Some recent activities of Phil Lucht **1.10.03**

I used to teach physics at the University of Utah, but then spent about 15 years doing hardware and software design in the "video industry". For the last 4 years I have worked as a contract engineer for our company Rimrock Digital Technology. In the last several months, motivated as described below, I have been doing physics again, with an emphasis on optics and related programming. I find that I still enjoy looking up integrals in my Gradshetyn and Ryzhik, as well as making physical models to describe things I see in the real world. By coincidence, Bob Cady mentioned to me that Orbtek was looking for someone to work on an optics/programming project. Perhaps I can help.

Floaters as Motivation. During the last several years I have been experiencing "floaters" in both my eyes. They have the appearance of dim transparent disks of small size, hundreds in each eye. I believe these are red blood cells which have bled from retinal blood vessels into a liquefied region between the retina and the vitreous membrane. They may diffuse into the hyaluronic matrix of the vitreous, or may exist in liquefied pockets within the matrix. In this low-salt environment, osmosis causes blood cells to bloat out from 5μ crenellated disks to 8μ spheres. The hemoglobin all leaks out through the stretched membrane, perhaps explaining why these floaters appear colorless to the viewer. They only appear at all because, being in a liquid instead of a gel, they can move relative to the retina, getting through the eye/brain filtering system which normally blocks visibility of small static objects. The floaters become visible when the intensity of their shadows on the retina differs by more than roughly 1/4 of 1% from to the background intensity, depending somewhat on lighting level and shadow diameter, according to the so-called contrast sensitivity function (CSF) of the eye. Surprisingly, my floaters periodically become very annoying, and then become completely invisible for weeks at a time. This has motivated me to try to construct a "physics model" for what I see. Floaters are of very little interest to ophthalmologists because they are fairly benign and there are no standard treatments to remove them. There is of course much web activity on the subject.

Physics work done on a Floater Model. I first reviewed my knowledge of optics, doing readings in my "Jackson" E&M book, then in Goodman's Fourier Optics. I learned that Born and Wolf have a new 7th hardback edition in Oct 1999 for a bargain Amazon price of $65, Hecht at $106, but did not buy, reading at Marriott library instead. For a simple model, I made the floater be an opaque disk so I could apply the usual scalar Kirchhoff or Sommerfeld integrals to a hole in a screen, then I used Babinet's Principle to get the field of the complementary disk, both in terms of a far-off point source. I did the Fresnel and then Fraunhofer limits of the integral to recover the Airy disk pattern of the hole. The Babinet intensity pattern of a disk is then $|1 - \text{Airy}|^2$ where Airy has the usual quadratic phase and $J_1(x)/x$. The interference here causes the disk diffraction pattern to have a Poisson/Arago bright spot in the center, which is a bit counter-intuitive if one thinks incorrectly of $1 - |\text{Airy}|^2$. My floaters have 1 or 2 very clear diffraction rings, and some have bright spots in the center. Next, I generalized the model to an extended incoherent light source (blue sky, computer screen. etc) formed by the cone of light passing through the pupil and terminating on a floater blood cell. This required a two dimensional integration over the incoherent source of the Babinet/Airy intensity from a disk.

Computer work done on a Floater Model. I started with Maple release V (a competitor of Mathematica), but found that Maple was not giving stable integration results for the accelerating highly oscillatory function $|1 - \text{Airy}|^2$. I then learned Maple's Algol68-like programming language and wrote my own approximate integration routine. It was at that point that I learned that even with the so-called evalfh

operator, my Maple release cannot do Bessel functions using floating point hardware, so the result was painfully slow. I then re-coded this routine in Visual Basic that I normally use with Excel, and things sped up by a factor of 50, despite the well-known slowness of Visual Basic. At least VB uses the Pentium FPU. Since I needed still more speed, I next used a trick from my past and wrote a custom DLL containing C++ routines that are callable from Excel. This provided another factor of 500 speedup. With all this in place, I was able to make an Excel spreadsheet and set in parameters like pupil diameter, distance z of floater from the retina, wavelength, floater radius, and so on, and was able to generate and plot the profile of floater shadows on the retina when the eye is looking at an incoherent uniform light source such as the blue sky (where floaters are most visible). The result is that the detailed shape of a floater shadow even in the Fraunhofer regime is highly variable and dependent on distance z: some shadows have bright spots in the center, others have dark spots in the center, and so on. As expected, outer diffraction rings on the floater shadows are quickly extinguished as the pupil opens wider.

A Microsoft Surprise. In C, it is sometimes useful to use "long doubles" to verify the accuracy of a complicated "double" precision calculation. It is well known that IEEE floating point hardware units such as that in the Pentium type CPU normally use 80-bit numbers internally, regardless of what accuracy one requires. In C or C++ one should be able to declare variables to be long doubles and the compiler should then take full advantage of the 80-bit computations, albeit with some speed cost due to the 64-bit Pentium data bus size. The surprise was that, on 32-bit machines, Microsoft compilers such as that in Visual C++ force a reduction of long doubles to regular 64-bit doubles (with no possible override), causing a loss of 12 bits of mantissa. However, gnu and Borland compilers allow real 80-bit long doubles.

Future Work. Floater blood cells are really spheres, not disks, and my crude disk model for a floater does not work well if the floater is partially transparent -- I don't think Kirchhoff and Babinet can be applied in this case. So I have been thinking of improving things by modeling the floater as a dielectric sphere of index close to that of the vitreous and using the electric dipole polarization radiation formulas to get the diffraction pattern. The second improvement is that with the high-speed Excel-DLL C-coding method, I can probably do the radiation integrals numerically and thereby get results in all zones, not just the Fraunhofer. Thirdly, I need to learn more about the eye's contrast sensitivity function as applied to isolated stimulus patterns such as disks. The traditional CSF function is based on 1D modulated sine patterns.