelength of light is the spatial component and the wavevector the momentum component. Larger nanoparticles that are still subwavelength in dimension can be thought of

as nanoantennas. In either view, the electric field of light can couple to the dipole plasmon resonance of the particle. The inverse of a nanoparticle is a slit in a metal film, which can also act as an antenna. Thus, a single slit or groove nanopatterned into either bulk metal or a metal thin film is sufficient to couple free space light to SPPs. [7]

The Heisenberg principle view shows us that any nanoscopic metallic feature may generate plasmons. Therefore, surface roughness may generate plasmons, as previously mentioned. Surface roughness may be thought of as similar to dispersed nanoparticles on a flat metal film. In fact, recently electrocatalytic reduction of CO_2 to CO and H_2 on Ag electrodes was enhanced by simply shining light at the electrode which excited surface plasmons from the roughness. The plasmons were implicated in the enhancement mechanism. [8] Surface roughness-generated plasmons is also used in surface enhanced Raman scattering (SERS), where the plasmons enhance the Raman signal through their concentration of light.

One may also take advantage of nanoscale probes to generate plasmons. Near-field scanning optical microscopy (NSOM, also known as scanning near-field optical microscopy or SNOM) uses metal-coated nanoprobes coupled to optical fibers to generate and detect near-field light. Near-field light has high- k components and so can be used to generate plasmons on a surface. The same effect is used in tip-enhanced Raman scattering (TERS), where a nanoprobe concentrates light via plasmons at its tip to enhance the Raman signal.

Another useful way to study plasmons is with electrons. High energy electron beams provide a source of electrons with kinetic energies orders of magnitude higher than plasmonic resonances with huge momenta compared to that of surface plasmon polaritons, which have the small momenta associated with photonic particles. The initial plasmonics experiments used electron energy loss spectroscopy (EELS) to study the resonances in thin metals films and identified the bulk and surface plasmon modes. [9] Interestingly, another surface mode was also identified, the higher energy dipole surface plasmon. [10] The usual surface plasmons can be thought of as monopole in nature, being a ripple in the electron sea at the metal surface. In the dipole plasmon,

only electrons at the surface are displaced up or down, while the sea below stays stationary, such that a local displacement of electrons above the metal surface is accompanied by a lack of electrons just below. Thus, in these locations a dipole is formed across the interface. EELS can also be used to probe localized surface plasmon resonances in nanostructures [11] as can cathodoluminescence. [12]

One may also be more controlled in the components of k that are coupled to by employing period gratings. If light with wavevector k is incident on a bulk metal film with angle θ , the k -component generated parallel to the surface is $k_x = k \sin(\theta)$. By adding a periodic grating to the surface, for example by milling subwavelength lines with a focused ion beam (FIB), additional k_g may be added according to $k_g = mg$ where $g = 2\pi/a$, m is an integer referring to the different possible grating modes, and a is the spacing of the lines. The wavevector now becomes $k_x = k \sin(\theta) \pm mg$ and the appropriate choice of a can lead to large enough momentum to couple to plasmons.

Efficient coupling to plasmons in planar metal films can be achieved without relying on nanostructure to generate high- k components. Consider that the wavevector of light scales with the refractive index, n , of the propagation medium, $k = k_0 n$, where k_0 is the vacuum wavevector. Naively, we may just try to increase the refractive index of the dielectric. However, the plasmon behavior depends on its dielectric environment and the dispersion of Figures 1-1, 1-2, and 1-5 will shift to keep the light line to the left of the bound plasmon. However, we could imagine shining light at an angle through a high index dielectric, glass for example. At high enough angles the light will be totally internally reflected (TIR) from the opposite face. TIR light has no real part of the transmission coefficient. There is, however, an imaginary component and thus at the surface of the glass exists an evanescent wave. We could then bring this surface to within the near-field of a plasmonic metal surface in air and the high- k evanescent wave would be able to generate plasmons on the metal. Such a design is called the Otto configuration.

The Otto configuration is useful for studying surface plasmons in bulk materials, however it can be difficult to control the tens of nm gap needed between the glass and plasmonic surfaces for experiments in the visible spectrum. The Kretschmann-

Raether (K-R) method is similar to the Otto configuration but easier to implement. Consider again the dispersion a thin Ag film embedded in SiO₂ (glass), shown in Figure 1-5. Here, the light line in glass is to the left of the plasmon. What happens if one side of the Ag is replaced with air? Figure 1-6 shows the dispersion of the glass / 50 nm Ag / air case.

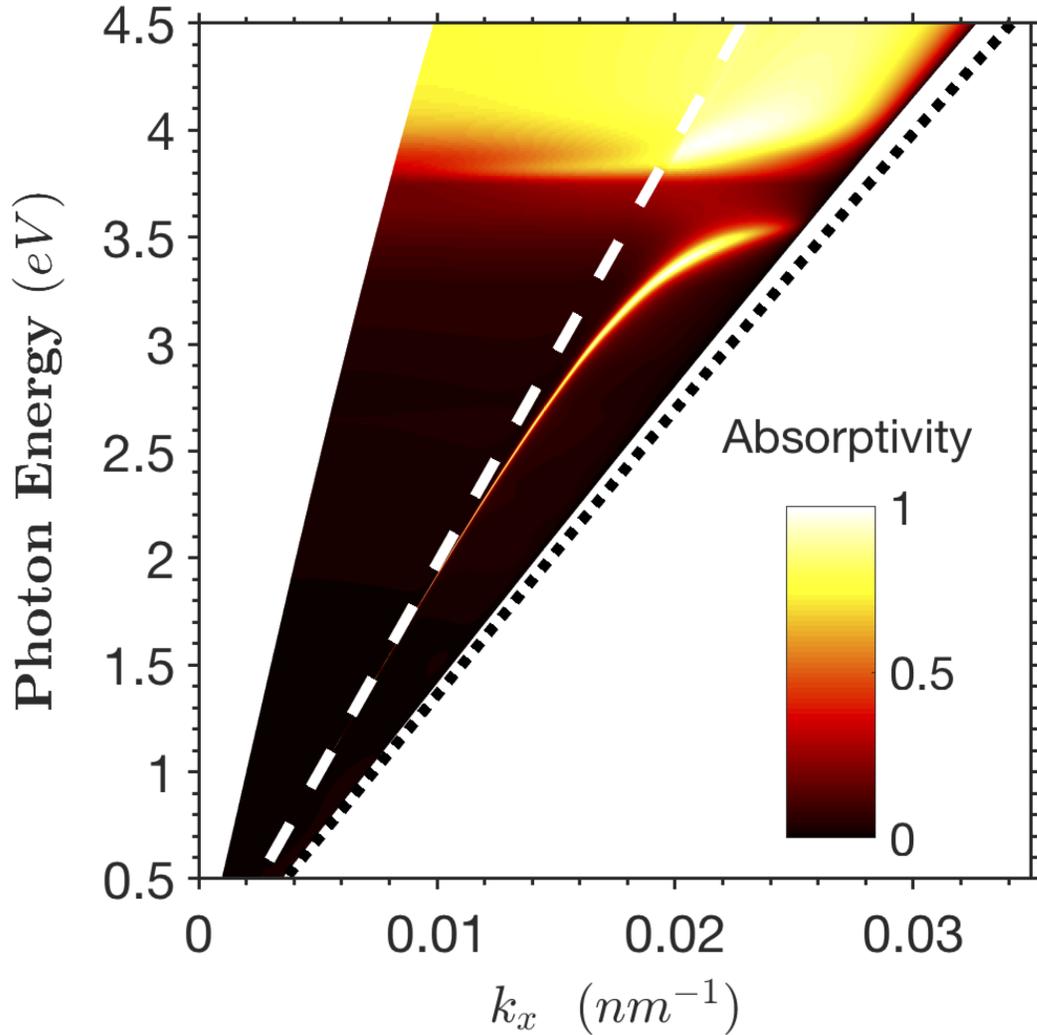


Figure 1-6: Dispersion of plasmon in 50 nm thin film Ag on semi-infinite fused silica (glass), in air. Experimental refractive indices are used in transfer matrix modeling. [13] P-polarized plane waves are excited from the glass side. Absorptivity A , is calculated as $A = 1 - R$, where R is reflectivity. Transmission is assumed zero due to total internal reflection conditions. White dashed line indicates the light line in air. Black dotted line is light line in fused silica.

The light line in glass (black dotted line) and air (white dashed line) are shown with the dispersion in Figure 1-6. We can see that the dispersion still follows the light line in air. Now, however, the light line in glass is to the right of the plasmon. Therefore, we can imagine reflecting p-polarized light off the Ag surface from the glass side where it has enough momentum to couple through the thin film to the air/Ag plasmon. This is how the K-R method works. Further details of implementation are given in the Experimental section 2.2 of Chapter 2.

Note that Figure 1-6 was made using the transfer matrix method of modeling the optical scattering properties of stacks of thin films based on the Fresnel equations. This method provides an analytical way to view the plasmon modes by studying the absorption properties (equivalently, the reflection and transmission properties) of the system. Further details are given in the Theory section 2.3 of Chapter 2.

A question arises looking at the dispersion of Figure 1-6: if the plasmon follows the light line in air, is there an equivalent mode to the right of the light line in glass? It turns out there is, and it is inaccessible in the usual K-R configuration because, again, higher k -values are needed to access the mode. Ref [14] numerically solved the glass/Au/air case, which is very similar to that with Ag, and found that there are two modes (symmetric and anti-symmetric) at each interface. Thus, breaking the symmetry of the system reveals another set of modes that were degenerate in the symmetric case. In Figure 1-5, the two interfaces are degenerate, so only two modes are seen. The modes at the glass/Ag aren't seen in Figure 1-6 because the transfer matrix method only models modes accessible with plane waves.

We can however imagine a way to probe such modes experimentally using a combination of the K-R and Otto methods. Imagine, for example a glass substrate (fused silica, $n = 1.46$) coated with 50 nm Ag, in air, as studied in Figure 1-6. Now, if we approach the Ag surface from the air side with a high index glass (SLAH-79, $n \simeq 2$) to within 100 nm and reflect plane waves under TIR from this surface, these waves should have enough k to couple to both sides of the Ag film. Indeed, this appears to work, as shown in Figure 1-7.

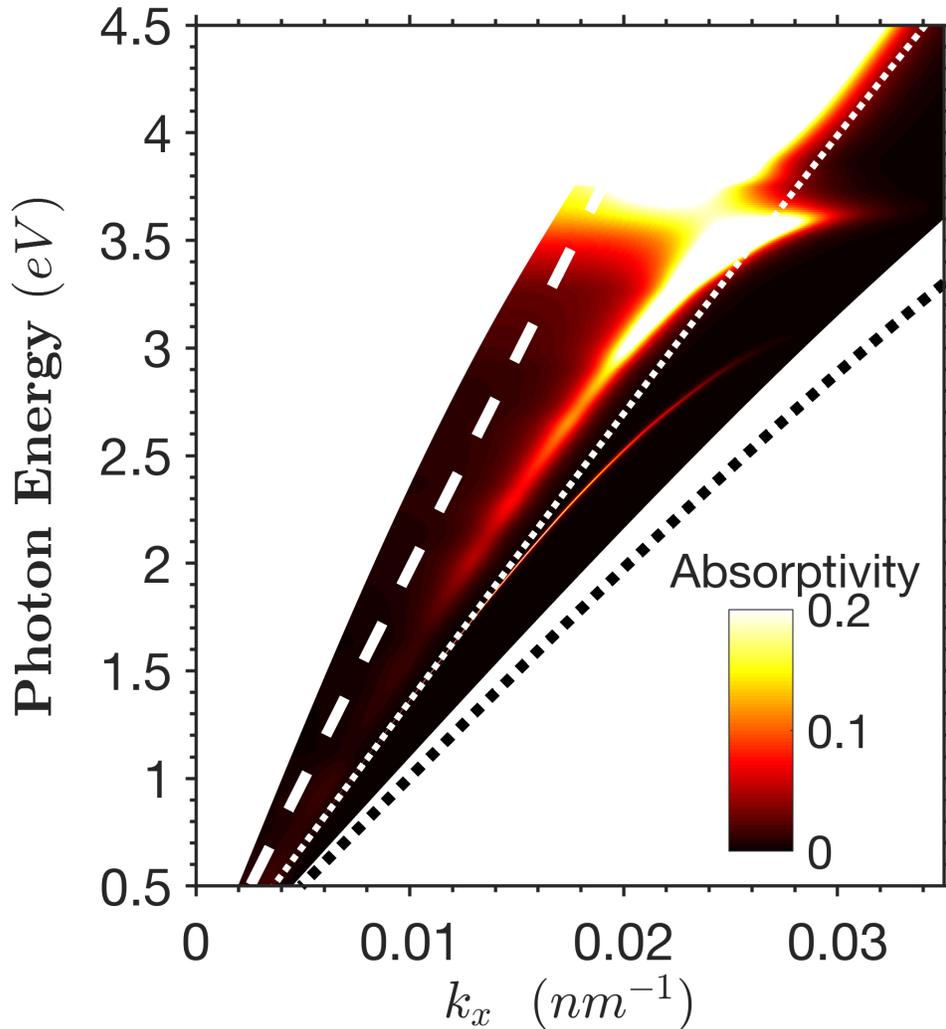


Figure 1-7: Dispersion of two interfacial plasmon modes in 50 nm thin film Ag. As in Figure 1-6, Ag is on semi-infinite fused silica (low index glass, refractive index $n = 1.46$), in air. However, in this model, light is shined through a semi-infinite slab of high index glass (SLAH-79, refractive index $n \simeq 2$) which is 100 nm above the Ag surface, so that air fills the gap. The high k -components of the evanescent field at the surface of the high index glass may couple to modes at the fused silica / Ag and Ag / air interfaces, allowing both modes to be seen in the dispersion. White dashed line indicates the light line in air. White dotted line is light line in low index glass. Black dotted line is light line in high index glass. Modeling is done similar to that in Figure 1-6.

In Figure 1-7, the light lines in air (white, dashed), in low-index fused silica (white, dotted), and in high index SLAH-79 (black, dotted) are shown. Two plasmon modes are seen, one associated with the air/Ag interface, which is the one studied in the earlier figure, and one with the fused silica/Ag interface. A couple features are interesting. First, the plasmon on the glass side has a much sharper resonance. Second, the plasmon at the air interface is now shifted to the right due to the local dielectric being an effective medium between the air and high index glass. Finally, the strong dispersion of the high index glass shows up as curvature at the bounds of the color plot.

1.3 Theory of quantum emitter–optical cavity coupling

To understand light-matter coupling, we consider a coupled oscillator model where one oscillator is a single quantum emitter, i.e. a two-level system such as an exciton found at the band edge of a semiconductor. The other oscillator is a photonic resonance, usually taken to be a cavity resonance where the allowed modes in a double Bragg mirror, superlattice structure, or photonic crystal have very high $Q = \frac{\hbar\omega}{\Delta\omega} \sim 10^3 - 10^7$. [15] However, the photonic resonance may take any form that co-locates light with the other emitter, such as a plasmonic structure. [16] High Q is also not necessary; the coupling depends on the line widths/decay rates of the modes involved. We'll see the effect the line widths has on the coupling below.

The solution to the equation of state of a system of damped driven coupled harmonic oscillators has normal modes given by eq 1.7. [16, 17]

$$\Delta\omega = -i \left(\frac{\gamma + \kappa}{4} \right) \pm \frac{1}{2} \sqrt{(2g)^2 - \left(\frac{\gamma - \kappa}{2} \right)^2} \quad (1.7)$$

In eq 1.7, $\Delta\omega$ is the change in radial frequency of the coupled modes from their resonance frequency, γ is the line width of the emitter, κ is the line width of the cavity mode, and g is the coupling strength. Evidently, for $g = 0$, $\Delta\omega$ is purely imaginary

and collapses to $\Delta\omega = -i\left(\frac{\gamma+\kappa}{4}\right) \pm i\left(\frac{\gamma-\kappa}{4}\right) = -\frac{i}{2}(\gamma, \kappa)$. The line widths of the coupled modes are given by $\gamma', \kappa' = -2\text{Im}(\Delta\omega) = \gamma, \kappa$. Therefore at $g = 0$, the line widths are unmodified. When the term under the square root becomes real, the coupled mode line widths collapse to a single value, $\gamma', \kappa' = \frac{\gamma+\kappa}{2}$. The behavior of the coupled mode line widths is shown in Figure 1-8.

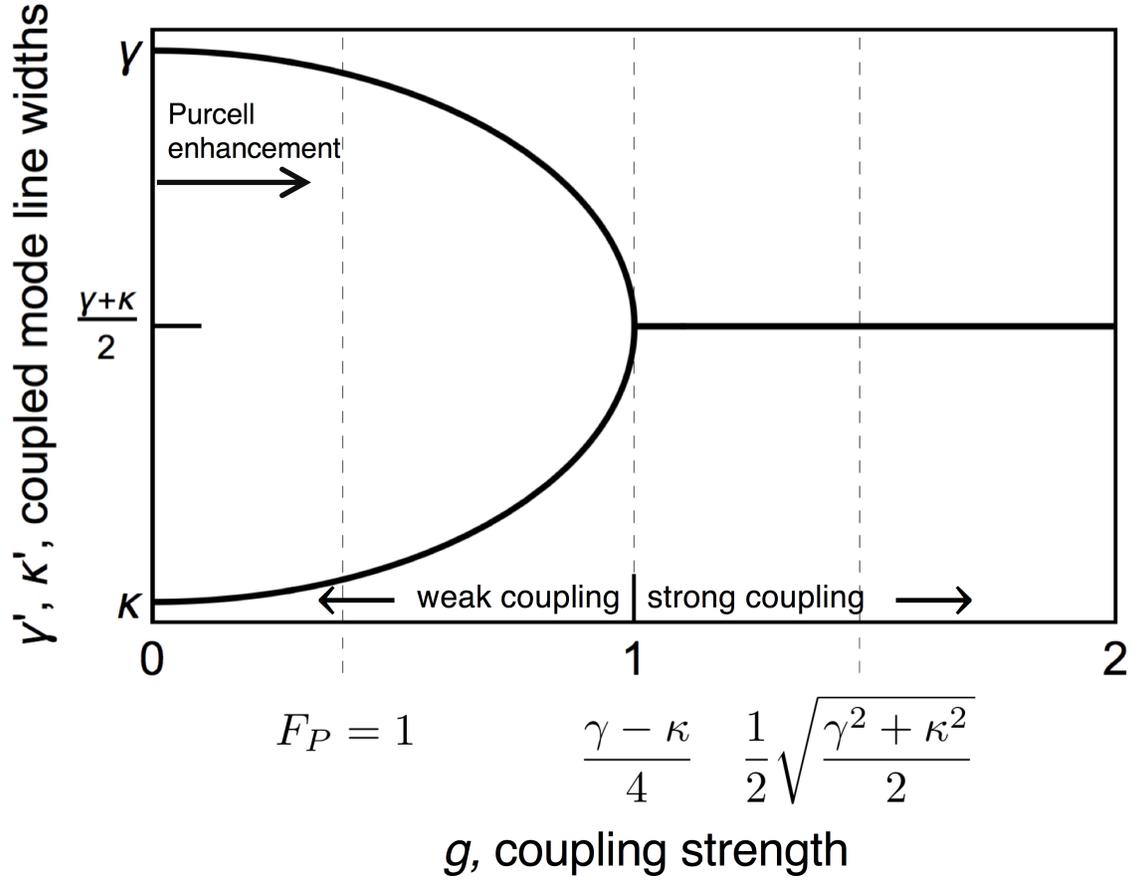


Figure 1-8: Imaginary part of eq 1.7 showing effect of coupling strength on line width of coupled quantum emitter and optical mode. Under strong coupling conditions, the line widths collapse to one value as the modes become completely hybridized. $F_P = 1$ denotes coupling strength corresponding to a Purcell Factor of 1. $g = 1$ shows the weak-to-strong coupling crossover in this scenario. The line near $g = 1.5$ shows where the Rabi splitting is larger than the average line width and thus approximately where strong coupling starts to be observable and long-living.

From Figure 1-8, a clear transition from weak to strong coupling is seen at $g = \frac{\gamma-\kappa}{4}$, the point at which the square root term becomes real. At this critical point, the

emitter and cavity modes hybridize and their line widths equalize.

Let us now plot the real solutions of eq 1.7 (called $\Delta\omega_-, \Delta\omega_+$) in Figure 1-9a. Here, the strong-to-weak coupling transition is clear, as is splitting of the spectral positions of the new hybrid modes. The uncoupled quantum emitter and cavity modes were chosen to have the same resonance energy and under weak coupling, their resonant energy doesn't change; only the line width changes. At the critical point, they hybridize into two new modes split by $\Delta\omega_- + \Delta\omega_+ = \Omega$, where Ω is known as the Rabi splitting. Under these strong coupling conditions, energy oscillates coherently between the two hybrid modes at the Rabi splitting frequency. However, at small Ω , this coherence will be quickly overcome by other loss mechanisms, given by the mode line width. The rule of thumb commonly used to define the true turn-on of strong coupling is when the Rabi splitting value is larger than the average line width of the hybrid modes, $\Omega > \frac{\gamma+\kappa}{2}$. This occurs for $g > \frac{1}{2}\sqrt{\frac{\gamma^2+\kappa^2}{2}}$, as indicated by position (iii) on Figure 1-9a.

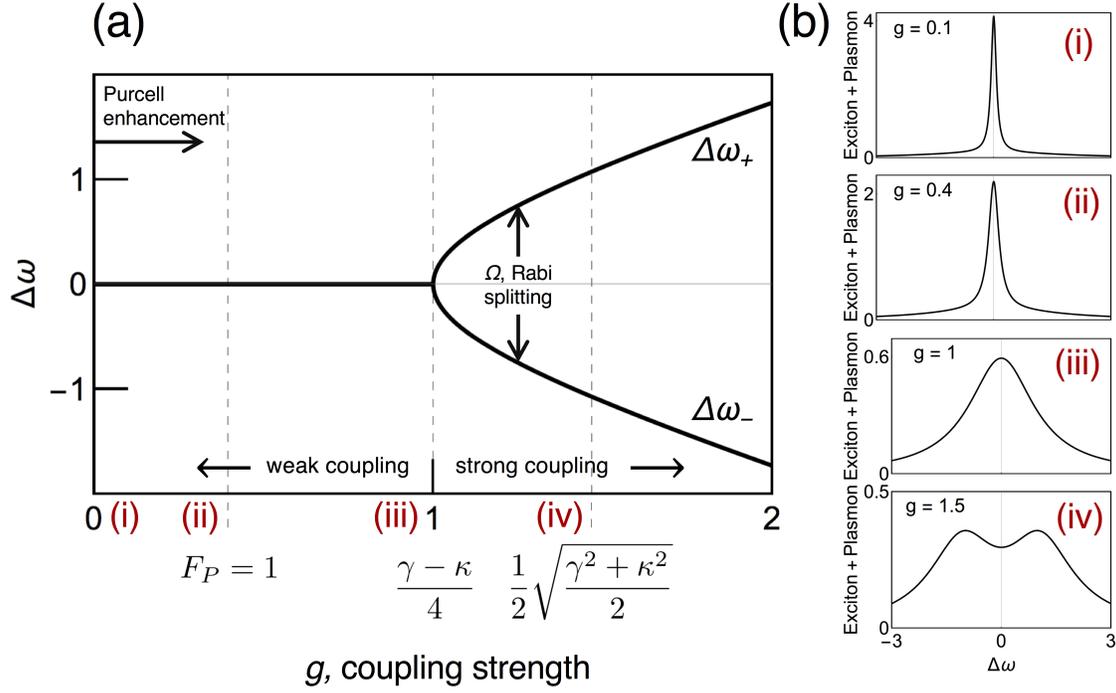


Figure 1-9: Effect of coupling strength on spectrum. a) Real part of eq 1.7 showing effect of coupling strength on mode energy. Under strong coupling conditions, the modes fully hybridize and a gap opens up the size of the Rabi energy. Near the critical point, the observed Rabi splitting may be smaller than the $2g$. At higher coupling strength, $\Omega \rightarrow 2g$. b) Spectral shapes of sum of two Lorentz oscillators with line widths and separation dictated by Figures 1-8 and 1-10, at coupling strengths indicated with red Roman numerals. The Purcell effect modifies the line width of the single mode prior to onset of strong coupling (i, ii). At strong coupling threshold, the hybrid modes are not resolvable (iii). Strong coupling effects compete with other decay mechanisms near the point when the coupling strength equals the average line width (iv), and Rabi splitting is resolvable.

If we assume the two modes have observable spectra shapes (e.g. by measuring transmission or reflection from the system) that can be characterized by Lorentzians, we can characterize the shape of the coupled modes as the sum of two Lorentzians with amplitude: $A_{coupled} = \frac{1}{2\pi} \left[\frac{\gamma'}{(\Delta\omega - \Delta\omega_-)^2 + (\frac{\gamma'}{2})^2} + \frac{\kappa'}{(\Delta\omega - \Delta\omega_-)^2 + (\frac{\kappa'}{2})^2} \right]$. Letting $\gamma = 4.15$ and $\kappa = 0.15$, $A_{coupled}$ is plotted for values of g for four regions (labelled i-iv) in Figure 1-9b. For weak coupling (i), we see a single sharp peak with width primarily from the sharp cavity mode. At the Purcell enhancement threshold (described later) (ii), the behavior is similar with some broadening. At the strong coupling threshold (iii), the shape of the coupled modes is broadened with width equal to the average of the two modes, however, as $\Omega = 0$, the coupled modes overlap. Finally, at the point where $\Omega > \frac{\gamma + \kappa}{2}$ (iv), the peaks become distinguishable. Thus, the condition $\Omega > \frac{\gamma + \kappa}{2}$ is not strictly necessary to resolve the Rabi splitting. The more general rule of thumb to characterize the presence strong coupling is to note whether or not the Rabi splitting is observable; at this frequency of energy exchange between the modes, the interesting quantum coherent effects can start to be observed. Further evidence is provided by anti-crossing behavior of the modes in dispersion plots, as will be seen in Chapters 4 and 5. [16, 18]

The Purcell effect is the modification of the vacuum radiative emission rate when an emitter is placed in a medium with modified photonic density of states such as near another resonator. The radiative decay rate is modified by a factor of $1 + F_P$, where $F_P = 6\pi c^3 Q / (\omega_e^3 V)$ is the Purcell factor, $Q = \omega_c / \kappa$, the cavity Q-factor, ω_e and ω_c are the emitter and cavity resonant frequencies, respectively, and V is the mode volume. F_P can be related to the coupling strength via $g = \mu \sqrt{\omega_c / (2\epsilon_0 \hbar V)}$ and the emitter decay width $\gamma = \omega_e^3 \mu^2 / (3\pi \hbar c^3 \epsilon_0)$. Some algebra yield $F_P = 4g^2 / (\gamma \kappa)$. The Purcell effect is seen in the weak coupling regime, until the modes are completely hybridized and have common line widths. In Figure 1-8, the Purcell effect can be seen as the weakly coupled mode widths move toward each other; the sharper resonator (κ) increases its line width/decay rate and the broader resonator (γ) decreases in width. The coupling strength for $F_P = 1$ is noted in Figures 1-8 and 1-9 for reference. See refs [15, 17, 19] for further details related to the Purcell effect and strong coupling.

In Figures 1-8 and 1-9, we assumed $\gamma = 4.15$ and $\kappa = 0.15$ for demonstration purposes. The ratio $\gamma : \kappa$ determines the weak-strong coupling landscape. In Figure 1-10, γ is set to 1 and κ is swept to show the coupling strengths required for three conditions: 1) the orange curve shows when $F_P = 1$, 2) the green curve shows the weak-strong coupling threshold, and 3) the blue curve shows the coupling strength required for the observable Rabi splitting to equal the average coupled mode line width. In Figure 1-10, we see for $\gamma = \kappa$, the weak-strong coupling threshold goes

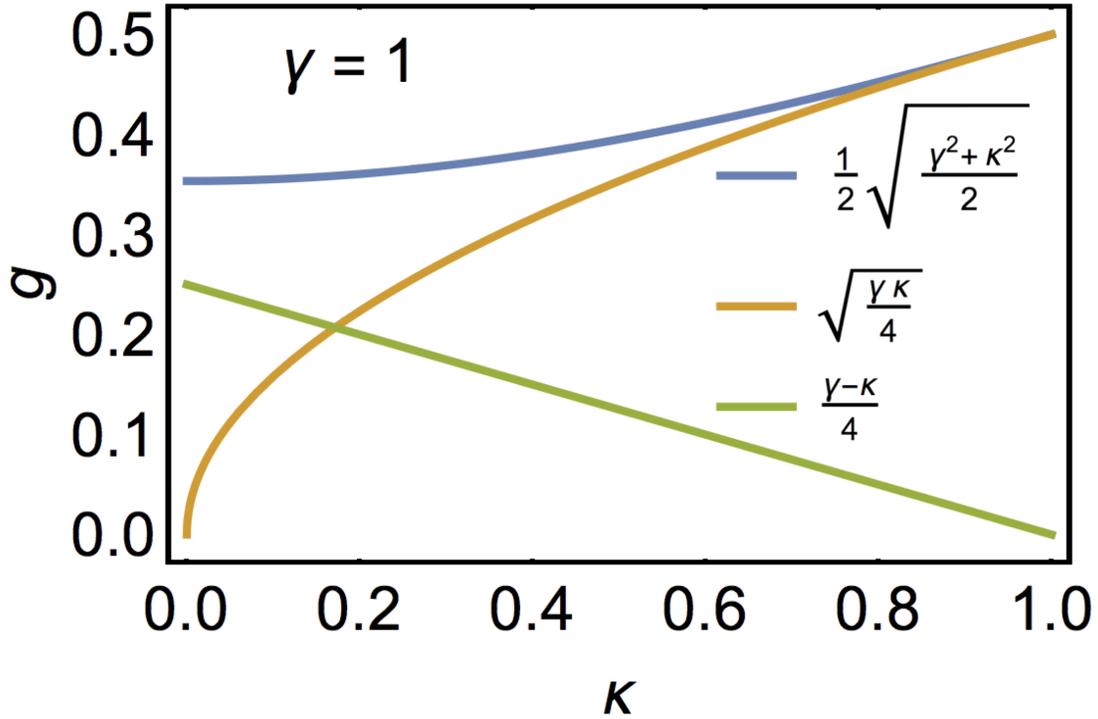


Figure 1-10: Effect of emitter:cavity linewidth ratio ($\kappa : \gamma$) on coupling behavior. The Purcell effect (orange line indicates $F_P = 1$) onset is earliest for small ratios. The coupling strength required to enter the strong coupling regime goes to zero as the ratio goes to 1 (green line) while the condition to resolve strong coupling (green line) gradually increases.

to $g = 0$ while the g needed for Rabi oscillations to compete with inherent decay is maximized. The point at which $F_P = 1$ intersects the weak-strong crossover is for $\kappa : \gamma = 0.17, g = 0.2$, therefore the Purcell effect is only seen for $\kappa < \gamma$ and can only reach large values for $\kappa \ll \gamma$.

1.4 Plasmonic interactions from weak to strong coupling and beyond

1.4.1 Zero coupling

In the weakest limit of no coupling, plasmon excitations have no interactions with non-local media. The plasmon energy eventually decays into phonon modes in the local medium or by photon emission. As an example, consider a plasmon wave traveling along a silver nanowire, much like photons propagating through an optical fiber. Such plasmonic transmission lines may transfer optical energy at subwavelength length scales, where coupling with other media is considered an undesired loss mechanism and is minimized. Applications include optical information processing, where lossy metal wires in a computer chip may be replaced by plasmonic transmission lines with faster transfer rates. Another application is sub-diffraction limited optics, where the Abbe diffraction limit is beaten by squeezing light into dimensions much smaller than the free space wavelength. Conventional optics are limited to resolution of a couple hundred nanometers, or more generally $\sim \frac{\lambda}{2n}$, where λ is the wavelength of light and n is the refractive index of the imaging medium. Resolving smaller features requires more expensive techniques that may not be suitable for some microscopy—e.g. *in vivo* imaging is not possible with a scanning electron microscope. Inexpensive methods of imaging subwavelength features, such as found in biology, could revolutionize optics, especially in medicine. For example, the HIV virion has a diameter of about 100 nm, just below the diffraction limit. Plasmonic superlenses may offer a technique to imagine such size scales and could possibly be integrated with conventional microscopes. In a superlens, a sample is placed on the lens, illuminated, and subwavelength near-field information is collected and transmitted through the lens, where it is magnified until it is no longer diffraction limited. A conventional microscope can then image the transmitted light. One method to create the superlens is to use an array of diverging plasmonic nanowires. On one side of the lens, the wires have subwavelength spacing, while on the other side they have \sim wavelength spacing. The tips of the nanowires

collect optical information from the sample as plasmons which are then transmitted to the other side of the lens. In such a configuration, one of the key limiting factors is coupling of plasmons between wires, or cross-talk. If the wires are brought too close together, plasmons on one wire will couple with plasmon modes in the other wire. Thus, a superlens must operate in the extreme weak limit, where coupling between plasmonic transmission lines is brought as close to zero as possible. The limitations and properties of such a system are described in Chapter 3.

1.4.2 Weak coupling

As we move up in coupling strength to the weak coupling regime, plasmonics can be used to modify absorption properties. For example, plasmonics has been used to enhance light absorption in thin film photovoltaic cells and photodetectors or in photocatalytic systems. To understand the increased absorption, consider a plasmonic nanoparticle (NP) embedded in a thin film semiconductor. On-resonance, the SPPs, called localized surface plasmon resonance (LSPR) when confined to a nanostructure, concentrate light around the NP. This is equivalent to increasing the photonic density of states in the semiconductor, making absorption more efficient. Alternately, one can view the semiconductor's optical properties as unchanged and instead the intensity of light is increased near the NP. As absorption is generally linear with intensity, more light will be absorbed in the region of the NP than far away from it. The net effect is that to absorb the same number of photons, less semiconductor volume is needed when resonant plasmonic NPs are incorporated. [20–29]

Emission is the equivalent process to absorption under time reversal. [30] Therefore, it should be expected that plasmonics can also modify emission properties. If an excited state 'sees' a region of higher optical density of states, the emission rate will be enhanced. This phenomenon is termed the Purcell effect. An excited state undergoes radiative and non-radiative decay processes. By enhancing the radiative decay rate via the Purcell effect, the radiative efficiency relative to the non-radiative is enhanced. The Purcell effect can thus be used to improved the luminescence efficiency of light emitting diodes (LEDs). In practice, there is a trade off between the

enhanced emission properties and increased non-radiative decay into phonon modes into the nearby resonant NP or other structure. [19, 31, 32]

Similarly, SPPs can be used to enhance Raman signal in surface-enhanced Raman scattering (SERS) and tip-enhanced Raman scattering (TERS). In fact, Raman enhancements can be greater than 10^{10} . Raman scattering is inelastic scattering of light whereby low-energy vibronic modes of the atomic bonds in a crystalline material or of molecular bonds are excited, reducing the photon energy of the scattered light (and sometimes increasing it as well). The change in energy is correlated with the vibrational modes of the system and can be used to fingerprint many materials. Typically, the photon energy used to detect a Raman signal is non-resonant with the system, although resonant Raman is also possible. Thus, it is a similar effect to the Purcell effect, where the primary driver is using light to increase the local optical density of states, which influences the emission properties of a local emitter. In the case of SERS/TERS, the increase of optical density is equivalent to an enhancement in electric field strength or intensity. This applies to absorption of the incident light as well as the emitted photons, therefore the enhanced Raman signal is proportional to the square of the enhanced optical intensity. Plasmonic Raman enhancement therefore combines both the absorption and emission enhancements, discussed above, into one process. [31]

It should be noted that the optical density of states can also be decreased through appropriately engineered nanostructures such as photonic crystals. Thus, the Purcell effect can also be used to suppress emission or fluorescence efficiency. [33]

1.4.3 Strong coupling

Consider a plasmon in a metallic NP weakly coupled to an exciton in a semiconductor or molecular system. The light absorbed and emitted by the exciton is mostly plasmonic in nature, not photonic and the degree of absorption and emission enhancement is controlled the coupling strength between the plasmon and exciton. In the weak coupling limit, the absorption and re-emission process only happens one time as there are competitive non-radiative decay mechanisms. In theory, absent

non-radiative decay or radiative decay to free space, the energy could be passed back and forth between the two particles indefinitely. The energy would then be delocalized between the two particles, hybridizing them into a new quasiparticle. As with other hybridized systems, a splitting in energy-momentum space occurs at the intersection of the two modes, creating two new hybrid modes. By maximizing the coupling strength between the two particles and minimizing other decay pathways, such a regime may be achieved, called the strong coupling regime. The typical condition to define the strong coupling regime is when the coupling rate between the particles is faster than the average decay of the particles, $\gamma_{SC} > \frac{\gamma_{SPP} + \gamma_{ex}}{2}$. Here, γ_{SC} is the strong coupling rate, γ_{SPP} is the decay rate of surface plasmons and γ_{ex} is the decay rate of excitons. [31]

How does one maximize the coupling strength between two particles? First, they must be near each other in real space. Next, as with any interaction, energy and momentum must be conserved, therefore, the two particles are required to overlap in k-space. Finally, the better the spectral overlap of their absorptive line shapes, the stronger the coupling; an oscillator that can absorb 100 % of incident photons will wash out an oscillator that absorbs 0.01 %.

Thus, the strongly coupled system is one in which two particles hybridize, coherently transferring energy back and forth between them. Being a coherent interaction, strong coupling has implications for quantum physical phenomena. Much like the overlap of electrons can collapse into Bose-Einstein condensation, strongly coupled polariton-excitons may condense to the ground state as a polariton condensate. Polariton condensation has been shown to emit monochromatic coherent light upon incoherent excitation, with a very low excitation threshold. Quantum coherent effects in strongly coupled systems are studied for their possible applications to room temperature quantum computing, information transport, and cryptography. [15, 31, 34–43]

The strong coupling regime is explored in Chapters 4 and 5 in two-dimensional transition metal dichalcogenide – plasmon systems.

1.4.4 Ultrastrong coupling

The ultrastrong coupling regime is defined by coupling strengths on the order of the transition energy. Typically 10 % is taken to be the minimum ratio, η , of coupling strength to transition energy: $\eta > 0.1g/\omega$. Note that this definition does not follow that of strong coupling. Where strong coupling is defined by comparing the coupling strength to the line widths of the involved resonators, ultrastrong coupling compares coupling strength to the energy scale involved. [44]

In fact, some of the interesting features of ultrastrong coupling may still be observed in the presence of large line widths that would place the system in the weak coupling regime. [45] Some of the unique physics of this regime include virtual photons in a squeezed ground state, correlated photon emission, and entangled photon emission. [44, 45]

1.5 Summary

The overarching theme of this dissertation is the exploration of plasmonic coupling to nanoscopic systems from the weak to the strong coupling domains. In Chapter 3, coupling between SPPs and the surrounding medium is minimized to determine the feasibility and performance limits of subwavelength, non-diffraction limited imaging via nanowaveguides. Using simulations, a single 'pixel' 'nanoscope' is shown to have a resolution limit of $\frac{\lambda}{32}$, outperforming the theoretical diffraction limit by more than an order of magnitude. In Chapter 4, coupling between SPPs and excitons in two-dimensional semiconductors is maximized to enter the strong coupling regime, where exchange of energy between the plasmon and excitons outcompetes other decay mechanisms, so that the SPPs and excitons coherently hybridize into a new quasiparticle with altered optical properties. It is shown that such plasmon hybridization can coherently couple excitons of different energies. Possible applications to quantum information processing is discussed. Chapter 5 extends the work of Chapter 4, increasing the strong coupling by over a factor of two at higher order excitons, far from the band edge, and showing that significant modification of the optical band

structure is possible in a inorganic semiconductor at room temperature.

Chapter 2

Methods

2.1 Fabrication

2.1.1 Physical vapor deposition

Physical vapor deposition (PVD) is a technique to coat surfaces with thin films (1 nm to several μm) of solid materials. There are several types of PVD including sputtering, electron beam evaporation, and thermal evaporation. In each method, the material to be deposited is ejected from the surface of a large piece of the material placed in a crucible (also call a boat) that can withstand high temperatures and into a vacuum chamber, coating everything including the sample placed opposite the crucible.

In Chapters 4 and 5, electron beam (e-beam) evaporation is used. In e-beam evaporation, a current is passed through a tungsten filament, heating it until thermionic emission of electrons occurs. The current of free electrons is moved in a circular arc using a magnetic field, tuned to position the current so the electrons land on the material in the crucible. The kinetic energy of the electrons is imparted to the surface atoms heating them to gas phase and also providing them with enough kinetic energy to eject from the surface. The ejected atoms then condense as a thin film on the sample.

Many materials may be deposited by PVD, including metals, semiconductors, and insulators. In this work, gold (Au) and silver (Ag) are coated onto glass (which

generally is amorphous SiO_2). A common problem when coating gold and silver on glass is poor adhesion of the metal to the glass due to poor wetting. Titanium (Ti), on the other hand, adheres well to glass as well as to gold and silver. Therefore, it is common to use a thin (several nm) layer of Ti between glass and Au or Ag. Approximately 1 nm of Ti was used as an adhesion layer in this work. The thickness was not determined rigorously; rather, the reliable deposition parameters for 10 nm of Ti were used for 10 % of the duration. Adhesion improvement was verified by placing a piece of Scotch tape onto the surfaces of Ag on glass, Au on glass, Ti/Ag on glass, and Ti/Au on glass and peeling it off. Without the Ti, the metal films were removed by the tape. With the Ti, no peeling was observed. Further, the Ti adhesion layer acts to dampen the plasmonic resonance of thin Ag or Au films. Thus, a minimum of Ti was desired. The thickness of Ti was determined to be acceptably thin by measuring the surface plasmon dispersion, as explained elsewhere, and finding that the plasmon reflection minimum was very sharp and reached near zero, as it should without Ti.

The preceding discussion about an adhesion layer also applies for depositions of Ag or Au on silicon, unless the silicon has undergone surface treatment to remove the native oxide layer. Silicon exposed to air or oxygen has a 2-3 nm native oxide layer coating its surface. The oxide is made of silicon dioxide SiO_2 , which is what most glass is made of. Therefore, adhesion of Ag or Au to silicon is very similar to that of glass.

The Ti/Ag and Ti/Au depositions were performed by electron beam physical vapor deposition at the National Renewable Energy Laboratory (NREL) in Golden, Colorado. Approximately 1 nm of Ti was deposited followed by 40 nm of Ag or 50 nm of Au, without breaking vacuum. The thickness was verified using profilometry.

2.1.2 Chemical vapor deposition

Chemical vapor deposition (CVD) is a vacuum deposition technique for growing films of solid materials. Typically, a substrate is placed in a heated vacuum chamber called a tube furnace. The tube furnace consists of a glass tube that can withstand high

temperatures of greater than 1,000 °C. Heating elements around the tube control the temperature of the tube. The substrate is inserted into the tube as are chemical precursors, often in powder form. Gas may also be flowed through the tube. As the precursors are heated, they start to change to the gaseous phase and mix with the flowed gas over the substrate where a chemical reaction may occur on the substrate to initiate growth of the desired material.

In Chapters 4 and 5, molybdenum disulphide (MoS_2) was grown. The few-layer MoS_2 (FL- MoS_2) growth procedures were adopted and modified from previous methods developed by Yu et al. [46] while the growth was performed by Hanyu Zhang of NREL. The CVD growth was carried out by a three-temperature-zone furnace. 500 mg of sulfur pellets (Sigma Aldrich) were placed at Zone 1, while the sapphire wafer (Universitywafer.com) was located at Zone 3. An insert tube in Zone 2 was used to create an isolated local environment. 2 mg of MoO_3 powder (Sigma Aldrich) was loaded into the insert tube. 25 sccm Ar/O_2 (4 vol. % of O_2) premix gas was flowed over the MoO_3 powder into Zone 2. The O_2 allows the MoO_3 precursor to remain in oxidized form as it is carried through the insert. 125 sccm of Ar was supplied at Zone 1 to carry the sulfur and balance the growth pressure at 1 Torr. During the growth, the temperatures in Zones 1, 2, and 3 were ramped at the rate of 35, 35, and 70 °C/min and then held for 30 min at 180, 530, and 930 °C, respectively. This method can be used to produced monolayer to thick films of MoS_2 . In this work, approximately 10 nm films were produced.

2.1.3 Transfer of quasi two-dimensional films from growth substrate to alternate substrate

Mono-layer and few-layer thin films are considered quasi-two-dimensional (2D) materials. Although they have finite thickness, their electronic properties can approach those predicted for theoretical 2D materials, hence the designation of 2D. In this work, 2D few-layer MoS_2 (FL- MoS_2) of 15 nm or less thickness is studied. The material is grown on a sapphire substrate due to sapphire's tolerance to the growth temperatures

used as well as its low surface roughness. Thermal-oxide-coated silicon wafers are also sometimes used to grow MoS₂. By contrast, growth on the plasmonic silver or gold films is not possible due to their low heat tolerance and relatively high surface roughness. Therefore, to study FL-MoS₂ on silver or gold coated substrates, the MoS₂ must be transferred from the growth substrate to the sample substrate. Fortunately, MoS₂ bonds to sapphire via weak Van der Waals bonds and so is easily separated from the growth substrate.

To transfer CVD-grown FL-MoS₂ film onto alternate substrates, the procedures described by Xu et al. were followed. [47] The CVD-grown MoS₂ was coated by polystyrene (PS, average molecular weight 192000, Sigma-Aldrich) dissolved in toluene (50 mg/ml) by spin-coating at 3,000 rpm for 60 seconds. The resulting PS/MoS₂/sapphire film was dried in a 150 °C oven for 5 min. One edge of the film was scribed by a utility knife to create an opening for etchant to access the MoS₂/sapphire interface. The substrate was soaked in 80 °C 2 M NaOH solution until the NaOH etched the interfacial sapphire across the wafer and the PS/MoS₂ film detached from the sapphire wafer. The PS/MoS₂ film was carefully transferred to an ultra-pure water (18.2 MΩ) bath to minimize NaOH residue. Then, the PS/MoS₂ film, which naturally floats on the water surface, was transferred to the alternate substrate by lifting the substrate up from underneath the film. The PS coating was removed by soaking the substrate in toluene.

2.1.4 Substrate cleaning

Special care must be taken when preparing substrates for deposition of nanoscopic films. Particulates and impurities can affect film quality, e.g. surface roughness and continuity of the film, as well as electronic characteristics. In this work, substrates were cleaned by sonicating in acetone then isopropyl alcohol (IPA) for several minutes, soaking in an aqua regia bath (3:1 hydrochloric acid:nitric acid) for an hour, rinsing twice with Milli-Q water, drying with a nitrogen spray gun, and further desiccating on a hot plate at 150 °C for several minutes. The acetone will clean oils and organic compounds. Acetone, however, leaves its own film behind, which is removed by IPA.

Sonication helps dislodge debris. Aqua regia is an acidic solution that will dissolve many remaining impurities including metals. Cleaned substrates were stored in a dry nitrogen box between cleaning and using, although this time was minimized.

2.2 Experimental

2.2.1 Angle-resolved reflectivity in the Kretschmann-Raether configuration and experimental dispersion

Dispersion plots, as explained in Chapter 1 and shown in Chapters 4 and 5, can be experimentally derived by measuring angle-resolved optical reflectivity in the Kretschmann-Raether (K-R) configuration. The K-R method is also discussed in Chapter 1. In the K-R configuration, shown in Figure 2-1, plane-polarized white light incident from the glass side (prism side) of a plasmonic thin film gains the extra momentum needed to couple to surface plasmon polaritons in the film due to increase in momentum by a factor of the index of refraction of the glass. The component of momentum of light tangential to the film, and thus along the direction of propagation of surface plasmons, is proportional to the sine of the incidence angle of light in the glass. Therefore, by changing the angle of incidence, the light may be coupled to plasmons of different momentum (and energy). The momentum-angle relationship is given by $k_{//} = (2\pi/\lambda)n_g \sin \theta_i$, where λ is the incident photon wavelength, n_g is the index of refraction of the prism, and θ_i is the incident angle of light inside the prism. Dips in reflectivity spectra at given angles are indications that coupling to the plasmon mode has occurred for the photon energy at the minima (assuming negligible transmission, which is usually true due to total internal reflection). Thus, by plotting the photon energy of reflection minima for various angles (and therefore momenta), one can construct experimental energy-momentum "dispersion curves." Further, by plotting $A = 1 - R$, where A is absorptivity and R is reflectivity, versus incident photon energy and momentum in a two-dimensional color plot, where the color represents the value of A , one can map out the full momentum space dispersion of the plasmon mode.

The above discussion is generally true—the dispersion of any system located in the evanescent field at the surface in the K-R configuration may be mapped out. The surface plasmon is merely an interesting case of a surface optical mode that can

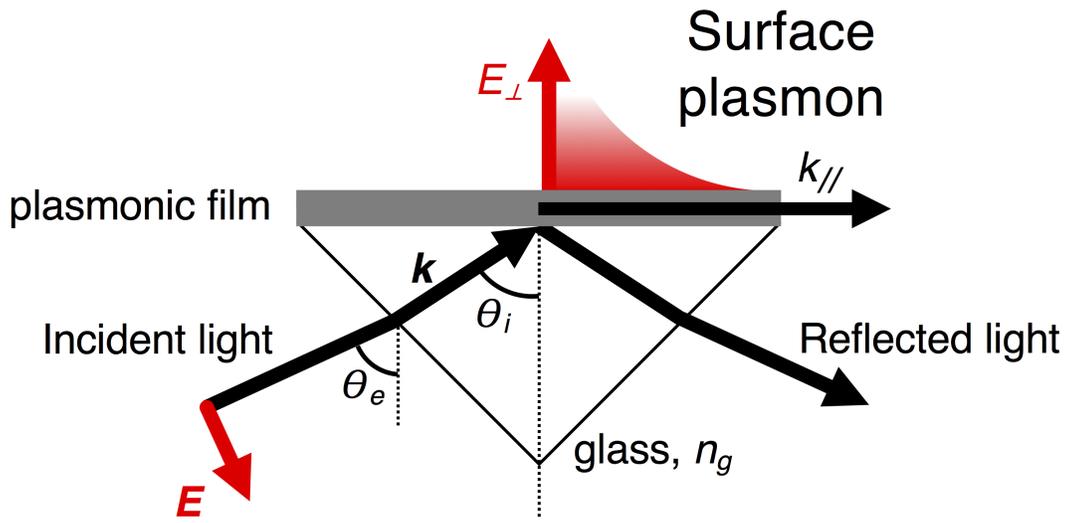


Figure 2-1: Plasmon excitation and angle-resolved reflectivity using the Kretschmann-Raether method: The plasmon is excited via p-polarized white light incident on the plasmonic thin film through the backside of the glass prism. Angle-resolved reflection is collected by varying θ_e and measuring the intensity of the reflected light relative. The excited plasmon decays evanescently away from the surface of the prism.

measured in this way.

The dispersion data shown in this dissertation, Chapters 4 and 5 specifically, were collected using the reflectivity mode of a J. A. Woollam M-2000 ellipsometer at ambient conditions. Using an ellipsometer was found to be an efficient collection method with good angle and wavelength resolution. Unless otherwise noted, a Ag reflectivity reference was created by depositing 1 nm Ti / 200 nm Ag on a control prism of the same type used for the sample data and reference spectra were collected in the same manner as the sample. At each angle, the reflected intensity collected from the sample was normalized to that of the reference to calculate reflectivity. The prism causes the beam to become misaligned at the collection detector and collected intensity to drop as the input angle of the ellipsometer is changed. In theory, such deviations are accounted for by normalizing to the reference prism which experiences the same effect. In practice, as the intensity drops, the signal-to-noise ratio decreases

and thus the quality of the data decreases. Thus, the system was realigned every few degrees, such that alignment error was maintained to $< 1\%$ in the region of interest. The data were collected every 0.1° , unless otherwise noted.

2.2.2 Photoluminescence

In photoluminescence (PL), a material absorbs a photon by exciting an electron into a higher energy state, which decays back to the ground state by emitting a photon of the same or lower energy. In a semiconductor, the excited electron is generally promoted from the valence band to the conduction band, creating a bound (exciton) or unbound electron-hole pair. If excited at energies above the bandgap, the carriers decay to the local band minima and then may decay (i.e. recombine) by photon emission, also called photoluminescence. The emitted photon energy is equal to the bandgap energy of the semiconductor.

In indirect bandgap semiconductors, the efficiency of photoluminescence is generally lower than in direct bandgap semiconductors. This is because emission of a photon must be accompanied by emission of a phonon to account for the momentum mismatch between photons (~ 0 compared to electrons or holes) and the conduction band minimum to valence band maximum transition. The extra step of phonon emission gives competing decay mechanisms, such as decay to trap states, more weight.

Transition metal dichalogenides (TMDCs), such as MoS_2 , are indirect bandgap semiconductors in the bulk, with very inefficient photoluminescence. However, as their thickness decreases, approaching one monolayer, they transition to direct bandgap semiconductors. Thus, mono- to few-layer TMDCs exhibit increasing photoluminescence with decreasing thickness. Furthermore, with decreasing thickness, their bandgaps increase and exciton positions in absorption spectra blue shift. Such behavior is useful in characterizing the thickness of the TMDCs. Any appreciable photoluminescence signal is a sign that the number of layers is no more than ~ 10 - 20 .

When PL is characterized in Chapter 4, it is acquired with an InVia Renishaw Raman/PL confocal microscope with a 532 nm laser. The sample is excited and the PL (or Raman) spectra collected with a long working distance $50\times$ objective lens

(Olympus SLMPLN) with a numerical aperture of 0.35. The 532-nm laser spot size is about 1 μm . A Thorlabs PM100D Power Meter measured the laser power. For the PL measurements, the laser intensities were between 6.5 μW and 12.9 mW with 2 s integration time and 5 accumulations. The PL signal was dispersed on a 600 lines/mm grating and detected by a charge-coupled device (CCD) array.

2.2.3 Raman microscopy

Raman microscopy employs Raman scattering to characterize materials and molecules by detecting infrared resonances. In Raman scattering, light incident onto a material interacts with that material via inelastic scattering, where some photon energy is transferred to vibrational modes of the atoms or molecules of the material. Such an effect can be detected by shining a high power single wavelength laser onto a material and analyzing the spectrum of reflected light. A large peak will be seen at the incident photon energy due to the low probability of undergoing Raman scattering. Smaller peaks will be observed at lower energies as well, with the difference in energy between the incident photons and any other peaks giving the energies of the excited modes. This is termed the Stokes shift. There is also the possibility that energetic vibrations existent in the material prior to laser excitation will lose energy, scattering with the incident photon energy to emit higher energy photons. This is termed the Anti-Stokes shift.

The two-dimensional TMDCs exhibit characteristic Raman modes, one of which is independent of thickness and one that shifts with the transition from bulk to monolayer. Therefore, characterizing the separation between these two modes can be used to determine the thickness of mono- and few-layer films.

Raman data in this dissertation in Chapter 4 were taken in the same way and system as PL with two differences: 1) the laser intensity was 1.84 kW/cm² and 2) a 1800 lines/mm grating was used to disperse the collected light.

2.2.4 Atomic force microscopy

Atomic force microscopy (AFM) is a form of scanning probe microscopy (SPM), used (among other reasons) for characterizing topography over nanometer to micrometer domains and angstrom to micrometer heights. AFM topographic imaging works by scanning a sharp tip (radius of curvature of 10s of nanometers) across a sample of interest and measuring the vertical deflection of that tip as a function of horizontal position. The tip is usually located at the end of a cantilever and its deflection is measured by reflecting a laser from the portion of the cantilever above the tip to a photosensitive arrayed detector, such as a CCD. When the tip deflects, the laser spot on the CDD moves. As the tip is lowered to the surface, the tip will deflect as it interacts with the surface through various forces, e.g. Van der Waals or Coulombic forces. There are a few different methods to control the position of the cantilever. In this work, the so-called tapping mode is used. In tapping mode, a piezoelectric driver oscillates the cantilever at or near its primary resonance frequency (in the vertical direction) and an electronic servo controls the average height of the cantilever. The frequency and amplitude of the oscillation can be controlled by the piezo. As the tip is scanned across the sample surface and the height of the surface varies, the amplitude and frequency of oscillation will change because the forces between tip and sample change. The values of amplitude and frequency are fed into a dynamic control system that adjusts the piezo to maintain constant amplitude and frequency (within tolerance). Thus, the average distance between tip and sample are maintained at a constant value, and the changes in vertical position of the cantilever, with respect to the detector, are measured to determine topography.

In Chapter 4, a Park AFM system in tapping mode was used to characterize the thickness of 2D FL-MoS₂ films.

2.2.5 Profilometry

Profilometry serves the same purpose as AFM for larger dimensions, both vertically and horizontally. Scan lengths of millimeters and heights of several nanometers to

tens or hundreds of micrometers are measurable. The probe tip, or stylus, of a profilometer typically is much larger than AFM, with radius of curvature on the order of a micrometers, although nanoscopic tips are also possible. A feedback loop is used to maintain a constant force between the stylus and sample surface, moving the stylus vertically as needed and recording the movements as topography.

Profilometry was performed for Chapters 4 and 5 using a Veeco DekTak 8 to characterize the thickness of deposited thin metal films. To measure thin film thickness, Kapton tape was used to mask one region of the substrate being deposited onto. The tape was then removed and the step height measured at various locations and averaged. Kapton tape has been found to perform well in masking for profilometry: it is thin which minimizes shadowing deposition onto the nearby substrate; it can tolerate temperatures > 250 °C; and typically it can be removed with no residue left behind. By contrast, thick metal shadow masks tend to shadow deposition of the uncovered nearby substrate, leading to a variation in thickness near the step edge and making profilometry analysis more difficult.

2.2.6 Optical microscopy

Optical microscopy captures light reflected from or transmitted through a sample and maps the signal onto an image plane to be viewed by eye or camera. In Chapters 4 and 5, optical microscopy was performed using the built-in microscopes of other techniques—AFM, profilometry, PL, Raman—to locate regions of interest for their respective measurements.

2.2.7 Ellispometry

Ellipsometry is a technique used to characterize thin films, as discussed in Section 2.2.1. The thickness and complex refractive index of thin films may be modeled from ellipsometric data, even for multi-stack films with optical anisotropies. The basic measurement reflects polarized light from a sample near the Brewster's angle and measures changes in the polarization of the reflected light. Often, this is done at

several incident angles and spectroscopically.

The ellispometry data presented here in Chapters 4 and 5 were taken with a J.A. Woollam M-2000 variable angle spectroscopic ellipsometer.

2.3 Theory

2.3.1 Electromagnetic finite element modeling

The interaction of light with matter can be described by solving Maxwell's equations with appropriate boundary conditions. However, numerical methods must be used for arbitrary and low symmetry geometries. A modeling scheme known as the finite element method, applied to electromagnetic waves, breaks a given structure into an array of smaller discrete regions and solves Maxwell's equations in each region, propagating the solutions of one region into the initial conditions of the next region. The result is that the fields and energy flow and loss of a given structure in the presence of electromagnetic stimulation can be simulated and visualized on a computer.

Chapter 3 shows results using CST Microwave Studio to simulate surface plasmons in nanowires. The complex refractive index for Ag was taken from Johnson and Christy. [6]

2.3.2 Transfer matrix modeling of optical properties of stacks of thin films

The transfer matrix method (TMM) is a mathematical formalism to solve for the reflection and transmission coefficients for stacks of thin films embedded in semi-infinite media. The method works for an arbitrary number and thickness of thin films, any input angle, and any polarization.

The Fresnel equations are the solutions to Maxwell's equations at the interface between two semi-infinite media and describe the behavior of light incident on that interface. Snell's Law, the Brewster's angle, and total internal reflection are all mathematical solutions to the Fresnel equations. The TMM is a generalization of the Fresnel equations to multiple interfaces into the language of linear algebra, where each additional layer is represented by an additional matrix and calculation of the reflection and transmission coefficients is reduced to trivial matrix multiplication. The only inputs are complex refractive indices and thicknesses for each layer.

The TMM is used in Chapters 4 and 5 to model the dispersion of strongly coupled exciton-plasmon systems. Prior to fabrication of actual devices, the thicknesses and materials used were optimized for maximum strong coupling. After fabrication, the actual devices were modeled again and fit to the experimental data to determine thicknesses of the actual materials and verify that the underlying theory explains the data. Further, the TMM was used to model the uncoupled plasmons in these systems, for which there is no experimental equivalent, but is nonetheless a rational and necessary input into the semiclassical harmonic oscillator model used later. See Chapter 4 for further details.

To create the dispersion plots, the TMM is solved for at all angles and wavelengths of interest to obtain angle-resolved reflectivity, R , transmissivity, T , and absorptivity, $A = 1 - R - T$. The A spectra are then plotted together as a two-dimensional color plot (or surface plot) with photon energy (or wavelength) on the y-axis, angle of incidence on the x-axis, and A represented by a color scale. Finally, true dispersion plots should be expressed in energy-momentum space, so a transformation is performed from angle to momentum (or wave vector). The in-plane momentum/wavevector is calculated as $k_{//} = (2\pi/\lambda)n_g \sin \theta_i$, where λ is the incident photon wavelength, n_g is the index of refraction of the input substrate, and θ_i is the incident angle of light inside the input substrate.

The TMM and subsequent data processing were implemented in MATLAB, starting with some code in ref [48]. Please see ref [13] as well the appendices of this dissertation for the source code. Material properties, i.e. complex refractive indices, were either taken from the literature or measured with spectroscopic ellipsometry.

2.3.3 Semiclassical oscillator model of strong coupling

Electromagnetic field-based models such as finite element numerical simulations or the Fresnel equation-based transfer matrix method can be used to predict properties of light-matter interactions, such as reflection and transmission, field strength, and energy flow or loss. Further information can be extracted from quantum models. The semiclassical coupled oscillator model treats interactions between quantized systems,

mediated by classical electromagnetic fields. A full quantum treatment quantizes the fields in addition to the allowable transitions in matter.

In this work, the semiclassical oscillator model is used to characterize the coupling strength and model the coupled mode dispersion in strongly-coupled exciton-polariton systems. The Hamiltonian matrix of the system is written down; terms along the diagonal represent the uncoupled mode energies and off-diagonal terms are the coupling strengths between the uncoupled modes. Solving for the eigenvalues of the Hamiltonian gives the coupled modes in the basis of the uncoupled modes, i.e. predicts the photonic band structure or dispersion of the system. The eigenvectors are a linear superposition of the uncoupled modes and the weights of each uncoupled mode gives the character of each hybrid mode. For example, in a two-level system at the point of maximum coupling in momentum space, the hybrid mode should be equally comprised of both coupled modes, while away from this point the character becomes dominated by a single uncoupled mode. See Chapters 4 and 5 for examples of this technique put into practice.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Nanoscope based on nanowaveguides

3.1 Abstract

The far field spatial resolution of conventional optical lenses is of the order of the wavelength of light, due to loss in the far field of evanescent, near electromagnetic field components. We showed via simulations that subwavelength details can be restored in the far field with an array of divergent nanowaveguides, which map the discretized, subwavelength image of an object into a magnified image observable with a conventional optical microscope. We demonstrated in simulations that metallic nanowires, nanocoaxes, and nanogrooves can be used as such nanowaveguides. Thus, an optical microscope capable of subwavelength resolution — a nanoscope — can be produced, with possible applications in a variety of fields where nanoscale optical imaging is of value.

3.2 Background

This chapter is based on my published work. [49] Further details of the methods used can be found in Chapter 2.

With conventional lenses, image details much smaller than the wavelength of the light being employed in microscopy propagate only as evanescent waves and thus are lost at distances beyond a wavelength, or the far field [50, 51]. However, there is no fundamental limitation on the recovery of this lost information, and a far field super-lens has been proposed based on a negative refractive index medium [52]. The lack

of negative refractive media operational in the optical frequency range has thus far prevented experimental demonstration of such a far field superlens. Another way to achieve a far field superlens is to place a near field optical magnifier (NFOM) within the near field of an image, configured so that the output of the magnifier is no longer subwavelength and can thus be propagated into the far field. Metamaterial NFOMs based on alternating thin metallic and dielectric films have been proposed [53,54] and experimentally demonstrated [55,56]. NFOMs based on arrays [57], including divergent [58,59], of continuous nanowires with inter-wire wave propagation and segmented wires capable of multi-wavelength operation in the optical regime [60] have also been proposed and simulated but not yet experimentally demonstrated. Another NFOM, based on divergent arrays of coaxial nanowaveguides, with exclusive intra-waveguide wave propagation (either by TEM photon modes or surface plasmon polaritons) has been proposed by some of us [61]. An NFOM based on generalized nanowaveguides is shown schematically in Figure 3-1 as a part of a nanoscale microscope, or nanoscope. The nanowaveguides can have their input ends located on a planar (as depicted) or curved stage. In this depiction, the distal ends of these diverging waveguides are located on a much larger (also planar or curved) image surface. The light emitted or scattered from the illuminated object (“object”) couples to the input ends of the waveguides whose lateral spacing is smaller (even much smaller) than the illumination wavelength λ and propagates as waveguided light or surface plasmon polaritons (SPP), which subsequently re-emit as photons at the output ends of the waveguides, where the spacing is larger than λ . The ratio of the inter-waveguide spacing at the opposite sides of the NFOM determines its magnification. Conventional optics can then be used to capture the magnified output image, making the entire device a nanoscale optical microscope, or nanoscope. While in the original proposal [61] the waveguides were envisioned as nanoscale coaxial transmission lines (nanocoax), here we further explore nanowires and nanogrooves as alternatives. We also investigate a

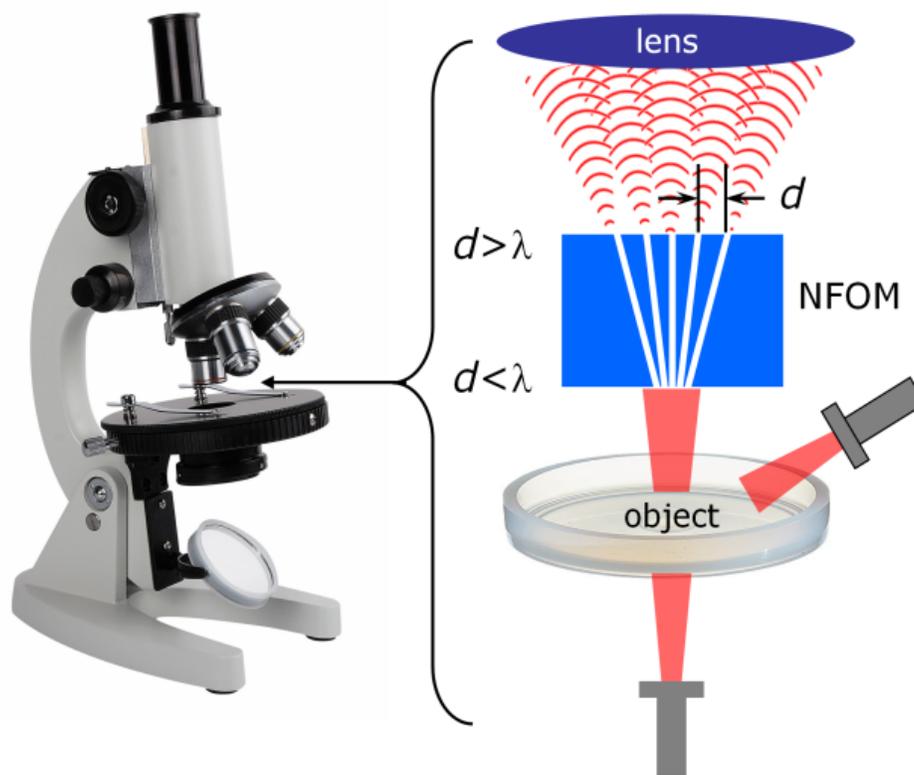


Figure 3-1: Schematic of imaging with proposed near field optical magnifier comprised of nanowaveguides (white lines) supported by dielectric filler (blue). The subwavelength-size “object” is placed at the input ends of the nanowaveguides array (having waveguide separation $d < \lambda$). When illuminated, the image of the object is propagated along the waveguides and re-emitted as a magnified image at the output ends of the waveguides (with separation $d > \lambda$). Traditional optics are used to capture the magnified image.

diverging pair of nanowires as a basic building block of our NFOM, illustrating its subwavelength, nanoscopic resolution.

