The endeavour undertaken for this independent study contract was written and directed by Dr. Farisa Morales and performed by Cody King for Physics M122.

The project undertaken herein consisted of two primary goals. First I was utilize previous knowledge and any available resources to integrate a functioning version of MIE Theory- which calculates the emissivity of particles as a function the optical constants associated with the particles composition, its radius, porosity, and the wavelength of incident light upon it- into the code I had written for the previous Independent Study; while initially this proved to be a challenge due to my limited understanding of MIE theory itself, it was decided that an in-depth understanding of MIE theory was not necessary to effectively implement it, and so a functioning module of MIE Theory was acquired from Dr. Bryden of NASA's Jet Propulsion Laboratory to suit this purpose. After testing, I made some very minor modifications to this code, removing portions which were unnecessary for the purpose it was to be used. After such modifications, numerous functions were written in the main body of EMT&MIE.py to utilize this implementation of MIE Theory on the optical constants generated by my implementation of Effective Medium Theory ('EMT')- the subject of the previous independent study. After modifying functions I had written for the reading and writing of data to disk, and of plotting values, the results of MIE Theory were then passed to these modified functions to be saved in an array-like fashion (with indices of grain radius and wavelength of incident light; both in microns) to Comma Separated Value ('CSV') tables and plotted, wherein the plots were saved as Portable Network Graphics pictures ('PNGs'). The validity of these results were verified by visual comparison between plots I had generated and those that Dr. Morales had generated from her program written in IDL, as well as through scrutiny by Dr. Morales herself. With the plots made and values saved, my next task was to ensure that these values could actually be utilized by Justin Bracks' portion of code, wherein he would take these emissivity values to calculate the equilibrium temperatures of various grains based on their emissivity, radial distance from their orbiting star, and the luminosity of said star. This was easy and done mostly by reversing the process through which the emissivity tables had been made, and providing this function for use by Justin Bracks. With all requested emissivities calculated and now available for usage by colleagues, my assigned goals have been accomplished for this independent study assignment; despite this, numerous improvements are to be made to the code I have written and maintain the intent to implementing these improvements, such as improving the ability to recursively impliment EMT through run-time keywords. Below is an example of the plots generated through *Matplotlib's* **Pyplot**. Emissivity lists are far too extensive to be shown here but are available on Github for viewing and/or download (see Appendix).

Generated Plot Samples-

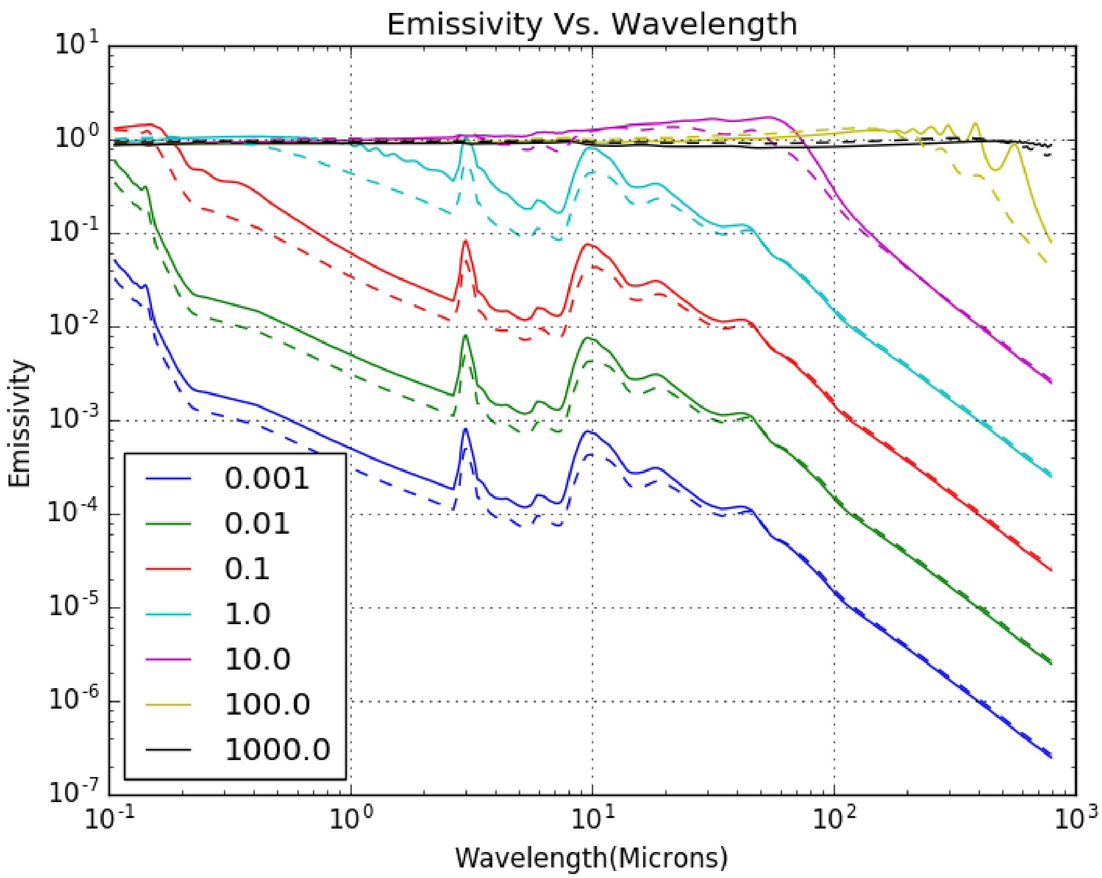

Figure 1: Plot of Emissivity of a 50-50 volume fraction ratio of a Dirty Ice matrix with AstroSil inclusions as a function of the wavelength of incident light and grain radius. Colored lines indicate grain radius in microns; a small sampling shown here. Dashed lines are the result of re-implementing EMT on the result of the aforementioned mixture, using it as the matrix and vacuum as the inclusion in another 50-50 volume fraction ratio.

