Bibliography

Ade, P., et al., First Season QUaD CMB Temperature and Polarization Power Spectra, *apj*, *674*, 22–28, doi:10.1086/524922, 2008.

Agilent, *Advanced Design Suite Momentum v6u2, User Manual*, Agilinet, 2006.

Apsotle, T., *Calculus, Vol. 1: One-Variable Calculus with an Introduction to Linear Algebra*, Wiley, 1967.

Arnold, K., personal communication, 2008.

Behdad, N., and K. Sarabandi, Wideband double-element ring slot antenna, *Electronics Letters*, *40*(7), 408 – 409, doi:10.1049/el:20040292, 2004.

Bell, R., *Introductory Fourier Transform Spectroscopy*, Academic Pr, 1972.

Benford, D. J., M. C. Gaidis, and J. W. Kooi, Optical properties of Zitex in the infrared to submillimeter, *ao*, *42*, 5118–5122, doi:10.1364/AO.42.005118, 2003.

Benson, B. A., et al., Peculiar Velocity Limits from Measurements of the Spectrum of the Sunyaev-Zeldovich Effect in Six Clusters of Galaxies, *apj*, *592*, 674–691, doi:10.1086/375864, 2003.

Bond, J., I could tell you, but then I'd have to kill you, 2010.

Booker, H., Slot aerials and their relation to complementary wire aerials (Babinet's principle), *Electrical Engineers - Part IIIA: Radiolocation, Journal of the Institution of*, *93*(4), 620 –626, doi:10.1049/ji-3a-1.1946.0150, 1946.

Born, M., and E. Wolf, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, Cambridge University Press, 1999.

Bussmann, R. S., W. L. Holzapfel, and C. L. Kuo, Millimeter Wavelength Brightness Fluctuations of the Atmosphere above the South Pole, *apj*, *622*, 1343–1355, doi:10.1086/427935, 2005.

Carlstrom, J. E., et al., The 10 Meter South Pole Telescope, *ArXiv e-prints*, 2009.

Carroll, S., *Spacetime and Geometry: An Introduction to General Relativity*, Benjamin Cummings, 2003.

Challinor, A., and H. Peiris, Lecture notes on the physics of cosmic microwave background anisotropies, in *American Institute of Physics Conference Series*, *American Institute of Physics Conference Series*, vol. 1132, edited by M. Novello & S. Perez, pp. 86–140, doi:10.1063/1.3151849, 2009.

Chang, J., Xenon Difluoride Etching System (XETCH), *Manual Nanoolab Manual section 7.13*, UC Berkeley, Berkeley, Ca, USA, 1998.

Chang, J., CPA Sputtering System, *Manual Nanoolab Manual section 6.04*, UC Berkeley, Berkeley, Ca, USA, 2010a.

Chang, J., Oxford Plasmalab 80plus PECVD System, *Manual Nanoolab Manual section 6.29*, UC Berkeley, Berkeley, Ca, USA, 2010b.

Chang, J., Tystar17 Non-MOS Low Stress Nitride and High Temp. Oxide LPCVD Furnace, *Manual Nanoolab Manual section 5.17*, UC Berkeley, Berkeley, Ca, USA, 2010c.

Chattopadhyay, G., and J. Zmuidzinas, A dual-polarized slot antenna for millimeter waves, *Antennas and Propagation, IEEE Transactions on*, *46*(5), 736 –737, doi:10.1109/8.668920, 1998.

Chattopadhyay, G., J. Glenn, J. Bock, B. Rownd, M. Caldwell, and M. Griffin, Feed horn coupled bolometer arrays for SPIRE - design, simulations, and measurements, *Microwave Theory and Techniques, IEEE Transactions on*, *51*(10), 2139 – 2146, doi:10.1109/TMTT.2003.817428, 2003.

Chattopadhyay, G., C.-L. Kuo, P. Day, J. Bock, J. Zmuidzinas, and A. Lange, Planar antenna arrays for CMB polarization detection, in *Infrared and Millimeter Waves, 2007 and the 2007 15th International Conference on Terahertz Electronics. IRMMW-THz. Joint 32nd International Conference on*, pp. 184 –185, 2007.

Chervenak, J. A., K. D. Irwin, E. N. Grossman, J. M. Martinis, C. D. Reintsema, and M. E. Huber, Superconducting multiplexer for arrays of transition edge sensors, *Applied Physics Letters*, *74*(26), 4043–4045, doi:10.1063/1.123255, 1999.

Chiang, H. C., et al., Measurement of Cosmic Microwave Background Polarization Power Spectra from Two Years of BICEP Data, *apj*, *711*, 1123–1140, doi:10.1088/0004-637X/711/2/1123, 2010.

Collin, R., *Antennas and Radiowave Propagation*, Mcgraw-Hill, 1985.

de Bernardis, P., et al., Detection of anisotropy in the Cosmic Microwave Background at horizon and sub-horizon scales with the BOOMERanG experiment, *ArXiv Astrophysics e-prints*, 2000.

Deschamps, G., Impedance properties of complementary multiterminal planar structures, *Antennas and Propagation, IRE Transactions on*, *7*(5), 371 –378, doi:10.1109/TAP.1959.1144717, 1959.

Dodelson, S., *Modern cosmology*, Academic Press, 2003.

Doe, J., `http://universe-review.ca/`, 2010.

DuHamel, R., Dual polarized sinuous antennas, United States Patent 4658262, 1987.

DuHamel, R., and D. Isbell, Broadband logarithmically periodic antenna structures, in *IRE International Convention Record*, vol. 5, pp. 119 – 128, 1957.

Dyson, J., The equiangular spiral antenna, *Antennas and Propagation, IRE Transactions on*, *7*(2), 181 –187, doi:10.1109/TAP.1959.1144653, 1959.

Edwards, J., personal communication, 2008.

Engargiola, G., Tapered microstrip balun for integrating a low noise amplifier with a nonplanar log periodic antenna, *Review of Scientific Instruments*, *74*, 5197–5200, doi: 10.1063/1.1622975, 2003.

Engargiola, G., W. Holzapfel, A. Lee, M. Myers, R. O'Brient, P. Richards, H. Tran, and H. Spieler, Planar channelized log-periodic antenna, in *Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, 2005. MAPE 2005. IEEE International Symposium on*, vol. 1, pp. 306 – 309 Vol. 1, doi:10.1109/MAPE.2005.1617910, 2005.

Filipovic, D., S. Gearhart, and G. Rebeiz, Double-slot antennas on extended hemispherical and elliptical silicon dielectric lenses, *Microwave Theory and Techniques, IEEE Transactions on*, *41*(10), 1738 –1749, doi:10.1109/22.247919, 1993.

Finelli, F., J. Hamann, S. M. Leach, and J. Lesgourgues, Single-field inflation constraints from CMB and SDSS data, *Journal of Cosmology and Astro-Particle Physics*, *4*, 11–+, doi:10.1088/1475-7516/2010/04/011, 2010.

Finkbeiner, D. P., M. Davis, and D. J. Schlegel, Extrapolation of Galactic Dust Emission at 100 Microns to Cosmic Microwave Background Radiation Frequencies Using FIRAS, *apj*, *524*, 867–886, doi:10.1086/307852, 1999.

Fixsen, D. J., E. S. Cheng, J. M. Gales, J. C. Mather, R. A. Shafer, and E. L. Wright, The Cosmic Microwave Background Spectrum from the Full COBE FIRAS Data Set, *apj*, *473*, 576–+, doi:10.1086/178173, 1996.

Galbraith, C., Cochlear-inspired channelizing filters for Wideband Radio Systems, Ph.D. thesis, The University Michigan Ann Arbor, 2008.

Galbraith, C., and G. Rebeiz, Higher Order Cochlea-Like Channelizing Filters, *Microwave Theory and Techniques, IEEE Transactions on*, *56*(7), 1675 –1683, doi:10.1109/TMTT.2008.925574, 2008.

