Draft June 22, 2012

Fact-finding about turbulence codes for the atmosphere module (Issue #169)

ABSTRACT

This document describes the results of a fact-finding exercise undertaken by Jörg Dietrich and Barney Rowe for adding the stochastic modelling of PSFs due to turbulent atmospheres to GalSim. This investigation included making a preliminary judgement as to what might be used or borrowed from existing codes that are publicly available and suitably licensed.

1. Motivation

The Code Plan document drawn up with the help of the full GREAT3 collaboration stated one of the basic requirements for the software to be:

> *2. The ability to generate realistic PSF models, by which we mean a physically-motivated model combining both an atmospheric component and a specifiable telescope component, as well as simple parameterized PSF models.*[1]

One of the primary initial goals of the GalSim project is addressing the software needs of GREAT3, so there is interest in tackling this problem from the lensing community. The code is already capable of generating PSFs due to aberrated telescope optics and combining these with physically well-motivated approximations to an ensemble average atmospheric seeing kernel, broadly equivalent to long duration exposures. These models take the form of the Double Gaussian and Kolmogorov PSFs currently implemented in GalSim.

However, the code cannot currently generate a PSF that encompasses a model of *instantaneous* atmospheric turbulence in order to create a more realistic, exposure time-dependent PSF model. This is a regime of interest when modelling short-exposure survey imaging from the ground, as is being seriously contemplated by a several upcoming surveys. These PSFs may look instantaneously much 'better' or 'worse' than the averaged pattern, and the impact this variability has on survey data analysis is not yet fully quantified.

[1] Full plan document available on the Wiki page:
http://great3.pbworks.com/w/page/49339085/Plan%20documents%20for%20the%20GREAT3%20Code%20and%20Challenge

2. Simplified physical model

2.1. Phase screens

Photons passing through the atmosphere encounter a complex, dynamic system of variable refractive index in three dimensions. An approximate model that has given good results in simulations of adaptive optics is to consider the propagation of a plane wavefront through multiple, discrete, planar *phase screens* to mimic the dominant phase distortion effects of the atmosphere. This model is seemingly used in most codes, although see also Section 2.4.

These phase screens are typically modelled as independent realizations of a random field with a theoretically-motivated power spectrum, such as the von Kàrmàn:

$$P_\phi(\nu) = 0.023 r_0^2 \left(\nu^2 + 1/L_0^2\right)^{-11/6}, \tag{1}$$

where r_0 is the wavelength-dependent *Fried parameter* in metres, ν is a 2D spatial frequency in the plane of the phase screen in units of cycles metre^{-1}, and L_0 is the *outer scale* of turbulence in metres. Physically, L_0 corresponds to the largest size of turbulent cells in the atmosphere, the scale on which progenitor eddies are created, measured to be $L_0 \sim$ 1-100m depending on altitude and atmospheric conditions. For $\nu \gg 1/L_0$, the scale-free Kolmogorov turbulence spectrum is recovered. As expressed, the units of $P_\phi(\nu)$ are (radians of wavefront phase)^2/(cycles metre^{-1})^2.

Generating images of phase screen realizations $\phi(x, y)$ with a power spectrum such as $P_\phi(\nu)$ is relatively straightforward using the discrete Fourier transform (DFT):

- Generate an image of a circular complex Gaussian random field with real and imaginary parts having unit variance and zero mean, and associate $[i, j]$ pixel locations with spatial frequencies $[\nu_x, \nu_y]$.
- Multiply this image by the square root of the power spectrum corresponding to $\nu^2 = \nu_x^2 + \nu_y^2$ at each location.
- The DFT of the result provides in its real and imaginary parts two independent images of phase screens $\phi(x, y)$ with the appropriate statistical properties.

Some care must be taken in ensuring that all scales are sufficiently well-represented in this model. The smallest physical scales (the phase screen image pixel scale) should be chosen to correspond to the Nyquist frequency for the eventual angular bandlimit of the telescope PSF, meaning this choice may therefore also be dependent on the phase screen altitude. The largest scales (the image extent in metres) should be large compared to L_0 to ensure that the lowest frequency modes are not heavily folded. Mitigating strategies for efficiently overcoming this latter requirement are discussed in Lane, Glindemann & Dainty (1992), but it is not clear whether this will be an issue given advances in computing resources over the last two decades. However, phase screen motion due to wind (see Section 2.3) also increases the required image extent.

2.2. Propagation between multiple phase screens

An approximate model of the phase distortion effects of a finite depth atmosphere can then be constructed by combining multiple phase screens at different altitudes. The simplest approximation to this process is simply to take a weighted average of multiple phase screens.