The basic factors considered when selecting an appropriate nanowaveguide for our NFOM were: i) light coupling to the waveguide, ii) attenuation of light or SPPs along the waveguide, iii) inter-waveguide crosstalk, and iv) manufacturability. Depending on the sensitivity of the illuminated object to radiation intensity, there may be an additional constraint placed on the allowed strength of the illumination source, therefore setting a lower limit on the minimum allowable transmission of the device. It has been demonstrated that nanocoaxes can couple to and transmit visible light with relatively long propagation length, estimated to be up to 50 microns [62, 63]. These waveguides are expected to have negligible crosstalk, since the propagating light is enclosed entirely within a metallic shield. Similarly, it has been demonstrated that nanowires are capable of transmitting light [64–66]. Ditlbacher, et al. [64] showed experimentally that nanowires can be used as waveguides of light energy via SPP modes, with propagation length of at least 20 microns and tight field confinement to the wire, shown in Figure 3-2. Recently, it has been demonstrated that grooves in metal can also act as nanowaveguides [67, 68]. As fabrication of metallic nanowires is less complicated than concentric metal-dielectric-metal nanocoaxes, and surface grooves easier still, we investigated the performance of all these structures. Silver was our material of choice for its relatively low optical loss in the visible. [6]

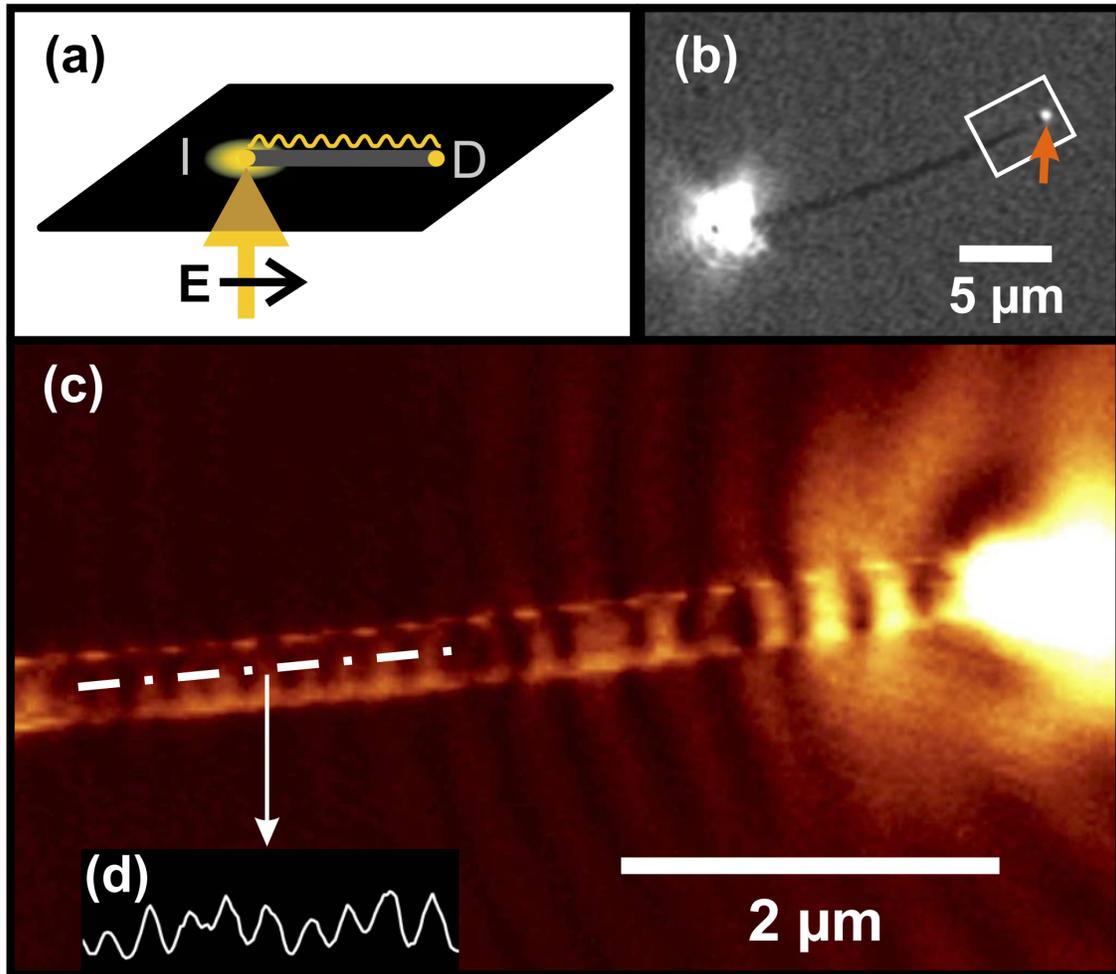


Figure 3-2: SPP propagation along $18.6 \mu\text{m}$ long silver nanowire. (a) Sketch of optical excitation; I is input and D is distal end of the wire. (b) Microscopic image—the bright spot to the left is the focused exciting light. The arrow indicates light scattered from the distal wire end. (c) SNOM image—the image area corresponds to the white box in (b). (d) $2 \mu\text{m}$ long cross-cut along the chain dotted line in (c). Optical excitation is with 785 nm laser. Reprinted figure and caption with permission from Ditlbacher, H., et al. *Phys. Rev. Lett.* 95, 257403 (2005). Copyright 2005 by the American Physical Society.

3.3 Results and discussion

To compare nanowaveguide structures, we simulated transmission of light through the aforementioned waveguide architectures by employing the finite element method [69]. See Subsection 2.3.1 of the Methods Chapter 2 for further details. To excite a waveguide in our NFOM, we use a circular pinhole aperture in a metallic wall (simulated as a sheet of a perfect electrical conductor). A monochromatic wave propagating on one side of the wall produces a dipole-like excitation of the waveguide, placed on the other side of the wall, 10 nm away from the aperture. Figure 3-3 shows electric field maps for a silver nanowire 40 nm in diameter and 5 μm long taken at wavelengths spanning the visible range, from $\lambda = 500$ to 800 nm. Negligible mode attenuation is seen for the $\lambda = 800$ nm wavelength, while appreciable reduction occurs for $\lambda = 500$ nm. Further simulations have shown that minimal attenuation occurs for λ between 600 and 800 nm. Similar behavior is found for nanowires from 20 nm to 200 nm diameter, generally increasing in transmission with increasing diameter.

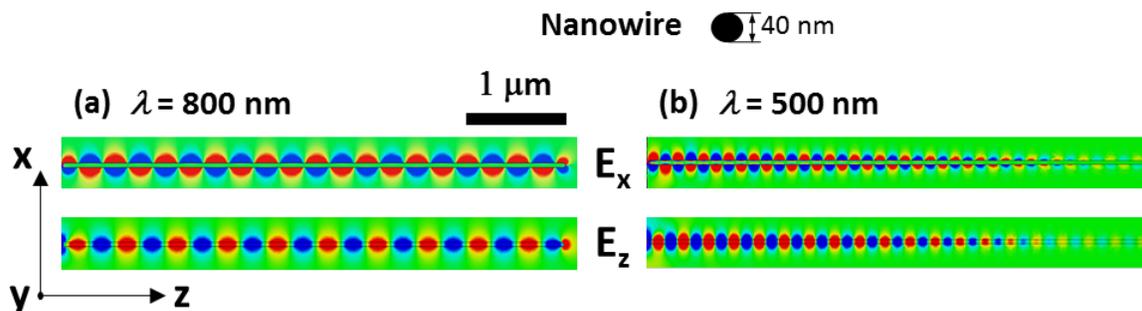


Figure 3-3: X and z components of the electric field along silver nanowires 40 nm in diameter and 5 μm in length, excited with light of vacuum wavelength (a) $\lambda = 800$ nm and (b) 500 nm. SPP modes are clearly visible in each case, but with case (b) showing high attenuation over the length of the wire.

From Figure 3-3, it is clear that the nanowire mode is SPP with a strongly plasmonic character. This is evidenced by a large, along-the-axis E_z component of the

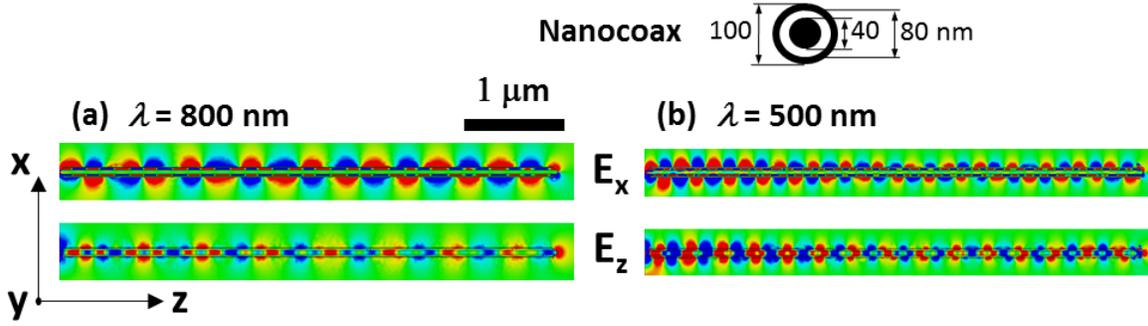


Figure 3-4: X and z components of the electric field along nanocoaxes $5 \mu\text{m}$ in length, excited with light of vacuum wavelength (a) $\lambda = 800 \text{ nm}$ and (b) 500 nm . In both cases, the coaxes have a 40 nm diameter inner silver core, a 20 nm thick vacuum annulus, and a 10 nm thick silver shield. The nanocoax modes appear to be a combination of classical coaxial cable TEM modes and SPP modes. Case (a) shows little attenuation while case (b) shows clear attenuation, albeit less than for the nanowire in Figure 3-3b.

electric field, with large penetration into the nanowire bulk. This causes appreciable metallic loss and, consequently, the large attenuation at $\lambda = 500 \text{ nm}$. The corresponding result for the nanocoax structure is shown in 3-4, where relatively less attenuation is seen at 500 nm , due to a smaller plasmonic character of the SPP in this architecture.

The classic analog of this nanoscale waveguide, the conventional coaxial cable, is a non-plasmonic TEM transmission line, and much of that character is retained in the nanocoax, although our coaxes lack perfect enclosure of the light within the coax due to the thinness of the metallic shield. We have also tested two open, “nanogroove” waveguide scenarios, shown in Figure 3-5. Here, we find excellent light confinement and negligible attenuation, though only at the 500 nm wavelength shown. The electric field of the transmitted mode is almost entirely the E_y component, i.e. it is strongly non-plasmonic, which reduces attenuation due to metallic losses. Above 600 nm , however, attenuation due to radiative loss is large, and the wave is no longer confined to the guide.

In general, we observe excellent wave propagation (in either a plasmonic or pho-

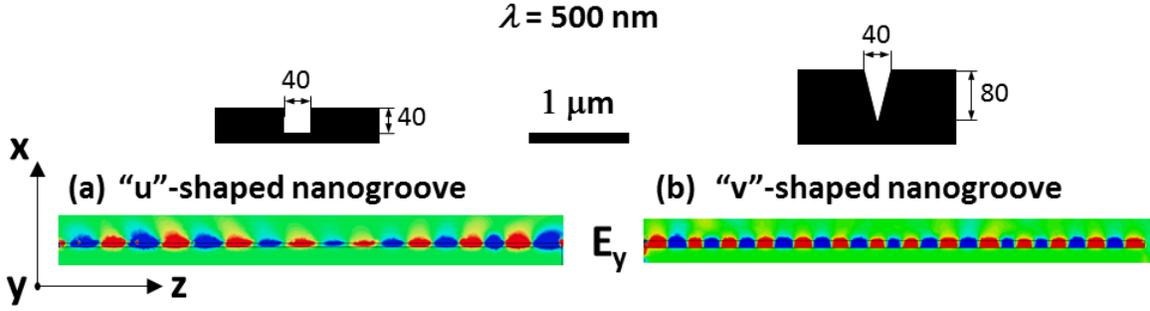


Figure 3-5: X and z components of the electric field along nanocoaxes $5 \mu\text{m}$ in length, excited with light of vacuum wavelength (a) $\lambda = 800 \text{ nm}$ and (b) 500 nm . In both cases, the coaxes have a 40 nm diameter inner silver core, a 20 nm thick vacuum annulus, and a 10 nm thick silver shield. The nanocoax modes appear to be a combination of classical coaxial cable TEM modes and SPP modes. Case (a) shows little attenuation while case (b) shows clear attenuation, albeit less than for the nanowire in Figure 3-3b.

tonic regime) along all waveguides studied, at least in certain frequency bands. Furthermore, we studied the effect of perturbations to the nanowire’s geometry and found that 10 nm deviations in a wire of 40 nm diameter changed the transmission of the wire by no more than 10% . However, there is significant impedance mismatch at the input and the output ends, leading to significant loss of transmission efficiency due to reflections. This mismatch can be strongly reduced (in principle, eliminated) by proper engineering of waveguide terminations (antennae). Techniques might include dielectric grading, diameter modulation, or other antenna schemes, and will be employed in future studies and designs.

In the following study, we chose nanowires as the waveguides of an NFOM. Optical waveguiding in nanowires has recently been demonstrated experimentally [64] and via simulations [66]; in ref. [64], $20 \mu\text{m}$ long nanowires, laser-excited at one end, emitted light from the distal end clearly visible with a conventional optical microscope. Figure 3-6 shows the geometry of the pair simulated here, as well as field maps for the E_x and E_z components. We chose the nanowires to be 40 nm in diameter and 2

μm in length, with no dielectric coating. The wires were excited with the pinhole described above, with light of vacuum wavelength $\lambda = 800 \text{ nm}$, but with the location of the pinhole aperture varied along the x -axis. The input ends of the wires were placed $\lambda/10 = 80 \text{ nm}$ apart (center-to-center), while the distal ends of the wires were separated by $1 \mu\text{m}$ (center-to-center). Figures 3-6a and 3-6c are for the case of the excitation pinhole directly in front of the upper nanowire, and Figures 3-6b and 3-6d for the pinhole halfway between the nanowires. Clearly, the excitation location dramatically determines the excitation degree of a given nanowire. This determines the resolution of the system, in this case at least 80 nm , i.e. strongly subwavelength. The transmitted power density of a particular wire is at a maximum when the aperture is near that wire and quickly drops to zero away from the wire. We conclude that inter-wire crosstalk in our proposed plasmonic nanowire NFOM is sufficiently low that subwavelength image integrity is maintained at the distal end of the device. The crosstalk issue has been studied previously [66] in a pair of dielectric-coated nanowires placed certain distances apart. The results showed that crosstalk rapidly decayed with increasing inter-wire distance (as trivially expected), and can be further reduced by coating with a dielectric. It is worth noting that the studies we present here were also performed for silver-based nanocoaxes, yielding similar results to the nanowire case.

The subwavelength resolution of our simulated NFOM is further confirmed in Figure 3-7, which plots transmitted field intensity (defined as the normalized magnitude of the time-averaged Poynting vector (extracted from simulation), averaged over the distal end of the wire) against the location of the pinhole light source. In Figure 3-7, it can be seen that each wire only propagates light when the aperture source is localized near its end (e.g. centered at $x \sim 45 \text{ nm}$ for the upper wire, blue data), with negligible light collection/plasmon propagation when the source is move to the adjacent wire (see blue data near $x \sim -45 \text{ nm}$). The FWHM is about 25 nm for the

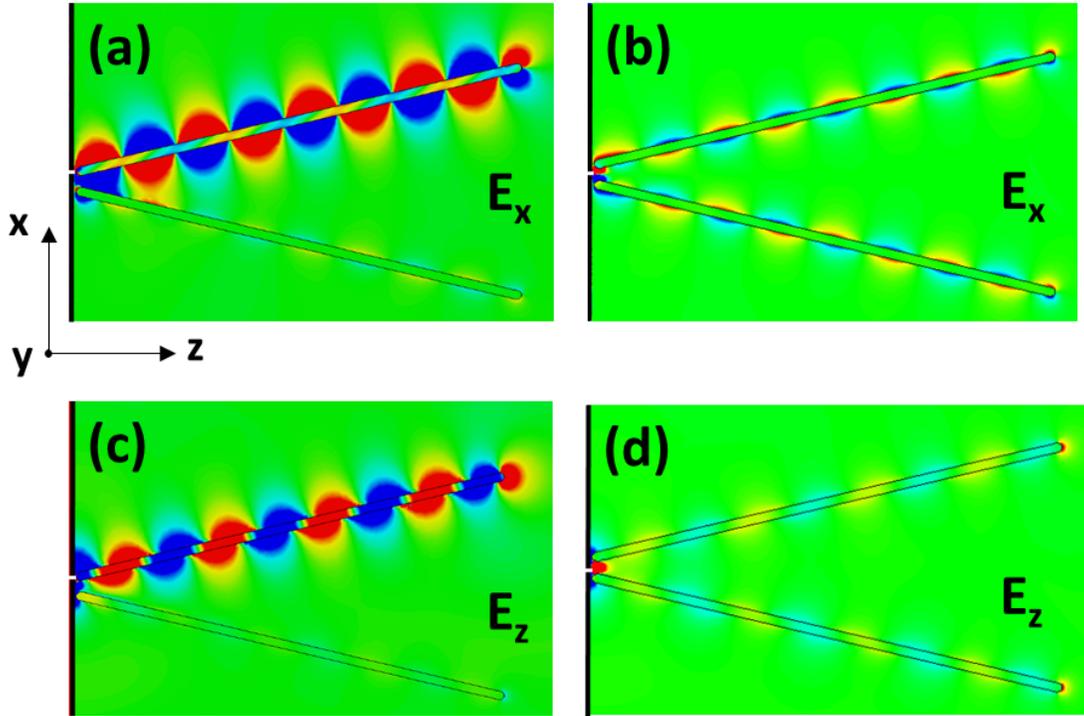


Figure 3-6: X and z components of the electric field in a two nanowire NFOM. The wires are 40 nm in diameter, 2 μm in length, and made of silver. Their close ends are separated by 80 nm, center-to-center, while their distal ends are separated by 1 μm , center-to-center. They are excited by $\lambda = 800$ nm vacuum wavelength light, with the aperture positioned adjacent to the center of the top wire ((a) and (c)) or halfway between the two wires ((b) and (d)). Excitation of the top wire results in SPP modes highly localized to that wire alone, while excitation between the wires yields symmetric, low amplitude SPP modes on both wires.

40 nm diameter wires employed. This value can be adopted as the spatial resolution for the nanoscope structure. Considering potential issues fabricating precisely controlled-diameter nanowires, we simulated an NFOM with one wire 40 nm and the other 60 nm in diameter and found the device achieved $\sim 90\%$ of the resolution of the symmetric 40 nm device.

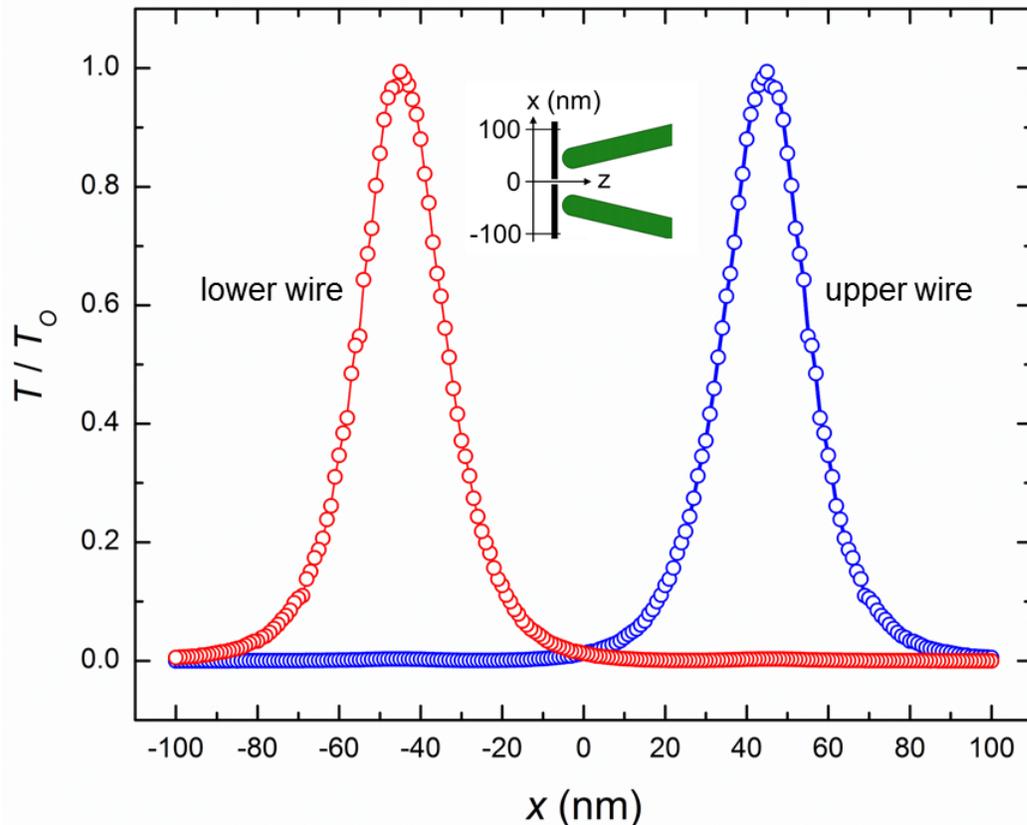


Figure 3-7: Calculated normalized transmitted electric field intensity along the converging wires of Figure 3-6, as the excitation aperture is moved along the x -axis, from halfway between both wires ($x=0$) to above and below both wires ($x=\pm 100$ nm). This intensity is found by averaging the normalized magnitude of the time-averaged Poynting vector in a cylinder 60 nm in radius surrounding the distal half of the wires. The inset shows the geometry.

3.4 Conclusions

In conclusion, we have shown that a near field optical magnifier based on plasmonic nanowires can in principle be used to image subwavelength features. Our simulated device has resolution of $\sim \lambda/32 = 25$ nm. Such a device could be used in conjunction with conventional optical microscopy for real-time noninvasive imaging of living systems at the nanoscale. We have also shown that the real driver of efficiency for such a device is not propagation decay along the wires — or even precise control of the wires' geometry — but impedance matching into and out of the device.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Plasmon-mediated coherent superposition of discrete excitons under strong exciton–plasmon coupling in few-layer MoS₂ at room temperature

4.1 Abstract

Here, we demonstrate room temperature coherent hybridization of the valley-polarized A- and B-excitons in few-layer MoS₂ (FL-MoS₂), mediated by simultaneous strong coupling to surface plasmon polaritons. FL-MoS₂ was placed on a tunable plasmonic structure and the dispersion of the system was measured by tuning the plasmon energy across the exciton energies. Strong coupling was found as double Rabi splitting at the A- and B-excitons of 81 meV and 93 meV, respectively. A coupled harmonic oscillator model sheds light on the nature of the interaction and reveals a coherent quantum superposition of the A- and B-excitons, mediated by the plasmon interaction. This observation suggests the possibility of room temperature intra- or intervalley quantum information transport and/or spin entanglement. The experiment confirms a previous theoretical prediction of room temperature exciton–exciton hybridization

in two-dimensional MoS₂. Further, through modeling we find that room temperature strong coupling of valley-polarized excitons is a general phenomenon among 2D TMDC exciton–plasmon systems.

4.2 Background

This chapter is based on my work submitted for publication. [70] Further details of the methods used can be found in Chapter 2.

Two-dimensional transition metal dichalcogenides (2D TMDCs) are under intense study for their optical, electronic, and catalytic properties [71, 72]. They exhibit semiconducting behavior with bandgaps in the visible to near-IR, high mobilities, extraordinarily high absorption coefficients, valley-polarized carriers, and catalytic activity [71, 72]. In addition, their sharp excitonic transitions make them interesting candidates for strong room temperature exciton-polariton coupling while their atomic thickness is well-suited for location within the plasmon polariton optical near-field.

Exciton-polariton strong coupling has been studied in the field of cavity quantum electrodynamics for over three decades [73]. The field has produced many exciting observations such as thresholdless lasing [36, 74], room temperature condensation [74], and other quantum coherent phenomena [75] with potential applications in quantum information [15, 76, 77] and control of chemical reactions [78, 79].

Strong coupling between an electronic transition and an optical mode of equal energy occurs when the coupling strength between the two is similar to or greater than their average decay [15, 16]. Distinct from the weak coupling regime, where the Purcell effect can modify the spontaneous emission rate, the energy band splitting that occurs under strong coupling conditions implies fundamentally new optoelectronic and chemical properties [79]. Strongly coupled quantum dot cavity systems exhibit anti-crossing, or Rabi splitting, of the exciton-cavity modes, i.e. show suppressed emission on resonance and hybridization of the two modes into a split band, whereas under weak coupling a single Purcell-enhanced emission is seen [32]. The band recon-

figuration under strong coupling has also been used to modify decay pathways and rates of excited carriers in light-sensitive organic molecules [78–80]. See Section 1.3 for further details on the weak and strong coupling regimes.

In 1974 Agranovich and Malshukov [34] predicted and in 1978 Pockrand and Swalen [81] demonstrated that the cavity polaritons can be replaced by surface plasmon polaritons to create exciton-plasmon strong coupling, opening up a new platform for strong coupling investigations. A wide array of plasmonic thin films and nanostructures is available to experimentalists [82] and allows for the excitonic system to be incorporated in a convenient open planar geometry. To date, most exciton-plasmon strong coupling has been demonstrated with organic dye molecules or quantum dots [16, 82] due to their sharply defined exciton transitions—which survive at room temperature—and the ability to create thin films in the plasmon near field. Conventional inorganic semiconductors have exciton binding energies of ~ 10 meV in the bulk [83] and a few times higher in two-dimensions [84], near the thermal energy at room temperature of 26 meV, and thus are generally cooled to observe strong coupling. More recently, 2D TMDCs have emerged as a new class of inorganic semiconductors with large binding energies of ~ 100 s of meV at mono- to few-layer thickness [71], allowing strong coupling to be observed at room temperature [85]. Further, their large absorption coefficients allow nanometer films to absorb tens of percent of incident light [71], well within the plasmon penetration depth. Strong coupling of the 2D TMDCs has been studied since 2015, with observations in cavity and plasmonic systems at room temperature and cooled [85–88].

In this letter, we demonstrate room temperature exciton-plasmon polariton strong coupling of the A- and B-excitons and exciton-exciton hybridization in few-layer MoS₂ (FL-MoS₂) on thin Ag films. To the best of our knowledge, observation of neither room temperature exciton-polariton strong coupling of the B-exciton in MoS₂, nor room temperature exciton-exciton hybridization in any 2D TMDC has been previ-

ously reported. Evidence for the strong coupling is found as double Rabi splitting at the A- and B-excitons in the experimental dispersion relation of 65 and 72 meV, respectively. A coupled harmonic oscillator model sheds light on the nature of the interaction and reveals a coherent quantum superposition of the A- and B-excitons, mediated by the plasmon interaction. We also show that room temperature strong coupling appears to be a general phenomenon in 2D TMDC exciton-plasmon systems by calculating dispersion plots for many systems in the Kretschmann-Raether configuration. Further, this is the first work to apply the Kretschmann-Raether configuration to strong plasmon coupling in the 2D TMDCs. The results obtained are important to the understanding of the properties of MoS₂ and other 2D TMDCs and could have implications in the design of polaritonic and valleytronic devices.

4.3 Fabrication and material characterization

To fabricate the sample shown schematically in Figure 4-1, Ag (40 nm) with a thin Ti adhesion layer (~ 1 nm) was deposited on the hypotenuse of a right-angle glass prism (Edmund Optics, stock number 32-334, uncoated N-BK7 glass, 20 mm leg length) via electron beam deposition. FL-MoS₂ (~ 5 nm) grown by chemical vapor deposition was transferred three consecutive times on top of the Ag for a total thickness of ~ 15 nm. Data collection with the device of Figure 4-1 will be described in the next section.

The transferred MoS₂ films were studied by atomic force microscope (AFM). Analysis of multiple locations of multiple AFM scans gives an average thickness for each FL-MoS₂ transfer of 5.0 ± 2.1 nm, which corresponds to 7 layers assuming 0.71 nm per layer. [89] Therefore, the total thickness of the studied film, comprised of three transfers of FL-MoS₂ is ~ 15 nm. The left panel of Figure 4-2 shows the optical microscope image of the region probed by AFM shown in the panel on the right; many such regions were probed to obtain MoS₂ thickness. In the optical image, the light-colored

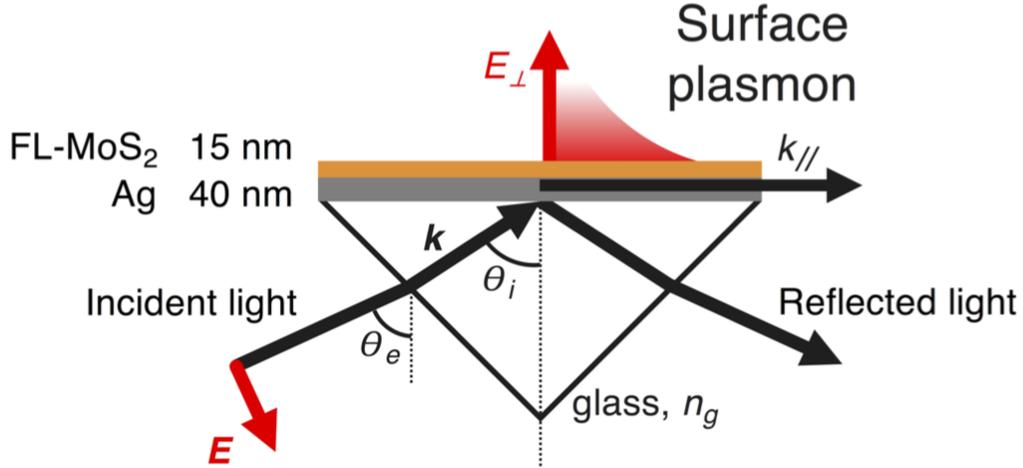


Figure 4-1: Plasmon excitation and angle-resolved reflectance using the Kretschmann-Raether method: The plasmon is excited via p-polarized white light incident on the FL-MoS₂-Ag thin film through the backside of the glass prism. Angle-resolved reflectance is collected by varying θ_e and measuring the intensity of the reflected light relative to a thick (~ 200 nm) silver reference.

stripe to the left of the AFM tip corresponds to the first MoS₂ transfer while the darker region to the right of the tip is a stack of two transfers.

To characterize the films further, photoluminescence (PL) spectra were collected of two regions of the MoS₂ film, as shown in Figure 4-3. The PL spectra show strong peaks corresponding to the A-exciton and corroborate the classification of the CVD-grown MoS₂ as few-layer in thickness — PL intensity goes to zero for films greater than 10 monolayers of MoS₂. [71] Raman spectra show $E_{2g}^1 - A_{1g}$ peak separation and positions characteristic of few-layer MoS₂. [71]

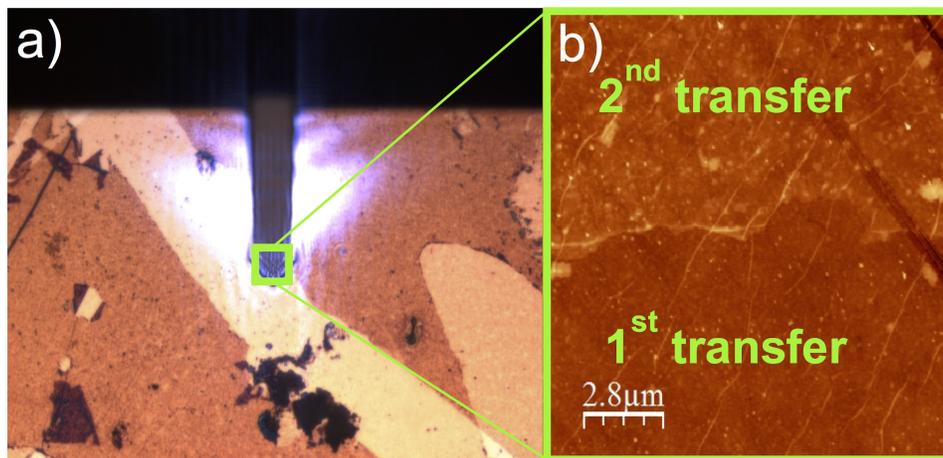


Figure 4-2: (a) Optical microscope view of AFM tip over region scanned in panel b. (b) AFM topography of MoS₂ on Ag. The two regions refer to one transfer and two transfers of FL-MoS₂.

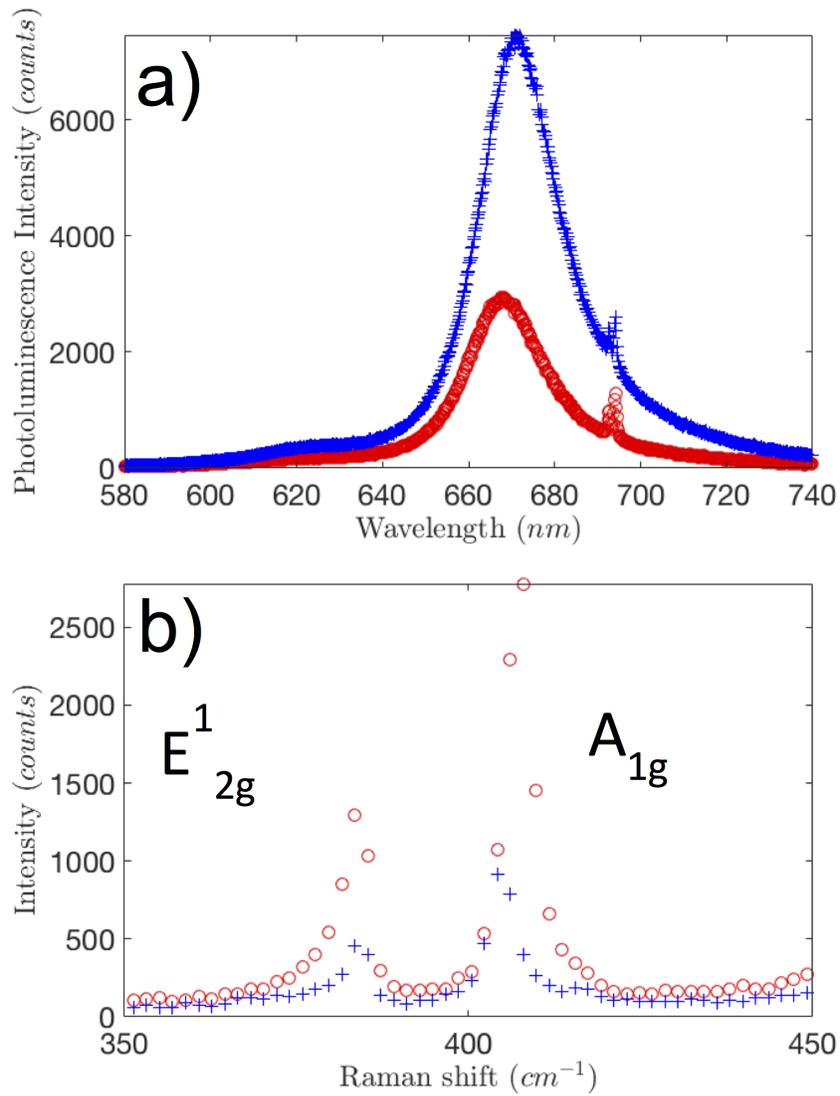


Figure 4-3: (a) PL signal at two regions of FL-MoS₂ on sapphire growth wafer. The sharp signal provides evidence of few-layer thickness. (b) Raman signal of two regions.

The refractive index of the studied three-transfers of FL-MoS₂ on Ag on glass prism was modeled from ellipsometry data. Ellipsometry data was collected from the top side of the prism, i.e. the air side of the MoS₂ film. During the Ag deposition onto the prism, a fused silica witness was also deposited onto and used as a Ag control. The data was modeled in J.A. Woollam WVASE software. First, the Ag control was modeled as: fused silica substrate / 1 nm Ti / 41.5 nm Ag / 0.25 nm Ag₂S. The fused silica refractive index was taken from the WVASE library, Ti from Palik [90] and Ag₂S from Bennett. [91] Starting parameters for Ag were taken from Palik. [92] The refractive index and thickness of Ag were set as fitting parameters. The complex refractive index of Ag was found to compare well to Palik [92] and Johnson and Christy. [6] This model was then extended to model the MoS₂ on the Ag-coated prism. The fused silica was replaced with NBK7 glass, with refractive index specified by manufacturer Schott. Starting parameters for MoS₂ were taken from Jung [93] and were fit along with thickness, while other model parameters were held constant. The thickness was determined to be 14 nm and the optical constants are in-line with other literature values. The resultant fit of the real and imaginary parts of the 3-transfer FL-MoS₂ refractive index are shown in Figure 4-4. The thickness fit of 14 nm matches the AFM, PL, and Raman data well. Note the variation in the real part of the refractive index at the exciton energies is directly evident, as required by the Kramers-Kronig relationship between the real and imaginary parts of the refractive index.

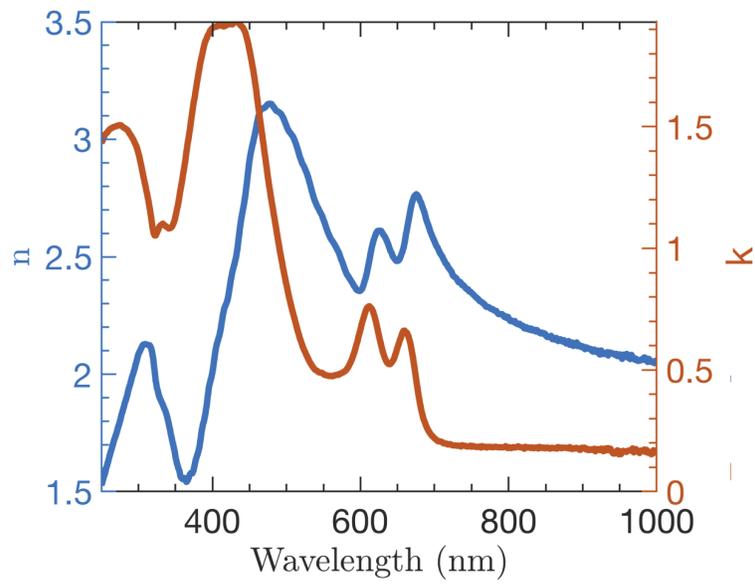


Figure 4-4: Complex refractive index of FL-MoS₂ derived from ellipsometry.

Transfer matrix simulations [13] were used to find target values for the thickness of the Ag and MoS₂ layers [94]. For example, the simulations showed that similar plasmon and exciton peak sizes are needed to optimize strong coupling, hence the large plasmon peak intensity was reduced by using thinner Ag (40 nm) than optimal for the sharpest resonance (~50 nm). Further, the simulations revealed that with increasing MoS₂ thickness, the plasmon energy redshifts until it approaches the A-exciton energy and avoids coupling altogether. Thus, a balance must be struck in finding the optimal thickness of the excitonic layer — too thin and the coupling will be weak, too thick and the surface plasmon energy will not overlap with the exciton energy. Three transfers of few-layer MoS₂ were found to provide a good balance in this regard. Few-layer MoS₂ was chosen over monolayer to minimize the number of samples and transfers needed to achieve the desired thickness. Few-layer and mono-layer TMDCs have very similar complex refractive indices qualitatively, with blue shifting of the exciton energies and increasing magnitudes as thickness is decreased. [95] Thus, although the refractive index for MoS₂ used for the model was for its mono-layer form and underestimated the optimal thickness of FL-MoS₂, the insight that thicker films were necessary for strong coupling in the given configuration was helpful.

4.4 Data collection, analysis, and discussion

To study FL-MoS₂-Ag strong coupling, we employed the simple Kretschmann-Raether configuration, shown in Figure 4-1, used in the pioneering exciton-plasmon coupling experiments [81, 96, 97]. The Kretschmann-Raether method [98] for exciting surface plasmons in a planar thin film offers a simple method to incorporate a tunable plasmon resonance and observe strong coupling. Detuning of the coupling (i.e. probing the interaction at energies away from that of maximum coupling) is achieved by sim-

ply varying the incident angle of light through the prism, allowing dispersion plots to be obtained in a straightforward manner.

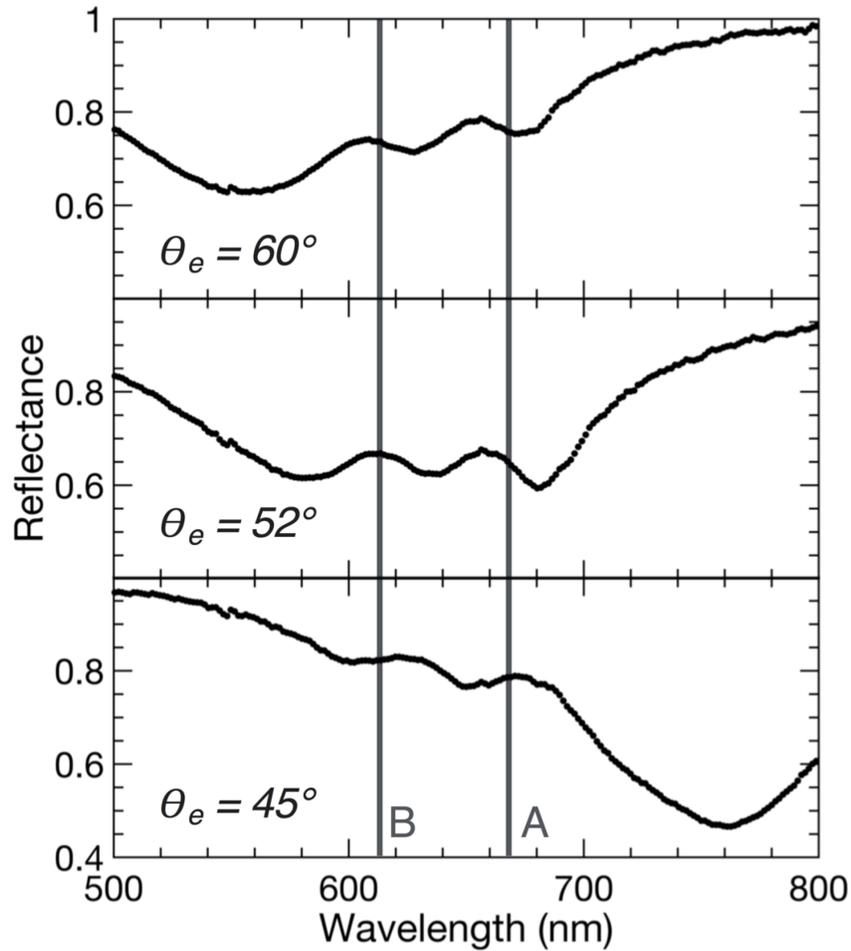


Figure 4-5: Select reflectance spectra: Reflectance is measured via the Kretschmann-Raether method shown in Figure 4-1. The vertical gray lines mark positions of the uncoupled A- and B-excitons. At $\theta_e = 45^\circ$ the plasmon is resonant at 760 nm and the exciton positions are slightly blue-shifted. In the vicinity of the exciton energies ($\theta_e = 52^\circ$), the plasmon completely hybridizes with the excitons. At $\theta_e = 60^\circ$ the plasmon is resonant at 560 nm and the exciton positions are slightly red-shifted.

Evidence for strong coupling can be found as anti-crossing, or Rabi splitting [15], of the coupled modes in energy-momentum dispersion plots. Dispersion plots were constructed by collecting angle-resolved reflectance spectra in the Kretschmann-Raether configuration (Figure 4-1) using an ellipsometer. The ellipsometer input beam was realigned every 2.5° over a collection range of 40° – 80° , which yielded alignment error $< 1\%$ in the region of interest. The data was collected every 0.1° . Figure 4-5 shows the reflectance at a few incident angles, $\theta_e = 45^\circ$, 52° , and 60° . The uncoupled exciton energies, E_A and E_B , are taken from dips in reflectance from the top side of the prism, where coupling to the plasmon is not allowed and the exciton positions are angle-independent.

The plasmon energy blue shifts with increasing angle, dominating the spectrum at the largest and smallest angles and completely hybridizing with the excitons at the intermediate angle. Color dispersion plots were constructed by plotting absorptivity, A , as a function of incident photon energy, E , and in-plane wavevector, $k_{//}$. Absorptivity was calculated as $A = 1 - R$, where R is reflectance. Transmission was assumed zero as incident light in the prism undergoes total internal reflection. Although the reflectance data is collected as a function of angle, the in-plane wavevector is the appropriate basis to use when studying strong coupling; using energy-angle dispersion plots results in an over-estimation of the Rabi splitting energy [16, 99]. The in-plane wavevector is calculated as $k_{//} = (2\pi/\lambda)n_g \sin \theta_i$, where λ is the incident photon wavelength, n_g is the index of refraction of the prism, and θ_i is the incident angle of light inside the prism. All measurements were performed at room temperature in air. Further details of the measurement method are provided in the supporting information.

The experimentally determined dispersion of the few-layer MoS₂-Ag system is presented in Figure 4-6a.

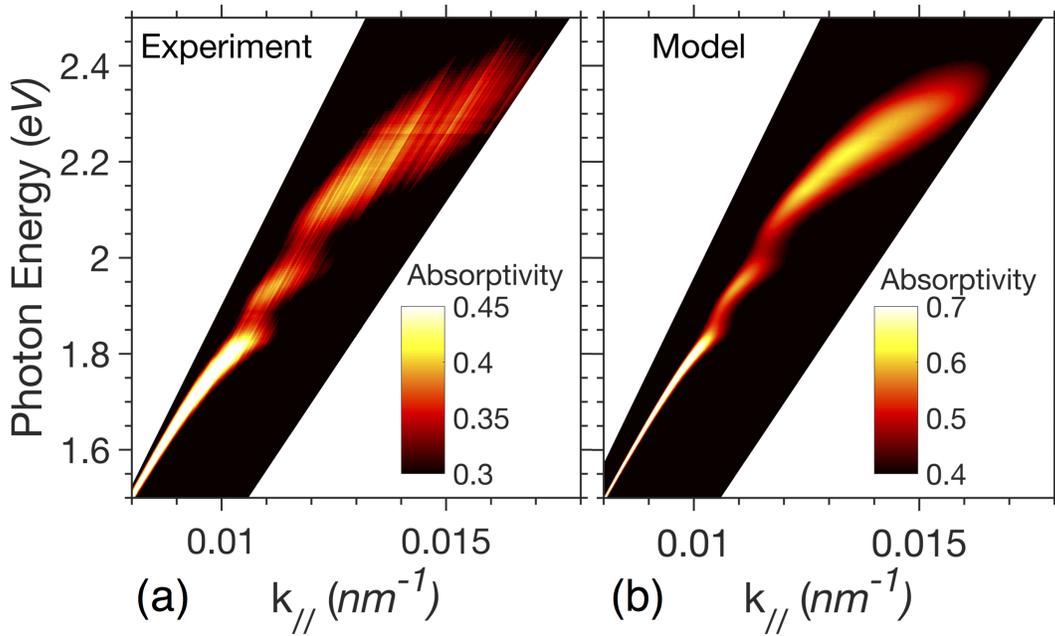


Figure 4-6: Dispersion of FL-MoS₂-Ag: (a) Experimental energy versus transverse wavevector dispersion obtained from angle-resolved reflectance. Rabi splitting of 81 meV and 93 meV is evident at the A- and B-exciton energies of 1.86 eV and 2.01 eV, respectively. (b) Modeled dispersion of the sample measured in panel a, calculated using the transfer matrix method. See main text for calculation of $k_{//}$ and other quantities.

Avoided crossing and thus Rabi splitting is evident at both the uncoupled A- and B-exciton energies of 1.86 eV and 2.01 eV, respectively, indicating strong coupling between the surface plasmon and each exciton. The uncoupled exciton energies were determined from dips in the reflection spectrum taken from the air side of the prism, where coupling to the surface plasmon is disallowed. Among the 2D TMDC exciton-plasmon strong coupling literature, there has been limited work with MoS₂ at room temperature [88,100]. Liu et al. showed exciton-plasmon strong coupling in monolayer MoS₂ for both the A- and B-excitons at 77 K and for the A-exciton only at room temperature using a Ag nanodisk array [88]. Vasilevskiy et al. predicted strong room temperature coupling of the A- and B-excitons in both MoS₂ and WS₂ monolayers in cavities [101]. To the best of our knowledge, room temperature exciton-plasmon strong coupling of the B-exciton in MoS₂ has not been previously reported.

In Figure 4-6b, a transfer matrix model [13] (TMM) shows good agreement with the data presented in Figure 4-6a. The model consisted of the following layers, with p-polarized light incident from the NBK7 glass (prism) side: NBK7 substrate / 1 nm Ti / 41.5 nm Ag / 0.25 nm Ag₂S / 16 nm MoS₂ and used refractive indices from refs. [90] (Ti) and [91] (Ag₂S) and those derived from ellipsometry (Ag and MoS₂). The thickness and complex refractive index of the Ag and MoS₂ thin films were measured with ellipsometry from the air side of a Ag control and the actual sample, respectively. The best fit of the TMM to the data was found by adjusting the thickness of MoS₂ to 16 nm. The two thicknesses found by optical methods, 14 nm from fitting ellipsometry data and 16 nm from fitting a transfer matrix model, average to 15 nm, which is the expected value from AFM analysis. Implementation of the transfer matrix method was done in MATLAB. In Figure 4-6a,b, the lack of data (white space) is a consequence of converting the dispersion plots from energy-angle to energy-momentum space, where $k_{//} \sim \lambda^{-1} \sim E$, where E , is the incident photon energy. The anticrossings do not go quite to zero because of an absorption background as

well as overlap of the plasmon and exciton spectral shapes.

The strong coupling results in a dispersion with three branches with each branch representing a new hybrid exciton-plasmon mode, retaining some character of each of its constituent parts. It is expected that the new quasiparticles, which are discovered by their new optical properties, should also possess altered electrical and chemical behavior. Study of such properties is the subject of ongoing study. To gain further insight into the character of the three optical modes, we fit the data to a semi-classical coupled oscillator model and extracted the exciton and plasmon fraction of each mode. The Hamiltonian representing the system in the coupled oscillator model is shown in eq 4.1:

$$\begin{pmatrix} E_{SPP}(k_{//}) & \frac{\Omega_A}{2} & \frac{\Omega_B}{2} \\ \frac{\Omega_A}{2} & E_A & 0 \\ \frac{\Omega_B}{2} & 0 & E_B \end{pmatrix} \quad (4.1)$$

The semiclassical coupled oscillator model treats the problem as a three-level quantized system of classical fields where a full quantum model would use quantized fields. Both models give the same result for the Rabi splitting [16]. Loss is neglected and coupling between the A- and B-excitons is assumed zero (off diagonal zeros). The diagonal elements Ω_A and Ω_B are the Rabi splitting energies at the A- and B-excitons, equal to twice the coupling strength, $\Omega = 2g$. E_A and E_B are the uncoupled exciton energies.

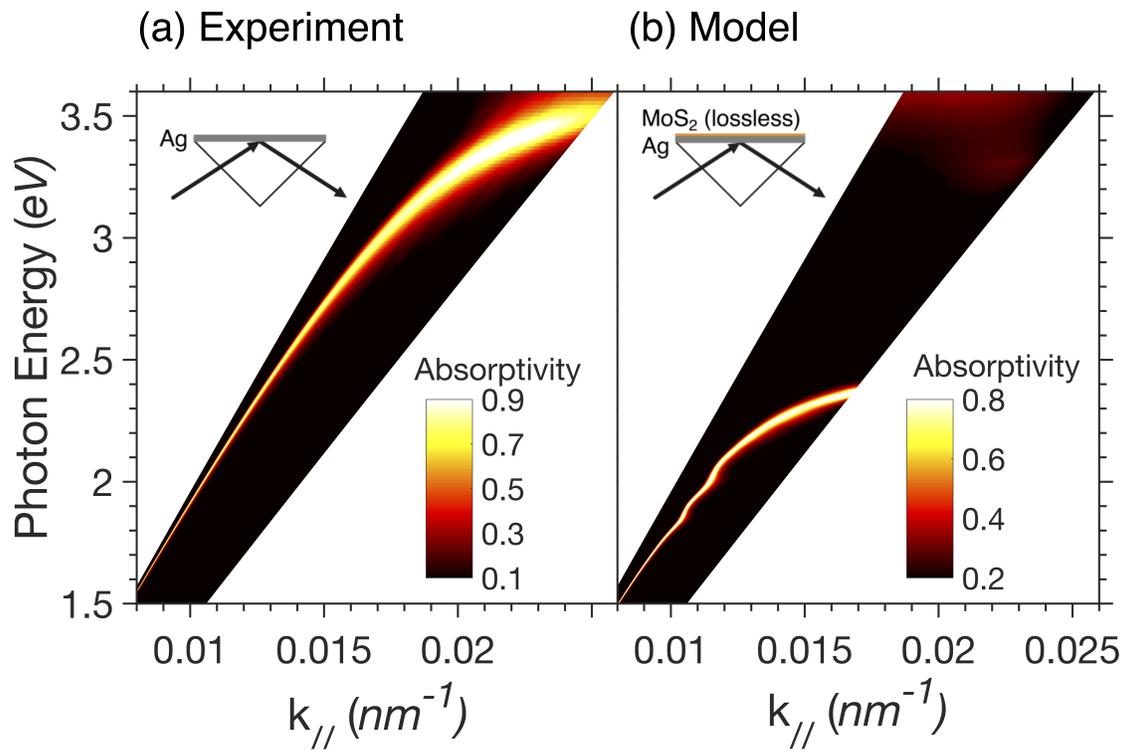


Figure 4-7: (a) Experimental dispersion of Ag control. The plasmon asymptotes at a higher energy without the MoS₂ than with it. (b) Transfer matrix model dispersion of MoS₂-Ag with the lossy part of the MoS₂ refractive index set to zero. This method approximates the behavior of the plasmon absent coupling.

For the fit, Ω_A , Ω_B , E_A , and E_B are set as fit parameters. As such, the only input into eq 4.1 is $E_{SPP}(k_{//})$, the uncoupled plasmon energy. However, the plasmon dispersion from a control sample with no MoS₂, shown in Figure 4-7a, does not accurately represent the uncoupled plasmon energy of a sample with MoS₂ because the media on both sides of the metal thin film influence the behavior of the surface plasmon. Particularly, the real part of the MoS₂ refractive index controls the behavior of the plasmon in the same way the addition of a lossless dielectric thin film would. The imaginary part is responsible for the exchange of energy and thus controls the coupling. Bare Ag exhibits the dispersion shown in Figure 4-7a, with the plasmon asymptotically approaching ~ 3.5 eV. Note that a transfer matrix model reproduces this shape well. However, the addition of “lossless” MoS₂ shifts the plasmon down in energy, approaching 2.4 eV, which agrees well with the upper branch in Figure 4-6a. This is shown in Figure 4-7b, which is the same model shown in Figure 4-6b, but with the imaginary part, k , of the complex refractive index (Figure 4-4) set to 0 for all wavelengths. This simulated bare plasmon dispersion is fitted to a single Lorentzian at each k -point and the peak value represents $E_{SPP}(k_{//})$ in the Hamiltonian and is plotted in Figure 4-8 as the red curve. Note that the features seen in Figure 4-7b at the A- and B-exciton positions are due to the features seen in the real part of the refractive index of MoS₂, which are in turn related to the features in the imaginary part by the Kramers-Kronig relationship. Thus, Kramers-Kronig also informs us that if we set the imaginary part to 0, those feature should be smoothed out in the real part, so while the approximation for the bare plasmon provides is sufficient for to fit the coupled oscillator model shown in Figure 4-8, it could be improved by substituting a smooth approximation of the refractive index for MoS₂. See Chapter 5 for an example of this type of approximation.

To fit the model to the data in Figure 4-6a, the peak positions of the three branches were found by fitting the absorptivity $A(E)$ at each $k_{//}$ to three Lorentzians, repre-

sented by the black data points in Figure 4-8. The three k -dependent eigenvalues of the Hamiltonian represent the three coupled modes of the system and thus the three branches of the dispersion. A least square fitting routine is used to fit these eigenvalues to the experimental data (black points), shown as the orange lines in Figure 4-8. The best fit Rabi splittings are 65 meV and 72 meV at the A- and B-exciton energy positions, respectively. This underestimates the experimentally measured values (81 meV and 93 meV, A- and B-excitons, respectively), taken from the difference between the experimental peaks at the k -value at the intersection between the uncoupled exciton energies and uncoupled plasmon (gray and red lines in Figure 4-8). The fit A- and B-exciton energies are 1.85 eV and 2.00 eV, in good agreement with the experimentally determined values of 1.86 eV and 2.01 eV. The fit could be improved by using a better approximation of the bare plasmon, as discussed in the previous paragraph. Further, the Hamiltonian in eq 4.1 ignores the imaginary part, i.e. the line widths or damping portion, of the modes. As shown in Section 1.3, the imaginary part causes the evident Rabi splitting to appear smaller than the actual coupling strength, g . Thus, an improved model would also fit the line widths.

Each of the three coupled modes is a quantum superposition of the uncoupled modes, each coherently exchanging energy at the Rabi frequency. The three eigenvectors, $|\phi\rangle_k$, of eq 4.1 can be expressed in the basis of the uncoupled plasmon, $|SPP\rangle$, A-exciton, $|A\rangle$, and B-exciton, $|B\rangle$, modes, shown in eq 4.2, with the square modulus of their respective coefficients, $|c_n|^2$, giving the contribution of each uncoupled mode to the final state. Figure 4-9a shows these mixing or Hopfield [102] coefficients for each of the three branches.

$$[ht] |\phi\rangle_k = c_1 |SPP\rangle + c_2 |A\rangle + c_3 |B\rangle \quad (4.2)$$

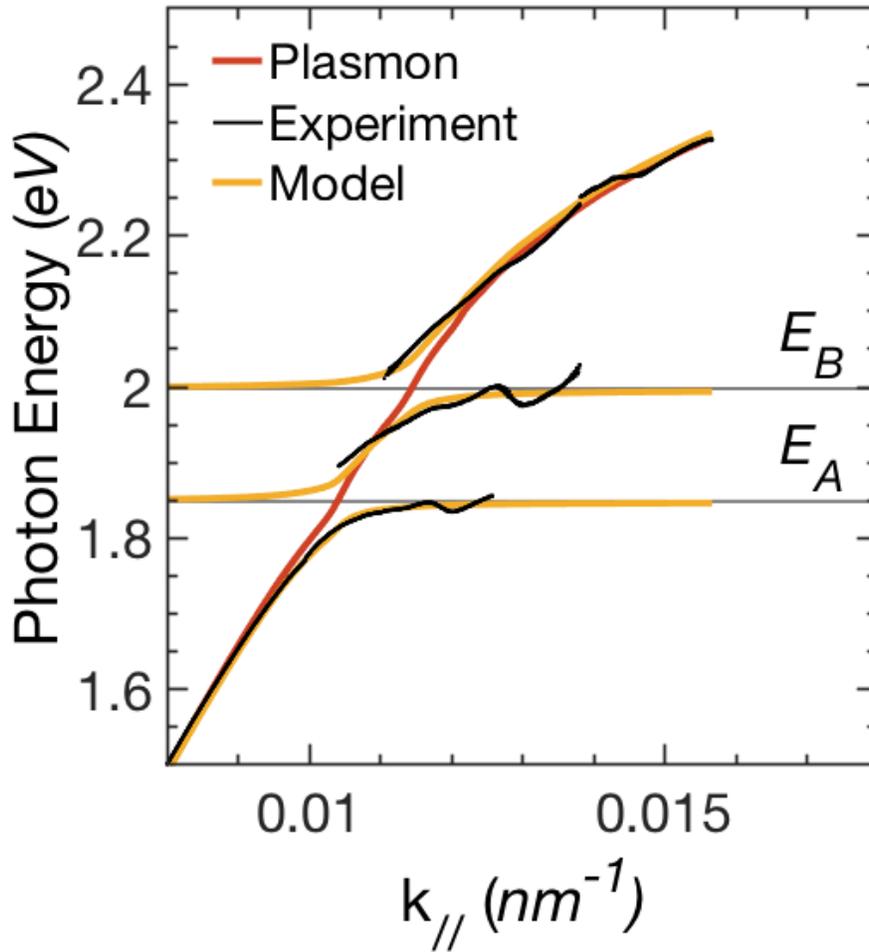


Figure 4-8: Semiclassical coupled oscillator model fit: Black data are peaks of the three hybrid modes taken from the experimental dispersion. Red data are peaks from the modeled uncoupled plasmon. Orange lines are the coupled harmonic oscillator fit to the black data. The gray lines labeled E_A and E_B are the uncoupled exciton positions, as fit by the model. Anticrossing of the three hybrid modes is a clear indication of a strongly coupled system. The modelled Rabi splittings are 65 meV and 72 meV at the A- and B-exciton energy positions, respectively.

At the limits of low energy and $k_{//}$ (lower branch) and high energy and $k_{//}$ (upper branch), the modes are purely plasmonic in nature, as the energies are far from the exciton energies. In the lower and upper branches at the crossing points of the uncoupled plasmon and exciton energies, the modes are half plasmonic, half excitonic in nature. The lower branch contains no contribution from the B-exciton and the upper branch none from the A. Interestingly, the middle branch contains non-negligible contributions from both the A- and B-excitons (15% each at their intersection), indicating the excitons and plasmon are coherently hybridized and this hybridization is mediated by the plasmon. Such polariton-mediated exciton-exciton hybridization has been previously observed in cavities coupled with organic molecules [103] and with plasmons coupled to quantum dots [104], both at room temperature. More recently in the TMDCs, Liu et al. showed such hybridization at 77 K in monolayer MoS₂ coupled to a Ag nanodisk array, evident in one of the hybrid mode branches [88], Wang et al. observed exciton-exciton hybridization in many-layer WS₂ Fabry-Perot modes at 77 K [105], and Cuadra et al. found exciton-trion hybridization at 6 K in WS₂ coupled to plasmonic nanoantennas [106]. Vasilevskiy et al. predicted room temperature hybridization of the A- and B-excitons in both MoS₂ and WS₂ monolayers in cavities [101]. Our observation of exciton-exciton hybridization in MoS₂ is the first at room temperature in any of the 2D TMDCs, to the best of our knowledge.

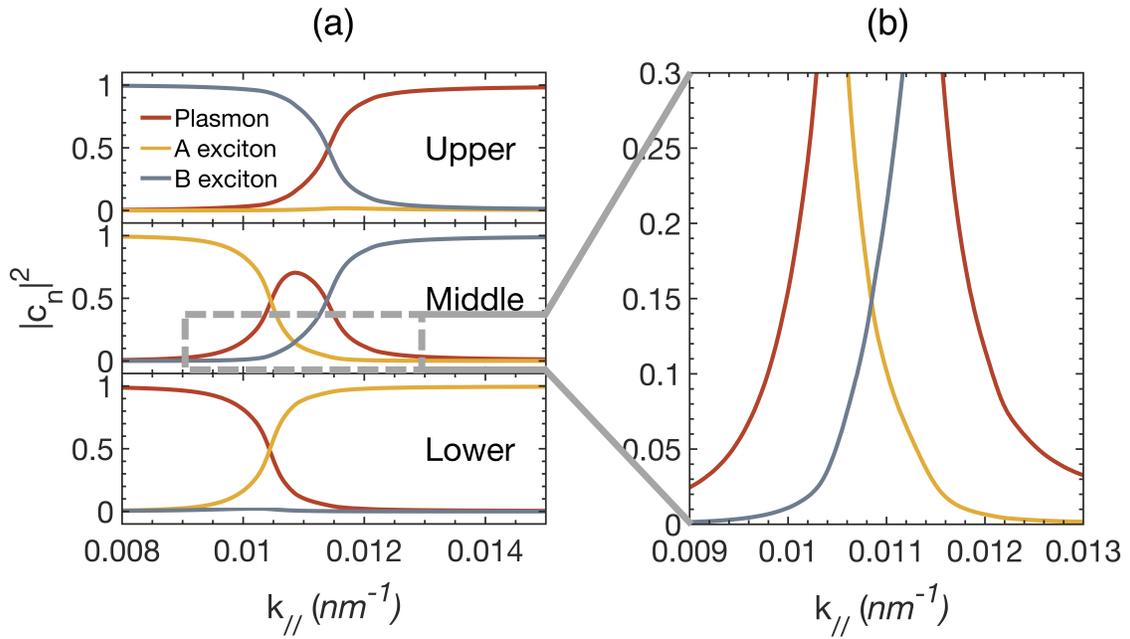


Figure 4-9: (a) Mixing coefficients for each of the three hybrid mode dispersion branches, as a function of transverse wavevector. The coefficients are obtained from the coupled harmonic oscillator model. (b) Mixing coefficients of the middle branch showing overlap of A- and B-excitons, implying coherent quantum superposition of the two modes, mediated by strong coupling to the plasmon.

The coherent exciton-exciton hybridization presents an interesting question: how do the hybridized excitons influence the decay dynamics of the valley-polarized carriers in MoS₂? Monolayer and odd numbers of FL-MoS₂ TMDCs exhibit broken inversion symmetry which leads to energy-degenerate spin-polarized valleys [107]. The valley-polarized excitons are normally short-lived at room temperature [108]; however, they survive at room temperature under strong coupling with polaritons in monolayer MoS₂ [109] where the coherent exchange between A-exciton and polariton modes increases the spin-polarized carrier lifetime and outcompetes decoherent intervalley scattering. The impact on the carrier lifetimes with the inclusion of coherent coupling to the B-exciton remains an open question. In theory, A- and B-excitons of like spin polarization exist in separate valleys, thus hybridization between the two may provide a mechanism to control intervalley scattering, possibly including coherent transfer of information between different valleys. Further studies are needed on the valley physics of the 2D TMDCs under strong coupling.

4.5 Application to other systems

Finally, the transfer matrix method was used to study exciton-plasmon coupling combinatorially in the exciton-plasmon MX₂-P system where M = (Mo, W), X = (S, Se), and P = (Ag, Au, Al, Cu). The dispersion was modeled similarly to that of Figure 4-6b, but using a simplified stack of NBK7 / metal / TMDC. Optical properties were taken from the literature: Ag and Au were from Johnson and Christy, [110] Al and Cu from McPeak et al., [111] TMDCs from Liu et al., [89] and NBK7 from a Schott specification sheet. Note that the complex refractive indices for the TMDCs are from monolayers. The refractive index differences between monolayer and few-layer TMDCs are small enough [95] to warrant using those of monolayer: the general takeaway from the model—that strong coupling should be observable in most of the

studied systems—should hold. The models were roughly optimized by varying the thickness of the metals and TMDCs to achieve the most prominent coupling features by eye. The following thicknesses were modeled: Ag (40 nm), Au (50 nm), Al (20 nm), Cu (50 nm), MoS₂ (5 nm), MoSe₂ (15 nm), WS₂ (5 nm), and WSe₂ (10 nm).