Galbraith, C., R. White, L. Cheng, K. Grosh, and G. Rebeiz, Cochlea-Based RF Channelizing Filters, *Circuits and Systems I: Regular Papers, IEEE Transactions on*, *55*(4), 969 –979, doi:10.1109/TCSI.2008.916537, 2008.

Gitin, M., F. Wise, G. Arjavalingam, Y. Pastol, and R. Compton, Broad-band characterization of millimeter-wave log-periodic antennas by photoconductive sampling, *Antennas and Propagation, IEEE Transactions on*, *42*(3), 335 –339, doi:10.1109/8.280719, 1994.

Goldin, A., J. J. Bock, C. Hunt, A. E. Lange, H. Leduc, A. Vayonakis, and J. Zmuidzinas, SAMBA: Superconducting antenna-coupled, multi-frequency, bolometric array, *Low Temperature Detectors*, *605*, 251–254, doi:10.1063/1.1457640, 2002.

Goodman, J., *Introduction to Fourier Optics*, McGraw-Hill Science, 1968.

Griffin, M. J., J. J. Bock, and W. K. Gear, Relative Performance of Filled and Feedhorn-Coupled Focal-Plane Architectures, *Appl. Opt.*, *41*(31), 6543–6554, 2002.

Guildemeister, J., Voltage-Biased Superconducting Bolometers for infrared and mm-waves, Ph.D. thesis, University of California Berkeley, 2000.

Gupta, K., G. Garg, B. Bahl, and P. Bhartia, *Microstrip Lines and Slotlines 2nd Ed.*, Artech House Publishers, 1996.

Halverson, N., NEP and Mapping speed for 92 GHz and 150 GHz, *Internal Technical Memo EPEX-SZ-041211a*, UC Berkeley, Berkeley, Ca, USA, 2004.

Halverson, N. W., et al., Degree Angular Scale Interferometer First Results: A Measurement of the Cosmic Microwave Background Angular Power Spectrum, *apj*, *568*, 38–45, doi:10.1086/338879, 2002.

Hamilton, B., Plasma-Therm Parallel Plate Etcher, *Manual Nanoolab Manual section 7.09*, UC Berkeley, Berkeley, Ca, USA, 2010.

Hanany, S., et al., MAXIMA-1: A Measurement of the Cosmic Microwave Background Anisotropy on Angular Scales of 10'-5deg, *apjl*, *545*, L5–L9, doi:10.1086/317322, 2000.

Hu, W., and S. Dodelson, Cosmic Microwave Background Anisotropies, *araa*, *40*, 171–216, doi:10.1146/annurev.astro.40.060401.093926, 2002.

Hu, W., and M. White, A CMB polarization primer, *New Astronomy*, *2*, 323–344, doi:10.1016/S1384-1076(97)00022-5, 1997.

Irwin, K., and G. Hilton, Transition-Edge Sensors, in *Cryogenic Particle Detection*, pp. 63–149, Springer-Verlag, Heidlberg, 2005.

Jackson, J. D., *Classical Electrodynamics Third Edition*, Wiley, 1998.

Johnson, B. R., et al., MAXIPOL: a balloon-borne experiment for measuring the polarization anisotropy of the cosmic microwave background radiation, *New Astronomy Reviews*, *47*(11-12), 1067 – 1075, doi:DOI:10.1016/j.newar.2003.09.034, 2003.

Kaplinghat, M., L. Knox, and Y. Song, Determining Neutrino Mass from the Cosmic Microwave Background Alone, *Physical Review Letters*, *91*(24), 241,301–+, doi:10.1103/PhysRevLett.91.241301, 2003.

Kerr, A., Surface Impedance of Superconductors and Normal Conductors in EM Simulators, *Internal Technical Memo Alma Memo No. 245*, National Radio Astronomy Observatory, Green Bank, W. Va, USA, 1999.

Kittel, C., *Introduction to Solid State Physics, 8th Ed*, wiley, 2004.

Kogut, A., et al., PAPPA: Primordial anisotropy polarization pathfinder array, *New Astronomy Review*, *50*, 1009–1014, doi:10.1016/j.newar.2006.09.024, 2006.

Kogut, A., et al., Three-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Foreground Polarization, *apj*, *665*, 355–362, doi:10.1086/519754, 2007.

Komatsu, E., et al., Seven-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Interpretation, *ArXiv e-prints*, 2010.

Kormanyos, B., P. Ostdiek, W. Bishop, T. Crowe, and G. Rebeiz, A planar wideband 80-200 GHz subharmonic receiver, *Microwave Theory and Techniques, IEEE Transactions on*, *41*(10), 1730 –1737, doi:10.1109/22.247918, 1993.

Kosowsky, A., Cosmic microwave background polarization., *Annals of Physics*, *246*, 49–85, doi:10.1006/aphy.1996.0020, 1996.

Kovac, J. M., Detection of polarization in the cosmic microwave background using DASI, Ph.D. thesis, THE UNIVERSITY OF CHICAGO, 2004.

Kovac, J. M., E. M. Leitch, C. Pryke, J. E. Carlstrom, N. W. Halverson, and W. L. Holzapfel, Detection of polarization in the cosmic microwave background using DASI, *nat*, *420*, 72–787, doi:10.1038/nature01269, 2002.

Kumar, S., Submillimeter wave camera using a novel photon detector technology, Ph.D. thesis, California Institute of Technology, 2007.

Lanting, T., Multiplexed Readout of Superconducting Bolometers for Cosmological Observation, Ph.D. thesis, University of California Berkeley, 2006.

Lanting, T. M., et al., Frequency-domain multiplexed readout of transition-edge sensor arrays with a superconducting quantum interference device, *Applied Physics Letters*, *86*(11), 112511, doi:10.1063/1.1884746, 2005.

Larson, D., et al., Seven-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Power Spectra and WMAP-Derived Parameters, *ArXiv e-prints*, 2010a.

Larson, D., et al., Seven-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Power Spectra and WMAP-Derived Parameters, *ArXiv e-prints*, 2010b.

Lee, A. T., et al., POLARBEAR: Ultra-high Energy Physics with Measurements of CMB Polarization, in *American Institute of Physics Conference Series*, *American Institute of Physics Conference Series*, vol. 1040, edited by H. Kodama & K. Ioka, pp. 66–77, doi: 10.1063/1.2981555, 2008.

Lewis, A., and A. Challinor, Weak gravitational lensing of the CMB, *physrep*, *429*, 1–65, doi:10.1016/j.physrep.2006.03.002, 2006.

Liddle, A. R., An Introduction to Cosmological Inflation, in *High Energy Physics and Cosmology, 1998 Summer School*, edited by R. L. Walsworth & D. F. Phillips, pp. 260–+, 1999.

Liddle, A. R., and D. H. Lyth, *Cosmological Inflation and Large-Scale Structure*, Cambridge University Press, 2000.

Lis, D., CSO Atmospheric Transmission Interactive Plotter, `http://www.submm.caltech.edu/cso/weather/atplot.shtml`, 2010.

Liu, L., H. Xu, R. R. Percy, D. L. Herald, A. W. Lichtenberger, J. L. Hesler, and R. M. Weikle, Development of Integrated Terahertz Broadband Detectors Utilizing Superconducting Hot-Electron Bolometers, *IEEE Transactions on Applied Superconductivity*, *19*, 282–286, doi:10.1109/TASC.2009.2018268, 2009.

Ludwig, A., The definition of cross polarization, *Antennas and Propagation, IEEE Transactions on*, *21*(1), 116 – 119, 1973.

Lyth, D. H., Introduction to Cosmology, *ArXiv Astrophysics e-prints*, 1993.

Matthaei, G., E. Jones, and L. Young, *Microwave Filters, Impedance-Matching Networks, and Coupling Structures*, Artech House Publishers, 1980.

