Transfer Matrix Optical Modeling

George F. Burkhard and Eric T. Hoke
McGehee Group (Stanford Univ.)
February, 2010

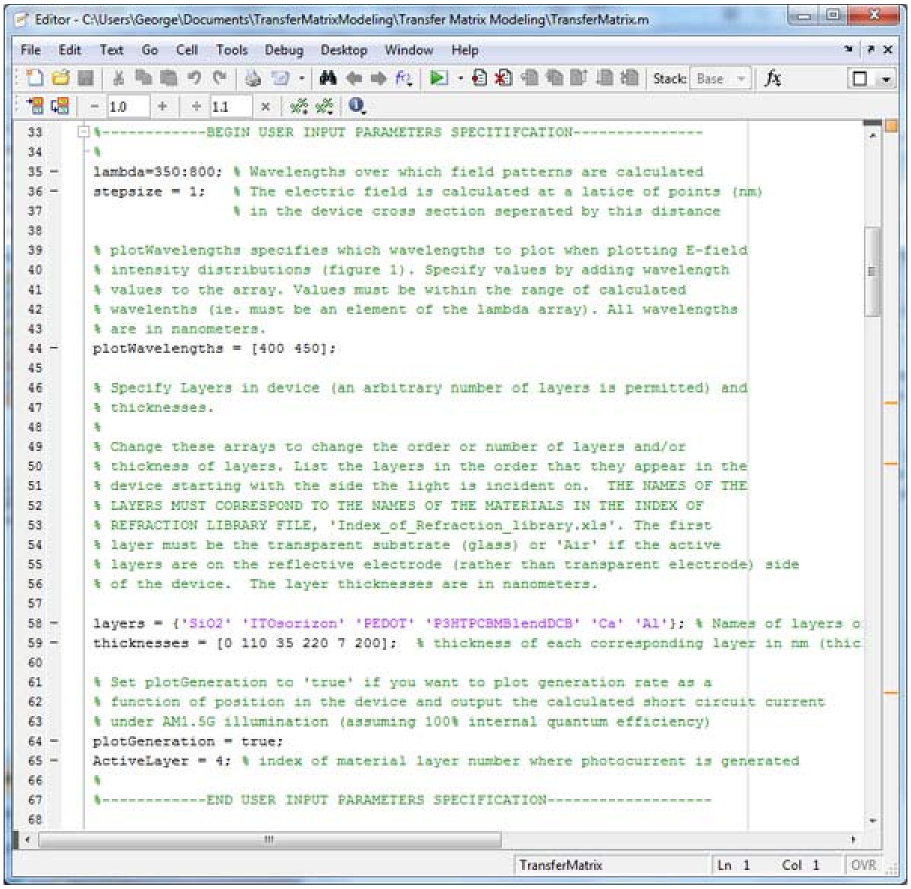
This document describes, briefly, how to use the Matlab script, TransferMatrix.m to calculate optical interference and absorption in multilayer stacks. This software uses the transfer matrix method, where transmission and reflection are calculated for each interface in the stack as well as attenuation in each layer. The theory behind this calculation is described in detail in the literature[1, 2] and will not be covered here. This model assumes normal incidence and does not take into account scattering.

The program consists of three files; the TransferMatrix.m file contains the main program as well as three helper functions. Using the program requires no knowledge of the code itself; however we have made every effort to thoroughly comment the code in case the user wishes to modify it. We will focus on how to use the program as designed to calculate optical fields in multilayers.

User definable parameters

All of the user definable parameters are defined at the top of the script:

lambda	array of wavelengths for which all values will be calculated. wavelength values must lie within the bounds of the wavelengths for which the optical constants (*n* and *k*) are defined in *Index_of_Refraction_library.xls*.
stepsize	numerical constant that defines the stepsize (in nanometers) between each point calculated in the spectrum.
plotWavelengths	array of wavelengths for which to plot interference patterns in the device. to add wavelengths, simply add elements to the vector.
layers	cell array of strings corresponding to the materials that compose each layer in the stack. **these strings must correspond to the names used in the *Index_of_Refraction_library.xls* file (minus the '_n' or '_k' extension).** the order of materials matters! light is incident on the stack "from the left"
thicknesses	vector describing the thicknesses of each layer (in nanometers). the thickness value at a given index in this vector corresponds to the thickness of the layer of the same index in *layers*.
plotGeneration	boolean switch. when true, TransferMatrix will include a plot of the generation rate as a function of thickness within the active layer of the device. if *plotGeneration* is set to *true*, *activeLayer* must be defined.
activeLayer	numerical constant that defines the index of the material responsible for photocurrent generation. this index corresponds to the index of the active material in the *layers* and *thicknesses* vectors. for instance, if *layers* = {'material1' 'material2' 'material3'} and material2 is the active layer, then *activeLayer* should be set to 2.