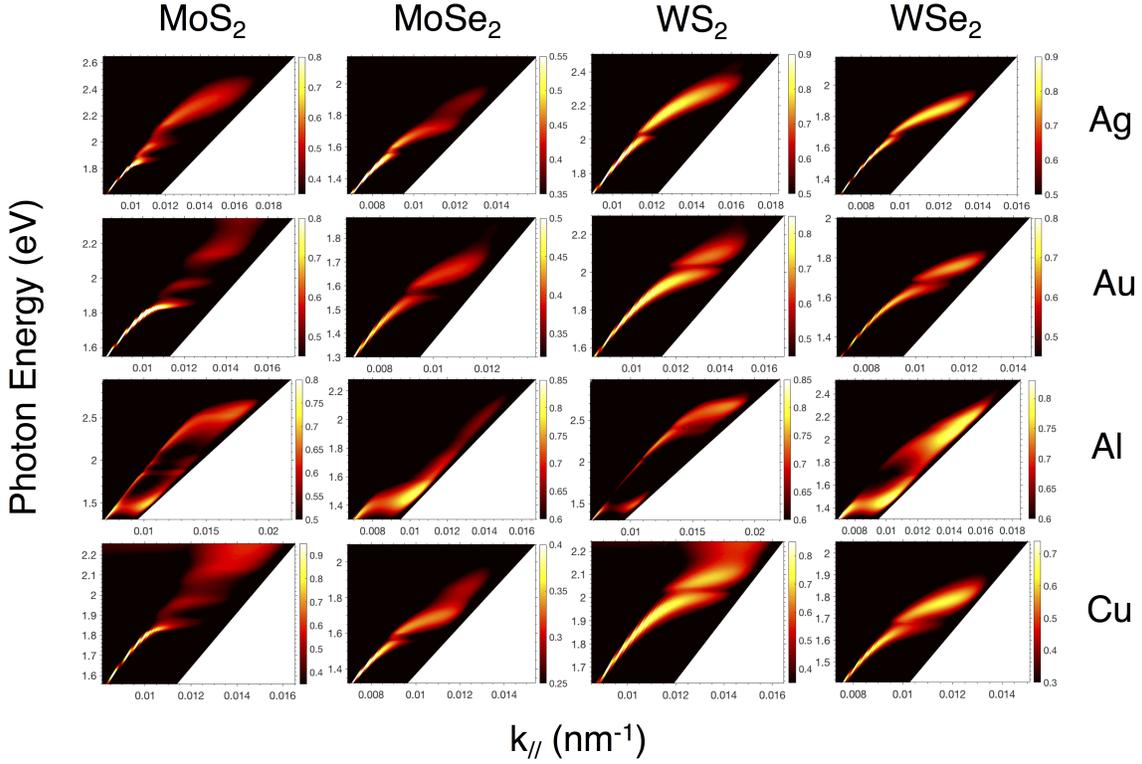


Figure 4-10: Generalized strong plasmonic coupling in 2D TMDC systems. A transfer matrix model is used and the thickness of each film is tuned to give the largest splitting. Rabi splitting is evident for each system except those with Al, where interpretation is more difficult. The color bar is absorptivity.

In Figure 4-10, Rabi splitting is clearly seen in each system, with the exception of Al. In Al, the plasmonic tuneability is limited to energies greater than ~ 1.8 eV and sharp resonances only available at higher energies, limiting strong coupling in the TMDCs studied here. The TMDCs were modeled with few-layer thickness, which as noted above, is necessary to achieve large Rabi splitting at the A- and B-excitons. The results suggest that the Kretschmann-Raether method may be a useful platform for studies under strong coupling conditions while varying other properties. For example, optical, electronic, and catalytic properties can be tuned by exchanging one metal or TMDC for another, while maintaining strong coupling.

4.6 Conclusions

To conclude, we have reported room temperature strong coupling of the A- and B-excitons of MoS₂ with plasmons as well as exciton-exciton hybridization. Our room temperature findings represent an important advance in the field of strong coupling as they point to a new possible platform for studies on room temperature threshold-less lasing and quantum information processing. Our finding of A- and B-exciton hybridization adds to the understanding of the valleytronic properties of the 2D TMDCs under strong coupling and suggests a potential method to coherently control quantum information. For example, one can imagine intervalley information transport via exciton-exciton hybridization, controlled in an active strong coupling device. More generally, our result shows it is important to consider exciton-exciton hybridization as a possible decay pathway in 2D TMDC exciton-polariton systems. Further, we demonstrate that the 2D TMDCs are ripe for strong coupling studies in the simple Kretschmann-Raether configuration, allowing researchers wide latitude in their choice of plasmonic metal or TMDC. The Kretschmann-Raether method, with its open geometry is conducive to biological and chemical studies as well. While there has been

much work on plasmonic biosensors, there has been little application of strong coupling to biology or chemistry, either of which could provide many opportunities for interesting research.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Enhanced strong coupling of band-nested excitons in few-layer TMDCs

5.1 Abstract

We show Rabi splitting of 244 meV at the C-exciton in the two-dimensional (2D) transition metal dichalcogenide (TMDC) MoS₂. The splitting is 9.5 % of the transition energy (2.57 eV), which is $\sim 2.5\times$ the ratio observed in similar experiments at the A- and B-excitons and approaches the ultrastrong coupling limit. The large enhancement in the coupling strength is likely due to the nested conduction and valence bands found in the 2D TMDCs which leads to strong absorption. We measured the k -space dispersion of few-layer MoS₂ (FL-MoS₂) in a tunable plasmonic cavity at room temperature to observe the Rabi splitting. The band-nesting region of 2D TMDCs is under study for spontaneous charge-separation and slowed hot carrier cooling. Our observation suggests that strong coupling of the C-exciton in the 2D TMDCs may improve carrier lifetimes for these processes as well as enhance other quantum coherent phenomena such as polariton condensation or molecular hybridization for chemical reactions.

5.2 Background

5.2.1 Strong coupling and the 2D TMDCs

This chapter is based on my work intended for publication. [112] Further details of the methods used can be found in Chapter 2.

As described in the previous chapter, two-dimensional (2D) transition metal dichalcogenides (TMDCs) have intriguing optical, electronic, and catalytic properties. [71,72] For example, they have extraordinarily high absorption coefficients—a single monolayer can absorb $\sim 10\%$ of incident light. Their absorption is characterized by sharp excitonic features, making the TMDCs good candidates for strong exciton–polariton coupling. Strong coupling is the coherent interaction of two equal-energy atomic transitions, yielding new hybridized modes, with characteristic energy splitting, called Rabi splitting, about their uncoupled resonant energy. [16] Strong coupling is possible when the two modes overlap in real space and energy–momentum space and when their coupling rate exceeds the average dissipation rate of the individual modes.

In exciton–polariton systems, strong coupling conditions have been exploited to achieve polariton condensation, [75] analogous to Bose-Einstein condensation, where the energy of the system contained in the hybrid modes collapses to the minimum in momentum space. Coherent light emission has been observed from polariton condensation and theory shows that thresholdless lasing, that is lasing upon excitation of as few as a single photon, should be possible from such a state. [36, 75] There is some evidence to support this prediction. [74] Such a quantum coherent phenomenon on the single photon level would be useful for quantum computing applications. [15] Work in this field is still ongoing and new systems that exhibit strong coupling at room temperature are needed.

Another application of strong coupling is to modify the chemical behavior of materials. Strong coupling results in shifted band positions, which can lead to fundamen-

tally new electronic and catalytic properties. Thus, there has been some work showing that chemical photoreaction pathways and rates can be manipulated by strongly coupling the reactants with polariton modes. [78–80, 113] As an example of how strong coupling can influence chemistry, consider that the band alignment between photoexcited modes and empty orbitals of chemical reactants is a factor in determining the transfer of photoexcited charge to a chemical species. By varying the band alignment through strong coupling, the charge transfer rate and therefore the reaction rate can be modified. See Section 1.3 for further details on the weak and strong coupling regimes.

5.2.2 The C-exciton of 2D TMDCs

Most strong exciton–polariton coupling research is focused on coupling to the lowest energy excitons. In this chapter we consider coupling to a higher order C-exciton in 2D molybdenum disulphide (MoS_2). In the 2D TMDCs, near the band edge, the two lowest energy exciton transitions are called the A- and B-excitons. These excitons correspond to transitions between the conduction band minimum and the spin–orbit-split valence band maxima at the K point in Figure 5-2. The C-exciton oscillator strength is significantly higher than the A or B, as seen by comparing the peak heights in Figure 5-1. The A-, B-, and C-exciton peaks are found at 689 nm, 605 nm, and 482 nm, respectively. Interestingly, the C-exciton does not arise from transitions between two extrema, but rather between regions of parallel valence and conduction band, referred to as nested bands, shown by the green arrows on either side of the Λ point in Figure 5-2. [114] To understand the nested band C-exciton absorption better, consider that 1. the density of states in the electronic band structure is proportional to the inverse slope of the bands and 2. photon absorption is optimized for transitions between the valence and conduction bands when they have equivalent densities of states—i.e., for every unoccupied state in the valence band there is an

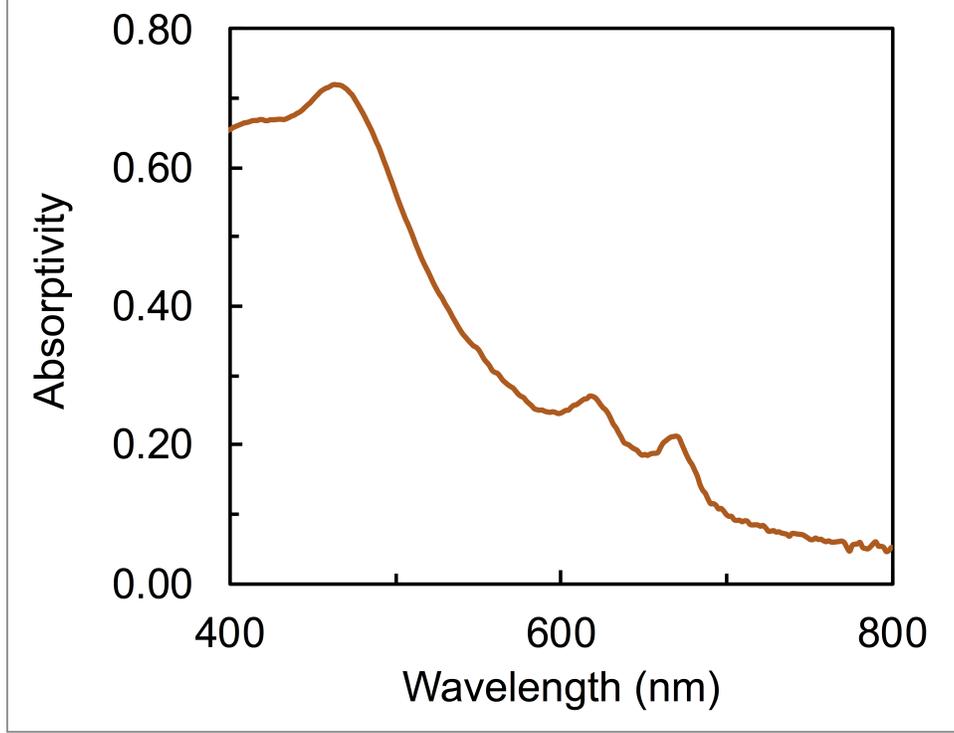


Figure 5-1: Absorptivity, A , of MoS₂ film on 40 nm Ag on glass. Reflectance of film was taken and A calculated as $A = 1 - R$, where transmission is assumed to be negligible.

occupied state in the conduction band, and vice versa. This relationship of allowed optical transitions to electronic density of states is described by the joint density of states. The absorption coefficient is proportional to the joint density of states which is proportional to $\frac{1}{\nabla(E_{CB}-E_{VB})}$, where E_{CB} is the conduction band and E_{VB} is the valence band. [114] Therefore, in regions of parallel bands, $E_{CB} - E_{VB} = 0$ and the absorption is maximized. Further, the k -space over which this condition is met is much larger than regions of aligned parabolic bands, where $E_{CB} - E_{VB} = 0$ for just a single k -point. Thus, the high C-exciton oscillator strength is a direct consequence of the large region of parallel bands found in the band structure.

An unusual consequence of the nested bands is that C-exciton carriers are expected to spontaneously self separate in momentum space as well as have slowed hot carrier cooling relative to the A- and B-excitons. [115] Hot carriers are defined as

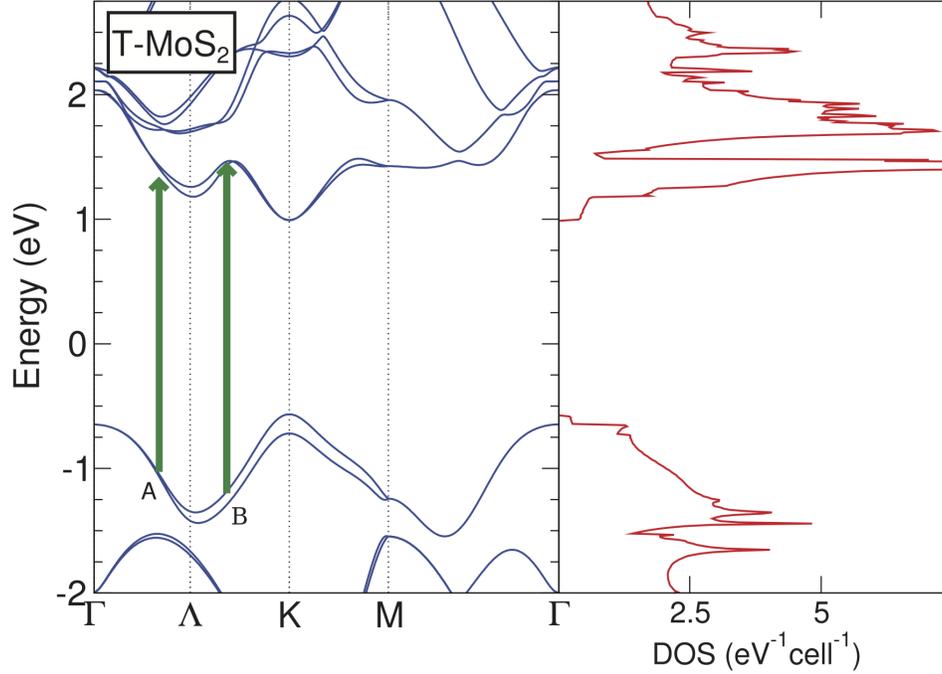


Figure 5-2: Band structure and density of states of MoS₂, predicted from density functional theory. The arrows labeled A and B point out transitions between nested bands that lead to peaks in the absorption spectrum (and do not refer to the A- or B-excitons). Reprinted with permission from ref [114]. Copyright 2013 by the American Physical Society.

photoexcited electron-hole pairs with greater collective energy than the band gap. Generally, these carriers relax to the conduction band minimum/valence band maximum faster than they may be extracted, limiting the photovoltage to the bandgap voltage. The potential slowing of hot carrier relaxation processes make the TMDCs a candidate for hot carrier photovoltaic studies, where the photovoltage is limited to the photon voltage. Further, the self-separation of charge may be useful for carrier-selective contacts. [116] Let's examine the nature of the C-exciton absorption and decay mechanisms. On the one hand, decay via phonon emission should be higher for the C-exciton than the A- and B-excitons which are trapped in potential wells across the bandgap, thus the C-exciton may exhibit reduced lifetime compared to the A- and B-excitons. On the other hand, C-exciton thermalization occurs until the carriers reach their own respective potential wells at the Γ and Λ points of the Brillouin

Zone (see Figure 5-2). This separation in momentum space reduces the likelihood of photon emission as at least one phonon is needed for momentum matching, thus the carriers lifetimes should be enhanced over the A- and B-excitons. Indeed, the sharpness of the C-exciton absorption features gives some indication that the lifetime isn't ultrashort.

The literature, however isn't clear on the lifetimes of the C-excitons compared to the A and B. Li et al. showed C-exciton biexponential decay constants of 1.4 and 13.5 ps and set a lifetime upper limit of 70 ps from intervalley scattering. [117] Wang et al. reported carrier cooling of the C-excitons with biexponential lifetimes of 23 and 350 ps, and significantly slower cooling than the A- and B-excitons. [115] Wang et al. cited upconversion from the band edge to the C-exciton as a possible explanation for the slowed cooling; exciting band edge excitons gave identical decay of the C-exciton as when excited directly. Kozawa et al. shows greatly subgap photoluminescence (PL) following excitation of the C-exciton which is ascribed to indirect emission from the cooled carriers sitting in the Γ and Λ valleys; this region of the spectrum is not probed in the other works. [118] Kozawa also notes that defect, trap, and impurities can greatly influence the non-radiative decay in the TMDCs. Finally, Wang et al. also coupled MoS₂ to graphene and calculated 80 % charge extraction efficiency from the C-exciton, showing the potential for harvesting hot carriers. [115] Strong coupling has been shown to increase carrier lifetimes and so may be beneficial to hot carrier extraction from the C-exciton.

Another intriguing demonstration of C-exciton physics is the 400-fold enhancement of second harmonic generation (SHG) from the C-exciton in monolayer WS₂ when weakly coupled to surface plasmons. [119] SHG may occur in mono- and odd numbered few-layer TMDCs due to the broken inversion symmetry. [107] SHG may also be possible in even-numbered few-layer TMDCs when gated, since gating lifts their inversion symmetry. [120] Strong coupling has also been used to enhance SHG

in organics and TMDCs while also modifying the emission spectra via Rabi splitting. [121, 122] Thus, strong coupling may offer a way to further tune and enhance the SHG, as well as other non-linear properties, at the C-exciton in the TMDCs.

5.2.3 Strong coupling at the C-exciton

This work studies strong exciton–polariton coupling at the C-exciton in 2D few-layer MoS₂ (FL-MoS₂) at room temperature. We find a large coupling strength of $g = 122$ meV (Rabi splitting of $\Omega = 2g = 244$ meV) which is $\eta = 4.7$ % of the uncoupled C-exciton transition energy, $E_C = 2.57$ eV, where $\eta = \frac{g}{E_C}$. Compared to splitting at the A- and B-excitons in similar systems, where ~ 70 meV was reported in ref. [70], this represents a ~ 150 % increase in η , and approaches the ultra-strong coupling regime ($\eta > 10$ %) where quantum coherent phenomena among coupled states becomes very efficient. [44] Our findings suggests that the C-exciton of the TMDCs is a suitable platform for studying strong and potentially ultrastrong coupling physics. Possible applications include slowed hot carrier cooling for photovoltaics, enhanced SHG for coherent light sources in the violet/blue, and enhanced photocatalysis by charge injection from the C-exciton to unoccupied molecular orbitals of chemical reaction intermediates.

5.3 Fabrication and data collection

Details of fabrication and experimental methods are given in Chapters 2 and 4. Brief details are provided in this section.

Strong exciton–plasmon coupling was studied with FL-MoS₂ on Ag on a glass prism in the Kretschmann-Raether configuration. For this experiment, a single FL-MoS₂ film was grown by chemical vapor deposition and transferred to the Ag-coated glass prism. Approximately 40 nm Ag was deposited onto the prism with a ~ 1 nm

Ti film for adhesion between the Ag and glass. The FL-MoS₂ was found to be ~10 nm thick by ellipsometry and ~8 nm thick by fitting a transfer matrix model to the data of the exciton–plasmon dispersion.

Dispersion data was obtained from angle-resolved reflectance through the back side of the prism. Reflectance, R is transformed first to absorptivity, A , via $A = 1 - R$. Then, the data which is collected for various angles is transformed to k -space by $k_{//} = (2\pi/\lambda)n_g \sin \theta_i$, where $k_{//}$ is the wavevector parallel to the surface plasmon propagation and surface of the prism, λ is the incident photon wavelength, n_g is the index of refraction of the prism, and θ_i is the incident angle of light inside the prism. All analysis is repeated in the original angle basis as well for comparison. All measurements were performed at room temperature in air.

5.4 Analysis in k -space

5.4.1 Experimental and simulated exciton–plasmon dispersion

We start by observing the experimental dispersion, shown in Figure 5-3a. Compared to the work of Chapter 4, the features near the A- and B-excitons at 1.85 eV and 2.0 eV, respectively, are only slight perturbations to the plasmon dispersion, indicating minimal plasmon coupling to these excitons. However, at the C-exciton position of 2.57 eV, Rabi splitting, or avoided crossing, is observed. The effect here is quite large and clear, with minimal overlap of two coupled modes due to large splitting and therefore large coupling strength.

We compared the experimental dispersion to transfer matrix model simulations of the same system, shown in Figure 5-3b. [13] The simulated system reproduces the experimental data well. The system in Figure 5-3b was modeled as NBK7 substrate / 1 nm Ti / 43 nm Ag / 0.27 nm Ag₂S / 8 nm MoS₂. The substrate refractive index and thicknesses of Ag and MoS₂ were fit until the model reproduced the data. The

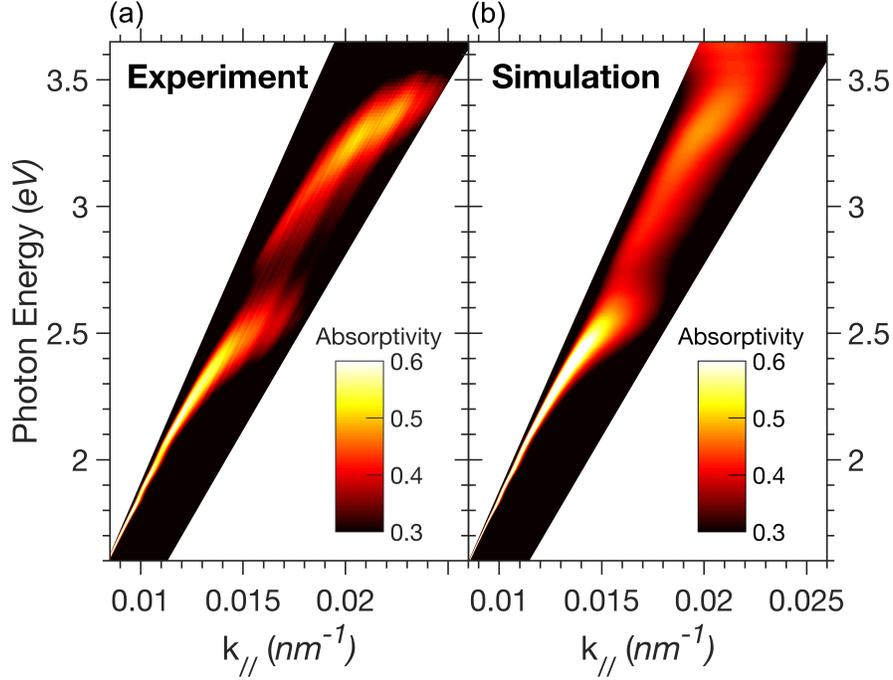


Figure 5-3: Experimental (a) and simulated (b) dispersion in k -space. (a) Stack is NBK7, 40 nm, ~ 10 nm FL-MoS₂. Little variance is seen at the A- and B-exciton positions near 1.8 and 2.05 eV. Avoided crossing is seen at the C-exciton, 2.57 eV. (b) Transfer matrix method simulation of the stack in (a). The model matches the experimental data well, with similar features and magnitude. The gap region is less pronounced and cut off near 3.5 eV less sharp. Absorptivity is calculated as $1 - R$, where R is reflectance.

NBK7 refractive index fit matches specifications of Schott within $\sim 2\%$. The fit value of 8 nm is near the value measured by ellipsometry of 10 nm. Refractive indices for Ag were modeled from ellipsometry data and are in-line with those of Palik [92] and Johnson and Christy [6] and the fit thickness was within 2 nm of that measured with ellipsometry. Refractive indices for Ti and Ag₂S were taken from refs [90] and [91], respectively. The thickness of Ti was set to be 1 nm while the thickness of Ag₂S was derived from ellipsometry.

5.4.2 Coupled oscillator model fit

The transfer matrix simulations serve to verify that our observations are to be expected based on applying Maxwell's equations to the system at hand. However, they do not quantify the strength of the coupling or the composition of the new hybrid modes. To gain this insight, we solve the problem using semiclassical coupled oscillator model. The Hamiltonian can be written as eq 5.1:

$$\begin{pmatrix} E_{SPP}(k_{//}) & \frac{\Omega_A}{2} & \frac{\Omega_B}{2} & \frac{\Omega_C}{2} \\ \frac{\Omega_A}{2} & E_A & 0 & 0 \\ \frac{\Omega_B}{2} & 0 & E_B & 0 \\ \frac{\Omega_C}{2} & 0 & 0 & E_C \end{pmatrix} \quad (5.1)$$

In this model, the problem is treated as a four-level quantum system coupled to classical electromagnetic fields. $E_{SPP}(k_{//})$ is the uncoupled or "bare" plasmon energy, which is a function of in-plane wave vector as given by its dispersion. E_A , E_B , and E_C are the bare exciton energies. Ω_A , Ω_B , and Ω_C are the Rabi splitting values, equal to twice the coupling strength. The off-diagonal zeros represent that there is no coupling between excitons, which is expected to be the case. The Hamiltonian is solved and the eigenvalues give the coupled modes. Therefore, our 4×4 matrix produces four coupled modes. The model is fit by the least square routine to the experimental data. To fit to the data, each branch is fitted to a Lorentzian for each k -point, and the model fits to the peak energies of these Lorentzians which are plotted in black in Figure 5-4. The fit modes are plotted in orange. To do the fitting, E_A , E_B , and E_C are set as fit parameters, with initial values taken from the positions of minima in reflectance off the top of the MoS₂, where coupling to plasmons is disallowed to

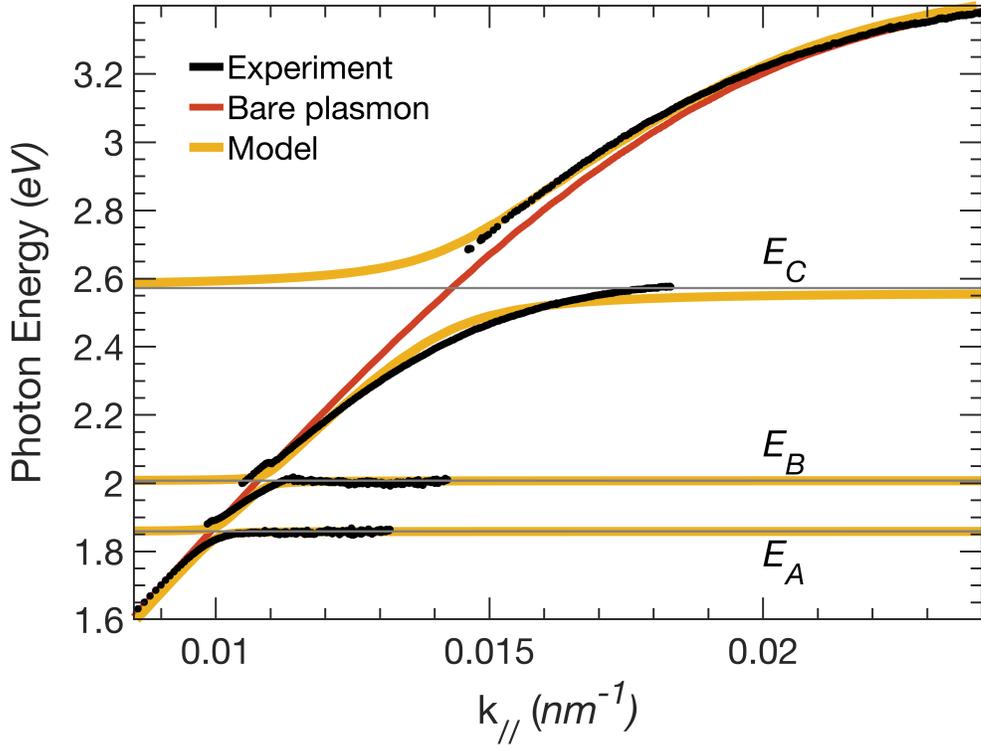


Figure 5-4: Coupled oscillator model fit to data. Black data points are from experiment. Red line is simulated uncoupled plasmon used in the model. Orange lines are coupled oscillator model fit to the data. Gray lines are model fit of A-, B-, and C-exciton energy positions.

momentum mismatch. Ω_A , Ω_B , and Ω_C are also set as fit parameters. Initial values of Ω are found by subtracting the branches on either side of each exciton and taking the minimum difference. $E_{SPP}(k_{//})$ is not set as a fitting parameter, but is fed into the model.

Determining the bare plasmon energy, shown by the red line in Figure 5-4, is more complicated than simply measuring the dispersion of a control Ag sample with no MoS₂. As noted in Chapter 4, plasmons are sensitive to their local environment, where local is defined by the near-field of the electromagnetic plasmon wave, ~ 10 nm. Therefore, the presence of several nanometers of any material, lossy or not, will modify the plasmon dispersion. In this case with MoS₂ on Ag, the plasmon asymptotically

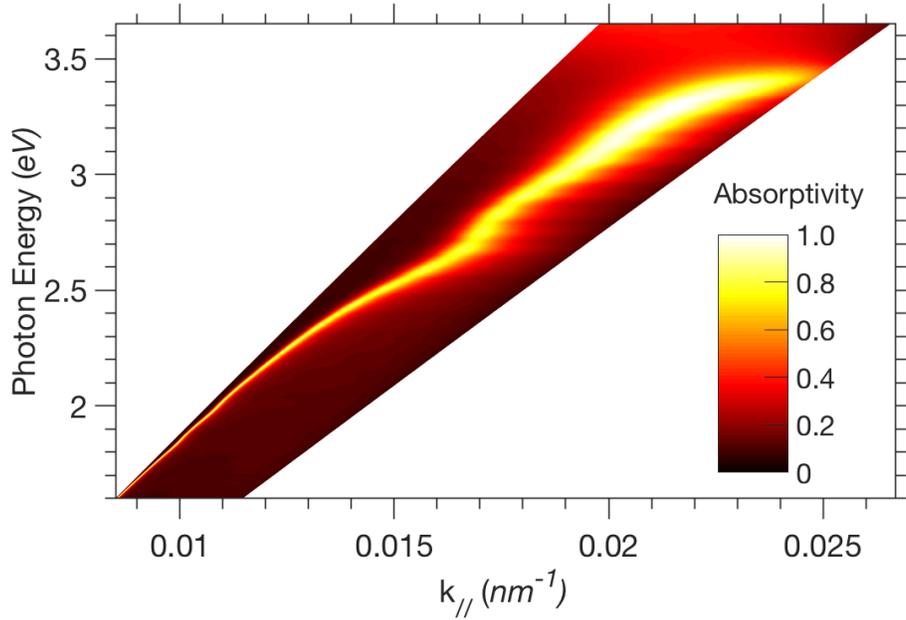


Figure 5-5: Transfer matrix model dispersion of bare plasmon with "lossless" MoS₂. Same as Figure 5-3b, but with the imaginary part of the complex refractive index of MoS₂ set to zero. This method of approximating the bare plasmon is not Kramers-Kronig consistent and results in a poor fit of eq 5.1 to the data.

approaches ~ 3.3 eV, which is red-shifted from the bare Ag case (see Figure 4-7a). In solving the similar problem in Chapter 4, the bare plasmon was approximated by solving the transfer matrix model of the system with lossless MoS₂. The lossless MoS₂ was modeled by setting the imaginary part of the complex MoS₂ refractive index to zero. In the case here, however, that approach had to be modified. Due to the Kramers-Kronig relationship, the real and imaginary parts of the refractive index are coupled such that a peak or valley in one corresponds to an inflection point in the other. Therefore, setting the imaginary part of the refractive index to zero to model a lossless version of that material is not physically accurate, as features are maintained in the real part that should disappear, as shown in Figure 5-5. Plugging the modeled plasmon from Figure 5-5 into the Hamiltonian of eq 5.1 will yield a poor fit to the data. Instead, a Kramers-Kronig consistent transformation must be used, whereby the features of the real part of the refractive index must be smoothed out.

Table 5.1: Best fit parameters from coupled oscillator model for strong coupling to the C-exciton in MoS₂. Solved in k -space.

	E_A (eV)	E_B (eV)	E_C (eV)	Ω_A (meV)	Ω_B (meV)	Ω_C (meV)	$\eta_C = \frac{\Omega_C}{2E_C}$
Experiment	1.86	2.01	2.66	-	-	-	-
Model	1.86	2.01	2.57	35.3	41.4	244	4.7 %

Since we're setting the imaginary part to zero, or a constant, the real part should also be constant. We found setting the index to $n = 1.5$, while keeping all else the same in the model from Figure 5-3b yielded a good fit of the plasmon to the data at the energy extrema, as shown by the bare plasmon in Figure 5-4. Note though that the Hamiltonian in eq 5.1 ignores the imaginary part, i.e. the line widths or damping portion, of the modes. As shown in Section 1.3, the imaginary part causes the evident Rabi splitting to appear smaller than the actual coupling strength, g . Thus, an improved model would also fit the line widths.

The best fit values for the bare exciton energies and Rabi splitting values are summed up in Table 5.1. The bare exciton energies obtained from minima in the reflectance spectra from the top of the MoS₂ film are also shown. The fit values of the bare A- and B-excitons are the same as experiment within error. The C-exciton energies differ by 9 meV. We note that in our experience the position of the C-excitons of mono-layer and FL-MoS₂ has a larger variability across experiments and substrates than the A- and B-excitons.

Figure 5-6 shows the oscillator model fit to the dispersion near the C-exciton. We see that the estimated Rabi splitting is 244 meV, which corresponds to $\eta = 4.7\%$. The fit Rabi value matches the separation between the data points near this k -point well. However, the coupling strength is likely larger than half the Rabi splitting, as

shown in Section 1.3 in Chapter 1, when the line widths involved are similar to the Rabi splitting, the observed splitting is less than $2g$. A model including the uncoupled mode line widths would more accurately capture this effect.

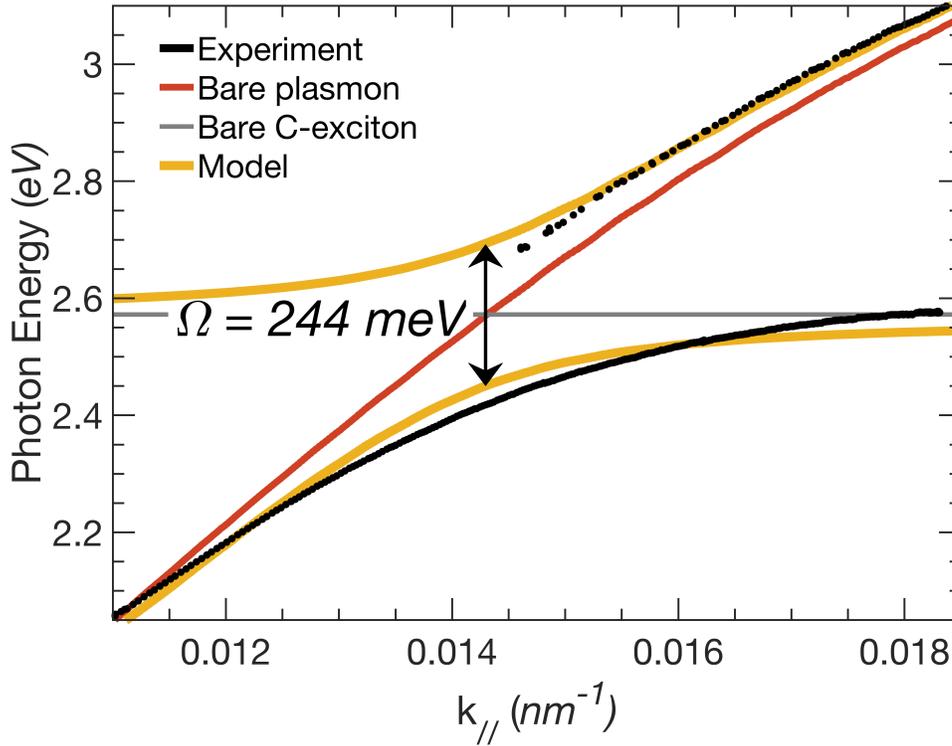


Figure 5-6: Coupled oscillator model fit, near C-exciton. Fit value of Rabi splitting is $\Omega = 244 \text{ meV}$ or 9.5% of splitting energy, which fits the data well.

Since 2005, strong coupling has started entering the ultrastrong coupling regime. One of the first papers used an organic J-aggregate dye molecule coupled to a plasmonic nanohole array to achieve 250 meV Rabi splitting at 1.8 eV transition energy, a ratio of $\eta = 6.9\%$. [123] In 2009, $\eta = 10\%$ was reached with J-aggregates coupled to a plasmonic nanodisk array. [124] This was pushed to $\eta = 16\%$ in another organic molecule, spiropyran, couple do a dielectric cavity mode. [125] In 2014, the organic semiconductor squaraine was coupled again to a microcavity to achieve $\eta = 30\%$, truly entering the ultrastrong coupling regime. [126] In the inorganic semiconductors,

the ultrastrong coupling regime was first seen in 2009 by coupling microcavity modes to intersubband transitions (~ 100 meV) in GaAs to achieve $\eta = 22\%$. [127] The largest demonstration of ultrastrong coupling at optical frequencies in the TMDCs was in many-layer WS_2 in 2016. The Fabry-Perot resonance of a 32 nm flake was coupled to the A- and B-excitons to achieve 270 meV and 780 meV, or $\eta = 6.7\%$, 15.6% , respectively. [105] There are many other reports of $\eta = 2 - 5\%$ in the TMDCs. [106, 128–130]

All of the reports of strong/ultrastrong coupling with TMDCs couple to the lowest energy A- or B-excitons. This work shows coupling to the C-exciton is also possible. The C-exciton has a large oscillator strength, the previously described band-nesting properties, and sits higher in energy than the A- or B-excitons. These features may find use in charge- or energy-transfer mediated photocatalytic chemical reactions, where the C-exciton may have favorable energy and band-alignment for coupling to molecular orbitals of reaction species, intermediate, or products.

5.4.3 Coupled mode Hopfield coefficients

The new hybrid modes, $|\phi\rangle_k$, may be written in the basis of the uncoupled modes according to eq 5.2.

$$|\phi\rangle_k = c_1 |SPP\rangle + c_2 |A\rangle + c_3 |B\rangle + c_4 |C\rangle \quad (5.2)$$

The Hopfield coefficients [102] $|c_n^2|$ represent the plasmonic and excitonic (A-, B-, or C-) weighting of the hybrid modes, as a function of wavevector. Figure 5-7 plots these coefficients for each mode.

In Figure 5-7, we see that unlike earlier work with stronger coupling to the A- and B-excitons, there is very little hybridization between the A- and B-exciton. Also, due to the large energy difference between the B- and C-excitons, there is very little hybridization between the two. The analysis shows the expected result that at the

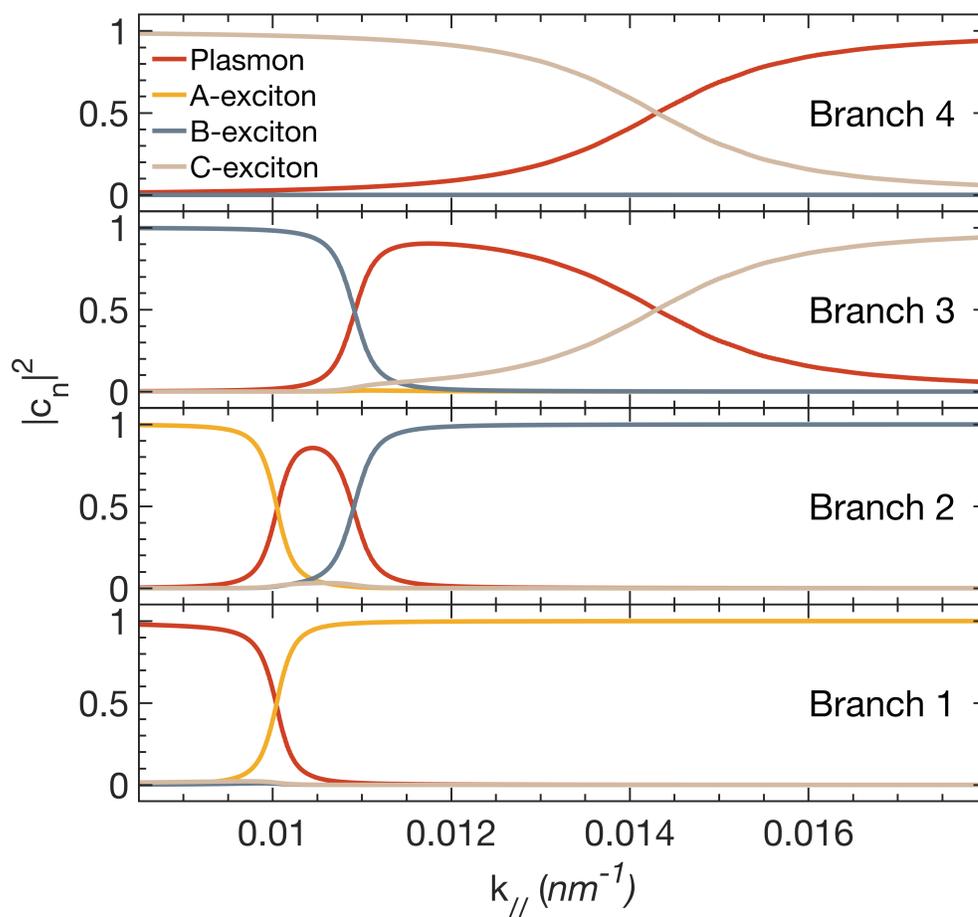


Figure 5-7: Hopfield/weighting coefficients of coupled modes in terms of uncoupled modes, taken from semiclassical oscillator model fit to experimental data. Branch 1 is the lowest energy branch seen in the dispersion (Figure 5-4) while Branch 4 is the highest branch. Coupling between excitons is marginal.

crossing point of the uncoupled modes the weighting is a 50-50 exciton–plasmon hybrid.

5.5 Analysis in angle-space

5.5.1 Experimental and simulated exciton–plasmon dispersion

The following figures show how the analysis unfolds if the reflectance/absorptivity vs. angle data is used without transformation into momentum space. This analysis results in impressive Rabi splitting, but is not the appropriate basis to work in. [99] Still, it does show that at the point of minimum separation between coupled modes near the C-exciton, the Rabi splitting seen experimentally is 607 meV, which corresponds to $\eta_\theta = 12\%$ for C-exciton transition energy of 2.57 eV. In k -space, $\eta_k = 4.7\%$ is seen.

Figure 5-8 shows the experimentally obtained and simulated dispersion. The transfer matrix simulation fit isn't as good as in the k -space simulation, but it still shows qualitatively that splitting of the C-exciton is to be expected.

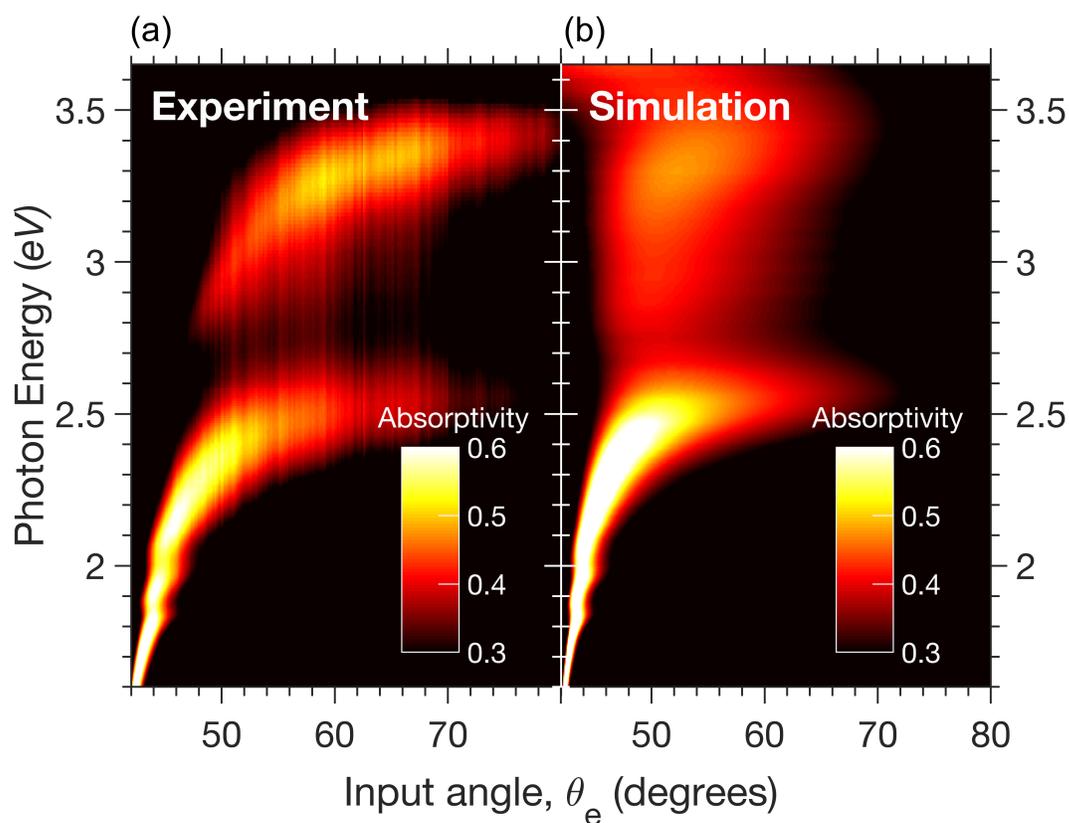


Figure 5-8: Experimental (a) and simulated (b) dispersion in angle-space. (a) Stack is NBK7, 40 nm, ~ 10 nm FL-MoS₂. Little variance is seen at the A- and B-exciton positions near 1.8 and 2.05 eV. Avoided crossing is seen at the C-exciton, 2.57 eV. (b) Transfer matrix method simulation of the stack in (a). The model matches the experimental data well, with similar features and magnitude. The gap region is less pronounced and cut off near 3.5 eV less sharp, while the shape of the simulated data compressed slightly in angle-space. Absorptivity is calculated as $1 - R$, where R is reflectance.

5.5.2 Coupled oscillator model fit

Applying the coupled oscillator model in angle-space, we see from Figure 5-9 that the fit is not as good again as in k -space, but shows the same general trends.

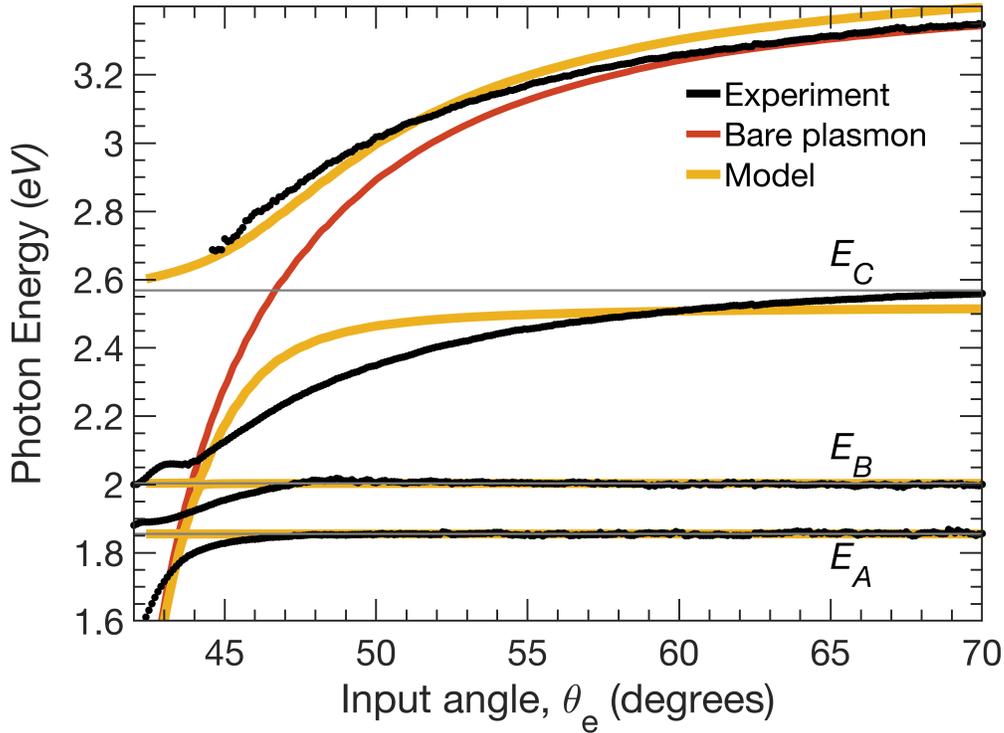


Figure 5-9: Coupled oscillator model fit to data in angle-space. Black data points are from experiment. Red line is simulated uncoupled plasmon used in the model. Orange lines are coupled oscillator model fit to the data. Gray lines are model fit of A-, B-, and C-exciton energy positions. No Rabi splitting is predicted at the A- and B-excitons.

Zooming in on the C-exciton fit, Figure 5-10 shows the model-extracted Rabi splitting of 423 meV at the bare exciton-plasmon crossing angle.

The extracted Rabi splitting is obviously underestimated, so in Figure 5-11, we extract the Rabi splitting at the same angle by taking the difference in energy between the two experimental curves. The experimental Rabi splitting here is 607 meV, which corresponds to $\eta = 12\%$.

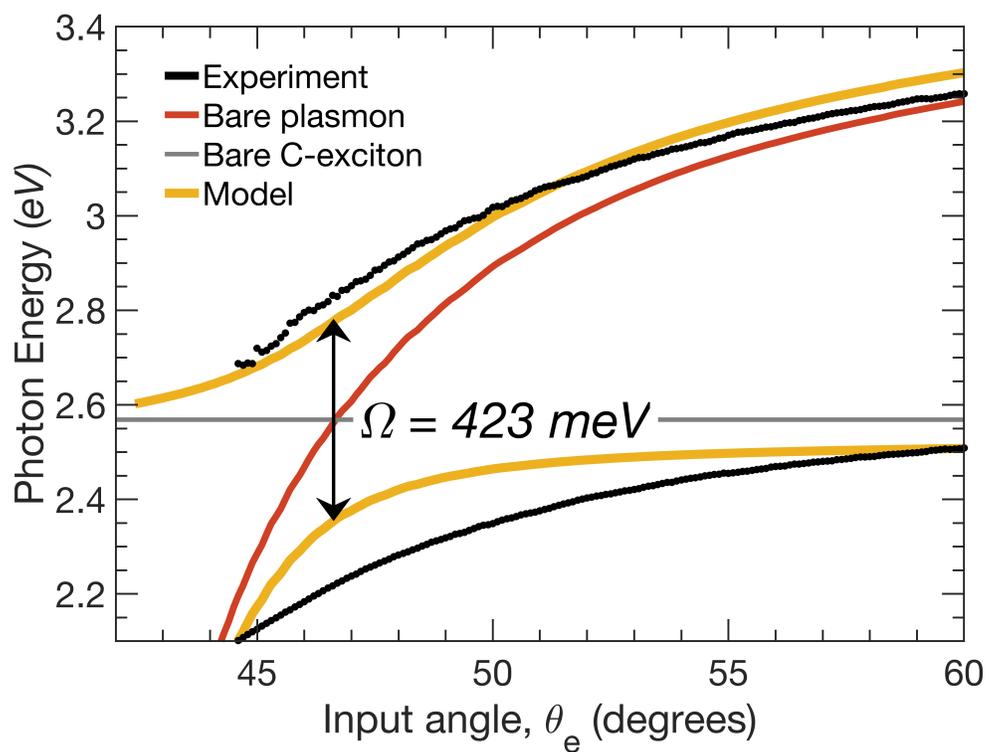


Figure 5-10: Coupled oscillator model fit, near C-exciton, in angle-space. Fit value of Rabi splitting is $\Omega = 423 \text{ meV}$. Fit is poor in angle-space.

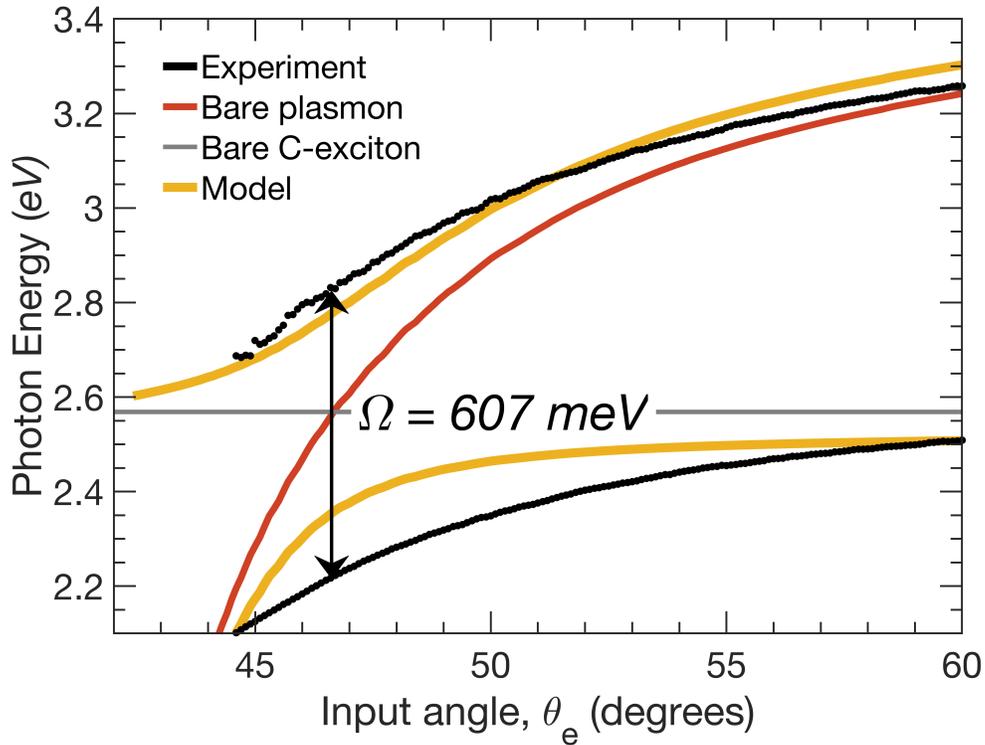


Figure 5-11: Rabi splitting extracted from experimental data at bare exciton-plasmon crossing point, in angle-space. Angle space shows massive Rabi splitting of 607 meV or 23.6 % of bare exciton (using fit value of C-exciton, 2.57 eV), while k-space shows 9.5 %. However, the more fundamental characterization of system is in k-space.

Table 5.2: Best fit parameters from coupled oscillator model for strong coupling to the C-exciton in MoS₂. Solved in angle-space.

	E_A (eV)	E_B (eV)	E_C (eV)	Ω_A (meV)	Ω_B (meV)	Ω_C (meV)	$\eta_C = \frac{\Omega_C}{2E_C}$
Experiment	1.86	2.01	2.66	-	-	607	11.4 %
Model	1.86	2.00	2.57	-0.01	-0.03	423	8.2 %

The best fit parameters from the coupled oscillator model in angle-space are presented in Table 5.2. The model fits effectively zero Rabi splitting at the A- and B-excitons and the fit value for the C-exciton transition energy is 3.5 % higher than the actual value.

5.5.3 Coupled mode Hopfield coefficients

The Hopfield coefficients, shown in Figure 5-12 show similar behavior as in k -space, with an asymmetry due to the angle-space presentation.

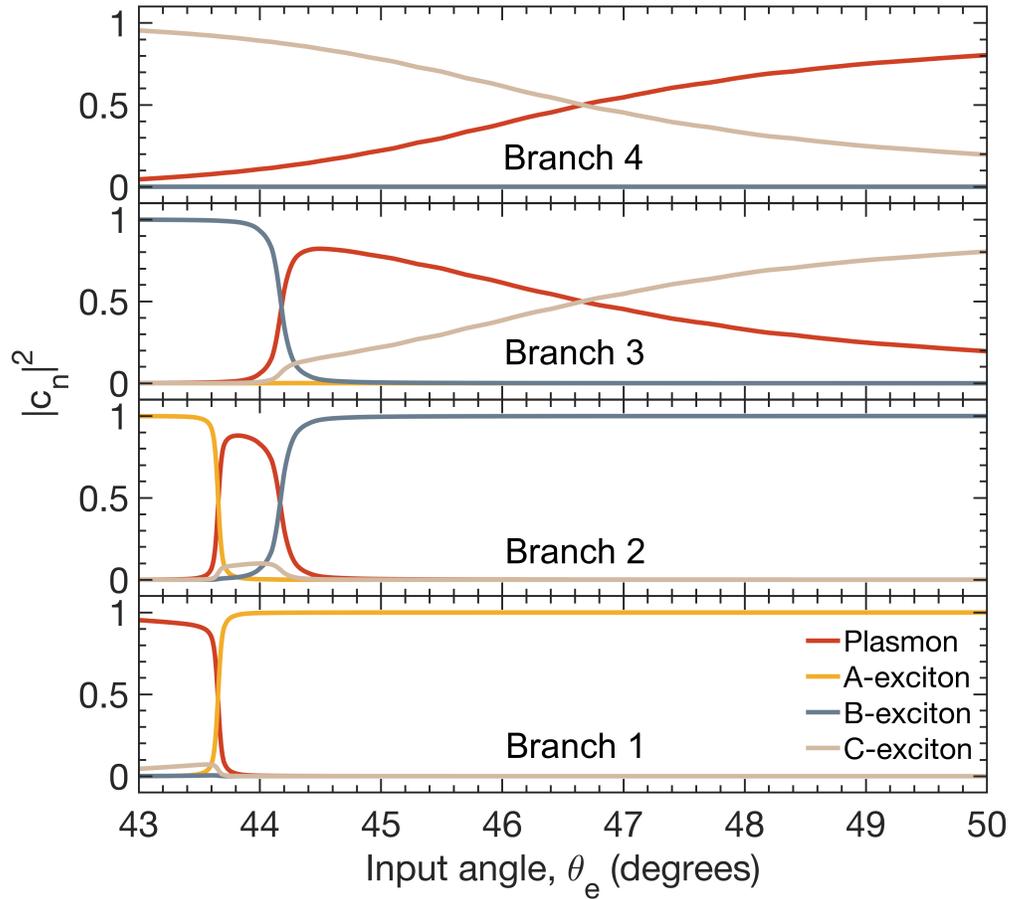


Figure 5-12: Hopfield/weighting coefficients of coupled modes in terms of uncoupled modes, taken from semiclassical oscillator model fit to experimental data, in angle-space. Branch 1 is the lowest energy branch seen in the dispersion (Figure 5-4) while Branch 4 is the highest branch. Coupling between excitons is marginal. Asymmetry of curves is due to angle-space representation.

5.5.4 Conclusion

In summary, the coupling strength at the C-exciton in FL-MoS₂ is significantly larger than that found in a similar experiment at the A- and B-excitons, due to its large oscillator strength. Further experiments are needed to probe the carrier dynamics under C-exciton strong coupling, but there may be unique physics due to the nested-bands. Further, the large coupling may mean that the interesting ultrastrong effects such as ground state virtual photons may be found, which are predicted to exist even at room temperature with lossy bare excitons. [45]

Chapter 6

Conclusions

I have shown plasmonics where coupling between components in a system is minimized to model non-diffraction limited optics and maximized to change the very properties of the system. Although the fundamental property of high spatial confinement of light remains a constant, these examples show the wide range of applications possible by tuning the coupling strength of plasmons to their environment. Plasmonics may be used to guide light, enhance optical absorption or emission in other systems, detect minute changes in local refractive index for sensing, enhance or suppress chemical reactions, or alter the photonic and electronic states of a system to produce polariton condensates and entirely new chemistry.

Beyond fundamental scientific studies and demonstration of applications, plasmonics has seen commercial success in sensing applications in the biomedical and scientific industries. Applications such as photothermal cancer therapy will require extensive trials. For other envisaged plasmonics applications to succeed, it is likely that replacements to the common gold and silver will have to be found. They are lossy, expensive, and incompatible with current CMOS processing for on-chip integration.

Applications taking advantage of strong coupling for thresholdless lasing and

quantum information transport and processing are still far out and more research is required to fully understand the potential. However, the ability of plasmonics to concentrate light on nanometer length scales makes further studies of quantum interactions warranted, especially in the strong to ultrastrong coupling regimes, where research is still in its infancy. Chemistry involving plasmonics is still relatively unexplored beyond simple heating effects. There are many early studies implicating plasmonics in control of chemical reactions, but the mechanistics are not fully understood. Combining distinct research fields can often lead to unexpected discoveries; coupling plasmons to molecules may be such a case. Going one step further, one can ask whether is plasmonics could be integrated with biological systems to control cellular processes.

Bibliography

- [1] Wang, Y., Plummer, E. W. & Kempa, K. Foundations of Plasmonics. *Advances in Physics* **60**, 799–898 (2011).
- [2] Raether, H. Surface plasmons on smooth surfaces. In Höhler, G. (ed.) *Surface Plasmons on Smooth and Rough Surfaces and on Gratings*, Springer Tracts in Modern Physics 111, 4–39 (Springer, Berlin, 1988).
- [3] Dionne, J. A., Sweatlock, L. A., Atwater, H. A. & Polman, A. Planar metal plasmon waveguides: Frequency-dependent dispersion, propagation, localization, and loss beyond the free electron model. *Physical Review B* **72**, 075405 (2005).
- [4] Yang, H. U. *et al.* Optical dielectric function of silver. *Physical Review B* **91**, 235137 (2015).
- [5] Ritchie, R. H. Plasma Losses by Fast Electrons in Thin Films. *Physical Review* **106**, 874–881 (1957).
- [6] Johnson, P. B. & Christy, R. W. Optical Constants of the Noble Metals. *Physical Review B* **6**, 4370–4379 (1972).
- [7] Merlo, J. M. *et al.* Wireless communication system via nanoscale plasmonic antennas. *Scientific Reports* **6**, 31710 (2016).
- [8] Creel, E. B. *et al.* Directing Selectivity of Electrochemical Carbon Dioxide Reduction Using Plasmonics. *ACS Energy Letters* **4**, 1098–1105 (2019).
- [9] Powell, C. J. & Swan, J. B. Origin of the Characteristic Electron Energy Losses in Aluminum. *Physical Review* **115**, 869–875 (1959).
- [10] Tsuei, K.-D., Plummer, E. W., Liebsch, A., Kempa, K. & Bakshi, P. Multipole plasmon modes at a metal surface. *Physical Review Letters* **64**, 44–47 (1990).
- [11] Rossouw, D., Couillard, M., Vickery, J., Kumacheva, E. & Botton, G. A. Multipolar Plasmonic Resonances in Silver Nanowire Antennas Imaged with a Subnanometer Electron Probe. *Nano Letters* **11**, 1499–1504 (2011).
- [12] Knight, M. W. *et al.* Aluminum Plasmonic Nanoantennas. *Nano Letters* **12**, 6000–6004 (2012).

- [13] Rose, A. H. RTACalc. version 1.0 (2018). URL <https://github.com/rose3fa/RTACalc>. Transfer-matrix method code for MATLAB. (accessed June 1, 2019).
- [14] Brissinger, D., Salomon, L. & De Fornel, F. Unguided plasmon-mode resonance in optically excited thin film: Exact modal description of Kretschmann–Raether experiment. *Journal of the Optical Society of America B* **30**, 333–337 (2013).
- [15] Khitrova, G., Gibbs, H. M., Kira, M., Koch, S. W. & Scherer, A. Vacuum Rabi splitting in semiconductors. *Nature Physics* **2**, 81–90 (2006).
- [16] Törmä, P. & Barnes, W. L. Strong coupling between surface plasmon polaritons and emitters: A review. *Rep. Prog. Phys.* **78**, 013901 (2015).
- [17] Plankensteiner, D., Sommer, C., Reitz, M., Ritsch, H. & Genes, C. Enhanced collective Purcell effect of coupled quantum emitter systems. *Physical Review A* **99**, 043843 (2019).
- [18] Limonov, M. F., Rybin, M. V., Poddubny, A. N. & Kivshar, Y. S. Fano resonances in photonics. *Nature Photonics* **11**, 543–554 (2017).
- [19] Purcell, E. M. Spontaneous emission probabilities at radio frequencies. *Physical Review* **69**, 681 (1946).
- [20] Stuart, H. R. & Hall, D. G. Absorption enhancement in silicon-on-insulator waveguides using metal island films. *Applied Physics Letters* **69**, 2327–2329 (1996).
- [21] Rand, B. P., Peumans, P. & Forrest, S. R. Long-range absorption enhancement in organic tandem thin-film solar cells containing silver nanoclusters. *Journal of Applied Physics* **96**, 7519–7526 (2004).
- [22] Pillai, S., Catchpole, K. R., Trupke, T. & Green, M. A. Surface plasmon enhanced silicon solar cells. *Journal of Applied Physics* **101**, 093105 (2007).
- [23] Catchpole, K. R. & Polman, A. Design principles for particle plasmon enhanced solar cells. *Applied Physics Letters* **93**, 191113 (2008).
- [24] Saeta, P. N., Ferry, V. E., Pacifici, D., Munday, J. N. & Atwater, H. A. How much can guided modes enhance absorption in thin solar cells? *Optics Express* **17**, 20975 (2009).
- [25] Yu, Z., Raman, A. & Fan, S. Fundamental limit of nanophotonic light trapping in solar cells. *PNAS* **107**, 17491–17496 (2010).
- [26] Munday, J. N. & Atwater, H. A. Large Integrated Absorption Enhancement in Plasmonic Solar Cells by Combining Metallic Gratings and Antireflection Coatings. *Nano Letters* **11**, 2195–2201 (2011).

- [27] Callahan, D. M., Munday, J. N. & Atwater, H. A. Solar Cell Light Trapping beyond the Ray Optic Limit. *Nano Letters* **12**, 214–218 (2012).
- [28] Ye, F., Burns, M. J. & Naughton, M. J. Embedded metal nanopatterns for near-field scattering-enhanced optical absorption. *Physica Status Solidi (A)* **209**, 1829–1834 (2012).
- [29] Ganapati, V., Miller, O. D. & Yablonovitch, E. Light Trapping Textures Designed by Electromagnetic Optimization for Subwavelength Thick Solar Cells. *IEEE Journal of Photovoltaics* **4**, 175–182 (2014).
- [30] McCumber, D. E. Einstein Relations Connecting Broadband Emission and Absorption Spectra. *Physical Review* **136**, A954–A957 (1964).
- [31] Novotny, L. & Hecht, B. *Principles of Nano-Optics* (Cambridge University Press, New York, 2006).
- [32] Reithmaier, J. P. *et al.* Strong coupling in a single quantum dot–semiconductor microcavity system. *Nature* **432**, 197–200 (2004).
- [33] Yablonovitch, E. Inhibited Spontaneous Emission in Solid-State Physics and Electronics. *Physical Review Letters* **58**, 2059–2062 (1987).
- [34] Agranovich, V. & Malshukov, A. Surface polariton spectra if the resonance with the transition layer vibrations exist. *Optics Communications* **11**, 169–171 (1974).
- [35] Yokoyama, H. & Brorson, S. D. Rate equation analysis of microcavity lasers. *Journal of Applied Physics* **66**, 4801–4805 (1989).
- [36] Yokoyama, H. Physics and Device Applications of Optical Microcavities. *Science* **256**, 66–70 (1992).
- [37] Weisbuch, C., Nishioka, M., Ishikawa, A. & Arakawa, Y. Observation of the coupled exciton-photon mode splitting in a semiconductor quantum microcavity. *Physical Review Letters* **69**, 3314–3317 (1992).
- [38] Shore, K. A. & Ogura, M. Threshold characteristics of microcavity semiconductor lasers. *Optical and Quantum Electronics* **24**, S209–S213 (1992).
- [39] Rice, P. R. & Carmichael, H. J. Photon statistics of a cavity-QED laser: A comment on the laser–phase-transition analogy. *Physical Review A* **50**, 4318–4329 (1994).
- [40] Andreani, L. C. Exciton-polaritons in superlattices. *Physics Letters A* **192**, 99–109 (1994).
- [41] Houdré, R. *et al.* Measurement of Cavity-Polariton Dispersion Curve from Angle-Resolved Photoluminescence Experiments. *Physical Review Letters* **73**, 2043–2046 (1994).