A more accurate approach is to model the propagation of light between screens using the Fresnel approximation to the Huygens' integral for a complex wavefront propagating paraxially from a plane at $z = 0$ to a plane at $z = H$. If an infinite uniform plane wavefront has a field distribution $\mathcal{E}(x, y, z = 0)$, then its Fourier transform in the plane at $z = H$ is given by

$$\tilde{\mathcal{E}}(\nu_x, \nu_y, H) = \mathrm{e}^{-\mathrm{i}\pi\nu^2 H\lambda} \times \tilde{\mathcal{E}}(\nu_x, \nu_y, 0), \tag{2}$$

omitting the wavefront amplitude distortions in this approximate treatment (***BARNEY TODO: Must check this calculation thoroughly and will add refs.***). Considering phase only, we will set $\mathcal{E}(x, y, 0) = \mathrm{e}^{\mathrm{i}\phi(x,y,0)}$ and $\mathcal{E}(x, y, H) = \mathrm{e}^{\mathrm{i}\phi(x,y,H)}$ when using equation (2) to propagate distortions from one screen to the next. For separations H between phase screens of order kilometres at optical wavelengths, it can be seen that the modifications of the phase pattern caused by Fresnel diffraction will be small, particularly for the lowest $\nu \sim 1/L_0$ modes which contain the greatest power in $P_\phi(\nu)$.

To propagate through multiple phase screens using the Fresnel approximation will therefore multiple DFT calculations at each layer, one for generating the initial screens as described in the previous Section, and another two (forward and inverse) for propagating light according to equation (2). It remains to be tested whether the Fresnel diffraction step can be safely omitted to speed things up while retaining sufficient accuracy.

It should finally be noted that the atmospheric PSF incident across the telescope field of view needs to be given in angular units, requiring appropriate conversion for each phase screen and further approximation if using Fresnel diffraction.

2.3. Screen motion due to wind

To simulate time variation in the patterns, phase screens can be 'rolled' using assumed wind velocities for each layer of the atmosphere, summing to generate an exposure time-dependent average pattern. The time step between these summations depends on the wind velocity and the pixel scale of the phase screens, which in turn is governed by the resolution of the telescope, as discussed in Section 2.1. Time steps should be short enough that the pixel shift between summations is ~ 0.5 pixels. As the power spectrum $P_\phi(\nu)$ due to turbulence is isotropic in general it may be possible to approximate independent wind velocities at multiple layers using rectangular strips of phase screen realizations rather than large square images. This may be a useful computational saving as the need to roll phase screens might add significantly to their extent.

2.4. A non-DFT approach?

It is tantalisingly hinted in documents describing the Large Synoptic Survey Telescope (LSST) ImSim data simulator[2] that these authors use a different method:

> *"The single photon history is traced through each layer of the atmosphere via a newly invented technique that avoids the need to do Fourier transforms of the wavefront perturbations. The approach is only valid for large aperture telescopes with exposure times of at least 10-20 seconds."*

It would, of course, be very interesting to learn more about this technique if it is available in the public domain.

3. Some existing codes

The adaptive optics community has created a large variety of codes, most of which are private and were never intended for public release. Two codes that are publicly available are

- Arroyo – http://cfao.ucolick.org/software/arroyo.php

  A C++ class library for time domain studies of electromagnetic waves. A very large and exhaustive library, which goes well beyond what we believe we need in GalSim.

- LAOPack – http://lao.ucolick.org/data/Pyramid/MTF%20Data/Code/LAOPack/LAO/pro/... simulationAnalysisTools.html

  A collection of IDL routines for adaptive optics, including the generation of Kolmogorov/von Kàrmàn phase screens and the propagation of photons.

Also known to exist is

- ImSim – http://lsst.astro.washington.edu/

  The image simulation software for the LSST, which apparently implements an entirely new method mentioned above.

4. Preliminary conclusions

None of the existing codes explored appeared to represent an ideal fit for GalSim. They were found to add a significant extra dependency (e.g. Arroyo, ImSim) and are perhaps over-powered

[2] http://lsst.astro.washington.edu/docs/overview/

(Arroyo) or specifically tailored to a single instrument (ImSim). Or they were found to use a non-supported language (IDL in the case of LAOPack).

However, given the relative simplicity of the commonly-adopted physical model described above, it should be feasible to write a home-grown turbulence code that is a better match to GalSim requirements. Much of the computation would proceed using similar DFT techniques to those used in the `galsim/optics.py` and `galsim/atmopshere.py` modules already. There is freedom in the construction of a multiple phase screen atmosphere model, including choice of power spectra, number of screens, screen altitudes, wind velocities etc., and so some thought will need to go into the user interface to make this powerful and while being as clean as possible.

It would be interesting to learn more about the new method developed by LSST, and so Jörg and Barney would welcome any additional information from those more closely connected with the LSST team, or some more thorough documentation of the photon shooting method adopted. It may be possible to implement both approaches, DFT and photon shooting, mirroring the twin implementations of object rendering in GalSim.