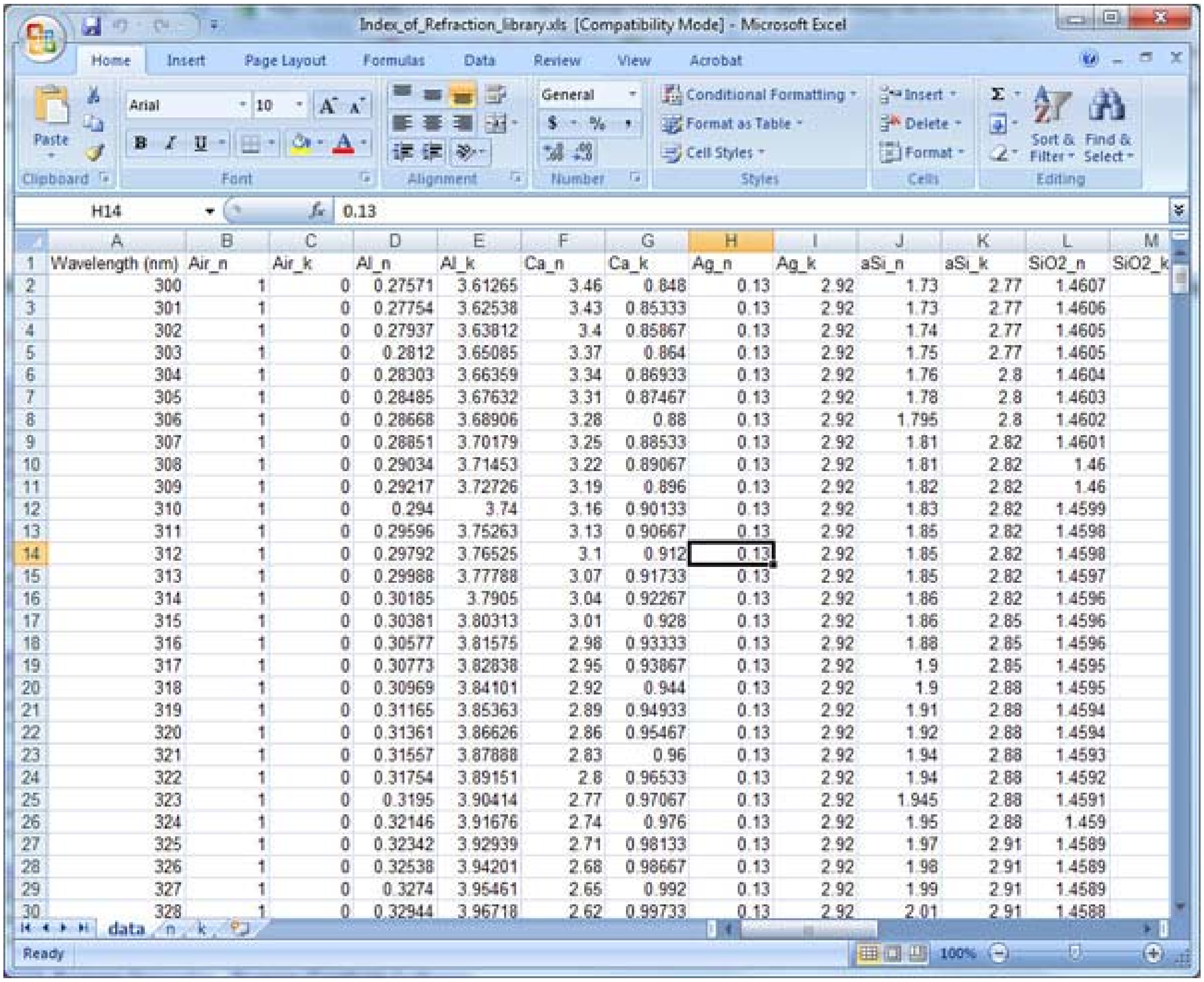


All user definable parameters are set within this section of code.

Index of refraction library file

The optical constant data for each available material is defined in the *Index_of_Refraction_library.xls* file. This file consists of wavelength-dependent optical constant data for each material. *n* and *k* are the real and imaginary parts of the complex index of refraction and are defined for each material at each wavelength. The names of the columns containing each set of data must follow the format, *material_n* and *material_k*, where *material* can be any designation for a material and *_n* and *_k* designate which column refers to n and k respectively. The first column must always be domain of wavelengths for which the material optical constants are defined.

