Draft version March 4, 2013
Preprint typeset using LaTeX style emulateapj v. 11/10/09

ECLIPSE MAPPING CODE

Woody Austin

Draft version March 4, 2013

ABSTRACT

I describe a C code that can read in time series photometry data of a transiting planet host star and produce a two dimensional surface brightness map of it. Using the Nelder-Mead Simplex Algorithm to minimize a chi-squared function, I am able to solve for relative brightness on the surface of the star. This code is meant to be easily extensible to any Kepler object. Planetary transits in the Kepler data allow for higher resolution of this brightness map along the path of the transit.

1. INTRODUCTION

2. DESCRIPTION OF THE MODEL

Overall characterization of the necessary pieces of information. This includes terminology, geometry of a star in the general case, input parameters, model flux, limb darkening, and a description of the brightness values for which we are solving.

2.1. *Set-up and Terminology*

This section should give an overview of the three types of regions in our stellar brightness map: boxes, stripes, and longitudes. It should also describe why each of these types occupies the region of the star that it does.

2.2. *Describe the inputs to the program*

This section should describe the stellar input parameters, planetary input parameters, and the model input parameters. This will be a relatively short and straightforward section.

2.3. *Model flux*

This section should describe the general case for $f_{mod,i}$. It should also talk about the specific variables, $V_{i,j}$ and b_j and motivate the reason for the visibility calculations below. It should describe the timescale for the evolution of starspots and our decision to include "windowing" in our model. And finally, it should give a brief description of the Simplex algorithm.

3. VISIBILITY AS A FUNCTION OF TIME

This section should discuss all of the math involved in the visibility profile of a general case star.

3.1. *Region Visibilities*

This should outline the rationale behind the integral used to calculate the projected surface area of each region. It should show the integral once and then show that for a box and a longitude, the θ limits are the only thing that change. It should also discuss the fact that the stripe visibilities are calculated within the Amoeba Algorithm and why (or I should change it in the code).

3.2. *Transiting Planet Model*

This section should discuss the idea that the planet is nothing more than a blocking of visibility while it is occluding the star. It should explain the basic geometry of this calculation and show the ten different cases involved.

3.3. *Limb Darkening*

The final visibility section will outline the limb darkening. Important facts to include are that we use a quadratic limb darkening equation, we include limb darkening both in and out-of-transit, and the specific math behind our calculations for each region (yay for analytically solving things).

Note

Everything is done as a function of time over the orbit.

4. VALIDATING OUR MODEL

Now that we have properly defined how our model works theoretically, we need to prove that it does work in practice

4.1. *What is variable?*

Give a description of the parameters that we still can mess with. For example, the number of stripes, number of boxes, and binning cadences both in and out-of-transit.

4.2. *Making the model light curves*

Describes the process of generating a model light curve and how, in our case, it was intrinsically linked to the stellar and planetary parameters of Kepler 17. It should briefly mention that I use a python script to generate Gaussian noise based on Kepler magnitude noise. It should also outline the list of spot models that we tried and our motivation for doing so.

4.3. *Testing Complexity*

We have only tested the ability of our program to solve for brightness values in two parameter spaces, complexity of the spots and noise of the light curve.

4.4. *Results*

Show plots (RMS, 2d boxes over time, stripes, etc.)

5. APPLICATION OF OUR PROGRAM TO REAL DATA

Kepler 17

6. POSSIBLE FIGURES

1. Brightness maps
2. Model lightcurve fits
3. Stripes over time
4. Boxes over time
5. Errors and residuals
6. Unbinned vs. binned lightcurves
7. . . . ?