

Eclipse Mapping Code

Woody Austin

Received _____; accepted _____

– 2 –

ABSTRACT

pass

1. Introduction

Magnetic fields can be directly detected and measured on a star through observations of the Zeeman effect. When an atom or molecule is in the presence of a magnetic field, once degenerate energy levels undergo a small separation. Advanced spectroscopic techniques have been developed to measure the Zeeman effect on stars so that the total magnetic field strength and/or the large-scale field topology have been directly observed for a reasonable number of bright stars with a range of masses and rotation periods (e.g. ??????). The results of these surveys suggest that a significant fraction of the total magnetic field in low mass stars is contained in small-scale, local features that are spatially unresolved (?). Thus, in order to understand the global magnetic properties of cool stars, we must be able to observe and characterize the small scale magnetic features that are similar in size to the starspots and faculae observed on the Sun ($\sim 10^4$ km) magnetic fields.

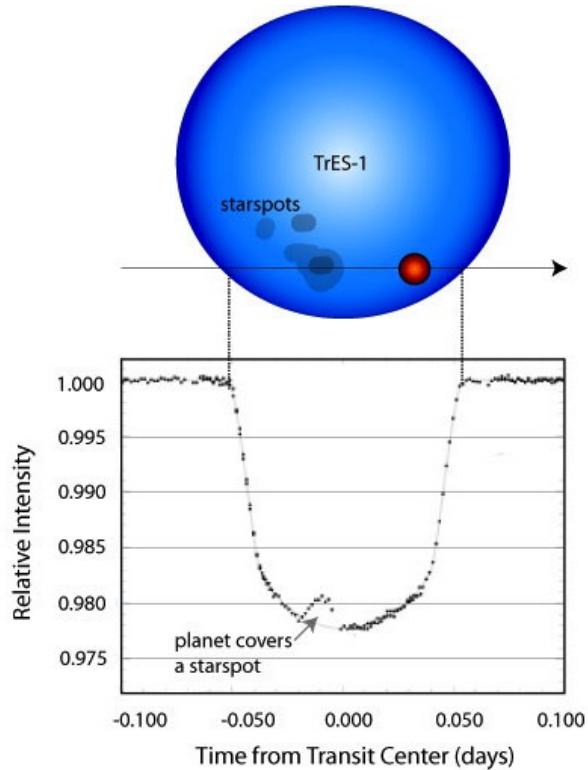


Fig. 1.— Hubble Space Telescope light curve of the transiting planet host star, Tres-1, showing an increase in brightness during the transit at phase = -0.02. The feature is most likely due to the planet crossing in front of a starspot on the stellar photosphere. Modeling of the feature by ? found the spot to have a diameter of 4.2×10^4 km with a temperature of 560 K below the photospheric temperature of the star.

This has not been possible up until now. Small scale fields are currently undetected in Zeeman splitting and polarization measurements because adjacent magnetic regions of opposite polarity produce spectral line profile signatures of opposite sign which cancel each

other out when the full stellar disk is observed. In addition, very small starspots produce virtually no spectral profile signatures, and thus are not easily detected in the surface brightness maps derived from Doppler Imaging alone. Furthermore, the effect of small starspots on the total brightness of the star is extremely small compared to the typical precision of ground based time series photometry which is limited to ≥ 1 mmag. Therefore, traditional spot modeling of the rotational variability observed in many low mass stars has only been able to identify large asymmetric starspots which cover significant fractions ($\sim 1/3$) of the stellar surface. However, with the recent availability of continuous, extremely high precision light curves from *Kepler* of many transiting planet host stars (Borucki et al. 2011; ?), we can now observe local magnetic field features on stars other than the Sun by detecting and characterizing small-scale starspots. Since Sunspots are known to be the location of strong concentrations of magnetic fields on the Sun, individual starspots can be used as tracers of the underlying local magnetic fields on low mass stars.

An apparent increase in brightness that appears as a “bump” in the light curve during a planetary transit is the signature of a planet crossing in front of a small starspot that is cooler and darker than the surrounding stellar surface (see Figure ??) A planet with radius, $R_p = 4 R_\oplus$, transiting a Solar-like star with $R_s = 1.0 R_\odot$ and uniform surface brightness would exhibit a 1.3 mmag periodic dip in brightness due to the transit. However, if the planet instead crossed directly in front of a dark starspot with a radius of 10^4 km, the transit depth would only be 1.1 mmag. This would manifest itself in the light curve as a “bump” in the flat-bottomed part of the transit with an amplitude of 0.2 mmag. Such anomalous brightness variations are detectable in high precision photometry with submillimag precision and have now been observed on some known host stars (e.g. CoRot-2, Tres-1, HAT-P-11, GJ 1214, Kepler-17). This technique is similar to “eclipse mapping”, where the distribution of starspots on the occulted surface of an eclipsing binary component is reconstructed from the flux variations observed during eclipse (??).

The starspot lifetime is an observable property that is defined by the rate of magnetic flux diffusion – one of the fundamental mechanisms that regulates the flux transport once it emerges on the surface. This parameter has been measured to high precision in the Sun (e.g. ?), but **no** measurements currently exist for other stars. Thus, the majority of stellar dynamo and magnetic flux transport models adopt the Solar value, attempting to model stars with fundamentally different properties (i.e. mass, dynamo type, convection zone depth). By measuring the lifetime of starspots for stars with a range of masses and rotation rates, we will constrain this important property as a function of convection zone depth and empirically determine the effect of rotation on the flux diffusion. In addition, by modeling the duration and amplitude of the observed photometric features as due to opaque spots, the spot sizes and temperatures can be estimated for individual features (e.g. ?) and the total spot filling factor can be derived.

2. The Model

The eclipse mapping program described in this paper derives a model for the relative two-dimensional surface brightness of a transiting planet host star that matches an observed one-dimensional light curve of its integrated flux. We derive the brightness values for a set of discrete regions on the star as shown in Figure 2 (Huber et al. 2009). Each iteration of the program produces a static map in which brightness values less than one correspond to regions of the star that contain starspots. By applying the code to a series of short segments of the light curve, we are able to extract information about the time evolution of the star’s surface brightness as described in Huber et al. (2010).

2.1. Set-up and Terminology

In order to efficiently describe the details of our program, the geometry of the problem must be defined and some basic terminology must be established. The star is assumed to be a uniformly rotating, solid body with a single planet on a nearly circular orbit that periodically transits the star. The stellar surface is divided into a series of large regions, as shown in Figure 2. While describing our program in this paper, the regions along the line of transit will be called *boxes*; the total longitudinal regions will be called *longitudes*; and the longitudinal regions with the *box* areas subtracted will be referred to as *stripes*. When talked about as an ensemble or when the distinction is unimportant, these regions will be referred to as just that - *regions*. The resolution of our brightness map is set by two parameters, the number of *boxes*, n_b , and the number of *stripes*, n_s . All of the *stripes* are the same size, and likewise so are the *boxes*. The width of each *stripe* (or *box*) is defined so that n_s (or n_b) regions exactly covers the full 360° stellar surface. The density of *boxes* is higher than that of the *stripes* because there is more information contained in the transit portions of the light curve. The number of *boxes*, n_b , is typically set to be an integer multiple of the number of *stripes* so that the boundaries of each longitude will always contain an integer number of *boxes*.