We have provided optical constants for the materials we typically use. **Note: many materials' optical properties depend on processing conditions!** ITO deposited under different conditions can have drastically different optical constants and electrical conductivity (in our case, we purchase ITO coated glass substrates from Sorizon technologies and measured the optical constants using variable angle spectroscopic ellipsometry). Similarly, the optical constants of PEDOT:PSS and the P3HT:PCBM blend depend on doping density (in the case of PEDOT:PSS) and deposition technique, solvent, post-annealing temperature and time, and of course material quality (contamination and regioregularity). Therefore, while these results are typical, you may wish to measure the optical constants of your materials or procure them from a supplier or from the literature.



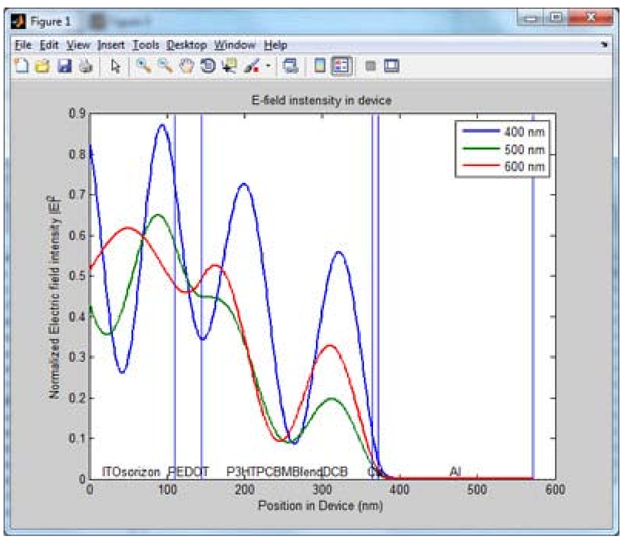
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Wavelength (nm)	Air_n	Air_k	Al_n	Al_k	Ca_n	Ca_k	Ag_n	Ag_k	aSi_n	aSi_k	SiO2_n	SiO2_k
2	300	1	0	0.27571	3.61265	3.46	0.848	0.13	2.92	1.73	2.77	1.4607	
3	301	1	0	0.27754	3.62538	3.43	0.85333	0.13	2.92	1.73	2.77	1.4606	
4	302	1	0	0.27937	3.63812	3.4	0.85867	0.13	2.92	1.74	2.77	1.4605	
5	303	1	0	0.2812	3.65085	3.37	0.864	0.13	2.92	1.75	2.77	1.4605	
6	304	1	0	0.28303	3.66359	3.34	0.86933	0.13	2.92	1.76	2.8	1.4604	
7	305	1	0	0.28485	3.67632	3.31	0.87467	0.13	2.92	1.78	2.8	1.4603	
8	306	1	0	0.28668	3.68906	3.28	0.88	0.13	2.92	1.795	2.8	1.4602	
9	307	1	0	0.28851	3.70179	3.25	0.88533	0.13	2.92	1.81	2.82	1.4601	
10	308	1	0	0.29034	3.71453	3.22	0.89067	0.13	2.92	1.81	2.82	1.46	
11	309	1	0	0.29217	3.72726	3.19	0.896	0.13	2.92	1.82	2.82	1.46	
12	310	1	0	0.294	3.74	3.16	0.90133	0.13	2.92	1.83	2.82	1.4599	
13	311	1	0	0.29596	3.75263	3.13	0.90667	0.13	2.92	1.85	2.82	1.4598	
14	312	1	0	0.29792	3.76525	3.1	0.912	0.13	2.92	1.85	2.82	1.4598	
15	313	1	0	0.29988	3.77788	3.07	0.91733	0.13	2.92	1.85	2.82	1.4597	
16	314	1	0	0.30185	3.7905	3.04	0.92267	0.13	2.92	1.86	2.82	1.4596	
17	315	1	0	0.30381	3.80313	3.01	0.928	0.13	2.92	1.86	2.85	1.4596	
18	316	1	0	0.30577	3.81575	2.98	0.93333	0.13	2.92	1.88	2.85	1.4596	
19	317	1	0	0.30773	3.82838	2.95	0.93867	0.13	2.92	1.9	2.85	1.4595	
20	318	1	0	0.30969	3.84101	2.92	0.944	0.13	2.92	1.9	2.88	1.4595	
21	319	1	0	0.31165	3.85363	2.89	0.94933	0.13	2.92	1.91	2.88	1.4594	
22	320	1	0	0.31361	3.86626	2.86	0.95467	0.13	2.92	1.92	2.88	1.4594	
23	321	1	0	0.31557	3.87888	2.83	0.96	0.13	2.92	1.94	2.88	1.4593	
24	322	1	0	0.31754	3.89151	2.8	0.96533	0.13	2.92	1.94	2.88	1.4592	
25	323	1	0	0.3195	3.90414	2.77	0.97067	0.13	2.92	1.945	2.88	1.4591	
26	324	1	0	0.32146	3.91676	2.74	0.976	0.13	2.92	1.95	2.88	1.459	
27	325	1	0	0.32342	3.92939	2.71	0.98133	0.13	2.92	1.97	2.91	1.4589	
28	326	1	0	0.32538	3.94201	2.68	0.98667	0.13	2.92	1.98	2.91	1.4589	
29	327	1	0	0.3274	3.95461	2.65	0.992	0.13	2.92	1.99	2.91	1.4589	
30	328	1	0	0.32944	3.96718	2.62	0.99733	0.13	2.92	2.01	2.91	1.4588	

