Wavelength Dependence of the Scattering of Small Particles by Sunlight

By

Scott A. Bain

A PROJECT

submitted to

Oregon State University

University Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Bachelors of Science in Physics (Honors Scholar)

Presented May 30, 2002
Commencement June 2002

AN ABSTRACT OF THE THESIS OF

Scott A. Bain for the degree of Honors Baccalaureate of Science in Physics presented on May 30, 2002. Title: Wavelength Dependence of the Scattering of Sunlight by Small Particles.

Abstract approved:

David Griffiths

This project calculates the value of β for ice particles based on wavelength dependent values of their refractive and absorptive indices. β is the ratio of the force due to radiation to the force due to gravity acting on a particle in the solar system. This parameter can later be used to determine the angle by which the trajectory of a particle changes upon entering the solar system. The radiation pressure is found using equations from the scattering theories of Mie and Debye. A computer program was used to create a table of values of the efficiency factor for radiation pressure, Q_{pr}. Each value of Q_{pr} in the table corresponds to value of λ. Each value of Q_{pr} was multiplied by the value of the Planck function at that wavelength. The integral of this product function, which is proportional to β, was found using a list integration function in Mathematica. Using wavelength dependent values of the absorptive and refractive indices gives a Q_{pr} function significantly different than those using constant values. This leads to the calculation of a realistic β value.

Wavelength Dependence of the Scattering
of Small Particles by Sunlight

By

Scott A. Bain

A PROJECT

submitted to

Oregon State University

University Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Bachelors of Science in Physics (Honors Scholar)

Presented May 30, 2002
Commencement June 2002

Honors Bachelor of Science in Physics project of Scott A. Bain presented on May 30, 2002.

APPROVED:

David Griffiths, representing Physics

Phil Siemens, representing Physics

Kenneth Krane, representing Physics

Henri Jansen, Department of Physics

Joe Hendricks, Dean of the University Honors College

Scott A. Bain, Author

ACKNOWLEDGEMENTS

I would like to acknowledge and thank all of those who helped me with this project. Dr. Griffiths was invaluable as my mentor, helping me whenever I needed it and explaining things to me up to the last minute. Dr. Siemens was a source of encouragement and helped point me to this project. I would also like to thank all of my friends and classmates for their patience with me as I rewrote and reran many programs and for having me explain this project to them, greatly increasing my understanding and my ability to communicate my work and results. Finally, I thank my parents. Without their help and encouragement, I would not have been admitted to the Honors College and would not have written this thesis as it now stands.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDIX FIGURES

LIST OF APPENDIX TABLES

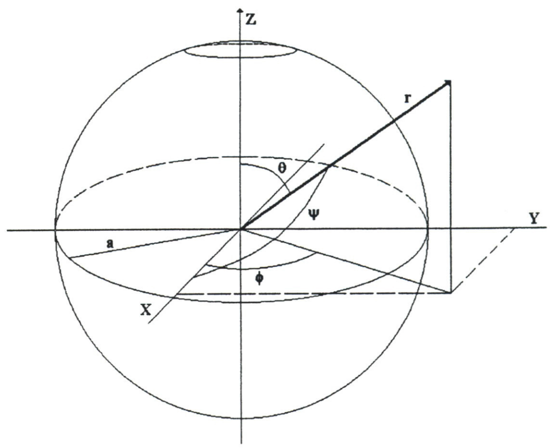
Wavelength Dependence of the Scattering of Sunlight by Small Particles

Background

Rayleigh Scattering[1]

Why is the sky blue? Most physicists know the blue color in the sky comes from the scattering of light off of the atoms and molecules in our atmosphere. Scattering of light is at the center of this thesis. Rayleigh scattering is the theory that describes the above phenomenon, our blue sky. This theory applies to cases where the radius of the particle, a, is much less than the wavelength of the incident light. Light acts just like a massive particle and scatters off of true massive particles, almost like billiard balls. The intensity, a unitless quantity, of radiation scattered at an angle ψ and distance r is given by

Figure 1
Scattering Diagram



$$I(r,\psi) = \frac{16\pi^4 a^6}{r^2\lambda^4}\left|\frac{m^2-1}{m^2+2}\right|\sin^2\psi \qquad (1)$$

where m is the index of refraction for the particle. By integrating the intensity function over the scattering angle ψ and the azimuthal angle ϕ, we can calculate the scattering cross-section, C_{sca}. The scattering cross-section has units of area.

$$C_{sca} = \int_0^{\pi}\int_0^{2\pi} Ir^2 \sin\psi d\psi d\phi \tag{2}$$

$$C_{sca} = \frac{128\pi^5 a^6}{3\lambda^4}\left(\frac{m^2-1}{m^2+2}\right)^2 \tag{3}$$

The efficiency factor of the scattering, which is unitless, can be calculated from the cross-section. The equation for this is $C_{sca}/\pi a^2$, which is the scattering cross-section over the cross-sectional area of the particle with radius a. The efficiency factor for scattering, Q_{sca}, is

$$Q_{sca} = \frac{128\pi^4 a^4}{3\lambda^4}\left(\frac{m^2-1}{m^2+2}\right)^2 \tag{4}$$

which can be rewritten as

$$Q_{sca} = \frac{8}{3}x^4\left(\frac{m^2-1}{m^2+2}\right)^2 \tag{5}$$

using the size parameter $x = 2\pi a/\lambda$. While this is a good overview of light scattering, Rayleigh scattering is a very special case, applying only when $a \ll \lambda$, as mentioned above. Mie scattering theory is the most general solution for light scattering off of a sphere.

Mie Scattering and Radiation Pressure

Light has momentum. The momentum of a photon is given by the equation

$$p = \frac{E}{c} \tag{6}$$

where E is the energy and c is the velocity of the photon. When a photon scatters off of a particle, it is absorbed or deflected. Either of these interactions causes a change in momentum of the photon, which becomes a force exerted upon the particle. This force is

$$F = \frac{(C_{ext} - \langle \cos\theta \rangle C_{sca})}{c} S \tag{7}$$

S in this equation is the magnitude of the Poynting vector; C_{ext} is the cross-section for extinction. C_{ext} is the sum of the cross-sections for scattering and absorption, leading to the following two equations:

$$C_{ext} = C_{sca} + C_{abs} \tag{8}$$

$$Q_{ext} = Q_{sca} + Q_{abs} \tag{9}$$

<cosθ> is the average of cosθ weighted by the angular intensity of the scattered beam.[1] It is given by

$$\langle \cos\theta \rangle = \frac{\int_0^\pi S(\theta) \cos\theta \sin\theta d\theta}{\int_0^\pi S(\theta) \sin\theta d\theta} \tag{10}$$

where S(θ) is the fraction of energy incident on the particle that is scattered into a cone opening forward with sides at angle θ to the forward z axis.[2]

The pressure on the particle is[1]

$$P = \frac{F}{\pi a^2} = \frac{S}{c} Q_{pr} \tag{11}$$

where

$$Q_{pr} = Q_{ext} - \langle \cos\theta \rangle Q_{sca} \tag{12}$$

The function $\langle\cos\theta\rangle Q_{sca}$ can be expressed in terms of spherical Bessel functions, eliminating the dependence on S(θ), as follows.

$$\langle \cos\theta \rangle Q_{sca} = \frac{4}{x^2}\sum_{n=1}^{\infty}\left\{\frac{n(n+2)}{n+1}[\mathrm{Re}(a_n)\mathrm{Re}(a_{n+1}) + \mathrm{Im}(a_n)\mathrm{Im}(a_{n+1}) + \mathrm{Re}(b_n)\mathrm{Re}(b_{n+1}) + \mathrm{Im}(b_n)\mathrm{Im}(b_{n+1})] + \frac{2n+1}{n(n+1)}[\mathrm{Re}(a_n)\mathrm{Re}(b_n) + \mathrm{Im}(a_n)\mathrm{Im}(b_n)]\right\} \tag{13}$$

Both Debye and Mie derived this result independently in the early 1900's. The above function uses the following factors and functions.

$$a_n = \frac{x\psi'_n(y)\psi_n(x) - y\psi'_n(x)\psi_n(y)}{x\psi'_n(y)\zeta_n(x) - y\zeta'_n(x)\psi_n(y)} \tag{14}$$

$$b_n = \frac{y\psi'_n(y)\psi_n(x) - x\psi'_n(x)\psi_n(y)}{y\psi'_n(y)\zeta_n(x) - x\zeta'_n(x)\psi_n(y)} \tag{15}$$

$$\psi_n(\alpha) = \left(\frac{\pi\alpha}{2}\right)^{\frac{1}{2}} J_{n+\frac{1}{2}}(\alpha) \tag{16}$$

$$\chi_n(\alpha) = (-1)^n \left(\frac{\pi\alpha}{2}\right)^{\frac{1}{2}} J_{-n-\frac{1}{2}}(\alpha) \tag{17}$$

$$\zeta_n(\alpha) = \left(\frac{\pi\alpha}{2}\right)^{\frac{1}{2}} \left[J_{n+\frac{1}{2}}(\alpha) + i(-1)^n J_{-n-\frac{1}{2}}(\alpha)\right] \tag{18}$$

Here, $J_k(\alpha)$ is the spherical Bessel function of half-integral (k) order. These account for the radial behavior of the light scattered from the particle. Equations 16-18 are the Riccati-Bessel functions. While $\chi_n(\alpha)$ is not used explicitly, it is noted that

$$\zeta_n(\alpha) = \psi_n(\alpha) + i\chi_n(\alpha) \tag{19}$$

The variable y = mx = (n+ik)x when k is small compared to n. n and k are the refractive and absorptive indexes, respectively, of the material of which the particle is composed.

Generally, m is $[K-2i\sigma\lambda/c]^{1/2}$, where K is the dielectric constant of the material and σ is its electrical conductivity and its value is small.

Written in this general format, Q_{sca} is

$$Q_{sca} = \frac{2}{x^2}\sum_{n=1}^{\infty}(2n+1)\left[|a_n|^2 + |b_n|^2\right] \tag{20}$$

We finally write Q_{ext} in terms of the factors a_n and b_n as follows.[2]

$$Q_{ext} = \frac{2}{x^2}\sum_{n=1}^{\infty}(2n+1)\operatorname{Re}(a_n + b_n) \tag{21}$$

This allows us to rewrite equation 12 as

$$\begin{aligned} Q_{pr} = {} & \frac{2}{x^2}\sum_{n=1}^{\infty}(2n+1)\left[|a_n|^2 + |b_n|^2\right] - \frac{4}{x^2}\sum_{n=1}^{\infty}\left\{\frac{n(n+2)}{n+1}\right. \\ & [\operatorname{Re}(a_n)\operatorname{Re}(a_{n+1}) + \operatorname{Im}(a_n)\operatorname{Im}(a_{n+1}) + \\ & \operatorname{Re}(b_n)\operatorname{Re}(b_{n+1}) + \operatorname{Im}(b_n)\operatorname{Im}(b_{n+1})] + \\ & \left.\frac{2n+1}{n(n+1)}[\operatorname{Re}(a_n)\operatorname{Re}(b_n) + \operatorname{Im}(a_n)\operatorname{Im}(b_n)]\right\} \end{aligned} \tag{22}$$

and reduce it to the equation used in the computer program,

$$\begin{aligned} Q_{pr} = {} & \frac{2}{x^2}\sum_{n=1}^{\infty}\left\{(2n+1)\left[|a_n|^2 + |b_n|^2\right] - 2\frac{n(n+2)}{n+1}\right. \\ & [\operatorname{Re}(a_n)\operatorname{Re}(a_{n+1}) + \operatorname{Im}(a_n)\operatorname{Im}(a_{n+1}) + \\ & \operatorname{Re}(b_n)\operatorname{Re}(b_{n+1}) + \operatorname{Im}(b_n)\operatorname{Im}(b_{n+1})] - \\ & \left.2\frac{2n+1}{n(n+1)}[\operatorname{Re}(a_n)\operatorname{Re}(b_n) + \operatorname{Im}(a_n)\operatorname{Im}(b_n)]\right\} \end{aligned} \tag{23}$$

Beta

The goal of this thesis is to find the parameter β, the ratio between force due to the radiation pressure of the sun and gravitational force exerted on the particle by the sun.