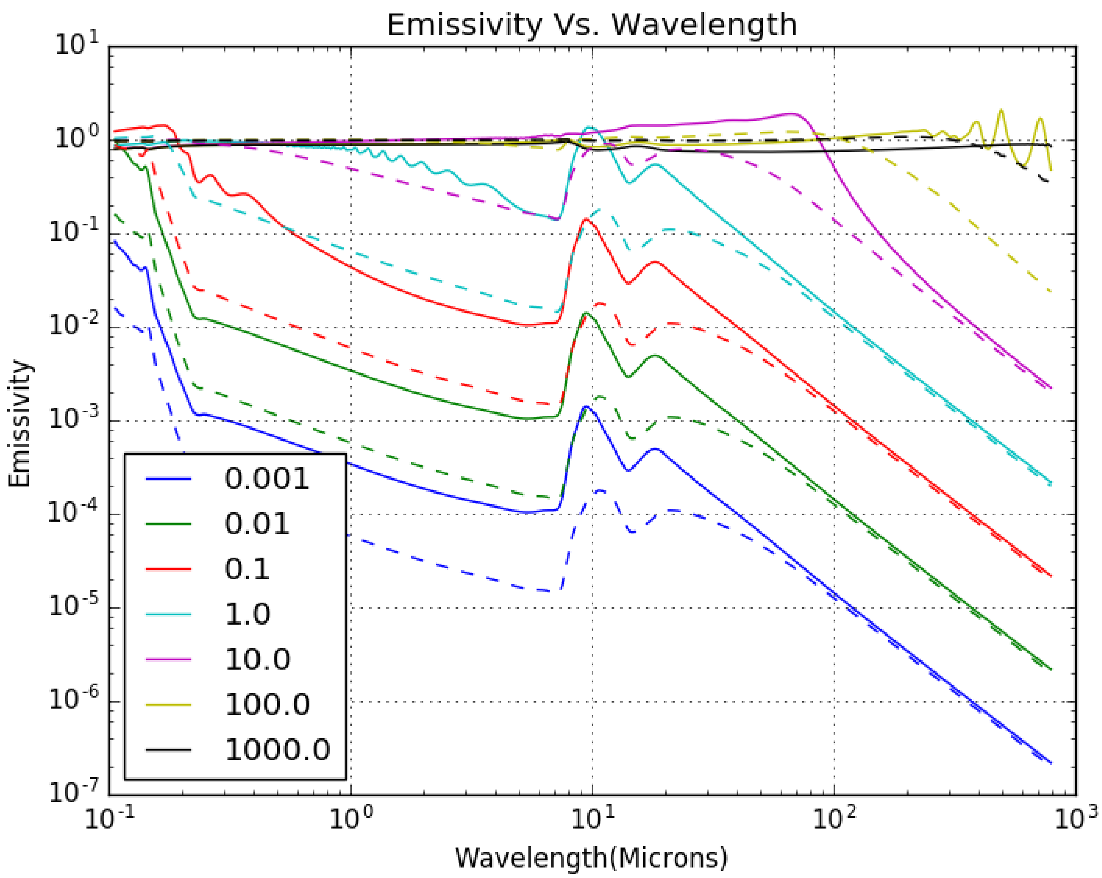

Figure 2: Plot of Emissivities of pure AstroSil particles as a function of the wavelength of incident light and grain radius. Colored lines indicate grain radius in microns; a small sampling shown here. Dashed lines are the result of incorporating high vacuous porosity (10% AstroSil, 90% Vacuum) through EMT. --

Code Sample-

```
from kmod import listifier, whiteSpaceParser
def readEmiss(FILE):
	FILE.seek(0)
	Lamda = listifier(whiteSpaceParser(FILE.readline()))
	Emiss = []
	for ele in FILE:
		temp = listifier(whiteSpaceParser(ele))
		Emiss.append(temp)
	return Lamda, Emiss
```

Figure 2: The short and simple function used to read in the emissivity tables

---

Appendix

All code and data tables are available online on github in the repository C3PO-for-H2D2 by user farisamorles (github.com/farisamorales/C3PO-for-H2D2/EMT-MIE); files discussed or shown here under the names:

- EMT&MIE.py (The main code body of my portion of this project)
- Emissivities[DI;AS;50-50].csv (The emissivity table from which samples had been plotted in Figure 1; the solid lines)
- Emissivities[DI;AS;50-50;Porous;50-50].csv (The emissivity table from which samples had been plotted in Figure 1; the dashed lines)
- EmVsLam[DI;AS;50-50]PorousOverlay.png (Figure 1)
- Emissivities[AS].csv (The emissivity table from which samples had been plotted in Figure 2; the solid lines)
- Emissivities[AS;Porous;10-90].csv (The emissivity table from which samples had been plotted in Figure 2; the dashed lines)
- EmVsLam[AS]PorousOverlay(10-90).png (Figure 2)