The program takes a single-band, high-cadence light curve of a transiting planet host star along with its rotation period (P_{rot}) as inputs. Deriving a realistic brightness map also depends on including a precise model of the transit, thus certain physical properties of the star and the planet must be provided. The following parameters are used to produce the detailed transiting planet model: orbital period (P_{orb}), orbital separation (a), impact parameter (b), relative radius of the planet compared to the star ($\frac{R_p}{R_*}$), and two quadratic limb-darkening coefficients. Finally, the configuration of our *regions* requires that the spin-axis of the star is aligned with the orbital axis of the planet. ***** Ask Leslie for a line (Many stars aligned citation) ***** These criteria are both met for our test object, Kepler-17 (Borucki et al. 2011).

In order to derive a discrete, two-dimensional brightness map that best reproduces the input light curve, the program calculates a model light curve by integrating over the surface of the visible sphere at each timestep according to:

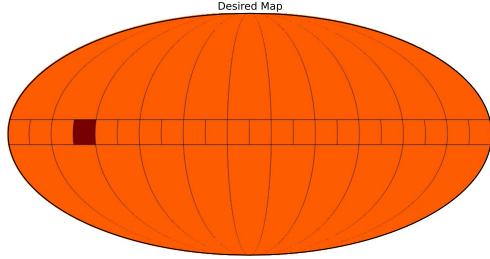


Fig. 2.— Recovered surface brightness map from our eclipse mapping program when using a simulated light curve with one dark spot group placed in the path of the planet. The figure shows the geometric set-up of a star that we are modeling. The *regions* approximately along the equator are in the path of the planet and are called *boxes*. Those out of the path of the planet are called *stripes*. In this case, the surface of the star is divided into 11 *stripes* and 22 *boxes*. Each stripe is 32.72° and each box is 16.36° . In this simulation, we recover relative brightness values of approximately one everywhere except for the visibly darkened *box* which corresponds to a brightness value of 0.90.

$$f_{mod,i} = \sum_j V_{i,j} b_j \quad (1)$$

The model flux includes b_j , the brightness per unit area for *region* j and $V_{i,j}$, the *visibility* of *region* j at time, i . As the star rotates, the brightness of a *region* does not change but its contribution to the total flux does. The amount of flux from each *region* that reaches the observer varies predictably and is captured in its visibility. Here, visibility is a measure of projected surface area along the line of sight to the observer. The transit by the planet also modifies the amount of flux that reaches the observer. Therefore, the area of the star blocked by the planet is explicitly included in the visibility calculations. The transit is accounted for by subtracting the area that the planet obscures of a given *box* from the normal visibility of the same *box* at each time step. It is important to include limb-darkening, which has a significant effect on the shape of the transit. We incorporate the quadratic limb-darkening law provided in Claret (2004).

2.2. Visibility Calculations of Regions

The visibility of each *region* is defined in Equation 2 to be the projected area along the line-of-sight modified by the limb-darkened intensity. This quantity varies as the star

rotates. For example, flux from a *region* in the middle of the backside of the star will contribute nothing to the overall luminosity at that time. Half a rotation period later, that same *region* will be in the center of the front of the star and contribute a maximal amount of flux.

$$V_{i,j} = \int_{\phi_1}^{\phi_2} \int_{\theta_1}^{\theta_2} \frac{I(\theta, \phi)}{I(0)} \sin^2 \theta \cos \phi d\theta d\phi \quad (2)$$

The visibility equation is simply the integral of the dot product of the spherical surface element, dA and the unit vector, \hat{x} , along the line-of-sight to the observer multiplied by the quadratic limb-darkening function provided in Claret (2004). ϕ_1 , ϕ_2 , θ_1 , and θ_2 are in standard spherical coordinates and denote the angular limits of an individual *region*. For any type of *region* (*box*, *stripe*, or *longitude*), the limits change, but the equation remains the same. We solve this integral analytically and then substitute in the appropriate limits of integration.

To calculate the *longitude* visibilities, ϕ_1 and ϕ_2 are determined by the number of stripes, n_s , defined at the start of the program and then modified at each timestep based on the rotational phase of the star. The latitude limits, θ_1 and θ_2 are always 0 and π . For the *box* visibilities, the ϕ_1 and ϕ_2 limits are determined in the same way using the number of *boxes*, n_b . The latitude limits depend on the impact parameter and the radius of the planet such that $\theta_{1,2} = \cos^{-1}(b \pm R_p)$. The *stripe* visibilities are calculated by subtracting the sum of the *box* visibilities in a given longitude range from the *longitude* visibility in the same range. Simplifying the calculation of the *stripe* visibilities is the primary reason for choosing the number of *boxes* to be an integer multiple of the number of *stripes*. The sum of all *stripe* and *box* visibilities at any timestep should be equal to 1.0 by definition except during a transit.

When the planet passes in front of the star, the apparent brightness of the system diminishes by an amount related to the area of the planet. To incorporate this effect in the model flux, we modify the visibilities of the *boxes* that are blocked by the planet during a transit. At each timestep, we use the orbital properties of the planet and the rotational phase of the star to determine which *boxes* are blocked, either fully or partially, by the planet. Then we calculate the intersection of the planet with each of those *boxes* on the projected surface of the star. The resulting value is multiplied by the average limb-darkened intensity in the *box* and subtracted from its unocculted visibility. We approximate the occulted *box* by a cartesian rectangle intersecting a circle and use standard Euclidean geometry to determine the area of the sector that is contained within the *box*. To verify that the simplifying approximation that each of the *boxes* is a cartesian rectangle on the surface of the star is reasonable, we show a plot of our transit model over the Mandel and Agol (2002) model generated from the same physical planet properties. The difference between the two models is less than <*****%*. A typical *box* visibility curve is shown in Figure 4. In addition, we provide additional details about the visibility calculations in Appendix A.

*****Make this eps

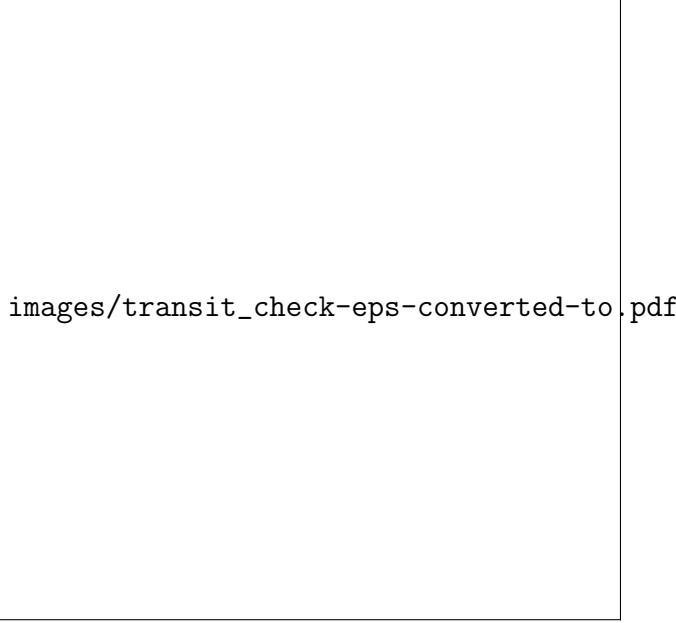


Fig. 3.— Our model transit is shown on top while the Mandel & Agol model is shown on the bottom. The difference between the two models is less than x%.