- [42] Houdré, R., Stanley, R. P., Oesterle, U., Ilegems, M. & Weisbuch, C. Room-temperature cavity polaritons in a semiconductor microcavity. *Physical Review B* **49**, 16761–16764 (1994).
- [43] Panzarini, G. & Andreani, L. C. Double quantum well in a semiconductor microcavity: Three-oscillator model and ultrafast radiative decay. *Physical Review B* **52**, 10780–10783 (1995).
- [44] Kockum, A. F., Miranowicz, A., De Liberato, S., Savasta, S. & Nori, F. Ultrastrong coupling between light and matter. *Nature Reviews Physics* **1**, 19–40 (2019).
- [45] De Liberato, S. Virtual photons in the ground state of a dissipative system. *Nature Communications* **8**, 1465 (2017).
- [46] Yu, H. *et al.* Wafer-Scale Growth and Transfer of Highly-Oriented Monolayer MoS₂ Continuous Films. *ACS Nano* **11**, 12001–12007 (2017).
- [47] Xu, Z.-Q. *et al.* Synthesis and Transfer of Large-Area Monolayer WS₂ Crystals: Moving Toward the Recyclable Use of Sapphire Substrates. *ACS Nano* **9**, 6178–6187 (2015).
- [48] Pascoe, K. J. Reflectivity and Transmissivity through Layered, Lossy Media: A User-Friendly Approach. Tech. Rep. AFIT/EN-TR-01-07, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio School of Engineering (2001).
- [49] Rose, A. H. *et al.* Nanoscope based on nanowaveguides. *Optics Express* **22**, 5228–5233 (2014).
- [50] Abbe, E. A Contribution to the theory of the microscope, and the nature of microscopic vision. *Proceedings of the Bristol Naturalists' Society* (1874).
- [51] Born, M. & Wolf, E. (Pergamon Press, New York).
- [52] Pendry, J. Negative refraction makes a perfect lens. *Physical Review Letters* **85**, 3966–9 (2000).
- [53] Jacob, Z., Alekseyev, L. V. & Narimanov, E. Optical Hyperlens: Far-field imaging beyond the diffraction limit. *Optics Express* **14**, 8247–56 (2006).
- [54] Salandrino, A. & Engheta, N. Far-field subdiffraction optical microscopy using metamaterial crystals: Theory and simulations. *Physical Review B* **74**, 075103 (2006).
- [55] Liu, Z., Lee, H., Xiong, Y., Sun, C. & Zhang, X. Far-Field Optical Hyperlens Magnifying Sub-Diffraction-Limited Objects. *Science* **315**, 1686 (2007).
- [56] Smolyaninov, I., Hung, Y.-J. & Davis, C. Magnifying superlens in the visible frequency range. *Science* **315**, 1699–1701 (2007).

- [57] Silveirinha, M., Belov, P. & Simovski, C. Subwavelength imaging at infrared frequencies using an array of metallic nanorods. *Physical Review B* **75**, 035108 (2007).
- [58] Shvets, G., Trendafilov, S., Pendry, J. B. & Sarychev, A. Guiding, Focusing, and Sensing on the Subwavelength Scale Using Metallic Wire Arrays. *Physical Review Letters* **99**, 053903 (2007).
- [59] Zhao, Y. Investigation of image magnification properties of hyperlenses formed by a tapered array of metallic wires using a spatially dispersive finite-difference time-domain method in cylindrical coordinates. *Journal of Optics* **14**, 035102 (2012).
- [60] Kawata, S., Ono, A. & Verma, P. Subwavelength colour imaging with a metallic nanolens. *Nature Photonics* **2**, 438–442 (2008).
- [61] Kempa, K., Wang, X., Ren, Z. F. & Naughton, M. J. Discretely guided electromagnetic effective medium. *Applied Physics Letters* **92**, 043114 (2008).
- [62] Rybczynski, J. *et al.* Subwavelength waveguide for visible light. *Applied Physics Letters* **90**, 021104 (2007).
- [63] Rizal, B. *et al.*
- [64] Ditlbacher, H. *et al.* Silver Nanowires as Surface Plasmon Resonators. *Physical Review Letters* **95**, 257403 (2005).
- [65] Manjavacas, A. & de Abajo, F. G. Robust Plasmon Waveguides in Strongly Interacting Nanowire Arrays. *Nano Letters* **9**, 1285–1289 (2009).
- [66] Peng, Y. & Kempa, K. Controlling light propagation with nanowires. *Applied Physics Letters* **100**, 171903 (2012).
- [67] Kuttge, M., García de Abajo, F. J. & Polman, A. How grooves reflect and confine surface plasmon polaritons. *Optics Express* **17**, 10385 (2009).
- [68] Smith, C. L. C. *et al.* Plasmonic V-groove waveguides with Bragg grating filters via nanoimprint lithography. *Optics express* **20**, 5696–706 (2012).
- [69] Jin, J. *The Finite Element Method in Electromagnetics* (Wiley-IEEE Press, 2002), 2 edn.
- [70] Rose, A. H., Dunklin, J. R., Zhang, H., Merlo, J. M. & van de Lagemaat, J. Coherent superposition of valley-polarized excitons under strong exciton–plasmon coupling in 2D MoS₂ at room temperature. *In Submission* (2019).
- [71] Wang, Q. H., Kalantar-Zadeh, K., Kis, A., Coleman, J. N. & Strano, M. S. Electronics and optoelectronics of two-dimensional transition metal dichalcogenides. *Nature Nanotechnology* **7**, 699–712 (2012).

- [72] Lu, Q., Yu, Y., Ma, Q., Chen, B. & Zhang, H. 2D Transition-Metal-Dichalcogenide-Nanosheet-Based Composites for Photocatalytic and Electrocatalytic Hydrogen Evolution Reactions. *Advanced Materials* **28**, 1917–1933 (2016).
- [73] Berman, P. R. *Cavity Quantum Electrodynamics* (Academic Press, 1994).
- [74] Christopoulos, S. *et al.* Room-Temperature Polariton Lasing in Semiconductor Microcavities. *Physical Review Letters* **98**, 126405 (2007).
- [75] Byrnes, T., Kim, N. Y. & Yamamoto, Y. Exciton–polariton condensates. *Nature Physics* **10**, 803–813 (2014).
- [76] Cao, E., Lin, W., Sun, M., Liang, W. & Song, Y. Exciton-plasmon coupling interactions: From principle to applications. *Nanophotonics* **7**, 145–167 (2018).
- [77] Hennessy, K. *et al.* Quantum nature of a strongly coupled single quantum dot–cavity system. *Nature* **445**, 896–899 (2007).
- [78] Wiederrecht, G. P., Hall, J. E. & Bouhelier, A. Control of Molecular Energy Redistribution Pathways via Surface Plasmon Gating. *Physical Review Letters* **98**, 083001 (2007).
- [79] Hutchison, J. A., Schwartz, T., Genet, C., Devaux, E. & Ebbesen, T. W. Modifying Chemical Landscapes by Coupling to Vacuum Fields. *Angewandte Chemie International Edition* **51**, 1592–1596 (2012).
- [80] Azarova, N. *et al.* Coupling between a Molecular Charge-Transfer Exciton and Surface Plasmons in a Nanostructured Metal Grating. *The Journal of Physical Chemistry Letters* **4**, 2658–2663 (2013).
- [81] Pockrand, I. & Swalen, J. D. Anomalous dispersion of surface plasma oscillations. *J. Opt. Soc. Am.* **68**, 1147–1151 (1978).
- [82] Vasa, P. & Lienau, C. Strong Light–Matter Interaction in Quantum Emitter/Metal Hybrid Nanostructures. *ACS Photonics* **5**, 2–23 (2018).
- [83] Kittel, C. Chapter 11. In *Introduction to Solid State Physics* (John Wiley & Sons, Inc., New York, 1986), 6 edn.
- [84] Gurioli, M. *et al.* Well-width and aluminum-concentration dependence of the exciton binding energies in GaAs/Al_xGa_{1-x}As quantum wells. *Physical Review B* **47**, 15755–15762 (1993).
- [85] Schneider, C., Glazov, M. M., Korn, T., Höfling, S. & Urbaszek, B. Two-dimensional semiconductors in the regime of strong light-matter coupling. *Nature Communications* **9**, 2695 (2018).
- [86] Liu, X. *et al.* Strong light–matter coupling in two-dimensional atomic crystals. *Nature Photonics* **9**, 30–34 (2015).

- [87] Dufferwiel, S. *et al.* Exciton–polaritons in van der Waals heterostructures embedded in tunable microcavities. *Nature Communications* **6**, 8579 (2015).
- [88] Liu, W. *et al.* Strong Exciton–Plasmon Coupling in MoS₂ Coupled with Plasmonic Lattice. *Nano Letters* **16**, 1262–1269 (2016).
- [89] Liu, H.-L. *et al.* Optical properties of monolayer transition metal dichalcogenides probed by spectroscopic ellipsometry. *Applied Physics Letters* **105**, 201905 (2014).
- [90] Lynch, D. W. & Hunter, W. R. Introduction to the Data for Several Metals. In Palik, E. D. (ed.) *Handbook of Optical Constants of Solids*, vol. 3, 245–247 (Academic Press, Burlington, 1998).
- [91] Bennett, J. M., Stanford, J. L. & Ashley, E. J. Optical Constants of Silver Sulfide Tarnish Films. *JOSA* **60**, 224–232 (1970).
- [92] Lynch, D. W. & Hunter, W. R. Comments on the Optical Constants of Metals and an Introduction to the Data for Several Metals. In Palik, E. D. (ed.) *Handbook of Optical Constants of Solids*, vol. 1, 355–356 (Academic Press, Burlington, 1985).
- [93] Jung, G.-H., Yoo, S. & Park, Q.-H. Measuring the optical permittivity of two-dimensional materials without a priori knowledge of electronic transitions. *Nanophotonics* **8**, 263–270 (2019).
- [94] Balci, S. *et al.* Tuning surface plasmon-exciton coupling via thickness dependent plasmon damping. *Physical Review B* **86**, 235402 (2012).
- [95] Hsu, C. *et al.* Thickness-Dependent Refractive Index of 1L, 2L, and 3L MoS₂, MoSe₂, WS₂, and WSe₂. *Advanced Optical Materials* **7**, 1900239 (2019).
- [96] Pockrand, I., Brillante, A. & Möbius, D. Exciton–surface plasmon coupling: An experimental investigation. *The Journal of Chemical Physics* **77**, 6289–6295 (1982).
- [97] Bellessa, J., Bonnard, C., Plenet, J. C. & Mugnier, J. Strong Coupling between Surface Plasmons and Excitons in an Organic Semiconductor. *PHYSICAL REVIEW LETTERS* **93**, 036404 (2004).
- [98] Kretschmann, E. & Raether, H. Radiative Decay of Non Radiative Surface Plasmons Excited by Light. *Zeitschrift für Naturforschung A* **23a**, 2135–2136 (1968).
- [99] Symonds, C. *et al.* Particularities of surface plasmon–exciton strong coupling with large Rabi splitting. *New Journal of Physics* **10**, 065017 (2008).

- [100] Liu, W. *et al.* Understanding the Different Exciton–Plasmon Coupling Regimes in Two-Dimensional Semiconductors Coupled with Plasmonic Lattices: A Combined Experimental and Unified Equation of Motion Approach. *ACS Photonics* **5**, 192–204 (2018).
- [101] Vasilevskiy, M. I., Santiago-Pérez, D. G., Trallero-Giner, C., Peres, N. M. R. & Kavokin, A. Exciton polaritons in two-dimensional dichalcogenide layers placed in a planar microcavity: Tunable interaction between two Bose-Einstein condensates. *Physical Review B* **92**, 245435 (2015).
- [102] Hopfield, J. J. Theory of the Contribution of Excitons to the Complex Dielectric Constant of Crystals. *Physical Review* **112**, 1555–1567 (1958).
- [103] Holmes, R. J. & Forrest, S. R. Strong Exciton-Photon Coupling and Exciton Hybridization in a Thermally Evaporated Polycrystalline Film of an Organic Small Molecule. *Physical Review Letters* **93**, 186404 (2004).
- [104] Gómez, D. E., Vernon, K. C., Mulvaney, P. & Davis, T. J. Coherent superposition of exciton states in quantum dots induced by surface plasmons. *Applied Physics Letters* **96**, 073108 (2010).
- [105] Wang, Q. *et al.* Direct observation of strong light-exciton coupling in thin WS₂ flakes. *Optics Express* **24**, 7151–7157 (2016).
- [106] Cuadra, J. *et al.* Observation of Tunable Charged Exciton Polaritons in Hybrid Monolayer WS₂-Plasmonic Nanoantenna System. *Nano Letters* **18**, 1777–1785 (2018).
- [107] Li, Y. *et al.* Probing Symmetry Properties of Few-Layer MoS₂ and h-BN by Optical Second-Harmonic Generation. *Nano Letters* **13**, 3329–3333 (2013).
- [108] Mak, K. F., He, K., Shan, J. & Heinz, T. F. Control of valley polarization in monolayer MoS₂ by optical helicity. *Nature Nanotechnology* **7**, 494–498 (2012).
- [109] Chen, Y.-J., Cain, J. D., Stanev, T. K., Dravid, V. P. & Stern, N. P. Valley-polarized exciton–polaritons in a monolayer semiconductor. *Nature Photonics* **11**, 431–435 (2017).
- [110] Johnson, P. & Christy, R. Optical Constants of the Noble Metals. *Physical Review B* **6**, 4370 (1972).
- [111] McPeak, K. M. *et al.* Plasmonic Films Can Easily Be Better: Rules and Recipes. *ACS Photonics* **2**, 326–333 (2015).
- [112] Rose, A. H., Dunklin, J. R., Zhang, H. & van de Lagemaat, J. Coherent superposition of valley-polarized excitons under strong exciton–plasmon coupling in 2D MoS₂ at room temperature. *In Preparation* (2019).

- [113] Thomas, A. *et al.* Tilting a ground-state reactivity landscape by vibrational strong coupling. *Science* **363**, 615–619 (2019).
- [114] Carvalho, A., Ribeiro, R. M. & Castro Neto, A. H. Band nesting and the optical response of two-dimensional semiconducting transition metal dichalcogenides. *Physical Review B* **88**, 115205 (2013).
- [115] Wang, L. *et al.* Slow cooling and efficient extraction of C-exciton hot carriers in MoS₂ monolayer. *Nature Communications* **8**, 13906 (2017).
- [116] Würfel, P. Solar energy conversion with hot electrons from impact ionisation. *Solar Energy Materials and Solar Cells* **46**, 43–52 (1997).
- [117] Li, Y. *et al.* Slow Cooling of High-Energy C Excitons Is Limited by Intervalley-Transfer in Monolayer MoS₂. *Laser & Photonics Reviews* **13**, 1800270 (2019).
- [118] Kozawa, D. *et al.* Photocarrier relaxation pathway in two-dimensional semiconducting transition metal dichalcogenides. *Nature Communications* **5**, 1–7 (2014).
- [119] Shi, J. *et al.* Plasmonic Enhancement and Manipulation of Optical Nonlinearity in Monolayer Tungsten Disulfide. *Laser & Photonics Reviews* **12** (2018).
- [120] Wu, S. *et al.* Electrical tuning of valley magnetic moment through symmetry control in bilayer MoS₂. *Nature Physics* **9**, 149–153 (2013).
- [121] Chervy, T. *et al.* High-Efficiency Second-Harmonic Generation from Hybrid Light-Matter States. *Nano Letters* **16**, 7352–7356 (2016).
- [122] Drobnyh, E., Pachter, R. & Sukharev, M. Harmonic Generation by Metal Nanostructures Optically Coupled to Two-Dimensional Transition-Metal Dichalcogenide. *The Journal of Physical Chemistry C* **123**, 6898–6904 (2019).
- [123] Dintinger, J., Klein, S., Bustos, F., Barnes, W. L. & Ebbesen, T. W. Strong coupling between surface plasmon-polaritons and organic molecules in subwavelength hole arrays. *Physical Review B* **71**, 035424 (2005).
- [124] Bellessa, J. *et al.* Giant Rabi splitting between localized mixed plasmon-exciton states in a two-dimensional array of nanosize metallic disks in an organic semiconductor. *Physical Review B* **80** (2009).
- [125] Schwartz, T., Hutchison, J. A., Genet, C. & Ebbesen, T. W. Reversible Switching of Ultrastrong Light-Molecule Coupling. *Physical Review Letters* **106**, 196405 (2011).
- [126] Gambino, S. *et al.* Exploring Light–Matter Interaction Phenomena under Ultrastrong Coupling Regime. *ACS Photonics* **1**, 1042–1048 (2014).
- [127] Günter, G. *et al.* Sub-cycle switch-on of ultrastrong light–matter interaction. *Nature* **458**, 178–181 (2009).

- [128] Wang, S. *et al.* Coherent Coupling of WS₂ Monolayers with Metallic Photonic Nanostructures at Room Temperature. *Nano Letters* **16**, 4368–4374 (2016).
- [129] Wen, J. *et al.* Room-Temperature Strong Light–Matter Interaction with Active Control in Single Plasmonic Nanorod Coupled with Two-Dimensional Atomic Crystals. *Nano Letters* **17**, 4689–4697 (2017).
- [130] Kleemann, M.-E. *et al.* Strong-coupling of WSe₂ in ultra-compact plasmonic nanocavities at room temperature. *Nature Communications* **8**, 1296 (2017).
- [131] Naughton, M. J. *et al.* Efficient nanocoax-based solar cells. *physica status solidi (RRL) – Rapid Research Letters* **4**, 181–183 (2010).
- [132] Ye, F., Burns, M. J. & Naughton, M. J. Embedded metal nanopatterns as a general scheme for enhanced broadband light absorption. *physica status solidi (a)* **212**, 561–565 (2015).
- [133] Solomon, M. L. *Embedded Metal Nanopatterns in Thin Film Microcrystalline Silicon for Increased Absorption*. Undergraduate Thesis, Boston College, Chestnut Hill, MA (2015).
- [134] Kempa, K. Plasmonic protection of the hot-electron energy. *physica status solidi (RRL) – Rapid Research Letters* **7**, 465–468 (2013).
- [135] Kong, J. *et al.* Hot electron plasmon-protected solar cell. *Optics Express* **23**, A1087–A1095 (2015).

Appendix A

List of Publications

Rose, A. H., Dunklin, J. R., Zhang, H., and van de Lagemaat, J. Enhanced strong coupling of band-nested excitons in few-layer TMDCs. *In preparation* (2019).

Rose, A. H., Dunklin, J. R., Zhang, H., Merlo, J. M., and van de Lagemaat, J. Plasmon-mediated coherent superposition of discrete excitons under strong exciton-plasmon coupling in few-layer MoS₂ at room temperature. *In submission* (2019).

Dunklin, J. R., Rose, A. H., Miller, E. M., and van de Lagemaat, J. Plasmonic Hot Hole Transfer in Gold Nanoparticle-decorated Transition Metal Dichalcogenide Nanosheets. *Under review, ACS Photonics* (2019).

Xue, J., Wang, R., Chen, X., Yao, C., Wang, Z.-K., Weber, M., Rose, A. H., Nuryyeva, S., Yang, J., Zhu, J., Huang, T., Zhao, Y., Tan, S., Yan, Y., Zhu, K., Yang, Y. Bring to Light the Surface of Halide Perovskite Thin Film. *Under review, Nature Mat.* (2019).

Zhai, T., Wang, K., Zhang, F., Xiao, C., Rose, A. H., Zhu, K., Beard, M. C. Individual Electron and Hole Mobilities in Lead Halide Perovskites Revealed by Noncontact Methods. *Accepted for publication, ACS Energy Lett.* (2019). DOI: 10.1021/acsenerylett.9b02310.

Yang, C., Merlo, J. M., D’Imperio, L. A., Rose, A. H., Calm, Y. M., Han, B.,

Gao, J., Zhou, G., Burns, M. J., Kempa, K., and Naughton, M. J. All-solution-processed micro/nanowires with electroplate welding as transparent conducting electrodes. *Phys. Status Solidi RRL* **13**, 1900010 (2019).

Merlo, J. M., Nesbitt, N. T., Calm, Y. M., Rose, A. H., D'Imperio, L., Yang, C., Naughton, J. R., Burns, M. J., Kempa, K., and Naughton, M. J. Wireless communication system via nanoscale plasmonic antennas. *Sci. Rep.* **6**, 31710 (2016).

Nesbitt, N. T., Merlo, J. M., Rose, A. H., Calm, Y. M., Kempa, K., Burns, M. J., and Naughton, M. J. Aluminum nanowire arrays via directed assembly. *Nano Lett.* **15**, 7294 (2015).

Kong, J., Rose, A. H., Yang, C., Wu, X., Merlo, J. M., Burns, M. J., Naughton, M. J., and Kempa, K. Hot electron plasmon-protected solar cell. *Opt. Express* **23**, A1087 (2015).

Rose, A. H., Wirth, B. M., Hatem, R. E., Ahmed, A. P., Burns, M. J., Naughton, M. J., and Kempa, K. Nanoscope based on nanowaveguides. *Opt. Express* **22**, 5228 (2014).

Appendix B

Supplementary figures for Chapter 3

The integrity of the plasmonic modes along the nanowaveguides is an important consideration. The following figure gives an indication of the effect geometrical imperfections of a nanowire have on signal propagation. The excitation scheme of the simulation in Figure B-1 below is the same as in Chapter 3. To quantify the transmis-

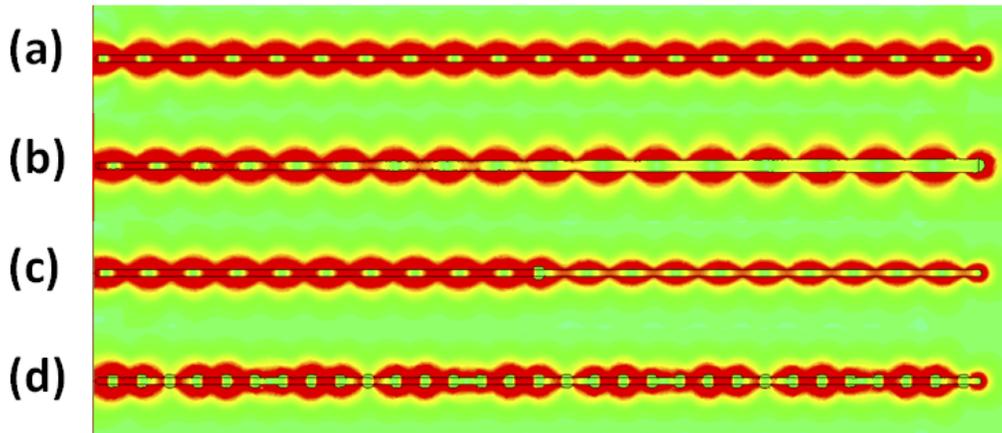


Figure B-1: Absolute value of the in-plane magnitude of the electric field for a silver nanowire $5 \mu\text{m}$ in length and 40 nm in diameter for (a) a smooth wire (control case), (b) a 40 nm diameter wire tapered to 80 nm at the distal end, (c) a 40 nm diameter wire with an 80 nm diameter silver nanosphere embedded halfway along the length of the wire, and (d) a 40 nm diameter wire with a series of 80 nm diameter silver spheres embedded along its full length, spaced 160 nm center-to-center. The excitation vacuum wavelength is 800 nm . The medium surrounding the wires is vacuum.

sion of the wires in Figure B-1, we integrated the maximum electric energy density around their input and distal ends and divided the distal value by the input value. This method allows us to compare the inherent transmission properties of each wire, independent of their ability to couple to free space. The tapered wire, Figure B-1b, achieved 70 % of the transmission of the control smooth wire in Figure B-1a. The straight wire with the large bump halfway along its length, Figure B-1c, had 50 % the performance of the control. Interestingly, a wire with many such bumps, Figure B-1d, had 80 % the performance of the control. The feature size of the defects in Figure B-1 is equal to the radius of the control wire—the tapered wire doubles its radius at the widest end and the bumps protrude from the surface by a height equal to the radius. We repeated the above simulations with a smaller feature size (half the wire radius), so the wire tapers from 40 nm to 60 nm and the bumps are 60 nm in diameter (with the same spacing), i.e. they protrude by 10 nm. In this case, the tapered wire achieves 95 %, the single bump wire 90 %, and the multi-bump wire 100 % of the transmission performance of the control. Thus, our simulations show that significant perturbations of the diameter of the wire are allowable while still maintaining reasonably high transmission of the signal. Impedance mismatch at the input and output of the waveguide is a more dominant factor than attenuation along the wire when considering transmission of the signal.

One question one may pose is: How accurately do the diameters of the two nanowires need to be equal to each other, for example for achieving 90 % of optimal performance? In short, the diameters of the two wires do not have to be closely controlled to maintain high resolution of the two-wire NFOM. In Figure B-2, the upper wire is 40 nm in diameter as in the manuscript, while the lower wire diameter is increased to 60 nm. Due to time constraints, fewer datum points are plotted and the simulation mesh quality has been reduced from that of the figure in Chapter 3. The FWHM of the lower wire has increased to 27 nm compared to 25 nm for the

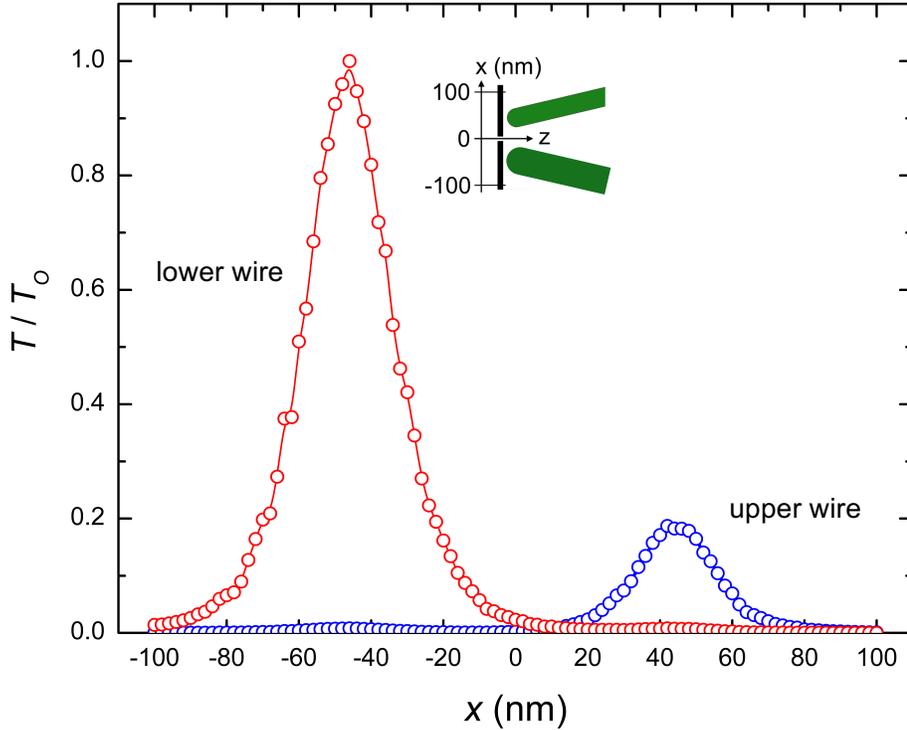


Figure B-2: Calculated normalized transmitted electric field intensity along converging 40 nm (upper) and 60 nm (lower) diameter wires as the excitation aperture is moved along the x-axis, from halfway between both wires ($x=0$) to above and below both wires ($x=\pm 100$ nm). This intensity is found by averaging the normalized magnitude of the time-averaged Poynting vector in a cylinder 160 nm in radius surrounding the distal half of the wires. The inset shows the geometry.

thinner wire, thus maintaining more than 90 % resolution. The increase in the signal of the lower wire can be attributed to the geometrical dependence of Fabry-Perot-type plasmonic modes as well as impedance variations. The variation of transmission properties of the wires can be accounted for with proper calibration of a device comprised of such wires.

The plasmonic modes along metallic waveguides are dependent on both the frequency of excitation and the geometry of the waveguides, and there is not a monotonic relationship between mode strength and excitation frequency. Figure B-3 shows that resonance for our 40 nm diameter, 5 μm length silver nanowires is stronger when

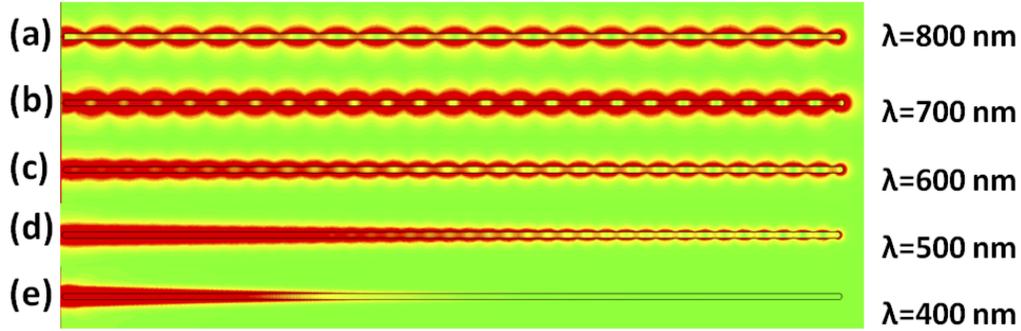


Figure B-3: Absolute value of the in-plane magnitude of the electric field for a silver nanowire $5 \mu\text{m}$ in length and 40 nm in diameter excited by light of vacuum wavelength (a) $\lambda = 800 \text{ nm}$, (b) 700 nm , (c) 600 nm , (d) 500 nm , and (e) 400 nm . Note that in this figure, the maximum values of the field of any phase are plotted. In some cases, this helps when comparing various plots. In other cases (i.e. plots in Chapter 3), it is unnecessary to plot the maximum values and plotting instead at a constant phase value helps show the shape of the modes.

excited by light with vacuum wavelength of 700 nm than by either 600 nm or 800 nm . Below 600 nm , there is clear attenuation. Any wavelength in the $600\text{-}800 \text{ nm}$ range, however, will provide both reasonable mode strength and attenuation.

Geometry has an effect on the modes excited on the nanowires. Figure B-4 shows the maximum electric field values for wires of various diameters. The number of

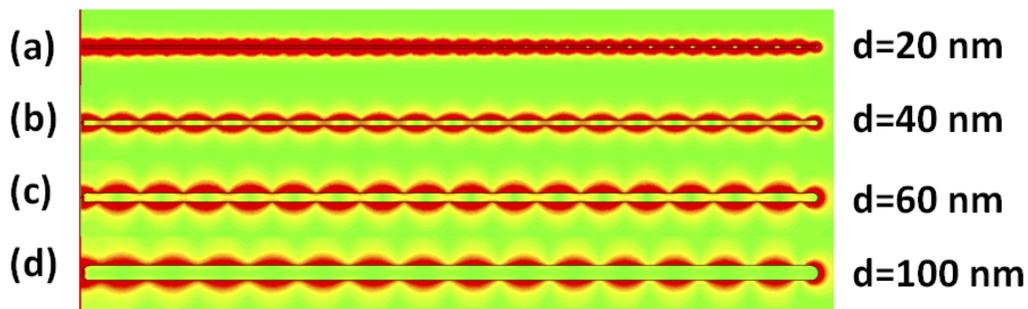


Figure B-4: Maximum value of the absolute value of the in-plane magnitude of the electric field for a silver nanowire with length of $5 \mu\text{m}$ and diameter of (a) 20 nm , (b) 40 nm , (c) 60 nm , and (d) 100 nm . The excitation vacuum wavelength is 800 nm .

plasmonic nodes along a metallic nanowire for a given excitation wavelength is a

function of both the length and diameter of the wire. As Figure B-4 shows, wider wires support fewer nodes. The strength of the modes is probably most closely tied to proximity to the resonance condition for each wire. This so-called etalon effect causes slight deviations in the mode strength, but does not allow for a complete anti-resonant condition. As a measure of overall transmission, the maximum electric energy density was integrated at the distal ends of wires from 20 nm to 200 nm diameter. The general trend was for increasing transmission with increasing diameter, with small local maxima and minima indicative of the etalon effect. Therefore, high signal transmission can be achieved along wires of varying diameter, with coupling into and out of the wires playing the dominant role in signal transmission. We added the following sentence to page 4 of the manuscript to address this comment: “Similar behavior is found for nanowires from 20 nm to 200 nm diameter generally increasing in transmission with increasing diameter.”

Silver has the lowest loss of common plasmonic metals. In Figure B-5, the use of Ag compared to Au for the nanowires is shown.

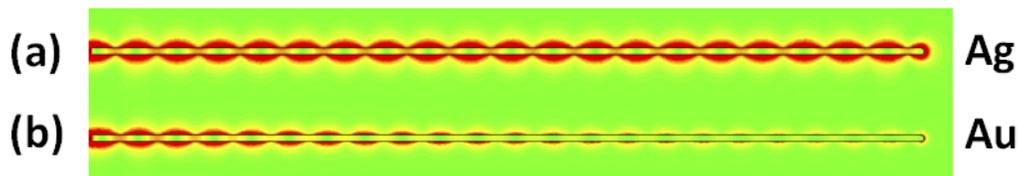


Figure B-5: Absolute value of the in-plane magnitude of the electric field for a (a) silver and (b) gold nanowire 5 μm in length and 40 nm in diameter. The excitation vacuum wavelength is 800 nm.

Figure B-5 clearly illustrates the increased attenuation in a gold nanowire.

Based on the above discussion, we conclude that metallic nanowires act as robust waveguides of surface plasmons, making them good candidates as building blocks of a near field optical magnifier in the search for an optical nanoscope.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix C

Supplementary strong coupling data

C.1 Reproducibility in MoS₂–Ag

The strong coupling effects shown in Chapters 4 and 5 was reproduced in various samples. Below are images showing the dispersion from these experiments.

First, Figure C-1 shows strong exciton–plasmon coupling at the C-exciton. The dispersion data are from a distinct region of the same sample used for Chapter 5. Strong coupling of similar magnitude to that shown in Chapter 5 is evident.

Figure C-2 shows the dispersion from a sample similar to that of Chapter 4, but made from a separate CVD run of MoS₂, using its own prism and another Ag deposition.

Figure C-3 shows the dispersion from a sample produced from a soda lime glass prism, 40 nm Ag, and chemically exfoliated MoS₂ that was airbrushed (sprayed) onto the Ag. The chemically exfoliated MoS₂ is flakes that are order 100 nm in diameter and mostly few-layer in thickness. They were diluted in ethanol and airbrushed onto the prism, heated to ~80°C, then soaked for at least an hour in de-ionized water, before drying with a nitrogen gun, and repeating the process. Five coats were applied. This work shows that wet chemistry methods may also be applied to producing films for

strong coupling experiments.

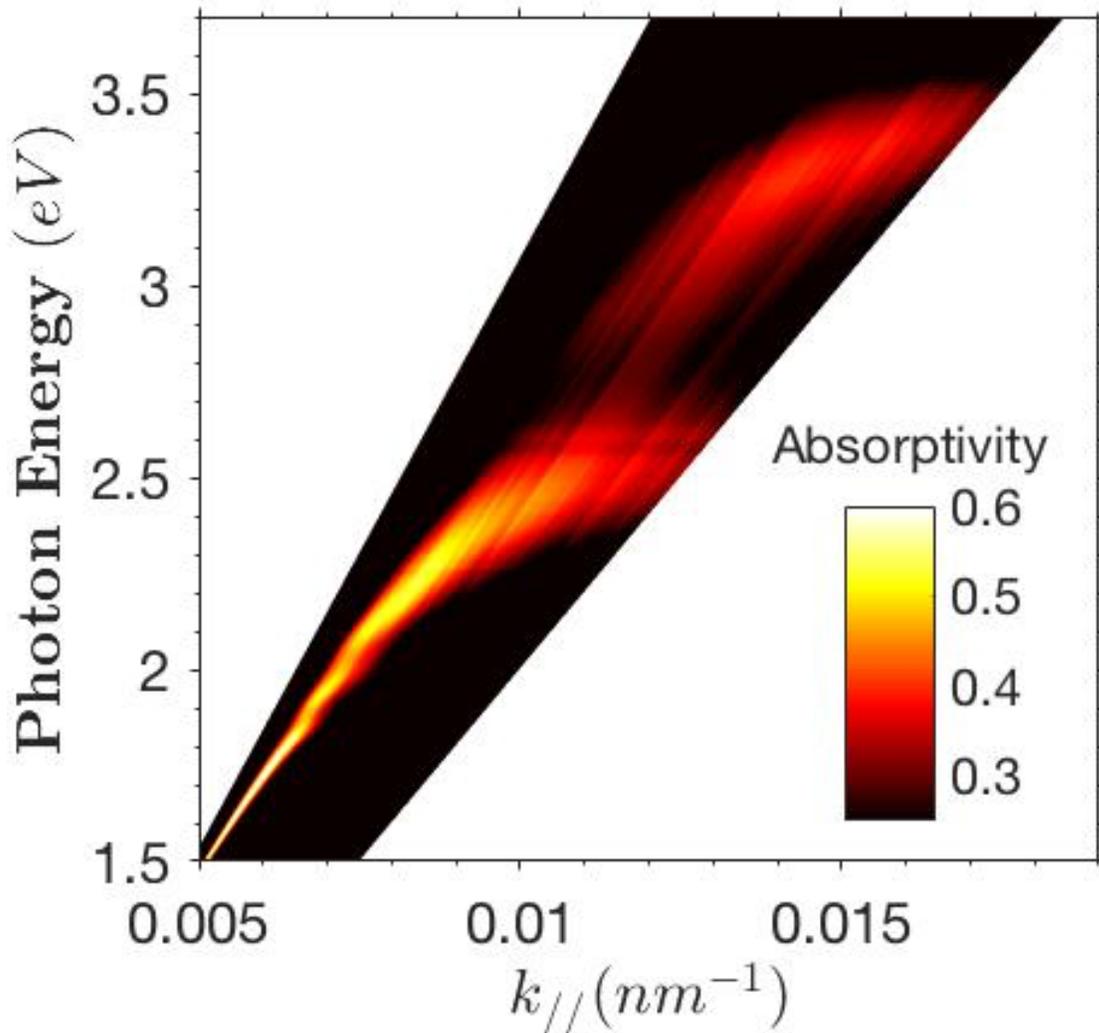


Figure C-1: Strong exciton–plasmon coupling at the C-exciton. The dispersion data taken for this figure are from a distinct region of the same sample used for Chapter 5. Strong coupling of similar magnitude to that shown in Chapter 5 is evident.

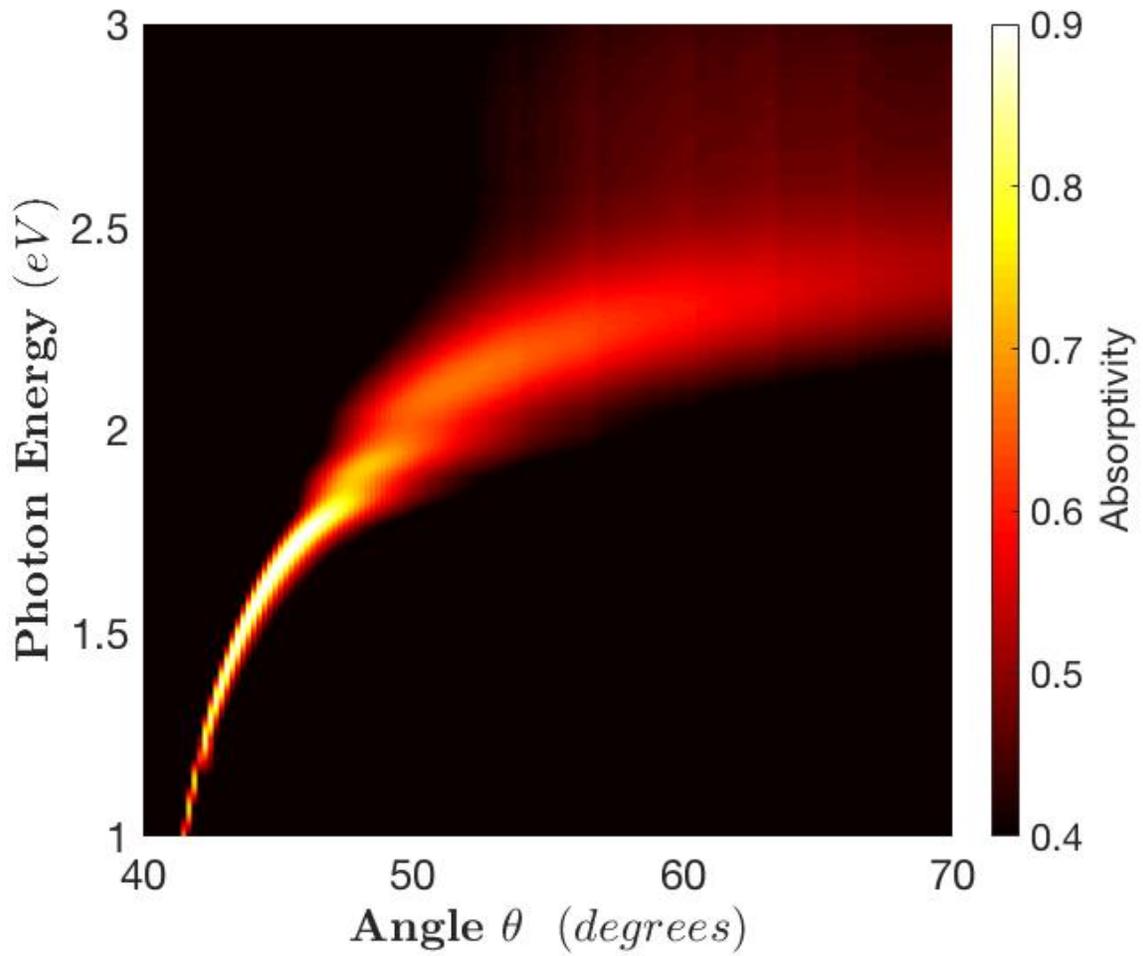


Figure C-2: Strong exciton–plasmon coupling at the A- and B-excitons. The data are taken from a sample similar, but distinct, to that of Chapter 4. Note that the x -axis is the angle of incidence in degrees, rather than wavevector.

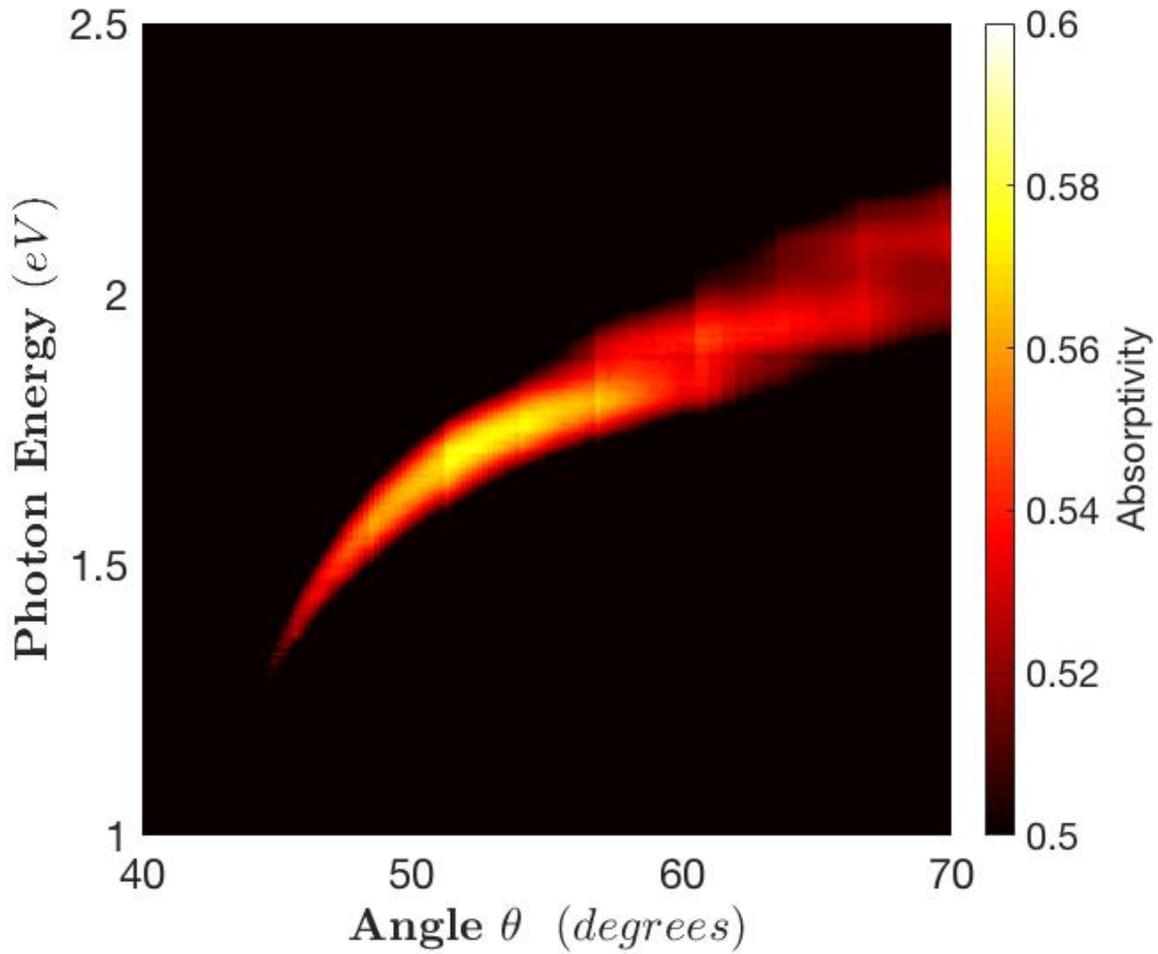


Figure C-3: Strong exciton–plasmon coupling at the A- and B-excitons. The data are taken from a sample produced from a soda lime glass prism, 40 nm Ag, and chemically exfoliated MoS₂ that was airbrushed (sprayed) onto the Ag. Note that the x -axis is the angle of incidence in degrees, rather than wavevector.

C.2 Strong coupling in MoS₂–Au

Finally, as Figure 4-10 in Chapter 4 shows, it is expected that strong coupling features should be visible when the Ag is exchanged with Au. Figure C-4 shows ~ 10 nm MoS₂ grown via CVD, transferred a single time to 1 nm Ti / 40 nm Au on a glass prism (high index glass, SF11, with refractive index 1.8). Strong coupling features at 1.8 and 2.0 eV are evident, although the splitting is not strong and could potentially be further optimized. Also evident is the merging of the plasmon mode with a continuum of states at 2.3-2.4 eV, due to D-band transitions in gold that turn on in this region and higher energies.

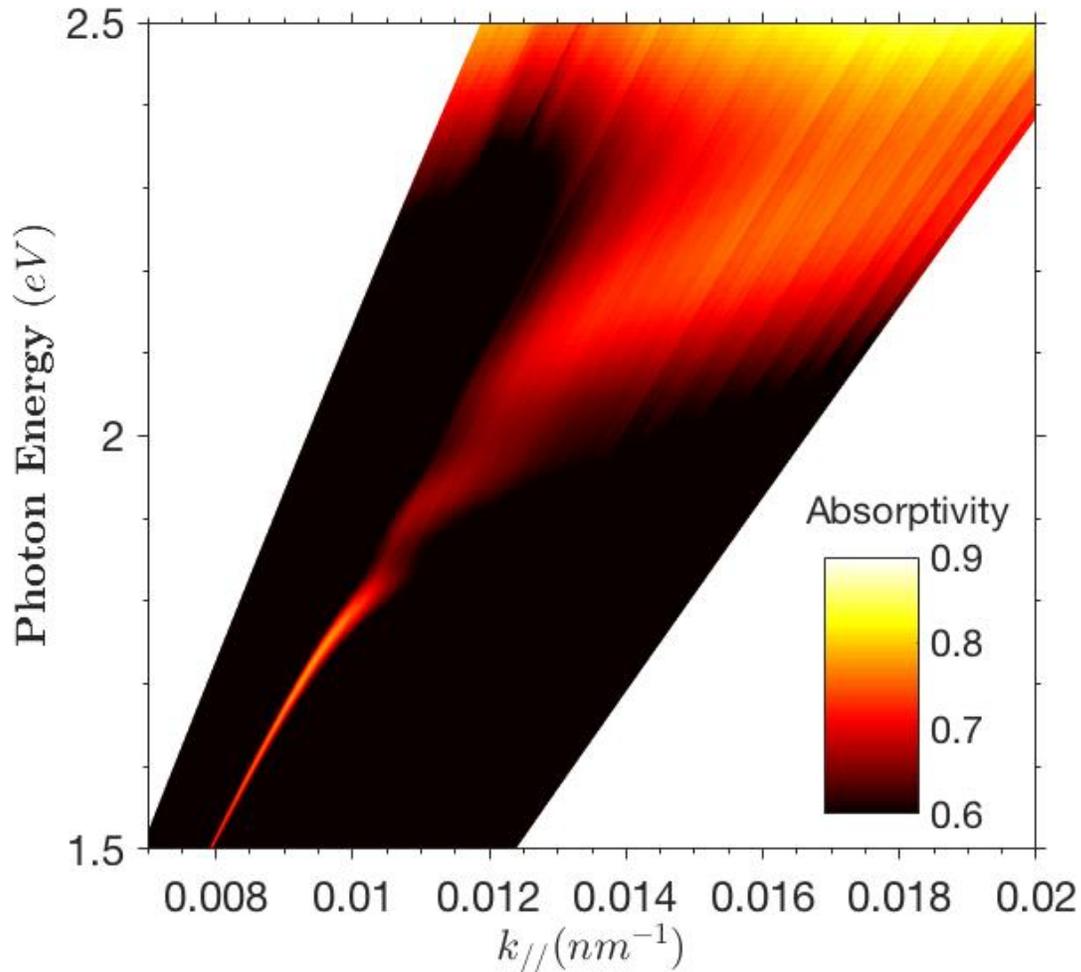


Figure C-4:

C.3 Strong coupling in aqueous electrolyte

As discussed in Chapters 4 and 5, strong coupling can have implications for chemical reactions. The band-splitting that occurs optically, also changes the electronic properties of the system, and therefore the chemical properties as well. The Kretschmann-Raether configuration provides a platform for studying photoelectrochemical properties of a system under strong coupling interface with chemical species in electrolyte. One of the first challenges in performing such experiments is redesigning them for the dielectric environment of the liquid electrolytes. Up until now, all work presented was performed or simulated in air. However, just as the introduction of a dielectric on top of a plasmonic film changes the resonant behavior and dispersion of the plasmon, placing the film in electrolyte also changes the dispersion of the plasmon. This behavior is shown qualitatively in the panels of Figure C-5, which simulate a similar sample to that described in Chapter 4, but with different media on top of the MoS₂ (rows, air or water) and different types of prism glass with varying refractive index (columns, refractive index 1.5 (conventional glass) or 2 (high index glass)). Figure C-5a shows simulated strong coupling (via transfer matrix method) at the A- and B-excitons in MoS₂ on Ag on a conventional glass (soda lime glass, fused silica, or BK-7, for example) prism, with air above the MoS₂. In panel b, the behavior changes slightly when the prism glass is changed for high index glass (SLAH-79, refractive index of 2.0), with the modes "squeezing" to lower angles. When the conventional glass sample is placed with water coating the MoS₂, the modes shift toward higher angles of incident, and much of the dispersion is not accessible given the limits of the k -value achievable in the glass. Thus, as we saw from panel b, we may utilize a high index glass substrate to shift the modes back toward lower angles. Indeed, in panel d, this appears to move the modes back to a region in k -space accessible within experimentally possible conditions.

Repeating the same experiments as in Chapter 4, but with water placed on top

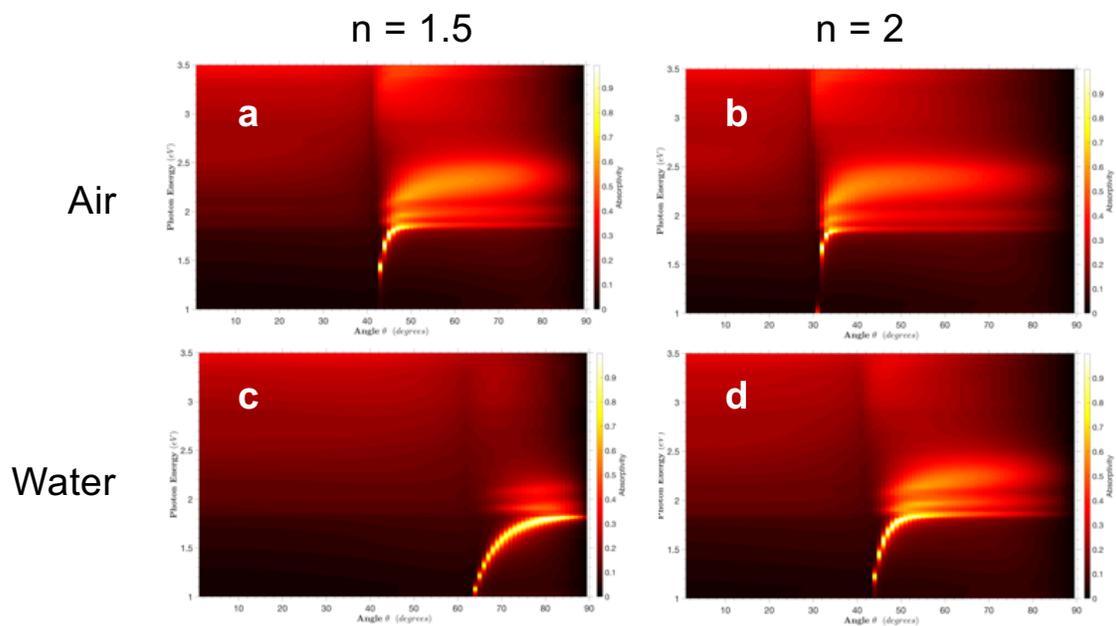


Figure C-5: Transfer matrix model simulations of strongly coupled MoS₂-Ag on different substrates and in different media. Panels a) and c) show the dispersion using substrate with refractive index of 1.5, in air and water, respectively. Panels b) and d) show the dispersion using substrate with refractive index of 2.0, in air and water, respectively. Use of the high index substrate is necessary when performing experiments in aqueous media. Note that the dispersion is shown with angle in degrees on the x -axis.

of the MoS₂ indeed revealed no spectral features. Therefore, the samples were reproduced on high index glass (SF11, refractive index of 1.8). A new CVD run of MoS₂ produced ~ 10 nm thick material, that was transferred a single time onto the typical 1 nm Ti / 40 nm Ag. Figure C-6 shows the dispersion of this sample.

Figure C-7 shows the dispersion of the same sample with the MoS₂ submersed in neutral (PH = 7) buffered phosphate solution. The behavior expected from the simulations is seen—the modes are shifted to higher k values, and are almost not accessible with the current setup. Even higher index substrates would help move the modes back to lower k .

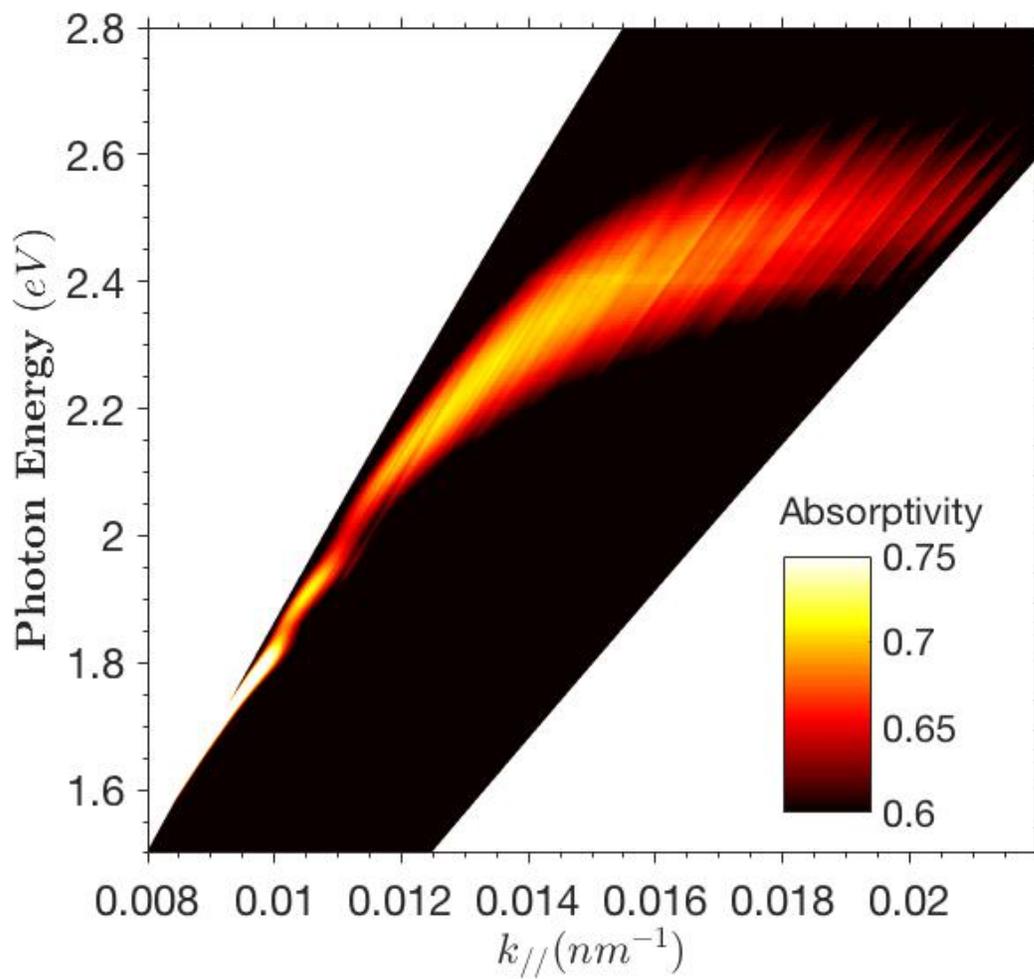


Figure C-6: Strong coupling at A- and B-excitons in ~ 10 nm MoS₂ on 40 nm Ag on high index glass (SF11, refractive index of 1.8).

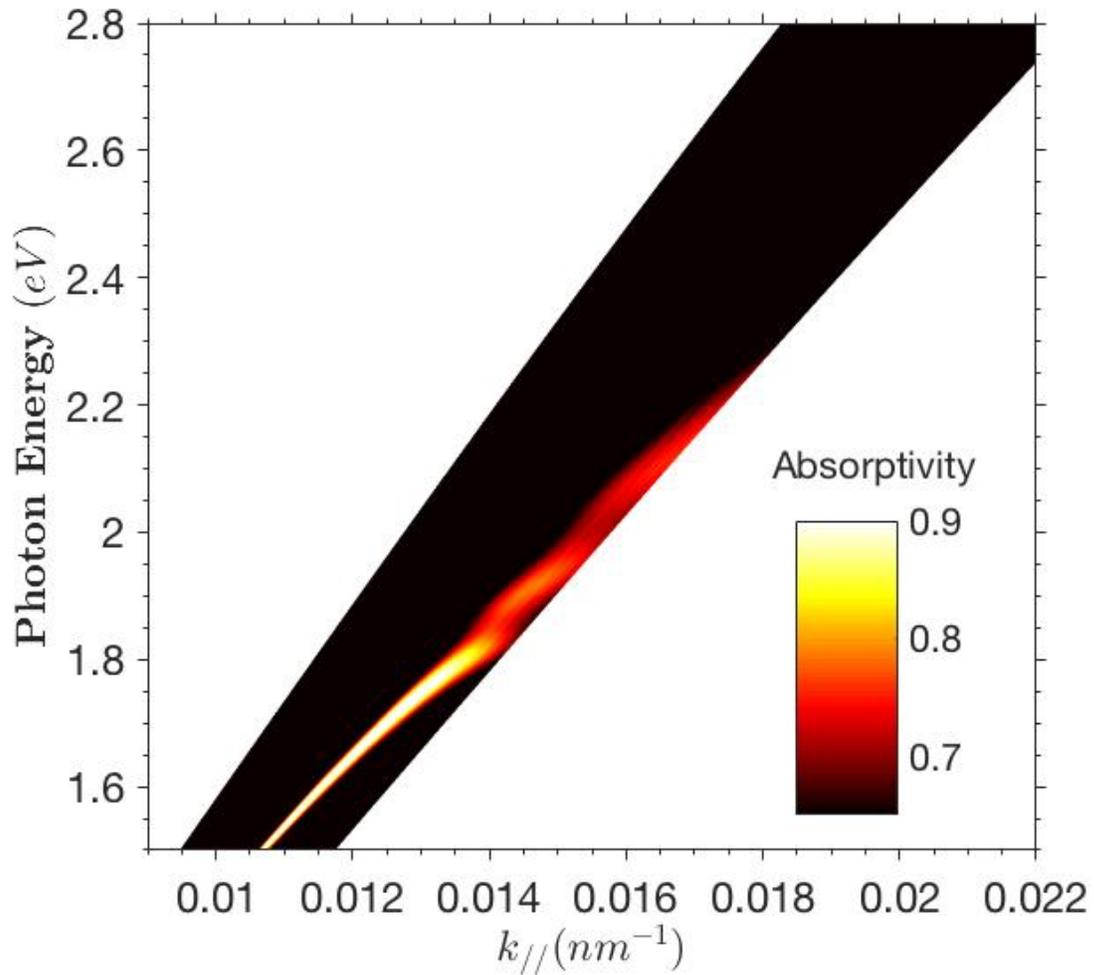


Figure C-7: Strong coupling at A- and B-excitons in ~ 10 nm MoS₂ on 40 nm Ag on high index glass (SF11, refractive index of 1.8), with the MoS₂ side submersed in aqueous buffered phosphate solution. The modes are shifted to higher k .

Appendix D

Embedded Metal Nanopatterns for Light Management in Ultrathin Solar Absorbers

D.1 Background

As mentioned in Chapter 1, in the weak coupling regime, plasmonics can enhance the local optical density of states via its confinement of light at interfaces. Thus, a plasmonic nanostructure near or embedded within a semiconducting absorber, can increase the optical density of states in the absorber, thereby enhancing absorption. An alternate view is to consider that the electric field strength of light is enhanced when it is coupled into a plasmon. Intensity is proportional to the square of the electric field, and absorption is linear with intensity, at low fluence, therefore the absorption is enhanced near the plasmon. This enhancement mechanism may be used to reduce the thickness of solar absorber materials. Thus, less material may be used, which reduces the cost, along with reduced carrier recombination described below.

In crystalline semiconductors used for solar cells, silicon and gallium arsenide, for example, the free carrier path length tends to be longer than the thickness of cell needed to absorb approximately 100 % of incident photons with energy greater than the band gap. In thin film solar materials, which utilize amorphous semiconductors, often the opposite is true. This leads to what is known as the "thick-thin problem:" when the cell is thick enough to absorb all incident light, it is too thick to collect all photo-generated carriers and when the cell is thin enough to collect all carriers, it is too thin to absorb all light. Thus, schemes are desired to enhance either the free carrier path length or reduce the optical absorption path length.

Considering the optical path, engineering absorption in a solar absorber is known as "light management." Forms of light management include anti-reflection coatings, back reflectors, surface texturing, and plasmonic absorption enhancement. [20–29] Another idea is to orthogonalize the problem by making the solar material thick in the direction of light propagation and thin orthogonal to that, in the form of nanopillars or nanocoax. [131] Considering the plasmonic enhancement method, the plasmonic structure may be placed on the top or bottom of the absorber, or embedded within it [28, 132]. Embedding a plasmonic nanostructure maximizes its contact area with the solar absorber, maximizing the absorption enhancement.

Following on the work by Fan Ye [28, 132], I performed follow-up experiments and modeling of embedded plasmonic nanopatterns (termed embedded metal nanopatterns (EMN)) in amorphous silicon (a-Si) and cadmium telluride (CdTe) ultrathin-film solar absorbers. I was aided by undergraduate researchers Laura Simko and Michelle L. Solomon. Solomon presented the results of much of the work in her undergraduate thesis. [133]

The general scheme is shown in Figure D-1. The substrates are glass coated with a transparent conductive oxide (TCO) thin film. Two solar absorbers were studied: aSi and CdTe. The TCO for the aSi experiments was indium tin oxide (ITO). Fluorine-

doped tin oxide (FTO) was used for the CdTe experiments. The general procedure to fabricate the devices is as follows (see Solomon's thesis for further details [133]). The TCO glass was purchased from a commercial supplier and cleaned. A thin film of aSi or CdTe is deposited on the TCO. A metal nanopattern of Ag is formed next, using nanosphere lithography. The nanopattern is embedded by depositing another layer of absorber material. Finally, an opaque Ag back reflector is deposited. The film thickness will be given in sections for each absorber material.

All work was performed at the Integrated Sciences Cleanroom and Nanofabrication Facility at Boston College.

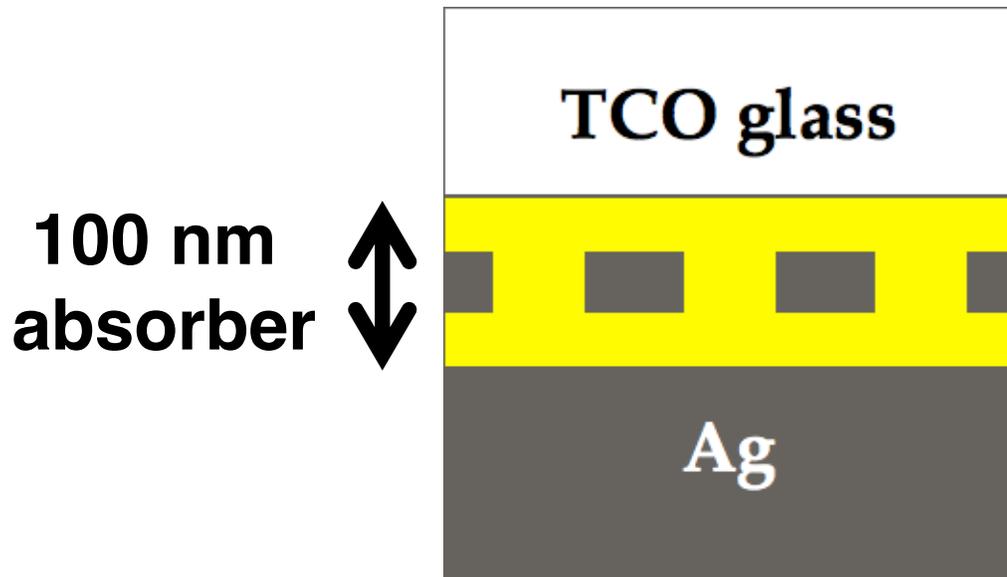


Figure D-1: Cross section representation of embedding metal nanopatterns (EMN) in solar absorber material. The substrates are glass coated with a transparent conductive oxide (TCO) thin film. A thin film of solar absorber is deposited on the TCO. A metal nanopattern of Ag is formed next, using nanosphere lithography. The nanopattern is embedded by depositing another layer of absorber material. Finally, an opaque Ag back reflector is deposited.

D.2 Amorphous silicon

To fabricate the aSi samples, 20 nm of aSi was deposited by plasma-enhanced chemical vapor deposition (PECVD). NSL was performed using 500 nm polystyrene spheres, etched slightly in an oxygen reactive ion etch (RIE), to yield an array of holes in a Ag thin film. The Ag was 20 nm thick, deposited by electron beam evaporation. Finally, 80 nm of aSi was deposited by PECVD, followed by 250 nm of Ag as a back reflector.