McGinnis, D., and J. Beyer, A broad-band microwave superconducting thin-film transformer, *Microwave Theory and Techniques, IEEE Transactions on*, *36*(11), 1521 –1525, doi:10.1109/22.8916, 1988.

Mehl, J., et al., TES Bolometer Array for the APEX-SZ Camera, *Journal of Low Temperature Physics*, *151*, 697–702, doi:10.1007/s10909-008-9738-1, 2008.

Meng, X., GCA 6200 Wafer Stepper, *Manual Nanoolab Manual section 4.13*, UC Berkeley, Berkeley, Ca, USA, 2010.

Miller, A. D., et al., A Measurement of the Angular Power Spectrum of the Cosmic Microwave Background from L = 100 to 400, *apjl*, *524*, L1–L4, doi:10.1086/312293, 1999.

Mushiake, Y., *Self-Complementary Antennas: Principle of Self-Complementarity for Constant Impedance*, New York Academic Press, 1996.

Myers, M., personal communication, 2008.

Myers, M. J., et al., An antenna-coupled bolometer with an integrated microstrip bandpass filter, *Applied Physics Letters*, *86*(11), 114103, doi:10.1063/1.1879115, 2005.

Nurnberger, M., and J. Volakis, A new planar feed for slot spiral antennas, *Antennas and Propagation, IEEE Transactions on*, *44*(1), 130 –131, doi:10.1109/8.477538, 1996.

O'Brient, R., et al., A Multi-Band Dual-Polarized Antenna-Coupled TES Bolometer, *Journal of Low Temperature Physics*, *151*, 459–463, doi:10.1007/s10909-007-9698-x, 2008a.

O'Brient, R., et al., Sinuous antennas for cosmic microwave background polarimetry, in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 7020, doi:10.1117/12.788526, 2008b.

Olive, K. A., G. Steigman, and T. P. Walker, Primordial nucleosynthesis: theory and observations, *physrep*, *333*, 389–407, doi:10.1016/S0370-1573(00)00031-4, 2000.

Pardo, J., J. Cernicharo, and E. Serabyn, Atmospheric transmission at microwaves (ATM): an improved model for millimeter/submillimeter applications, *Antennas and Propagation, IEEE Transactions on*, *49*(12), 1683 –1694, doi:10.1109/8.982447, 2001.

Partridge, R., *3K: The Cosmic Microwave Background Radiation*, Cambridge University Press, Palo Alto, 1995.

Pathria, R., *Statistical Mechanics, 2nd Edition*, Butterworth-Heinemann, 1996.

Pearson, T. J., et al., The Anisotropy of the Microwave Background to l = 3500: Mosaic Observations with the Cosmic Background Imager, *apj*, *591*, 556–574, doi:10.1086/375508, 2003.

Pisano, G., G. Savini, P. A. R. Ade, V. Haynes, and W. K. Gear, Achromatic half-wave plate for submillimeter instruments in cosmic microwave background astronomy: experimental characterization, *ao*, *45*, 6982–6989, doi:10.1364/AO.45.006982, 2006.

Pozar, D., *Microwave Engineering*, Wiley, 2004.

Raman, S., and G. Rebeiz, Single- and dual-polarized millimeter-wave slot-ring antennas, *Antennas and Propagation, IEEE Transactions on*, *44*(11), 1438 –1444, doi:10.1109/8.542067, 1996.

Rauscher, C., Efficient design methodology for microwave frequency multiplexers using infinite-array prototype circuits, *Microwave Theory and Techniques, IEEE Transactions on*, *42*(7), 1337 –1346, doi:10.1109/22.299727, 1994.

Reichardt, C. L., et al., High-Resolution CMB Power Spectrum from the Complete ACBAR Data Set, *apj*, *694*, 1200–1219, doi:10.1088/0004-637X/694/2/1200, 2009.

Richards, P. L., Bolometers for infrared and millimeter waves, *Journal of Applied Physics*, *76*(1), 1–24, doi:10.1063/1.357128, 1994.

Rumsey, V., *Frequency Independent Antennas*, New York Academic Press, 1966.

Rybicki, G. B., and A. P. Lightman, *Radiative processes in astrophysics*, Wiley, 1979.

Saini, L., and R. Bradley, The Sinuous Antenna - a Dual Polarized Element for Wideband Phased Array Feed Application, *Internal Technical Memo Electronic Division Internal Report-No. 31*, National Radio Astronomy Observatory, Green Bank, W. Va, USA, 1996.

Sarpeshkar, R., R. Lyon, and C. Mead, *A Low-Power Wide-Dynamic-Range Analog VLSI Cochlea*, vol. 447, pp. 49–103, Springer, doi:10.1007/b102308, 2007.

Simons, R., *Coplanar Waveguide Circuits Components and Systems*, Wiley-IEEE Press, 2001.

Smoot, G. F., M. V. Gorenstein, and R. A. Muller, Detection of anisotropy in the cosmic blackbody radiation, *Physical Review Letters*, *39*, 898–901, doi:10.1103/PhysRevLett.39.898, 1977.

Smoot, G. F., et al., Structure in the COBE differential microwave radiometer first-year maps, *apjl*, *396*, L1–L5, doi:10.1086/186504, 1992.

Staniszewski, Z., et al., Galaxy Clusters Discovered with a Sunyaev-Zel'dovich Effect Survey, *apj*, *701*, 32–41, doi:10.1088/0004-637X/701/1/32, 2009.

Swanson, D., Thin-film lumped-element microwave filters, in *Microwave Symposium Digest, 1989., IEEE MTT-S International*, pp. 671 –674 vol.2, doi:10.1109/MWSYM.1989.38814, 1989.

Takahashi, Y., personal communication, 2008.

Van Duzer, T., *Principles of Superconducting Devices and Circuits*, Prentice Hall, 1998.

Vieira, J. D., et al., Extragalactic millimeter-wave sources in South Pole Telescope survey data: source counts, catalog, and statistics for an 87 square-degree field, *ArXiv e-prints*, 2009.

Watts, L., Cochlear Mechanics, Analysis and Analog VLSI, Ph.D. thesis, California Institute of Technology, 1993.

Weisend, J., *The Handbook Of Cryogenic Engineering*, CRC Press, 1998.

Werthamer, N. R., Theory of the Superconducting Transition Temperature and Energy Gap Function of Superposed Metal Films, *Phys. Rev.*, *132*(6), 2440–2445, doi:10.1103/PhysRev.132.2440, 1963.

Woodcraft, A. L., and A. Gray, A low temperature thermal conductivity database, in *A low temperature thermal conductivity database*, vol. 1185, edited by B. Young, B. Cabrera, and A. Miller, pp. 681–684, AIP, doi:10.1063/1.3292433, 2009.

Woody, D. P., and P. L. Richards, Spectrum of the cosmic background radiation, *Physical Review Letters*, *42*, 925–929, doi:10.1103/PhysRevLett.42.925, 1979.

Yassin, G., and S. Withington, Loss In Normal And Superconducting Millimetre-Wave And Submillimetre Wave Microstrip Transmission Line, in *Computation in Electromagnetics, Third International Conference on (Conf. Publ. No. 420)*, pp. 149 –154, 1996.

Yoon, K. W., et al., Feedhorn-Coupled TES Polarimeters for Next-Generation CMB Instruments, in *American Institute of Physics Conference Series*, *American Institute of Physics Conference Series*, vol. 1185, edited by B. Young, B. Cabrera, & A. Miller, pp. 515–518, doi:10.1063/1.3292392, 2009.

Zmuidzinas, J., Cramér-Rao sensitivity limits for astronomical instruments: implications for interferometer design, *Journal of the Optical Society of America A*, *20*, 218–233, doi:10.1364/JOSAA.20.000218, 2003a.

Zmuidzinas, J., Thermal noise and correlations in photon detection, *ao*, *42*, 4989–5008, doi:10.1364/AO.42.004989, 2003b.