2.3. Solving for the brightness values

The goal of the program is to identify a realistic brightness distribution of the stellar surface. We employ a standard chi-squared minimization routine, the Amoeba Algorithm, to find the optimum brightness values, b_j , that, when multiplied by the visibilities as described above, produces a model light curve according to that matches the observed data. This model light curve is produced according to Equation ???. The Amoeba Algorithm is a minimization tool that works well for small numbers of dimensions (regions in this case) and which will always converge to some minima (although not necessarily absolute) (Press et al. 1992). Given an initial set of brightness values given pre-defined visibilities as described in section 2.2. Both of these, brightness values and visibilities, apply to a set of $j = n_{boxes} + n_{stripes}$ regions. The planet will occlude only the boxes and will do so only during a transit. The longitudes' brightness values are defined as:

$$\chi^2 = \sum \frac{(f_{mod,i} - f_{obs,i})^2}{\sigma_i^2} \quad (3)$$

For a given set of data the brightness values cannot change using our technique. To remedy this, each month (or quarter) of data is divided into smaller intervals called *windows*. By moving the start and end times of the windows by small amounts, information about the evolution of starspots can be inferred. Previous estimates on the timescale of starspot evolution report that it occurs on the order of 10-30 days *****cite someone*****. This works well with the model described here. The model does a better job of fitting smaller sections of data than larger ones. *****Note about striping*****

Fig. 4.— An example visibility curve for one *box*. This is "box 0", or the box whose left edge is on the left limb of the star at time 0 (the given epoch). This was generated for a system with the same parameters as Kepler-17, later shown in Section 3. θ_1 and θ_2 are 81.52° and 96.4° respectively.

The Amoeba Algorithm is used to do the actual minimization. The algorithm runs quickly and will always converge given any set of inputs. The largest downside (of most optimization algorithms) is that it cannot distinguish between local and absolute minima. This becomes especially apparent when choosing an initial set of brightness guesses for the simplex structure. Because the average brightness of the star is 1.0, the initial conditions for the simplex structure are $n_{sb} + 1$ brightness sets of n_{sb} values where $n_{sb} = n_{stripes} + n_{boxes}$ and every element is in the simplex is 1.0. A different (orthogonal) basis vector e_i times a scale factor, s , is then added to each of the $n_{sb} + 1$ sets. When the scale factor is too large, the Amoeba converges on answers that are not likely real. When the scale factor is too small or non-existent, the Amoeba is more likely to vary all regions in order to converge to a solution rather than just the regions that differ from the average value.

$$S_j = Z_j(1 - c) + \frac{c}{q} \sum_{j=1}^q B_j \quad (4)$$

Where q is the ratio of n_{boxes} to $n_{stripes}$ and c is the ratio of the total eclipsed area to the non-eclipsed area. c can be calculated by the same set of integrals that will be used for the visibilities. The Amoeba algorithm is given the set of box and longitude visibilities $\{b_1, \dots, b_j, z_1, \dots, z_n\}$. When the planet is not transiting, the only information available is about the total longitudinal brightness information. Within the chi-squared call of the Amoeba algorithm, the box and stripe visibilities and brightnesses are calculated as a way to blend the parameters together and get information about the overall longitude. This allows the use of boxes and stripes whose sum can be thought of as the longitude values in the Amoeba while still getting information about the boxes and the longitudes independently. This is called parameter interdependence. This appears to encourage the

Amoeba Algorithm to navigate to a better local minima than if this process of creating parameter interdependence and using optimal variables were omitted.

3. Validating our model

Extracting a two dimensional brightness map by fitting a model to a one dimensional light curve can result in degenerate solutions. There is no way of knowing what the brightness values of a real system should be *a priori*. To remedy this, we produce synthetic light curves using our definition of model flux, a set of visibilities, and a known set of input brightness values. The average brightness on the surface of the star is defined to be 1.0. This is the default value for this known set of input brightness values. To simulate starspots, we darken regions by giving them a brightness value of less than 1.0. For example, if we want to simulate a group of spots in only one *box*, we would set all of the input brightness values to 1.0 except for one *box* that has the value 0.90. Once the spots are darkened, the flux is then integrated at each time step over all of the *regions*. There are currently 11 synthetic light curves with different spot configurations that we test with our system, but only a subset will be detailed in the paper.

We want our synthetically produced light curves to mimic light curves from real systems. To ensure that this happens, we adopt the stellar, planetary, and orbital properties directly from the Kepler-17 system provided in ? ***Cite the correct parameter place and shown in Table ???. These parameters are then used to create the visibilities, as described in Section ??, and determine an appropriate number of stripes and boxes for the system as described in Sections ?? and 2.3. For the Kepler-17 system and for our synthetic systems, we choose the number of *stripes* and *boxes* to be 11 and 22 respectively.

A typical spot on a real star is anywhere from 500 to 1000 degrees cooler than the rest of the stellar surface. This corresponds to about 0.67 the brightness of an average part of the star in the starspot (Walkowicz et al. 2013). The *regions* in our synthetic systems are designed to imitate a real star with at most 30% spot coverage. Therefore, the brightness

Parameter Name	Value
Rotation Period	11.89
$\frac{R_p}{R^*}$	0.129530
Limb Darkening Coefficient 1	0.4282
Limb Darkening Coefficient 2	0.2514
Orbital Period	1.485711
Orbital Epoch	352.678035
Orbital Separation	5.670
Impact Parameter	0.01800
Transit Duration	0.094775

values in our *regions* vary from 1.0 down to 0.88 in the *boxes* (higher percentage of spot coverage) and down to 0.91 in the *stripes*.

When the light curves are produced in this way, they do not have any noise. This is clearly a bad imitation of a real system. We introduce Gaussian noise with standard deviations based on the average photon error counts provided in the table at ?****Noise level table. We try to recover the brightness values for noiseless, 12th magnitude noise, and 14th magnitude noise synthetic light curves. We show examples of the same light curve with different levels of noise in Figure 5. Kepler-17 itself is a fourteenth magnitude star.

The in and out-of-transit binning cadences also mimic how we would solve for a real system. Because so many points exist out of transit and the trends from larger, possibly polar spots tend to have longer timescales, a low binning cadence is desired for out-of-transit binning. For these example solves, the out-of-transit binning cadence is 95 Kepler short-cadence time steps (58.85 seconds) per bin. The in-transit binning cadence is set at a higher frequency in order to remain sensitive to small scale variations in the light curves due to small scale starspots in the path of the planet. For these example solves, the in-transit binning cadence is set to three Kepler short-cadence time steps per bin.

Our predetermined set of brightness guesses $B = \{b_1, b_2, \dots, b_{n\text{boxes}}, b_{\text{stripes}}, \dots, b_j\}$ remains constant over then entire period of data in question (usually a month of Kepler short-cadence data). This is not a good approximation to real data as it does not include any notion of starspot evolution. However, over the short period of each *window* the evolution is not a noticeable factor, and it makes more sense to produce the lightcurves in this way.

3.1. Testing Complexity

The ability of the program to solve for brightness values has only been tested in two parameter spaces, complexity of the spots and noise of the light curve. The noise of the curve is generated by a python script based on Kepler Magnitude noises. Models were tested for a noiseless, a 12th magnitude star, and a 14th magnitude star. There is no noticeable difference between these.