As stated in equation 7, the force exerted by radiation is C_{pr} S/c. This force can be rewritten in terms of λ as

$$F_{rad} = \int \frac{2\pi hc^2}{c\lambda^5}\left(\frac{d\lambda}{\exp[hc/\lambda kT_s]-1}\right)\pi a^2 Q_{pr}\frac{R_s^2}{r^2} \tag{24}$$

where a is the radius of the particle and r is the distance from the sun. This also includes the Planck blackbody radiation function

$$B(\lambda, T_s) = \frac{1}{\lambda^5(\exp[hc/\lambda kT_s]-1)} \tag{25}$$

The gravitational force is

$$F_{grav} = \frac{GM_s m}{r^2} \tag{26}$$

M_s and R_s are the mass and radius of the sun, respectively. Knowing these two forces, we can write β as

$$\beta = \frac{F_{rad}}{F_{rav}} = \frac{2\pi hc^2}{cGM_s}\frac{R_s^2}{r^2}\frac{\pi a^2}{\rho}\frac{3r^2}{4\pi a^3}\int\frac{Q_{pr}d\lambda}{\lambda^5(\exp[hc/\lambda kT_s]-1)} \tag{27}$$

ρ is the mass density of the particle. More concisely, β is given as follows

$$\beta = \frac{3\pi hcR_s^2}{2GM_s\rho a}\int_{\lambda_l}^{\lambda_u}\frac{Q_{pr}[m^*(\lambda),(a/\lambda)]}{\lambda^5\exp[hc/\lambda kT_s]-1}d\lambda \tag{28}$$

β can be rewritten in terms of the size parameter x as:

$$\beta = \frac{3hcR_s^2}{32\pi^3 GM_s a^5\rho}\int_{x_l}^{x_u} Q_{pr}[m^*(x),x]\frac{x^3}{\exp[(\hbar c/kT_s a)x]-1}dx \tag{29}$$

Note that the value of the x^3 dx term is negative. We are interested in the magnitude of β, which can be accounted for by switching the limits of the integral. Since x is inversely proportional to λ, the limits on the integral go from low x to high x. Using the values

Table 1:
Values of Important Constants

h	$6.626\ 10^{-27}$ g cm^2 / s
c	$3.00\ 10^{10}$ cm / s
R_s	$6.96\ 10^{10}$ cm
G	$6.67\ 10^{-5}$ cm^3 / g s^2
M_s	$1.99\ 10^{33}$ g
k	$1.38\ 10^{-16}$ g cm^2 / s^2 K
T_s	5800 K

given in Table 1 and using a=0.25 micrometers, we can rewrite equation 29 as a more convenient expression.

$$\beta = \frac{2.2470}{\rho} \int_{x_l}^{x_u} Q_{pr}[m^*(x), x] \left(\frac{x^3}{\exp[1.582x] - 1} \right) dx \tag{30}$$

For this equation, λ is in micrometers and the density is in g/cm^3.

Project

Q_{pr}

The first object of this project was to write a program to find values of Q_{pr} and β. Before the program began, certain parameters had to be set. A range for x was chosen from 0.1 to 10.1, roughly fitting a range of λ values from 0.15 to 15 microns. This was later modified to span x=0.3 to x=10.1 because the λ value of ~15 was so far removed from this new range, which spans wavelengths from about 0.15 microns to about 5 microns. This smaller range still covers a sufficiently large percentage of the Planck curve of the sun, which will be discussed later.

Another parameter that needed to be decided upon was the number of orders of the Bessel functions to be summed over. Using a Q_{pr} program written by Dr. Griffiths[(3)], sums over up to 15 orders were tested. While there were significant differences between summing over ten orders and summing over fewer orders, there seemed to be no difference in the values of Q_{pr} for sums to orders 10, 12, and 15. However, summing over fifteen orders took quite a long time to run for fifty values of Q_{pr} (about half an hour). Out of the three options, summing over ten orders took the least time, fifteen to twenty minutes, and was as accurate as the other two choices.

This initial program was written to use values of n and k that are constant over all wavelengths. The first attempts at this had the functions a_n and b_n explicitly written out in terms of the Riccati-Bessel functions. This did not work in either Maple or Mathematica. Instead of spending hours debugging this code and figuring out why it wouldn't work, it was decided to use the program written by Dr. Griffiths, mentioned above, as a basis for a new program. The algorithm for this program defined each part of the quotients in a_n and b_n separately, named their quotient a_n and b_n, and used equation 23 to create tables of Q_{pr} values. The value used for ρ in this and subsequent programs was that of water ice, 0.92 g/cm^3.

This program created five tables of values, each term of which was the value of the function for a certain x value. The values of x for all of the tables ranged from x=0.3 to x=10.1 in increasing steps of 0.2. The first table listed the values of Q_{pr} calculated for each value of the x range. The second listed the appropriate values of the blackbody function. The third table listed the values of the integrand for β, the product of the first two tables. After this table, a value of β was calculated. The last two tables listed coordinate pairs of x and the corresponding value of Q_{pr} and the blackbody function, respectively. Finally, graphs of these tables were plotted. With the test program completed and working, the next step was to incorporate wavelength dependent values of the absorptive and refractive indices.

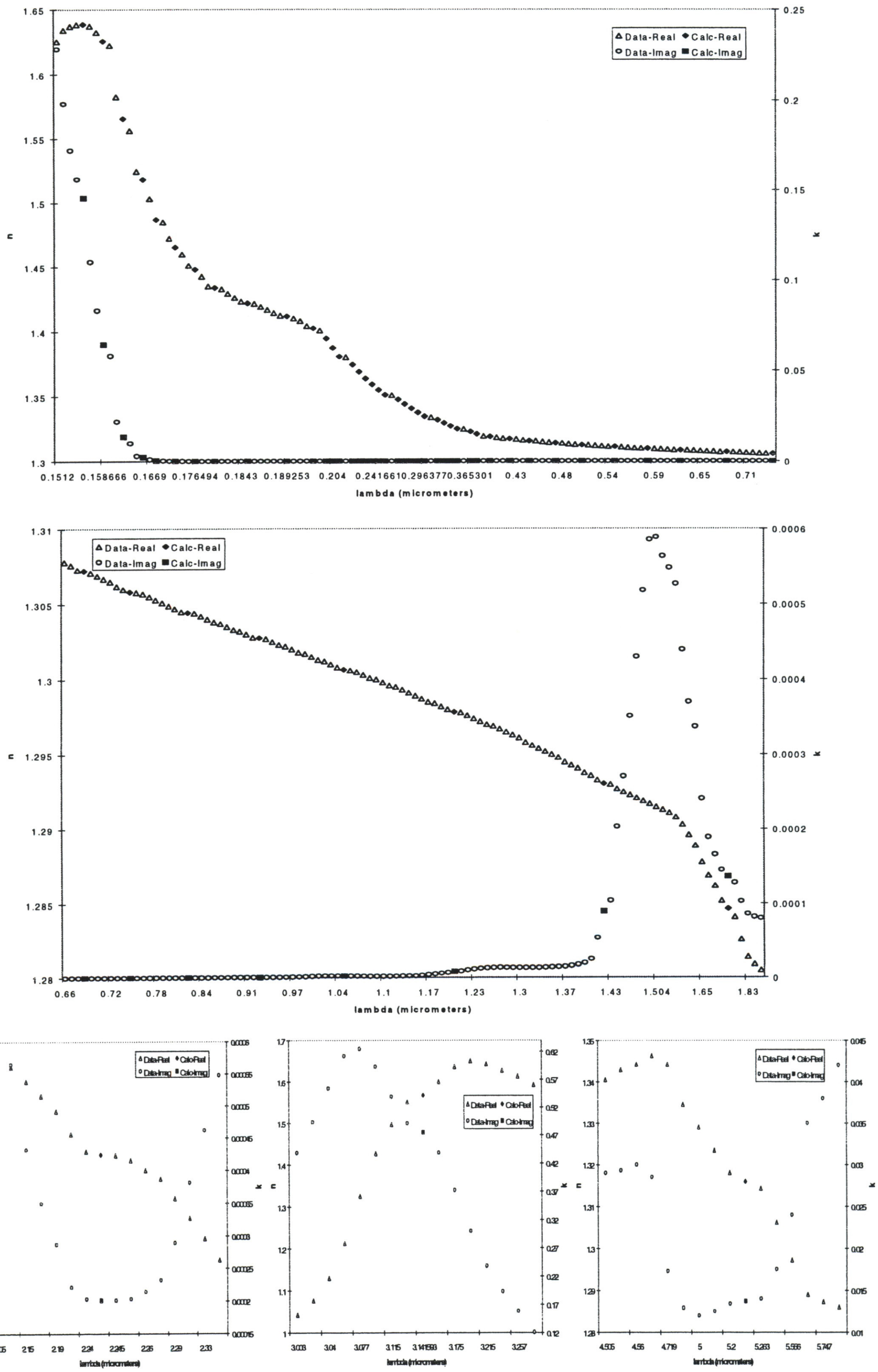
n(λ)+ik(λ)

The ultimate goal of this project is to use realistic, wavelength dependent values for the refractive and absorptive indices, n(λ) and k(λ), of a particle in space in order to determine how it is deflected from its original path when it enters the solar system. These data were found in an article published by Dr. Stephen Warren from the University of Washington.[(4)] The article includes a table containing values of λ, n, and k for ice over a wide range of wavelengths, including the infrared, visible, and ultraviolet portions of the electromagnetic spectrum. Unfortunately, none of the specific values of λ that were given match the values of λ used in our program. The most important implication of this fact is that we then needed to somehow perform a curve fit to the data in order to interpolate values for the λ values we were using. The second problem was that there was no way to perform a direct comparison between the values from the actual data and the curve fit. Without this direct comparison, error on the calculated data points can only be extrapolated.

A curve fit was needed. After looking for curve fits in Mathematica and Maple, it was discovered that Maple could perform a spline fit to data read from a pair of files. The spline fit command in Maple creates a third-degree polynomial function that best approximates the function in the space between any two subsequent λ values given. Each part of the piecewise function is labeled by the term $\lambda < \lambda_j$ where λ_j is the j^{th} value of lambda in the read data file. Because nearly every required value of λ had a different function describing the corresponding values for both $n(\lambda)$ and $k(\lambda)$, a "brute force" approach was needed to find all 100 values within our range.

Two Maple worksheets were created for this brute force method, one to find $n(\lambda)$ values and one to find values of $k(\lambda)$. In each worksheet, the required values of λ were calculated. For each value, the corresponding function was copied and pasted into the worksheet, λ was defined, and the value of $n(\lambda)$ or $k(\lambda)$ was calculated by Maple. Plots comparing the calculated value of $n(\lambda)$ and $k(\lambda)$ to the data are shown in Figure 2. After all fifty values of $n(\lambda)$ or $k(\lambda)$ were found, they were copied and pasted and formatted into a column. Each value was then assigned a name of n1-n50 or k1-k50, corresponding to the appropriate value of x, increasing from 0.3. This formatting was

Figure 2
Calculated and given data values of n(λ) and k(λ)



such that Mathematica could read it without any alteration. Both of these columns were then copied and pasted into a second Mathematica program that solves for β.