Zmuidzinas, J., personal communication, 2010.

Appendix A

MATLAB raytracing software

A.1 Overview

This appendix contains a hardcopy of the raytracing script and all functions. This code is included here as a reference and to ensure that it is not lost. It was written with Matlab v 7.0. The function progmeter.m is need as well, but was written by another author and is available for download from http://www.mathworks.com/matlabcentral/fileexchange.

A.2 Main script

```
%BEAMSCRIPT
%
%This script modifies the far-field antenna patterns from ADS-momentum to
%account for a contacting extended hemispherical lens.  It accounts for
%refraction at the lens surface as well as diffraction.  The user should
%modify the *non-indented* variables at the begining of the script between
%the BEGIN and END USER DEFINED INPUT
%
%This script uses the following functions which must be present in the same
%directory for it to function properly: build_lens.m, raytrace.m,
%readfff.m, normalize.m, rect2plane_incidence.m, plane_incidence2rect.m,
%rect2sphere.m, sphere2rect.m, refraction_nocoat.m, refraction_1coat.m,
%refraction_3coat.m, matrix_mult4d.m, surfacecurrents.m, Diffraction.m,
%progmeter.m, writefff.m
%
%Because the program was developed on a laptop with 512MB RAM, there was
%insufficient memory to store all variables at once, so a series of loops
%are used instead, with a timer so the user can watch progress.  If all
%discritization is done at 3 degrees, then the integral takes 30 sec to
%finish.
%Special thanks to Jen Edwards for help troubleshooting this code, pointing
%out that Matlab's dot function conjugates it's first argument, and
%suggesting the use of image currents.
%Roger O'Brient         Jan 2010

home;
```

```
clear all;

c=3*10^8;%[m/s] speed of light free space
eta=377;%[Ohms] impedance free space

f_GHz=150; %[GHz] Frequency
f=f_GHz*10^9; %[Hz]
f_str=[num2str(round(f_GHz))];

%Lens Properties
er=11.7; % rel permativity of lens material (silicon)
%er=10.5;%rel permativity if lens material (sintered alumina)
    nlens=sqrt(er); %index refraction of lens material
R=13.7/2; %[mm] Lens Radius
    R=R/1000; %[m] Lens Radius
%Lext=.2767*R; %[mm] Hyperhemisphere for er=11.7
Lext=.3876*R; %[mm] Synth ellipse for er=11.7

%Change coating-flag for the three posible coatings:
%coatingflag='no coating';
coatingflag='single layer';
%coatingflag='three layers';

%Single Layer AR-coating properites
thickAR=0.30/2; %[mm] Currently lambda/4 for 300GHz
    thickAR=thickAR/1000; %[m]
nAR=2; %Index of stycast-2850, Lamb compendium data, 4.8K, 100GHz

%Multi-layer AR properites.  I've set all thicknesses to lambda/4 for
%160GHz center frequency
thick1=0.19; %[mm] TMM6
thick2=0.27; %[mm] TMM3
thick3=0.3906; %[mm] Zitex
    thick1=thick1/1000; %[m]
    thick2=thick2/1000; %[m]
    thick3=thick3/1000; %[m]
n1=2.45; %index refraction of TMM6, from Erin Quealy
n2=1.73; %index refraction of TMM3, from Erin Quealy
n3=1.2; %%index refraction of Zitex, from Dominec Benford et all.

%antenna location relative to hemispherical center
centerx=0; %[mm]
centery=0; %[mm]
centerz=-Lext; %[mm]
    ant_loc=[centerx;centery;centerz] ;

%To include image currents, set inc_img=1.  To exclude, set it to 0
inc_img=1 ;

%Angular steps for integration and far field pattern display
%Code takes 30-40 sec with all set to 3 deg.
hemthetastep=3; %[deg]
hemphistep=3; %[deg]
    hemthetastep=pi/180*hemthetastep; %[rad]
    hemphistep=pi/180*hemphistep; %[rad]
% PATTERN THETA/PHI
ffthetastep=3; %[deg]
ffphistep=3; %[deg]
    ffthetastep=pi/180*ffthetastep; %[rad]
```

```
    ffphistep=pi/180*ffphistep; %[rad]

%Input file from ADS momentum
pathin='TestResults\';
fnamein=[ 'Momentum_input.fff' ] ;

%Output file to write to.
pathout=pathin;
fnameout=['Sample_output_', f_str, 'GHz.txt'];

commentary = { [ 'Frequency = ' num2str(f_GHz), ' GHz' ];
               [ 'er = ', num2str(er,'%2.2f') ] ;
               [ 'Lens Radius = ' , num2str(R*1e3,'%2.4f'), ' mm, ' ,...
                 'Extension Length = ' , num2str(Lext*1e3,'%2.4f'),' mm '];
               [ 'antenna at x = ', num2str(centerx*1e3,'%2.6f'),' mm, '...
                 ' y = ', num2str(centery*1e3,'%2.6f'),...
                 ' mm relative to lens central axis' ] ;
                 [ 'Antireflection model: ', coatingflag]; }   ;



disp('Performing Raytracing inside the lens') ;
tic ;

%Diffraction happens at the outer-most surface, so we need to adjust the
%radius of the lens surface according to the number of layers in the
%AR-coating.
switch coatingflag
    case {'no coating'}
        Rsurf=R;
    case {'single layer'}
        Rsurf=R+thickAR;
    case {'three layers'}
        Rsurf=R+thick1+thick2+thick3;
end
%Calculate lens geometry:
%hemtheta,hemphi: angular postions of surface patches to hemispherical
%       center
%normal=unit vector normal to surface.
%anttheta,antphi: angular poistions of surface patches to antenna position
%dA, dA_img: patch area & image patch areas
%dist: distance to patch from antenna
%patch_pos img_pos: vectorial positions of patches relative to antenna.
%khat= unit wavevector incident to each patch
[hemphi hemtheta antphi anttheta normal dA dA_img patch_pos img_pos...
    distance khat]=build_lens(hemthetastep, hemphistep, Rsurf, ant_loc);

%Calculate the fields internal to the lens surface.  Use data from the
%provided momentum file as well as geometry calculated above.
%antE & antH are the fields radiated from the antenna just inside the lens
%surface patches.
lambda=c/nlens/f; %[m] wavelength inside lens
[antE antH]=raytrace([pathin,fnamein],anttheta,antphi,patch_pos,...
            lambda,nlens);

%Convert the internal fields to the basis wrt to the plane of incidence in
%preparation for refraction.
%TE_hat & TM_hat are unit vectors perpendicular to & within the plane of
%       incidence, both normal to k_hat
```

```
[TM_hat,TE_hat,Etransverse,Htransverse]=rect2transverse(normal,khat,...
                                        antE,antH);

%refractErect & refractHrect are electric and magnetic fields just outside
%the lens surface, in a rectangular basis
%The transmission coefficients are different depending on the style of
%AR-coating.  The three cases are:
switch coatingflag
    case {'no coating'}
        [refractErect,refractHrect] =refraction_nocoat(Etransverse,...
            Htransverse,khat,TM_hat,TE_hat,normal,nlens,1);
        disp('Refraction through an uncoated surface');
    case {'single layer'}
        [refractErect,refractHrect] =refraction_1coat(Etransverse,...
            Htransverse,khat,TM_hat,TE_hat,normal,thickAR,nlens,nAR,1,f);
        disp('Refraction through a single layer AR-coating');
    case {'three layers'}
        [refractErect,refractHrect]=refraction_3coat(Etransverse,...
         Htransverse,khat,TM_hat,TE_hat,normal,nlens,n1,n2,n3,1,thick1,...
         thick2,thick3,f);
        disp('Refraction through a triple-layer AR-coating');
end

%Calculate surface currents (J,M) and their images (J_img,M_img).
[J,M,J_img,M_img]=surfacecurrents(refractErect,refractHrect,normal,Lext);
disp(['Finished after: ',num2str(toc),' sec']);

%Do the Diffraction Integral to calculate far-fields (Efarfield, Hfarfield)
%at the angular positions (fftheta,ffphi)
[fftheta ffphi Efarfield Hfarfield]=Diffraction(ffthetastep,ffphistep,...
    J,M,J_img,M_img,dA,dA_img,patch_pos,img_pos,inc_img,Lext,f,...
                                                hemtheta,hemphi);

writefff(180/pi*ffphi,180/pi*fftheta,Efarfield,[pathout,fnameout],...
                                                    commentary);
```