The complexity of the spots is determined by the number of regions that vary from the default brightness value of 1.0. The model has been tested on 11 different spot configurations. These are shown in Table 3.1.

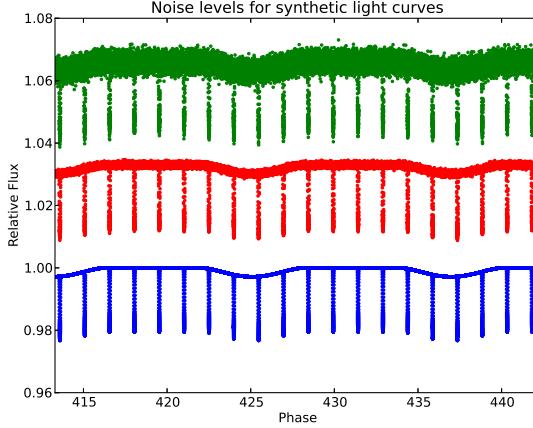


Fig. 5.— Comparison of noise of models. The noise for these models was based off of error counts similar to a given magnitude Kepler star. Our tests included models with no noise in blue, simulated 12th magnitude noise (441 counts per million) in red, and simulated 14th magnitude noise (1620 counts per million) in green (?).

Model ID	Short Name	Dark Boxes	Dark Stripes
1	1b	b4=0.90	
2	1b_1s	b16=0.90	s4=0.95
3	2b	b9=0.91 b16=0.89	
4	2b_1s	b13=0.92 b15=0.94	s6=0.92
5	2b_2s	b14=0.89 b19=0.93	s5=0.92 s6=0.96
6	3b_1s	b1=0.91 b8=0.93 b10=0.91	s4=0.86
7	3b_2s	b1=0.89 b12=0.89 b22=0.95	s6=0.96 s9=0.93
8	3b_3s	b3=0.91 b13=0.92 b21=0.90	s5=0.92 s7=0.96 s9=0.92
9	4b_3s	b4=0.91 b10=0.90 b19=0.90 b22=0.93	s2=0.94 s9=0.97 s11=0.94
10	all_b_2s	All Random	s2=0.97 s8=0.94
11	all_varied	All Random	All Random

3.2. Results

The program does a good job of recovering the input brightness values and matching the light curves. The level of noise has some effect on the reproduction of the brightness values, but clear trends can still be seen even with simulated 14th magnitude noise. One important thing to note is that the program does not recover the correct brightness values every time, but does a good job when averaged over many windows. This means that the results are better interpreted as long term trends than as exactly correct in every window. For the synthetic light curves, starspot evolution was not introduced. However, in real systems starspot evolution would exist. This means that our program can give information about overall trends in starspot evolution, but it cannot be trusted to give the exact brightness values of a region for any given time step.

Further, the algorithm needs at least one full period per window in order to accurately reproduce the box values. With higher noise levels, more of the lightcurve must be included at once. There is a tradeoff in lightcurve fit versus avoiding systematics that stem from not having enough information to recover the brightness values. It is counterintuitive to think that the brightness values are recovered more accurately overall the worse the lightcurve fit gets, but this is true. One important thing to note is that the lightcurve fit still fits extremely well overall, but there is more variation from window to window.

In all cases, the light curve fits look good for synthetic curves and real data alike. The fit of the light curve is good irrespective of noise level in the synthetic curve. The RMS of the recovered brightness values is dependent upon the noise levels, but they are still reliably recovered even with simulated 14th magnitude noise.

We have created seven diagnostic plots to compare our synthetic light curve inputs to the model outputs recovered by the code for those same inputs. The first of these plots is the input brightness map shown in Figure 6. This shows every region on the star and the relative brightness values of each region that was used to create the corresponding synthetic light curve that is then fed into the eclipse mapping code.

To visualize the brightness values, we use Figures 7 and 8. Along the x-axis is window number defined in Section ???. The y-axis shows the longitude of the star. The top of the plot starts at longitude 0, and the bottom of the plot ends at longitude 360 and then wraps around. These longitudes correspond to the region numbers. This is why we must separate the box and stripe visualizations. The stripes and boxes occupy the same longitude space. Each "pixel's" color is indicative of the brightness value for a given region during a given window. Each vertical strip shows a complete set of box or stripe brightness values for one window. Each horizontal strip shows the recovery for the same region over every window. The value on the far right, beyond the black separator, is the input brightness value for the synthetic light curves. You can compare the recovered brightness in that pixel with all of those in the same region to the left on the same horizontal slice.

An important thing to note, that should be evident from the discussion of the box and stripe brightness visualizations is that we are working with multiple windows in all cases. This makes it hard to show the brightness recovered for each individual window on the surface of a star in a compact and meaningful way. Because of this, we only show the

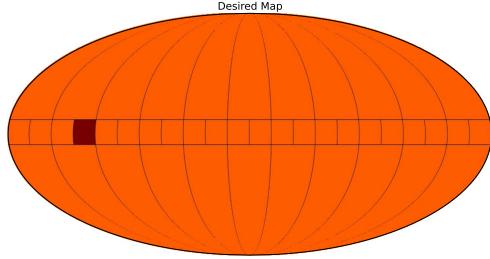


Fig. 6.— The set of brightness values used to produce a specific synthetic lightcurve. In this case, this was used to produce the 1b/ set of lightcurves (no noise, 12th magnitude noise, 14th magnitude noise).

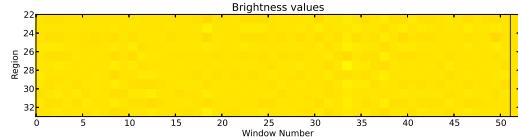


Fig. 7.— Recovered stripe brightness plot versus window number. This plot is for the no noise model with 1 box darkened. Note the desired values on the far right.

average brightness recovered over all windows and plot it on the star similar to the input brightness map. You can see this in Figure 9. For our synthetic lightcurves, this map should match the input map because there is no starspot evolution included. In a real system, a model brightness map is inadequate because evolution could occur. The average over the given time period will be correct, but it will not be reporting all of the information that has been recovered. In this situation, it would be best to produce a separate brightness map for every window or just revert to the box and stripe brightness visualizations shown in Figures 7 and 8.

The next figure is the RMS of the average brightness recovered for a given region over all windows versus the input brightness used to produce the synthetic lightcurve shown in Figure 10. The red diamonds are the regions that have been modified from the default brightness of 1.0. This is only useful to verify that we recover the values used to create

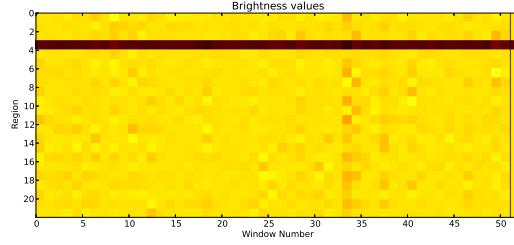


Fig. 8.— Recovered box brightness plot versus window number. This plot is for the no noise model with 1 box darkened. Note the desired values on the far right.

