# AR-atmospheres-py

Python version of autoregressive atmosphere generator

Relevant paper:

\_A computationally efficient autoregressive method for generating phase screens with frozen flow and turbulence in optical simulations\_

Srikar Srinath, Univ. of California, Santa Cruz; Lisa A. Poyneer, Lawrence Livermore National Lab.;

Alexander R. Rudy, UCSC; S. Mark Ammons, LLNL

Published in Optics Express 23, 33335-33349 (2015)

http://dx.doi.org/10.1364/OE.23.033335

##### Programs for basic operations:

\_make\_ar\_atmos\_: A program that generates a datacube (2 spatial dimensions x time) in fits format (hdf5 outout not implemented yet)

of stacked phase screens with the power-scaling law supplied (Kolmogorov, von Karman etc.) and specified turbulence strength.

Required inputs are exposure time (in seconds), system rate (in Hz), alpha magnitude (< 1), number of subapertures across the

wavefront sensor (n) and resolution of the output phase screen (m, pixels / subaperture). The eventual datacube is (n \* m) x

(n \* m) x (exposure time \* rate). For non-AO applications, (n, m) and (expsore time, rate) could just be combined into one

variable. Simulations where memory is not constrained, but computation resources are sould use this to generate a phase screen

cube beforehand.

\_create\_multilayer\_phasecube\_: AR phase screen generator for use in LSST sims by Jim Chiang

Use these two functions in connjunction for AR phase screen generation & evolution on the fly in simulations. Allows

for arbitrarily long exposure times (where arbitrary is limited by storage and a few other constraints.).

\_create\_multilayer\_arbase\_: Use outside the simulation loop to generate alpha vectors and powerlaw scaling for each layer

\_get\_ar\_atmos\_: Takes initial FT-ed phase screen (generated by \_get\_phase\_streamlined\_, for example), alpha and powerlaw (from

\_create\_multilayer\_arbase\_) and returns new phase and FT of phase for the next time step in a simulation. The first time

this function is called, input FT of phase can just be an array of zeroes.

\_ARscreens\_: Class structure for AR screen generation by Jim Chiang

##### Utilities:

\_cdr\_create\_parameters.py\_: Returns an array with three selected wind layer characteristics (measurements based on GPI data).

For each selected layer, the array contains r0, wind velocity (m/s), wind direction (degrees) and layer altitude (m).

\_check\_ar\_atmos\_: Currently set up for GPI only. Checks phase variance and PSDs of phase screen data cubes for sanity.

\_depiston\_: Removes piston from residual phase. Takes a 2-D phase array as input and, optionally, the aperture mask.

\_detilt\_ : Remves tip/tilt from residual phase. Takes a 2-D phase array and the aperture mask as inputs.

\_gen\_avg\_per\_unb\_: Generates an averaged, unbiased periodogram. Takes 1-D data, a periodogram length and various options such as

half overlapping (to improve SNR), mean removal, hanning or hamming windows, whether to use fftw or fft (in python the scipack

fft utilities are just as efficient as fftw).

\_generate grids\_: Generates x, y grids that are needed for things like spatial frequency grid generation etc.

\_get\_phase\_streamlined\_: Returns a phase screen of given dimension (n x n) starting from scaled (Kolmogorov power spectrum),

white noise. Used to generate the starting phase screen in any simulation/datacube.

\_radialprofile\_: return an azimuthally averaged radial profile of a 2-D array. Useful for PSF profile analysis. Initial source

of power differences found between AR and frozen flow phase screens

\_rebin\_: Resizes a 2d array by averaging or repeating elements.