A.3 Construct Lens Geometry

```
function [hemphi hemtheta antphi anttheta normal dA dA_img patch_pos...
    img_pos distance khat]=build_lens(hemthetastep, hemphistep, R, ant_loc)

%BUILD_LENS      build_lens(hem_theta_step, hem_phi_step, R, ant_loc)
%
%build_lens calculates lens geometry before any physics.
%This function accepts as arguments the angular step sizes for meshing the
%surface (hemthetastep & hemphistep) as well as lens radius (R) and the
%vectorial offset of the hemisphere's center from the antenna (ant_loc).
%It returns the folling in meshgrid format:
%HEMPHI & HEMTHETA: patch angular positions wrt center
%NORMAL: unit normal vectors to each patch
%dA: area if each patch, repeated for all 3-dimensions
%dA_img: area of image patchs
%PATCH_POS: vector position of each patch wrt antenna
%img_pos: location of image patches
```

```
%DISTANCE: distance to each patch from antenna
%KHAT: unit angle of incidence
%ANTPHI & ANTTHETA: patch angular positions wrt antenna
%All output vectors are in rectangular basis.
%Roger O'Brient              Oct 2009
%                            updated for image currents Jan 2010


%construct phi&theta coords on the hemisphere.  Exclude theta=0 (tip) and
%phi=360 (duplicate points).
[hemphi,hemtheta] = meshgrid(0:hemphistep:(2∗pi−hemphistep),...
    hemthetastep:hemthetastep:pi/2);
numtheta=size(hemtheta,1);
numphi=size(hemphi,2);

%unit normal
normal=cat(3,sin(hemtheta).∗cos(hemphi),...
             sin(hemtheta).∗sin(hemphi),...
             cos(hemtheta));

Lext=ant_loc(3);
%area of each patch & of image patches
dA=R^2.∗sin(hemtheta);
dA=repmat(dA,[1 1 3]);
dA_img=dA;
%location of each patch
patch_pos=R∗normal−...
            repmat(permute(ant_loc,[3 2 1]),[numtheta,numphi,1]);

%img_pos = patch_pos(1:end−(Lext==0),:,:) ;
img_pos=patch_pos;
img_pos(:,:,3)=−img_pos(:,:,3);

distance=repmat(sqrt(dot(patch_pos,patch_pos,3)),[1 1 3]);
khat = patch_pos ./ distance ;

%atan is defined on [−90,90], but we need it over the full [0,360]
antphi=atan(khat(:,:,2)./khat(:,:,1))+...
    pi∗((khat(:,:,1)<0)&(khat(:,:,2)>0))+...
    pi∗((khat(:,:,1)<0)&(khat(:,:,2)<0))+...
    2∗pi∗((khat(:,:,1)>0)&(khat(:,:,2)<0));
anttheta=acos((khat(:,:,3)));
```

A.4 Construct Fields just inside lens

```
function [antE antH]=raytrace(fff_file,anttheta,antphi,patch_pos,lambda,n)

%RAYTRACE raytrace(fff_file,anttheta,antphi,patch_pos,normal,lambda,khat,n)
%
%This fucntion calculates the internal fields of the lens just inside the
%surface.  It reads an ADS momentum generated file fff_file='∗.fff' and
%interpolates electric field values at patches at (ANTTHETA,ANTPHI).  It
%also accepts as input the rectangular location of each patch PATCH_POS,
%the wavelength in the material LAMBDA, and the wavespeed n to construct a
```

```
%propagator (greens function) that accounts for phase delays and 1/R field
%decay between the antenna and surface patches.  It returns NxMx3 arrays of
%field antE and antH in rectangular coords.  Note that the field must be
%the 1st argument of any cross product since Matlab conjugates the
%2nd argument.
%This function uses other functions NORMALIZE and SPHERE2RECT.
%
%Roger O'Brient     Jan 2010

eta=377; %[ohms]
%construct normalized wavevector
khat=cat(3,sin(anttheta).*cos(antphi),...
           sin(anttheta).*sin(antphi),...
           cos(anttheta));
ki=2*pi/lambda*khat;%[1/m] wavevector

%Read simulation file, interpolate field @ lens, convert to rect coords.
[simtheta,simphi,simEtheta,simEphi] = readfff(fff_file) ;
interpEphi = interp2(simphi,simtheta,simEphi,antphi,anttheta) ;
interpEtheta = interp2(simphi,simtheta,simEtheta,antphi,anttheta) ;
interpEsphere=cat(3,zeros(size(anttheta)),interpEtheta,interpEphi);
interpErect=sphere2rect(interpEsphere,anttheta,antphi);

%Construct propagator to account for phase delays to lens and 1/R field
%decay
R=repmat(sqrt(dot(patch_pos,patch_pos,3)), [1 1 3]);
propagator = repmat( exp( -j* dot( ki,patch_pos, 3 ) ), [ 1 1 3 ] ) ./R ;

%Now consctruct E and H fields at lens.
antE=interpErect.*propagator;
antH=-n*cross(antE,khat,3)/eta;
```

```
function [B]=normalize(A)
%Accepts an array of vectors (mxnx3), calculates the magnitude
%of each vector, and then divides that out to return an array of
%vectors that each have unit magnitude.  The vectors can be in
%any basis.
%Roger O'Brient Oct 07
Amag=repmat(sqrt(dot(A,A,3)),[1 1 3]);
B=A./Amag;
```

A.5 File reading & writing

```
function [theta phi Etheta Ephi] = readfff(fname)

fid=fopen( fname ) ;
if fid==-1
    error(['error: cannot find file ',fname]);
end

data=[];

%fget1 reads a line, ignores new line character
```

```
line=fgetl(fid);


%feof=1 if at end of file, 0 otherwise.
while ¬feof(fid)
    %igonre if the line is empty or begins with '#', otherwise,
    %concatinate with the data array as the next row.
    if ¬isempty( line ) && ¬strcmp( line(1), '#')
        %convert line string into a vector of floating point variables
        data = cat( 1, data, (sscanf( line, '%f' ))' ) ;
    end

    %read next line before repeat
    line=fgetl(fid);
end

%Data format is columns of:
%theta phi real(E_theta) imag(E_theta) real(E_phi) imag(E_phi)

%Now search the 1st two columns & ignore all the repeats
theta=unique(data(:,1))⋆pi/180 ;
phi=unique(data(:,2))⋆pi/180 ;

%Combine the real and imag components & put in meshgrid form
Etheta=reshape(complex(data(:,3),data(:,4)),...
                    length(theta),length(phi));
Ephi=reshape(complex(data(:,5),data(:,6)),...
                    length(theta),length(phi));
fclose(fid);
```

```
function wr=writefff(phi,theta,Efarfield,fname,commentary)
if nargin==5
    commentary={};
end

%put the data in ⋆.fff format so it could be fed back into ADS momentum
ffEtheta=Efarfield(:,:,2);
ffEphi=Efarfield(:,:,3);
dataout=permute(cat(3,theta,phi,...
                    real(ffEtheta),imag(ffEtheta),...
                    real(ffEphi),imag(ffEphi)),...
                                                  [3 1 2]);
fid=fopen(fname,'wt');
%first write comments, each line proceeded by a '#'
for i=1:length(commentary) ;
    fprintf(fid,'# %s \n',commentary{i});
end

%Write data.  Speparate each theta cut with 'Begin Cut' & 'End Cut' as
%Momentum does.
for i=1:size(phi,2)
    fprintf(fid,'# %s \n','Begin cut');
        fprintf(fid,'%2.8f  %2.8f  %2.8e  %2.8e  %2.8e  %2.8e  \n', ...
                dataout(:,:,i));
    fprintf(fid,'# %s \n\n','End cut');
end
fclose(fid);
```