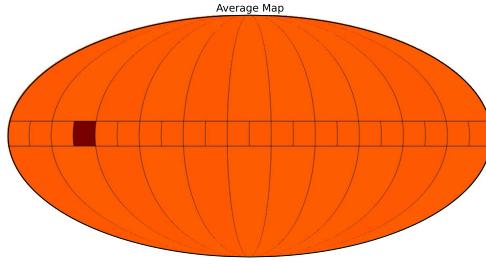


Fig. 9.— Average brightness value in each region over all windows of an eclipse-mapping code run that was attempting to recover the 1b/no_noise lightcurve.

our synthetic lightcurves. It has no use for real data because we don't know the brightness values before running the code.

The next plot is the lightcurve fit shown in Figure 11. The red points are the input, synthetic lightcurve data. The black lines are the model fits. One thing to note here is that there are actually as many black lines as there are windows. Each black line is overlaid on top of the corresponding portion of lightcurve. As you can see in all cases, it looks like one line, and the fits are incredibly good.

The final plot is that of the transit fits. Just like the overall lightcurve fit, there are multiple windows per transit that are being plotted. As the complexity grows, you can see this more and more. Each of these transit pages has the transits in order as you would read in English. We based our synthetic lightcurves off of the third month of Kepler-17 data. This particular data set has 20 transits, so all of our models have exactly 20 transits to

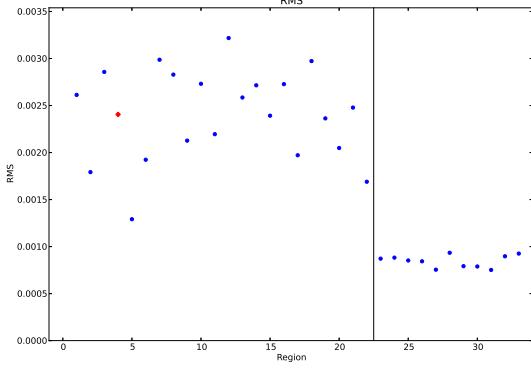


Fig. 10.— Comparison of the average value over all windows for each region versus the input value for each region. This particular plot is for the 1b/no_noise lightcurve.

recover.

A. Visibility calculations

*****Details of the vis preamble

A.1. Basic integration for visibility calculations

Recall equation 2 from Section ???. In this equation, the quadratic limb darkening law is taken directly from Claret (2004):

$$\frac{I(\mu)}{I(0)} = 1 - c_1(1 - \mu) - c_2(1 - \mu)^2 \quad (\text{A1})$$

Where $\mu = \sin \theta \cos \phi$ in spherical coordinates. Substituting equation ?? into equation 2 yields:

$$V_{i,j} = \int_{\phi_1}^{\phi_2} \int_{\theta_1}^{\theta_2} (1 - c_1 - c_2) \sin^2 \theta \cos \phi + (c_1 + 2c_2) \sin^3 \theta \cos^2 \phi - c_2 \sin^4 \theta \cos^3 \phi d\theta d\phi \quad (\text{A2})$$

We calculate this integral at every timestep for every region with ϕ_1 and ϕ_2 changing based on the region number and the rotational phase. θ_1 and θ_2 are constant in time, but different for boxes and longitudes as described in Section ??.

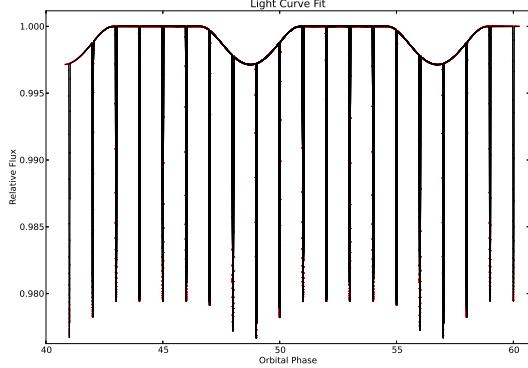


Fig. 11.— Light curve fit of the recovered models versus the synthetic lightcurve. This particular fit has 51 model fits overlaid on top of the red data and is the recovery for the 1b/no_noise lightcurve.

A.2. The transiting planet calculations

The simplest case in which the planet is entirely contained within a box is:

$$A_{occluded} = \pi r_p^2 \quad (\text{A3})$$

The planet can also be partially contained in a box in many ways, shown in Table 1. The sliver of the planet that is not within the box will be calculated and then extended to take care of all of the various cases. To do this, the area of a sector of a circle is found and then the triangle that is formed within it is subtracted. The sector is the combination of the light blue and yellow regions in Figure 12. *****Make this eps

$$A_{sector} = \frac{r_p \alpha^2}{2} \quad (\text{A4})$$

where:

$$\alpha = 2 \cos^{-1} \left(\frac{r'}{r_p} \right) \quad (\text{A5})$$

The area of the triangle is cr' . c is one half the length of the chord in the figure. c can be found with the Pythagorean theorem. Knowing this, the area of the triangle becomes:

$$A_{triangle} = r' \sqrt{r_p^2 - r'^2} \quad (\text{A6})$$

Finally, the area of the segment that indicates the part of the planet just over the border of the box is:

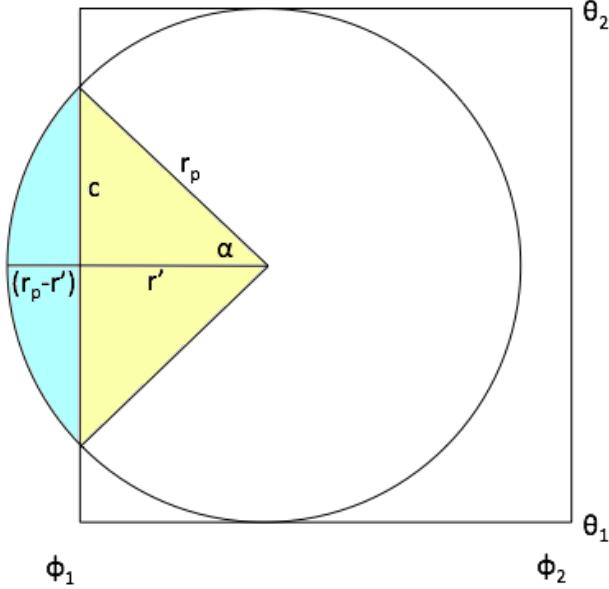


Fig. 12.— Geometry of eclipse path visibility.

$$A_{seg} = r_p^2 \cos^{-1} \left(\frac{r'}{r_p} \right) - r' \sqrt{r_p^2 - r'^2} \quad (A7)$$

After completing the calculation of the area eclipsed by the planet, we multiply it by the average limb-darkened intensity in the containing box.

B. Results Continued

Using A_{seg} and Table 1, a believable box-car transit model is produced. Note that $A_{seg,l}$ refers to a segment that is off the left side of a box relative to the center of the planet and likewise for $A_{seg,r}$. The subscripts $A_{seg,r1}$ and $A_{seg,r2}$ and likewise for the left refer to cases where there are two portions off one side of a box relative to the center of the planet. Limb darkening must also be included in order to give a better approximation to the actual shape of a real transit. This is introduced in the visibility rather than in the model flux calculation. This is okay because $f_{mod,i} = V_{i,j} b_j$. If limb-darkening were calculated during every flux calculation of every chi-squared call during the Amoeba algorithm run, it would require much more operational complexity in the code. However, limb-darkening can be calculated before the many chi-squared calls via the Amoeba algorithm and avoid this. The geometric analysis of the transit and limb darkening is continued in the next section.