Beta Using Wavelength Dependent m Values

A second Mathematica program was written to solve the wavelength dependent values of Q_{pr}. The values of Q_{pr} were solved for using the same method as in the previous program. However, this program solved for fifty different values of Q_{pr}. Each value was defined as q1-q50, each using the corresponding value of n1-n50 and k1-k50. The output for each of these Q_{pr} values was a table one item long, the Q_{pr} value for the appropriate value of x. The end of the program created a table of q1-q50. Unfortunately, this created a "table of tables", as each item in this final table was a table itself.
This final table was copied and pasted into a third Mathematica program and edited so that each item was no longer its own table. This was then multiplied by a table of Planck function values, as in the first program, to give a table for the integrand. The integral was then performed, using the ListIntegrate function of Mathematica, and multiplied by the appropriate constants to give an accurate, wavelength dependent value of λ. Since this was a modification of the first program, the plot of the Planck function was left as a part of this program. However, the plot of Q_{pr} was removed due to the lack of ease of producing a table to create a plot.

Results and Conclusions

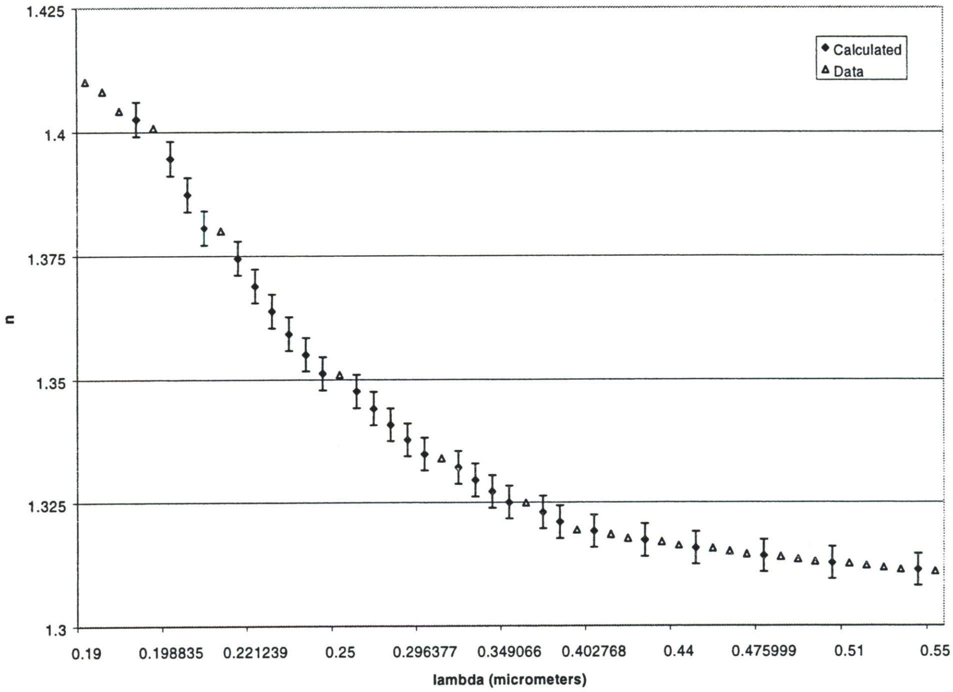
Planck Function

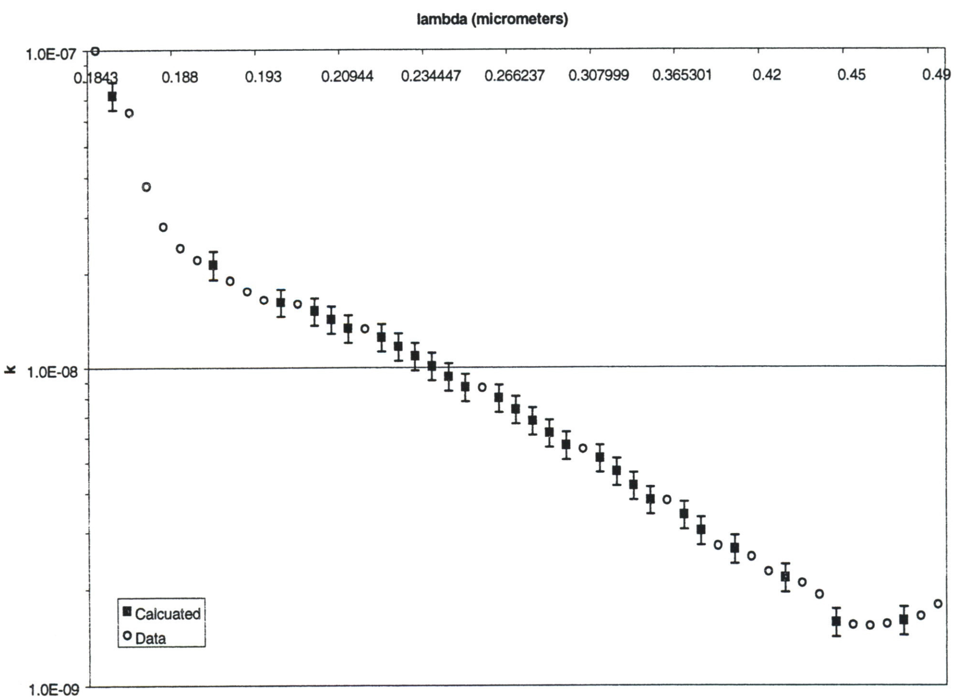
This project looks at the wavelength dependence of radiation pressure on a particle in our solar system. The wavelengths that we use should be those at which the sun radiates with significant intensity. The Planck blackbody radiation curve gives us, for an object at a certain temperature, intensity as a function of wavelength. The blackbody equation used in the program is

$$B(x, T_s) = \frac{x^3}{\exp[1.582x] - 1} \tag{31}$$

This is written in terms of the dimensionless size parameter x for a particle of radius 0.25 microns. The question arises as to how much of the sun's total blackbody curve does our program integrate over. In the program, x ranges from 0.3 to 10.1. The blackbody radiation curve covers all wavelengths from zero to infinity. This corresponds to the same range for x. Since the Planck curve is small for very small x, we used $x=10^{-4}$ as the lower limit in our total integration. Infinity, likewise, can be approximated by a value of x large compared to 10. Two values were used in two different comparisons, 10^3 and 10^4. The comparisons used the ListIntegrate program to integrate over a table of values produced using the above limits. The integral of these two curves was the same to six figures, returning a value of 1.03674. The integral from x=0.3 to x=10.1 gave a value of 1.03192, differing from the total value by 0.00482. This gives a percent difference of about 0.46%, meaning that our program uses 99.54% of the total solar Planck curve. This leaves little room for error to arise from this portion of the project.

Figure 3
Graphs with error bars on spline fit of n(λ) and k(λ)
Error on n(λ) is 0.25%. Error on k(λ) is 10%.





Error Analysis of Spline Fit

While the approximations from the spline fit are good, there is still some definite variation from the data. This is shown by the graphs in Figure 3. Most of the values for the refractive index seem to be good to within 0.25 %. Most of the k values, on the other hand, seem to be good to within 10%. This is an uncomfortably large value. However, an error of 10% here means a difference of less than 10^{-5} for over 80% of our calculated values. A difference this small in the absorptive index would not affect our results for Q_{pr} in any significant manner. The point with the largest value, x=0.5, is good to within 5%, an acceptable value.

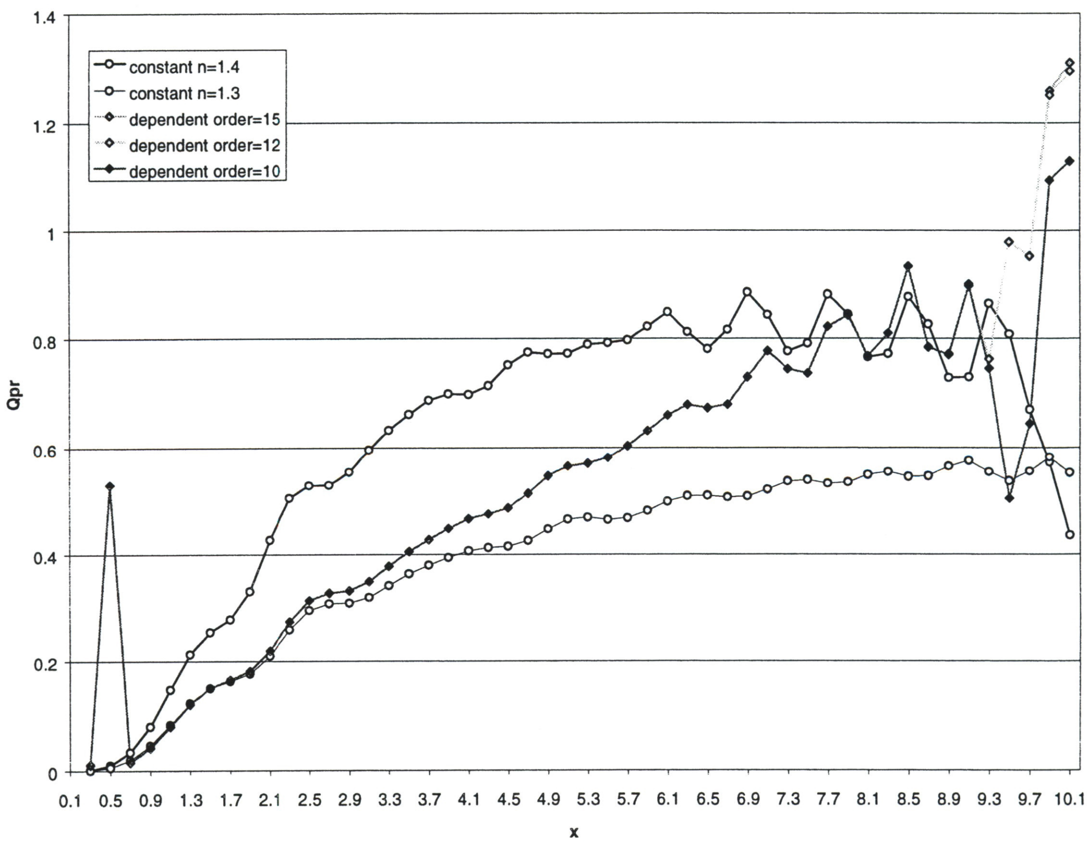
Comparison Between Orders of Bessel Functions

As shown in Figure 4, there are differences in the Q_{pr} values when using more than ten orders of the spherical Bessel functions. However, these differences become significant only above x=9. Past this point, the Planck curve is small, as is the product of the Planck curve and Q_{pr}. The β values differ by less than 0.00011 between n=10 and n=12, and by about 0.000001 between n=12 and n=15. (See Table 2) For n=10, the β value differs from that of n=12 by 0.016%. This percent difference is essentially the same between orders of 10 and 15. Using the sum of Bessel functions to the twelfth order is more accurate than summing to the tenth order. Anything above twelve orders, however, seems too insignificant to spend the processing time on.

Table 2
Comparison of results for β

n value	k value	Order	β value	Difference	% Diff.
λ dependent	λ dependent	10	0.69104	0	0
λ dependent	λ dependent	12	0.69115	0.00011	0.0159
λ dependent	λ dependent	15	0.69115	0.00011	0.0159
1.4	0.15	10	2.6194	1.9284	279
1.4	0	10	1.0785	.3874	56.07
1.3	0	10	0.59908	0.09196	13.31
1.3	0	15	0.59908	0.09196	13.31

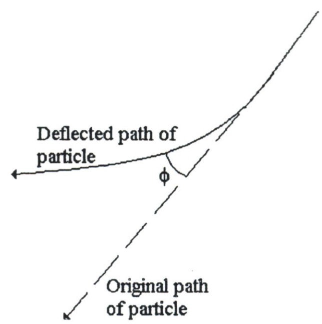
Figure 4
Q_{pr} using different values of n and k=0



Comparison Between Constant and Dependent m Values

The original program, written by Dr. Griffiths, used a value of 1.4 for the refractive index and 0.15 for the absorptive index. The values from this program for both β and Q_{pr} differ largely from the wavelength-dependent values, as shown in Table 2 and Figure 4. For a more accurate comparison, we first decided to keep n=1.4, but to set k to zero, as most of the values for k are very smaller than 10^{-5}. This gave a better β than the previous refractive index values, but it was still too large. The β value is over 1.5 times the value from the λ dependence.