A.6 Convert to & from spherical coordinates

```
function v_sphere=rect2sphere(v_rect,theta, phi)
%This converts an array of 3-d vectors in a rectangualr basis to a
%spherical basis.  All angles are in RADIANS.  The theta and
%phi matrices should have been generated by meshgrid of
%dimensions mxn, while v_sphere should be mxnx3.  The program
%constructs |r><x|+|r><y|+|r><z| etc and dots this against the
%vectors in an xyz basis: V_r|r≥(|r><x|+|r><y|+|r><z|)|v_rect>
%etc.
%Roger O'Brient Oct 07

rhat=cat(3, sin(theta).*cos(phi), sin(theta).*sin(phi)...
    ,cos(theta));  %|r><x|+|r><y|+|r><z|
thetahat=cat(3, cos(theta).*cos(phi), cos(theta).*sin(phi)...
    ,-sin(theta)); %|theta><x|+|theta><y|+|theta><z|
phihat=cat(3, -sin(phi), cos(phi),0*phi);
                        %|phi><x|+|phi><y|+|phi><z|

v_sphere=cat(3,...
    dot(rhat,v_rect,3),... %V_r|r≥(|r><x|+|r><y|+|r><z|)|v_rect>
    dot(thetahat,v_rect,3),...%V_th|th≥(|th><x|+|th><y|+|th><z|)|v_rect>
    dot(phihat,v_rect,3)); %V_ph|ph≥(|ph><x|+|ph><y|+|ph><z|)|v_rect>
```

```
function vrect=sphere2rect(vsphere,theta, phi)
%This converts an array of 3-d vectors in a spherical basis to a
%rectangular basis.  All angles are in RADIANS.  The theta and
%phi matrices should have been generated by meshgrid of
%dimensions mxn, while v_sphere should be mxnx3.  The program
%constructs |x><r|+|x><th|+|x><ph| etc and dots this against the
%vectors in an xyz basis: V_x|x≥(|x><r|+|x><th|+|x><ph|)|v_sph>
%etc.
%Roger O'Brient Oct 07

xhat=cat(3, sin(theta).*cos(phi), cos(theta).*cos(phi)...
    ,-sin(phi));  %|x><r|+|x><th|+|x><ph|
yhat=cat(3, sin(theta).*sin(phi), cos(theta).*sin(phi)...
    ,cos(phi)); %|y><r|+|y><th|+|y><ph|
zhat=cat(3, cos(theta), -sin(theta),zeros(size(phi)));
                        %|z><r|+|z><th|+|z><ph|

vrect=cat(3,...
          dot(xhat,vsphere,3),... %V_x|x≥(|x><r|+|x><th|+|x><ph|)|v_sph>
          dot(yhat,vsphere,3),...%V_y|y≥(|y><r|+|y><th|+|y><ph|)|v_sph>
          dot(zhat,vsphere,3)); %V_z|z≥(|z><r|+|z><th|+|z><ph|)|v_sph>
```

A.7 Convert to & from POI coordinates

```
function [TM_hat,TE_hat,Eplane,Hplane]=rect2transverse(normal,k_hat,...
                                                        Erect,Hrect)
%            This function accepts unit normal (NORMAL) and unit incident
```

```
%           (KHAT) vectors and constructs a basis perpendicular (TE_HAT)
%           and within the plane of incidence (TM_HAT).  The basis is
%               TM_HAT=TE_HAT X KHAT
%
%           TE stands for Transverse-Electric component while TM for
%           Transvere-Magnetic component.
%           It then resolves the provided E and H vectors into components
%           parallel (E_TM) and perpendicular (E_TE), where the output
%           format is an mxnx3 matrix.  The mxn columns refer to specific
%           angles of the lens surface patchs and the components on the
%           3-element dimension are:
%               1. along K_HAT (which will be zero)
%               2. along TM_HAT
%               3. along TE_HAT
%           The function returns the two new basis vectors as well as the
%           fields in that basis.
%           This function calls the custom function NORMALIZE which forces a
%           matrix of 3-vectors to be normal.
%           WARNING: Matlab's native "dot" function takes the conjugate of
%           it's first argument, and "cross" conjugates the second!
%           Roger O'Brient Oct 07

%set up the new basis.
normal=normalize(normal);
k_hat=normalize(k_hat);
TE_hat=cross(normal,k_hat,3);
TE_hat=normalize(TE_hat);
TM_hat=cross(TE_hat,k_hat,3);
TM_hat=normalize(TM_hat);

%Find components in new basis.  Basis vectors must be the first argument
%since they are real and Matlab will automatically conjugate those.
E_k=dot(k_hat,Erect,3);
E_tm=dot(TM_hat,Erect,3);
E_te=dot(TE_hat,Erect,3);

H_k=dot(k_hat,Hrect,3);
H_te=dot(TM_hat,Hrect,3);
H_tm=dot(TE_hat,Hrect,3);

%construct vectors in new basis as described in header
Eplane=cat(3,E_k,E_tm,E_te);
Hplane=cat(3,H_k,H_te,H_tm);
```

```
function [Erect,Hrect]=transverse2rect(normal,k_hat,...
                                        Etransverse,Htransverse)
%This function accepts unit normal (NORMAL) and unit incident
%(KHAT) vectors and constructs a basis perpendicular (TE_HAT)
%and within the plane of incidence (TM_HAT).  These names are in reference
%to the orientation of the electric fields of those componenents; e.g the
%electric field of the TM-component resides in the plane of incidence.
%The basis is
%       TM_HAT=TE_HAT x K_HAT
%and all vectors are ordered:
%       1.k_hat
%       2.TM_hat
%       3.TE_hat
```

```
%It then resolves the E and H vectors provided by the user in
%that basis back into a rectangular basis, where the output
%format is an mxnx3 matrix.  The mxn columns refer to specific
%anglular positions of the lens surface patchs and the components on the
%3-element dimension are:
%       1. x_hat
%       2. y_hat
%       3. z_hat
%This function calls the custom function NORMALIZE which forces a
%matrix of 3-vectors each be of unit length.
%WARNING: Matlab's native "dot" function takes the conjugate of
%it's first argument, and "cross" conjugates the second!
%Roger O'Brient Oct 2009

%Set up the new basis.  These are in cartesian coordinates.
normal=normalize(normal);
k_hat=normalize(k_hat);
TE_hat=cross(normal,k_hat,3);
TE_hat=normalize(TE_hat);
TM_hat=cross(TE_hat,k_hat,3);
TM_hat=normalize(TM_hat);

%extract components in the transverse-basis.
E_k=repmat(Etransverse(:,:,1),[1 1 3]);
E_tm=repmat(Etransverse(:,:,2),[1 1 3]);
E_te=repmat(Etransverse(:,:,3),[1 1 3]);

H_k=repmat(Htransverse(:,:,1),[1 1 3]);
H_te=repmat(Htransverse(:,:,2),[1 1 3]);
H_tm=repmat(Htransverse(:,:,3),[1 1 3]);

%Assemble vectors in rectangular coordinates.
Erect=E_k.*k_hat+E_tm.*TM_hat+E_te.*TE_hat;
Hrect=H_k.*k_hat+H_te.*TM_hat+H_tm.*TE_hat;
```