Figure D-2 shows the topography of a control device with no metal nanopattern measured by atomic force microscopy (AFM) before deposition of back reflector. Grain size of ~ 100 nm is evident.

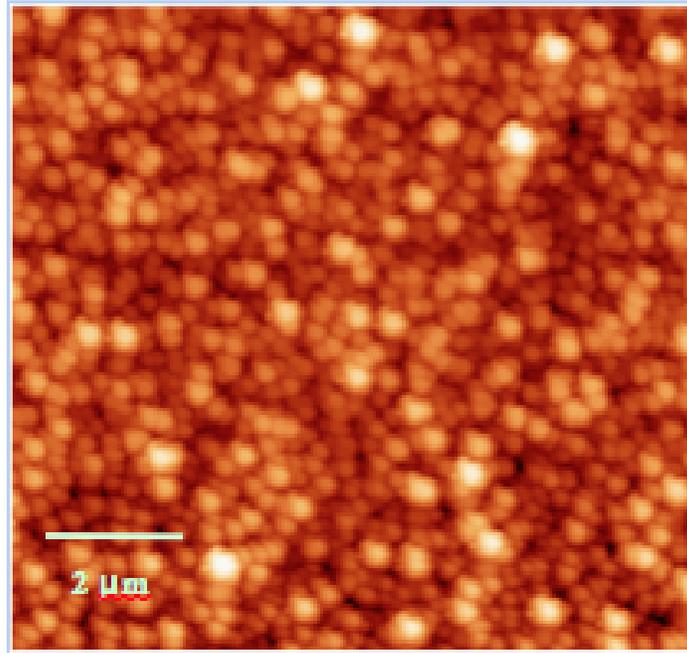


Figure D-2: Topography of aSi control film consisting of device without metal nanopattern, measured by atomic force microscopy (AFM) before deposition of back reflector.

Figure D-3 shows an image of a similar control device measured by scanning electron microscopy (SEM). Grain size of ~ 100 nm is again evident.

The grain structure of the aSi suggested non-amorphous material. In e-mail com-

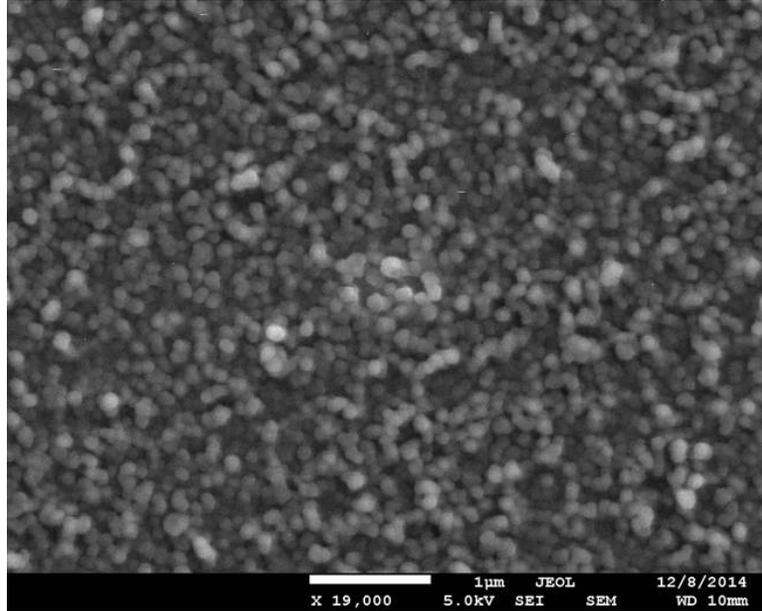


Figure D-3: Image of aSi control film consisting of device without metal nanopattern, measured by scanning electron microscopy (SEM) before deposition of back reflector.

munication with Franz-Josef Haug of L'Ecole Polytechnique Fédérale de Lausanne, Franz suggested the following: "The granular structure is a little strange but not unheard of. Try to get the book "Physics of hydrogenated amorphous silicon Part I" by Joannopoulos and Lucovsky. On page 23 they show rather similar morphology for He-diluted films." Our aSi recipe utilized 4% silane in helium, with no additional hydrogen. Most aSi solar work is done with added hydrogen during PECVD. Therefore, we concluded that the "quality" of our films was "low" and we doubted the usefulness to the community of publishing data without controlling for film quality. Thus, this work was never published. However, there are still some interesting results, shown below.

Figure D-4 shows a SEM of an aSi device after nanopatterning, before second aSi deposition. The hexagonal closed packed array of holes in Ag is clearly visible.

Figure ?? shows absorbance data for a set of experimental devices. Here, we define absorbance, A , equivalently to absorptance or absorptivity, as $A = 1 - R - T$, where R

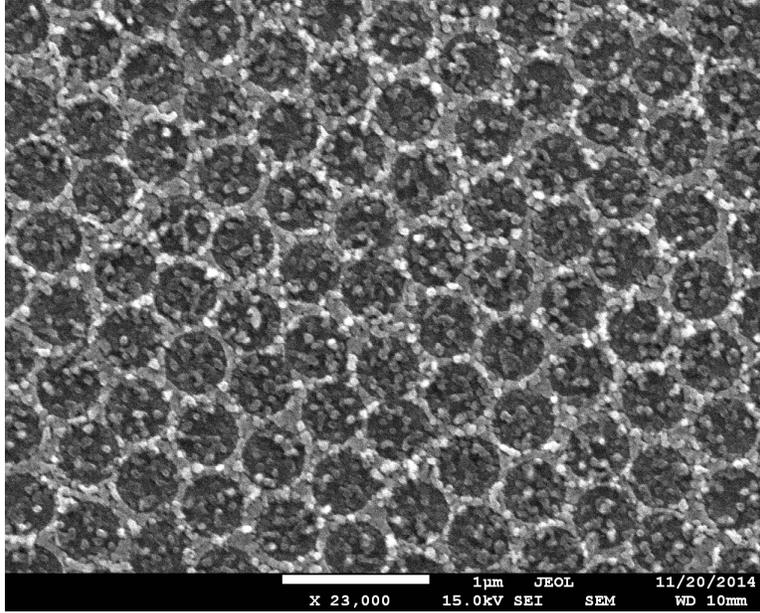


Figure D-4: SEM image of aSi device after nanopatterning, before second deposition of aSi. The array of holes in Ag is clearly visible, with 500 nm pitch. The metal is 20 nm thick.

is reflectance and T is transmittance. The brown and red lines shows two nominally identical devices with EMN. The orange line shows a control with embedded metal layer (EML), where a metal thin film is deposited during the same run as the EMN devices, but no nanopatterning is performed. The purple line shows a "thin" control with no embedded metal and same overall thickness of aSi. The blue line shows a device with 500 nm aSi, which is comparable to a conventional thin film aSi solar cell, a "thick" control.

There are a couple of results to take away from Figure ???. The EMN devices show an 11 % integrated absorption enhancement over the thin control from 300 nm to 689 nm. It shows a 3 % enhancement over the thick control. This is intriguing as it shows approximately the same absorption is achievable in using 80 % less material and thickness, by incorporating the nanopattern. The reduction in thickness should also reduce carrier recombination, leading to further improvement in a complete photovoltaic solar cell.

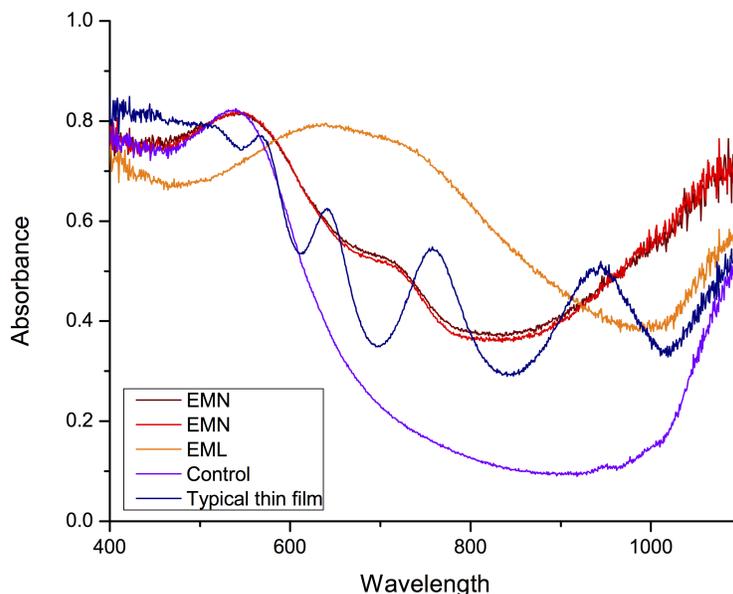


Figure D-5: Absorbance (defined as $1 - \text{transmittance} - \text{reflectance}$) of aSi devices. The brown and red lines shows two nominally identical devices with EMN. The orange line shows a control with embedded metal layer (EML), where a metal thin film is deposited during the same run as the EMN devices, but no nanopatterning is performed. The purple line shows a control with no embedded metal and same overall thickness of aSi. The blue line shows a device with 500 nm aSi, which is comparable to a conventional thin film aSi solar cell.

The control with 20 nm unpatterned Ag sandwiched between the aSi layers, shows surprisingly high absorption. Solomon and I tested the hypothesis that the absorption enhancement in this device was due to thin film interference effects by performing transfer matrix method (TMM) simulations. We took spectroscopic ellipsometry data of control or witness samples of all materials in the devices and I analyzed the data to obtain the complex refractive indices of each material. Solomon implemented the TMM. We did not find any evidence that in perfect thin film stacks of aSi and Ag, large absorption in the visible would be expected. See [133] for full details. Another hypothesis is coating the rough aSi grains with Ag created localized surface plasmon resonances (LSPR). Such features would not be captured by the TMM, but would lead to large absorption. Finite element method numerical electromagnetic

simulations based on the actual topography of the aSi thin films could be run to test this hypothesis.

D.3 Cadmium telluride

The CdTe devices were fabricated similarly to the aSi with the following exceptions. The first deposition of CdTe was 50 nm, done by RF magnetron sputtering at the University of Toledo in Alvin Compaan's lab. As mentioned, the substrate was commercial FTO-coated glass. The Ag pattern was 30 nm thick. The second CdTe deposition was identical to the first, 50 nm thick.

Figure D-6 shows a SEM of the CdTe EMN device. Similar to aSi, ~100 nm grains were observed. This observation again brings into question the quality of the semiconducting thin film and the usefulness of the results of this study to the community. Indeed, a CdTe researcher at the National Renewable Energy Laboratory remarked on the poor quality of the films when seeing this work presented at a poster session.

Figure ?? shows the absorbance (same definition as above) of 100 nm of CdTe with EMN compared to a control with no EMN. The x-axis in figure ?? is plotted up to the band gap of CdTe. Total absorbance of the EMN film is 93 % and control film is 86 %, leading to a 8 % (rounding error) absorption enhancement with EMN. At the band edge, the enhancement is 16 %. Conventional CdTe thin film solar cells use 2,000 nm thick CdTe layers, so this represents high absorption using 20 times less material.

Figure ?? shows the absorbance of 150 nm of CdTe with EMN compared to a control with no EMN. Total absorbance of the EMN film is 94 % and control film is 92 %, leading to a 2 % absorption enhancement with EMN. Thus, the CdTe films used here are more highly absorbing than that used in conventional devices. One hypothesis

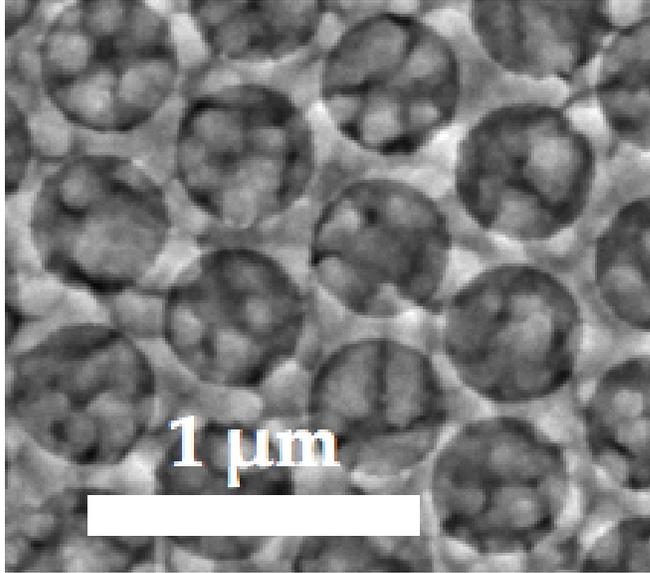


Figure D-6: SEM image of CdTe device after nanopatterning, before second deposition of CdTe. The array of holes in Ag is clearly visible, with 500 nm pitch. The metal is 30 nm thick.

is the grains lead to higher absorption than a pure amorphous film. However, it is expected the increased absorption would come at the cost of reduced free carrier path length.

Finite element method numerical electromagnetic three-dimensional simulations of the devices were performed using Comsol Multiphysics software. Figure D-9 shows the integrated power loss density in the simulated structures, normalized to the incident power; this is equivalent to absorbance. The solid lines show the absorbance of the entire device, while the dashed lines show the absorbance that occurs in CdTe. Absorbance may also occur in the metal back reflector, EMN, or FTO, accounting for the difference. Comparing the area between the two curves for the two samples shows loss from the Ag EMN must be relatively small. The poor agreement between the simulations and experimental data is likely due to the lack of surface roughness modeling in the simulations. Real FTO is textured to improve light trapping while the simulated FTO was smooth. Further, the CdTe, as discussed was granular, not

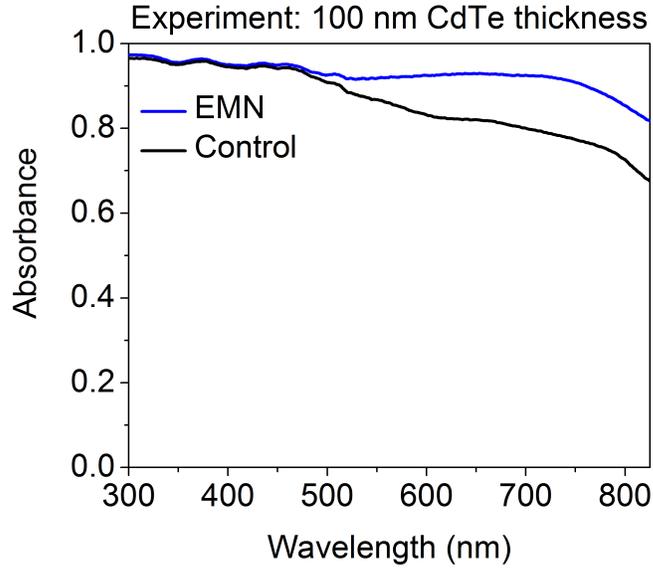


Figure D-7: Absorbance of 100 nm CdTe with EMN vs. control without EMN. The x-axis is plotted up to the band gap of CdTe. Total absorbance of the EMN film is 93 % and control film is 86 %, leading to a 8 % absorption enhancement with EMN. At the band edge, the enhancement is 16 %.

amorphous in texture, which is not captured by simulations. The oscillations in the simulated data are likely due to Fabry-Perot resonances in the FTO, which are not allowed or are damped in the real textured film.

Figure D-10 shows the power loss density in a cross section of the simulated CdTe EMN device, excited with 800 nm light. Light is incident from $-z$. Polarization is in y . The presence of resonant modes is visible due to the nanopattern. Without EMN, the color would be constant along y . The sharp corners show high localization of light, which in reality is probably damped by experimentally achievable finite corner radius.

As in the aSi case, we find that EMN may lead to improvements in the absorption of thin films of CdTe. The conclusions are complicated by the poor quality solar absorber, which may not be useful in a real device due to presumed high carrier recombination. In any case, we can extrapolate the changes in solar cell performance

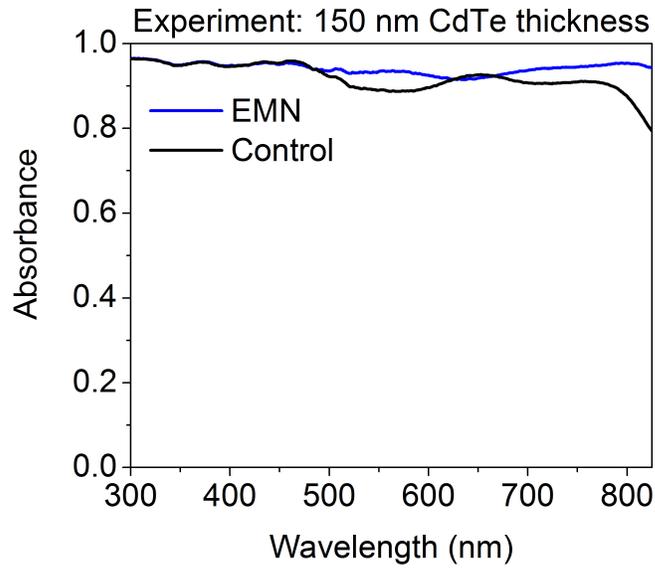


Figure D-8: Absorbance of 150 nm CdTe with EMN vs. control without EMN. The x-axis is plotted up to the band gap of CdTe. Total absorbance of the EMN film is 94 % and control film is 92 %, leading to a 2 % absorption enhancement with EMN.

expected from the enhanced absorption. We make the following assumptions: the solar spectrum considered for absorption is the AM1.5G, IQE is 89 %, fill factor is 0.77, and open circuit voltage is 0.875 V. Given these assumptions, we used the absorbance over the solar spectrum to calculate the expected current density and then efficiency. The 100 nm CdTe control sample would be expected to make a 14.4 % efficient solar cell. The 100 nm EMN device would yield 15.8 %. This represents a 10 % relative improvement in efficiency. The simulated devices show 12.4 % and 15.6 % efficiency, overall, with the EMN expected to show a 26 % relative improvement.

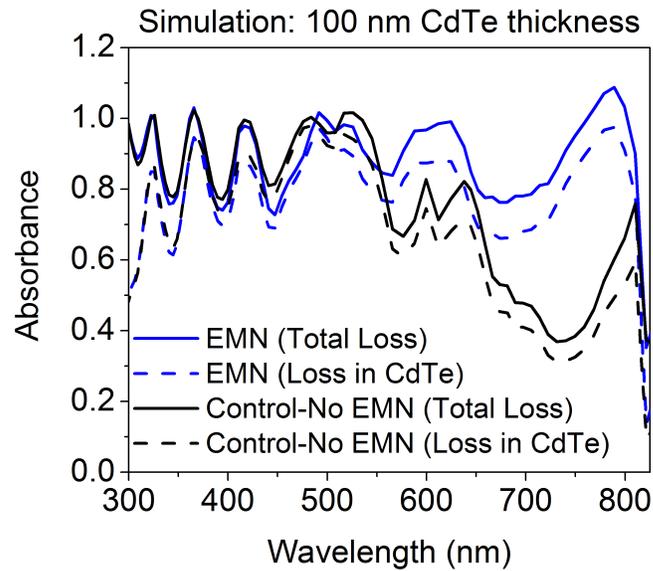


Figure D-9: Simulated absorbance of 100 nm CdTe devices with and without EMN. The dashed lines show absorption in the CdTe only. The balance of absorption may be due to absorption in the EMN or metal back reflector or FTO. The EMN device has overall absorbance 88 %, while the control has 75 %. Considering absorption in the CdTe, the EMN device absorbs 78 % and the control 66 % of incident light. The data indicate little loss occurs in the Ag EMN.

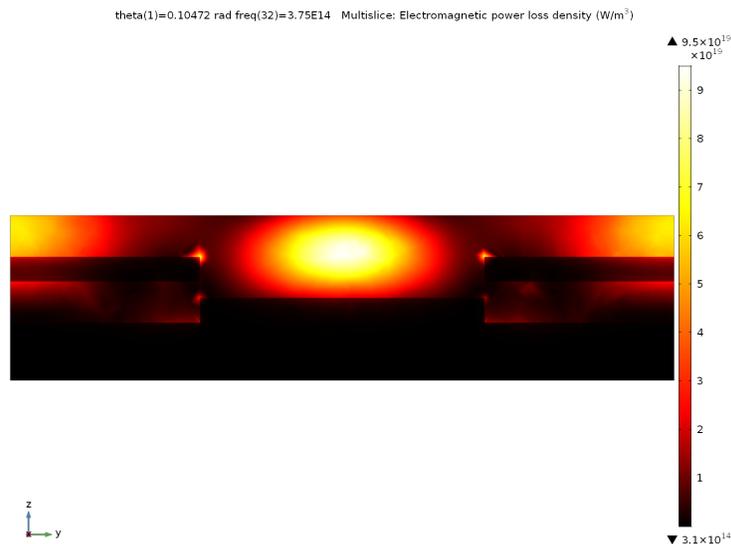


Figure D-10: Power loss density in cross section of simulated CdTe EMN device. 800 nm light is incident from $-z$. Polarization is in y . The presence of resonant modes is visible due to the nanopattern.

Appendix E

Near-infrared plasmonic nanostructures

It was proposed in refs. [134, 135] that a low energy plasmonic resonance in the near-infrared (NIR) may scatter with hot carriers in a semiconductor and that this scattering rate may be higher than for phonons in the semiconductor. Thus, the hot carriers may decay to the band edges by plasmon emission rather than phonon emission. By integrating the plasmonic resonator with a contact of a photovoltaic device, the excess energy may be extracted from the cell, which is one of the requirements for a hot carrier solar cell (the other being isentropic cooling of the hot energy, not considered here).

To this end, I fabricated plasmonic nanostructures on amorphous silicon designed to have resonance in the NIR. All fabrication was performed at the Boston College Integrated Sciences Cleanroom and Nanofabrication Facility. The nanostructures were designed based on electromagnetic finite element simulations performed by Xueyuan "Shirley" Wu.

E.1 Nanostructure

The devices were fabricated up to the point of nanostructuring by other lab members as follows. A glass substrate was used. The glass was coated with Ag to be opaque to visible light, typically 100 - 250 nm. Alumina was deposited by atomic layer deposition (ALD) to be 10 nm thick. Amorphous silicon (aSi) was deposited by sputtering (AJA sputterer). Another layer of 10 nm ALD alumina was deposited.

To create the nanostructure, I employed electron beam lithography (EBL). I used PMMA as the resist, spinning on a thin, ~ 200 nm, layer. After patterning, I deposited 20 nm Ag in a Sharon electron beam deposition system. I then lifted off the resist using Shipley Microposit Remover 1165 at 80 °C with a stir bar. Further details are found in my lab notebooks.

Three nanostructures were fabricated, designed to resonate at 1) 1,550 nm and 2) 980 nm, independent of polarization (termed "crosses") and 3) at 1,550 nm for one polarization (termed "lines"). The polarization-independent patterns were an array of square holes in the 20 nm Ag film; this can also be thought of as an array of lines crossed perpendicularly with another array of crossed lines with equivalent dimensions. The polarization-dependent pattern was an array of lines, designed to be excited with polarization perpendicular to the lines. Table E.1 shows the pitch and line width for each pattern.

Figure E-1 shows a zoomed out view of the nanostructured region and electrical contacts of the fabricated 980 nm crosses device. The nanostructures are formed in Ag, separated from the aSi by 10 nm of alumina. Also seen is an Ag back gate, which is underneath the aSi and also separated from it by 10 nm alumina. The Ag back gate also serves as a back reflector, helping define the optical modes.

Figure E-2 shows a close up of the 980 nm crosses.

Figure E-3 shows a zoomed out view of the nanostructured region and electrical contacts of the fabricated 1550 nm crosses device.

Table E.1: Parameters of fabricated nanostructures.

Name	Resonance (nm)	Pitch (nm)	Line width (nm)
980 nm crosses	980	380	80
1550 nm crosses	1,550	600	150
1550 nm lines	1,550	600	150

Figure E-4 shows a close up of the 1550 nm crosses.

Figure E-5 shows a zoomed out view of the nanostructured region and electrical contacts of the fabricated 1550 nm lines device.

Figure E-6 shows a close up of the 1550 nm lines.

E.2 Results

After fabrication, the three devices were characterized by optical reflectance measurements.

Figure E-7 shows the reflectance of the 980 nm crosses sample. Note that the blue data, from a Silicon photodetector-based spectrometer is only valid below the Si bandgap of 1,100 nm; the spectrometer outputs data beyond this for unknown reasons and here the raw data is shown. The FTIR spectrum was taken at Boston College using a Bruker Hyperion microscope coupled to a Bruker Vertex FTIR. The FTIR covers longer wavelengths, thus we only obtain reflectance data from it after 1,200 nm. Similar features, except for an extra resonance at 1,500 nm are seen compared to the simulated device, with some shifting to be expected from the imperfect matching of experiment and simulation.

Figure E-8 shows the absorbance (defined as $1 - R$, where R is reflectance) of the

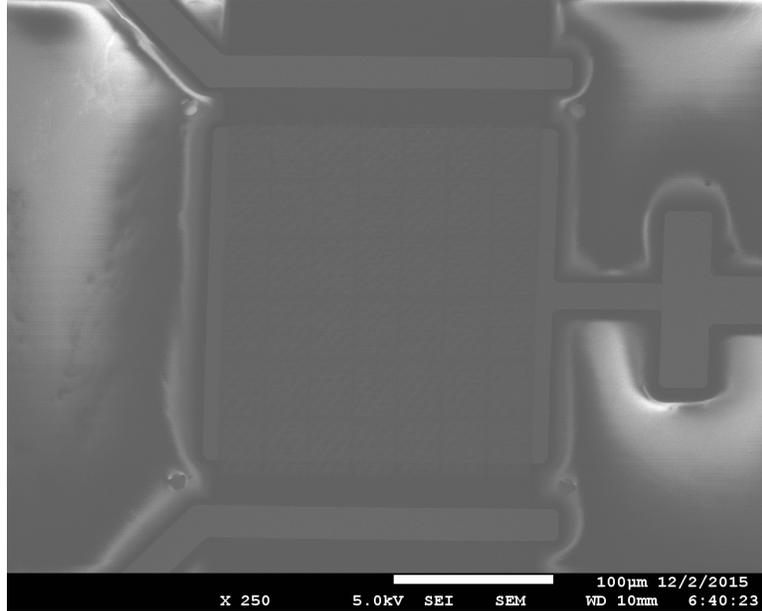


Figure E-1: SEM of 980 nm crosses device. The stripe down the center of the image is the aSi sample region. The lighter thick lines at the top and bottom are contact pads to the aSi. The lighter line that starts on the right of the image, makes a cross, and then a large square under the aSi is a gate and also serves as a back reflector, helping define the optical modes. The darker region that spans the gate vertically except for the side edges is the nanostructured region. The aSi is sandwiched between the gate on bottom and the nanostructure on top, with 10 nm alumina between each layer. The contact pads directly contact the aSi.

1550 nm crosses device. There is good agreement between the simulated data and FTIR data, within experimental error of the device geometry that causes the peaks to shift.

Finally, figure E-9 shows the reflectance of the 1550 nm lines device. There is good agreement between the simulated data and FTIR data, within experimental error of the device geometry that causes the peaks to shift.

To conclude, we fabricated devices with resonances tuned in the near-infrared (NIR). The qualitative behavior of the spectral responses of the devices matches that of simulation well, with the shapes of the curves shifted. This shift could be easily compensated for by fabricating a series of devices with varying pitch. The pitch will tune the resonance condition and the fabrication parameters could be tuned to match

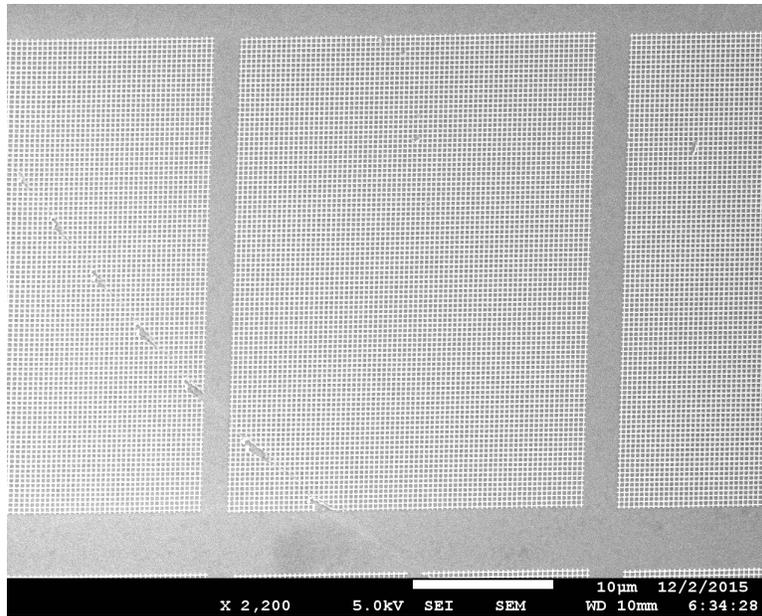


Figure E-2: SEM of 980 nm crosses device showing the nanostructures on 10 nm alumina on aSi. A scratch can be seen across the middle and left regions.

the desired resonance.

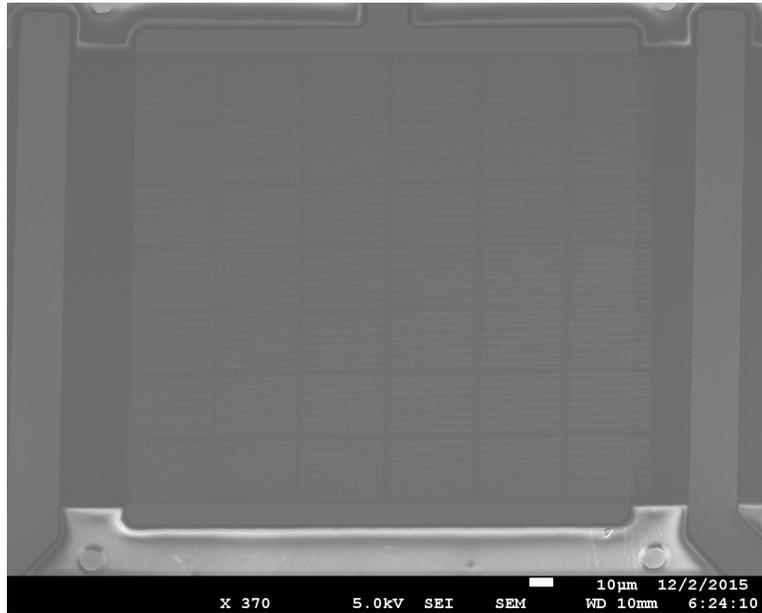


Figure E-3: SEM of 1550 nm crosses device. Similar to the previous figure, but turned counter-clockwise 90°, with 1550 nm crosses pattern.

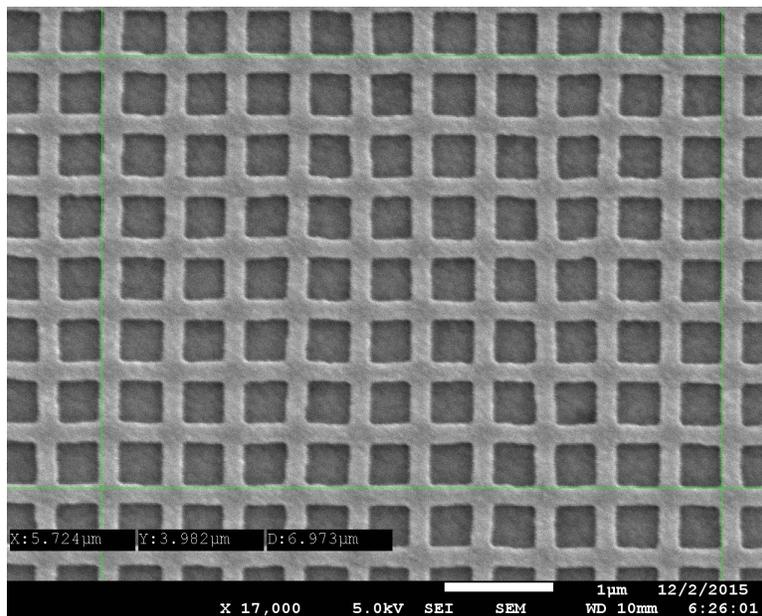


Figure E-4: SEM of 1550 nm crosses device showing the nanostructures on 10 nm alumina on aSi.

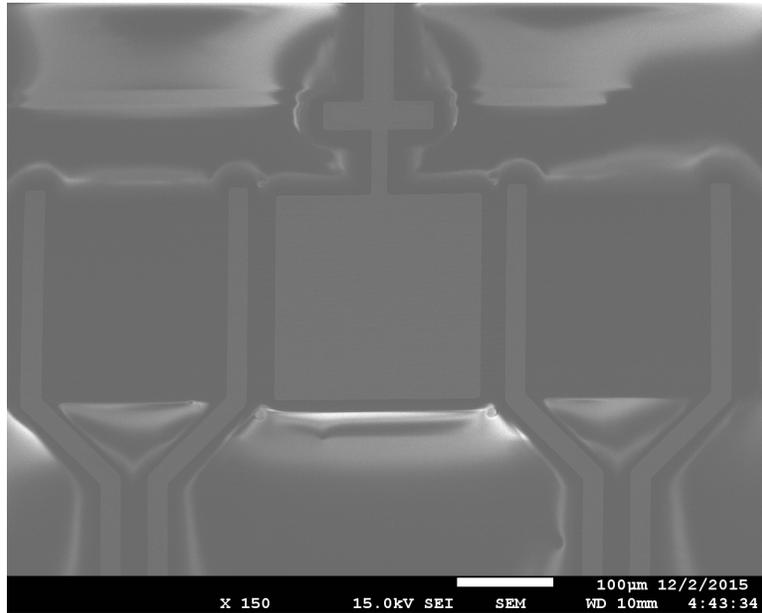


Figure E-5: SEM of 1550 nm lines device. Similar to the previous figure, but zoomed out further, with 1550 lines pattern.

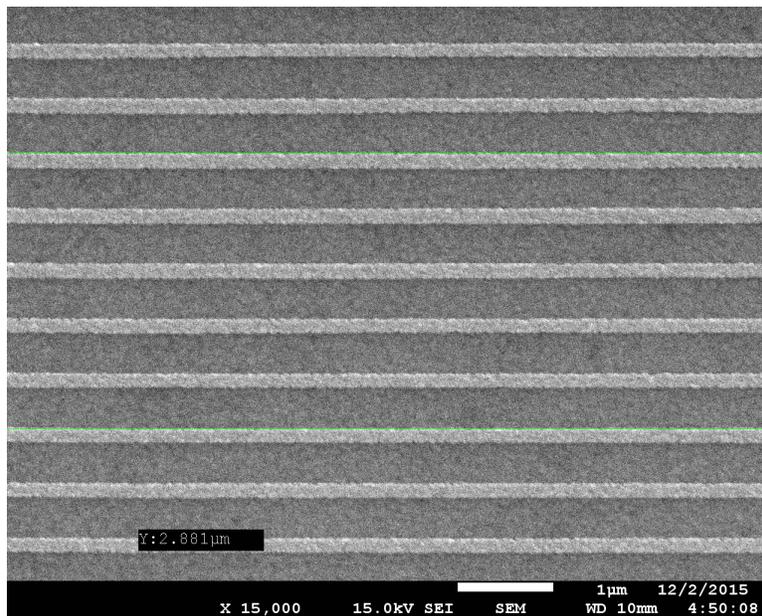


Figure E-6: SEM of 1550 nm lines device showing the nanostructures on 10 nm alumina on aSi.

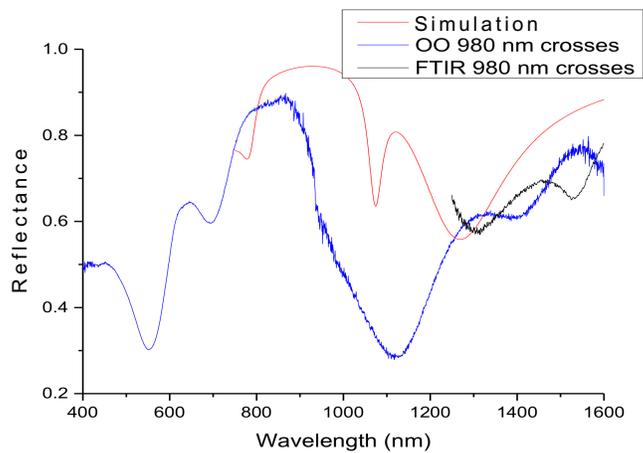


Figure E-7: Reflectance of 980 nm crosses device. The red line is the simulated device, showing actual resonance as a dip near 1,050 nm. The blue data is from an Ocean Optics (OO) spectrometer. NOTE: OO data for wavelengths longer than 1,100 nm is invalid. The black data is from unpolarized FTIR. A broad resonance near 1,300 nm is picked up by the FTIR and matches the simulated spectrum with a slight shift.

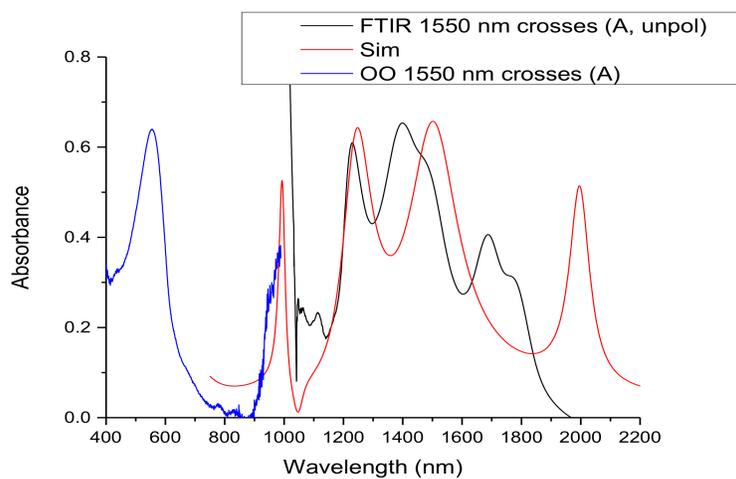


Figure E-8: Absorbance of 1550 nm crosses device. The red line is the simulated device, showing actual resonances as peaks near 1,250 nm, 1,500 nm, and 2,00 nm. The blue data is from an Ocean Optics spectrometer. The black data is from unpolarized FTIR. From the FTIR data, peaks appear at 1,200 nm, 1,400 nm, and 1,700 nm, matching the FTIR data well, although shifted. In addition to matching of the peaks (albeit shifted), the dips also match well (again, shifted).

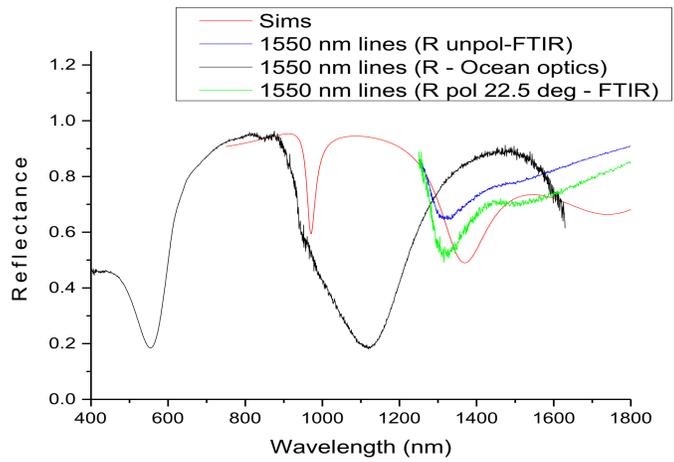


Figure E-9: Reflectance of 1550 nm lines device. The red line is the simulated device, showing actual resonance as dips near 950 nm and 1,350 nm. The black data is from an Ocean Optics (OO) spectrometer. NOTE: OO data for wavelengths longer than 1,100 nm is invalid. The blue data is from unpolarized FTIR, while the green data is from polarized FTIR, tuned to minimize the dip near 1,300 nm. From the FTIR data, the simulated 1,350 nm resonance is shifted slightly to 1,300, but matches shape well.

Appendix F

MATLAB code for transfer matrix model

The following is the code used to for the transfer matrix method modeling. It can be downloaded from GitHub or MathWorks's website. [13]

The first section shows the dependencies of each script, or which other scripts call or are called by the script. Subsequent sections show the actual code. The final section describes the materials properties input files.

F.1 Dependencies

Table F.1 shows the dependencies of each MATLAB script.

Table F.1: Transfer matrix model MATLAB scripts' dependencies.

MATLAB File List	Children (called functions)	Parents (calling functions)
ExampleUsage	AvsAngle, plotDispersion	
AvsAngle	thinfilmRTA	ExampleUsage
plotDispersion		ExampleUsage
thinfilmRTA	jreftran_rt	AvsAngle
jreftran_rt		thinfilmRTA

F.2 Code

F.2.1 ExampleUsage

See next page.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Plot A as dispersion to help compare to spectrometer output
% Setup output directory
date = '2019-10-16'; % Used for saving files
suffix = 'Ag_1L-MoS2-using-exp-nk'; % Used to create new folders if repeating same runs
on same day
experimentFolder = fullfile('Output', date, suffix);
directory = fullfile(pwd, experimentFolder);
mkdir(directory);

% Input parameters
lam0 = 250; % Smallest wavelength of interest (nm)
lam1 = 1650; % Largest wavelength of interest (nm)
dlam = 1; % Wavelength interval/resolution (nm)
layers = {'NBK7-5' 'ti-palik' 'nrelAg' 'Ag2S' '1-film-mos2-on-ag' 'slg' 'nrelAg' 'air'};
thicknesses = [1 43 0.493*0.55 40 500 100]; % nm
angle = [1:2:89];
polarization = 1;
dispersionX = 1; % 0 for kx or 1 for theta on the x-axis

range = [1 80 0.7494 3.715]; %[0.0039 0.0267 0.7494 3.715];

% Used for AvsAngle
wl = lam0:dlam:lam1; % Creates a vector from the desired wavelength range.
% If lam0 + n*dlam doesn't equal lam1 for any n, it
% will round down to the nearest value to lam1

for n = 20:20:180
    thicknesses(5) = n;
    [A, n_substrate]=AvsAngle(lam0, lam1, dlam, layers, thicknesses, angle, polarization,
    wl);
    plotDispersion(angle, wl, A, n_substrate, layers, thicknesses, polarization, dispersionX,
    range);
end

clearvars date suffix lam0 lam1 dlam layers angle polarization R directory
experimentFolder n_substrate note thicknesses wl;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

F.2.2 AvsAngle

See next page.

```

function [A, n_substrate]=AvsAngle(lam0, lam1, dlam, layers, thicknesses, angle,
polarization, wl)
% Preallocate memory
A = zeros(length(wl), length(angle));
% Calculate A for every angle
for q = 1:length(angle)
    [~, n_substrate,~,~,A(:,q),~,~,~,~]=thinfilmRTA(lam0, lam1, dlam, layers,
thicknesses, angle(q), polarization);
end

```

F.2.3 plotDispersion

See next page.

```

function [] = plotDispersion(angle, wl, A, n_substrate, layers, thicknesses,
polarization, dispersionX, range)
%% This function plots dispersion of a thin film stack
    % dispersionX, 0 for kx or 1 for theta on the x-axis

%% Set up matrices for plotting
    % pcolor is used to make the surface plot, however, it throws away data
    % rather than just plotting a color pixel for every data point
    % The code below creates matrices out of the x and y axes that
    % correspond to the edges of where each pixel should be rather than the
    % center coordinates of each pixel. The, the absorptivity matrix has a
    % blank row and column added so that when pcolor deletes the last row
    % and column it won't lose any information.

    % Compute E vector based on wavelength range of interest
    eV = 1240 ./ wl'; % Convert wavelengths to energy (eV)

    % Turn eV and angle vectors into matrices the same size as A matrix
    eVtemp = eV;
    angleTemp = angle;
    Nangles = length(angle);
    for l=1:Nangles
        eV(:,l) = eVtemp;
    end
    Nwavelengths = length(wl);
    for l=1:Nwavelengths
        angle(l,:) = angleTemp;
    end

    % Create the matrices for the pixel edges for y axis as energy and x axis
    % as angle
    x = angle;
    y = eV;

    % Find differences between adjacent points
    xSplit = diff(x,1,2)/2;
    ySplit = diff(y,1,1)/2;

    % Create data points for the edge of each pixel by adding or
    % subtracting the difference between adjacent points
    xEdges = [x(:,1)-xSplit(:,1) x(:,2:end)-xSplit x(:,end)+xSplit(:,end)];
    yEdges = [y(1,:)-ySplit(1,:); y(2:end,:)-ySplit; y(end,:)+ySplit(end,:)];

    % Add an extra row or column so the x and y matrices will have the
    % same size
    xEdges = [xEdges; xEdges(1,:)];
    yEdges = [yEdges yEdges(:,1)];

    % Make the absorptivity data matrix the same size as the pixel edge
    % matrices. The last row and column are ignored. The default shading of
    % the pcolor function will color each pixel according to the value at
    % the corner of that pixel with the smallest indices
    data = A; % save a copy of the original datafile
    A = [[A zeros(size(A,1),1)] ; zeros(1,size(A,2)+1)];

    % Plot vs kx or angle
    if dispersionX == 1 % then x-axis is angle (deg)
        xlabel = '\textbf{Angle} $\theta$ \ (degrees)$';
        xaxis = xEdges;
    else % x-axis is wavenumber (nm-1)
        xlabel = '\textbf{Wavenumber} $\frac{2\pi}{\lambda n \sin\theta}$ \ (nm^{-1})$';
    end

    % Create a matrix for kx. This uses the matrices of pixel edge
    % positions for energy (wavelength), angle, and a new one for the
    % substrate refractive index.

    % Create the new substrate index matrix
    n=n_substrate;
    n_substrateTemp = n;
    for l=1:Nangles
        n(:,l) = n_substrateTemp;
    end
end

```

```

        % Make it the same size as the other pixel edge matrices
nSplit = diff(n,1,1)/2;
nEdges = [n(1,:)-nSplit(1,:); n(2:end,:)-nSplit; n(end,:)+nSplit(end,:)];
nEdges = [nEdges nEdges(:,1)];

        % Calculate the new kx matrix
%       wlEdges = 1240 ./ yEdges;
kxEdges = 2*pi.*nEdges.*yEdges/1240.*sin(pi/4+asin(sin(pi*xEdges/180-
pi/4)./nEdges));
xAxis = kxEdges;

%           % While we're here, let's output new E vs kx datafiles
%           directory = fullfile(pwd, 'Output Data'); % create folder name to save
%           dlmwrite(fullfile(directory, 'eV.txt'), eV, 'delimiter', '\t'); % Use the
original eV data
%           % Have to create kx that corresponds to the original data, not
%           % the pixel edges for the plots
%           kx = 2*pi.*n.*eV/1240.*sin(pi/4+asin(sin(pi*angle/180-pi/4)./n));
%           %
%           dlmwrite(fullfile(directory, 'kx.txt'), kx, 'delimiter', '\t');
%           dlmwrite(fullfile(directory, 'Adata_eV_kx.txt'), data, 'delimiter', '\t');

end

yLabel = '\textbf{Photon Energy} $(eV)$';

%% Setup titles
% Create title text
stack = [layers{1}, ' / '];
note = [ ' ' layers{2} ': ' num2str(thicknesses(1)) ];
for q=2:length(layers)-1
    stack = [stack, layers{q}, ' / '];
end
if length(layers) > 3
    for q = 3:length(layers)-1
        note = [note, ' ' layers{q} ': ' num2str(thicknesses(q-1)) ];
    end
end
stack = [stack, layers{end}];
% Polarization
if polarization==0
    pol = 'TE';
else
    pol = 'TM';
end
% Put it all together
plotTitle = {'Dispersion with ' pol '-polarization'}, ['Layers: ' stack], note,
' '];

% Create name to save image
saveStack = strcat(layers{1}, '-');
for q=2:length(layers)-1
    saveStack = strcat(saveStack, layers{q}, '-');
end
saveStack = strcat(saveStack, layers{end});
saveTitle = ['Dispersion with ' pol '-polarization ', saveStack, ' ', note];
saveTitle = replace(saveTitle, ':', ''); % colons aren't allowed in Windows
filenames, so remove them

%% Set up plot
% set(0, 'DefaultFigureVisible', 'off'); % Don't display the plot--just save it
font = 24;

Plot = figure;
set(Plot, 'Units', 'normalized');
set(Plot, 'Position', [0 0 1 1]);
axes('FontSize', font) ;
xlabel(xLabel, 'FontSize', font, 'Interpreter', 'latex');
ylabel(yLabel, 'FontSize', font, 'Interpreter', 'latex');

hold on

```

```

plot = pcolor(xAxis,yEdges,A);
plot.EdgeColor = 'none';
colormap('hot')
axis(range)
c=colorbar;
caxis([0 inf])
c.Label.String = 'Absorptivity';
title(plotTitle, 'FontSize', font+1)
ax = gca;
ax.TickDir = 'out';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
hold off

% Save figures
f = get(groot,'CurrentFigure'); % f.Number will give current fig number
directory = fullfile(pwd, 'Figures'); % create folder name to save
mkdir(directory); % create the folder
saveas(Plot, fullfile(directory,[saveTitle '_' num2str(f.Number) '.png']));
savefig(Plot, fullfile(directory,[saveTitle '_' num2str(f.Number) '.fig']));

% output new E vs kx datafiles
directory = fullfile(pwd, 'Output Data'); % create folder name to save\
mkdir(directory);
dlmwrite(fullfile(directory, [saveTitle '_' num2str(f.Number) 'eV.txt']),
eV, 'delimiter', '\t'); % Use the original eV data

if dispersionX == 1 % then x-axis is angle (deg)
dlmwrite(fullfile(directory, [saveTitle '_' num2str(f.Number) 'angle.txt']),
x, 'delimiter', '\t');
dlmwrite(fullfile(directory, [saveTitle '_' num2str(f.Number) 'Adata_eV_angle.txt']),
data, 'delimiter', '\t');
else % x-axis is wavenumber (nm-1)

% Have to create kx that corresponds to the original data, not
% the pixel edges for the plots
kx = 2*pi.*n.*eV/1240.*sin(pi/4+asin(sin(pi*angle/180-pi/4)./n));

dlmwrite(fullfile(directory, [saveTitle '_' num2str(f.Number) 'kx.txt']),
kx, 'delimiter', '\t');
dlmwrite(fullfile(directory, [saveTitle '_' num2str(f.Number) 'Adata_eV_kx.txt']),
data, 'delimiter', '\t');
end

```

F.2.4 `thinfilmRTA`

See next page.

```

function [wl, n_substrate, R, T, A, nData, nDataInterp, kData, kDataInterp, N] =
thinfilMRTA(lam0, lam1, dlam, layers, thicknesses, angle, polarization)

%% This script outputs spectral R, T, A for thin film stacks

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% About this script %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% How to use this function, and functions in general (basic usage)
% This file is a function, so you call it from either the command
% window or another script with thinfilMRTA(...) and filling in each of
% the variables according to the description below.
% To access the output variables either in the workspace or in
% another script, you would use: [wl, n_substrate, R, T, A, nData,
% nDataInterp, kData, kDataInterp, N] = thinfilMRTA(...), where each of
% the variables in [] can be called anything (i.e., you don't have to
% use the names above) and if a variable is unneeded it can be replaced
% with ~ to save computation time. Further, to suppress output of each
% variable in the Command Window, just add ; at the end of the command.

%% Input variables
% lam0 Smallest wavelength of interest (nm)
% lam1 Largest wavelength of interest (nm)
% dlam Wavelength interval/resolution (nm)
% layers is a cell array entered e.g. {'fused silica' 'Ag' 'air'}
% The first and last layers will be modeled as lossless
% semi-infinite slabs; they can have dispersion but only the
% real part of the refractive index will be taken.
% Light is incident from first layer
% Dielectric function are pulled from the 'Refractive Indices'
% folder in the same directory as this function.
% Naming of refractive index files is, e.g. 'Ag_nm_n' where all
% files use 'nm,' 'n' can be 'n' or 'k,' and the material name
% changes.
% Call the layer based on the name of the file before the first
% underscore, e.g. 'Ag'
% thicknesses is a vector (nm) that doesn't include the first or last
% layers since they are always semi-infinite slabs
% angle of incidence (deg)
% polarization, 0 for TE (s-polarized), otherwise (any value other than
% 0) TM (p-polarized)

%% Output variables
% wl, a 1xn vector containing the wavelengths studied (nm)
% n_substrate, a nxl vector containing the real part of the index of
% refraction of the substrate for each corresponding wavelength.
% The underlying program neglects the imaginary part of the
% substrate. Having n_substrate is using for calculating dispersion
% plots
% R, T, and A are the reflectance, transmittance, and absorptivity (or
% absorptance, but not absorbance...) at each wavelength, as a
% decimal value, in a 1xn vector
% nData and kData are cells (which are just arrays that can contain any
% kind of information (not just numbers) with varying sizes) that
% contain the actual imported real and imaginary refractive index
% and wavelength data
% nDataInterp and kDataInterp are the interpolated/extrapolated
% refractive index data. Wavelengths are not included as they
% correspond to the wl variable. Included so one can check that the
% refractive index data being used makes sense, especially if it's
% being extrapolated.
% N is the number of layers in the stack

%% jreftran_rt.m
% This script (thinfilMRTA.m) calls upon the jreftran_rt.m script
% written by someone else
% function [r,t,R,T,A]=jreftran_rt(l,d,n,t0,polarization)
% l = free space wavelength, nm
% d = layer thickness vector, nm
% same length as n vector
% Ex: [NaN,200,NaN], 1st and last layers are semi-infinite slabs
% n = layer complex refractive index vector n = n + ik
% same length as d vector

```

```

%      Ex: [1,0.9707 + 1.8562i,1], 1st and last layers are lossless
%      t0 = angle of incidence, radians
%      polarization should be 0 for TE (s-polarized), otherwise (any value
%      other than 0) TM (p-polarized)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Start importing and interpolating the dielectric functions at the
% appropriate wavelengths
% Note: dielectric function = refractive index

wl = lam0:dlam:lam1; % Creates a vector from the desired wavelength range.
% If lam0 + n*d lam doesn't equal lam1 for any n, it
% will round down to the nearest value to lam1

% Infer file names from function input and import the Refractive Index data
delimiter = '\t'; % Dispersion files are tab delimited
N = length(layers); % Number of layers

% Preallocate memory for faster computation
% Cells are used for generic arrays or vectors of strings (or any type
% of data) while actual vectors or arrays can only contain numbers and
% have to have a regular shape (e.g. cell {1 2 3; 4 5} is OK, but array
% [1 2 3; 4 5] is not OK
nFileName = cell(1,N); % n for real part of refractive index
kFileName = cell(1,N); % k for imaginary part of refractive index
nFullFileName = cell(1,N);
kFullFileName = cell(1,N);
nData = cell(1,N);
kData = cell(1,N);
nDataInterp = cell(1,N);
kDataInterp = cell(1,N);

for k = 1:N
    nFileName{k} = strcat(layers{k}, '_nm_n.txt'); % Create the n file names
    kFileName{k} = strcat(layers{k}, '_nm_k.txt'); % Create the k file names
    % Create the n file names including the full path
    % Note that using fullfile to create directory paths should be OS
    % agnostic. E.g., '~/Refractive Indices/Ag_nm_n' would work on a Mac, but
    % not on a PC
    nFullFileName{k} = fullfile(pwd, 'Refractive Indices', nFileName{k});
    % Create the k file names including the full path
    kFullFileName{k} = fullfile(pwd, 'Refractive Indices', kFileName{k});

    % Import the data
    % Call wavelength (nm) as, eg, nData{1}(:,1), and
    % call index as, eg, nData{1}(:,2)
    % where {1} says to use the first layer's data, : says to take all
    % rows, and 1 says to take the first column, so you'll end up with
    % nx1 vectors for this example
    nData{k} = importdata(nFullFileName{k}, delimiter);
    kData{k} = importdata(kFullFileName{k}, delimiter);

    % Interpolate and extrapolate the data at the requested wavelengths
    % pchip will also be used to extrapolate. If extrapolation method isn't
    % good, add data points to dispersion text files to correct (if valid
    % to do so)
    nDataInterp{k} = interp1(nData{k}(:,1),nData{k}(:,2),wl,'pchip');
    kDataInterp{k} = interp1(kData{k}(:,1),kData{k}(:,2),wl,'pchip');
end

%% Construct a vector of n and k data
% Ignore wavelength data as the interpolated values all share the same
% wavelength vector, wl
n = zeros(length(wl),N); % preallocate memory
n(:,1) = nDataInterp{1}; % First medium is lossless, so only taken n data
n(:,end) = nDataInterp{end}; % Final medium is lossless, so only taken n data
n_substrate = n(:,1); % Output substrate index to help with dispersion plots

% For other films, n = n + ik
for k = 2:N-1
    n(:,k) = nDataInterp{k}+1i*kDataInterp{k};

```

```

end

%% Construct a vector of the thicknesses for each layer
d = zeros(1,length(thicknesses)+2); % preallocate memory
d(1) = NaN; % First and last media are semi-infinite
d(end) = NaN;

% Other films take thickness specified when calling function
for p = 2:length(d)-1
    d(p) = thicknesses(p-1);
end

%% Call the transfer matrix R, T, A calculator function and build spectra
t0 = pi*angle/180; % Convert angle to radians
l = length(wl);
% Preallocate memory
R = zeros(1,l);
T = zeros(1,l);
A = zeros(1,l);
for m = 1:l
    % Reflection and transmission coefficients, r, t, not used, so
    % replace output with ~. Can add back in if needed.
    % [r(m),t(m),R(m),T(m),A(m)]=jreftran_rt(wl(m),d,n(m,:),t0,polarization);
    % full form of jreftran_rt
    [~,~,R(m),T(m),A(m)]=jreftran_rt(wl(m),d,n(m,:),t0,polarization);
end

```

F.2.5 jreftran_rt

See next page.

```

% jreftran_rt.m
% Reflection and transmission coefficient calculator of lossy multilayer
% media for optical scientists

% Shawn Divitt
% ETH Zurich
% Photonics Group
% Ver. 2.1, 15 Nov 2016

%-iwt time dependence

%This program produces the complex reflection and transmission coefficients
%of a multilayer stack given the angle of incidence, polarization,
%wavelength, complex-refractive index of each layer, and thickness of each
%layer. The program assumes lossless dielectric incident and exit media.
%The program also assumes that all layers are non-magnetic. Magnetic media
%can be handled by more general theory (see technical report cited below).

%This program was inspired by the technical report written by K. Pascoe,
%"Reflectivity and Transmissivity through Layered, Lossy Media: A
%User-Friendly Approach," 2001. See the following link for the technical
%paper:
%http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA389099
% This link worked better for me: http://www.dtic.mil/dtic/tr/fulltext/u2/a389099.pdf

% usage:
% [r,t,R,T,A]=jreftran_rt(l,d,n,t0,polarization)
% where
% l = free space wavelength, in nanometers
% d = layer thickness vector, in nanometers
% n = layer complex refractive index vector
% t0= angle of incidence
% polarization should be 0 for TE (s-polarized), otherwise TM (p-polarized)
% r = reflection coefficient
% t = transmission coefficient
% R = reflectivity (fraction of intensity that is reflected)
% T = transmissivity (fraction of intensity that is transmitted)
% A = absorptivity (fraction of intensity that is absorbed in the film
% stack), referenced paper notes why it does not use the A = 1-R-T
% formalism
% Note:
% The layer thickness vector "d" should be the same length as the refractive index
vector, even though the first and last entries are not used.
% Example: Finding the coefficients for a 200nm gold layer surrounded by air, using
the Johnson and Christy data
% input:
% [r,t,R,T,A]=jreftran_rt(500,[NaN,200,NaN],[1,0.9707 + 1.8562i,1],0,0)
% output:
% r = -0.4622 - 0.5066i
%
% t = -0.0047 + 0.0097i
%
% R = 0.4702
%
% T = 1.1593e-04
%
% A = 0.5296

function [r,t,R,T,A]=jreftran_rt(l,d,n,t0,polarization)
%l = free space wavelength, nm
%d = layer thickness vector, nm
%n = layer complex refractive index vector
%t0= angle of incidence
%polarization should be 0 for TE (s-polarized), otherwise TM (p-polarized)

Z0=376.730313; %impedance of free space, Ohms

%the line below had mistakenly been a Z0/n instead of a n/Z0 in version 1!
Y=n./Z0; %admittance in terms of impedance of free space and refractive index, assuming
non-magnetic media
g=li*2*pi*n/l; %propagation constant in terms of free space wavelength and refractive
index

```

```

%all of the calculations rely on cosine of the complex angle, but we can
%only find the sine of the complex angle from snells law. So we use the
%fact that cos(asin(x))=sqrt(1-x^2)
%t=asin(n(1)./n*sin(t0)), complex theta for each layer
ct=sqrt(1-(n(1)./n*sin(t0)).^2); %cosine theta

if polarization==0
    eta=Y.*ct; %tilted admittance, TE case
else
    eta=Y./ct; %tilted admittance, TM case
end

delta=1i*g.*d.*ct;

M=zeros(2,2,length(d));

for j=1:length(d)
    M(1,1,j)=cos(delta(j));
    M(1,2,j)=1i./eta(j).*sin(delta(j));
    M(2,1,j)=1i*eta(j).*sin(delta(j));
    M(2,2,j)=cos(delta(j));
end

M_t=[1,0;0,1]; %M total
for j=2:(length(d)-1)
    M_t=M_t*M(:, :, j);
end

r=(eta(1)*(M_t(1,1)+M_t(1,2)*eta(end))-
(M_t(2,1)+M_t(2,2)*eta(end)))/(eta(1)*(M_t(1,1)+M_t(1,2)*eta(end))+(M_t(2,1)+M_t(2,2)*eta
(end)));
t=2*eta(1)/(eta(1)*(M_t(1,1)+M_t(1,2)*eta(end))+(M_t(2,1)+M_t(2,2)*eta(end)));

R=abs(r)^2;
T=real(eta(end)/eta(1))*abs(t)^2;
A=(4*eta(1)*real((M_t(1,1)+M_t(1,2)*eta(end))*conj(M_t(2,1)+M_t(2,2)*eta(end))-
eta(end)))/abs(eta(1)*(M_t(1,1)+M_t(1,2)*eta(end))+(M_t(2,1)+M_t(2,2)*eta(end)))^2;

```

F.3 Refractive Indices

The code relies on text files containing the refractive indices of each material entered into the model. These files are placed in a folder called 'Refractive Indices' placed in the same directory as the MATLAB scripts. The format of the refractive index files is: tab delimited text files with '.txt' extension; first column is wavelength in nm; second column is either n or k; separate files for n and k; must have both n and k files; uses $n = n+ik$ formalism; naming is 'x_nm_y' where x is the material name, nm is the unit, and y is either n or k. For materials with negligible k, create a file with 0s for k.

Appendix G

MATLAB code Strong coupling analysis of Chapter 5

The following is the code used to analyze the data and produce the dispersion plots shown in Chapter 5. It can be easily applied to analyze the data of Chapter 4.