Figure 5
Path Deflection by ϕ



Furthermore, the value of β is greater than one. This is significant, as it will change the quadrant of the deflection angle. This angle is related to β as follows.

$$\phi \propto \tan^{-1}\left(\frac{1-\beta}{\kappa}\right) \tag{32}$$

κ is a product of constants that is not significant here. As can be seen, a β greater than one will change the sign of the arctan function, which will make it appear as if the particle deflected in the other direction. The major difference between this and the previous program is that this does match the general form of the wavelength dependent Q_{pr} values, including a departure from the clean line appearance of the first portion of the function.

Another set of values were run, this time with n=1.3 and k=0. Both of these values correspond to the approximate value for most of the wavelengths. The Q_{pr} values here match the wavelength dependent function better than the n=1.4 curve. In fact, the two functions match very well for $x<1.5$. The biggest discrepancy here is the commonality between this and the $n=1.4$ program: This curve maintains its nice appearance throughout the x range, not "scattering" at high x. While the β value produced with this constant n value is better than the β from $n=1.4$, it is still significantly different from β found using the wavelength dependence. Furthermore, the percent difference of 13.31 is much larger than the percent error from the approximated data points.

It is interesting to note that the local maxima and minima for all of the Q_{pr} plots with small k seem to be consistent but offset in the different functions. It also appears that the "scattering" of Q_{pr} values occurs after the same number of local maxima, excluding the point $x=0.5$. This scattering appears to happen in this data range on for n values greater than 1.3.

Conclusions

There is no easy and accurate substitute for the wavelength dependent n+ik values when calculating Q_{pr}. While some constant approximations come closer than others, our closely matching values were chosen using the function of n+ik vs. wavelength. A closely matching constant approximation of any wavelength dependent n+ik value would need to be based on the data. We have looked only at this type of data for ice. Other materials may behave differently, with an easier constant approximation, but we need to find data for their refractive and absorptive indices as a function of wavelength to know this. If this is done after setting up a computer program, there is the advantage of finding the data for the specific values of λ that are needed to run the program and solve for β. This would avoid the approximations that were necessary in this project.

Investigation into the β, $n(\lambda)$, and $k(\lambda)$ curves of other materials is one possibility for future work. Carbon dioxide, methane, and other types of space dust have different properties than water, and will produce a different β value and Q_{pr} curve. The behavior of a cloud of particles differing in radius could also be investigated. Further investigation with water ice could be done performing a study on the exact Q_{pr} curves for λ less than the particle radius and for the regions around resonance wavelengths, with produce relatively high $n(\lambda)$ and $k(\lambda)$ values.

Bibliography

1. Kerker, Milton. 1969. *The Scattering of Light and Other Electromagnetic Radiation.* New York: Academic Press, Inc.

2. Wickramasinghe, N.C. 1973. *Light Scattering Functions For Small Particles with Applications in Astronomy.* New York: Halsted Press.

3. Griffiths, David J. 2001. *beta2.nb.* Mathematica program.

4. Warren, Stephen G. Optical Constants of Ice from the Ultraviolet to the Microwave. *Applied Optics* Vol.23 No.8: 1206-25.

APPENDICES

APPENDIX A

Figure 1
Comparison of Calculated and Given n(λ) and k(λ) values

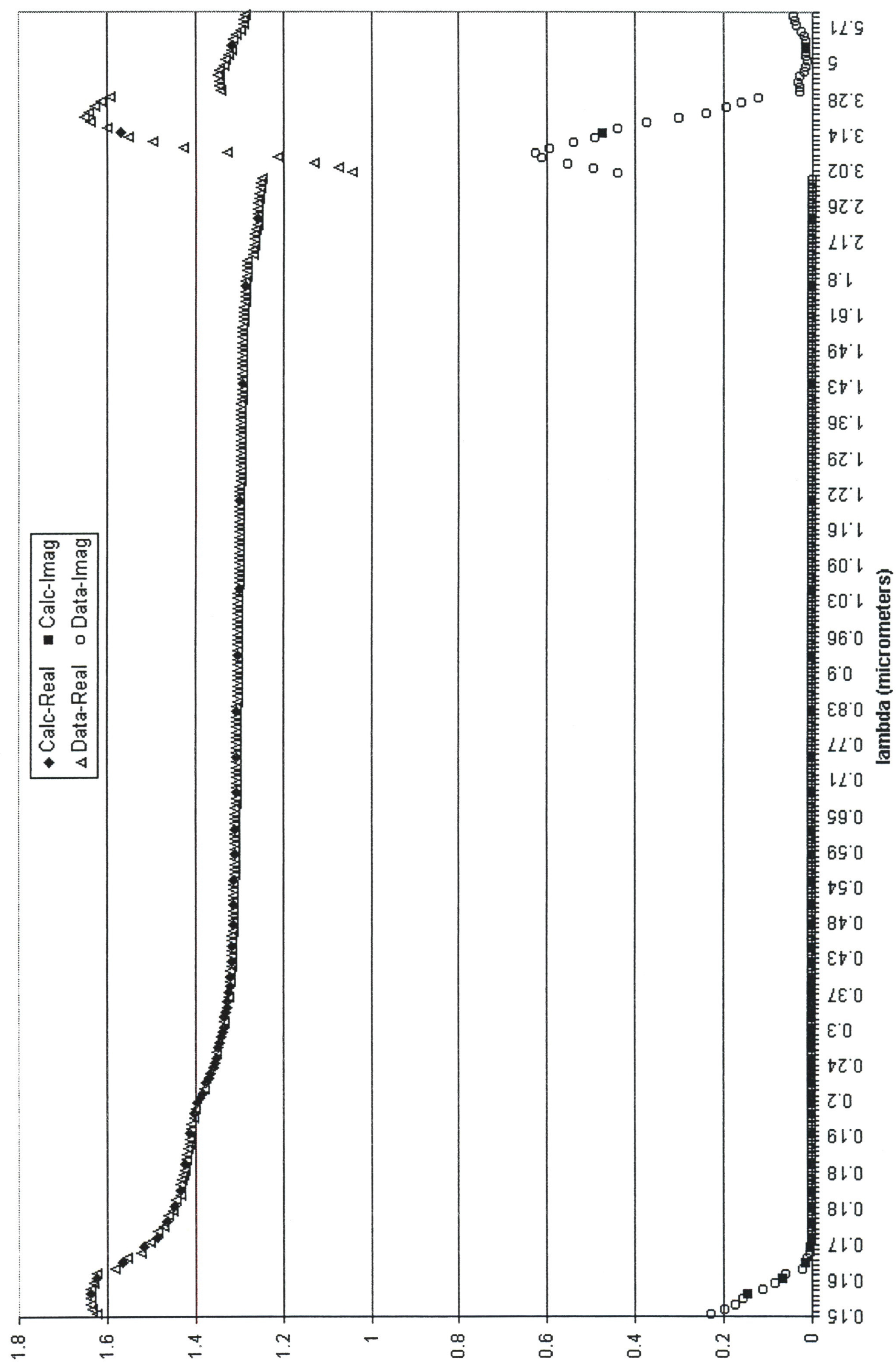


Figure 2
Second Comparison of Calculated and Given n(λ) and k(λ) values

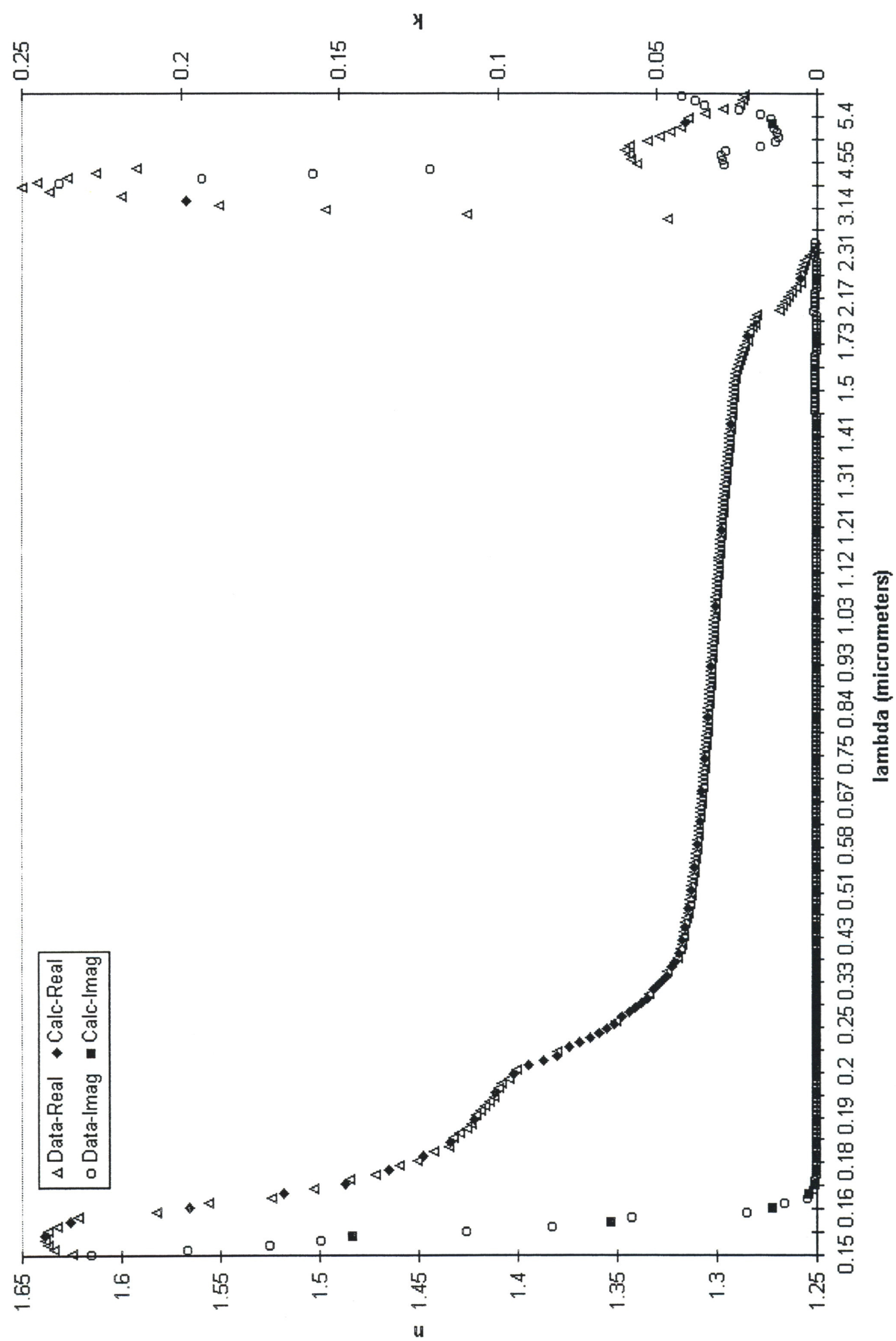


Figure 3
Comparison of Calculated and Given n(λ) and k(λ) values for λ from 0.152μm to 0.748μm