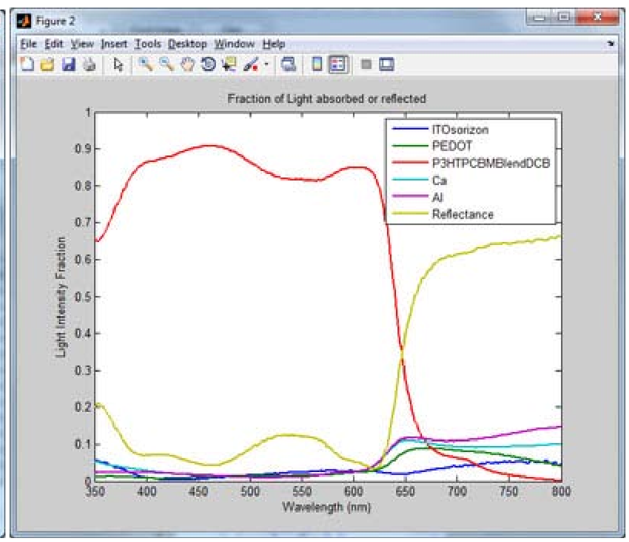
Example calculation

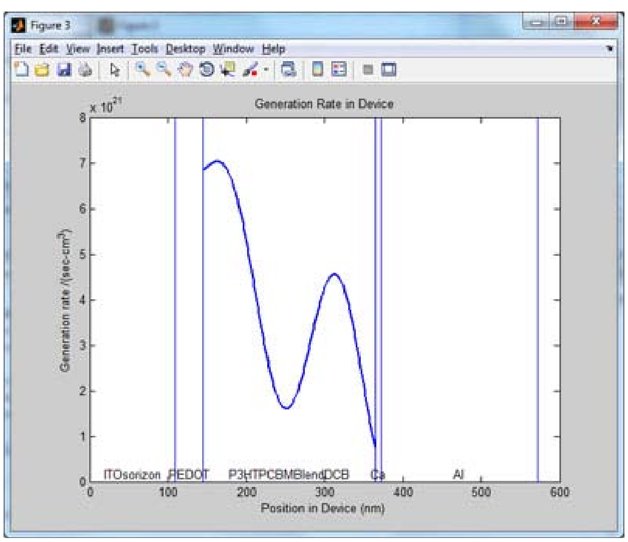
Suppose we want to calculate the optical properties for a typical poly-3-hexylthiophene:[6,6]-phenyl-C_{61}-butyric acid methyl ester (P3HT:PCBM) solar cell. A typical device structure consists of a glass substrate/ITO/PEDOT:PSS/P3HT:PCBM/Ca/Al (where PEDOT:PSS refers to poly(3,4-ethylenedioxythiophene) poly(styrenesulfonate)). The thickness of the glass substrate is not important as it is thick enough that interference effects have practically no effect. Typical thicknesses for the remaining layers are (in nm), 110/35/220/7/200. We are interested in the visible spectrum where the materials absorb (~350 nm - ~800 nm) and suppose we wish to see the interference patterns of 400, 500, and 600 nm light within the device. We would also like to see the position-dependent generation rate in the device. The user-defined parameters would then take the following values:

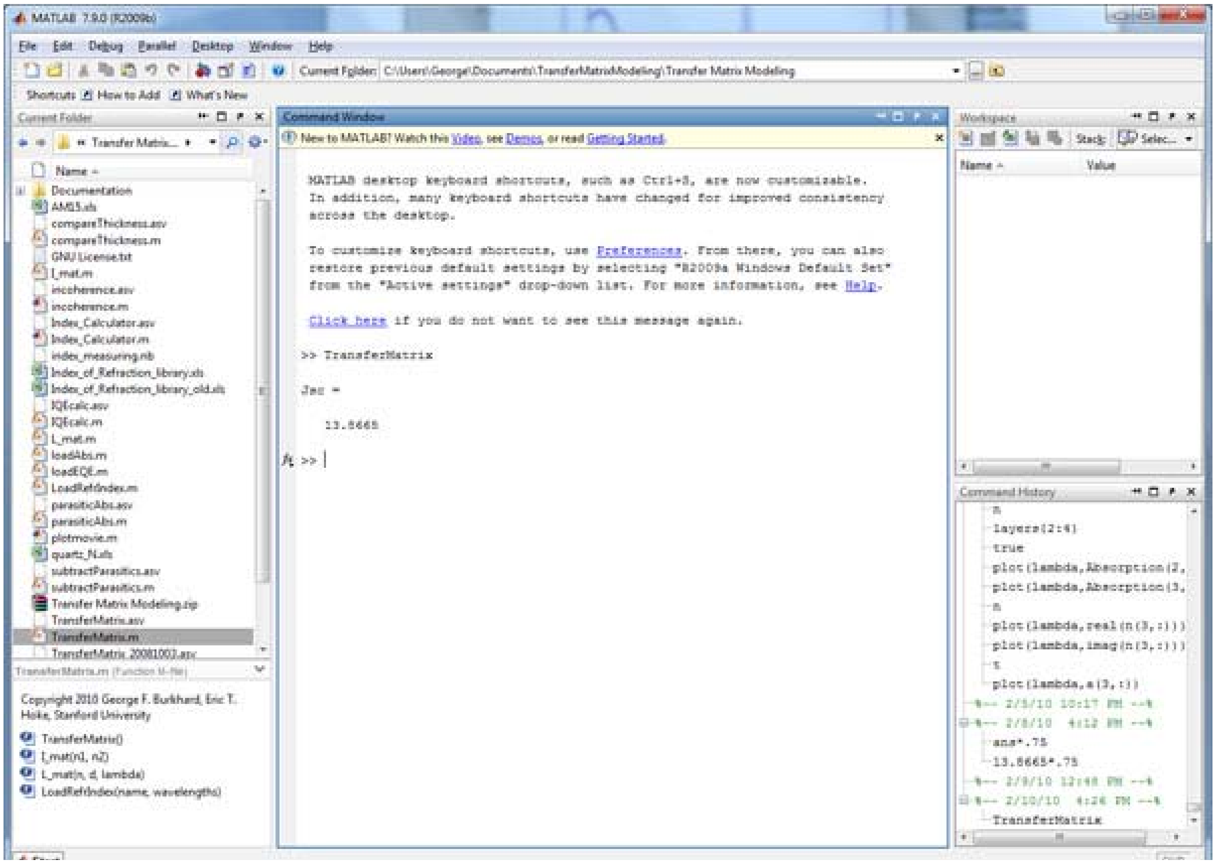
```
lambda=350:800;
stepsize = 1;
plotWavelengths = [400 500 600];
layers = {'SiO2' 'ITOsorizon' 'PEDOT' 'P3HTPCBMBlendDCB' 'Ca' 'Al'};
thicknesses = [0 110 35 220 7 200];
plotGeneration = true;
activeLayer = 4;
```

The script is then run either by pressing *F5* in the editor or by typing, *TransferMatrix* at the Matlab command prompt (the current directory must be the directory containing the *TransferMatrix.m* file and associated *.xls* files). After a few seconds of computation, the results are plotted in four figure windows and the predicted short-circuit current (J_{sc}) is output in the command window (this assumes 100% *internal quantum efficiency* at all wavelengths).









By default, *TransferMatrix.m* creates variables, *absorption*, *reflection*, and *lambda* in the base workspace when run. These arrays contain the absorption data for each material in the stack, the total reflection one would expect to measure from the device, and the list of wavelengths for which these values correspond. This should be sufficient for most users, however, if you wish to output other local variables to the workspace (eg. E-field), you can add more *assignin('base'*,variable_name,local_variable*)* commands to the end of the main function.

References

[1] L. A. A. Pettersson, L. S. Roman, O. Inganas, *J. Appl. Phys.* **1999**, *86*, 487.

[2] P. Peumans, A. Yakimov, S. Forrest, *J. of Appl. Phys.* **2003**, *93*, 3693.