Step1_processAndPlotData

```
%% Imports the JA Woollam ellipsometry reflectance data and combines it, normalizes it,
shifts the IR offset, then plots R and dispersion in k and angle space
addpath('scripts called by other scripts');
%% How to use:
% Add the original .dat data files to Input Data folder
% Add a control .txt tab-delimited file that has the same format as the
merged.txt files (see merge_1 script)
% Update the section 'Update These' as needed.
% Then run the script.
% Check the plots and find the limits of angle and wavelength that are
% acceptable to process using the data cursor on the plots
% and bring those values to step 2.

%% Update these
dateToday = '2019-09-03'; % Used to name .tsvs
plotTitle = 'Reflectance of Silver with 1 Film Few-layer CVD MoS_2, Kretschmann
Configuration'; % For R(wavelength) plot
layers = {'NBK7 prism' 'Ag' 'CVD MoS_2' 'air'}; % used for dispersion plot labels
thicknesses = {'40 nm' '~5 nm, single transfer'}; % nm used for dispersion plot labels
polarization = 1; % used for dispersion plot labels, 1 means p-pol, 0 means s-pol
substrateName = 'NBK7'; % Name used for RTA transfer matrix calculator refractive index
directory (note that k is assumed 0)

%% Import data and output ordered and detector changeover corrected data
merge_1 % First, import the data and merge into one file
normalize_2 % Then normalize vs. control
shift_3 % Correct the detector changeover offset at 1,000 nm

% Import the merged, normalized, and shifted data
dataFolder = 'Output Data';
inputData = fullfile(dataFolder, 'mergedNormalizedShifted.txt');
mergedNormalizedData = dlmread(inputData);

% Extract wavelength and angle information
wl = mergedNormalizedData(2:end, 1)';
angle = mergedNormalizedData(1,2:end);

% Then strip out the angle and wavelength data leaving only R
data = mergedNormalizedData;
data(1, :) = [];
data(:, 1) = [];

% A = 1 - R
for j = 1:length(angle)
A(:,j) = 1-data(:,j);
end

% Set limits for plots
wlLimits = [-inf inf]; % (nm) x-axis limits for R(theta)
dR = [-inf inf]; % (R) y-axis limits for R(theta)
range = [-inf inf -inf inf]; % used for dispersion plot, use -inf or inf to allow matlab
to automatically set a limit

% Pull real part of refractive index of substrate and make a vector with
% length equal to the number of wavelength points
delimiter = '\t'; % Dispersion files are tab delimited
nFileName = strcat(substrateName, '_nm_n.txt'); % Create the n file name
indexDirectory = fullfile('/Users', 'aaronrose', 'Google Drive', 'MATLAB', 'Transfer Matrix
Reflection Calculator', 'RTA Transfer Matrix Calculator v1', 'Refractive Indices'); % Path
of Refractive Indices folder
nFullFileName = fullfile(indexDirectory, nFileName);
substrateData = importdata(nFullFileName, delimiter);
n_substrate = interp1(substrateData(:,1),substrateData(:,2),wl, 'pchip');

% Save the substrate info for future reference
dlmwrite(fullfile('Output Data', 'substrateData.txt'), substrateData, 'delimiter', '\t');
```

```
plotData % plot the data
```

```
clearvars dataFolder inputData wLimits dR range dateToday
```

```
clearvars mergedNormalizedData delimiter nFileName indexDirectory nFullFileName
```

Step2_cutBadData

```
%% How to use
% Update section below. Check the plots. If they look good, continue to
% Step3. If not, adjust the values below and iterate until good.

%% Update these
minWl = 340; % nm
maxWl = 1660; % nm
minAngle = 42; % deg
maxAngle = 80; % deg

% Find the indices for that range
wlIndex = [1:length(wl)];
angleIndex = [1:length(angle)];
minWlIndex = round(interp1(wl,wlIndex,minWl));
maxWlIndex = round(interp1(wl,wlIndex,maxWl));
minAngleIndex = round(interp1(angle,angleIndex,minAngle));
maxAngleIndex = round(interp1(angle,angleIndex,maxAngle));

% cut out annoying data
angle = angle(minAngleIndex:maxAngleIndex);
wl = wl(minWlIndex:maxWlIndex);
data = data(minWlIndex:maxWlIndex,minAngleIndex:maxAngleIndex);
clear A n_substrate;

% A = 1 - R. Redo A with new limits
for j = 1:length(angle)
A(:,j) = 1-data(:,j);
end

% Redo n with new limits
n_substrate = interp1(substrateData(:,1),substrateData(:,2),wl,'pchip');

% Set limits for plots
wlLimits = [-inf inf]; % (nm) x-axis limits for R(theta)
dR = [-inf inf]; % (R) y-axis limits for R(theta)
range = [-inf inf -inf inf]; % used for dispersion plot, use -inf or inf to allow matlab
to automatically set a limit

% plot the data
plotData

clearvars wlLimits dR range minWl maxWl minAngle maxAngle wlIndex angleIndex
clearvars minWlIndex maxWlIndex minAngleIndex maxAngleIndex
```

Step3_savePlotsAndDispersionData

```
%% Plot the reflectance and dispersion and save the plots and dispersion data
% Set limits for plots
wlLimits = [-inf inf]; % (nm) x-axis limits for R(theta)
dR = [-inf inf]; % (R) y-axis limits for R(theta)
range = [-inf inf -inf inf]; % used for dispersion plot, use -inf or inf to allow matlab
to automatically set a limit

%% Reflectance plot
% Set up plot defaults
xlabel = 'Wavelength (nm)';
ylabel = 'Reflectance';

for l = 1:length(angle)
plotLabels{l} = num2str(angle(l));
end

% This will save the reflectance figure
processReflectance(wl,data,xLabel,yLabel,plotLabels,plotTitle, wlLimits, dR);

%% Plot the dispersion
% Plot it
dispersionX = 1; % 0 for kx, 1 for theta; if plotting kx, 3 data files for E, kx, data
will also be output
processDispersion(angle, wl, A, n_substrate, layers, thicknesses, polarization,
dispersionX, range);
dispersionX = 0;
processDispersion(angle, wl, A, n_substrate, layers, thicknesses, polarization,
dispersionX, range);

% clearvars dispersionX

% clearvars layers angle polarization A directory dispersionX experimentFolder
n_substrate note thicknesses wl;
%
clear all;
%
%
% Now, take the dispersion data and interpolate it so that columns of
% k-data are all the same
delimiter = '\t';
eV = importdata(fullfile(pwd,'Output Data', 'eV.txt'), delimiter);
kx = importdata(fullfile(pwd,'Output Data', 'kx.txt'), delimiter);
Adata = importdata(fullfile(pwd,'Output Data', 'Adata_eV_kx.txt'), delimiter);

minkx = min(min(kx));
maxkx = max(max(kx));
kxresfactor = 10;
kxres = (maxkx-minkx)/(size(kx,2)-1)/kxresfactor;
kxInterp = minkx:kxres:maxkx;
kxInterpGrid = repmat(kxInterp,size(eV,1),1);
eVInterpGrid = repmat(eV(:,1),1,size(kxInterpGrid,2));

for i = 1:size(eV,1)
Vq1D(i,:)=interp1(kx(i,:), Adata(i,:), kxInterp, 'makima', NaN);
end

% these data files will have constant kx along the columns
dlmwrite(fullfile('Output Data','eV_interp.txt'), eVInterpGrid,'delimiter','\t');
dlmwrite(fullfile('Output Data','kx_interp.txt'), kxInterpGrid,'delimiter','\t');
dlmwrite(fullfile('Output Data','Adata_eV_kx_interp.txt'), Vq1D,'delimiter','\t');

%% Plot it to check that it looks OK
Plot = figure;
hold on
plot = pcolor(kxInterpGrid,eVInterpGrid,Vq1D);
```

```
plot.EdgeColor = 'none';
colormap('hot')
c=colorbar;
caxis([0 inf])
c.Label.String = 'Absorptivity';
ax = gca;
ax.TickDir = 'out';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
hold off

clear all
```

Step4_fit

```
%% Fit in angle space
% This is kludgy
% See steps below

% Import data
eVgrid = dlmread(fullfile('Output Data','eV.txt'), '\t');
angleGrid = dlmread(fullfile('Output Data','angle.txt'), '\t');
A = dlmread(fullfile('Output Data','Adata_eV_angle.txt'), '\t');

% Reduce axis data to vectors
angle = angleGrid(1,:);
eV = eVgrid(:,1);

%% Step 1, run this section to find a region in angle space where all the peaks that need
to be fit can be seen in one absorptivity vs energy plot
% Update angle0 and rerun this section as needed to find such a slice of the data. The
purpose is
% to find fitting parameters and we need well defined peaks to get started
% with initial values for the fit

% Fit will start where there are all features and then fit to the ends
% where some features disappear
angle0 = 47; % Start fit here
i0 = round(interp1(angle,[1:length(angle)],angle0)); % get index value for this location

plot(eV, A(:,i0)); % Check that this is a good starting point

%% Step 2, Now open the curve fitting tool, load x vs y, and check that the init, etc
values below work to fit the data
% Update the values below for whatever gives a good fit in the tool
x = eV;
y = A(:,i0);

% Update these values from curve fitting tool
% This is the model to fit to: (a*g0/((x-x0)^2+(g0/2)^2) + b*g1/((x-x1)^2+(g1/2)^2) +
c*g2/((x-x2)^2+(g2/2)^2) + d*g3/((x-x3)^2+(g3/2)^2) )/(2*pi)
% It can also be updated to remove or add peaks to get a good fit. Be sure
% to update the number of init, upper, and lower parameters accordingly.
% Represents a sum of Lorentzians
init = [0.010 0.045 0.28 0.26 0.068 0.17 0.33 0.63 1.85 1.99 2.24 2.83]; %
Starting fit parameters, tuned to work for angle(i0)
upper = [1.000 1.000 1.000 1.000 2.000 2.000 2.000 2.000 1.96 2.15 2.94 3.55];
lower = [0.0001 0.001 0.001 0.001 0.0001 0.001 0.001 0.001 min(eV) 1.82 1.92 2.47];

%% Step 3, try to fit the absorptivity vs energy plots at every angle
% The script runFitRoutine should be updated for whichever model was used
% in the fitting app
% Also, the peakinfo, coefficients, and init2 at the end of the loop need
% to be updated based on how many fit parameters there are in the model
% At this stage, you could clear all and run the whole script
fitRegion = 1:length(angle); % Indices corresponding to angle range of interest
l=length(fitRegion);
fitSequence = [i0:1:fitRegion(end) (i0-1):-1:fitRegion(1)]; % Start with i0, then go up
one index at a time, using the solved for fit parameters as the initial parameters of the
next solution; at the last value, go back to starting value and work back down

fitresult = cell(1,1); % Initialize cell to store fit results
gof = cell(1,1); % Initialize to store goodness of fit results
init2 = init;
%(a*g0/((x-x0)^2+(g0/2)^2) + b*g1/((x-x1)^2+(g1/2)^2) + c*g2/((x-x2)^2+(g2/2)^2) +
d*g3/((x-x3)^2+(g3/2)^2) )/(2*pi)
for i = fitSequence
% n=i - totalFitRegion(1)+1;
n = i - fitRegion(1)+1;
y = A(:,i);
```

```

% Note that script runFitRoutine also needs to be updated for current
% data--not very general yet, code not well developed at all
% Note that it used to be runFitRoutine, but that go over written to
% fit the plasmon, by accident. I saved the backup copy as
% runFitRoutineData--hopefully this still works. i copied and pasted
% the fit equation from this backup above
[fitresult{n,1}, gof{n,1}] = runFitRoutineData(x, y, init2, upper, lower);

peakinfo(n,1) = fitresult{n,1}{1}.a;
peakinfo(n,2) = fitresult{n,1}{1}.b;
peakinfo(n,3) = fitresult{n,1}{1}.c;
peakinfo(n,4) = fitresult{n,1}{1}.d;

peakinfo(n,5) = fitresult{n,1}{1}.g0;
peakinfo(n,6) = fitresult{n,1}{1}.g1;
peakinfo(n,7) = fitresult{n,1}{1}.g2;
peakinfo(n,8) = fitresult{n,1}{1}.g3;

peakinfo(n,9) = fitresult{n,1}{1}.x0;
peakinfo(n,10) = fitresult{n,1}{1}.x1;
peakinfo(n,11) = fitresult{n,1}{1}.x2;
peakinfo(n,12) = fitresult{n,1}{1}.x3;

peakinfo(n,13) = angle(i);

% Coefficients
a = fitresult{n,1}{1}.a;
b = fitresult{n,1}{1}.b;
c = fitresult{n,1}{1}.c;
d = fitresult{n,1}{1}.d;

g0 = fitresult{n,1}{1}.g0;
g1 = fitresult{n,1}{1}.g1;
g2 = fitresult{n,1}{1}.g2;
g3 = fitresult{n,1}{1}.g3;

x0 = fitresult{n,1}{1}.x0;
x1 = fitresult{n,1}{1}.x1;
x2 = fitresult{n,1}{1}.x2;
x3 = fitresult{n,1}{1}.x3;

% Update initial coefficients
init2 = [a b c d g0 g1 g2 g3 x0 x1 x2 x3];
if i==(i0-1)
    init2 = init;
end
end

%% Step 4, Plot it to check that it looks OK
% These are the uncoupled exciton energies of interest, update as
% necessary
EA = 1240/666;
EB = 1240/617;
EC = 1240/487;
minAngle = min(angle);
maxAngle = max(angle);

Plot = figure;
hold on
colorPlot = pcolor(angleGrid, eVgrid, A); % Plot the dispersion
colorPlot.EdgeColor = 'none';
colormap('hot')
c=colorbar;
caxis([0 inf])
c.Label.String = 'Absorptivity';
ax = gca;

```

```

ax.TickDir = 'out';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';

% overlay the fit peak values
scatter(peakinfo(:,13),peakinfo(:,9), 'black')
scatter(peakinfo(:,13),peakinfo(:,10), 'black')
scatter(peakinfo(:,13),peakinfo(:,11), 'black')
scatter(peakinfo(:,13),peakinfo(:,12), 'black')

% overlay the uncoupled exciton positions
plot([minAngle maxAngle],[EA EA], '--k')
plot([minAngle maxAngle],[EB EB], '--k')
plot([minAngle maxAngle],[EC EC], '--k')
hold off

% Output raw peak fits to data file
dlmwrite(fullfile('Output Data','peakinfoAngleSpace.txt'),peakinfo,'delimiter','\t');

clear all;

% %% Fit in k-space
%
% eVInterpGrid = dlmread(fullfile('Output Data','eV_interp.txt'), '\t');
% kxInterpGrid = dlmread(fullfile('Output Data','kx_interp.txt'), '\t');
% A = dlmread(fullfile('Output Data','Adata_eV_kx_interp.txt'), '\t');
%
% kxInterp = kxInterpGrid(1,:);
% x = eVInterpGrid(:,1);
% % z = A;
%
%
% % xyrange = [-inf inf -inf inf]; % 0.0075 0.018
% % crange = [0 0.8]; % color range
% %
% % %% Plot it
% % font = 32;
% %
% % Plot = figure(1);
% % set(Plot, 'Units', 'normalized');
% % set(Plot, 'Position', [0 0 1 1]);
% % axes('FontSize', font) ;
% % xlabel(xLabel, '\textbf{Wavenumber} \frac{2\pi}{\lambda}n\sin{\theta} \ \ (nm^{-1})$');
% % ylabel(yLabel, 'FontSize', font, 'Interpreter','latex');
% % xlabel(xLabel, 'FontSize', font, 'Interpreter','latex');
% % ylabel(yLabel, 'FontSize', font, 'Interpreter','latex');
% %
% % hold on
% % cplot = pcolor(x,y,z);
% % cplot.EdgeColor = 'none';
% % colormap('hot')
% % axis(xyrange)
% % c=colorbar;
% % caxis(crangle)
% % c.Label.String = 'Absorptivity';
% % % title(plotTitle, 'FontSize', font+1)
% % ax = gca;
% % ax.TickDir = 'out';
% % ax.Layer = 'top';
% % ax.Box = 'on';
% % ax.XMinorTick = 'on';
% % ax.YMinorTick = 'on';
% % hold off
%
%
%

```

```

%
% % start k 422, end k end
%
% % x = kxInterp;
% % xx = eV(:,1);
% % z = Vq1D;
% % totalFitRegion = 422:length(kxInterp);
% k0 = 0.012; % Start fit here
% i0 = round(interp1(kxInterp,[1:length(kxInterp)],k0));
%
% init = [0.02531 0.07626 0.02723 0.09534 0.1484 0.1034 1.839 1.957 2.054]; % Starting
fit parameters, tuned to work for i0
% upper = [1.000 1.000 1.000 2.000 2.000 2.000 1.900 2.100 2.600];
% lower = [0.001 0.001 0.001 0.001 0.001 0.001 1.500 1.800 2.000];
% fitRegion = 430:1947; % Indices corresponding to kx range of interest 836, 422 1844
% i0 = 1250; % initial index
% % l = length(totalFitRegion);
% l=length(fitRegion);
% fitSequence = [i0:1:fitRegion(end) (i0-1):-1:fitRegion(1)]; % Start with i0, then go up
one index at a time, using the solved for fit parameters as the initial parameters of the
next solution; at the last value, go back to starting value and work back down
%
%
% fitresult = cell(1,1); % Initialize cell to store fit results
% gof = cell(1,1); % Initialize to store goodness of fit results
% init2 = init;
%
% for i = fitSequence
% % n=i - totalFitRegion(1)+1;
% n=i - fitRegion(1)+1;
% y = A(:,i);
%
% % Note that script runFitRoutine also needs to be updated for current
% % data--not very general yet, code not well developed at all
% [fitresult{n,1}, gof{n,1}] = runFitRoutine(x, y, init2, upper, lower);
%
% peakinfo(n,1) = fitresult{n,1}{1}.a;
% peakinfo(n,2) = fitresult{n,1}{1}.b;
% peakinfo(n,3) = fitresult{n,1}{1}.c;
%
% peakinfo(n,4) = fitresult{n,1}{1}.g0;
% peakinfo(n,5) = fitresult{n,1}{1}.g1;
% peakinfo(n,6) = fitresult{n,1}{1}.g2;
%
% peakinfo(n,7) = fitresult{n,1}{1}.x0;
% peakinfo(n,8) = fitresult{n,1}{1}.x1;
% peakinfo(n,9) = fitresult{n,1}{1}.x2;
%
% peakinfo(n,10) = kxInterp(i);
%
% % Coefficients
% a = fitresult{n,1}{1}.a;
% b = fitresult{n,1}{1}.b;
% c = fitresult{n,1}{1}.c;
%
% g0 = fitresult{n,1}{1}.g0;
% g1 = fitresult{n,1}{1}.g1;
% g2 = fitresult{n,1}{1}.g2;
%
% x0 = fitresult{n,1}{1}.x0;
% x1 = fitresult{n,1}{1}.x1;
% x2 = fitresult{n,1}{1}.x2;
%
% % Update initial coefficients
% init2 = [a b c g0 g1 g2 x0 x1 x2 ];
% if i==(i0-1)
% % init2 = init;
% end
% end
%
%
%
%
%

```

```

%
% % Output raw peak fits to data file
% dlmwrite(fullfile('Output Data','peakinfo-raw-
noOffset.txt'),peakinfo,'delimiter','\t');
%
%
%
% %
% % d1 = 6;
% % d2 = 6;
% % d3 = 6;
% %
% % for n = 1:length(peakinfo)
% % % Delete points at bounds
% % if round(peakinfo(n,1),d1,'significant') == round(lower(1),d1,'significant') ||
round(peakinfo(n,1),d1,'significant') == round(upper(1),d1,'significant')
% %     peakinfo(n,1) = NaN;
% % end
% %
% % if round(peakinfo(n,2),d1,'significant') == round(lower(2),d1,'significant') ||
round(peakinfo(n,2),d1,'significant') == round(upper(2),d1,'significant')
% %     peakinfo(n,2) = NaN;
% % end
% %
% % if round(peakinfo(n,3),d1,'significant') == round(lower(3),d1,'significant') ||
round(peakinfo(n,3),d1,'significant') == round(upper(3),d1,'significant')
% %     peakinfo(n,3) = NaN;
% % end
% %
% % if round(peakinfo(n,4),d2,'significant') == round(lower(4),d2,'significant') ||
round(peakinfo(n,4),d2,'significant') == round(upper(4),d2,'significant')
% %     peakinfo(n,4) = NaN;
% % end
% %
% % if round(peakinfo(n,5),d2,'significant') == round(lower(5),d2,'significant') ||
round(peakinfo(n,5),d2,'significant') == round(upper(5),d2,'significant')
% %     peakinfo(n,5) = NaN;
% % end
% %
% % if round(peakinfo(n,6),d2,'significant') == round(lower(6),d2,'significant') ||
round(peakinfo(n,6),d2,'significant') == round(upper(6),d2,'significant')
% %     peakinfo(n,6) = NaN;
% % end
% %
% % if round(peakinfo(n,7),d3,'significant') == round(lower(7),d3,'significant') ||
round(peakinfo(n,7),d3,'significant') == round(upper(7),d3,'significant')
% %     peakinfo(n,7) = NaN;
% % end
% %
% % if round(peakinfo(n,8),d3,'significant') == round(lower(8),d3,'significant') ||
round(peakinfo(n,8),d3,'significant') == round(upper(8),d3,'significant')
% %     peakinfo(n,8) = NaN;
% % end
% %
% % if round(peakinfo(n,9),d3,'significant') == round(lower(9),d3,'significant') ||
round(peakinfo(n,9),d3,'significant') == round(upper(9),d3,'significant')
% %     peakinfo(n,9) = NaN;
% % end
% % end
% %
% % % Before kx = 0.008046, or index 257, set curves 1 values to NaN
% % for n = 1:256
% %     peakinfo(n,4) = NaN;
% % end
% %
% % % Before kx = 0.01006, or index 590, set curves 2 and 3 values to NaN
% % for n = 1:590
% % %     peakinfo(n,2) = NaN;
% % %     peakinfo(n,3) = NaN;
% %     peakinfo(n,5) = NaN;
% %     peakinfo(n,6) = NaN;
% %     peakinfo(n,8) = NaN;

```

```

% %      peakinfo(n,9) = NaN;
% % end
% %
% % % After kx = 0.013823, or index 1213, set curves 1 and 2 values to NaN
% % for n = 1213:length(peakinfo)
% % %      peakinfo(n,1) = NaN;
% % %      peakinfo(n,2) = NaN;
% %      peakinfo(n,4) = NaN;
% %      peakinfo(n,5) = NaN;
% %      peakinfo(n,7) = NaN;
% %      peakinfo(n,8) = NaN;
% % end
% %
% % dlmwrite(fullfile('Output Data','peakinfo.txt'),peakinfo,'delimiter','\t');
% %
% %
% % % figure
% % % hold on
% % % scatter(peakinfo(:,10),peakinfo(:,1), 'black')
% % % scatter(peakinfo(:,10),peakinfo(:,2), 'red')
% % % scatter(peakinfo(:,10),peakinfo(:,3), 'blue')
% % %
% % % hold off
% % %
% % % figure
% % % hold on
% % % scatter(peakinfo(:,10),peakinfo(:,4), 'black')
% % % scatter(peakinfo(:,10),peakinfo(:,5), 'red')
% % % scatter(peakinfo(:,10),peakinfo(:,6), 'blue')
% % %
% % % hold off
% % %
% % % figure
% % % hold on
% % % scatter(peakinfo(:,10),peakinfo(:,7), 'black')
% % % scatter(peakinfo(:,10),peakinfo(:,8), 'red')
% % % scatter(peakinfo(:,10),peakinfo(:,9), 'blue')
% % %
% % % hold off
% % %
% % %
% % %
% % % weight = [peakinfo(:,1:3)];
% % % weight = weight./max(max(weight));
% % %
% % % % colors2x = redblue(length(peakinfo)*2-1);
% % % % colors1x = colors2x(1:length(peakinfo),:);
% % % % colors1 = colors1x.*weight(:,1);
% % %
% % % k1 = 0.01116;
% % % k2 = 0.011859;
% % % dEmin1 = 1.8;
% % % dEmax1 = 1.95;
% % % dEmin2 = 1.9;
% % % dEmax2 = 2.2;
% % %
% % % EA = 1240/666;
% % % EB = 1240/617;
% % % EC = 1240/487;
% % % kmin = 0.008;
% % % kmax = 0.016;
% % %
% % %
% % % figure
% % % hold on
% % % s1=scatter(peakinfo(:,10),peakinfo(:,7), 'black', 'filled');
% % % s2=scatter(peakinfo(:,10),peakinfo(:,8), 'red', 'filled');
% % % s3=scatter(peakinfo(:,10),peakinfo(:,9), 'blue', 'filled');
% % % s1.MarkerEdgeColor = 'none';
% % % s2.MarkerEdgeColor = 'none';
% % % s3.MarkerEdgeColor = 'none';
% % % % s1.MarkerFaceColor = 'k';

```

```

% % % s1.SizeData = weight(:,1)*200;
% % % s2.SizeData = weight(:,2)*200;
% % % s3.SizeData = weight(:,3)*200;
% % % plot([k1 k1],[dEmin1 dEmax1], '--k')
% % % plot([k2 k2],[dEmin2 dEmax2], '--r')
% % % plot([kmin kmax],[EA EA], '--k')
% % % plot([kmin kmax],[EB EB], '--k')
% % % hold off
% % %
% % % weight2 = [peakinfo(:,4:5)];
% % % weight2 = weight2./max(max(weight2));
% % %
% % % figure
% % % hold on
% % % s1=scatter(peakinfo(:,10),peakinfo(:,7), 'black', 'filled');
% % % s2=scatter(peakinfo(:,10),peakinfo(:,8), 'red', 'filled');
% % % s3=scatter(peakinfo(:,10),peakinfo(:,9), 'blue', 'filled');
% % % s1.MarkerEdgeColor = 'none';
% % % s2.MarkerEdgeColor = 'none';
% % % s3.MarkerEdgeColor = 'none';
% % % % s1.MarkerFaceColor = 'k';
% % % s1.SizeData = weight2(:,1)*200;
% % % s2.SizeData = weight2(:,2)*200;
% % % s3.SizeData = weight2(:,3)*200;
% % % plot([k1 k1],[dEmin1 dEmax1], '--k')
% % % plot([k2 k2],[dEmin2 dEmax2], '--r')
% % % plot([kmin kmax],[EA EA], '--k')
% % % plot([kmin kmax],[EB EB], '--k')
% % % hold off
% % %
% % %
% % %
% % %
% % % % first FWHM crossing at kx = 0.011116, FWHM = 1123 index 765 ->0.1036 eV
splitting
% % % omega1 = peakinfo(765,8)-peakinfo(765,7)
% % %
% % % % second FWHM crossing at kx = 0.011859, FWHM = 0.10549 index 888 -> 0.2427
% % % % eV splitting *** first and 3rd peaks -> 0.2427 eV
% % % omega2 = peakinfo(888,9)-peakinfo(888,7)
% % % % eV splitting *** second and 3rd peaks -> 0.1148 eV
% % % omega4 = peakinfo(888,9)-peakinfo(888,8)
% % %
% % % %third FWHM crossing at kx = 0.012935, FWHM = 0.19773 index 1066 -> 0.1895 eV
splitting
% % % omega3 = peakinfo(1066,9)-peakinfo(1066,8)
% % %
% % % % determine splitting
% % % dE01 = peakinfo(:,8)-peakinfo(:,7);
% % % dE12 = peakinfo(:,9)-peakinfo(:,8);
% % %
% % % k1 = 0.011116;
% % % k2 = 0.012935;
% % % dEmin = 0.05;
% % % dEmax = 0.25;
% % %
% % % figure
% % % hold on
% % % scatter(peakinfo(:,10),dE01, 'black')
% % % scatter(peakinfo(:,10),dE12, 'red')
% % % plot([k1 k1],[dEmin dEmax], '--k')
% % % plot([k2 k2],[dEmin dEmax], '--r')
% % % hold off
% % %
% % %
% % %
% % % fitRegion = 50:2:90; % Indices corresponding to angles to fit
% % % l = length(fitRegion);
% % %
% % % n=1;
% % % fitresult = cell(1,1); % Initialize cell to store fit results
% % % gof = cell(1,1); % Initialize to store goodness of fit results

```

```

% % % for i = fitRegion
% % %     y = A(:,i);
% % % [fitresult{n,1}, gof{n,1}] = createFits(x, y);
% % % peakinfo(n,1)=angle(i);
% % % peakinfo(n,2) = fitresult{n,1}{1}.g0;
% % % peakinfo(n,3) = fitresult{n,1}{1}.g1;
% % % peakinfo(n,4) = fitresult{n,1}{1}.g2;
% % % peakinfo(n,5) = fitresult{n,1}{1}.g3;
% % % peakinfo(n,6) = fitresult{n,1}{1}.g4;
% % % peakinfo(n,7) = fitresult{n,1}{1}.g5;
% % % n=n+1;
% % % end
% % %
% % % figure
% % % hold on
% % % scatter(peakinfo(:,1),peakinfo(:,2), 'red')
% % % scatter(peakinfo(:,1),peakinfo(:,3), 'blue')
% % % scatter(peakinfo(:,1),peakinfo(:,4), 'black')
% % % scatter(peakinfo(:,1),peakinfo(:,5), 'green')
% % % scatter(peakinfo(:,1),peakinfo(:,6), 'm')
% % % scatter(peakinfo(:,1),peakinfo(:,7), 'c')
% % % hold off

```

Step5_stripPts

```
% Raw fit data is saved in step 2. The fit routine allows fitting outside
% the domain of input data. We don't want data points outside the domain,
% so we need to delete those. Also, data points that were fit at the limit
% of allowed fit values will be ignored.

% Read in the raw fit data
peakinfo = dlmread(fullfile('Output Data','peakinfoAngleSpace.txt'),'t');

%% Update this
nP = 4; % Assuming the fitting model was a sum of Lorentzians, this is the number of
peaks that were fit
    % This is used to find the columns in 'peakinfo' containing each
    % type of coefficient, since they are grouped in nP number of
    % columns

% Import data
eVgrid = dlmread(fullfile('Output Data','eV.txt'),'t');
angleGrid = dlmread(fullfile('Output Data','angle.txt'),'t');
A = dlmread(fullfile('Output Data','Adata_eV_angle.txt'),'t');

% Reduce axis data to vectors
angle = angleGrid(1,:);
eV = eVgrid(:,1);

% These values are copied and pasted from previous step
angle0 = 47; % Start fit here
i0 = round(interp1(angle,[1:length(angle)],angle0)); % get index value for this location
x = eV;
y = A(:,i0);

init = [0.010 0.045 0.28 0.26    0.068 0.17 0.33 0.63    1.85 1.99 2.24 2.83]; %
Starting fit parameters, tuned to work for i0
upper = [1.000 1.000 1.000 1.000 2.000 2.000 2.000 2.000 1.96 2.15 2.94 3.55];
lower = [0.0001 0.0001 0.001 0.001 0.0001 0.0001 0.001 0.001 min(eV) 1.82 1.92 2.47];
fitRegion = 1:length(angle); % Indices corresponding to angle range of interest
l=length(fitRegion);
fitSequence = [i0:1:fitRegion(end) (i0-1):-1:fitRegion(1)]; % Start with i0, then go up
one index at a time, using the solved for fit parameters as the initial parameters of the
next solution; at the last value, go back to starting value and work back down

%% These are the fitting parameters for reference
% peakinfo(n,1) = fitresult{n,1}{1}.a;
% peakinfo(n,2) = fitresult{n,1}{1}.b;
% peakinfo(n,3) = fitresult{n,1}{1}.c;
% peakinfo(n,4) = fitresult{n,1}{1}.d;
%
%
% peakinfo(n,5) = fitresult{n,1}{1}.g0;
% peakinfo(n,6) = fitresult{n,1}{1}.g1;
% peakinfo(n,7) = fitresult{n,1}{1}.g2;
% peakinfo(n,8) = fitresult{n,1}{1}.g3;
%
%
% peakinfo(n,9) = fitresult{n,1}{1}.x0;
% peakinfo(n,10) = fitresult{n,1}{1}.x1;
% peakinfo(n,11) = fitresult{n,1}{1}.x2;
% peakinfo(n,12) = fitresult{n,1}{1}.x3;
%
%
% peakinfo(n,13) = angle(n);

% These are the additional columns generated later in the script for
% working in k space
% peakinfo(n,14) = kx(1);
% peakinfo(n,15) = kx(2);
% peakinfo(n,16) = kx(3);
```

```

% peakinfo(n,17) = kx(4);

% Get a look at the current peak values
figure
hold on
    scatter(peakinfo(:,13),peakinfo(:,9), 'black')
    scatter(peakinfo(:,13),peakinfo(:,10), 'black')
    scatter(peakinfo(:,13),peakinfo(:,11), 'black')
    scatter(peakinfo(:,13),peakinfo(:,12), 'black')
hold off

%% Strip out fit coefficient where there is no data
for i = fitSequence
    n = i - fitRegion(1)+1;
    y = A(:,i);

    notNaNLoc = find(~isnan(y)); % gives vector containing indices of non-NaN data
    minE = min(x(notNaNLoc));
    maxE = max(x(notNaNLoc));

    for Efits = [(nP*2+1):nP*3] % Corresponds to peak energy location
        if peakinfo(n,Efits) > maxE || peakinfo(n,Efits) < minE
            peakinfo(n,Efits) = NaN;
        else
            end
        end
    end
end

% Check the updated peak values and compare to initial plot
figure
hold on
    scatter(peakinfo(:,13),peakinfo(:,9), 'black')
    scatter(peakinfo(:,13),peakinfo(:,10), 'black')
    scatter(peakinfo(:,13),peakinfo(:,11), 'black')
    scatter(peakinfo(:,13),peakinfo(:,12), 'black')
hold off

% Delete points at bounds. I.e., if the fit routine fit a parameter at the
% lower or upper limit for that parameter, it's probably not a useful data
% point, so delete it
d1 = 6; % These are used to set the precision of the rounding routine when comparing the
fit values to the boundary values. Just a consequence of the imprecision of computer
numbers
d2 = 6;
d3 = 6;
for n = 1:length(peakinfo)

if round(peakinfo(n,1),d1,'significant') == round(lower(1),d1,'significant') ||
round(peakinfo(n,1),d1,'significant') == round(upper(1),d1,'significant')
    peakinfo(n,1) = NaN;
end

if round(peakinfo(n,2),d1,'significant') == round(lower(2),d1,'significant') ||
round(peakinfo(n,2),d1,'significant') == round(upper(2),d1,'significant')
    peakinfo(n,2) = NaN;
end

if round(peakinfo(n,3),d1,'significant') == round(lower(3),d1,'significant') ||
round(peakinfo(n,3),d1,'significant') == round(upper(3),d1,'significant')
    peakinfo(n,3) = NaN;
end

if round(peakinfo(n,4),d1,'significant') == round(lower(4),d1,'significant') ||
round(peakinfo(n,4),d1,'significant') == round(upper(4),d1,'significant')
    peakinfo(n,4) = NaN;
end

if round(peakinfo(n,5),d2,'significant') == round(lower(5),d2,'significant') ||
round(peakinfo(n,5),d2,'significant') == round(upper(5),d2,'significant')
    peakinfo(n,5) = NaN;
end
end

```

```

end

if round(peakinfo(n,6),d2,'significant') == round(lower(6),d2,'significant') ||
round(peakinfo(n,6),d2,'significant') == round(upper(6),d2,'significant')
    peakinfo(n,6) = NaN;
end

if round(peakinfo(n,7),d2,'significant') == round(lower(7),d2,'significant') ||
round(peakinfo(n,7),d2,'significant') == round(upper(7),d2,'significant')
    peakinfo(n,7) = NaN;
end

if round(peakinfo(n,8),d2,'significant') == round(lower(8),d2,'significant') ||
round(peakinfo(n,8),d2,'significant') == round(upper(8),d2,'significant')
    peakinfo(n,8) = NaN;
end

if round(peakinfo(n,9),d3,'significant') == round(lower(9),d3,'significant') ||
round(peakinfo(n,9),d3,'significant') == round(upper(9),d3,'significant')
    peakinfo(n,9) = NaN;
end

if round(peakinfo(n,10),d3,'significant') == round(lower(10),d3,'significant') ||
round(peakinfo(n,10),d3,'significant') == round(upper(10),d3,'significant')
    peakinfo(n,10) = NaN;
end

if round(peakinfo(n,11),d3,'significant') == round(lower(11),d3,'significant') ||
round(peakinfo(n,11),d3,'significant') == round(upper(11),d3,'significant')
    peakinfo(n,11) = NaN;
end

if round(peakinfo(n,12),d3,'significant') == round(lower(12),d3,'significant') ||
round(peakinfo(n,12),d3,'significant') == round(upper(12),d3,'significant')
    peakinfo(n,12) = NaN;
end

end
end

% Check
figure
hold on
    scatter(peakinfo(:,13),peakinfo(:,9), 'black')
    scatter(peakinfo(:,13),peakinfo(:,10), 'black')
    scatter(peakinfo(:,13),peakinfo(:,11), 'black')
    scatter(peakinfo(:,13),peakinfo(:,12), 'black')
hold off

% For any NaN coefficients, set the other corresponding coefficients of that peak to NaN
as well
for n = 1:length(peakinfo)
    if isnan(peakinfo(n,1)) == 1 || isnan(peakinfo(n,5)) == 1 || isnan(peakinfo(n,9)) ==
1
        peakinfo(n,1) = NaN;
        peakinfo(n,5) = NaN;
        peakinfo(n,9) = NaN;
        end

    if isnan(peakinfo(n,2)) == 1 || isnan(peakinfo(n,6)) == 1 || isnan(peakinfo(n,10)) ==
1
        peakinfo(n,2) = NaN;
        peakinfo(n,6) = NaN;
        peakinfo(n,10) = NaN;
        end

    if isnan(peakinfo(n,3)) == 1 || isnan(peakinfo(n,7)) == 1 || isnan(peakinfo(n, 11))
== 1
        peakinfo(n,3) = NaN;
        peakinfo(n,7) = NaN;
        peakinfo(n,11) = NaN;
        end
end

```

```

    if isnan(peakinfo(n,4)) == 1 || isnan(peakinfo(n,8)) == 1 || isnan(peakinfo(n, 12))
== 1
        peakinfo(n,4) = NaN;
        peakinfo(n,8) = NaN;
        peakinfo(n,12) = NaN;
    end
end

% Check again
figure
hold on
    scatter(peakinfo(:,13),peakinfo(:,1), 'black')
    scatter(peakinfo(:,13),peakinfo(:,2), 'black')
    scatter(peakinfo(:,13),peakinfo(:,3), 'black')
    scatter(peakinfo(:,13),peakinfo(:,4), 'black')
hold off

figure
hold on
    scatter(peakinfo(:,13),peakinfo(:,5), 'black')
    scatter(peakinfo(:,13),peakinfo(:,6), 'black')
    scatter(peakinfo(:,13),peakinfo(:,7), 'black')
    scatter(peakinfo(:,13),peakinfo(:,8), 'black')
hold off

figure
hold on
scatter(peakinfo(:,10),peakinfo(:,1).*peakinfo(:,5), 'black')
scatter(peakinfo(:,10),peakinfo(:,2).*peakinfo(:,6), 'red')
scatter(peakinfo(:,10),peakinfo(:,3).*peakinfo(:,7), 'blue')
scatter(peakinfo(:,10),peakinfo(:,4).*peakinfo(:,8), 'blue')
hold off

figure
hold on
    scatter(peakinfo(:,13),peakinfo(:,9), 'black')
    scatter(peakinfo(:,13),peakinfo(:,10), 'black')
    scatter(peakinfo(:,13),peakinfo(:,11), 'black')
    scatter(peakinfo(:,13),peakinfo(:,12), 'black')
hold off

% % For very small peaks, set value equal to NaN
% ignoreFactor = 0.0001; % Peaks less than 1% of max peak, ignore
% for n = 1:length(peakinfo)
%
%     peakArea(n,1) = peakinfo(n,1)*peakinfo(n,5);
%     peakArea(n,2) = peakinfo(n,2)*peakinfo(n,6);
%     peakArea(n,3) = peakinfo(n,3)*peakinfo(n,7);
%     peakArea(n,4) = peakinfo(n,4)*peakinfo(n,8);
%
%
%
%     maxPeakArea(n) = max(peakArea(n,1:nP)); % Don't include the extra peak used for
fitting (this was previously set to nP-1?)
%
%     for k = 1:nP
%         if peakArea(n,k) < ignoreFactor*maxPeakArea(n)
%             peakinfo(n,k) = NaN;
%         else
%             end
%     end
% end
%
% % For any NaN coefficients, set the other corresponding coefficients of that peak to
NaN as well
% for n = 1:length(peakinfo)
%     if isnan(peakinfo(n,1)) == 1 || isnan(peakinfo(n,5)) == 1 || isnan(peakinfo(n,9))
== 1
%         peakinfo(n,1) = NaN;
%         peakinfo(n,5) = NaN;
%         peakinfo(n,9) = NaN;
%     end
%
%

```

```

%     if isnan(peakinfo(n,2)) == 1 || isnan(peakinfo(n,6)) == 1 || isnan(peakinfo(n,10))
== 1
%         peakinfo(n,2) = NaN;
%         peakinfo(n,6) = NaN;
%         peakinfo(n,10) = NaN;
%     end
%
%     if isnan(peakinfo(n,3)) == 1 || isnan(peakinfo(n,7)) == 1 || isnan(peakinfo(n, 11))
== 1
%         peakinfo(n,3) = NaN;
%         peakinfo(n,7) = NaN;
%         peakinfo(n,11) = NaN;
%     end
%
%     if isnan(peakinfo(n,4)) == 1 || isnan(peakinfo(n,8)) == 1 || isnan(peakinfo(n, 12))
== 1
%         peakinfo(n,4) = NaN;
%         peakinfo(n,8) = NaN;
%         peakinfo(n,12) = NaN;
%     end
% end
%
% % Check again
% figure
% hold on
%     scatter(peakinfo(:,13),peakinfo(:,1), 'black')
%     scatter(peakinfo(:,13),peakinfo(:,2), 'red')
%     scatter(peakinfo(:,13),peakinfo(:,3), 'blue')
%     scatter(peakinfo(:,13),peakinfo(:,4), 'green')
% hold off
%
% figure
% hold on
%     scatter(peakinfo(:,13),peakinfo(:,5), 'black')
%     scatter(peakinfo(:,13),peakinfo(:,6), 'red')
%     scatter(peakinfo(:,13),peakinfo(:,7), 'blue')
%     scatter(peakinfo(:,13),peakinfo(:,8), 'green')
% hold off
%
% figure
% hold on
% scatter(peakinfo(:,13),peakinfo(:,1).*peakinfo(:,5), 'black')
% scatter(peakinfo(:,13),peakinfo(:,2).*peakinfo(:,6), 'red')
% scatter(peakinfo(:,13),peakinfo(:,3).*peakinfo(:,7), 'blue')
% scatter(peakinfo(:,13),peakinfo(:,4).*peakinfo(:,8), 'green')
% hold off
%
% figure
% hold on
%     scatter(peakinfo(:,13),peakinfo(:,9), 'black')
%     scatter(peakinfo(:,13),peakinfo(:,10), 'red')
%     scatter(peakinfo(:,13),peakinfo(:,11), 'blue')
%     scatter(peakinfo(:,13),peakinfo(:,12), 'green')
% hold off

%% Finally, delete any remaining points that don't make sense
minAngle4 = 44.5; % (deg) data at angles below 44.5 is garbage
minAngle4index = round(interp1(angle,[1:length(angle)],minAngle4)); % get index value for
this location
    peakinfo(1:minAngle4index,4) = NaN;
    peakinfo(1:minAngle4index,8) = NaN;
    peakinfo(1:minAngle4index,12) = NaN;

% Check again
figure
hold on
    scatter(peakinfo(:,13),peakinfo(:,1), 'black')
    scatter(peakinfo(:,13),peakinfo(:,2), 'red')
    scatter(peakinfo(:,13),peakinfo(:,3), 'blue')
    scatter(peakinfo(:,13),peakinfo(:,4), 'green')
hold off

```

```

figure
hold on
    scatter(peakinfo(:,13),peakinfo(:,5), 'black')
    scatter(peakinfo(:,13),peakinfo(:,6), 'red')
    scatter(peakinfo(:,13),peakinfo(:,7), 'blue')
    scatter(peakinfo(:,13),peakinfo(:,8), 'green')
hold off

figure
hold on
scatter(peakinfo(:,13),peakinfo(:,1).*peakinfo(:,5), 'black')
scatter(peakinfo(:,13),peakinfo(:,2).*peakinfo(:,6), 'red')
scatter(peakinfo(:,13),peakinfo(:,3).*peakinfo(:,7), 'blue')
scatter(peakinfo(:,13),peakinfo(:,4).*peakinfo(:,8), 'green')
hold off

figure
hold on
    scatter(peakinfo(:,13),peakinfo(:,9), 'black')
    scatter(peakinfo(:,13),peakinfo(:,10), 'red')
    scatter(peakinfo(:,13),peakinfo(:,11), 'blue')
    scatter(peakinfo(:,13),peakinfo(:,12), 'green')
hold off

%% Overlay it on dispersion
EA = 1240/666;
EB = 1240/617;
EC = 1240/487;
minAngle = min(angle);
maxAngle = max(angle);

Plot = figure;
    hold on
        colorPlot = pcolor(angleGrid, eVgrid, A); % Plot the dispersion
        colorPlot.EdgeColor = 'none';
        colormap('hot')
        c=colorbar;
        caxis([0 inf])
        c.Label.String = 'Absorptivity';
        ax = gca;
        ax.TickDir = 'out';
        ax.Layer = 'top';
        ax.Box = 'on';
        ax.XMinorTick = 'on';
        ax.YMinorTick = 'on';

        % overlay the fit peak values
        scatter(peakinfo(:,13),peakinfo(:,9), 'black')
        scatter(peakinfo(:,13),peakinfo(:,10), 'black')
        scatter(peakinfo(:,13),peakinfo(:,11), 'black')
        scatter(peakinfo(:,13),peakinfo(:,12), 'black')

        % overlay the uncoupled exciton positions
        plot([minAngle maxAngle],[EA EA], '--k')
        plot([minAngle maxAngle],[EB EB], '--k')
        plot([minAngle maxAngle],[EC EC], '--k')
    hold off

% Output the new peak info
dlmwrite(fullfile('Output Data','peakinfoAngleStripped.txt'),peakinfo,'delimiter','\t');

%% It was easier to fit in angle space, but we also want k space, so now
% convert
% These will be the updated parameters: only the angle changes
% peakinfo(n,1) = fitresult{n,1}{1}.a;
% peakinfo(n,2) = fitresult{n,1}{1}.b;
% peakinfo(n,3) = fitresult{n,1}{1}.c;
% peakinfo(n,4) = fitresult{n,1}{1}.d;
%
%
% peakinfo(n,5) = fitresult{n,1}{1}.g0;

```

```

% peakinfo(n,6) = fitresult{n,1}{1}.g1;
% peakinfo(n,7) = fitresult{n,1}{1}.g2;
% peakinfo(n,8) = fitresult{n,1}{1}.g3;
%
%
% peakinfo(n,9) = fitresult{n,1}{1}.x0;
% peakinfo(n,10) = fitresult{n,1}{1}.x1;
% peakinfo(n,11) = fitresult{n,1}{1}.x2;
% peakinfo(n,12) = fitresult{n,1}{1}.x3;
%
%
% peakinfo(n,13) = angle(i);
% peakinfo(n,14) = kx(n,1);
% peakinfo(n,15) = kx(n,2);
% peakinfo(n,16) = kx(n,3);
% peakinfo(n,17) = kx(n,4);

% Inputs include the substrate data
substrateDataWl = dlmread(fullfile('Output Data','substrateData.txt'),'t');
substrateDataEV = [1240./substrateDataWl(:,1) substrateDataWl(:,2)]; % Convert to energy
space

% For each of the branches, the energy and angle are inputs into kx, so
% deal with each branch separately
for i = 1:nP % Number of peaks fitted
    j = (nP*2+i); % Counting the peak energy values
    k = j+nP+1; % Counting kx position for peakinfo

    % Need to calculate substrate index for each set of peak values
    n_substrate =
interpl(substrateDataEV(:,1),substrateDataEV(:,2),peakinfo(:,j),'pchip');

    % Caculate corresponding k value for each E and angle
    kx(:,i) = 2*pi.*n_substrate.*peakinfo(:,j)/1240.*sin(pi/4+asin(sin(pi*angle'/180-
pi/4)./n_substrate));
    peakinfo(:,k) = kx(:,i);
end

%% Overlay it on dispersion
% Import data
kxGrid = dlmread(fullfile('Output Data','kx.txt'),'t');
Akx = dlmread(fullfile('Output Data','Adata_eV_kx.txt'),'t');
EA = 1240/666;
EB = 1240/617;
EC = 1240/487;
minKx = min(min(kxGrid));
maxKx = max(max(kxGrid));

Plot = figure;
hold on
colorPlot = pcolor(kxGrid, eVgrid, Akx); % Plot the dispersion
colorPlot.EdgeColor = 'none';
colormap('hot')
c=colorbar;
caxis([0 inf])
c.Label.String = 'Absorptivity';
ax = gca;
ax.TickDir = 'out';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';

% overlay the fit peak values
scatter(peakinfo(:,14),peakinfo(:,9), 'black')
scatter(peakinfo(:,15),peakinfo(:,10), 'black')
scatter(peakinfo(:,16),peakinfo(:,11), 'black')
scatter(peakinfo(:,17),peakinfo(:,12), 'black')

% overlay the uncoupled exciton positions
plot([minKx maxKx],[EA EA], '--k')
plot([minKx maxKx],[EB EB], '--k')

```

```
    plot([minKx maxKx],[EC EC], '--k')  
hold off  
dlmwrite(fullfile('Output Data','peakinfoKstripped.txt'),peakinfo,'delimiter','\t');
```

Step6_fitBarePlasmon_8nm

```
%% Fit in angle and k space
% 1. Run the RTA Transfer Matrix Calculator to create a model and output
%   modeled dispersion data
% 2. Put the three data files into the 'Input Data' folder and then step
%   through steps below
clear all;
%% Step 1, update file names
% Import data--copy and paste data file name
% k space first
eVgrid = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-ti-palik-
nrelAg-Ag2S-1-film-mos2-on-ag-constant-1.5-lossless-air, ti-palik 1, nrelAg 43, Ag2S
0.27115, 1-film-mos2-on-ag-constant-1.5-lossless 8_11eV.txt'), '\t');
kxGrid = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-ti-palik-
nrelAg-Ag2S-1-film-mos2-on-ag-constant-1.5-lossless-air, ti-palik 1, nrelAg 43, Ag2S
0.27115, 1-film-mos2-on-ag-constant-1.5-lossless 8_11kx.txt'), '\t');
A = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-ti-palik-nrelAg-
Ag2S-1-film-mos2-on-ag-constant-1.5-lossless-air, ti-palik 1, nrelAg 43, Ag2S 0.27115,
1-film-mos2-on-ag-constant-1.5-lossless 8_11Adata_eV_kx.txt'), '\t');

% The kxGrid is not constant along the columns like the angleGrid is, due
% to the transformation from angle space to k-space. So run the following
% function to give an interpolated version of the data with constant k
% columns
interpolateKdispersion(eVgrid, kxGrid, A)
% Then read the data from that output back in
eVInterpGrid = dlmread(fullfile('Output Data','spp_eV_interp.txt'), '\t');
kxInterpGrid = dlmread(fullfile('Output Data','spp_kx_interp.txt'), '\t');
AInterp = dlmread(fullfile('Output Data','spp_Adata_eV_kx_interp.txt'), '\t');
% angle space, eVgrid and A are common to both
angleGrid = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-ti-palik-
nrelAg-Ag2S-1-film-mos2-on-ag-constant-1.5-lossless-air, ti-palik 1, nrelAg 43, Ag2S
0.27115, 1-film-mos2-on-ag-constant-1.5-lossless 8_11angle.txt'), '\t');

% Reduce x- and y-axis data to vectors
eV = eVgrid(:,1);
eVInterp = eVInterpGrid(:,1);
kxInterp = kxInterpGrid(1,:);
angle = angleGrid(1,:);

% Plot the 3 cases
% Check color plot, kx
Plot = figure;
hold on
colorPlot = pcolor(kxGrid, eV, A); % Plot the dispersion
colorPlot.EdgeColor = 'none';
colormap('hot')
c=colorbar;
caxis([0 inf])
c.Label.String = 'Absorptivity';
ax = gca;
ax.TickDir = 'out';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
hold off

% Check color plot, kxinterp
Plot = figure;
hold on
colorPlot = pcolor(kxInterp, eVInterp, AInterp); % Plot the dispersion
colorPlot.EdgeColor = 'none';
colormap('hot')
c=colorbar;
```

```

    caxis([0 inf])
    c.Label.String = 'Absorptivity';
    ax = gca;
    ax.TickDir = 'out';
    ax.Layer = 'top';
    ax.Box = 'on';
    ax.XMinorTick = 'on';
    ax.YMinorTick = 'on';
hold off

% Check color plot, angle
Plot = figure;
hold on
    colorPlot = pcolor(angle, eV, A); % Plot the dispersion
    colorPlot.EdgeColor = 'none';
    colormap('hot')
    c=colorbar;
    caxis([0 inf])
    c.Label.String = 'Absorptivity';
    ax = gca;
    ax.TickDir = 'out';
    ax.Layer = 'top';
    ax.Box = 'on';
    ax.XMinorTick = 'on';
    ax.YMinorTick = 'on';
hold off

%% Step 2, copy and paste these value from script Step2_cutBadData
minWl = 340; % nm
maxWl = 984; % nm

% We're working in eV here, come on!
maxEv = 1240.0/minWl;
minEv = 1240.0/maxWl;

% Find the indices for that range
EvIndex = [1:length(eV)]';
EvInterpIndex = [1:length(eVInterp)]';

% The others should be done this way
% If the desired limits are outside the limits of the data, then finding
% the index will return NaN, so you should just use the first/last index
% For now, I only ran into this problem with minEv
if eV(end) > minEv
    minEvIndex = EvIndex(end);
else
    minEvIndex = round(interp1(eV,EvIndex,minEv));
end
maxEvIndex = round(interp1(eV,EvIndex,maxEv));

if eVInterp(end) > minEv
    minEvInterpIndex = EvInterpIndex(end);
else
    minEvInterpIndex = round(interp1(eVInterp,EvInterpIndex,minEv));
end
maxEvInterpIndex = round(interp1(eVInterp,EvInterpIndex,maxEv));

% cut out annoying data
kxGrid = kxGrid(maxEvIndex:minEvIndex,:);
angleGrid = angleGrid(maxEvIndex:minEvIndex,:);
eV = eV(maxEvIndex:minEvIndex);
A = A(maxEvIndex:minEvIndex,:);

kxInterpGrid = kxInterpGrid(maxEvInterpIndex:minEvInterpIndex,:);
eVInterp = eVInterp(maxEvInterpIndex:minEvInterpIndex);
AInterp = AInterp(maxEvInterpIndex:minEvInterpIndex,:);

% Check color plot, kx
Plot = figure;
hold on
    colorPlot = pcolor(kxInterp, eVInterp, AInterp); % Plot the dispersion
    colorPlot.EdgeColor = 'none';

```

```

        colormap('hot')
        c=colorbar;
        caxis([0 inf])
        c.Label.String = 'Absorptivity';
        ax = gca;
        ax.TickDir = 'out';
        ax.Layer = 'top';
        ax.Box = 'on';
        ax.XMinorTick = 'on';
        ax.YMinorTick = 'on';
    hold off

    % Check color plot, angle
    Plot = figure;
    hold on
        colorPlot = pcolor(angle, eV, A); % Plot the dispersion
        colorPlot.EdgeColor = 'none';
        colormap('hot')
        c=colorbar;
        caxis([0 inf])
        c.Label.String = 'Absorptivity';
        ax = gca;
        ax.TickDir = 'out';
        ax.Layer = 'top';
        ax.Box = 'on';
        ax.XMinorTick = 'on';
        ax.YMinorTick = 'on';
    hold off

%% Step 3, run this section to find a region in angle space where all the peaks that need
to be fit can be seen in one absorptivity vs energy plot
% Update angle0 and rerun this section as needed to find such a slice of the data. The
purpose is
% to find fitting parameters and we need well defined peaks to get started
% with initial values for the fit

% Fit will start where there are all features and then fit to the ends
% where some features disappear
angle0 = 47; % Start fit here
k0 = 0.0226; % Start fit here

i0 = round(interp1(angle,[1:length(angle)],angle0)); % get index value for this location
i0Interp = round(interp1(kxInterp,[1:length(kxInterp)],k0)); % get index value for this
location
figure
plot(eV, A(:,i0)); % Check that this is a good starting point
figure
plot(eVInterp, AInterp(:,i0Interp)); % Check that this is a good starting point

% In k-space, hard to fit high-k data where there is no plasmon
% Note that I discovered this by trying the fit routine below and looking
% where it failed in peakinfo
kcutoff = 0.0255; % Delete data beyond this
kcutoffIndex = round(interp1(kxInterp,[1:length(kxInterp)],kcutoff)); % get index value
for this location
kxInterp(kcutoffIndex:end)=[]; % Remove k data beyond this point
AInterp(:, kcutoffIndex:end) = []; % Remove A data beyond this point
% Same thing happens on low k side
kcutoff = 0.0066155;
kcutoffIndex = round(interp1(kxInterp,[1:length(kxInterp)],kcutoff)); % get index value
for this location
kxInterp(1:kcutoffIndex)=[]; % Remove k data before this point
AInterp(:, 1:kcutoffIndex) = []; % Remove A data before this point
% Check color plot, kx
    Plot = figure;
    hold on
        colorPlot = pcolor(kxInterp, eVInterp, AInterp); % Plot the dispersion
        colorPlot.EdgeColor = 'none';
        colormap('hot')

```

```

c=colorbar;
caxis([0 inf])
c.Label.String = 'Absorptivity';
ax = gca;
ax.TickDir = 'out';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
hold off

%% Step 4, Now open the curve fitting tool, load x vs y, and check that the init, etc
values below work to fit the data
% Update the values below for whatever gives a good fit in the tool
% Angle space
x = eV;
y = A(:,i0);
% k-space
xInterp = eVInterp;
yInterp = AInterp(:,i0Interp);

% Update these values from curve fitting tool
% Angle space
% This is the model to fit to: (a*g0/((x-x0)^2+(g0/2)^2)+b*g1/((x-
x1)^2+(g1/2)^2))/(2*pi)+m*x
% It can also be updated to remove or add peaks to get a good fit. Be sure
% to update the number of init, upper, and lower parameters accordingly.
% Represents a sum of Lorentzians with a sloping line passing through E=0
% [a b g0 g1 m x0 x1 ]
init = [0.26 0.08 0.26 0.26 0.05 2.69 3.6 ]; % Starting fit parameters, tuned
to work for angle(i0)
upper = [5 5 3 5 1 5 10 ];
lower = [0.001 0.0001 0.001 0.001 0.0001 0.001 3.45];

% k-space, same model
initInterp = [0.17 0.19 0.14 0.36 0.0001 3.37 3.62 ]; % Starting fit
parameters, tuned to work for angle(i0)
upperInterp = [5 5 3 5 1 5 10 ];
lowerInterp = [0.001 0.0001 0.001 0.001 0.0001 0.001 3.45];

%% Step 3, try to fit the absorptivity vs energy plots at every angle
% The script runFitRoutine should be updated for whichever model was used
% in the fitting app
% Also, the peakinfo, coefficients, and init2 at the end of the loop need
% to be updated based on how many fit parameters there are in the model
% At this stage, you could clear all and run the whole script
fitRegion = 1:length(angle); % Indices corresponding to angle range of interest
l=length(fitRegion);
fitSequence = [i0:1:fitRegion(end) (i0-1):-1:fitRegion(1)]; % Start with i0, then go up
one index at a time, using the solved for fit parameters as the initial parameters of the
next solution; at the last value, go back to starting value and work back down

fitresult = cell(1,1); % Initialize cell to store fit results
gof = cell(1,1); % Initialize to store goodness of fit results
init2 = init;

for i = fitSequence
% n=i - totalFitRegion(1)+1;
n = i - fitRegion(1)+1;
y = A(:,i);

% Note that script runFitRoutine also needs to be updated for current
% data--not very general yet, code not well developed at all
[fitresult{n,1}, gof{n,1}] = runFitRoutine(x, y, init2, upper, lower);
% [a b g0 g1 m x0 x1 ]
peakinfo(n,1) = fitresult{n,1}{1}.a;
peakinfo(n,2) = fitresult{n,1}{1}.b;

peakinfo(n,3) = fitresult{n,1}{1}.g0;
peakinfo(n,4) = fitresult{n,1}{1}.g1;

peakinfo(n,5) = fitresult{n,1}{1}.m;

```

```

peakinfo(n,6) = fitresult{n,1}{1}.x0;
peakinfo(n,7) = fitresult{n,1}{1}.x1;

peakinfo(n,8) = angle(i);

    % Coefficients
    a = fitresult{n,1}{1}.a;
    b = fitresult{n,1}{1}.b;

    g0 = fitresult{n,1}{1}.g0;
    g1 = fitresult{n,1}{1}.g1;

    m = fitresult{n,1}{1}.m;

    x0 = fitresult{n,1}{1}.x0;
    x1 = fitresult{n,1}{1}.x1;

% Update initial coefficients
init2 = [a    b    g0    g1    m    x0    x1    ];
if i==(i0-1)
    init2 = init;
end
end

%% Step 5, Plot it to check that it looks OK
% These are the uncoupled exciton energies of interest, update as
% necessary
EA = 1240/666;
EB = 1240/617;
EC = 1240/487;
minAngle = min(angle);
maxAngle = max(angle);

Plot = figure;
hold on
colorPlot = pcolor(angle, eV, A); % Plot the dispersion
colorPlot.EdgeColor = 'none';
colormap('hot')
c=colorbar;
caxis([0 inf])
c.Label.String = 'Absorptivity';
ax = gca;
ax.TickDir = 'out';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';

% overlay the fit peak values
scatter(peakinfo(:,8),peakinfo(:,6), 'black')
% scatter(peakinfo(:,8),peakinfo(:,7), 'green')

% overlay the uncoupled exciton positions
plot([minAngle maxAngle],[EA EA], '--k')
plot([minAngle maxAngle],[EB EB], '--k')
plot([minAngle maxAngle],[EC EC], '--k')
hold off

% Output raw peak fits to data file
dlmwrite(fullfile('Output Data','peakinfoAngleSpace-8nm.txt'),peakinfo,'delimiter','\t');
% peakinfo = dlmread(fullfile('Output Data','peakinfoAngleSpace-8nm.txt'),'\t');
peakinfoAngle = peakinfo;

% I should have checked for bad data points
% Plot again, with 2nd peak, then delete bad points by hand
Plot = figure;
hold on
colorPlot = pcolor(angle, eV, A); % Plot the dispersion
colorPlot.EdgeColor = 'none';
colormap('hot')

```

```

c=colorbar;
caxis([0 inf])
c.Label.String = 'Absorptivity';
ax = gca;
ax.TickDir = 'out';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';

% overlay the fit peak values
scatter(peakinfoAngle(:,8),peakinfoAngle(:,6), 'black')
hold off

% % Weird outlier at 46.9 deg
% peakinfoAngle(50, :) = [];
% 1st 3 data points miss plasmon
peakinfoAngle(1:4,:) = [];

% Plot again to check
Plot = figure;
hold on
colorPlot = pcolor(angle, eV, A); % Plot the dispersion
colorPlot.EdgeColor = 'none';
colormap('hot')
c=colorbar;
caxis([0 inf])
c.Label.String = 'Absorptivity';
ax = gca;
ax.TickDir = 'out';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';

% overlay the fit peak values
scatter(peakinfoAngle(:,8),peakinfoAngle(:,6), 'black')
hold off

% Beautiful. Overwrite the saved data file
dlmwrite(fullfile('Output Data','peakinfoAngleSpace-
8nm.txt'),peakinfoAngle,'delimiter','\t');

%% Step 6, fit for kx
clear fitRegion l fitSequence fitresult gof init2 i n y x upper lower peakinfo init i0
fitRegion = 1:length(kxInterp); % Indices corresponding to angle range of interest
l=length(fitRegion);
fitSequence = [i0Interp:1:fitRegion(end) (i0Interp-1):-1:fitRegion(1)]; % Start with
i0Interp, then go up one index at a time, using the solved for fit parameters as the
initial parameters of the next solution; at the last value, go back to starting value and
work back down

fitresult = cell(1,1); % Initialize cell to store fit results
gof = cell(1,1); % Initialize to store goodness of fit results
init = initInterp;
init2 = initInterp;
upper = upperInterp;
lower = lowerInterp;
i0 = i0Interp;

x = eVInterp;

for i = fitSequence
% n=i - totalFitRegion(1)+1;
n = i - fitRegion(1)+1;
y = AInterp(:,i);

% Note that script runFitRoutine also needs to be updated for current

```

```

    % data--not very general yet, code not well developed at all
    [fitresult{n,1}, gof{n,1}] = runFitRoutine(x, y, init2, upper, lower);
    % [a   b   g0   g1   m   x0   x1 ]
    peakinfo(n,1) = fitresult{n,1}{1}.a;
    peakinfo(n,2) = fitresult{n,1}{1}.b;

    peakinfo(n,3) = fitresult{n,1}{1}.g0;
    peakinfo(n,4) = fitresult{n,1}{1}.g1;

    peakinfo(n,5) = fitresult{n,1}{1}.m;

    peakinfo(n,6) = fitresult{n,1}{1}.x0;
    peakinfo(n,7) = fitresult{n,1}{1}.x1;

peakinfo(n,8) = kxInterp(i);

    % Coefficients
    a = fitresult{n,1}{1}.a;
    b = fitresult{n,1}{1}.b;

    g0 = fitresult{n,1}{1}.g0;
    g1 = fitresult{n,1}{1}.g1;

    m = fitresult{n,1}{1}.m;

    x0 = fitresult{n,1}{1}.x0;
    x1 = fitresult{n,1}{1}.x1;

% Update initial coefficients
init2 = [a   b   g0   g1   m   x0   x1 ];
if i==(i0-1)
    init2 = init;
end
end

peakinfo = dlmread(fullfile('Output Data', 'peakinfoKxSpace-8nm.txt'), '\t')

%% Step 7, Plot it to check that it looks OK
% These are the uncoupled exciton energies of interest, update as
% necessary
EA = 1240/666;
EB = 1240/617;
EC = 1240/487;
minKx = min(kxInterp);
maxKx = max(kxInterp);

Plot = figure;
hold on
colorPlot = pcolor(kxInterp, eVInterp, AInterp); % Plot the dispersion
colorPlot.EdgeColor = 'none';
colormap('hot')
c=colorbar;
caxis([0 inf])
c.Label.String = 'Absorptivity';
ax = gca;
ax.TickDir = 'out';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';

% overlay the fit peak values
scatter(peakinfo(:,8),peakinfo(:,6), 'black')
% scatter(peakinfo(:,8),peakinfo(:,7), 'green')

% overlay the uncoupled exciton positions
plot([minKx maxKx],[EA EA], '--k')
plot([minKx maxKx],[EB EB], '--k')
plot([minKx maxKx],[EC EC], '--k')
hold off

```

```

% Delete bad points by looking at plot
% Data >= kx = 0.024263 is bad--this is where plasmon runs into light
% line
peakinfo(2075:end,:) = [];
% A weird outlier at kx = 0.022588
%
peakinfo(1879,:) = [];

% Plot again to check, ignore the second peak and just plot x0
Plot = figure;
hold on
colorPlot = pcolor(kxInterp, eVInterp, AInterp); % Plot the dispersion
colorPlot.EdgeColor = 'none';
colormap('hot')
c=colorbar;
caxis([0 inf])
c.Label.String = 'Absorptivity';
ax = gca;
ax.TickDir = 'out';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';

% overlay the fit peak values
scatter(peakinfo(:,8),peakinfo(:,6), 'black')

hold off

% This looks beautiful. Save it.
% Output raw peak fits to data file
dlmwrite(fullfile('Output Data', 'peakinfoKxSpace-8nm.txt'), peakinfo, 'delimiter', '\t');

```

Step7_semiClassicalModel_k

```
%% This file fits the data to a semiclassical harmonic oscillator model to
% extract coupling strengths/Rabi splitting values, in angle and kx space

%% k-space first
% Input data is the positions of the peaks of each of the 4 branches of the
% coupled mode taken from the dispersion color plots, as well as the
% position of the peaks from the simulated bare plasmon with lossless mos2
% position == energy vs kx of the peaks
peakinfoExp = dlmread(fullfile('Output Data','peakinfoKStripped.txt'), '\t');
peakinfoSPP = dlmread(fullfile('Output Data','peakinfoKxSpace-8nm.txt'), '\t');

% Experimental peak data is structured as follows
% peak amplitude
% peakinfo(n,1) = fitresult{n,1}{1}.a;
% peakinfo(n,2) = fitresult{n,1}{1}.b;
% peakinfo(n,3) = fitresult{n,1}{1}.c;
% peakinfo(n,4) = fitresult{n,1}{1}.d;
%
% peak width
% peakinfo(n,5) = fitresult{n,1}{1}.g0;
% peakinfo(n,6) = fitresult{n,1}{1}.g1;
% peakinfo(n,7) = fitresult{n,1}{1}.g2;
% peakinfo(n,8) = fitresult{n,1}{1}.g3;
%
% peak energy
% peakinfo(n,9) = fitresult{n,1}{1}.x0;
% peakinfo(n,10) = fitresult{n,1}{1}.x1;
% peakinfo(n,11) = fitresult{n,1}{1}.x2;
% peakinfo(n,12) = fitresult{n,1}{1}.x3;
%
%
% peakinfo(n,13) = angle(i);
% peakinfo(n,14) = kx(n,1);
% peakinfo(n,15) = kx(n,2);
% peakinfo(n,16) = kx(n,3);
% peakinfo(n,17) = kx(n,4);

% SPP data is structured as follows
% NOTE: only peak 1 is relevant (a, g0, x0); peak 2 was fit to the high
% energy d-band transitions
% peak amplitude
% peakinfo(n,1) = fitresult{n,1}{1}.a;
% peakinfo(n,2) = fitresult{n,1}{1}.b;
% peak width
% peakinfo(n,3) = fitresult{n,1}{1}.g0;
% peakinfo(n,4) = fitresult{n,1}{1}.g1;
% slope of line
% peakinfo(n,5) = fitresult{n,1}{1}.m;
% peak positions
% peakinfo(n,6) = fitresult{n,1}{1}.x0;
% peakinfo(n,7) = fitresult{n,1}{1}.x1;
%
%
% peakinfo(n,8) = kxInterp(i);

% Plot it
figure
hold on
scatter(peakinfoExp(:,14),peakinfoExp(:,9),'k');
scatter(peakinfoExp(:,15),peakinfoExp(:,10),'k');
scatter(peakinfoExp(:,16),peakinfoExp(:,11),'k');
scatter(peakinfoExp(:,17),peakinfoExp(:,12),'k');
```

```

scatter(peakinfoSPP(:,8),peakinfoSPP(:,6),'b');

hold off

% Find the approximate splitting values
% NOTE: k values aren't the same for each branch's peak energy values, so
% need to have common x-axis (kx) to subtract one dataset from another
% Let's find the small deltaK and then interpolate the data on a new k
% vector where the deltaK is 10x smaller than the smallest found in the
% data
% First we need to look at only data that is not NaN
Exp1x = peakinfoExp(:,14);
Exp1E = peakinfoExp(:,9);
Exp1xNonNan = Exp1x(~isnan(Exp1x));
Exp1ENonNan = Exp1E(~isnan(Exp1x));
Exp2x = peakinfoExp(:,15);
Exp2E = peakinfoExp(:,10);
Exp2xNonNan = Exp2x(~isnan(Exp2x));
Exp2ENonNan = Exp2E(~isnan(Exp2x));
Exp3x = peakinfoExp(:,16);
Exp3E = peakinfoExp(:,11);
Exp3xNonNan = Exp3x(~isnan(Exp3x));
Exp3ENonNan = Exp3E(~isnan(Exp3x));
Exp4x = peakinfoExp(:,17);
Exp4E = peakinfoExp(:,12);
Exp4xNonNan = Exp4x(~isnan(Exp4x));
Exp4ENonNan = Exp4E(~isnan(Exp4x));

% There are unique values of x, so remove them
[Exp1xNonNanUnique, index1] = unique(Exp1xNonNan);
Exp1ENonNanUnique = Exp1ENonNan(index1);
[Exp2xNonNanUnique, index2] = unique(Exp2xNonNan);
Exp2ENonNanUnique = Exp2ENonNan(index2);
[Exp3xNonNanUnique, index3] = unique(Exp3xNonNan);
Exp3ENonNanUnique = Exp3ENonNan(index3);
[Exp4xNonNanUnique, index4] = unique(Exp4xNonNan);
Exp4ENonNanUnique = Exp4ENonNan(index4);

% Smallest deltaK divided by 10 to make our grid resolution small
% enough to not affect interpolating the data
dK = (Exp4xNonNanUnique(end) - Exp4xNonNanUnique(1))/10;

kx = (Exp1xNonNanUnique(1)-5*dK):dK:(Exp4xNonNanUnique(end)+5*dK);

E1 = interp1(Exp1xNonNanUnique,Exp1ENonNanUnique, kx);
E2 = interp1(Exp2xNonNanUnique,Exp2ENonNanUnique, kx);
E3 = interp1(Exp3xNonNanUnique,Exp3ENonNanUnique, kx);
E4 = interp1(Exp4xNonNanUnique,Exp4ENonNanUnique, kx);

% Check
figure
hold on
scatter(peakinfoExp(:,14),peakinfoExp(:,9),'k');
scatter(peakinfoExp(:,15),peakinfoExp(:,10),'k');
scatter(peakinfoExp(:,16),peakinfoExp(:,11),'k');
scatter(peakinfoExp(:,17),peakinfoExp(:,12),'k');

scatter(kx, E1, 'red');
scatter(kx, E2, 'red');
scatter(kx, E3, 'red');
scatter(kx, E4, 'red');

hold off

% Repeat for SPP
% First we need to look at only data that is not NaN
sppX = peakinfoSPP(:,8);
sppE = peakinfoSPP(:,6);
sppXNonNan = sppX(~isnan(sppX));
sppENonNan = sppE(~isnan(sppX));
% There are unique values of x, so remove them