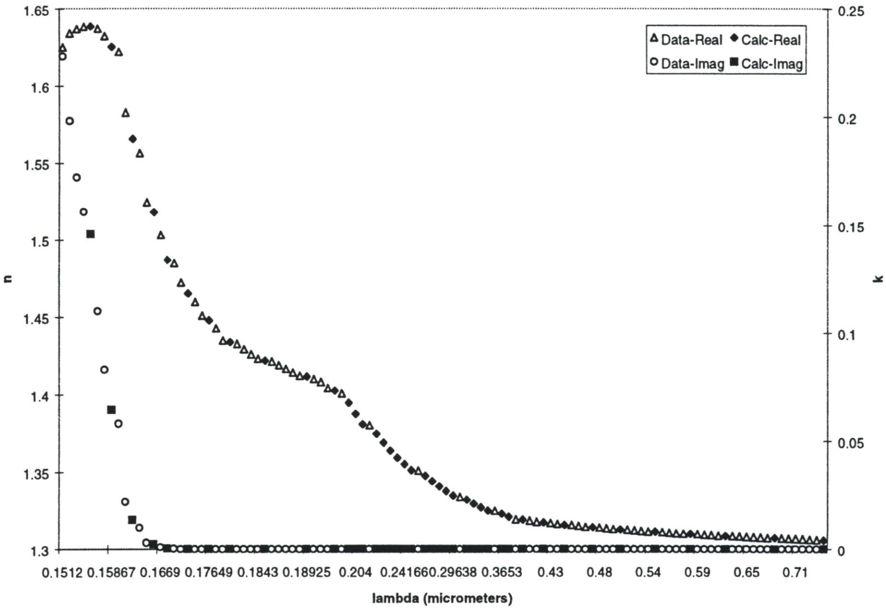


Figure 4
Comparison of Calculated and Given n(λ) and k(λ) values for λ from 0.66μm to 1.85μm

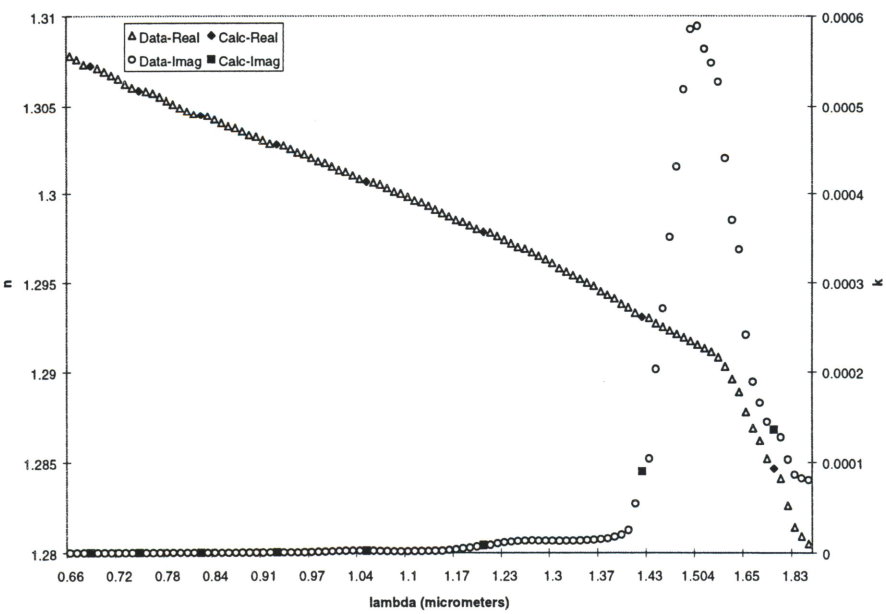


Figure 5
Comparison of Calculated and Given n(λ) and k(λ) values for λ from 2.10μm to 2.35μm

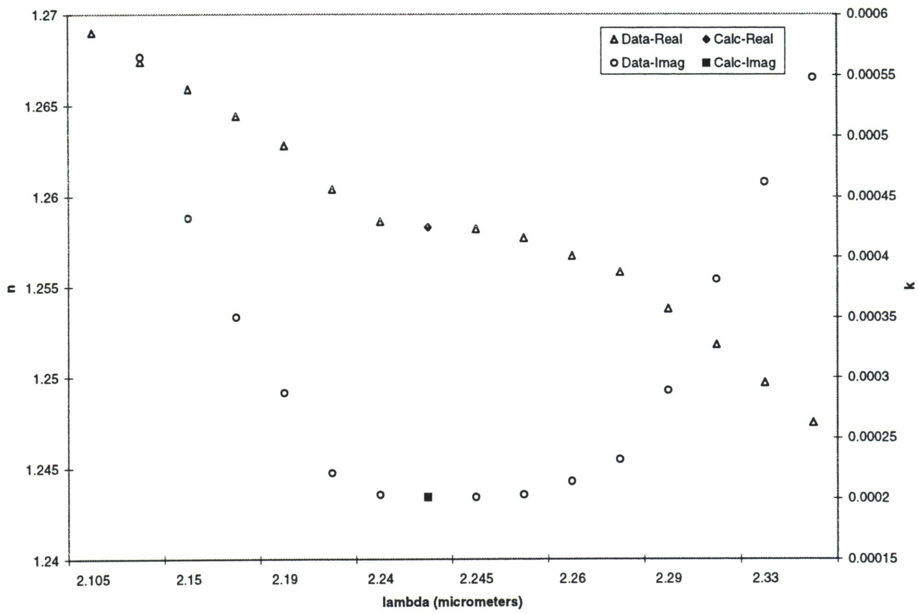


Figure 6
Comparison of Calculated and Given n(λ) and k(λ) values for λ from 3.00μm to 3.28μm

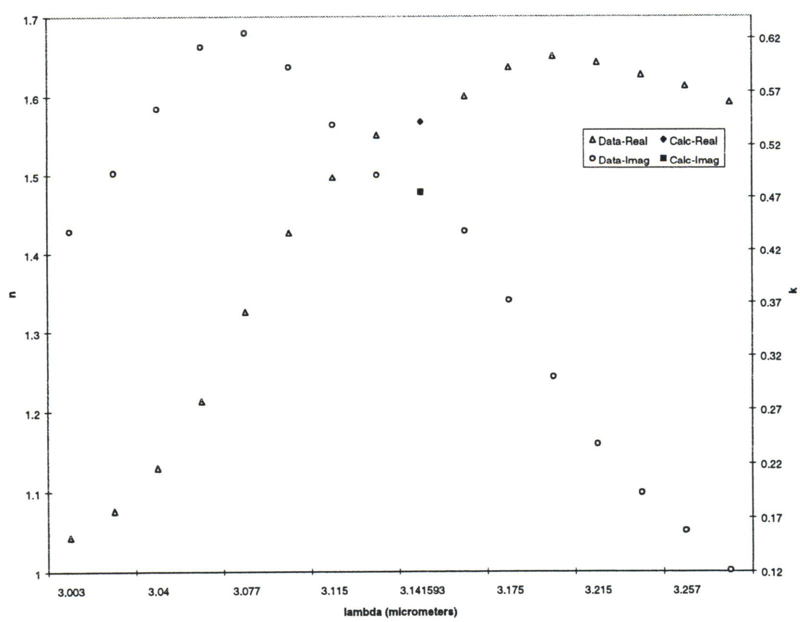


Figure 7
Comparison of Calculated and Given $n(\lambda)$ and $k(\lambda)$ values for λ from 4.5μm to 5.8μm