The top two plots show the brightness map as it appears on a projected star. The

Table 1: Subcases for planetary eclipse visibility

Case	Description	Area covered by planet in box
I	Planet is completely out of the box to the right	0
II	Planet is completely out of the box to the left	0
III	Planet is completely contained in the box	πr_p^2
IV	Planet is partially off the right side of the box. Center is inside of the box	$\pi r_p^2 - A_{seg,l}$
V	Planet is partially off the right side of the box. Center is outside of the box	A_{seg}
VI	Planet is partially off the left side of the box. Center is inside of the box	$\pi r_p^2 - A_{seg,r}$
VII	Planet is partially off the left side of the box. Center is outside of the box	A_{seg}
VIII	Planet is partially off both sides of the box. Center is inside of the box	$\pi r_p^2 - A_{seg,r} - A_{seg,l}$
IX	Planet is partially off the right side of the box. Center is outside of the box to the left	$A_{seg,l1} - A_{seg,l2}$
X	Planet is partially off the left side of the box. Center is outside of the box to the right	$A_{seg,r1} - A_{seg,r2}$

left map shows the input brightness values to the system for the production of the light curves. The map on the right shows the average recovered brightness over 24 windows that were modeled over the 30 day data set. Each window was 14 days wide and the increment between start times of the windows was 0.3 days. The next two plots show the box and stripe brightness values as a function of window. The region number corresponds to the tuple defined in Section ??, Equation ???. The hue represents the brightness value and should be the same scale for all four of these pictures reporting brightness values. Each column in these plots is one window. The far right column (separated by a line) shows the input brightness value for that region. The next plot shows the standard deviation of the average recovered value in a region compared to the input value for that region. The boxes are to the left of the black line, and the stripes are to the right. Note that the region numbers are the same as in the box and stripe brightness plots above. The final plot is the fit to the synthetic light curve. This plot shows the entire month of data that was created. Every model is over-plotted. The model light curves are shown in black while the synthetic light curves are the red points. Note that models overlap significantly due to the fact that the start times are only 0.3 days ($\frac{1}{5}$ orbital phase units) apart per window while the window length is 14 days.

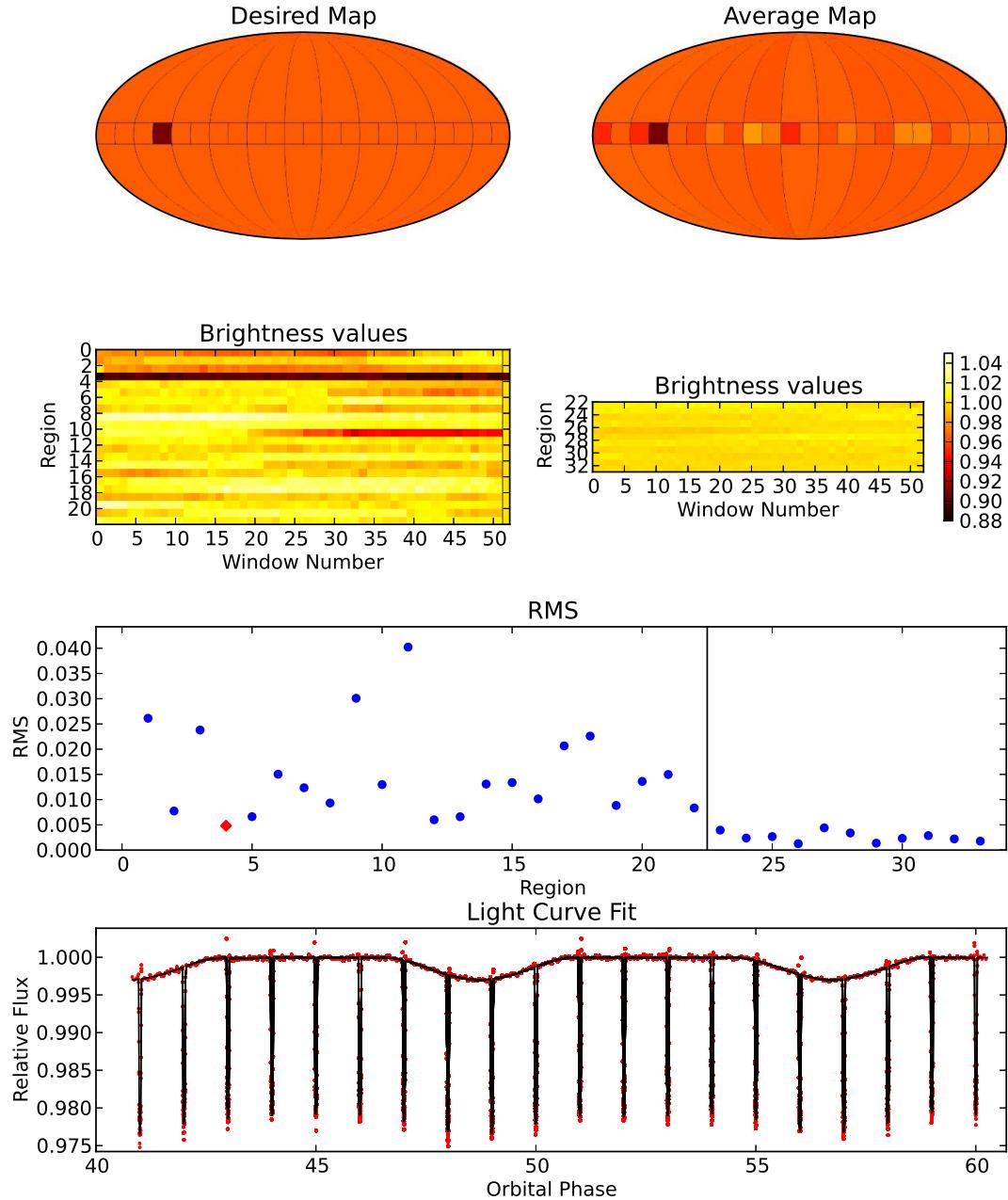


Fig. 13.— Diagnostic plots for synthetic starspot system recovery.