```

```

[sppXNonNanUnique, index5] = unique(sppXNonNan);
sppENonNanUnique = sppENonNan(index5);
% Now, interpolate over common kx
ESPP = interp1(sppXNonNanUnique,sppENonNanUnique, kx);

% Check
figure
hold on
scatter(peakinfoExp(:,14),peakinfoExp(:,9),'k');
scatter(peakinfoExp(:,15),peakinfoExp(:,10),'k');
scatter(peakinfoExp(:,16),peakinfoExp(:,11),'k');
scatter(peakinfoExp(:,17),peakinfoExp(:,12),'k');

scatter(kx, E1, 'red');
scatter(kx, E2, 'red');
scatter(kx, E3, 'red');
scatter(kx, E4, 'red');
scatter(kx, ESPP, 'blue');

hold off

% Get the difference between branches
diffA = E2-E1; % splitting at A exciton, vs k
diffB = E3-E2; % etc...
diffC = E4-E3;
%
% diffAnonNaN = diffA(~isnan(diffA));
% diffBnonNaN = diffB(~isnan(diffB));
% diffCnonNaN = diffC(~isnan(diffC));

% Plot difference between curves
figure
hold on
scatter(kx,diffA,'k');
scatter(kx,diffB,'k');
scatter(kx,diffC,'k');
hold off

% Find minimum values of difference curves
% [GA, index6] = min(diffAnonNaN);
% [GB, index7] = min(diffBnonNaN);
% [GC, index8] = min(diffCnonNaN);

[GA, index6] = min(diffA);
[GB, index7] = min(diffB);
[GC, index8] = min(diffC);

kGA = kx(index6);
kGB = kx(index7);
kGC = kx(index8);

figure
hold on
scatter(kx, E1, 'red');
scatter(kx, E2, 'red');
scatter(kx, E3, 'red');
scatter(kx, E4, 'red');
scatter(kx, ESPP, 'blue');
minE = min(E1);
maxE = max(E4);
plot([kGA kGA],[minE maxE],'b');
plot([kGB kGB],[minE maxE],'b');
plot([kGC kGC],[minE maxE],'b');
hold off

% For fit routine, use initial parameters of splitting and exciton
% position as:
% Splitting is from GML and GUM above, i.e. minimum distance between
% branches
% Exciton position is from halfway between energy position of splitting

EA0 = (E1(index6)+E2(index6))/2;

```

```

EBO = (E2(index7)+E3(index7))/2;
ECO = (E3(index8)+E4(index8))/2;

figure
hold on
scatter(kx, E1, 'red');
scatter(kx, E2, 'red');
scatter(kx, E3, 'red');
scatter(kx, E4, 'red');
scatter(kx, ESPP, 'blue');
plot([kGA kGA],[minE maxE],'b');
plot([kGB kGB],[minE maxE],'b');
plot([kGC kGC],[minE maxE],'b');
mink = min(kx);
maxk = max(kx);
plot([mink maxk],[EA0 EA0],'b');
plot([mink maxk],[EB0 EB0],'b');
plot([mink maxk],[ECO ECO],'b');
hold off

% Update based on plot
EB0 = 2.02;
ECO = 2.58;
% Plot again
figure
hold on
scatter(kx, E1, 'red');
scatter(kx, E2, 'red');
scatter(kx, E3, 'red');
scatter(kx, E4, 'red');
scatter(kx, ESPP, 'blue');
plot([kGA kGA],[minE maxE],'b');
plot([kGB kGB],[minE maxE],'b');
plot([kGC kGC],[minE maxE],'b');
mink = min(kx);
maxk = max(kx);
plot([mink maxk],[EA0 EA0],'b');
plot([mink maxk],[EB0 EB0],'b');
plot([mink maxk],[ECO ECO],'b');
hold off

%% Now, take that and solve for eigen values and plot them
% The bare plasmon is one input
Espp = ESPP;
% And here's it's corresponding k
kspp = kx;

% First, there are some NaN values in the SPP curve, let's remove them
SPPNaNs = find(isnan(Espp)); % these are the indices of the NaNs
Espp(SPPNaNs) = []; % Set those indices to null, ie delete them
kspp(SPPNaNs) = []; % Delete them for the k too

%% Fit the data
x0 = [EA0 EB0 ECO GA GB GC];
% can't fit to NaNs, so need to make a new x for each y where the xs have
% no NaNs
% initialize with original k values
% xdata = [peakinfoExp(:,10)';
%         peakinfoExp(:,10)'];
%         peakinfoExp(:,10)'];
xdata1 = kx;
ydata1 = E1;
y1nan = isnan(ydata1);
xdata1(y1nan)=[];
ydata1(y1nan)=[];
% xdata1 = xdata1';
% ydata1 = ydata1';

xdata2 = kx;
ydata2 = E2;
y2nan = isnan(ydata2);
xdata2(y2nan)=[];

```

```

ydata2(y2nan)=[];
%   xdata2 = xdata2';
%   ydata2 = ydata2';

xdata3 = kx;
ydata3 = E3;
y3nan = isnan(ydata3);
xdata3(y3nan)=[];
ydata3(y3nan)=[];
%   xdata3 = xdata3';
%   ydata3 = ydata3';

xdata4 = kx;
ydata4 = E4;
y4nan = isnan(ydata4);
xdata4(y4nan)=[];
ydata4(y4nan)=[];
%   xdata4 = xdata4';
%   ydata4 = ydata4';

xdata = {xdata1; xdata2; xdata3; xdata4};
ydata = {ydata1; ydata2; ydata3; ydata4};
ydataCat = horzcat(ydata{:});

% ydata = [peakinfoExp(:,7)';
%         peakinfoExp(:,8)';
%         peakinfoExp(:,9)'];
% Define the function
% you call fun(x,xdata) and it will call the external function EVofK,
% plugging in the appropriate values
% This function will find the eigenvalues of the matrix using the
% fitting values from x and the bare plasmon, then put those values in
% columns, each column corresponding to a kx
% NOTE: the function EVofK will have to be updated for cases not having 4
% branches
fun = @(x,xdata)EVofK(x,xdata,Espp,kspp);

% The lsqcurvefit fits fun(x,xdata), starting at x0, to the data ydataCat,
% then outputs the best fit data x. Here we used concatenated y data
% because the output eigenvalues from fun have been concatenated, even
% though the xdata hasn't been
x = lsqcurvefit(fun,x0,xdata,ydataCat);

% fullfun does what fun does, but uses a single string of x values, not
% sure if this is necessary
fullfun = @(x,xdata)EVofKunbounded(x,xdata,Espp,kspp);
xdatafull = kspp;

efull = fullfun(x,xdatafull);
efull1 = efull(1,:);
efull2 = efull(2,:);
efull3 = efull(3,:);
efull4 = efull(4,:);

figure
hold on
scatter(xdata1,ydata1,'k');
scatter(xdata2,ydata2,'k');
scatter(xdata3,ydata3,'k');
scatter(xdata4,ydata4,'k');
scatter(horzcat(xdata{:}), fun(x,xdata),'r','Marker','.', 'SizeData',100)
scatter(horzcat(xdata{:}), fun(x0,xdata),'g','Marker','x', 'SizeData',50)
scatter(xdatafull, efull1,'b','Marker','.', 'SizeData',25)
scatter(xdatafull, efull2,'b','Marker','.', 'SizeData',25)
scatter(xdatafull, efull3,'b','Marker','.', 'SizeData',25)
scatter(xdatafull, efull4,'b','Marker','.', 'SizeData',25)
hold off

%% stopped here 11:59 10-9
% Caculate the eigenvectors

[V, D] = EVectorsofKunbounded(x,xdatafull,Espp,kspp);

```

```

figure
hold on
scatter(xdatafull, V(1,1,:).^2, 'r')
scatter(xdatafull, V(2,1,:).^2, 'g')
scatter(xdatafull, V(3,1,:).^2, 'b')
scatter(xdatafull, V(4,1,:).^2, 'k')
hold off

figure
hold on
scatter(xdatafull, V(1,2,:).^2, 'r')
scatter(xdatafull, V(2,2,:).^2, 'g')
scatter(xdatafull, V(3,2,:).^2, 'b')
scatter(xdatafull, V(4,2,:).^2, 'k')
hold off

figure
hold on
scatter(xdatafull, V(1,3,:).^2, 'r')
scatter(xdatafull, V(2,3,:).^2, 'g')
scatter(xdatafull, V(3,3,:).^2, 'b')
scatter(xdatafull, V(4,3,:).^2, 'k')
hold off

figure
hold on
scatter(xdatafull, V(1,4,:).^2, 'r')
scatter(xdatafull, V(2,4,:).^2, 'g')
scatter(xdatafull, V(3,4,:).^2, 'b')
scatter(xdatafull, V(4,4,:).^2, 'k')
hold off
%x0 = [EA0 EB0 EC0 GA GB GC];
dlmwrite(fullfile('Output Data', 'k-space', 'kvalues.txt'), xdatafull, 'delimiter', '\t');
dlmwrite(fullfile('Output Data', 'k-space', 'evector1coeffs.txt'), V(:,1,:), 'delimiter', '\t');
dlmwrite(fullfile('Output Data', 'k-space', 'evector2coeffs.txt'), V(:,2,:), 'delimiter', '\t');
dlmwrite(fullfile('Output Data', 'k-space', 'evector3coeffs.txt'), V(:,3,:), 'delimiter', '\t');
dlmwrite(fullfile('Output Data', 'k-space', 'evector4coeffs.txt'), V(:,4,:), 'delimiter', '\t');
dlmwrite(fullfile('Output Data', 'k-space', 'evalues.txt'), efull, 'delimiter', '\t');
dlmwrite(fullfile('Output Data', 'k-space', 'fitparameters-EA-EB-EC-GA-GB-GC.txt'), x, 'delimiter', '\t');

```

Step8_semiClassicalModel_angle

```
%% This file fits the data to a semiclassical harmonic oscillator model to
% extract coupling strengths/Rabi splitting values, in angle and kx space

%% angle-space
% Input data is the positions of the peaks of each of the 4 branches of the
% coupled mode taken from the dispersion color plots, as well as the
% position of the peaks from the simulated bare plasmon with lossless mos2
% position == energy vs kx of the peaks
peakinfoExp = dlmread(fullfile('Output Data','peakinfoKStripped.txt'), '\t');
peakinfoSPP = dlmread(fullfile('Output Data','peakinfoAngleSpace-8nm.txt'), '\t');

% Experimental peak data is structured as follows
% peak amplitude
% peakinfo(n,1) = fitresult{n,1}{1}.a;
% peakinfo(n,2) = fitresult{n,1}{1}.b;
% peakinfo(n,3) = fitresult{n,1}{1}.c;
% peakinfo(n,4) = fitresult{n,1}{1}.d;
%
% peak width
% peakinfo(n,5) = fitresult{n,1}{1}.g0;
% peakinfo(n,6) = fitresult{n,1}{1}.g1;
% peakinfo(n,7) = fitresult{n,1}{1}.g2;
% peakinfo(n,8) = fitresult{n,1}{1}.g3;
%
% peak energy
% peakinfo(n,9) = fitresult{n,1}{1}.x0;
% peakinfo(n,10) = fitresult{n,1}{1}.x1;
% peakinfo(n,11) = fitresult{n,1}{1}.x2;
% peakinfo(n,12) = fitresult{n,1}{1}.x3;
%
%
% peakinfo(n,13) = angle(i);
% peakinfo(n,14) = kx(n,1);
% peakinfo(n,15) = kx(n,2);
% peakinfo(n,16) = kx(n,3);
% peakinfo(n,17) = kx(n,4);

% SPP data is structured as follows
% NOTE: only peak 1 is relevant (a, g0, x0); peak 2 was fit to the high
% energy d-band transitions
% peak amplitude
% peakinfo(n,1) = fitresult{n,1}{1}.a;
% peakinfo(n,2) = fitresult{n,1}{1}.b;
% peak width
% peakinfo(n,3) = fitresult{n,1}{1}.g0;
% peakinfo(n,4) = fitresult{n,1}{1}.g1;
% slope of line
% peakinfo(n,5) = fitresult{n,1}{1}.m;
% peak positions
% peakinfo(n,6) = fitresult{n,1}{1}.x0;
% peakinfo(n,7) = fitresult{n,1}{1}.x1;
%
%
% peakinfo(n,8) = angle(i);

% Plot it
figure
hold on
scatter(peakinfoExp(:,13),peakinfoExp(:,9),'k');
scatter(peakinfoExp(:,13),peakinfoExp(:,10),'k');
scatter(peakinfoExp(:,13),peakinfoExp(:,11),'k');
scatter(peakinfoExp(:,13),peakinfoExp(:,12),'k');
```

```

scatter(peakinfoSPP(:,8),peakinfoSPP(:,6),'b');

hold off

% Find the approximate splitting values
% NOTE: k values aren't the same for each branch's peak energy values, so
% need to have common x-axis (kx) to subtract one dataset from another
% Let's find the small deltaK and then interpolate the data on a new k
% vector where the deltaK is 10x smaller than the smallest found in the
% data
% First we need to look at only data that is not NaN
Exp1x = peakinfoExp(:,13);
Exp1E = peakinfoExp(:,9);
Exp1xNonNan = Exp1x(~isnan(Exp1x));
Exp1ENonNan = Exp1E(~isnan(Exp1x));
Exp2x = peakinfoExp(:,13);
Exp2E = peakinfoExp(:,10);
Exp2xNonNan = Exp2x(~isnan(Exp2x));
Exp2ENonNan = Exp2E(~isnan(Exp2x));
Exp3x = peakinfoExp(:,13);
Exp3E = peakinfoExp(:,11);
Exp3xNonNan = Exp3x(~isnan(Exp3x));
Exp3ENonNan = Exp3E(~isnan(Exp3x));
Exp4x = peakinfoExp(:,13);
Exp4E = peakinfoExp(:,12);
Exp4xNonNan = Exp4x(~isnan(Exp4x));
Exp4ENonNan = Exp4E(~isnan(Exp4x));

% There are unique values of x, so remove them
[Exp1xNonNanUnique, index1] = unique(Exp1xNonNan);
Exp1ENonNanUnique = Exp1ENonNan(index1);
[Exp2xNonNanUnique, index2] = unique(Exp2xNonNan);
Exp2ENonNanUnique = Exp2ENonNan(index2);
[Exp3xNonNanUnique, index3] = unique(Exp3xNonNan);
Exp3ENonNanUnique = Exp3ENonNan(index3);
[Exp4xNonNanUnique, index4] = unique(Exp4xNonNan);
Exp4ENonNanUnique = Exp4ENonNan(index4);

% Smallest deltaK divided by 10 to make our grid resolution small
% enough to not affect interpolating the data
dAngle = (Exp4xNonNanUnique(end) - Exp4xNonNanUnique(end-1))/10;

angle = (Exp1xNonNanUnique(1)-5*dAngle):dAngle:(Exp4xNonNanUnique(end)+5*dAngle);

E1 = interp1(Exp1xNonNanUnique,Exp1ENonNanUnique, angle);
E2 = interp1(Exp2xNonNanUnique,Exp2ENonNanUnique, angle);
E3 = interp1(Exp3xNonNanUnique,Exp3ENonNanUnique, angle);
E4 = interp1(Exp4xNonNanUnique,Exp4ENonNanUnique, angle);

% E1 = Exp1ENonNanUnique;
% E2 = Exp2ENonNanUnique;
% E3 = Exp3ENonNanUnique;
% E4 = Exp4ENonNanUnique;

% Check
figure
hold on
scatter(peakinfoExp(:,13),peakinfoExp(:,9),'k');
scatter(peakinfoExp(:,13),peakinfoExp(:,10),'k');
scatter(peakinfoExp(:,13),peakinfoExp(:,11),'k');
scatter(peakinfoExp(:,13),peakinfoExp(:,12),'k');

scatter(angle, E1, 'red');
scatter(angle, E2, 'red');
scatter(angle, E3, 'red');
scatter(angle, E4, 'red');

hold off

% Repeat for SPP
% First we need to look at only data that is not NaN

```

```

sppX = peakinfoSPP(:,8);
sppE = peakinfoSPP(:,6);
sppXNonNan = sppX(~isnan(sppX));
sppENonNan = sppE(~isnan(sppX));
% There are unique values of x, so remove them
[sppXNonNanUnique, index5] = unique(sppXNonNan);
sppENonNanUnique = sppENonNan(index5);
% Now, interpolate over common kx
ESPP = interp1(sppXNonNanUnique,sppENonNanUnique, angle);

% Check
figure
hold on
scatter(peakinfoExp(:,13),peakinfoExp(:,9),'k');
scatter(peakinfoExp(:,13),peakinfoExp(:,10),'k');
scatter(peakinfoExp(:,13),peakinfoExp(:,11),'k');
scatter(peakinfoExp(:,13),peakinfoExp(:,12),'k');

scatter(angle, E1, 'red');
scatter(angle, E2, 'red');
scatter(angle, E3, 'red');
scatter(angle, E4, 'red');
scatter(angle, ESPP, 'blue');

hold off

% Get the difference between branches
diffA = E2-E1; % splitting at A exciton, vs k
diffB = E3-E2; % etc...
diffC = E4-E3;
%
% diffAnonNaN = diffA(~isnan(diffA));
% diffBnonNaN = diffB(~isnan(diffB));
% diffCnonNaN = diffC(~isnan(diffC));

% Plot difference between curves
figure
hold on
scatter(angle,diffA,'k');
scatter(angle,diffB,'k');
scatter(angle,diffC,'k');
hold off

% Find minimum values of difference curves
% [GA, index6] = min(diffAnonNaN);
% [GB, index7] = min(diffBnonNaN);
% [GC, index8] = min(diffCnonNaN);

[GA, index6] = min(diffA);
[GB, index7] = min(diffB);
[GC, index8] = min(diffC);

angleGA = angle(index6);
angleGB = angle(index7);
angleGC = angle(index8);

figure
hold on
scatter(angle, E1, 'red');
scatter(angle, E2, 'red');
scatter(angle, E3, 'red');
scatter(angle, E4, 'red');
scatter(angle, ESPP, 'blue');
minE = min(E1);
maxE = max(E4);
plot([angleGA angleGA],[minE maxE],'b');
plot([angleGB angleGB],[minE maxE],'b');
plot([angleGC angleGC],[minE maxE],'b');
hold off

% For fit routine, use initial parameters of splitting and exciton
% position as:

```

```

% Splitting is from GML and GUM above, i.e. minimum distance between
% branches
% Exciton position is from halfway between energy position of splitting

EA0 = (E1(index6)+E2(index6))/2;
EB0 = (E2(index7)+E3(index7))/2;
EC0 = (E3(index8)+E4(index8))/2;

figure
hold on
scatter(angle, E1, 'red');
scatter(angle, E2, 'red');
scatter(angle, E3, 'red');
scatter(angle, E4, 'red');
scatter(angle, ESPP, 'blue');
plot([angleGA angleGA],[minE maxE],'b');
plot([angleGB angleGB],[minE maxE],'b');
plot([angleGC angleGC],[minE maxE],'b');
minangle = min(angle);
maxangle = max(angle);
plot([minangle maxangle],[EA0 EA0],'b');
plot([minangle maxangle],[EB0 EB0],'b');
plot([minangle maxangle],[EC0 EC0],'b');
hold off

% Update based on plot
EB0 = 2.02;
EC0 = 2.58;
% Plot again
figure
hold on
scatter(angle, E1, 'red');
scatter(angle, E2, 'red');
scatter(angle, E3, 'red');
scatter(angle, E4, 'red');
scatter(angle, ESPP, 'blue');
plot([angleGA angleGA],[minE maxE],'b');
plot([angleGB angleGB],[minE maxE],'b');
plot([angleGC angleGC],[minE maxE],'b');
minangle = min(angle);
maxangle = max(angle);
plot([minangle maxangle],[EA0 EA0],'b');
plot([minangle maxangle],[EB0 EB0],'b');
plot([minangle maxangle],[EC0 EC0],'b');
hold off

%% Now, take that and solve for eigen values and plot them
% The bare plasmon is one input
Espp = ESPP;
% And here's it's corresponding k
anglespp = angle;

% First, there are some NaN values in the SPP curve, let's remove them
SPPNaNs = find(isnan(Espp)); % these are the indices of the NaNs
Espp(SPPNaNs) = []; % Set those indices to null, ie delete them
anglespp(SPPNaNs) = []; % Delete them for the k too

%% Fit the data
x0 = [EA0 EB0 EC0 GA GB GC];
% can't fit to NaNs, so need to make a new x for each y where the xs have
% no NaNs
% initialize with original k values
% xdata = [peakinfoExp(:,10)';
%         peakinfoExp(:,10)';
%         peakinfoExp(:,10)'];
xdata1 = angle;
ydata1 = E1;
ylnan = isnan(ydata1);
xdata1(ylnan)=[];
ydata1(ylnan)=[];
% xdata1 = xdata1';
% ydata1 = ydata1';

```

```

xdata2 = angle;
ydata2 = E2;
y2nan = isnan(ydata2);
xdata2(y2nan)=[];
ydata2(y2nan)=[];
%   xdata2 = xdata2';
%   ydata2 = ydata2';

xdata3 = angle;
ydata3 = E3;
y3nan = isnan(ydata3);
xdata3(y3nan)=[];
ydata3(y3nan)=[];
%   xdata3 = xdata3';
%   ydata3 = ydata3';

xdata4 = angle;
ydata4 = E4;
y4nan = isnan(ydata4);
xdata4(y4nan)=[];
ydata4(y4nan)=[];
%   xdata4 = xdata4';
%   ydata4 = ydata4';

xdata = {xdata1; xdata2; xdata3; xdata4};
ydata = {ydata1; ydata2; ydata3; ydata4};
ydataCat = horzcat(ydata{:});

% ydata = [peakinfoExp(:,7)';
%         peakinfoExp(:,8)';
%         peakinfoExp(:,9)'];
% Define the function
% you call fun(x,xdata) and it will call the external function EVofK,
% plugging in the appropriate values
% This function will find the eigenvalues of the matrix using the
% fitting values from x and the bare plasmon, then put those values in
% columns, each column corresponding to a kx
% NOTE: the function EVofK will have to be updated for cases not having 4
% branches
fun = @(x,xdata)EVofK(x,xdata,Espp,anglespp);

% The lsqcurvefit fits fun(x,xdata), starting at x0, to the data ydataCat,
% then outputs the best fit data x. Here we used concatenated y data
% because the output eigenvalues from fun have been concatenated, even
% though the xdata hasn't been
x = lsqcurvefit(fun,x0,xdata,ydataCat);

% fullfun does what fun does, but uses a single string of x values, not
% sure if this is necessary
fullfun = @(x,xdata)EVofKunbounded(x,xdata,Espp,anglespp);
xdatafull = anglespp;

efull = fullfun(x,xdatafull);
efull1 = efull(1,:);
efull2 = efull(2,:);
efull3 = efull(3,:);
efull4 = efull(4,:);

figure
hold on
scatter(xdata1,ydata1,'k');
scatter(xdata2,ydata2,'k');
scatter(xdata3,ydata3,'k');
scatter(xdata4,ydata4,'k');
scatter(horzcat(xdata{:}), fun(x,xdata),'r','Marker','.', 'SizeData',100)
scatter(horzcat(xdata{:}), fun(x0,xdata),'g','Marker','x', 'SizeData',50)
scatter(xdatafull, efull1,'b','Marker','.', 'SizeData',25)
scatter(xdatafull, efull2,'b','Marker','.', 'SizeData',25)
scatter(xdatafull, efull3,'b','Marker','.', 'SizeData',25)
scatter(xdatafull, efull4,'b','Marker','.', 'SizeData',25)
hold off

```

```

%%
% Caculate the eigenvectors

[V, D] = EVectorsofKunbounded(x,xdatafull,Espp,anglespp);

figure
hold on
scatter(xdatafull, V(1,1,:).^2, 'r')
scatter(xdatafull, V(2,1,:).^2, 'g')
scatter(xdatafull, V(3,1,:).^2, 'b')
scatter(xdatafull, V(4,1,:).^2, 'k')
hold off

figure
hold on
scatter(xdatafull, V(1,2,:).^2, 'r')
scatter(xdatafull, V(2,2,:).^2, 'g')
scatter(xdatafull, V(3,2,:).^2, 'b')
scatter(xdatafull, V(4,2,:).^2, 'k')
hold off

figure
hold on
scatter(xdatafull, V(1,3,:).^2, 'r')
scatter(xdatafull, V(2,3,:).^2, 'g')
scatter(xdatafull, V(3,3,:).^2, 'b')
scatter(xdatafull, V(4,3,:).^2, 'k')
hold off

figure
hold on
scatter(xdatafull, V(1,4,:).^2, 'r')
scatter(xdatafull, V(2,4,:).^2, 'g')
scatter(xdatafull, V(3,4,:).^2, 'b')
scatter(xdatafull, V(4,4,:).^2, 'k')
hold off
%x0 = [EA0 EB0 EC0 GA GB GC];
dlmwrite(fullfile('Output Data','angle-
space','anglevalues.txt'),xdatafull,'delimiter','\t');
dlmwrite(fullfile('Output Data','angle-
space','evector1coeffs.txt'),V(:,1:),'delimiter','\t');
dlmwrite(fullfile('Output Data','angle-
space','evector2coeffs.txt'),V(:,2:),'delimiter','\t');
dlmwrite(fullfile('Output Data','angle-
space','evector3coeffs.txt'),V(:,3:),'delimiter','\t');
dlmwrite(fullfile('Output Data','angle-
space','evector4coeffs.txt'),V(:,4:),'delimiter','\t');
dlmwrite(fullfile('Output Data','angle-space','evalues.txt'),efull,'delimiter','\t');
dlmwrite(fullfile('Output Data','angle-space','fitparameters-EA-EB-EC-GA-GB-
GC.txt'),x,'delimiter','\t');

```

Step9_saveDataFigures

```
%% Plot experimental dispersion

% Import data
% Energy info common to k- and angle-space
eVgrid = dlmread(fullfile('Output Data','eV.txt'), '\t');

% angle-space (original data)
angleGrid = dlmread(fullfile('Output Data','angle.txt'), '\t');
Angle = dlmread(fullfile('Output Data','Adata_eV_angle.txt'), '\t');

% k-space (transformed from angle space)
kxGrid = dlmread(fullfile('Output Data','kx.txt'), '\t');
Akx = dlmread(fullfile('Output Data','Adata_eV_kx.txt'), '\t');

%% Setup plots
titlekx = 'expkx';
titleangle = 'expangle';
fontsize = 32;
w = 7; % figure width in inches
h = 5; % figure height in inches

%% Plot k-space data
ZData1 = Akx;
YData1 = eVgrid;
XData1 = kxGrid;
CData1 = Akx;

Plot=figure;

set(Plot, 'Units', 'inches');
set(Plot, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the figure
on the screen just to make it easier to see; resolution is set to 300 dpi later, which
equates to 600 dpi when figure is shrunk to desired size

colormap('hot')

% Create axes
axes1 = axes;
set(axes1, 'Position', [0.125 0.18 0.8 0.76]);
% axes1 = axes; % ('Position', [0.21 0.212942462559223 0.7441772476415 0.75]);
hold(axes1, 'on');

% Create surface
surface('ZData', ZData1, 'YData', YData1, 'XData', XData1, ...
        'AlignVertexCenters', 'on', ...
        'EdgeColor', 'none', ...
        'CData', CData1);

% Create ylabel
ylabel('Photon Energy (\it{eV})');

% Create xlabel
xl = xlabel('k_{//} (\it{nm}^{-1})', 'VerticalAlignment', 'baseline');
set(xl, 'Units', 'normalized');
xl.Position = [0.5000 -0.19];

% Create title
t = title({'Experiment'});
t.Units = 'normalized';
t.Position = [0.189 0.86];

xlim(axes1, [0.0085 0.026]);
ylim(axes1, [1.6 3.65]);
```

```

box(axes1,'on');
c = colorbar;
caxis([0.3 0.6])
c.Label.String = 'Absorptivity';
c.Units = 'normalized';
c.Position = [0.76 0.22 0.06 0.4];
c.Label.Units = 'normalized';
c.Label.Rotation = 0;
c.Label.Position = [0.9 1.23];
c.TickLength = 0.1;
c.LineWidth =1;

set(axes1,'FontSize',fontsize);
set(axes1,'Layer','top');
set(axes1,'LineWidth',1.7);
set(axes1,'TickDir','out');
set(axes1,'TickLength',[0.025 0.05]);
set(axes1,'XMinorTick','on');
set(axes1,'YMinorTick','on');

hold(axes1,'off');

print(fullfile('Figures',titlekx), '-dpng', '-r300')

%% Plot angle-space data
ZData1 = Aangle;
YData1 = eVgrid;
XData1 = angleGrid;
CData1 = Aangle;

Plot=figure;

set(Plot, 'Units', 'inches');
set(Plot, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the figure
on the screen just to make it easier to see; resolution is set to 300 dpi later, which
equates to 600 dpi when figure is shrunk to desired size

colormap('hot')

% Create axes
axes1 = axes;
set(axes1,'Position',[0.125 0.18 0.8 0.76]);
% axes1 = axes; %('Position',[0.21 0.212942462559223 0.7441772476415 0.75]);
hold(axes1,'on');

% Create surface
surface('ZData',ZData1,'YData',YData1,'XData',XData1,...
        'AlignVertexCenters','on',...
        'EdgeColor','none',...
        'CData',CData1);

% Create ylabel
ylabel('Photon Energy (\it{eV})');

% Create xlabel
xl = xlabel('Input angle, \theta_{e} (degrees)','VerticalAlignment','baseline');
set(xl, 'Units', 'normalized');
xl.Position = [0.5000 -0.19];

% Create title
t = title({'Experiment'});
t.Units = 'normalized';
t.Position = [0.189 0.86]; %[0.8565 0.8874];
t.Color = [0.99 0.99 0.99];

xlim(axes1,[42 80]);
ylim(axes1,[1.6 3.65]);

```

```

box(axes1, 'on');
c = colorbar;
caxis([0.3 0.6])
c.Label.String = 'Absorptivity';
c.Units = 'normalized';
c.Position = [0.76 0.22 0.06 0.4];
c.Label.Units = 'normalized';
c.Label.Rotation = 0;
c.Label.Position = [0.9 1.23];
c.TickLength = 0.1;
c.LineWidth = 1;
c.EdgeColor = [0.99 0.99 0.99];

set(axes1, 'FontSize', fontsize);
set(axes1, 'Layer', 'top');
set(axes1, 'LineWidth', 1.7);
set(axes1, 'TickDir', 'out');
set(axes1, 'TickLength', [0.025 0.05]);
set(axes1, 'XMinorTick', 'on');
set(axes1, 'YMinorTick', 'on');

hold(axes1, 'off');

print(fullfile('Figures', titleangle), '-dpng', '-r300')

```

Step10_saveSimFigures

```
%% Plot simulation dispersion

% Import data
%Energy info common to k- and angle-space
eVgrid = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-5-ti-palik-
nrelAg-Ag2S-1-film-mos2-on-ag-air, ti-palik 1, nrelAg 43, Ag2S 0.27115, 1-film-mos2-on-
ag 8_leV.txt'), '\t');

% angle-space (original data)
angleGrid = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-5-ti-
palik-nrelAg-Ag2S-1-film-mos2-on-ag-air, ti-palik 1, nrelAg 43, Ag2S 0.27115, 1-film-
mos2-on-ag 8_langle.txt'), '\t');
Aangle = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-5-ti-palik-
nrelAg-Ag2S-1-film-mos2-on-ag-air, ti-palik 1, nrelAg 43, Ag2S 0.27115, 1-film-mos2-on-
ag 8_lAdata_eV_angle.txt'), '\t');

% k-space (transformed from angle space)
kxGrid = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-5-ti-palik-
nrelAg-Ag2S-1-film-mos2-on-ag-air, ti-palik 1, nrelAg 43, Ag2S 0.27115, 1-film-mos2-on-
ag 8_2kx.txt'), '\t');
Akx = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-5-ti-palik-
nrelAg-Ag2S-1-film-mos2-on-ag-air, ti-palik 1, nrelAg 43, Ag2S 0.27115, 1-film-mos2-on-
ag 8_2Adata_eV_kx.txt'), '\t');

%% Setup plots
titlekx = 'simkx';
titleangle = 'simangle';
fontsize = 32;
w = 7; % figure width in inches
h = 5; % figure height in inches

%% Plot k-space data
ZData1 = Akx;
YData1 = eVgrid;
XData1 = kxGrid;
CData1 = Akx;

Plot = figure;

set(Plot, 'Units', 'inches');
set(Plot, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the figure
on the screen just to make it easier to see; resolution is set to 300 dpi later, which
equates to 600 dpi when figure is shrunk to desired size

colormap('hot')

% Create axes
axes1 = axes;
set(axes1, 'Position', [0.125 0.18 0.8 0.76]);

hold(axes1, 'on');

% Create surface
surface('ZData', ZData1, 'YData', YData1, 'XData', XData1, ...
'AlignVertexCenters', 'on', ...
'EdgeColor', 'none', ...
'CData', CData1);

% Create ylabel
ylabel('Photon Energy (\it{eV})');

% Create xlabel
xl = xlabel('k_{//} (\it{nm}^{-1})', 'VerticalAlignment', 'baseline');
set(xl, 'Units', 'normalized');
```

```

xl.Position = [0.5000 -0.19];

% Create title
t = title({'Simulation'});
t.Units = 'normalized';
t.Position = [0.189 0.86];

xlim(axes1,[0.0085 0.026]);
ylim(axes1,[1.6 3.65]);

box(axes1,'on');
c = colorbar;
caxis([0.3 0.6])
c.Label.String = 'Absorptivity';
c.Units = 'normalized';
c.Position = [0.76 0.22 0.06 0.4];
c.Label.Units = 'normalized';
c.Label.Rotation = 0;
c.Label.Position = [0.9 1.23];
c.TickLength = 0.1;
c.LineWidth =1;

set(axes1,'FontSize',fontsize);
set(axes1,'Layer','top');
set(axes1,'LineWidth',1.7);
set(axes1,'TickDir','out');
set(axes1,'TickLength',[0.025 0.05]);
set(axes1,'XMinorTick','on');
set(axes1,'YMinorTick','on');

hold(axes1,'off');

print(fullfile('Figures',titlekx), '-dpng', '-r300')

%% Plot angle-space data
ZData1 = Aangle;
YData1 = eVgrid;
XData1 = angleGrid;
CData1 = Aangle;

Plot=figure;

set(Plot, 'Units', 'inches');
set(Plot, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the figure
on the screen just to make it easier to see; resolution is set to 300 dpi later, which
equates to 600 dpi when figure is shrunk to desired size

colormap('hot')

% Create axes
axes1 = axes;
set(axes1,'Position',[0.125 0.18 0.8 0.76]);

hold(axes1,'on');

% Create surface
surface('ZData',ZData1,'YData',YData1,'XData',XData1,...
'AlignVertexCenters','on',...
'EdgeColor','none',...
'CData',CData1);

% Create ylabel
ylabel('Photon Energy (\it{eV})');

% Create xlabel
xl = xlabel('Input angle, \theta_{e} (degrees)','VerticalAlignment','baseline');
set(xl, 'Units', 'normalized');

```

```

xl.Position = [0.5000 -0.19];

% Create title
t = title({'Simulation'});
t.Units = 'normalized';
t.Position = [0.8565 0.8874];
t.Color = [0.99 0.99 0.99];

xlim(axes1,[42 80]);
ylim(axes1,[1.6 3.65]);

box(axes1,'on');
c = colorbar;
caxis([0.3 0.6])
c.Label.String = 'Absorptivity';
c.Units = 'normalized';
c.Position = [0.76 0.22 0.06 0.4];
c.Label.Units = 'normalized';
c.Label.Rotation = 0;
c.Label.Position = [0.9 1.23];
c.TickLength = 0.1;
c.LineWidth =1;
c.EdgeColor = [0.99 0.99 0.99];

set(axes1,'FontSize',fontsize);
set(axes1,'Layer','top');
set(axes1,'LineWidth',1.7);
set(axes1,'TickDir','out');
set(axes1,'TickLength',[0.025 0.05]);
set(axes1,'XMinorTick','on');
set(axes1,'YMinorTick','on');

hold(axes1,'off');

print(fullfile('Figures',titleangle), '-dpng', '-r300')

```

Step11_plotExpSimKx

```
%%% Plot experimental vs simulation dispersion in k-space

%% Import data
% Experimental
eVExp = dlmread(fullfile('Output Data','eV.txt'), '\t');
kxExp = dlmread(fullfile('Output Data','kx.txt'), '\t');
Aexp = dlmread(fullfile('Output Data','Adata_eV_kx.txt'), '\t');
% Simulation
eVsim = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-5-ti-palik-
nrelAg-Ag2S-1-film-mos2-on-ag-air, ti-palik 1, nrelAg 43, Ag2S 0.27115, 1-film-mos2-on-
ag 8_1eV.txt'), '\t');
kxSim = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-5-ti-palik-
nrelAg-Ag2S-1-film-mos2-on-ag-air, ti-palik 1, nrelAg 43, Ag2S 0.27115, 1-film-mos2-on-
ag 8_2kx.txt'), '\t');
Asim = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-5-ti-palik-
nrelAg-Ag2S-1-film-mos2-on-ag-air, ti-palik 1, nrelAg 43, Ag2S 0.27115, 1-film-mos2-on-
ag 8_2Adata_eV_kx.txt'), '\t');

%% Setup plots
titlekx = 'expsimkx'; % name for saving file
fontsize = 32;
w = 7; % figure width in inches
h = 5; % figure height in inches

%% Plot experimental data
ZData1 = Aexp;
YData1 = eVExp;
XData1 = kxExp;
CData1 = Aexp;

Plot=figure;

set(Plot, 'Units', 'inches');
set(Plot, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the figure
on the screen just to make it easier to see; resolution is set to 300 dpi later, which
equates to 600 dpi when figure is shrunk to desired size

subplot(1,2,1)

colormap('hot')

% Create axes
axes1 = gca;
set(axes1, 'Position', [0.125 0.18 0.375 0.76]);
% axes1 = axes; %('Position', [0.21 0.212942462559223 0.7441772476415 0.75]);
hold(axes1, 'on');

% Create surface
surface('ZData', ZData1, 'YData', YData1, 'XData', XData1, ...
        'AlignVertexCenters', 'on', ...
        'EdgeColor', 'none', ...
        'CData', CData1);

% Create ylabel
ylabel('Photon Energy (\it{eV})');

% Create xlabel
xl = xlabel('k_{//} (\it{nm}^{-1})', 'VerticalAlignment', 'baseline');
set(xl, 'Units', 'normalized');
xl.Position = [0.5000 -0.19];

% Create title
t = title({'Experiment'});
```

```

t.Units = 'normalized';
t.Position = [0.300 0.897];

xlim(axes1,[0.0085 0.026]);
ylim(axes1,[1.6 3.65]);

box(axes1,'on');
c = colorbar;
caxis([0.3 0.6])
c.Label.String = 'Absorptivity';
c.Units = 'normalized';
c.Position = [0.3612 0.2228 0.0500 0.25000];
c.Label.Units = 'normalized';
c.Label.Rotation = 0;
c.Label.Position = [0.9 1.23];
c.TickLength = 0.1;
c.LineWidth = 1;
c.FontSize = 25;

set(axes1,'FontSize',fontsize);
set(axes1,'Layer','top');
set(axes1,'LineWidth',1.7);
set(axes1,'TickDir','out');
set(axes1,'TickLength',[0.025 0.05]);
set(axes1,'XMinorTick','on');
set(axes1,'YMinorTick','on');
tl = get(axes1,'XTickLabel');
set(axes1,'XTickMode','manual');

tl(end) = {' '};
set(axes1,'XTickLabel',tl);

% hold(axes1,'off');
%
% print(fullfile('Figures',titlekx), '-dpng', '-r300')

%% Plot angle-space data
ZData1 = Asim;
YData1 = eVsim;
XData1 = kxSim;
CData1 = Asim;

ss = subplot(1,2,2);
% Plot=figure;
%
% set(Plot, 'Units', 'inches');
% set(Plot, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the figure
on the screen just to make it easier to see; resolution is set to 300 dpi later, which
equates to 600 dpi when figure is shrunk to desired size

colormap('hot')

% Create axes
axes2 = gca;
set(axes2,'Position',[0.5 0.18 0.375 0.76]); % 0.125 0.18 0.375 0.76
% axes2 = axes; %('Position',[0.21 0.212942462559223 0.7441772476415 0.75]);
hold(axes2,'on');

% Create surface
surface('ZData',ZData1,'YData',YData1,'XData',XData1,...
'AlignVertexCenters','on',...
'EdgeColor','none',...
'CData',CData1);

% Create ylabel
% ylabel('Photon Energy (\it{eV})');

```

```

% Create xlabel
xl = xlabel('k_{//} (\it{nm}^{-1})', 'VerticalAlignment', 'baseline');
set(xl, 'Units', 'normalized');
xl.Position = [0.5000 -0.19];

% Create title
t = title({'Simulation'});
t.Units = 'normalized';
t.Position = [0.30 0.897];
t.Color = 'k';

xlim(axes2,[0.0085 0.026]);
ylim(axes2,[1.6 3.65]);

box(axes2,'on');
c = colorbar;
caxis([0.3 0.6])
c.Label.String = 'Absorptivity';
c.Units = 'normalized';
c.Position=[0.74 0.2228 0.0500 0.25000];% [0.76 0.22 0.06 0.4];
c.Label.Units = 'normalized';
c.Label.Rotation = 0;
c.Label.Position = [0.9 1.23];
c.TickLength = 0.1;
c.LineWidth =1;
c.EdgeColor = 'k';
c.FontSize = 25;

set(axes2, 'FontSize', fontsize);
set(axes2, 'Layer', 'top');
set(axes2, 'LineWidth', 1.7);
set(axes2, 'TickDir', 'out');
set(axes2, 'TickLength', [0.025 0.05]);
set(axes2, 'XMinorTick', 'on');
set(axes2, 'YMinorTick', 'on');
set(axes2, 'YAxisLocation', 'right');
hold(axes2, 'off');

axes
axes3 = gca;
set(axes3, 'Position', get(axes2, 'Position'), 'TickDir', 'both', 'Box', 'off',
'color', 'none', 'xtick', []);
set(axes3, 'YTick', get(axes2, 'YTick'));
set(axes3, 'TickLength', [0.025 0.1]);
set(axes3, 'YLim', get(axes2, 'YLim'));
set(axes3, 'YTickLabel', []);
set(axes3, 'LineWidth', get(axes2, 'LineWidth'));
set(axes3, 'YMinorTick', 'on');

print(fullfile('Figures', titlekx), '-dpng', '-r300')

```

Step12_plotHOkx

```
%% Plot experimental modes, coupled oscillator modes, and bare plasmon
%% Import data
peakinfoExp = dlmread(fullfile('Output Data','peakinfoKStripped.txt'),'t');
peakinfoSPP = dlmread(fullfile('Output Data','peakinfoKxSpace-8nm.txt'),'t');
modelY = dlmread(fullfile('Output Data','k-space','values.txt'),'t');
modelY = modelY';
modelX = dlmread(fullfile('Output Data','k-space','kvalues.txt'),'t');
modelFitParams = dlmread(fullfile('Output Data','k-space','fitparameters-EA-EB-EC-GA-GB-
GC.txt'),'t');

% experimental x-values
% peakinfo(n,14) = kx(n,1);
% peakinfo(n,15) = kx(n,2);
% peakinfo(n,16) = kx(n,3);
% peakinfo(n,17) = kx(n,4);
% experimental y-values
% peakinfo(n,9) = fitresult{n,1}{1}.x0;
% peakinfo(n,10) = fitresult{n,1}{1}.x1;
% peakinfo(n,11) = fitresult{n,1}{1}.x2;
% peakinfo(n,12) = fitresult{n,1}{1}.x3;

% bare spp x values
% peakinfo(n,8) = kxInterp(i);
% bare spp y values
% peakinfo(n,6) = fitresult{n,1}{1}.x0;

%% Setup plots
titlekx = 'hoFitKx'; % name for saving file
fontsize = 32;
w = 7; % figure width in inches
h = 5; % figure height in inches
dataLabel = {'Experiment','Bare plasmon','Model'};
range = [0.0085 0.024 1.6 3.4];

% Inputs
EA = modelFitParams(1);
EB = modelFitParams(2);
EC = modelFitParams(3);
kspp = peakinfoSPP(:,8);
Espp = peakinfoSPP(:,6);
kexp1 = peakinfoExp(:,14)';
kexp2 = peakinfoExp(:,15)';
kexp3 = peakinfoExp(:,16)';
kexp4 = peakinfoExp(:,17)';
Eexp1 = peakinfoExp(:,9)';
Eexp2 = peakinfoExp(:,10)';
Eexp3 = peakinfoExp(:,11)';
Eexp4 = peakinfoExp(:,12)';
kexp = [kexp1 kexp2 kexp3 kexp4];
Eexp = [Eexp1 Eexp2 Eexp3 Eexp4];

%% Plot
figure1 = figure;
set(figure1, 'Units', 'inches');
set(figure1, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the
figure on the screen just to make it easier to see; resolution is set to 300 dpi later,
which equates to 600 dpi when figure is shrunk to desired size

% Create axes
axes1 = axes('Position',[0.21 0.212942462559223 0.7441772476415 0.75]);
set(axes1,'Position',[0.15 0.2 0.74 0.75]);
hold(axes1,'on');

% Bare plasmon
```

```

sppPlot = plot(kspp,Espp, 'LineWidth',6, 'Color',[0.811764705882353 0.250980392156863
0.141176470588235]);
% set(sppPlot, 'MarkerEdgeColor', 'none')
% set(sppPlot, 'MarkerFaceColor', 'none')
set(sppPlot, 'LineWidth',6);

% Model
modelPlot1 = plot(modelX,modelY(:,1),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot1,'LineWidth',8);
modelPlot2 = plot(modelX,modelY(:,2),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot2,'LineWidth',8);
modelPlot3 = plot(modelX,modelY(:,3),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot3,'LineWidth',8);
modelPlot4 = plot(modelX,modelY(:,4),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot4,'LineWidth',8);

% Experiment
expPlot = scatter(kexp,Eexp, 'MarkerFaceColor',[0 0 0],'MarkerEdgeColor',[0 0
0],'Marker','.');
set(expPlot,'SizeData',500);

% Bare exciton positions
plot([range(1) range(2)],[EA EA],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % '--'
plot([range(1) range(2)],[EB EB],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % ':'
plot([range(1) range(2)],[EC EC],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % ':'

% Set up axes
axis(range)
% title(plotTitle, 'FontSize', font+1)

% Create ylabel
ylab = ylabel('Photon Energy (\it{eV})','Units','normalized','FontSize',37.4);
set(ylab,'Position',[-0.1, 0.5]);

% Create xlabel
xlabel('k_{//} (\it{nm}^{-1})','FontSize',37.4);

ax = gca;
ax.TickDir = 'in';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
ax.TickLength = [0.025 0.05];
ax.LineWidth = 1.7;
ax.FontSize = 32;
%   xticks([0.010 0.015]);
%   yticks(1.6:0.2:2.4);

% Label bare exciton positions
text('VerticalAlignment','top','FontAngle','italic','FontSize',32,...
'String','E_A',...
'Position',[0.02 EA]);

text('VerticalAlignment','bottom','FontAngle','italic','FontSize',32,...
'String','E_B',...
'Position',[0.02 EB]);

text('VerticalAlignment','bottom','FontAngle','italic','FontSize',32,...
'String','E_C',...
'Position',[0.02 EC]);

% Experimental data on legend doesn't show up well
fakePlot = plot([range(1) range(2)],[range(4)+1 range(4)+1],'-','Color',[0 0 0],
'LineWidth', 7);
% set(fakePlot,'Visible','off');

% Create legend
a = legend([fakePlot sppPlot modelPlot1],dataLabel);
% set(a,'location','northwest');
set(a,'box','off');
set(a,'Position', [0.1944 0.7500 0.2183 0.1528]);

```

```
print(fullfile('Figures',titlekx), '-dpng', '-r300')
```

Step13_plotHOkxZoom

```
%% Plot experimental modes, coupled oscillator modes, and bare plasmon
%% Import data
peakinfoExp = dlmread(fullfile('Output Data','peakinfoKStripped.txt'), '\t');
peakinfoSPP = dlmread(fullfile('Output Data','peakinfoKxSpace-8nm.txt'), '\t');
modelY = dlmread(fullfile('Output Data','k-space','evalues.txt'), '\t');
modelY = modelY';
modelX = dlmread(fullfile('Output Data','k-space','kvalues.txt'), '\t');
modelFitParams = dlmread(fullfile('Output Data','k-space','fitparameters-EA-EB-EC-GA-GB-
GC.txt'), '\t');

% experimental x-values
% peakinfo(n,14) = kx(n,1);
% peakinfo(n,15) = kx(n,2);
% peakinfo(n,16) = kx(n,3);
% peakinfo(n,17) = kx(n,4);
% experimental y-values
% peakinfo(n,9) = fitresult{n,1}{1}.x0;
% peakinfo(n,10) = fitresult{n,1}{1}.x1;
% peakinfo(n,11) = fitresult{n,1}{1}.x2;
% peakinfo(n,12) = fitresult{n,1}{1}.x3;

% bare spp x values
% peakinfo(n,8) = kxInterp(i);
% bare spp y values
% peakinfo(n,6) = fitresult{n,1}{1}.x0;

%% Setup plots
titlekx = 'hoFitKxTOCzoom'; % name for saving file
fontsize = 32;
w = 7; % figure width in inches
h = 5; % figure height in inches
dataLabel = {'Experiment', 'Bare plasmon', 'Bare C-exciton', 'Model'};
range = [0.011 0.0185 2.05 3.1];

% Inputs
EA = modelFitParams(1);
EB = modelFitParams(2);
EC = modelFitParams(3);
kspp = peakinfoSPP(:,8);
Espp = peakinfoSPP(:,6);
kexp1 = peakinfoExp(:,14)';
kexp2 = peakinfoExp(:,15)';
kexp3 = peakinfoExp(:,16)';
kexp4 = peakinfoExp(:,17)';
Eexp1 = peakinfoExp(:,9)';
Eexp2 = peakinfoExp(:,10)';
Eexp3 = peakinfoExp(:,11)';
Eexp4 = peakinfoExp(:,12)';
kexp = [kexp1 kexp2 kexp3 kexp4];
Eexp = [Eexp1 Eexp2 Eexp3 Eexp4];

%% Plot
figure1 = figure;
set(figure1, 'Units', 'inches');
set(figure1, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the
figure on the screen just to make it easier to see; resolution is set to 300 dpi later,
which equates to 600 dpi when figure is shrunk to desired size

% Create axes
axes1 = axes('Position',[0.21 0.212942462559223 0.7441772476415 0.75]);
set(axes1, 'Position',[0.15 0.2 0.74 0.75]);
hold(axes1, 'on');

% Bare plasmon
```

```

sppPlot = plot(kspp,Espp, 'LineWidth',6, 'Color',[0.811764705882353 0.250980392156863
0.141176470588235]);
% set(sppPlot, 'MarkerEdgeColor', 'none')
% set(sppPlot, 'MarkerFaceColor', 'none')
set(sppPlot, 'LineWidth',6);

% Bare exciton positions
% plot([range(1) range(2)],[EA EA],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % '--'
% plot([range(1) range(2)],[EB EB],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % ':'
cPlot = plot([range(1) range(2)],[EC EC],'-','Color',[0.5 0.5 0.5], 'LineWidth', 4); %
':.'

% Model
modelPlot1 = plot(modelX,modelY(:,1),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot1,'LineWidth',8);
modelPlot2 = plot(modelX,modelY(:,2),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot2,'LineWidth',8);
modelPlot3 = plot(modelX,modelY(:,3),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot3,'LineWidth',8);
modelPlot4 = plot(modelX,modelY(:,4),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot4,'LineWidth',8);

% Experiment
expPlot = scatter(kexp,Eexp, 'MarkerFaceColor',[0 0 0],'MarkerEdgeColor',[0 0
0],'Marker','.');
set(expPlot,'SizeData',500);

% Set up axes
axis(range)
% title(plotTitle, 'FontSize', font+1)

% Create ylabel
ylabel('Photon Energy (\it{eV})','Units','normalized','FontSize',37.4);

% Create xlabel
xlabel('k_{//} (\it{nm}^{-1})','FontSize',37.4);

ax = gca;
ax.TickDir = 'in';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
ax.TickLength = [0.025 0.05];
ax.LineWidth = 1.7;
ax.FontSize = 32;
xticks([0.012 0.014 0.016 0.018]);
% yticks(1.6:0.2:2.4);

% % Label bare exciton positions
% text('VerticalAlignment','top','FontAngle','italic','FontSize',32,...
% 'String','E_A',...
% 'Position',[0.02 EA]);
%
% text('VerticalAlignment','bottom','FontAngle','italic','FontSize',32,...
% 'String','E_B',...
% 'Position',[0.02 EB]);
%
% text('VerticalAlignment','bottom','FontAngle','italic','FontSize',32,...
% 'String','E_C',...
% 'Position',[0.02 EC]);

% Experimental data on legend doesn't show up well
fakePlot = plot([range(1) range(2)],[range(4)+1 range(4)+1],'-','Color',[0 0 0],
'LineWidth', 7);
% set(fakePlot,'Visible','off');

% Create legend
a = legend([fakePlot sppPlot cPlot modelPlot1],dataLabel);
% set(a,'location','northwest');
set(a,'box','off');

```

```

set(a, 'Position', [0.1944 0.7500 0.2183 0.1528]);
set(a, 'AutoUpdate', 'off');

% Label the Rabi splitting
GC = modelFitParams(6);
% [minValue,closestIndex] = min(abs(Esp-EC));
% kRabi = kspp(closestIndex);
% plot([kspp kspp])

arrow = annotation('doublearrow',[0.4752 0.4752],[0.4861 0.6583]);
set(arrow, 'LineWidth',3);
set(arrow, 'Head1Width',40);
set(arrow, 'Head1Length',20);
set(arrow, 'Head2Width',40);
set(arrow, 'Head2Length',20);

removeLine = annotation('textbox',[0.1984 0.5694 0.2659 0.0120],'String','');
removeLine.BackgroundColor = [0.99999 0.99999 0.99999];
removeLine.LineStyle = 'none';

space = ' ';
% om = ['\Omega = ',space];
rabi = ['\Omega = ',space, num2str(GC*1000,3), ' meV'];
% rabi = strcat(om, num2str(GC*1000,3), ' meV');
omega = annotation('textbox',[0.1984 0.5444 0.2719 0.0653],'String',rabi);
omega.LineStyle = 'none';
omega.FontSize = 44;
omega.FontAngle = 'italic';
% omega.FontWeight = 'bold';
omega.FitBoxToText = 'off';
% omega.BackgroundColor = [0.99999 0.99999 0.99999];

print(fullfile('Figures',titlekx), '-dpng', '-r300')

```

Step14_plotCoeffs

```
%% Plot coupled modes' weights
%% Import data
k = dlmread(fullfile('Output Data','k-space','kvalues.txt'),'t');
b1 = dlmread(fullfile('Output Data','k-space','evector1coeffs.txt'),'t');
b2 = dlmread(fullfile('Output Data','k-space','evector2coeffs.txt'),'t');
b3 = dlmread(fullfile('Output Data','k-space','evector3coeffs.txt'),'t');
b4 = dlmread(fullfile('Output Data','k-space','evector4coeffs.txt'),'t');

%% Setup plots
titlekx = 'coeffs'; % name for saving file
fontsize = 32;
w = 7; % figure width in inches
h = 6; % figure height in inches
dataLabel = {'Plasmon', 'A-exciton', 'B-exciton', 'C-exciton'};
range = [0.0085 0.018 -0.1 1.1];
font = 32;
colorP = [0.812 0.251 0.141];
color1 = [0.949 0.678 0.161];
color2 = [0.412 0.490 0.561];
color3 = [211 186 163]./256;

%% Plot
figure1 = figure;
set(figure1, 'Units', 'inches');
set(figure1, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the
figure on the screen just to make it easier to see; resolution is set to 300 dpi later,
which equates to 600 dpi when figure is shrunk to desired size

% Create axes
axes1 = axes('Position',[0.21 0.212942462559223 0.7441772476415 0.75],'FontSize', font);
set(axes1,'Position',[0.15 0.15 0.74 0.8]);
hold(axes1,'on');
pos = axes1.Position;

yLabel = '|c_n|^2';
xLabel = 'k_{//} (\it{nm}^{-1})';
% xlabel(xLabel, 'FontSize', font,'VerticalAlignment','top');

% axes('FontSize', font) ;

% Plot branch 4
h(1) = subplot(4,1,1);
subplot(h(1));
hold on
plot(k,b4(:,1).^2,'Color',colorP, 'LineWidth', 4);
plot(k,b4(:,2).^2,'Color',color1, 'LineWidth', 4);
plot(k,b4(:,3).^2,'Color',color2,'LineWidth', 4);
plot(k,b4(:,4).^2,'Color',color3,'LineWidth', 4);

set(h(1), 'FontSize', font)
axis(range);
ax = gca;
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
ax.TickLength = [0.015 0.05];
ax.LineWidth = 1.7;
text(0.016, 0.5, 'Branch 4', 'FontSize', font);
legend(dataLabel,'location', 'west', 'FontSize', font-6, 'Box', 'off');
ax.XTickLabel = {};
ax.Position = [pos(1) pos(2)+pos(4)/4*3 pos(3) pos(4)/4];
pos = ax.Position;

hold off
```

```

% Plot branch 3
h(2) = subplot(4,1,2);
subplot(h(2));
hold on
plot(k,b3(:,1).^2,'Color',colorP, 'LineWidth', 4);
plot(k,b3(:,2).^2,'Color',color1, 'LineWidth', 4);
plot(k,b3(:,3).^2,'Color',color2, 'LineWidth', 4);
plot(k,b3(:,4).^2,'Color',color3, 'LineWidth', 4);

set(h(2), 'FontSize', font)
axis(range);
ax = gca;
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
ax.TickLength = [0.015 0.05];
ax.LineWidth = 1.7;
text(0.016, 0.5, 'Branch 3', 'FontSize', font);
ax.XTickLabel = {};
ypos = pos(2)-pos(4);
ax.Position = [pos(1) ypos pos(3) pos(4)];
pos = ax.Position;
ylab = ylabel(yLabel, 'FontSize', font);
ylp = get(ylab, 'Position');
set(ylab, 'Position', [ypl(1) 0]);

hold off

% Plot branch 2
h(3) = subplot(4,1,3);
subplot(h(3));
hold on
plot(k,b2(:,1).^2,'Color',colorP, 'LineWidth', 4);
plot(k,b2(:,2).^2,'Color',color1, 'LineWidth', 4);
plot(k,b2(:,3).^2,'Color',color2, 'LineWidth', 4);
plot(k,b2(:,4).^2,'Color',color3, 'LineWidth', 4);

set(h(3), 'FontSize', font)
axis(range);
ax = gca;
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
ax.TickLength = [0.015 0.05];
ax.LineWidth = 1.7;
text(0.016, 0.5, 'Branch 2', 'FontSize', font);
ax.XTickLabel = {};
ypos = pos(2)-pos(4);
ax.Position = [pos(1) ypos pos(3) pos(4)];
pos = ax.Position;
hold off

% Plot branch 1
h(4) = subplot(4,1,4);
subplot(h(4));
hold on
plot(k,b1(:,1).^2,'Color',colorP, 'LineWidth', 4);
plot(k,b1(:,2).^2,'Color',color1, 'LineWidth', 4);
plot(k,b1(:,3).^2,'Color',color2, 'LineWidth', 4);
plot(k,b1(:,4).^2,'Color',color3, 'LineWidth', 4);

set(h(4), 'FontSize', font)
axis(range);
ax = gca;
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
xticks = ax.XTickLabel;
xticks{end} = ' ';

```

```
        ax.XTickLabel = xticks;
ax.TickLength = [0.015 0.05];
ax.LineWidth = 1.7;
    text(0.016, 0.5, 'Branch 1', 'FontSize', font);
        ypos = pos(2)-pos(4);
ax.Position = [pos(1) ypos pos(3) pos(4)];
hold off

xlabel(xLabel, 'FontSize', font, 'VerticalAlignment', 'top');

hold off

print(fullfile('Figures',titlekx), '-dpng', '-r300')
```

Step15_plotExpSimAngle

```
%% Plot experimental vs simulation dispersion in k-space

%% Import data
% Experimental
eVexp = dlmread(fullfile('Output Data','eV.txt'), '\t');
angleExp = dlmread(fullfile('Output Data','angle.txt'), '\t');
Aexp = dlmread(fullfile('Output Data','Adata_eV_angle.txt'), '\t');
% Simulation
eVsim = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-5-ti-palik-
nrelAg-Ag2S-1-film-mos2-on-ag-air, ti-palik 1, nrelAg 43, Ag2S 0.27115, 1-film-mos2-on-
ag 8_leV.txt'), '\t');
angleSim = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-5-ti-
palik-nrelAg-Ag2S-1-film-mos2-on-ag-air, ti-palik 1, nrelAg 43, Ag2S 0.27115, 1-film-
mos2-on-ag 8_langle.txt'), '\t');
Asim = dlmread(fullfile('Input Data','Dispersion with TM-polarization NBK7-5-ti-palik-
nrelAg-Ag2S-1-film-mos2-on-ag-air, ti-palik 1, nrelAg 43, Ag2S 0.27115, 1-film-mos2-on-
ag 8_lAdata_eV_angle.txt'), '\t');

%% Setup plots
titleangle = 'expsimangle'; % name for saving file
fontsize = 32;
w = 7; % figure width in inches
h = 5; % figure height in inches
anglerange = [42 80];

%% Plot experimental data
ZData1 = Aexp;
YData1 = eVexp;
XData1 = angleExp;
CData1 = Aexp;

Plot=figure;

set(Plot, 'Units', 'inches');
set(Plot, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the figure
on the screen just to make it easier to see; resolution is set to 300 dpi later, which
equates to 600 dpi when figure is shrunk to desired size

subplot(1,2,1)

colormap('hot')

% Create axes
axes1 = gca;
set(axes1, 'Position', [0.125 0.18 0.375 0.76]);
% axes1 = axes; %('Position', [0.21 0.212942462559223 0.7441772476415 0.75]);
hold(axes1, 'on');

% Create surface
surface('ZData', ZData1, 'YData', YData1, 'XData', XData1, ...
'AlignVertexCenters', 'on', ...
'EdgeColor', 'none', ...
'CData', CData1);

% Create ylabel
ylabel('Photon Energy (\it{eV})');

% Create xlabel
xl = xlabel('Input angle, \theta_{e} (degrees)', 'VerticalAlignment', 'baseline');
set(xl, 'Units', 'normalized');
% xl.Position = [0.5000 -0.19];
xl.Position = [1 -0.19];
```

```

% Create title
t = title({'Experiment'});
t.Units = 'normalized';
t.Position = [0.300 0.897];
t.Color = [0.99 0.99 0.99];

xlim(axes1,anglerange);
ylim(axes1,[1.6 3.65]);

box(axes1,'on');
c = colorbar;
caxis([0.3 0.6])
c.Label.String = 'Absorptivity';
c.Units = 'normalized';
c.Position = [0.3612 0.2228 0.0500 0.25000];
c.Label.Units = 'normalized';
c.Label.Rotation = 0;
c.Label.Position = [0.9 1.23];
c.TickLength = 0.1;
c.LineWidth =1;
c.FontSize = 25;
c.EdgeColor = [0.99 0.99 0.99];

set(axes1,'FontSize',fontsize);
set(axes1,'Layer','top');
set(axes1,'LineWidth',1.7);
set(axes1,'TickDir','out');
set(axes1,'TickLength',[0.025 0.05]);
set(axes1,'XMinorTick','on');
set(axes1,'YMinorTick','on');
tl = get(axes1,'XTickLabel');
set(axes1,'XTickMode','manual');

tl(end) = {' '};
set(axes1,'XTickLabel',tl);

% hold(axes1,'off');
%
% print(fullfile('Figures',titlekx), '-dpng', '-r300')

%% Plot sim data
ZData1 = Asim;
YData1 = eVsim;
XData1 = angleSim;
CData1 = Asim;

ss = subplot(1,2,2);
% Plot=figure;
%
% set(Plot, 'Units', 'inches');
% set(Plot, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the figure
on the screen just to make it easier to see; resolution is set to 300 dpi later, which
equates to 600 dpi when figure is shrunk to desired size

colormap('hot')

% Create axes
axes2 = gca;
set(axes2,'Position',[0.5 0.18 0.375 0.76]); % 0.125 0.18 0.375 0.76
% axes2 = axes; %('Position',[0.21 0.212942462559223 0.7441772476415 0.75]);
hold(axes2,'on');

% Create surface
surface('ZData',ZData1,'YData',YData1,'XData',XData1,...
'AlignVertexCenters','on',...
'EdgeColor','none',...)