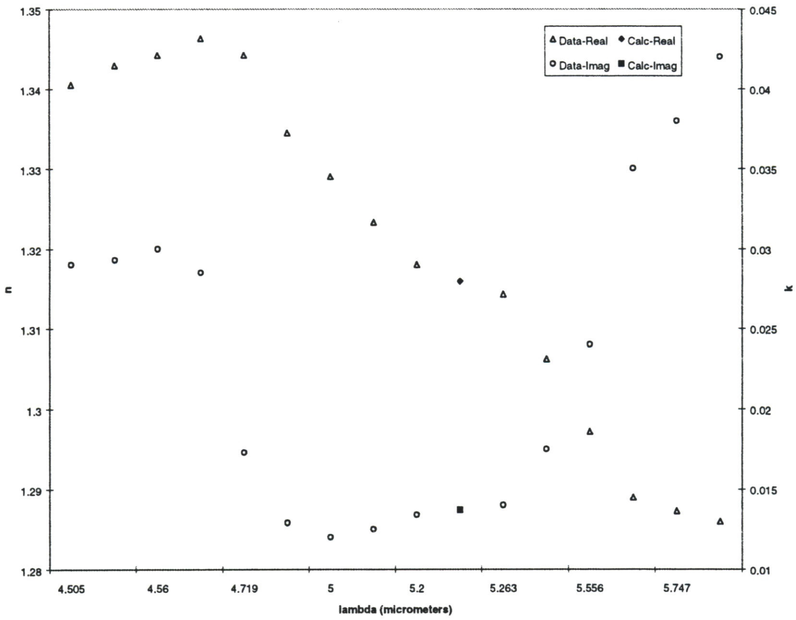


Figure 8
0.25% Error Bars on n Data from 0.19μm to 0.55μm

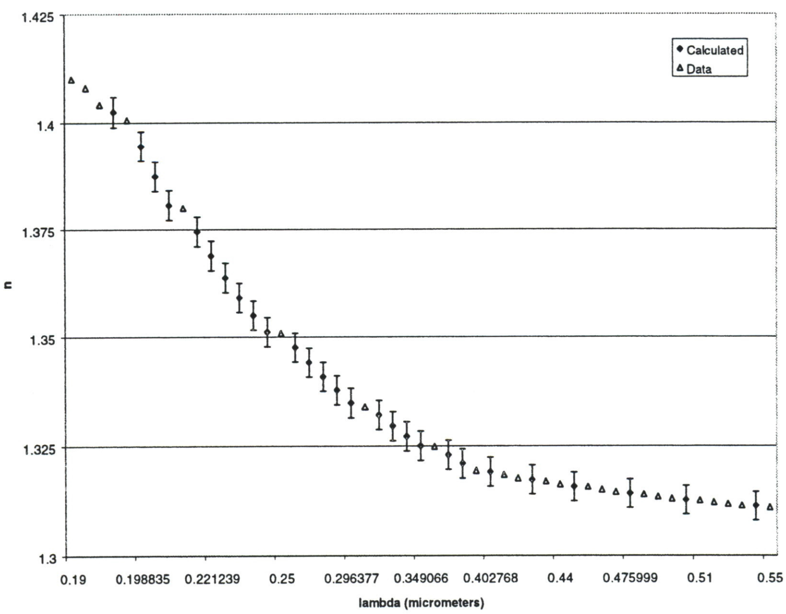


Figure 9
0.75% Error Bars on n Data from 0.75μm up to 2.8μm

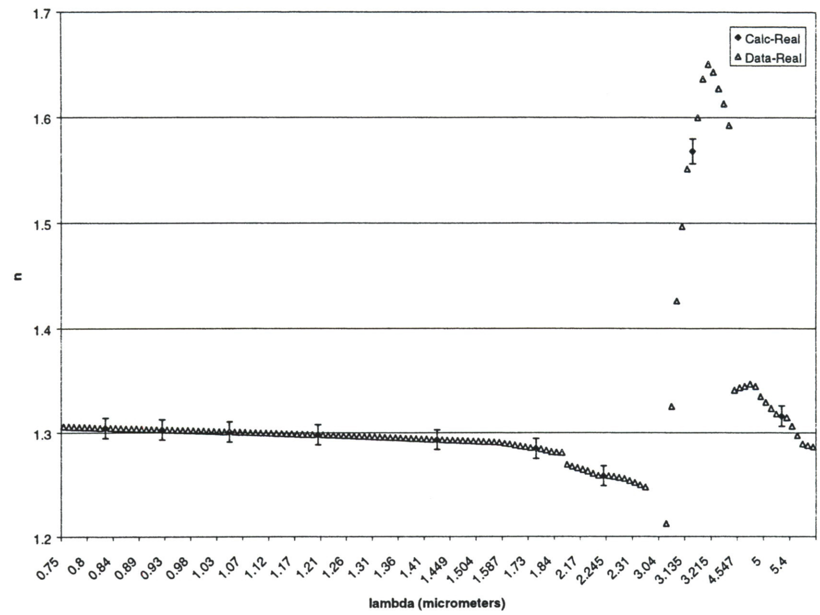


Figure 10
10% Error Bars on k Data from 0.184μm to 0.49μm

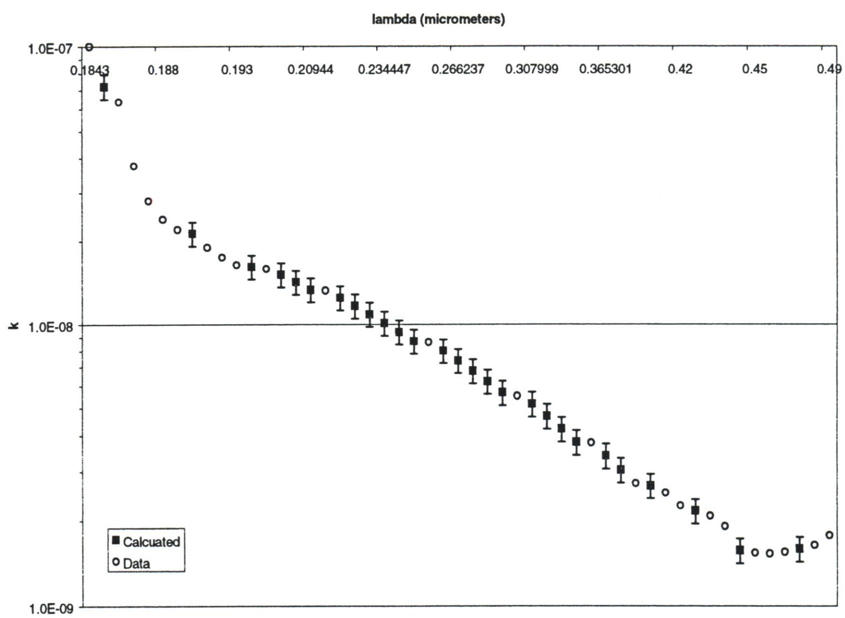


Figure 11
10% Error Bars on k Data from 0.49μm to 0.80μm

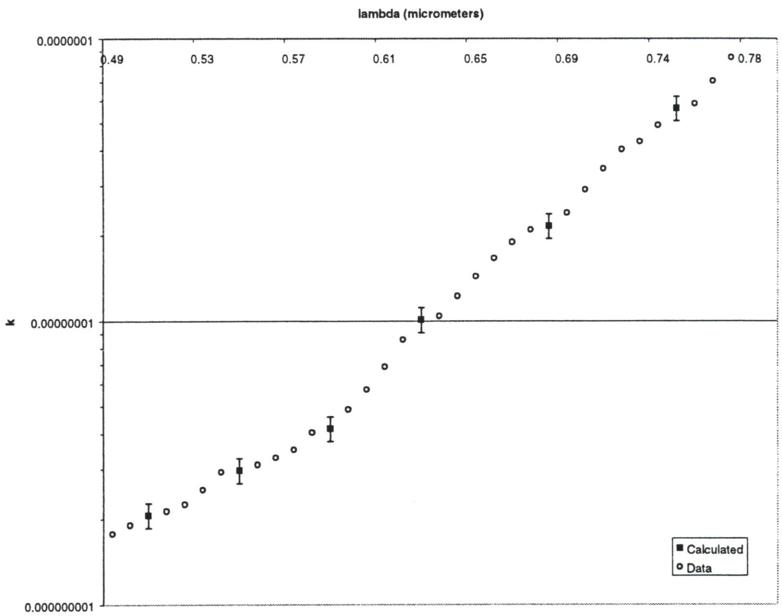


Figure 12
10% Error Bars on k Data from 0.152μm to 0.163μm, 3.0μm to 3.28μm, and 4.5μm to 5.8μm

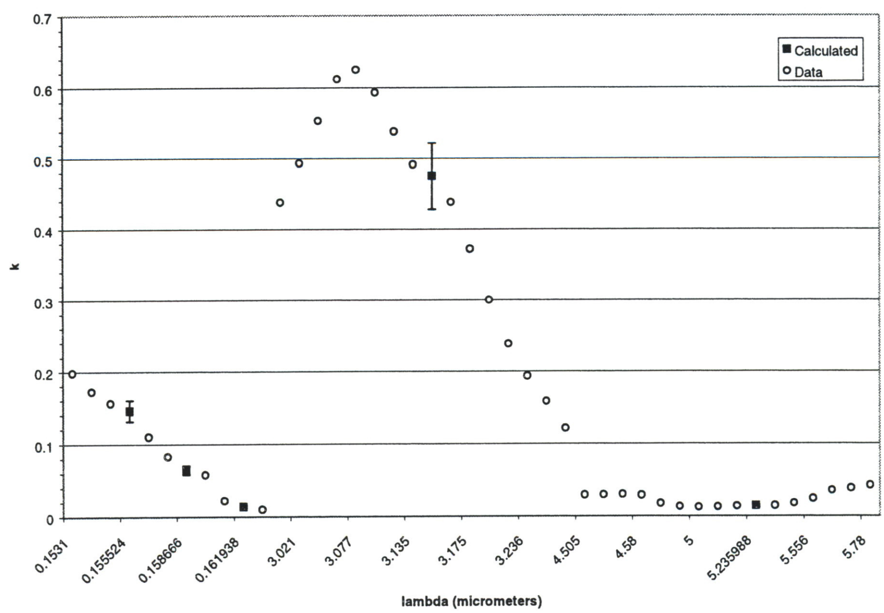
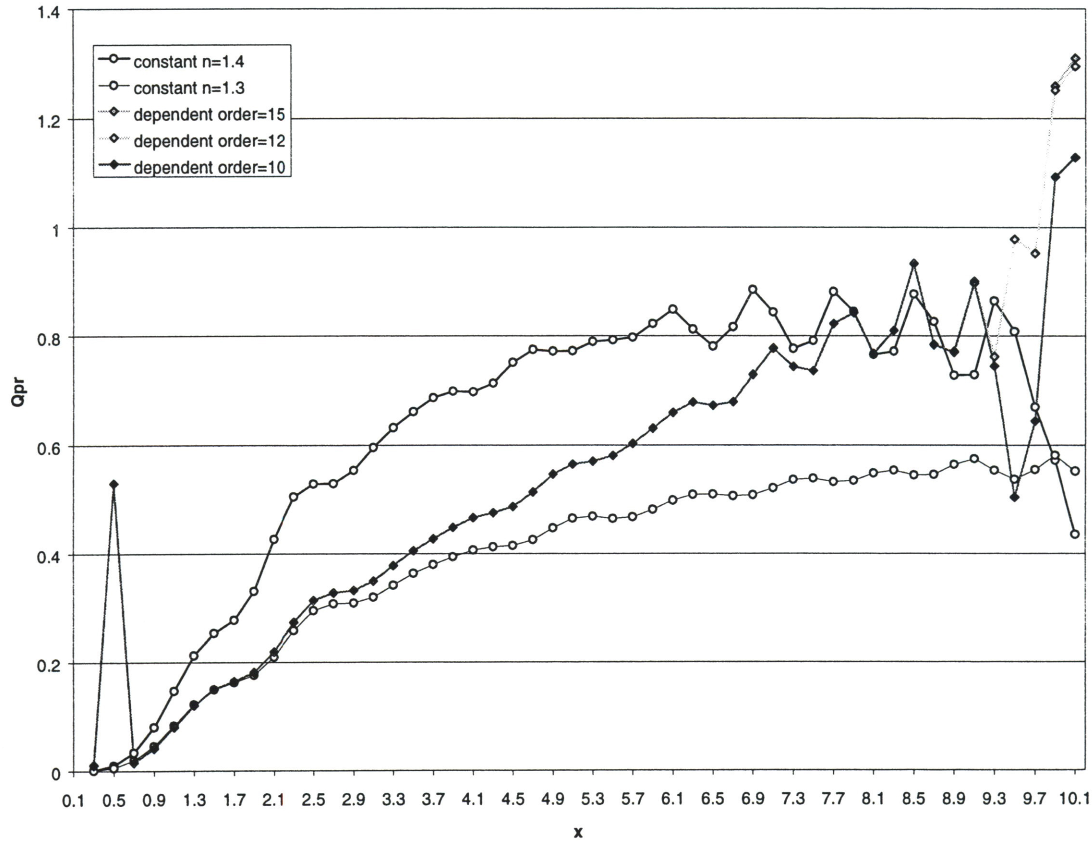


Figure 13
Comparison of Q_{pr} functions using constant and wavelength dependent n and k values
k, when constant, is zero. x ranges from 0.3 to 10.1.



APPENDIX B

Table 1
Values of x, λ, n, and k used in the programs using wavelength dependence

x	λ	n	k
0.3	5.235988	1.315913	1.37E-02
0.5	3.141593	1.567734	0.474845
0.7	2.243995	1.258288	2.01E-04
0.9	1.745329	1.284652	1.36E-04
1.1	1.427997	1.29306	8.97E-05
1.3	1.208305	1.297834	8.32E-06
1.5	1.047198	1.300646	2.23E-06
1.7	0.923998	1.302758	4.88E-07
1.9	0.826735	1.304436	1.45E-07
2.1	0.747998	1.305832	5.66E-08
2.3	0.682955	1.307231	2.16E-08
2.5	0.628319	1.308542	1.01E-08
2.7	0.581776	1.309946	4.17E-09
2.9	0.541654	1.311335	2.97E-09
3.1	0.506708	1.312726	2.07E-09
3.3	0.475999	1.314193	1.59E-09
3.5	0.448799	1.31577	1.57E-09
3.7	0.42454	1.317378	2.16E-09
3.9	0.402768	1.319144	2.66E-09
4.1	0.383121	1.32104	3.03E-09
4.3	0.365301	1.322983	3.40E-09
4.5	0.349066	1.325026	3.79E-09
4.7	0.334212	1.327198	4.22E-09
4.9	0.320571	1.32954	4.67E-09
5.1	0.307999	1.332071	5.16E-09