A.8 Refraction

There are three options: no coating, one-layer coating, and three layers. For brevity's sake (too late...), I have excluded the one-layer function which is similar to the three-layer.

```
function [transErect,transHrect] =refraction_nocoat(Etransverse,Hplane,...
    k_hat,TM_hat,TE_hat,normal,nlens,noutside)
%REFRACTION_NOCOAT refraction_nocoat(Etransverse,Hplane,...
%           k_hat,TE_hat,TM_hat,normal,nlens,noutside)
%This function reftacts the incident fields (EPLANE,HPLANE) in
%Plane-Of-Incidence (POI) coords into refracted fields
%(TRANSERECT,TRANSHRECT), rectangular coords.  All are mxnx3 arrays.
%It accepts the POI basis vectors (KHAT,TM_hat,TE_hat) as well as the
%surface normals NORMAL and the indicies inside (N1) and outside (N2) the
%lens.
%This function invokes Snell's law to calculate the new unit wavevector
%khatprime.  It calculates Fressnel Coefficients, and then the trnsmitted
%fields.  The TE component of the E-fields remains in the same
%positon, as does the TM component of the H-fields.  The others rotate
```

```
%about the basis vectors TE_hat.   HTRANS is calculated from ETRANS and
%khatprime.
%Roger O'Brient Jan 2010

eta=377; %[Ohms] Impedance Free Space

%Snell's law rotates pointing vector in khat-TM_hat plane by
%angleDelta:
angle_inc=acos(dot(k_hat,normal,3)); %[radians]
angle_trans=asin(nlens/noutside*sin(angle_inc)); %[radians]
angleDelta=angle_trans-angle_inc; %[radians]
%The wavevector for the transmitted ray is just a rotation of the incident
%wavevector by angle_Delta about the TE_hat axis.  transk_hat will be part
%of a new basis-set for the fields.
transk_hat=repmat(cos(angleDelta),[1 1 3]).*k_hat+...
    repmat(sin(angleDelta),[1 1 3]).*TM_hat;

%Fresnel Coefficients (unitless):
R_te=(nlens*cos(angle_inc)-noutside*cos(angle_trans))./...
    (nlens*cos(angle_inc)+noutside*cos(angle_trans));
T_te=1+R_te;
R_tm=(-noutside*cos(angle_inc)+nlens*cos(angle_trans))./...
    (noutside*cos(angle_inc)+nlens*cos(angle_trans));
T_tm=nlens/noutside*(1-R_tm);

%Transmitted E-field perpendicular to POI (Etransverse(:,:,3)) points in
%same direction before and after refraction.  The E-field within the POI
%Etransverse(:,:,2) rotates like khat by angleDelta

transEplane=cat(3,-T_tm.*Etransverse(:,:,2).*sin(angleDelta),...
                  T_tm.*Etransverse(:,:,2).*cos(angleDelta),...
                  T_te.*Etransverse(:,:,3));

%covert to rectangular coords
[transErect,transHrect]=transverse2rect(normal,k_hat,transEplane,...
                                        transEplane);

%Construct Magnetic-field
transHrect=1/eta*cross(transk_hat,transErect,3);
```

```
function [transErect,transHrect] =refraction_3coat(Etransverse,...
    Htransverse,k_hat,TM_hat,TE_hat,normal,nlens,n1,n2,n3,noutside,...
    thick1,thick2,thick3,f)
%REFRACTION_3COAT refraction_3coat(Etransverse,Htransverse,k_hat,TE_hat,...
%TM_hat,normal,nlens,n1,n2,n3,noutside,thick1,thick2,thick3,f)
%This function refracts the incident fields (ETRANSVERSE,HTRANSVERSE) in
%Plane-Of-Incidence (POI_ coords into refracted fields
%(TRANSERECT,TRANSHRECT).  All are mxnx3 arrays.  It accepts
%the basis vectors (KHAT,TM_HAT,TE_HAT) as well as the surface
%normals NORMAL and the indicies inside (NLEN) and outside (NOUTSIDE) the
%lens.  It also accepts the thicknesses of the 3 layers THICK1-THICK3 and
%their indices N1-N3.  Layer 1 is the inner-most, Layer 3 the outer-most.
%This function invokes Snell's law to calculate the new unit wavevector
%khatprime, which is just a rotation about the TE_hat basis.
%It calculates Fressnel Coefficients for a three layer AR-coating by
%forcing the fields to be continuous at the boundaries between the media,
%and then calculates transmitted fields.  The TE component of the E-fields
```

```
%remains in the same positon, as does the TM component of the H-fields.
%The others rotate about the basis vectors TE_hat just like the wavevector.
%TRANSHRECT is calculated from TRANSERECT and khatprime.
%This function uses the function matrix_mult4d, since matlab cannot
%natively do matrix multiplication on arrays of rank>2.
%Roger O'Brient         Jan 2010

c=3*10^8; %[m/s] Speed light free space
eta=377; %[Ohms] Impedance Free Space

lambda_o=c/f; %[m] wavelength outside lens
ko=(2*pi/lambda_o); %[1/m] wavenumber outside lens

%Snell's law rotates pointing vector in khat-TM_hat plane at every coating
%interface.  The net rotation is angleDelta, which is as if there never was
%no intermediate layers.  But we need the intermediate angles for
%calculating the transmission coefficients.
angle_inc=acos(dot(k_hat,normal,3)); %[radians]

angle1=asin(nlens/n1*sin(angle_inc)); %[radians]
angle2=asin(n1/n2*sin(angle1)); %[radians]
angle3=asin(n2/n3*sin(angle2)); %[radians]

angle_trans=asin(nlens/noutside*sin(angle_inc)); %[radians]
angleDelta=angle_trans-angle_inc; %[radians]
%The wavevector for the transmitted ray is just a rotation of the incident
%wavevector by angle_Delta about the TE_hat axis.  transk_hat can be part
%of a new basis-set for the fields, although I keep all vectors in the
%original incident basis
transk_hat=repmat(cos(angleDelta),[1 1 3]).*k_hat+...
    repmat(sin(angleDelta),[1 1 3]).*TM_hat;

%Construct Transmission coefficients using the well known algorithm
%discussed in section 9.7.1 of Hect's Optics using transfer matrices.
%Construct transfer matrics: [E1;H1]=M*[E2;H2] for each layer and then
%multiply them

Y_lens=nlens/eta*cos(angle_inc);

Y_1_te=n1/eta*cos(angle1);
Y_1_tm=n1/eta*sec(angle1);
Y_2_te=n2/eta*cos(angle2);
Y_2_tm=n2/eta*sec(angle2);
Y_3_te=n3/eta*cos(angle3);
Y_3_tm=n3/eta*sec(angle3);

Y_out=noutside/eta*cos(angle_trans);

h1=n1*thick1*cos(angle1);
h2=n2*thick1*cos(angle2);
h3=n3*thick3*cos(angle3);

M1_te=cat(3,cat(4,cos(ko*h1),              i*sin(ko*h1)./Y_1_te),...
           cat(4,i*sin(ko*h1).*Y_1_te,   cos(ko*h1)));
M1_tm=cat(3,cat(4,cos(ko*h1),              i*sin(ko*h1)./Y_1_tm),...
           cat(4,i*sin(ko*h1).*Y_1_tm,   cos(ko*h1)));

M2_te=cat(3,cat(4,cos(ko*h2),              i*sin(ko*h2)./Y_2_te),...
           cat(4,i*sin(ko*h2).*Y_2_te,   cos(ko*h2)));
```

```
M2_tm=cat(3,cat(4,cos(ko*h2),              i*sin(ko*h2)./Y_2_tm),...
           cat(4,i*sin(ko*h2).*Y_2_tm,   cos(ko*h2)));

M3_te=cat(3,cat(4,cos(ko*h3),              i*sin(ko*h3)./Y_3_te),...
           cat(4,i*sin(ko*h3).*Y_3_te,   cos(ko*h3)));
M3_tm=cat(3,cat(4,cos(ko*h3),              i*sin(ko*h3)./Y_3_tm),...
           cat(4,i*sin(ko*h3).*Y_3_tm,   cos(ko*h3)));

%Multiply the transfer matrices.
Mtot_te=matrix_mult4d(M1_te,matrix_mult4d(M2_te,M3_te));
Mtot_tm=matrix_mult4d(M1_tm,matrix_mult4d(M2_tm,M3_tm));

%Construct transmission coefficients for the E-fields
T_te=2*Y_out./(Y_out.*Mtot_te(:,:,1,1)+...
               Y_out.*Y_lens.*Mtot_te(:,:,1,2)+...
               Mtot_te(:,:,2,1)+...
               Y_lens.*Mtot_te(:,:,2,2));

T_tm=2*Y_out./(Y_out.*Mtot_tm(:,:,1,1)+...
               Y_out.*Y_lens.*Mtot_tm(:,:,1,2)+...
               Mtot_tm(:,:,2,1)+...
               Y_lens.*Mtot_tm(:,:,2,2));

%Transmitted E-field perpendicular to POI (Etransverse(:,:,3)) points in
%same direction before and after refraction.  The E-field within the POI
%(Etransverse(:,:,2))rotates by angleDelta like khat.

transEplane=cat(3,-T_tm.*Etransverse(:,:,2).*sin(angleDelta),...
                  T_tm.*Etransverse(:,:,2).*cos(angleDelta),...
                  T_te.*Etransverse(:,:,3));

%covert to rectangular coords
[transErect,transHrect]=transverse2rect(normal,k_hat,transEplane, ...
                                        transEplane);

%Construct Magnetic-field
transHrect=1/eta*cross(transk_hat,transErect,3);
```

```
function [C]=matrix_mult4d(A,B)
%4DMATRIX_MULT C=matrix_mult4d(A,B)
%This function multiplies two mxnx2x2 arrays A and B in the last two
%indices according to standard matrix multiplication definition.  It
%returns a mxnx2x2 array where each of the mth,nth 2x2 array is the matrix
%product of the corresponding ones from A and B.  Matlab
%does not have a native way of doing this, but this is needed for 2x2
%multiplication in the AR-coating algoerithm in our ray-tracing code.
%
%Roger O'Brient     Feb 2010

C11=A(:,:,1,1).*B(:,:,1,1)+A(:,:,1,2).*B(:,:,2,1);
C12=A(:,:,1,1).*B(:,:,1,2)+A(:,:,1,2).*B(:,:,2,2);
C21=A(:,:,2,1).*B(:,:,1,1)+A(:,:,2,2).*B(:,:,2,1);
C22=A(:,:,2,1).*B(:,:,1,2)+A(:,:,2,2).*B(:,:,2,2);

Ccol1=cat(4,C11,C12);
Ccol2=cat(4,C21,C22);
C=cat(3,Ccol1,Ccol2);
```

