Email 1 8/6/2018 fROM Szilard Gyalay

Hello Gordon,

To answer your question, I can't say I have code or have heard of code that applies specifically to asteroid spectra. However, I can't imagine the application to asteroid spectra would be too different from general applications of Shkuratov. I believe Poulet et al. 2002 actually examined some asteroids (Trojans I want to say?) with both a Hapke model and a Shkuratov model. Karly Pitman also uses Shkuratov often, although I forget which specific surfaces now that I think about it.

I'm assuming you've read the original Shkuratov et al. 1999 paper and have read through their derivation of the model. As long as you have read that, the theory should still hold up and be applicable to any planetary surface. I'll explain what exactly my code does, and then ask some more questions about your project and how you hope to apply the Shkuratov model and we can go from there.

So the Shkuratov code I have is actually not that well written (I wrote this about halfway through undergrad, so I wasn't the best at coding at the time) but is serviceable. I wrote it in IDL, and it takes in up to ten different sets of optical constants and associated wavelengths of different materials. Then, given the concentrations of each material, grain sizes, and the porosity, spits out the modeled albedo (and then I have a separate set of code that spits out the associated wavelengths for that albedo list, since again I was not the best at coding back then). As you probably know, Shkuratov is a 1D hemispheric/integral albedo, but can be converted to a bidirectional reflectance (Shkuratov and Grynko 2005), which is what I'm assuming what you're mostly getting from your asteroid observations.

I also have code written in Python that will derive the optical constants given a spectrum using Shkuratov et al. 1999, assuming the spectrum is of a surface of uniform composition.

So some questions:

I was doing mostly forward models, and never really got around to making a fitting program to derive composition given an observed spectrum. I'm not sure if that's what you're trying to do or not? So if you're good guessing compositions, my code will work for that, but won't derive or fit compositions without more code being written.

Just wanted to double check that you did indeed want to use Shkuratov. While I do like it for many reasons, I know many prefer Hapke, and a lot more people have code lying around for that (I want to say many people use Ted Roush's Hapke code). One big caveat of Shkuratov is that it is a 1D albedo, whereas most observations are bidirectional reflectance. There are ways to convert, but I do not think anyone has formalized a complete conversion (unless you want to use Hapke to develop a phase-function).

I don't know how far into IR your NIR observations go, but one thing you may have to worry about is how much of your NIR signal comes from reflectance, and how much comes from the emissivity of your asteroids. For lab samples I'm sure their emissivity doesn't factor in much if at all.

So let me know if my stuff sounds kind of what you're looking for, and we can talk from there.

Reply...From GG 8/8/2018

Thanks for your prompt reply and offer to help. I think your code, while not providing everything I need, can provide a solid foundation that I can build off to get the desired end result. I am excited to work with you to understand the code and how it functions. Your explanation of the basics was very clear.

Yes, I am familiar with the Shkuratov model and Poulet’s work. There are quite a few others who have worked on asteroids (including one D-type – 10199 Chariklo) using Shkuratov and I have been in contact with almost all of the investigators with little luck. I have more understanding of Shkuratov than of Hapke (although I have read some of Hapke’s book 😊). Having pondered this over the past months, my thoughts are Shkuratov gives a little more empirical flexibility than Hapke that may make it easier to model surfaces for featureless bodies such as D-types. Having said this, I am not closed off to Hapke should that be the only option where working code is available to do what I need.

On a macro level, my modeling objectives are to:

Input a D-type asteroid spectrum (0.69-2.5μm at this point) plus a set of proposed surface mineral components with appropriate parameters (abundances, grain sizes, porosities, and optical constants).

Use radiative transfer (RT) modeling equations, in this case Shkuratov, to create a model spectrum of the combined mineral components covering the same wavelength range as the asteroid spectrum.

Compare the asteroid spectrum to the modeled mineral components.

Deliver some measure of the goodness of fit.

Repeat until the best fit set(s) of components possible is found for each asteroid spectrum.

Yes, I am trying to guess the composition, but hopefully it will be a considered guess. I have lab access to physical samples of the Tagish Lake meteorite (TLM – proposed to be from a D-type parent body), many other meteorites (CM’s, CR’s etc), and a variety of terrestrial analog materials which could be plausible, in some combination, as components of a D-type surface. I also have access to a spectrograph. I have the RELAB spectra of multiple lithologies of TLM, of WIS91600 (another D-type analog meteorite), and agreement to get the spectra (unfortunately no samples) of MET00432, the other D-type analog meteorite (There are only three D-type analogs in the entire meteorite collection!). Use of an RT model gives me some level of accepted validation, to support my conclusions.

At first glance, your Python module would be very useful in developing optical constants (OC) for the mixtures that will need to be produced. OC’s for D-type analog stuff are rare, outdated (TLM), or non-existent. Your IDL code will provide guidance on handling the inputs and applying the equations. Yes, a conversion and a fitting routine each need to be produced. I am an awful coder, but I will take a shot (a good shot) at those with a little help from my friends 😊.

To answer, your last question, emissivity does not factor at all with my spectra. In my repository of past work done on D-types, I have TIR spectra, provided by Josh Emery, of a few D-types, but I am not planning on modeling those. As far as adding additional wavelength range, It is possible, I could be granted time to image D-types in UV/VIS at a telescope outside the U.S. but that proposal is still under consideration and emissivity is not a consideration there.

Hopefully, I have answered all your questions. Feel free please to shoot more at me. I would really like to obtain all the code you described plus any documentation, however cryptic, that you have. If you have a sample data set you have successfully run, plus results, that would also be very helpful as well as anything else you can think of.

If there is anything I can do to help you in your research, please let me know. Thanks again Szilard!

Reply from Szilard 8/8/2018

Okay, let me attach the functions. For my IDL code, I had the NASA Astro IDL library installed, but I don't think any of those functions in the library are actually used in this code. The NASA Astro Library and the mpfitfun library are both good IDL libraries to have installed if you don't use them already, especially if you want to write a fitting/goodness-of-fit routine.

Both genshkuratov and genshkuratovwaves take in the same inputs, but one outputs the modeled albedo, and the other outputs the corresponding wavelengths. For the wavelengths, I believe it finds the largest range that is a part of all input wavelengths, and interpolates optical constants to the highest resolution among them. (Funny thing, since this was early on when I was still learning how to code, I wrote up the interpolation myself without realizing there was an easy function I could use.) Not exactly the most efficient code I've ever written, but it should do its job.

The python code uses functions from NumPy and SciPy, so make sure to have those installed if you do not already. You should just be able to type import skkanalysis in a file or on the command line and then use skkanalysis.skkanalysis(\*args) to run the code. Again, this'll output the OC from an albedo spectrum. Glancing through, all these programs should be commented enough to explain what is going on, and I think I even cite which equations are being used from Shkuratov et al. 1999.

Let me know if you run into any trouble, and I guess keep me updated on how things go!