x	λ	n	k
5.3	0.296377	1.33479	5.67E-09
5.5	0.285599	1.337698	6.21E-09
5.7	0.275578	1.340802	6.77E-09
5.9	0.266237	1.344098	7.38E-09
6.1	0.257508	1.347573	8.01E-09
6.3	0.249333	1.351212	8.68E-09
6.5	0.241661	1.35504	9.38E-09
6.7	0.234447	1.359196	1.01E-08
6.9	0.227652	1.363789	1.09E-08
7.1	0.221239	1.368879	1.17E-08
7.3	0.215178	1.374494	1.25E-08
7.5	0.20944	1.380639	1.33E-08
7.7	0.204	1.387325	1.42E-08
7.9	0.198835	1.394607	1.51E-08
8.1	0.193925	1.402532	1.61E-08
8.3	0.189253	1.411668	2.13E-08
8.5	0.1848	1.421899	7.22E-08
8.7	0.180551	1.433891	8.37E-07
8.9	0.176494	1.447932	6.21E-06
9.1	0.172615	1.465275	4.97E-05
9.3	0.168903	1.487075	3.44E-04
9.5	0.165347	1.518342	2.28E-03
9.7	0.161938	1.565661	1.36E-02
9.9	0.158666	1.625413	6.43E-02
10.1	0.155524	1.638526	0.145613

```
(* Pressure Q factor and Beta - February 1, 2002

    Finally works after many modifications.
      Removed "{" brackets from qpressure equation in integrand.
      Final modification on Apr. 13,
  2002.  May go back to original code later.
      I think that most of this may be Dr. Griffith's beta2 program.
      The original code (his Qpress program) should work,
  too.  The planck and qpr functions are my own code.

      April 22, 2002  Found minus sign error in qpr,
  thanks to Dr. Griffiths.  Now corrected.
      4th version, adjusting below n and k values.  *)

(* The wavelength ranges from about 0.15 microns to 4.00 microns *)

(* This corresponds to X ranging from 10.47 to 0.3917 for a = 0.25\[Mu]
*)

(* In fact, here the range for X is from 10.10 to 0.30 *)

(* Give values for the real (n) and imaginary (k) parts of the index;
  assume k=0 here to roughly match data *)

<<NumericalMath`ListIntegrate`

n = 1.30  (* The real part of the refractive index *)

k =  0.0  (* The imaginary part of the refractive index *)

np =50(* The number of points for the graph *)

no = 12(* The number of orders of the Bessel Functions *)

rhod = 0.92 (*density of ice*)

step = 0.20 (* Step size for the abscissa *)

an1 = x D[Sqrt[0.5 Pi y] BesselJ[i + 1/2,y], y] Sqrt[0.5 Pi x] BesselJ
[i + 1/2, x];

an2 = y Sqrt[0.5 Pi y] BesselJ[i + 1/2, y] D[Sqrt[0.5 Pi x]BesselJ[i +
1/2, x], x];

ant = an1 - an2;

ad1 =  x D[Sqrt[0.5 Pi y] BesselJ[i + 1/2, y], y] Sqrt[0.5 Pi
x](BesselJ[ i + 1/2, x] + I ((-1)^i)  BesselJ[-i - 1/2, x]);

ad2 = y Sqrt[0.5 Pi y] BesselJ[i + 1/2,y] D[Sqrt[0.5 Pi x](BesselJ[ i +
1/2, x] + I ((-1)^i)  BesselJ[-i - 1/2, x]), x];

adt = ad1 - ad2;

ai = ant/adt;
```

```
(* Now, we generate the i + 1 term for the a coefficients. *)

an1p = x D[Sqrt[0.5 Pi y] BesselJ[i + 3/2,y], y] Sqrt[0.5 Pi x] BesselJ
[i + 3/2, x];

an2p = y Sqrt[0.5 Pi y] BesselJ[i + 3/2, y] D[Sqrt[0.5 Pi x]BesselJ[i +
3/2, x], x];

antp = an1p - an2p;

ad1p =  x D[Sqrt[0.5 Pi y] BesselJ[i + 3/2, y], y] Sqrt[0.5 Pi
x](BesselJ[ i + 3/2, x] + I ((-1)^(i + 1))  BesselJ[-i - 3/2, x]);

ad2p = y Sqrt[0.5 Pi y] BesselJ[i + 3/2,y] D[Sqrt[0.5 Pi x](BesselJ[ i
+ 3/2, x] + I ((-1)^(i + 1))  BesselJ[-i - 3/2, x]), x];

adtp = ad1p - ad2p;

aip = antp/adtp;

(* First, we generate the i term for the b coefficients. *)

bn1 = y D[Sqrt[0.5 Pi y] BesselJ[i + 1/2, y], y] Sqrt[0.5 Pi x] BesselJ
[i + 1/2, x];

bn2 = x Sqrt[0.5 Pi y] BesselJ[i + 1/2, y] D[Sqrt[0.5 Pi x]BesselJ[i +
1/2, x], x];

bnt = bn1 - bn2;

bd1 = y D[Sqrt[0.5 Pi y] BesselJ[i + 1/2, y], y] Sqrt[0.5 Pi
x](BesselJ[ i + 1/2, x] + I ((-1)^i)  BesselJ[-i - 1/2, x]);

bd2 = x Sqrt[0.5 Pi y] BesselJ[i + 1/2,y] D[Sqrt[0.5 Pi x](BesselJ[ i +
1/2, x] + I ((-1)^i)  BesselJ[-i - 1/2, x]), x];

bdt = bd1 - bd2;

bi = bnt/bdt;

(* Now, we generate the i + 1 term for the b coefficients. *)

bn1p = y D[Sqrt[0.5 Pi y] BesselJ[i + 3/2, y], y] Sqrt[0.5 Pi x]
BesselJ [i + 3/2, x];

bn2p = x Sqrt[0.5 Pi y] BesselJ[i + 3/2, y] D[Sqrt[0.5 Pi x]BesselJ[i +
3/2, x], x];

bntp = bn1p - bn2p;

bd1p = y D[Sqrt[0.5 Pi y] BesselJ[i + 3/2, y], y] Sqrt[0.5 Pi
x](BesselJ[ i + 3/2, x] + I ((-1)^(i + 1))  BesselJ[-i - 3/2, x]);

bd2p = x Sqrt[0.5 Pi y] BesselJ[i + 3/2,y] D[Sqrt[0.5 Pi x](BesselJ[ i
+ 3/2, x] + I ((-1)^(i + 1))  BesselJ[-i - 3/2, x]), x];

bdtp = bd1p - bd2p;
```

```
bip = bntp/bdtp;

Clear[qpress]


plfunct =
  Table[((0.1 + step j)^3)/(Exp[1.582(0.1 + step j)] - 1), {j, 1, np}]

qpr = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n - I k) (0.1 + step j)}, {i, 1,
no}], {j, 1, np}]

integrand = plfunct qpr


beta = (2.2470/rhod) (ListIntegrate[integrand, step])


planck = Table[{(0.1 + step j), ((0.1 + step j)^3)/(Exp[1.582(0.1 +
step j)] - 1)}, {j, 1, np}]

  qpress = Table[{(0.1 + step j),  Sum [(2 / (x^2))(2i + 1) Re[ai + bi]
- (4 / (x^2)) (i (i + 2)/(i + 1)) (Re[ai] Re[aip] + Im[ai] Im[aip] +
Re[bi] Re[bip] + Im[bi] Im[bip]) - (4 / (x^2))((2i + 1)/(i (i + 1)))
(Re[ai] Re[bi] + Im[ai] Im[bi]) /. {x \[Rule] (0.1 + step j), y \[Rule]
(n - I k) (0.1 + step j)}, {i, 1, no}]}, {j, 1, np}]

ListPlot[qpress, AxesOrigin  \[Rule] {0,0}]

ListPlot[planck, AxesOrigin  \[Rule] {0,0}]
```

```
(* Testing qpress function with n and k values for ice from data.
April 16,
  2002.
      Final adjustment May 2, 2002. *)

(* The wavelength ranges from about 0.15 microns to 4.00 microns *)

(* This corresponds to X ranging from 10.47 to 0.3917 for a = 0.25\[Mu]
*)

(* In fact, here the range for X is from 10.10 to 0.30 *)

(* Give values for the real (n) and imaginary (k) parts of the index *)

<<NumericalMath`ListIntegrate`

(* The real part of the refractive index *)
n1=1.31591336
n2=1.567734
n3=1.258288
n4=1.284651571
n5=1.29306044
n6=1.297834096
n7=1.30064645
n8=1.30275829
n9=1.30443558
n10=1.30583249
n11=1.30723065
n12=1.30854227
n13=1.309946234
n14=1.311334533
n15=1.312726173
n16=1.314192851
n17=1.315769744
n18=1.317377884
n19=1.319144177
n20=1.321039685
n21=1.322983370
n22=1.325025918
n23=1.327198086
n24=1.329540395
n25=1.332071304
n26=1.334790241
n27=1.337697849
n28=1.340801637
n29=1.344097526
n30=1.347573188
n31=1.351211737
n32=1.355039590
n33=1.359196065
n34=1.363788714
n35=1.368878570
n36=1.374493631
n37=1.380638711
n38=1.387324921
n39=1.394607387
n40=1.40253224
```

```
n41=1.4116681
n42=1.4218989
n43=1.4338911
n44=1.44793224
n45=1.46527488
n46=1.48707513
n47=1.5183424
n48=1.5656605
n49=1.6254134
n50=1.6385262


k1=.136934*10^-1
k2=.474845
k3=.20106*10^-3
k4=.1364010*10^-3
k5=.8974*10^-4
k6=.83210*10^-5
k7=.222685*10^-5
k8=.487610*10^-6
k9=.1446659*10^-6
k10=.5659619*10^-7
k11=.2162394*10^-7
k12=.10099060*10^-7
k13=.417479*10^-8
k14=.297342*10^-8
k15=.207023*10^-8
k16=.1592514*10^-8
k17=.157099*10^-8
k18=.21648526*10^-8
k19=.2658815*10^-8
k20=.302725145*10^-8
k21=.339666117*10^-8
k22=.378948065*10^-8
k23=.421501309*10^-8
k24=.467261255*10^-8
k25=.515894832*10^-8
k26=.567008777*10^-8
k27=.620674090*10^-8
k28=.677470713*10^-8
k29=.737657843*10^-8
k30=.801233379*10^-8
k31=.868034106*10^-8
k32=.937918451*10^-8
k33=.1010993762*10^-7
k34=.1087261274*10^-7
k35=.1166595681*10^-7
k36=.1248791850*10^-7
k37=.133359987*10^-7
k38=.142134981*10^-7
k39=.151386769*10^-7
k40=.1613788*10^-7
k41=.21290*10^-7
k42=.7215*10^-7
k43=.83655*10^-6
k44=.6207*10^-5
k45=.49657*10^-4
```

```
k46=.34437*10^-3
k47=.22804*10^-2
k48=.136025*10^-1
k49=.6429*10^-1
k50=.145613
(* The imaginary part of the refractive index *)


np =50(* The number of points for the graph *)

no = 10(* The number of orders of the Bessel Functions *)

rhod = 2.0(*density of dust*)

step = 0.20 (* Step size for the abscissa *)

an1 = x D[Sqrt[0.5 Pi y] BesselJ[i + 1/2,y], y] Sqrt[0.5 Pi x] BesselJ
[i + 1/2, x];

an2 = y Sqrt[0.5 Pi y] BesselJ[i + 1/2, y] D[Sqrt[0.5 Pi x]BesselJ[i +
1/2, x], x];

ant = an1 - an2;

ad1 =  x D[Sqrt[0.5 Pi y] BesselJ[i + 1/2, y], y] Sqrt[0.5 Pi
x](BesselJ[ i + 1/2, x] + I ((-1)^i)  BesselJ[-i - 1/2, x]);

ad2 = y Sqrt[0.5 Pi y] BesselJ[i + 1/2,y] D[Sqrt[0.5 Pi x](BesselJ[ i +
1/2, x] + I ((-1)^i)  BesselJ[-i - 1/2, x]), x];

adt = ad1 - ad2;

ai = ant/adt;

(* Now, we generate the i + 1 term for the a coefficients. *)

an1p = x D[Sqrt[0.5 Pi y] BesselJ[i + 3/2,y], y] Sqrt[0.5 Pi x] BesselJ
[i + 3/2, x];

an2p = y Sqrt[0.5 Pi y] BesselJ[i + 3/2, y] D[Sqrt[0.5 Pi x]BesselJ[i +
3/2, x], x];

antp = an1p - an2p;

ad1p =  x D[Sqrt[0.5 Pi y] BesselJ[i + 3/2, y], y] Sqrt[0.5 Pi
x](BesselJ[ i + 3/2, x] + I ((-1)^(i + 1))  BesselJ[-i - 3/2, x]);

ad2p = y Sqrt[0.5 Pi y] BesselJ[i + 3/2,y] D[Sqrt[0.5 Pi x](BesselJ[ i
+ 3/2, x] + I ((-1)^(i + 1))  BesselJ[-i - 3/2, x]), x];

adtp = ad1p - ad2p;

aip = antp/adtp;

(* First, we generate the i term for the b coefficients. *)
```

```
bn1 = y D[Sqrt[0.5 Pi y] BesselJ[i + 1/2, y], y] Sqrt[0.5 Pi x] BesselJ
[i + 1/2, x];

bn2 = x Sqrt[0.5 Pi y] BesselJ[i + 1/2, y] D[Sqrt[0.5 Pi x]BesselJ[i +
1/2, x], x];

bnt = bn1 - bn2;

bd1 = y D[Sqrt[0.5 Pi y] BesselJ[i + 1/2, y], y] Sqrt[0.5 Pi
x](BesselJ[ i + 1/2, x] + I ((-1)^i)  BesselJ[-i - 1/2, x]);

bd2 = x Sqrt[0.5 Pi y] BesselJ[i + 1/2,y] D[Sqrt[0.5 Pi x](BesselJ[ i +
1/2, x] + I ((-1)^i)  BesselJ[-i - 1/2, x]), x];

bdt = bd1 - bd2;

bi = bnt/bdt;

(* Now, we generate the i + 1 term for the b coefficients. *)

bn1p = y D[Sqrt[0.5 Pi y] BesselJ[i + 3/2, y], y] Sqrt[0.5 Pi x]
BesselJ [i + 3/2, x];

bn2p = x Sqrt[0.5 Pi y] BesselJ[i + 3/2, y] D[Sqrt[0.5 Pi x]BesselJ[i +
3/2, x], x];

bntp = bn1p - bn2p;

bd1p = y D[Sqrt[0.5 Pi y] BesselJ[i + 3/2, y], y] Sqrt[0.5 Pi
x](BesselJ[ i + 3/2, x] + I ((-1)^(i + 1))  BesselJ[-i - 3/2, x]);

bd2p = x Sqrt[0.5 Pi y] BesselJ[i + 3/2,y] D[Sqrt[0.5 Pi x](BesselJ[ i
+ 3/2, x] + I ((-1)^(i + 1))  BesselJ[-i - 3/2, x]), x];

bdtp = bd1p - bd2p;

bip = bntp/bdtp;

Clear[qpress]


q1 = Table[
      Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i
+ 1)) (Re[ai] Re[aip] + Im[ai] Im[aip] +   Re[bi] Re[bip] +  Im[bi]
Im[bip]) - (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai]
Im[bi]) /. {x \[Rule] (0.1 + step j), y \[Rule] (n1 - I k1) (0.1 + step
j)}, {i, 1, no}], {j, 1, 1}]

q2 = Table[
      Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i