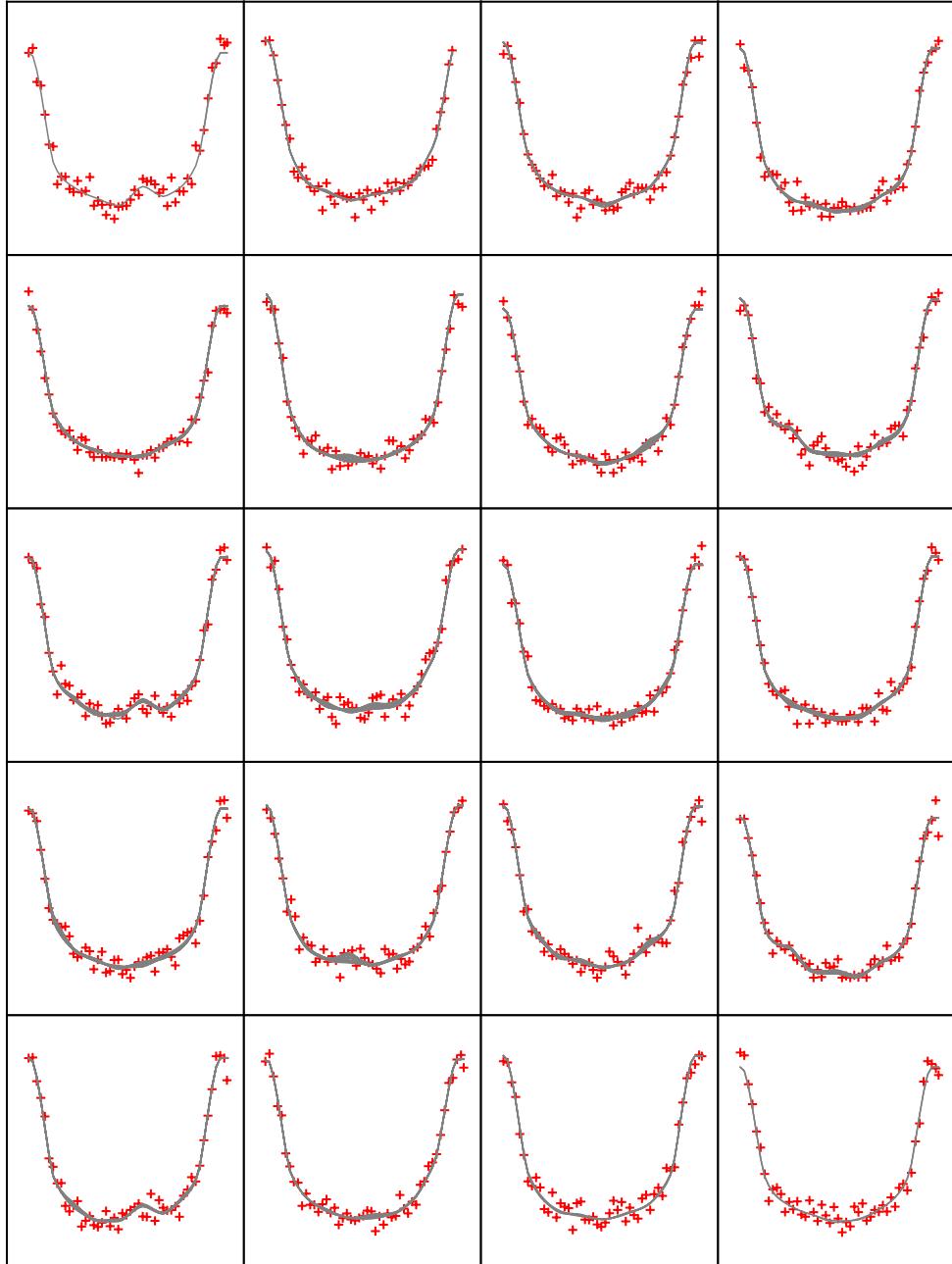


Fig. 14.— Each of the transits for the month of synthetic data. The transits are in order as English is read. Each transit plot shows the synthetic data in red points with every relevant model overlaid in grey lines.

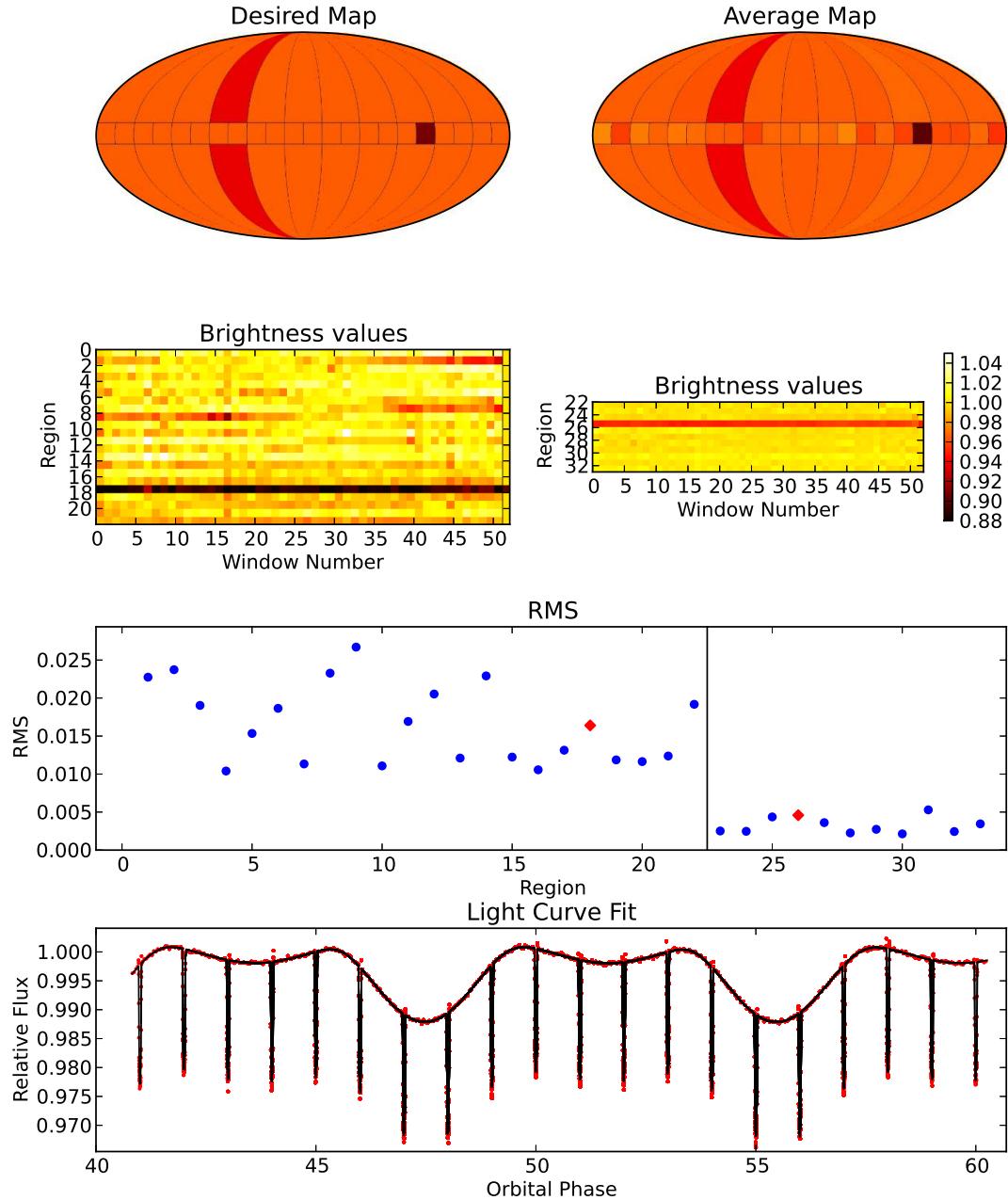


Fig. 15.— Diagnostic plots for synthetic starspot system recoveries.