A.9 diffraction

```matlab
function [J,M,J_img,M_img]=surfacecurrents(E,H,normal,Lext)
%           Function takes as input mxnx3 arrays of 3-D vectors in
%           rectangualr basis correspondong to Electric Fields E, Magnetic
%           Fields H, and unit normal vectors NORMAL at the lens sruface.
%           It constructs fictitious electrinc J and magnetic currents M on
%           the surface to assist with the Huygens Integral.  These vectors
%           are also in a rectangular coordinate basis.  It also returns
%           the image currents J_IMG & M_IMG, but reflects them to ensure
%           that the electric fields at the ground plane are normal and
%           magnetic fields are tangential.
%           Roger O'Brient          October 2009
%                                   updated for image currents Jan 2010

J= cross(normal,H,3);
M= -cross(normal,E,3);

J_img=J;
J_img(:,:,1:2)=-J_img(:,:,1:2);

M_img=M;
M_img(:,:,3)=-M_img(:,:,3) ;
```

```matlab
function [fftheta ffphi Efarfield Hfarfield]=Diffrcation(ffthetastep,...
    ffphistep,J,M,J_img,M_img,dA,dA_img,patch_pos,img_pos,inc_img,Lext,...
    f,hemtheta,hemphi)
%DIFFRACTION
%This function returns the far-fields (EFARFIELD,HFARFIELD) at angular
%positions (FFTHETA,FFPHI).  It's arguments are theta and phi steps for
%the far field, surface currents and their images, the locations of those
%curretns, patch surface areas, a flag to include the images, and the
%extension length LEXT.
%The code uses a common Fourier Transform algorithm outline in most
%antenna textbooks for radiation through an aperture.  This was developed
%on a laptop with 0.5Gb RAM, which was insuffient to store 5-index arrays
%needed for the diffaction calculations.  So it uses a loop instead and it
%reports progress to the user with the function PROGMETER.  If all angles
%steps are 3deg, then the code takes 30-40sec to execute on my laptop with
%1.8GHz processor.
%Roger O'Brient    Aug2009

c=3*10^8; %[m/s]
eta=377; %[Ohms]
if Lext==0
    remove=1;
else
    remove=0;
end
tic ;
[ffphi,fftheta]=meshgrid(0:ffphistep:(2*pi),0:ffthetastep:(pi/2));
Efarfield=zeros(cat(2,size(ffphi),3));
Hfarfield=zeros(cat(2,size(ffphi),3));
progmeter(0, 'Performing Diffraction Integral')

```

```
for p=1:size(ffphi,2)
    for t=1:size(fftheta,1)
        %Construct the wavevector of the total wave at the far-field
        %angular positon (fftheta,ffphi
        theta=fftheta(t,p);
        phi=ffphi(t,p);
        ffk=2*pi*f/c*cat(3,...
      repmat(sin(theta).*cos(phi),[size(hemtheta,1) size(hemphi,2) 1]),...
      repmat(sin(theta).*sin(phi),[size(hemtheta,1) size(hemphi,2) 1]),...
      repmat(cos(theta),[size(hemtheta,1) size(hemphi,2) 1]));
        %Construct Propagator to far field.  This only accounts for phase
        %difference since intensity decay is roughly the same at all
        %points.  The Propagator for the images is almost the same, except
        %excludes equator points if Lext=0.
        Propagator = repmat(exp(i*sum(ffk.*patch_pos,3)),[1 1 3]) ;
        if inc_img
            Propagator_img=repmat(exp(i*sum(ffk(1:end-remove,:,:).*...
                                                 img_pos,3)),[ 1 1 3 ]);
        end
        %Construct Far-field Magnetic (N) and Electric (L) Vector
        %potentials.  These exclude common factors of phase delay and 1/r
        %field decay.
        Nrect=sum(sum(J.*Propagator.*dA,1),2)+...
            inc_img*sum(sum(J_img.*Propagator_img.*dA_img,1),2);
        Lrect=sum(sum(M.*Propagator.*dA,1),2)+...
            inc_img*sum(sum(M_img.*Propagator_img.*dA_img,1),2);
        N=rect2sphere(Nrect,theta,phi);
        L=rect2sphere(Lrect,theta,phi);

        %Fields are deivatives of the potentials:
        Efarfield(t,p,2)=-(L(:,:,3)+eta*N(:,:,2));
        Efarfield(t,p,3)=(L(:,:,2)-eta*N(:,:,3));
        Hfarfield(t,p,2)=(N(:,:,3)-L(:,:,2)/eta);
        Hfarfield(t,p,3)=-(N(:,:,2)+L(:,:,3)/eta);
    end
    progmeter(phi/(2*pi));
end

%normalize beams to peak
EPower=dot(Efarfield,Efarfield,3);
HPower=dot(Hfarfield,Hfarfield,3);
Efarfield=Efarfield/sqrt(max(max(EPower)));
Hfarfield=Efarfield/sqrt(max(max(HPower)));
% ffEtheta=Efarfield(:,:,2);
% ffEphi=Efarfield(:,:,3);
% ffHtheta=Hfarfield(:,:,2);
% ffHphi=Hfarfield(:,:,3);
progmeter done
disp(['Diffraction Calculation finished after ',num2str(toc),' sec']);
```