```

```

    'CData',CData1);

% Create ylabel
% ylabel('Photon Energy (\it{eV})');

% Create xlabel
% xl = xlabel('Input angle, \theta_{e} (degrees)', 'VerticalAlignment', 'baseline');
% set(xl, 'Units', 'normalized');
% xl.Position = [0.5000 -0.19];

% Create title
t = title({'Simulation'});
t.Units = 'normalized';
t.Position = [0.30 0.897];
t.Color = [0.99 0.99 0.99];

    xlim(axes2,anglerange);
    ylim(axes2,[1.6 3.65]);

box(axes2,'on');
c = colorbar;
caxis([0.3 0.6])
c.Label.String = 'Absorptivity';
c.Units = 'normalized';
c.Position = [0.74 0.2228 0.0500 0.25000];% [0.76 0.22 0.06 0.4];
c.Label.Units = 'normalized';
c.Label.Rotation = 0;
c.Label.Position = [0.9 1.23];
c.TickLength = 0.1;
c.LineWidth = 1;
c.FontSize = 25;
c.EdgeColor = [0.99 0.99 0.99];

set(axes2, 'FontSize', fontsize);
set(axes2, 'Layer', 'top');
set(axes2, 'LineWidth', 1.7);
set(axes2, 'TickDir', 'out');
set(axes2, 'TickLength', [0.025 0.05]);
set(axes2, 'XMinorTick', 'on');
set(axes2, 'YMinorTick', 'on');
set(axes2, 'YAxisLocation', 'right');
hold(axes2, 'off');

axes
axes3 = gca;
set(axes3, 'Position', get(axes2, 'Position'), 'TickDir', 'both', 'Box', 'off',
'color', 'none', 'xtick', []);
set(axes3, 'YTick', get(axes2, 'YTick'));
set(axes3, 'TickLength', [0.025 0.1]);
set(axes3, 'YLim', get(axes2, 'YLim'));
set(axes3, 'YTickLabel', []);
set(axes3, 'LineWidth', get(axes2, 'LineWidth'));
set(axes3, 'YMinorTick', 'on');
set(axes3, 'YColor', [0.99 0.99 0.99]);

print(fullfile('Figures',titleangle), '-dpng', '-r300')

```

Step16_plotHOangle

```
%% Plot experimental modes, coupled oscillator modes, and bare plasmon, angle space
%% Import data
peakinfoExp = dlmread(fullfile('Output Data','peakinfoAngleStripped.txt'), '\t');
peakinfoSPP = dlmread(fullfile('Output Data','peakinfoAngleSpace-8nm.txt'), '\t');
modelY = dlmread(fullfile('Output Data','angle-space','evalues.txt'), '\t');
modelY = modelY';
modelX = dlmread(fullfile('Output Data','angle-space','anglevalues.txt'), '\t');
modelFitParams = dlmread(fullfile('Output Data','angle-space','fitparameters-EA-EB-EC-GA-
GB-GC.txt'), '\t');

% experimental x-values
% peakinfo(n,13) = angle(i);
% experimental y-values
% peakinfo(n,9) = fitresult{n,1}{1}.x0;
% peakinfo(n,10) = fitresult{n,1}{1}.x1;
% peakinfo(n,11) = fitresult{n,1}{1}.x2;
% peakinfo(n,12) = fitresult{n,1}{1}.x3;

% bare spp x values
% peakinfo(n,8) = angle(i);
% bare spp y values
% peakinfo(n,6) = fitresult{n,1}{1}.x0;

%% Setup plots
titleangle = 'hoFitAngle'; % name for saving file
fontsize = 32;
w = 7; % figure width in inches
h = 5; % figure height in inches
dataLabel = {'Experiment', 'Bare plasmon', 'Model'};
range = [42 70 1.6 3.4];

% Inputs
EA = modelFitParams(1);
EB = modelFitParams(2);
EC = modelFitParams(3);
anglespp = peakinfoSPP(:,8);
Espp = peakinfoSPP(:,6);
angleexp1 = peakinfoExp(:,13)';
angleexp2 = peakinfoExp(:,13)';
angleexp3 = peakinfoExp(:,13)';
angleexp4 = peakinfoExp(:,13)';
Eexp1 = peakinfoExp(:,9)';
Eexp2 = peakinfoExp(:,10)';
Eexp3 = peakinfoExp(:,11)';
Eexp4 = peakinfoExp(:,12)';
angleexp = [angleexp1 angleexp2 angleexp3 angleexp4];
Eexp = [Eexp1 Eexp2 Eexp3 Eexp4];

%% Plot
figure1 = figure;
set(figure1, 'Units', 'inches');
set(figure1, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the
figure on the screen just to make it easier to see; resolution is set to 300 dpi later,
which equates to 600 dpi when figure is shrunk to desired size

% Create axes
axes1 = axes('Position',[0.21 0.212942462559223 0.7441772476415 0.75]);
set(axes1, 'Position',[0.15 0.2 0.74 0.75]);
hold(axes1, 'on');

% Bare plasmon
sppPlot = plot(anglespp, Espp, 'LineWidth', 6, 'Color', [0.811764705882353 0.250980392156863
0.141176470588235]);
% set(sppPlot, 'MarkerEdgeColor', 'none')
% set(sppPlot, 'MarkerFaceColor', 'none')
```

```

set(sppPlot, 'LineWidth',6);

% Model
modelPlot1 = plot(modelX,modelY(:,1),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot1,'LineWidth',8);
modelPlot2 = plot(modelX,modelY(:,2),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot2,'LineWidth',8);
modelPlot3 = plot(modelX,modelY(:,3),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot3,'LineWidth',8);
modelPlot4 = plot(modelX,modelY(:,4),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot4,'LineWidth',8);

% Experiment
expPlot = scatter(angleexp,Eexp, 'MarkerFaceColor',[0 0 0],'MarkerEdgeColor',[0 0
0],'Marker','.');
set(expPlot,'SizeData',500);

% Bare exciton positions
plot([range(1) range(2)],[EA EA],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % '--'
plot([range(1) range(2)],[EB EB],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % ':'
plot([range(1) range(2)],[EC EC],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % ':'

% Set up axes
axis(range)
% title(plotTitle, 'FontSize', font+1)

% Create ylabel
ylab = ylabel('Photon Energy (\it{eV})','Units','normalized','FontSize',37.4);
set(ylab,'Position',[-0.1, 0.5]);

% Create xlabel
xl = xlabel('Input angle, \theta_{e} (degrees)','FontSize',37.4);

ax = gca;
ax.TickDir = 'in';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
ax.TickLength = [0.025 0.05];
ax.LineWidth = 1.7;
ax.FontSize = 32;
%   xticks([0.010 0.015]);
%   yticks(1.6:0.2:2.4);

% Label bare exciton positions
text('VerticalAlignment','top','FontAngle','italic','FontSize',32,...
'String','E_A',...
'Position',[range(2)-5 EA]);

text('VerticalAlignment','bottom','FontAngle','italic','FontSize',32,...
'String','E_B',...
'Position',[range(2)-5 EB]);

text('VerticalAlignment','bottom','FontAngle','italic','FontSize',32,...
'String','E_C',...
'Position',[range(2)-5 EC]);

% Experimental data on legend doesn't show up well
fakePlot = plot([range(1) range(2)],[range(4)+1 range(4)+1],'-','Color',[0 0 0],
'LineWidth', 7);
% set(fakePlot,'Visible','off');

% Create legend
a = legend([fakePlot sppPlot modelPlot1],dataLabel);
% set(a,'location','northwest');
set(a,'box','off');
set(a,'Position', [0.627931746031745,0.718055555555556,0.2183,0.1528]);

print(fullfile('Figures',titleangle), '-dpng', '-r300')

```


Step17_plotHOangleRabi

```
%% Plot experimental modes, coupled oscillator modes, and bare plasmon, angle space
%% Import data
peakinfoExp = dlmread(fullfile('Output Data','peakinfoAngleStripped.txt'), '\t');
peakinfoSPP = dlmread(fullfile('Output Data','peakinfoAngleSpace-8nm.txt'), '\t');
modelY = dlmread(fullfile('Output Data','angle-space','evalues.txt'), '\t');
modelY = modelY';
modelX = dlmread(fullfile('Output Data','angle-space','anglevalues.txt'), '\t');
modelFitParams = dlmread(fullfile('Output Data','angle-space','fitparameters-EA-EB-EC-GA-
GB-GC.txt'), '\t');

% experimental x-values
% peakinfo(n,13) = angle(i);
% experimental y-values
% peakinfo(n,9) = fitresult{n,1}{1}.x0;
% peakinfo(n,10) = fitresult{n,1}{1}.x1;
% peakinfo(n,11) = fitresult{n,1}{1}.x2;
% peakinfo(n,12) = fitresult{n,1}{1}.x3;

% bare spp x values
% peakinfo(n,8) = angle(i);
% bare spp y values
% peakinfo(n,6) = fitresult{n,1}{1}.x0;

%% Setup plots
titleangle = 'hoFitAngleZoomRabiFit'; % name for saving file
fontsize = 32;
w = 7; % figure width in inches
h = 5; % figure height in inches
dataLabel = {'Experiment', 'Bare plasmon', 'Bare C-exciton', 'Model'};
range = [42 60 2.1 3.4];

% Inputs
EA = modelFitParams(1);
EB = modelFitParams(2);
EC = modelFitParams(3);
anglespp = peakinfoSPP(:,8);
Espp = peakinfoSPP(:,6);
angleexp1 = peakinfoExp(:,13)';
angleexp2 = peakinfoExp(:,13)';
angleexp3 = peakinfoExp(:,13)';
angleexp4 = peakinfoExp(:,13)';
Eexp1 = peakinfoExp(:,9)';
Eexp2 = peakinfoExp(:,10)';
Eexp3 = peakinfoExp(:,11)';
Eexp4 = peakinfoExp(:,12)';
angleexp = [angleexp1 angleexp2 angleexp3 angleexp4];
Eexp = [Eexp1 Eexp2 Eexp3 Eexp4];

%% Plot
figure1 = figure;
set(figure1, 'Units', 'inches');
set(figure1, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the
figure on the screen just to make it easier to see; resolution is set to 300 dpi later,
which equates to 600 dpi when figure is shrunk to desired size

% Create axes
axes1 = axes('Position',[0.21 0.212942462559223 0.7441772476415 0.75]);
set(axes1, 'Position',[0.15 0.2 0.74 0.75]);
hold(axes1, 'on');

% Bare plasmon
sppPlot = plot(anglespp, Espp, 'LineWidth', 6, 'Color', [0.811764705882353 0.250980392156863
0.141176470588235]);
% set(sppPlot, 'MarkerEdgeColor', 'none')
% set(sppPlot, 'MarkerFaceColor', 'none')
```

```

set(sppPlot, 'LineWidth',6);

% Bare exciton positions
% plot([range(1) range(2)],[EA EA],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % '--'
% plot([range(1) range(2)],[EB EB],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % ':'
cPlot = plot([range(1) range(2)],[EC EC],'-','Color',[0.5 0.5 0.5], 'LineWidth', 4); %
':.'

% Model
modelPlot1 = plot(modelX,modelY(:,1),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot1,'LineWidth',8);
modelPlot2 = plot(modelX,modelY(:,2),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot2,'LineWidth',8);
modelPlot3 = plot(modelX,modelY(:,3),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot3,'LineWidth',8);
modelPlot4 = plot(modelX,modelY(:,4),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot4,'LineWidth',8);

% Experiment
expPlot = scatter(angleexp,Eexp, 'MarkerFaceColor',[0 0 0],'MarkerEdgeColor',[0 0
0],'Marker','.');
set(expPlot,'SizeData',500);

% Set up axes
axis(range)
% title(plotTitle, 'FontSize', font+1)

% Create ylabel
ylabel('Photon Energy (\it{eV})','Units','normalized','FontSize',37.4);

% Create xlabel
xl = xlabel('Input angle, \theta_{e} (degrees)','FontSize',37.4);

ax = gca;
ax.TickDir = 'in';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
ax.TickLength = [0.025 0.05];
ax.LineWidth = 1.7;
ax.FontSize = 32;
%     xticks([0.012 0.014 0.016 0.018]);
%     yticks(1.6:0.2:2.4);

% % Label bare exciton positions
% text('VerticalAlignment','top','FontAngle','italic','FontSize',32,...
%     'String','E_A',...
%     'Position',[0.02 EA]);
%
% text('VerticalAlignment','bottom','FontAngle','italic','FontSize',32,...
%     'String','E_B',...
%     'Position',[0.02 EB]);
%
% text('VerticalAlignment','bottom','FontAngle','italic','FontSize',32,...
%     'String','E_C',...
%     'Position',[0.02 EC]);

% Experimental data on legend doesn't show up well
fakePlot = plot([range(1) range(2)],[range(4)+1 range(4)+1],'-','Color',[0 0 0],
'LineWidth', 7);
% set(fakePlot,'Visible','off');

% Create legend
a = legend([fakePlot sppPlot cPlot modelPlot1],dataLabel, 'Location','northwest');
% set(a,'location','northwest');
set(a,'box','off');
set(a,'Position',
[0.188492063492063,0.717361166130783,0.234126984126984,0.202083333333333]);
set(a, 'AutoUpdate','off');

```

```

% Label the Rabi splitting
GC = modelFitParams(6);
% [minValue,closestIndex] = min(abs(Espp-EC));
% kRabi = kspp(closestIndex);
% plot([kspp kspp])

arrow = annotation('doublearrow',[0.34 0.34],[0.347 0.593]);
set(arrow, 'LineWidth',3);
set(arrow, 'Head1Width',40);
set(arrow, 'Head1Length',20);
set(arrow, 'Head2Width',40);
set(arrow, 'Head2Length',20);

removeLine =
annotation('textbox',[0.357130158730158,0.4638444444444445,0.2659,0.012],'String','');
removeLine.BackgroundColor = [0.99999 0.99999 0.99999];
removeLine.LineStyle = 'none';

space = ' ';
% om = ['\Omega = ',space];
rabi = ['\Omega = ',space, num2str(GC*1000,3), ' meV'];
% rabi = strcat(om, num2str(GC*1000,3), ' meV');
omega =
annotation('textbox',[0.356138095238094,0.445788888888889,0.2719,0.0653],'String',rabi);
omega.LineStyle = 'none';
omega.FontSize = 44;
omega.FontAngle = 'italic';
% omega.FontWeight = 'bold';
omega.FitBoxToText = 'off';
% omega.BackgroundColor = [0.99999 0.99999 0.99999];

print(fullfile('Figures',titleangle), '-dpng', '-r300')

%% Setup plots-show rabi splitting from experimental data
titleangle = 'hoFitAngleZoomRabiData'; % name for saving file
fontsize = 32;
w = 7; % figure width in inches
h = 5; % figure height in inches
dataLabel = {'Experiment','Bare plasmon','Bare C-exciton','Model'};
range = [42 60 2.1 3.4];

% Inputs
EA = modelFitParams(1);
EB = modelFitParams(2);
EC = modelFitParams(3);
anglespp = peakinfoSPP(:,8);
Espp = peakinfoSPP(:,6);
angleexp1 = peakinfoExp(:,13)';
angleexp2 = peakinfoExp(:,13)';
angleexp3 = peakinfoExp(:,13)';
angleexp4 = peakinfoExp(:,13)';
Eexp1 = peakinfoExp(:,9)';
Eexp2 = peakinfoExp(:,10)';
Eexp3 = peakinfoExp(:,11)';
Eexp4 = peakinfoExp(:,12)';
angleexp = [angleexp1 angleexp2 angleexp3 angleexp4];
Eexp = [Eexp1 Eexp2 Eexp3 Eexp4];

%% Plot
figure1 = figure;
set(figure1, 'Units', 'inches');

```

```

set(figure1, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the
figure on the screen just to make it easier to see; resolution is set to 300 dpi later,
which equates to 600 dpi when figure is shrunk to desired size

% Create axes
axes1 = axes('Position',[0.21 0.212942462559223 0.7441772476415 0.75]);
set(axes1,'Position',[0.15 0.2 0.74 0.75]);
hold(axes1,'on');

% Bare plasmon
sppPlot = plot(anglespp,Espp, 'LineWidth',6, 'Color',[0.811764705882353 0.250980392156863
0.141176470588235]);
% set(sppPlot, 'MarkerEdgeColor', 'none')
% set(sppPlot, 'MarkerFaceColor', 'none')
set(sppPlot, 'LineWidth',6);

% Bare exciton positions
% plot([range(1) range(2)],[EA EA],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % '--'
% plot([range(1) range(2)],[EB EB],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % ':'
cPlot = plot([range(1) range(2)],[EC EC],'-','Color',[0.5 0.5 0.5], 'LineWidth', 4); %
';

% Model
modelPlot1 = plot(modelX,modelY(:,1),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot1,'LineWidth',8);
modelPlot2 = plot(modelX,modelY(:,2),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot2,'LineWidth',8);
modelPlot3 = plot(modelX,modelY(:,3),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot3,'LineWidth',8);
modelPlot4 = plot(modelX,modelY(:,4),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot4,'LineWidth',8);

% Experiment
expPlot = scatter(angleexp,Eexp, 'MarkerFaceColor',[0 0 0],'MarkerEdgeColor',[0 0
0],'Marker','.');
set(expPlot,'SizeData',500);

% Set up axes
axis(range)
% title(plotTitle, 'FontSize', font+1)

% Create ylabel
ylabel('Photon Energy (\it{eV})','Units','normalized','FontSize',37.4);

% Create xlabel
xl = xlabel('Input angle, \theta_{e} (degrees)','FontSize',37.4);

ax = gca;
ax.TickDir = 'in';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
ax.TickLength = [0.025 0.05];
ax.LineWidth = 1.7;
ax.FontSize = 32;
% xticks([0.012 0.014 0.016 0.018]);
% yticks(1.6:0.2:2.4);

% % Label bare exciton positions
% text('VerticalAlignment','top','FontAngle','italic','FontSize',32,...
% 'String','E_A',...
% 'Position',[0.02 EA]);
%
% text('VerticalAlignment','bottom','FontAngle','italic','FontSize',32,...
% 'String','E_B',...
% 'Position',[0.02 EB]);
%
% text('VerticalAlignment','bottom','FontAngle','italic','FontSize',32,...
% 'String','E_C',...
% 'Position',[0.02 EC]);

```

```

% Experimental data on legend doesn't show up well
fakePlot = plot([range(1) range(2)],[range(4)+1 range(4)+1], '-','Color',[0 0 0],
'LineWidth', 7);
% set(fakePlot,'Visible','off');

% Create legend
a = legend([fakePlot sppPlot cPlot modelPlot1],dataLabel, 'Location','northwest');
% set(a,'location','northwest');
set(a,'box','off');
set(a,'Position',
[0.188492063492063,0.717361166130783,0.234126984126984,0.202083333333333]);
set(a, 'AutoUpdate','off');

% Label the Rabi splitting
%GC = modelFitParams(6);
[minValue,closestIndex] = min(abs(Espp-EC));
angleRabi = anglespp(closestIndex);
diff = abs(angleexpl'-angleRabi);
[q, rabiIndex] = min(diff);
GCdata = Eexp4(rabiIndex) - Eexp3(rabiIndex);% need to calculate for exp data and fix
arrow

arrow = annotation('doublearrow',[0.34 0.34],[0.267 0.618]);
set(arrow, 'LineWidth',3);
set(arrow, 'Head1Width',40);
set(arrow, 'Head1Length',20);
set(arrow, 'Head2Width',40);
set(arrow, 'Head2Length',20);

removeLine =
annotation('textbox',[0.357130158730158,0.463844444444445,0.2659,0.012],'String','');
removeLine.BackgroundColor = [0.99999 0.99999 0.99999];
removeLine.LineStyle = 'none';

space = ' ';
% om = ['\Omega = ',space];
rabi = ['\Omega = ',space, num2str(GCdata*1000,3), ' meV'];
% rabi = strcat(om, num2str(GC*1000,3), ' meV');
omega =
annotation('textbox',[0.356138095238094,0.445788888888889,0.2719,0.0653],'String',rabi);
omega.LineStyle = 'none';
omega.FontSize = 44;
omega.FontAngle = 'italic';
% omega.FontWeight = 'bold';
omega.FitBoxToText = 'off';
% omega.BackgroundColor = [0.99999 0.99999 0.99999];

print(fullfile('Figures',titleangle), '-dpng', '-r300')

```

Step18_plotCoeffsAngle

```
%% Plot coupled modes' weights, angle space
%% Import data
k = dlmread(fullfile('Output Data','angle-space','anglevalues.txt'),'t');
b1 = dlmread(fullfile('Output Data','angle-space','evector1coeffs.txt'),'t');
b2 = dlmread(fullfile('Output Data','angle-space','evector2coeffs.txt'),'t');
b3 = dlmread(fullfile('Output Data','angle-space','evector3coeffs.txt'),'t');
b4 = dlmread(fullfile('Output Data','angle-space','evector4coeffs.txt'),'t');

%% Setup plots
titleangle = 'coeffsAngle'; % name for saving file
fontsize = 32;
w = 7; % figure width in inches
h = 6; % figure height in inches
dataLabel = {'Plasmon', 'A-exciton', 'B-exciton', 'C-exciton'};
range = [43 50 -0.1 1.1];
font = 32;
colorP = [0.812 0.251 0.141];
color1 = [0.949 0.678 0.161];
color2 = [0.412 0.490 0.561];
color3 = [211 186 163]./256;

%% Plot
figure1 = figure;
set(figure1, 'Units', 'inches');
set(figure1, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the
figure on the screen just to make it easier to see; resolution is set to 300 dpi later,
which equates to 600 dpi when figure is shrunk to desired size

% Create axes
axes1 = axes('Position',[0.21 0.212942462559223 0.7441772476415 0.75],'FontSize', font);
set(axes1,'Position',[0.15 0.15 0.74 0.8]);
hold(axes1,'on');
pos = axes1.Position;

yLabel = '|c_n|^2';
xlabel = 'Input angle, \theta_{e} (degrees)';
% xl = xlabel('Input angle, \theta_{e} (degrees)','FontSize',37.4);

% xlabel(xLabel, 'FontSize', font,'VerticalAlignment','top');

% axes('FontSize', font) ;

% Plot branch 4
h(1) = subplot(4,1,1);
subplot(h(1));
hold on
plot(k,b4(:,1).^2,'Color',colorP, 'LineWidth', 4);
plot(k,b4(:,2).^2,'Color',color1, 'LineWidth', 4);
plot(k,b4(:,3).^2,'Color',color2, 'LineWidth', 4);
plot(k,b4(:,4).^2,'Color',color3, 'LineWidth', 4);

set(h(1), 'FontSize', font)
axis(range);
ax = gca;
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
ax.TickLength = [0.015 0.05];
ax.LineWidth = 1.7;
text(0.016, 0.5, 'Branch 4', 'FontSize', font);
% legend(dataLabel,'location','east','FontSize',font-6,'Box','off');
ax.XTickLabel = {};
ax.Position = [pos(1) pos(2)+pos(4)/4*3 pos(3) pos(4)/4];
pos = ax.Position;
```

```

hold off

% Plot branch 3
h(2) = subplot(4,1,2);
subplot(h(2));
hold on
plot(k,b3(:,1).^2,'Color',colorP, 'LineWidth', 4);
plot(k,b3(:,2).^2,'Color',color1, 'LineWidth', 4);
plot(k,b3(:,3).^2,'Color',color2,'LineWidth', 4);
plot(k,b3(:,4).^2,'Color',color3,'LineWidth', 4);

set(h(2), 'FontSize', font)
axis(range);
ax = gca;
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
ax.TickLength = [0.015 0.05];
ax.LineWidth = 1.7;
text(0.016, 0.5, 'Branch 3', 'FontSize', font);
ax.XTickLabel = {};
ypos = pos(2)-pos(4);
ax.Position = [pos(1) ypos pos(3) pos(4)];
pos = ax.Position;
ylab = ylabel(yLabel, 'FontSize', font);
ylp = get(ylab,'Position');
set(ylab, 'Position', [yyp(1) 0]);

hold off

% Plot branch 2
h(3) = subplot(4,1,3);
subplot(h(3));
hold on
plot(k,b2(:,1).^2,'Color',colorP, 'LineWidth', 4);
plot(k,b2(:,2).^2,'Color',color1, 'LineWidth', 4);
plot(k,b2(:,3).^2,'Color',color2,'LineWidth', 4);
plot(k,b2(:,4).^2,'Color',color3,'LineWidth', 4);

set(h(3), 'FontSize', font)
axis(range);
ax = gca;
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
ax.TickLength = [0.015 0.05];
ax.LineWidth = 1.7;
text(0.016, 0.5, 'Branch 2', 'FontSize', font);
ax.XTickLabel = {};
ypos = pos(2)-pos(4);
ax.Position = [pos(1) ypos pos(3) pos(4)];
pos = ax.Position;
hold off

% Plot branch 1
h(4) = subplot(4,1,4);
subplot(h(4));
hold on
plot(k,b1(:,1).^2,'Color',colorP, 'LineWidth', 4);
plot(k,b1(:,2).^2,'Color',color1, 'LineWidth', 4);
plot(k,b1(:,3).^2,'Color',color2,'LineWidth', 4);
plot(k,b1(:,4).^2,'Color',color3,'LineWidth', 4);

set(h(4), 'FontSize', font)
axis(range);
ax = gca;
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';

```

```

%         xticks = ax.XTickLabel;
%         xticks{end} = ' ';
%         ax.XTickLabel = xticks;
ax.TickLength = [0.015 0.05];
ax.LineWidth = 1.7;
    text(0.016, 0.5, 'Branch 1', 'FontSize', font);
        ypos = pos(2)-pos(4);
ax.Position = [pos(1) ypos pos(3) pos(4)];
    legend(dataLabel,'location', 'east', 'FontSize', font-6, 'Box', 'off');
hold off

    xlabel(xLabel, 'FontSize', font,'VerticalAlignment', 'top');

hold off

print(fullfile('Figures',titleangle), '-dpng', '-r300')

```

Step19_plotHOkxTOC

```
%% i never finished this
% TOC figure for nanoletters should fit within 3.25 x 1.75 inches, 8 point
% font preferable, 6 point minimum, TIFF 300 dpi, Label the graphic ?For Table of
Contents Only? and provide it on the last page of the submitted manuscript.
%

%% Plot experimental modes, coupled oscillator modes, and bare plasmon
%% Import data
peakinfoExp = dlmread(fullfile('Output Data','peakinfoKStripped.txt'), '\t');
peakinfoSPP = dlmread(fullfile('Output Data','peakinfoKxSpace-8nm.txt'), '\t');
modelY = dlmread(fullfile('Output Data','k-space','evalues.txt'), '\t');
modelY = modelY';
modelX = dlmread(fullfile('Output Data','k-space','kvalues.txt'), '\t');
modelFitParams = dlmread(fullfile('Output Data','k-space','fitparameters-EA-EB-EC-GA-GB-
GC.txt'), '\t');

% experimental x-values
% peakinfo(n,14) = kx(n,1);
% peakinfo(n,15) = kx(n,2);
% peakinfo(n,16) = kx(n,3);
% peakinfo(n,17) = kx(n,4);
% experimental y-values
% peakinfo(n,9) = fitresult{n,1}{1}.x0;
% peakinfo(n,10) = fitresult{n,1}{1}.x1;
% peakinfo(n,11) = fitresult{n,1}{1}.x2;
% peakinfo(n,12) = fitresult{n,1}{1}.x3;

% bare spp x values
% peakinfo(n,8) = kxInterp(i);
% bare spp y values
% peakinfo(n,6) = fitresult{n,1}{1}.x0;

%% Setup plots
titlekx = 'hoFitKxTOC'; % name for saving file
fontsize = 32;
w = 7; % figure width in inches
h = 5; % figure height in inches
dataLabel = {'Experiment', 'Bare plasmon', 'Bare C-exciton', 'Model'};
range = [0.011 0.0185 2.05 3.1];

% Inputs
EA = modelFitParams(1);
EB = modelFitParams(2);
EC = modelFitParams(3);
kspp = peakinfoSPP(:,8);
Espp = peakinfoSPP(:,6);
kexp1 = peakinfoExp(:,14)';
kexp2 = peakinfoExp(:,15)';
kexp3 = peakinfoExp(:,16)';
kexp4 = peakinfoExp(:,17)';
Eexp1 = peakinfoExp(:,9)';
Eexp2 = peakinfoExp(:,10)';
Eexp3 = peakinfoExp(:,11)';
Eexp4 = peakinfoExp(:,12)';
kexp = [kexp1 kexp2 kexp3 kexp4];
Eexp = [Eexp1 Eexp2 Eexp3 Eexp4];

%% Plot
figure1 = figure;
set(figure1, 'Units', 'inches');
set(figure1, 'Position', 2*[0 0 w h]); % Note that I double the desired size of the
figure on the screen just to make it easier to see; resolution is set to 300 dpi later,
which equates to 600 dpi when figure is shrunk to desired size

% Create axes
```

```

axes1 = axes('Position',[0.21 0.212942462559223 0.7441772476415 0.75]);
set(axes1,'Position',[0.15 0.2 0.74 0.75]);
hold(axes1,'on');

% Bare plasmon
sppPlot = plot(kspp,Espp, 'LineWidth',6, 'Color',[0.811764705882353 0.250980392156863
0.141176470588235]);
% set(sppPlot, 'MarkerEdgeColor', 'none')
% set(sppPlot, 'MarkerFaceColor', 'none')
set(sppPlot, 'LineWidth',6);

% Bare exciton positions
% plot([range(1) range(2)],[EA EA],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % '--'
% plot([range(1) range(2)],[EB EB],'-','Color',[0.5 0.5 0.5], 'LineWidth', 2); % ':'
cPlot = plot([range(1) range(2)],[EC EC],'-','Color',[0.5 0.5 0.5], 'LineWidth', 4); %
':';

% Model
modelPlot1 = plot(modelX,modelY(:,1),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot1,'LineWidth',8);
modelPlot2 = plot(modelX,modelY(:,2),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot2,'LineWidth',8);
modelPlot3 = plot(modelX,modelY(:,3),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot3,'LineWidth',8);
modelPlot4 = plot(modelX,modelY(:,4),'Color', [0.9290 0.6940 0.1250], 'LineWidth', 8);
set(modelPlot4,'LineWidth',8);

% Experiment
expPlot = scatter(kexp,Eexp, 'MarkerFaceColor',[0 0 0],'MarkerEdgeColor',[0 0
0],'Marker','.');
set(expPlot,'SizeData',500);

% Set up axes
axis(range)
% title(plotTitle, 'FontSize', font+1)

% Create ylabel
ylabel('Photon Energy (\it{eV})','Units','normalized','FontSize',37.4);

% Create xlabel
xlabel('k_{//} (\it{nm}^{-1})','FontSize',37.4);

ax = gca;
ax.TickDir = 'in';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
ax.TickLength = [0.025 0.05];
ax.LineWidth = 1.7;
ax.FontSize = 32;
xticks([0.012 0.014 0.016 0.018]);
% yticks(1.6:0.2:2.4);

% % Label bare exciton positions
% text('VerticalAlignment','top','FontAngle','italic','FontSize',32,...
% 'String','E_A',...
% 'Position',[0.02 EA]);
%
% text('VerticalAlignment','bottom','FontAngle','italic','FontSize',32,...
% 'String','E_B',...
% 'Position',[0.02 EB]);
%
% text('VerticalAlignment','bottom','FontAngle','italic','FontSize',32,...
% 'String','E_C',...
% 'Position',[0.02 EC]);

% Experimental data on legend doesn't show up well
fakePlot = plot([range(1) range(2)],[range(4)+1 range(4)+1],'-','Color',[0 0 0],
'LineWidth', 7);
% set(fakePlot,'Visible','off');

```

```

% Create legend
a = legend([fakePlot sppPlot cPlot modelPlot1],dataLabel);
% set(a,'location','northwest');
set(a,'box','off');
set(a,'Position', [0.1944    0.7500    0.2183    0.1528]);
set(a, 'AutoUpdate','off');

% Label the Rabi splitting
GC = modelFitParams(6);
% [minValue,closestIndex] = min(abs(Espp-EC));
% kRabi = kspp(closestIndex);
% plot([kspp kspp])

arrow = annotation('doublearrow',[0.4752 0.4752],[0.4861 0.6583]);
set(arrow, 'LineWidth',3);
set(arrow, 'Head1Width',40);
set(arrow, 'Head1Length',20);
set(arrow, 'Head2Width',40);
set(arrow, 'Head2Length',20);

removeLine = annotation('textbox',[0.1984 0.5694 0.2659 0.0120],'String','');
removeLine.BackgroundColor = [0.99999 0.99999 0.99999];
removeLine.LineStyle = 'none';

space = ' ';
% om = ['\Omega = ',space];
rabi = ['\Omega = ',space, num2str(GC*1000,3), ' meV'];
% rabi = strcat(om, num2str(GC*1000,3), ' meV');
omega = annotation('textbox',[0.1984 0.5444 0.2719 0.0653],'String',rabi);
omega.LineStyle = 'none';
omega.FontSize = 44;
omega.FontAngle = 'italic';
% omega.FontWeight = 'bold';
omega.FitBoxToText = 'off';
% omega.BackgroundColor = [0.99999 0.99999 0.99999];

print(fullfile('Figures',titlekx), '-dpng', '-r300')

```

merge_1

```
%% This script imports reflection data from JA Woolam .dat files

% First column is polarization and data type (as best I can tell), eg pR is
% p-polarized reflection
% Second column is wavelength in units given by line 4 (I'll try to stick
% to nm)
% Third column is angle
% Fourth column is reflectance
% Fifth column is deviation (would need to look up to understand how
% they're reporting)
% Finally, measurements may be taken at various angles, with subsequent
% angles just filling in subsequent rows
% Note that imported data (variable 'dataxy') will pick out the first column
% as text and put into 'dataxy.textdata' and so there will only be four
% columns in 'dataxy.data'

% Get a list of all files in the folder with the desired file name pattern.
myFolder = pwd;
dataFolder = fullfile(myFolder, 'Input Data');
extension = '.dat'; % Change to whatever extension you need.
delimiter = '\t'; % Change here too
filePattern = fullfile(dataFolder, strcat('*', extension));
theFiles = dir(filePattern); % Get file names

% Extract the data from each file
for k = 1 : length(theFiles)
    % Get the name of the file and the name+path of the file
    fileName = theFiles(k).name;
    fullFileName = fullfile(dataFolder, fileName);

    % Import the data
    dataxy(k) = importdata(fullFileName, delimiter);

    % Extract the x, y, and angle columns
    datacolumns(k).angle = dataxy(k).data(:,2);
    datacolumns(k).wavelength = dataxy(k).data(:,1);
    datacolumns(k).reflectance = dataxy(k).data(:,3);

    % Create new matrices that will contain the x and y data for each
    % angle in a separate column, along with a vector containing the angles
    % and another containing the wavelengths

    % Initialize the matrices
    datamatrix(k).angle(1) = datacolumns(k).angle(1);
    datamatrix(k).wavelength(1,1) = datacolumns(k).wavelength(1);
    datamatrix(k).reflectance(1,1) = datacolumns(k).reflectance(1);

    % Fill in the matrices
    l=1; % counter for angle
    m = 1; % counter for column
    n = 2; % counter for row
    for i = 2:length(dataxy(k).data)
        if datacolumns(k).wavelength(i) > datacolumns(k).wavelength(i-1)
            datamatrix(k).wavelength(n,m) = datacolumns(k).wavelength(i);
            datamatrix(k).reflectance(n,m) = datacolumns(k).reflectance(i);
            n=n+1;
        else
            l=l+1;
            m=m+1;
            n=1;
            datamatrix(k).angle(l) = datacolumns(k).angle(i);
            datamatrix(k).wavelength(n,m) = datacolumns(k).wavelength(i);
            datamatrix(k).reflectance(n,m) = datacolumns(k).reflectance(i);
            n=n+1;
        end
    end
end
```

```

        end
    end
end % of importing each data file

% Check that the wavelengths are the same in each datafile
% If not, manually figure out what to do
for i=2:length(datamatrix)
    tf = isequal(datamatrix(i-1).wavelength(:,1),datamatrix(i).wavelength(:,1));
    if tf==0
        sprintf(['wavelengths not the same for ' '' theFiles(i-1).name ' " and "'
theFiles(i).name ' "'])
    end
end

% The wavelengths are the same for every spectrum, so save it one time, and
% make it a row vector
wavelength = datamatrix(1).wavelength(:,1)';

% Construct a vector containing all of the angles and a matrix containing
% the reflectance values for each angle (columns) and wavelength (rows)
angle = datamatrix(1).angle;
reflectance = datamatrix(1).reflectance;
for i=2:length(datamatrix)
    angle = [angle datamatrix(i).angle];
    reflectance = [reflectance datamatrix(i).reflectance];
end

% Sort the reflectance matrix by inserting the angle vector into the first
% row, then sorting based on angle. To use sortrows, need to invert the
% matrix.
reflectance = sortrows([angle; reflectance]')';

% Sort the angle matrix
angle = sort(angle);

% Initialize a new matrix that will contain the angles in the first row,
% the wavelengths in the first column, and the reflectance values in the
% rest.
mergedMatrix = reflectance;

% % Delete the row of angles from the reflectance matrix
% reflectance(1,:) = [];

% Add the wavelengths to the first column of the merged matrix
mergedMatrix = [zeros(size(mergedMatrix,1),1) mergedMatrix];
mergedMatrix(2:end,1)=wavelength';

% Save data to a file
name = 'merged.txt';
outputName = fullfile('Output Data',name);
dlmwrite(outputName, mergedMatrix,'delimiter','\t');

clearvars myFolder extension filePattern k fileName nickNameB date fullFileName delimiter
m i dataxy theFiles l n
clearvars angle datacolumns datamatrix mergedMatrix reflectance tf wavelength dataFolder
name outputName

```

normalize_2

```
%% Normalizes the input reflectance vs angle to a control reflectance vs angle

% Import the sample and control data
mergedInput = fullfile('Output Data','merged.txt');
mergedData = dlmread(mergedInput);
controlInput = fullfile('Input Data','control.txt');
mergedControl = dlmread(controlInput);

% Check that the two datasets are the same size
% If not, correct by hand
tf = isequal(size(mergedData), size(mergedControl));
if tf==0
    sprintf(['Control and normal matrices are different sizes. Good luck.'])
end

% Import the baseline data
% The 200 nm Ag coated prism control data is all normalized to the
% reflection of itself at 50 deg. The spectrum used to normalize is
% called the 'originalControlBaseline'
% However, since the light source can change over time or with user
% adjustments, this control baseline at 50 deg is taken each session so
% the control data can be renormalized using it.
ogBaselineInput = fullfile('Input Data','originalControlBaseline.txt');
originalControlBaselineImport = importdata(ogBaselineInput,'\t');
originalControlBaseline = originalControlBaselineImport.data(:, 3);
newBaselineInput = fullfile('Input Data','newControlBaseline.txt');
newControlBaselineImport = importdata(newBaselineInput,'\t');
newControlBaseline = newControlBaselineImport.data(:, 3);

% Create separate vectors containing the angle and wavelength data
wavelength = mergedData(2:end, 1)';
angle = mergedData(1,2:end);

% Then strip out the angle and wavelength data leaving only R
data = mergedData;
data(1, :) = [];
data(:, 1) = [];

control = mergedControl;
control(1, :) = [];
control(:, 1) = [];

% Renormalize the control to the new baseline by multiplying out the old one
% and diving in the new one. R_control = Counts(theta)/Counts_baseline_original
control = control.*originalControlBaseline./newControlBaseline;
% Normalize vs the control
renormalizedData = data ./ control;

% Recreate matrix with angles along top row and wavelengths along 1st
% column
mergedMatrix = [angle; renormalizedData];
mergedMatrix = [zeros(size(mergedMatrix, 1), 1) mergedMatrix];
mergedMatrix(2:end, 1) = wavelength';

% Save data to a file
outputName = fullfile('Output Data','mergedNormalized.txt');
dlmwrite(outputName, mergedMatrix,'delimiter','\t');

clearvars controlInput mergedInput mergedMatrix newBaselineInput ogBaselineInput
outputName
```

```
clearvars mergedData mergedControl tf wavelength angle data control renormalizedData plot
originalControlBaselineImport originalControlBaseline newControlBaselineImport
newControlBaseline
```

plotData

```
%% Plot the reflectance and dispersion

%% Reflectance plot
% Set up plot defaults
xlabel = 'Wavelength (nm)';
ylabel = 'Reflectance';

for l = 1:length(angle)
    plotLabels{l} = num2str(angle(l));
end

plotReflectance(wl,data,xlabel,ylabel,plotLabels,plotTitle, wlLimits, dR);

%% Plot the dispersion
% Plot it
dispersionX = 1; % 0 for kx, 1 for theta; if plotting kx, 3 data files for E, kx, data
will also be output
plotDispersion(angle, wl, A, n_substrate, layers, thicknesses, polarization, dispersionX,
range);
dispersionX = 0;
plotDispersion(angle, wl, A, n_substrate, layers, thicknesses, polarization, dispersionX,
range);

% clearvars delimiter nFileName indexDirectory nFullFileName substrateData
clearvars j l xlabel ylabel plotLabels dispersionX
```

shift_3

```
% The ellipsometer in use for this dataset generally has an offset at 1,000
% nm. The data for my sample sets generally converge at long wavelengths,
% so let's assume that the values >=1,000 nm are of "correct" magnitude, and
% then shift everything <1,000 nm up or down to line up with the 1st data
% point >=1,000 nm

% Import the previously normalized data
inputData = fullfile('Output Data','mergedNormalized.txt');
mergedNormalizedData = dlmread(inputData);

% Extract wavelength and angle information
wavelength = mergedNormalizedData(2:end, 1)';
angle = mergedNormalizedData(1,2:end);

% Then strip out the angle and wavelength data leaving only R
data = mergedNormalizedData;
data(1, :) = [];
data(:, 1) = [];

% Do the shift
y = data;
xShift = 999; % Where does shift occur?
visShiftX = round(interp1(wavelength,[1:length(wavelength)],xShift));
extrapReg = 0.1; % Percentage of data points after detector crossover to consider in
fitting routine for finding offset in data
r = round(extrapReg*length(wavelength(visShiftX+1:end))); % Number of data points
corresponding to previous

for j = 1:length(angle)

    xIR = wavelength(visShiftX+1:visShiftX+r)'; % x points to consider for fitting
    yIR = y(visShiftX+1:visShiftX+r,j); % y points to consider for fitting

    xf = wavelength(visShiftX:visShiftX+r)'; % x points including discontinuity
region
    yf = y(visShiftX:visShiftX+r,j); % y points including discontinuity region

    p = polyfit(xIR, yIR, 2); % fit with order 2
    interpolatedY = polyval(p,xShift); % fitted value on visible side of
discontinuity

    visShiftY = y(visShiftX,j) - interpolatedY; % finds the difference between the
actual y value and the fitted y value
    % Start by just duplicating the data
    y_norm(:,j) = y(:,j);

    % For wavelengths longer than 1,000 nm, keep constant, leaving
    % matrix untouched

    % For wavelengths <= 1,000 nm, write over the values with shifted
    % values
    y_norm(1:visShiftX, j) = y(1:visShiftX, j) - visShiftY;

end

% Recreate matrix with angles along top row and wavelengths along 1st
% column
mergedMatrix = [angle; y_norm];
mergedMatrix = [zeros(size(mergedMatrix, 1), 1) mergedMatrix];
mergedMatrix(2:end, 1) = wavelength';

% Save data to a file
outputData = fullfile('Output Data','mergedNormalizedShifted.txt');
dlmwrite(outputData, mergedMatrix,'delimiter','\t');
```

```
clearvars mergedNormalizedData wavelength plot y y_norm visShiftX visShiftY mergedMatrix
clearvars outputData inputData j xIR yIR l p xf yf r extrapReg interpolatedY xShift
```

runFitRoutineData

```
function [fitresult, gof] = runFitRoutine(x, y, init, upper, lower)
%CREATEFITS(X,Y)
% Create fits.
%
% Data for 'untitled fit 10' fit:
%   X Input : x
%   Y Output: y
% Output:
%   fitresult : a cell-array of fit objects representing the fits.
%   gof : structure array with goodness-of fit info.
%
% See also FIT, CFIT, SFIT.

% from script Auto-generated by MATLAB on 25-Feb-2019 16:08:43

%% Initialization.

% Initialize arrays to store fits and goodness-of-fit.
fitresult = cell( 1, 1 );
gof = struct( 'sse', cell( 1, 1 ), ...
    'rsquare', [], 'dfe', [], 'adjrsquare', [], 'rmse', [] );

%% Fit: 'untitled fit 10'.
[xData, yData] = prepareCurveData( x, y );

% Set up fittype and options.
ft = fittype( '(a*g0/((x-x0)^2+(g0/2)^2) + b*g1/((x-x1)^2+(g1/2)^2) + c*g2/((x-
x2)^2+(g2/2)^2) + d*g3/((x-x3)^2+(g3/2)^2) )/(2*pi)', 'independent', 'x', 'dependent',
'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Display = 'Off';
opts.Lower = lower;
opts.MaxFunEvals = 1000;
opts.MaxIter = 1000;
opts.Robust = 'LAR';
opts.StartPoint = init;
opts.Upper = upper;

% Fit model to data.
[fitresult{1}, gof(1)] = fit( xData, yData, ft, opts );

% % Delete points at bounds
% if round(fitresult{1}.a,1,'significant') == round(opts.Lower(1),1,'significant') ||
round(opts.Upper(1),1,'significant')
%     fitresult{1}.a = NaN;
% end
%
% if round(fitresult{1}.b,1,'significant') == round(opts.Lower(2),1,'significant') ||
round(opts.Upper(2),1,'significant')
%     fitresult{1}.b = NaN;
% end
%
% if round(fitresult{1}.c,1,'significant') == round(opts.Lower(3),1,'significant') ||
round(opts.Upper(3),1,'significant')
%     fitresult{1}.c = NaN;
% end
%
% if round(fitresult{1}.g0,1,'significant') == round(opts.Lower(4),1,'significant') ||
round(opts.Upper(4),1,'significant')
%     fitresult{1}.g0 = NaN;
% end
%
% if round(fitresult{1}.g1,1,'significant') == round(opts.Lower(5),1,'significant') ||
round(opts.Upper(5),1,'significant')
```

```

%     fitresult{1}.g1 = NaN;
% end
%
% if round(fitresult{1}.g2,1,'significant') == round(opts.Lower(6),1,'significant') ||
round(opts.Upper(6),1,'significant')
%     fitresult{1}.g2 = NaN;
% end
%
% if round(fitresult{1}.x0,1,'significant') == round(opts.Lower(7),1,'significant') ||
round(opts.Upper(7),1,'significant')
%     fitresult{1}.x0 = NaN;
% end
%
% if round(fitresult{1}.x1,1,'significant') == round(opts.Lower(8),1,'significant') ||
round(opts.Upper(8),1,'significant')
%     fitresult{1}.x1 = NaN;
% end
%
% if round(fitresult{1}.x2,1,'significant') == round(opts.Lower(9),1,'significant') ||
round(opts.Upper(9),1,'significant')
%     fitresult{1}.x2 = NaN;
% end

% if mod(m,10)==0
% % Plot fit with data.
% figure( 'Name', 'untitled fit 10' );
% h = plot( fitresult{1}, xData, yData );
% legend( h, 'y vs. x', 'untitled fit 1', 'Location', 'NorthEast', 'Interpreter', 'none'
);
% % Label axes
% xlabel( 'x', 'Interpreter', 'none' );
% ylabel( 'y', 'Interpreter', 'none' );
% grid on
% end

```

interpolateKdispersion

```
function [] = interpolateKdispersion(eV, kx, Adata)
% Now, take the dispersion data and interpolate it so that columns of
% k-data are all the same

minkx = min(min(kx));
maxkx = max(max(kx));
kxresfactor = 10;
kxres = (maxkx-minkx)/(size(kx,2)-1)/kxresfactor;
kxInterp = minkx:kxres:maxkx;
kxInterpGrid = repmat(kxInterp,size(eV,1),1);
eVInterpGrid = repmat(eV(:,1),1,size(kxInterpGrid,2));

for i = 1:size(eV,1)
Vq1D(i,:)=interp1(kx(i,:), Adata(i,:), kxInterp, 'makima', NaN);
end

% these data files will have constant kx along the columns
dlmwrite(fullfile('Output Data','spp_eV_interp.txt'), eVInterpGrid,'delimiter','\t');
dlmwrite(fullfile('Output Data','spp_kx_interp.txt'), kxInterpGrid,'delimiter','\t');
dlmwrite(fullfile('Output Data','spp_Adata_eV_kx_interp.txt'), Vq1D,'delimiter','\t');
clear all
```

runFitRoutine

```
function [fitresult, gof] = runFitRoutine(x, y, init, upper, lower)
%CREATEFITS(X,Y)
% Create fits.
%
% Data for 'untitled fit 10' fit:
%   X Input : x
%   Y Output: y
% Output:
%   fitresult : a cell-array of fit objects representing the fits.
%   gof : structure array with goodness-of fit info.
%
% See also FIT, CFIT, SFIT.

% from script Auto-generated by MATLAB on 25-Feb-2019 16:08:43

%% Initialization.

% Initialize arrays to store fits and goodness-of-fit.
fitresult = cell( 1, 1 );
gof = struct( 'sse', cell( 1, 1 ), ...
    'rsquare', [], 'dfe', [], 'adjrsquare', [], 'rmse', [] );

%% Fit: 'untitled fit 10'.
[xData, yData] = prepareCurveData( x, y );

% Set up fittype and options.
ft = fittype( '(a*g0/((x-x0)^2+(g0/2)^2)+b*g1/((x-x1)^2+(g1/2)^2))/(2*pi)+m*x',
    'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Display = 'Off';
opts.Lower = lower;
opts.MaxFunEvals = 1000;
opts.MaxIter = 1000;
opts.Robust = 'LAR';
opts.StartPoint = init;
opts.Upper = upper;

% Fit model to data.
[fitresult{1}, gof{1}] = fit( xData, yData, ft, opts );
```

EVectorsofKunbounded

```
function [V, D] = EVectorsofKunbounded(x,xdata,Espp,kspp)

    % Different xdata or k values for the three branches, so solve it three
    % times, once for each k value
    EsppOfK = interp1(kspp,Espp,xdata, 'linear', 'extrap');
    M = zeros(4,4,length(xdata));
    % x, x0 = [EA0 EB0 EC0 GA GB GC]; % Format of fit values

    % Build the matrix
    M(1,1,:) = EsppOfK;
    M(1,2,:) = x(4)/2;
    M(1,3,:) = x(5)/2;
    M(1,4,:) = x(6)/2;
    M(2,1,:) = M(1,2,:);
    M(3,1,:) = M(1,3,:);
    M(4,1,:) = M(1,4,:);
    M(2,2,:) = x(1);
    M(3,3,:) = x(2);
    M(4,4,:) = x(3);

    for n = 1:length(xdata)
        [V(:,:,n), D(:,:,n)] = eig(M(:,:,n));
    end

end
```

EVofK

```
function [eCat] = EVofK(x,xdata4,Espp,kspp)

    % Different xdata or k values for the 4 branches, so solve it 4
    % times, once for each k value
    for m = 1:4
        xdata = xdata4{m};
        EsppOfK = interp1(kspp,Espp,xdata, 'linear', 'extrap');
        M = zeros(4,4,length(xdata));
        % x, x0 = [EA0 EB0 EC0 GA GB GC]; % Format of fit values

        % Build the matrix
        M(1,1,:) = EsppOfK;
        M(1,2,:) = x(4)/2;
        M(1,3,:) = x(5)/2;
        M(1,4,:) = x(6)/2;
        M(2,1,:) = M(1,2,:);
        M(3,1,:) = M(1,3,:);
        M(4,1,:) = M(1,4,:);
        M(2,2,:) = x(1);
        M(3,3,:) = x(2);
        M(4,4,:) = x(3);

        % for each k value, calculate the eigenvalues
        for n = 1:length(xdata)
            e{m}(:,n) = eig(M(:, :, n));
        end
    end

    % put the eigenvalues together
    ee = {e{1}(1,:);
          e{2}(2,:);
          e{3}(3,:);
          e{4}(4,:)};

    eCat = horzcat(ee{:});
end
```

EVofKunbounded

```
function [e] = EVofKunbounded(x,xdata,Espp,kspp)

    EsppOfK = interp1(kspp,Espp,xdata, 'linear', 'extrap');
    M = zeros(4,4,length(xdata));
    % x, x0 = [EA0 EB0 EC0 GA GB GC]; % Format of fit values

    % Build the matrix
    M(1,1,:) = EsppOfK;
    M(1,2,:) = x(4)/2;
    M(1,3,:) = x(5)/2;
    M(1,4,:) = x(6)/2;
    M(2,1,:) = M(1,2,:);
    M(3,1,:) = M(1,3,:);
    M(4,1,:) = M(1,4,:);
    M(2,2,:) = x(1);
    M(3,3,:) = x(2);
    M(4,4,:) = x(3);

    for n = 1:length(xdata)
        e(:,n) = eig(M(:,:,n));
    end

end
```

processDispersion

```
function [] = processDispersion(angle, wl, A, n_substrate, layers, thicknesses,
polarization, dispersionX, range)
%% This function plots dispersion of a thin film stack
    % dispersionX, 0 for kx or 1 for theta on the x-axis

%% Set up matrices for plotting
    % pcolor is used to make the surface plot, however, it throws away data
    % rather than just plotting a color pixel for every data point
    % The code below creates matrices out of the x and y axes that
    % correspond to the edges of where each pixel should be rather than the
    % center coordinates of each pixel. The, the absorptivity matrix has a
    % blank row and column added so that when pcolor deletes the last row
    % and column it won't lose any information.

    % Compute E vector based on wavelength range of interest
    eV = 1240 ./ wl';    % Convert wavelengths to energy (eV)

    % Turn eV and angle vectors into matrices the same size as A matrix
    eVtemp = eV;
    angleTemp = angle;
    Nangles = length(angle);
    for l=1:Nangles
        eV(:,l) = eVtemp;
    end
    Nwavelengths = length(wl);
    for l=1:Nwavelengths
        angle(l,:) = angleTemp;
    end

    % Create the matrices for the pixel edges for y axis as energy and x axis
    % as angle
    x = angle;
    y = eV;

    % Find differences between adjacent points
    xSplit = diff(x,1,2)/2;
    ySplit = diff(y,1,1)/2;

    % Create data points for the edge of each pixel by adding or
    % subtracting the difference between adjacent points
    xEdges = [x(:,1)-xSplit(:,1) x(:,2:end)-xSplit x(:,end)+xSplit(:,end)];
    yEdges = [y(1,:)-ySplit(1,:); y(2:end,:)-ySplit; y(end,:)+ySplit(end,:)];

    % Add an extra row or column so the x and y matrices will have the
    % same size
    xEdges = [xEdges; xEdges(1,:)];
    yEdges = [yEdges yEdges(:,1)];

    % Make the absorptivity data matrix the same size as the pixel edge
    % matrices. The last row and column are ignored. The default shading of
    % the pcolor function will color each pixel according to the value at
    % the corner of that pixel with the smallest indices
    data = A; % save a copy of the original data
    A = [[A zeros(size(A,1),1)] ; zeros(1,size(A,2)+1)];

    % Plot vs kx or angle
    if dispersionX == 1 % then x-axis is angle (deg)
        xlabel = '\textbf{Angle} $\theta$ \ (degrees)$';
        xaxis = xEdges;
        % While we're here, let's output new E vs angle datafiles
        dlmwrite(fullfile('Output Data','eV.txt'), eV,'delimiter','\t'); % Use the
original eV data
        dlmwrite(fullfile('Output Data','angle.txt'), angle,'delimiter','\t');
        dlmwrite(fullfile('Output Data','Adata_eV_angle.txt'), data,'delimiter','\t');
    else
        % x-axis is wavenumber (nm-1)
```

```

xLabel = '\textbf{Wavenumber} $\frac{2\pi}{\lambda}n\sin{\theta}$ \ \ (nm^{-1})$';

% Create a matrix for kx. This uses the matrices of pixel edge
% positions for energy (wavelength), angle, and a new one for the
% substrate refractive index.

% Create the new substrate index matrix
n=n_substrate;
n_substrateTemp = n;
for l=1:Nangles
    n(:,l) = n_substrateTemp;
end

% Make it the same size as the other pixel edge matrices
nSplit = diff(n,1,1)/2;
nEdges = [n(1,:)-nSplit(1,:); n(2:end,:)-nSplit; n(end,:)+nSplit(end,:)];
nEdges = [nEdges nEdges(:,1)];

% Calculate the new kx matrix
wlEdges = 1240 ./ yEdges;
kxEdges = 2*pi.*nEdges.*yEdges/1240.*sin(pi/4+asin(sin(pi*xEdges/180-
pi/4)./nEdges));
xAxis = kxEdges;

% While we're here, let's output new E vs kx datafiles
dlmwrite(fullfile('Output Data','eV.txt'), eV,'delimiter','\t'); % Use the
original eV data
% Have to create kx that corresponds to the original data, not
% the pixel edges for the plots
kx = 2*pi.*n.*eV/1240.*sin(pi/4+asin(sin(pi*angle/180-pi/4)./n));

dlmwrite(fullfile('Output Data','kx.txt'), kx,'delimiter','\t');
dlmwrite(fullfile('Output Data','Adata_eV_kx.txt'), data,'delimiter','\t');

end

yLabel = '\textbf{Photon Energy} $(eV)$';

%% Setup titles
% Create title text
stack = [layers{1}, ' / '];
note = [ ' ' layers{2} ': ' thicknesses{1} ];
for q=2:length(layers)-1
    stack = [stack, layers{q}, ' / '];
end
if length(layers) > 3
    for q = 3:length(layers)-1
        note = [note, ' ' layers{q} ': ' thicknesses{q-1} ];
    end
end
stack = [stack, layers{end}];
% Polarization
if polarization==0
    pol = 'TE';
else
    pol = 'TM';
end
% Put it all together
plotTitle = {'Dispersion with ' pol '-polarization'}, ['Layers: ' stack], note,
' '];

% Create name to save image
saveStack = strcat(layers{1}, '-');
for q=2:length(layers)-1
    saveStack = strcat(saveStack, layers{q}, '-');
end
saveStack = strcat(saveStack, layers{end});
saveTitle = ['Dispersion with ' pol '-polarization ', saveStack, ', ' note];
saveTitle = replace(saveTitle, ':', ''); % colons aren't allowed in Windows
filenames, so remove them

```

```

%% Set up plot
%   set(0,'DefaultFigureVisible','off'); % Don't display the plot--just save it
font = 24;

Plot = figure;
set(Plot, 'Units', 'normalized');
set(Plot, 'Position', [0 0 1 1]);
axes('FontSize', font) ;
xlabel(xLabel, 'FontSize', font, 'Interpreter','latex');
ylabel(yLabel, 'FontSize', font, 'Interpreter','latex');

hold on
plot = pcolor(xAxis,yEdges,A);
plot.EdgeColor = 'none';
colormap('hot')
axis(range)
c=colorbar;
caxis([0 inf])
c.Label.String = 'Absorptivity';
title(plotTitle, 'FontSize', font+1)
ax = gca;
ax.TickDir = 'out';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
hold off

% Save figures
f = get(groot,'CurrentFigure'); % f.Number will give current fig number
directory = fullfile(pwd, 'Figures'); % create folder name to save
mkdir(directory); % create the folder
saveas(Plot, fullfile(directory,[saveTitle '_' num2str(f.Number) '.png']));
savefig(Plot, fullfile(directory,[saveTitle '_' num2str(f.Number) '.fig']));

```

processReflectance

```
% _____ %  
% Plot R, T, A, etc. data %  
% _____ %  
  
function output = plotData(xData, yData, xLabel, yLabel, dataLabel, plotTitle, wl, dR)  
% xData, yData should be given as a vector, i.e. using []  
% dataLabel should be given as a cell (i.e., same as vector but using {}  
% Eg, [x{1} x{2}]; or for a single data set [x{1}] and not just x{1}  
% plotOpticsData([x{8} x{3}], [y{8} y{3}], 'Wavelength (nm)', '% Transmission', {labels{8}  
labels{3}}, 'title')  
  
% Set default font size of figure labels  
font = 24;  
  
Plot = figure;  
set(Plot, 'Units', 'normalized');  
set(Plot, 'Position', [0 0 1 1]);  
axes('FontSize', font)  
xlabel(xLabel, 'FontSize', font)  
ylabel(yLabel, 'FontSize', font)  
hold on  
  
for k=1:size(yData,2)  
scatter(xData, yData(:,k), 'filled')  
end  
  
% To not allow underscores to be interpreted as subscripts, use 'Interpreter', 'none'  
a=legend(dataLabel, 'location', 'southeast');  
a.Title.String = 'Angle of Incidence (deg)';  
title(plotTitle, 'FontSize', font+2)  
axis([wl dR])  
ax = gca;  
ax.Box = 'on';  
ax.Layer = 'top';  
ax.XMinorTick = 'on';  
ax.YMinorTick = 'on';  
  
hold off  
  
% Save figures  
f = get(groot, 'CurrentFigure'); % f.Number will give current fig number  
directory = fullfile(pwd, 'Figures'); % create folder name to save  
mkdir(directory); % create the folder  
saveas(Plot, fullfile(directory, [plotTitle '_' num2str(f.Number) '.png']));  
savefig(Plot, fullfile(directory, [plotTitle '_' num2str(f.Number) '.fig']));  
  
end
```

plotDispersion

```
function [] = plotDispersion(angle, wl, A, n_substrate, layers, thicknesses,
polarization, dispersionX, range)
%% This function plots dispersion of a thin film stack
    % dispersionX, 0 for kx or 1 for theta on the x-axis

%% Set up matrices for plotting
    % pcolor is used to make the surface plot, however, it throws away data
    % rather than just plotting a color pixel for every data point
    % The code below creates matrices out of the x and y axes that
    % correspond to the edges of where each pixel should be rather than the
    % center coordinates of each pixel. The, the absorptivity matrix has a
    % blank row and column added so that when pcolor deletes the last row
    % and column it won't lose any information.

    % Compute E vector based on wavelength range of interest
    eV = 1240 ./ wl';    % Convert wavelengths to energy (eV)

    % Turn eV and angle vectors into matrices the same size as A matrix
    eVtemp = eV;
    angleTemp = angle;
    Nangles = length(angle);
    for l=1:Nangles
        eV(:,l) = eVtemp;
    end
    Nwavelengths = length(wl);
    for l=1:Nwavelengths
        angle(l,:) = angleTemp;
    end

    % Create the matrices for the pixel edges for y axis as energy and x axis
    % as angle
    x = angle;
    y = eV;

    % Find differences between adjacent points
    xSplit = diff(x,1,2)/2;
    ySplit = diff(y,1,1)/2;

    % Create data points for the edge of each pixel by adding or
    % subtracting the difference between adjacent points
    xEdges = [x(:,1)-xSplit(:,1) x(:,2:end)-xSplit x(:,end)+xSplit(:,end)];
    yEdges = [y(1,:)-ySplit(1,:); y(2:end,:)-ySplit; y(end,:)+ySplit(end,:)];

    % Add an extra row or column so the x and y matrices will have the
    % same size
    xEdges = [xEdges; xEdges(1,:)];
    yEdges = [yEdges yEdges(:,1)];

    % Make the absorptivity data matrix the same size as the pixel edge
    % matrices. The last row and column are ignored. The default shading of
    % the pcolor function will color each pixel according to the value at
    % the corner of that pixel with the smallest indices
    data = A; % save a copy of the original datafile
    A = [[A zeros(size(A,1),1)] ; zeros(1,size(A,2)+1)];

    % Plot vs kx or angle
    if dispersionX == 1 % then x-axis is angle (deg)
        xlabel = '\textbf{Angle} $\theta$ \ (degrees)';
        xAxis = xEdges;
    else % x-axis is wavenumber (nm-1)
        xlabel = '\textbf{Wavenumber} $\frac{2\pi}{\lambda n \sin\theta}$ \ (nm^{-1})';
    end

    % Create a matrix for kx. This uses the matrices of pixel edge
    % positions for energy (wavelength), angle, and a new one for the
    % substrate refractive index.
```

```

        % Create the new substrate index matrix
n=n_substrate;
n_substrateTemp = n;
for l=1:Nangles
    n(:,l) = n_substrateTemp;
end

        % Make it the same size as the other pixel edge matrices
nSplit = diff(n,1,1)/2;
nEdges = [n(1,:)-nSplit(1,:); n(2:end,:)-nSplit; n(end,:)+nSplit(end,:)];
nEdges = [nEdges nEdges(:,1)];

        % Calculate the new kx matrix
%       wEdges = 1240 ./ yEdges;
kxEdges = 2*pi.*nEdges.*yEdges/1240.*sin(pi/4+asin(sin(pi*xEdges/180-
pi/4)./nEdges));
xAxis = kxEdges;

        % Have to create kx that corresponds to the original data, not
        % the pixel edges for the plots
kx = 2*pi.*n.*eV/1240.*sin(pi/4+asin(sin(pi*angle/180-pi/4)./n));
end

yLabel = '\textbf{Photon Energy} $(eV)$';

%% Setup titles
% Create title text
stack = [layers{1}, ' / '];
note = [ ' ' layers{2} ': ' thicknesses{1}  ];
for q=2:length(layers)-1
    stack = [stack, layers{q}, ' / '];
end
if length(layers) > 3
    for q = 3:length(layers)-1
        note = [note, ' ' layers{q} ': ' thicknesses{q-1} ];
    end
end
stack = [stack, layers{end}];
% Polarization
if polarization==0
    pol = 'TE';
else
    pol = 'TM';
end
% Put it all together
plotTitle = {'Dispersion with ' pol '-polarization'}, ['Layers: ' stack], note,
' '];

% Create name to save image
saveStack = strcat(layers{1},'-');
for q=2:length(layers)-1
    saveStack = strcat(saveStack, layers{q},'-');
end
saveStack = strcat(saveStack, layers{end});
saveTitle = ['Dispersion with ' pol '-polarization ', saveStack, ' ', note];
saveTitle = replace(saveTitle, ':', ''); % colons aren't allowed in Windows
filenames, so remove them

%% Set up plot
% set(0,'DefaultFigureVisible','off'); % Don't display the plot--just save it
font = 24;

Plot = figure;
set(Plot, 'Units', 'normalized');
set(Plot, 'Position', [0 0 1 1]);
axes('FontSize', font) ;
xlabel(xLabel, 'FontSize', font, 'Interpreter','latex');
ylabel(yLabel, 'FontSize', font, 'Interpreter','latex');

hold on
plot = pcolor(xAxis,yEdges,A);

```

```
plot.EdgeColor = 'none';
colormap('hot')
axis(range)
c=colorbar;
caxis([0 inf])
c.Label.String = 'Absorptivity';
title(plotTitle, 'FontSize', font+1)
ax = gca;
ax.TickDir = 'out';
ax.Layer = 'top';
ax.Box = 'on';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';
hold off

clearvars l
```

plotReflectance

```
% Plot R, T, A, etc. data
function output = plotData(xData, yData, xlabel, ylabel, dataLabel, plotTitle, wl, dR)
% xData, yData should be given as a vector, i.e. using []
% dataLabel should be given as a cell (i.e., same as vector but using {})
% Eg, [x{1} x{2}]; or for a single data set [x{1}] and not just x{1}
% plotOpticsData([x{8} x{3}], [y{8} y{3}], 'Wavelength (nm)', '% Transmission', {labels{8}
labels{3}}, 'title')

% Set default font size of figure labels
font = 24;

Plot = figure;
set(Plot, 'Units', 'normalized');
set(Plot, 'Position', [0 0 1 1]);
axes('FontSize', font)
xlabel(xlabel, 'FontSize', font)
ylabel(ylabel, 'FontSize', font)
hold on

for k=1:size(yData,2)
scatter(xData, yData(:,k), 'filled')
end

% To not allow underscores to be interpreted as subscripts, use 'Interpreter', 'none'
a=legend(dataLabel, 'location', 'southeast');
a.Title.String = 'Angle of Incidence (deg)';
title(plotTitle, 'FontSize', font+2)
axis([wl dR])
ax = gca;
ax.Box = 'on';
ax.Layer = 'top';
ax.XMinorTick = 'on';
ax.YMinorTick = 'on';

hold off

end
```