+ 1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi]
Im[bip]) - (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai]
Im[bi]) /. {x \[Rule] (0.1 + step j), y \[Rule] (n2 - I k2) (0.1 + step
j)}, {i, 1, no}], {j, 2, 2}]

q3 = Table[
```

```
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i
+ 1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi]
Im[bip]) - (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai]
Im[bi]) /. {x \[Rule] (0.1 + step j), y \[Rule] (n3 - I k3) (0.1 + step
j)}, {i, 1, no}], {j, 3, 3}]

q4 = Table[
   Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n4 - I k4) (0.1 + step j)}, {i,
1, no}], {j, 4, 4}]

q5 = Table[
   Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n5 - I k5) (0.1 + step j)}, {i,
1, no}], {j, 5, 5}]

q6 = Table[
   Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n6 - I k6) (0.1 + step j)}, {i,
1, no}], {j, 6, 6}]

q7 = Table[
   Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n7 - I k7) (0.1 + step j)}, {i,
1, no}], {j, 7, 7}]

q8 = Table[
   Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n8 - I k8) (0.1 + step j)}, {i,
1, no}], {j, 8, 8}]

q9 = Table[
   Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n9 - I k9) (0.1 + step j)}, {i,
1, no}], {j, 9, 9}]

q10 = Table[
   Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n10 - I k10) (0.1 + step j)}, {i,
1, no}], {j, 10, 10}]

q11 = Table[
```

```
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n11 - I k11) (0.1 + step j)}, {i,
1, no}], {j, 11, 11}]

q12 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n12 - I k12) (0.1 + step j)}, {i,
1, no}], {j, 12, 12}]

q13 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n13 - I k13) (0.1 + step j)}, {i,
1, no}], {j, 13, 13}]

q14 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] +   Re[bi] Re[bip] + [bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n14 - I k14) (0.1 + step j)}, {i,
1, no}], {j, 14, 14}]

q15 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n15 - I k15) (0.1 + step j)}, {i,
1, no}], {j, 15, 15}]

q16 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n16 - I k16) (0.1 + step j)}, {i,
1, no}], {j, 16, 16}]

q17 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n17 - I k17) (0.1 + step j)}, {i,
1, no}], {j, 17, 17}]

q18 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n18 - I k18) (0.1 + step j)}, {i,
1, no}], {j, 18, 18}]

q19 = Table[
```

```
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n19 - I k19) (0.1 + step j)}, {i,
1, no}], {j, 19, 19}]

q20 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n20 - I k20) (0.1 + step j)}, {i,
1, no}], {j, 20, 20}]

q21 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n21 - I k21) (0.1 + step j)}, {i,
1, no}], {j, 21, 21}]

q22 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n22 - I k22) (0.1 + step j)}, {i,
1, no}], {j, 22, 22}]

q23 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n23 - I k23) (0.1 + step j)}, {i,
1, no}], {j, 23, 23}]

q24 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + [bi] Re[bip] + Im[bi] Im[bip]) -
(4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /. {x
\[Rule] (0.1 + step j), y \[Rule] (n24 - I k24) (0.1 + step j)}, {i, 1,
no}], {j, 24, 24}]

q25 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n25 - I k25) (0.1 + step j)}, {i,
1, no}], {j, 25, 25}]

q26 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n26 - I k26) (0.1 + step j)}, {i,
1, no}], {j, 26, 26}]

q27 = Table[
```

```
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n27 - I k27) (0.1 + step j)}, {i,
1, no}], {j, 27, 27}]

q28 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n28 - I k28) (0.1 + step j)}, {i,
1, no}], {j, 28, 28}]

q29 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n29 - I k29) (0.1 + step j)}, {i,
1, no}], {j, 29, 29}]

q30 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] +   Re[bi] Re[bip] + Im[bi]
Im[bip]) - (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai]
Im[bi]) /. {x \[Rule] (0.1 + step j), y \[Rule] (n30 - I k30) (0.1 +
step j)}, {i, 1, no}], {j, 30, 30}]

q31 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n31 - I k31) (0.1 + step j)}, {i,
1, no}], {j, 31, 31}]

q32 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n32 - I k32) (0.1 + step j)}, {i,
1, no}], {j, 32, 32}]

q33 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n33 - I k33) (0.1 + step j)}, {i,
1, no}], {j, 33, 33}]

q34 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n34 - I k34) (0.1 + step j)}, {i,
1, no}], {j, 34, 34}]

q35 = Table[
```

```
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n35 - I k35) (0.1 + step j)}, {i,
1, no}], {j, 35, 35}]

q36 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n36 - I k36) (0.1 + step j)}, {i,
1, no}], {j, 36, 36}]

q37 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n37 - I k37) (0.1 + step j)}, {i,
1, no}], {j, 37, 37}]

q38 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n38 - I k38) (0.1 + step j)}, {i,
1, no}], {j, 38, 38}]

q39 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n39 - I k39) (0.1 + step j)}, {i,
1, no}], {j, 39, 39}]

q40 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n40 - I k40) (0.1 + step j)}, {i,
1, no}], {j, 40, 40}]

q41 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n41 - I k41) (0.1 + step j)}, {i,
1, no}], {j, 41, 41}]

q42 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n42 - I k42) (0.1 + step j)}, {i,
1, no}], {j, 42, 42}]

q43 = Table[
```

```
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n43 - I k43) (0.1 + step j)}, {i,
1, no}], {j, 43, 43}]

q44 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n44 - I k44) (0.1 + step j)}, {i,
1, no}], {j, 44, 44}]

q45 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n45 - I k45) (0.1 + step j)}, {i,
1, no}], {j, 45, 45}]

q46 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n46 - I k46) (0.1 + step j)}, {i,
1, no}], {j, 46, 46}]

q47 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n47 - I k47) (0.1 + step j)}, {i,
1, no}], {j, 47, 47}]

q48 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] +   Im[bi]
Im[bip]) - (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai]
Im[bi]) /. {x \[Rule] (0.1 + step j), y \[Rule] (n48 - I k48) (0.1 +
step j)}, {i, 1, no}], {j, 48, 48}]

q49 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n49 - I k49) (0.1 + step j)}, {i,
1, no}], {j, 49, 49}]

q50 = Table[
    Sum [(2 / (x^2))(2i + 1) Re[ai + bi] - (4 / (x^2)) (i (i + 2)/(i +
1)) (Re[ai] Re[aip] + Im[ai] Im[aip] + Re[bi] Re[bip] + Im[bi] Im[bip])
- (4 / (x^2))((2i + 1)/(i (i + 1))) (Re[ai] Re[bi] + Im[ai] Im[bi]) /.
{x \[Rule] (0.1 + step j), y \[Rule] (n50 - I k50) (0.1 + step j)}, {i,
1, no}], {j, 50, 50}]
```

```
qpr = Table[{q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12, q13,
q14, q15, q16, q17, q18, q19, q20, q21, q22, q23, q24 , q25, q26, q27,
q28, q29, q30, q31, q32,q33, q34, q35, q36, q37, q38, q39, q40, q41,
q42, q43, q44, q45, q46, q47, q48, q49, q50}]
```

```
(* Further Modification of qpress_planck3_beta_final_working.
      Adding in new qpr function,
  a table of values from qpress_final_test2.
      Plot of Qpr is to be done using MS Excel.  April 20, 2002 *)


(* The wavelength ranges from about 0.15 microns to 4.00 microns *)

(* This corresponds to X ranging from 10.47 to 0.3917 for a = 0.25\[Mu]
*)

(* In fact, here the range for X is from 10.10 to 0.10 *)

(* Give values for the real (n) and imaginary (k) parts of the index *)

<<NumericalMath`ListIntegrate`


np =50(* The number of points for the graph *)

no = 10(* The number of orders of the Bessel Functions *)

rhod = 0.92(*density of ice*)

step = 0.20 (* Step size for the abscissa *)

plfunct =
  Table[((0.1 + step j)^3)/(Exp[1.582(0.1 + step j)] - 1), {j, 1, np}]

qpr =
Table[{0.010421396356131696,0.5293231048667987,0.014594656423937641,
      0.04135246819448386,0.0795716338972744,0.12020243945520165,
      0.1497674027918325,0.16492966726550673,0.1812990887457252,
      0.21930321789244042,0.27334650231355617,0.31345746863574764,
      0.3272856295673084,0.3319163773599328,0.3492257976183164,
      0.3780273762310032,0.40525030107332716,0.42714429132619197,
      0.44810536961352865,0.4659033478156193,0.4750677539613635,
      0.48637730823691866,0.5133289172598007,0.5463928406320763,
      0.5646092618021665,0.570056239581049,0.5809073317790873,
      0.6031798239435618,0.6310231780608218,0.659400121359017,
      0.678463831239897,0.6729775074038127,0.6794636856598216,
      0.7297060081264833,0.777416164921534,0.7438393600231517,
      0.7362777735177926,0.8221276636959161,0.8426975269769251,
      0.7683383156098996,0.8100416033481724,0.9335448684096527,
      0.7843794152030004,0.7707948680452357,0.8979225178211006,
      0.7451802729332048,0.5034700402726577,0.6435785527148196,
      1.0929128332454132,1.1285159020514284}]

integrand = plfunct qpr


beta = (2.2470/rhod) (ListIntegrate[integrand, step])


planck = Table[{(0.1 + step j), ((0.1 + step j)^3)/(Exp[1.582(0.1 +
step j)] - 1)}, {j, 0, np}]
```

```
ListPlot[planck, AxesOrigin  \[Rule] {0,0}]
```

```
(* Attempt to determine percent of blackbody curve used in program.
      April 13, 2002.

    April 29,
  2002.  Changed second and third functions to cover range < x=0.1.  *)

<<NumericalMath`ListIntegrate`

np = 50
np2 = 1000
np3 = 10000

step = 0.20 (* Step size for the abscissa *)


plfunct =
    Table[((0.1 + step j)^3)/(Exp[1.582(0.1 + step j)] - 1), {j, 1,
np}];


blbody0 = (ListIntegrate[plfunct, step])


planck1 =
    Table[(( 0.00001+step j)^3)/(Exp[1.582(0.00001+step j)] - 1), {j,
0, np2}];

 blbody1 = ListIntegrate[planck1, step]

(blbody1 - blbody0)/blbody1*100

planck2 =
    Table[((0.00001+step j)^3)/(Exp[1.582(0.00001+step j)] - 1), {j, 0,
np3}];

 blbody2 = ListIntegrate[planck2, step]

(blbody2 - blbody1)/blbody2*100

(blbody2 - blbody0)/blbody2*100
```