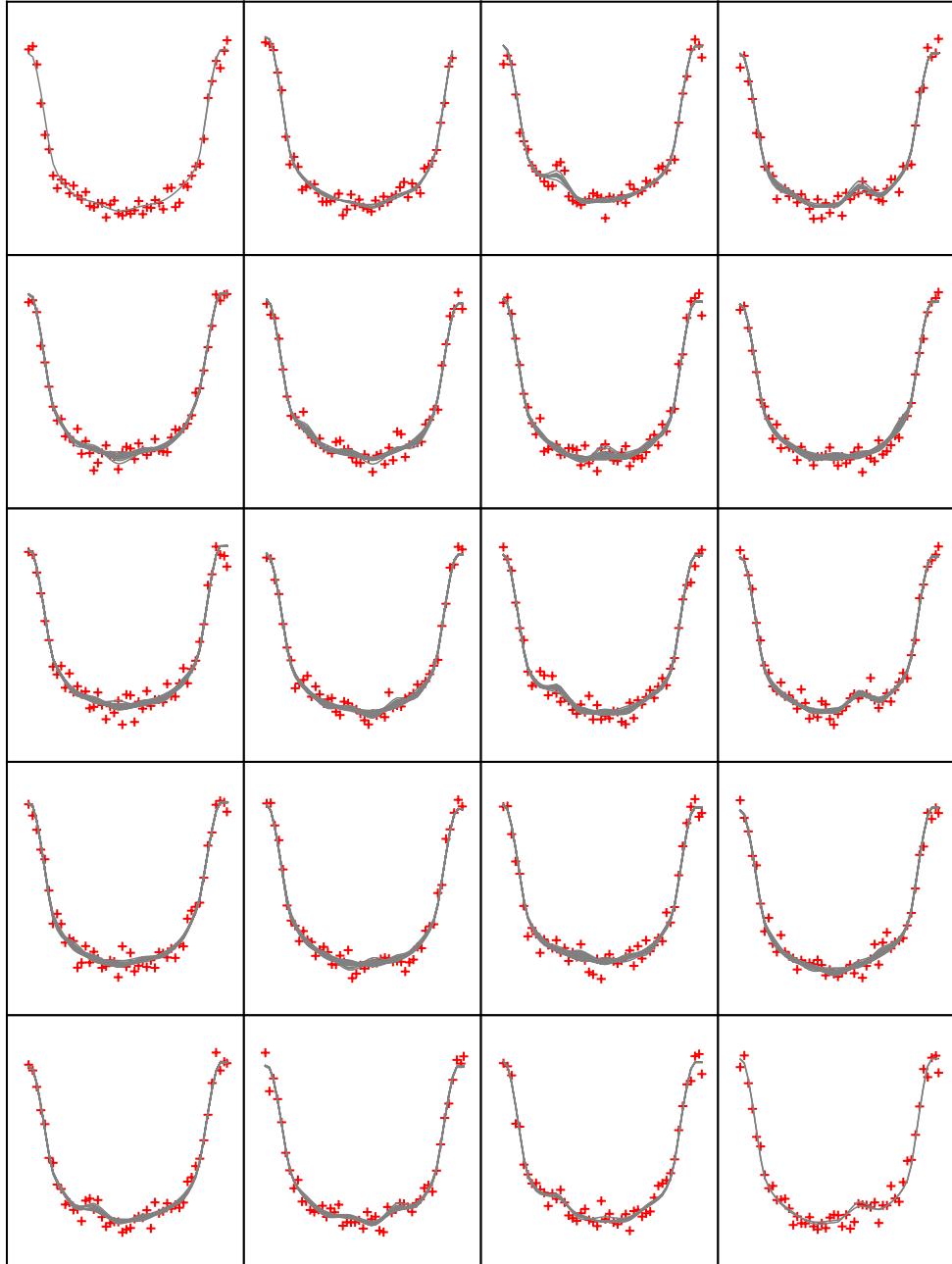


Fig. 16.— Each of the transits for the month of synthetic data. The transits are in order as English is read. Each transit plot shows the synthetic data in red points with every relevant model overlaid in grey lines.

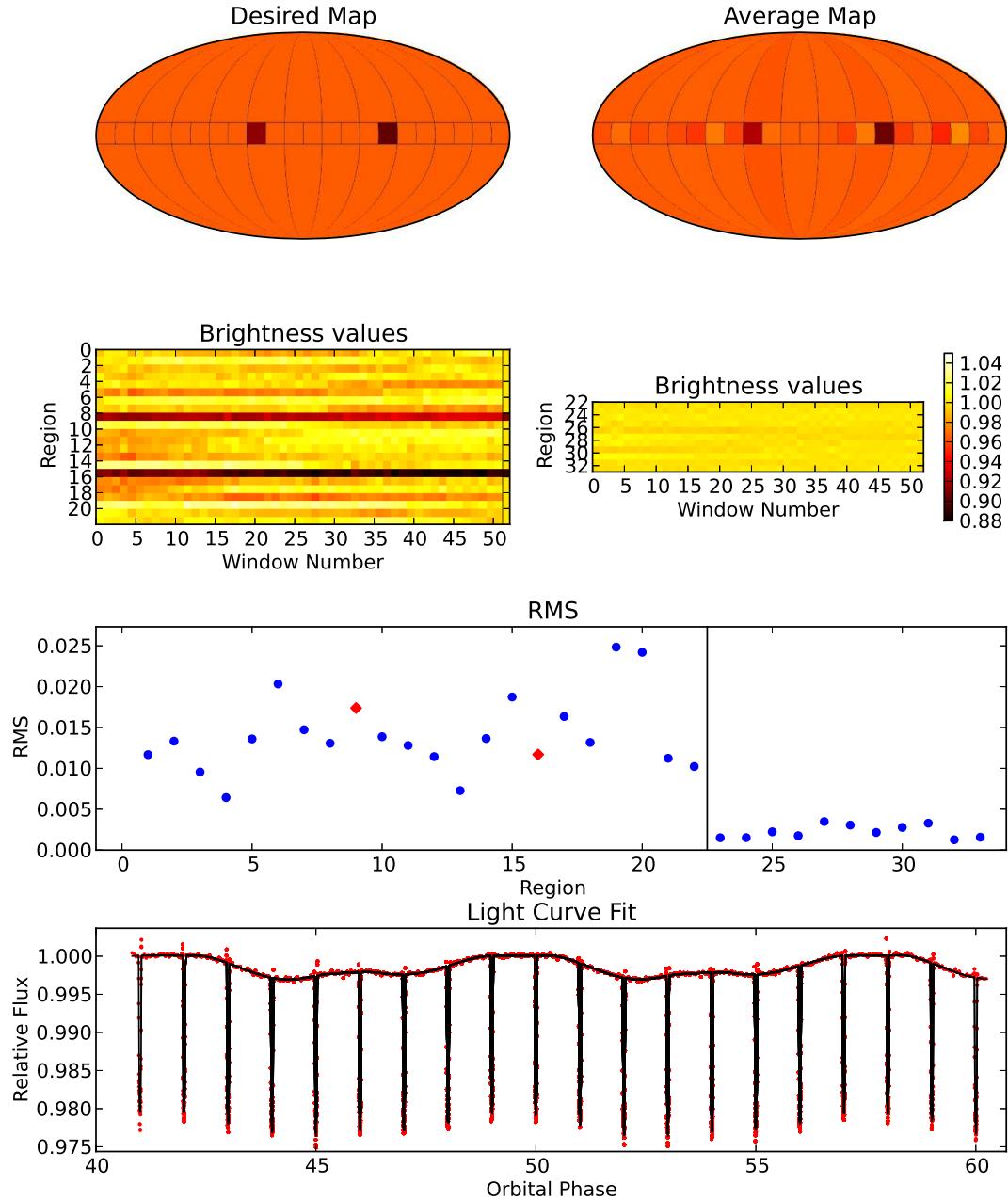


Fig. 17.— Diagnostic plots for synthetic starspot system recovery.

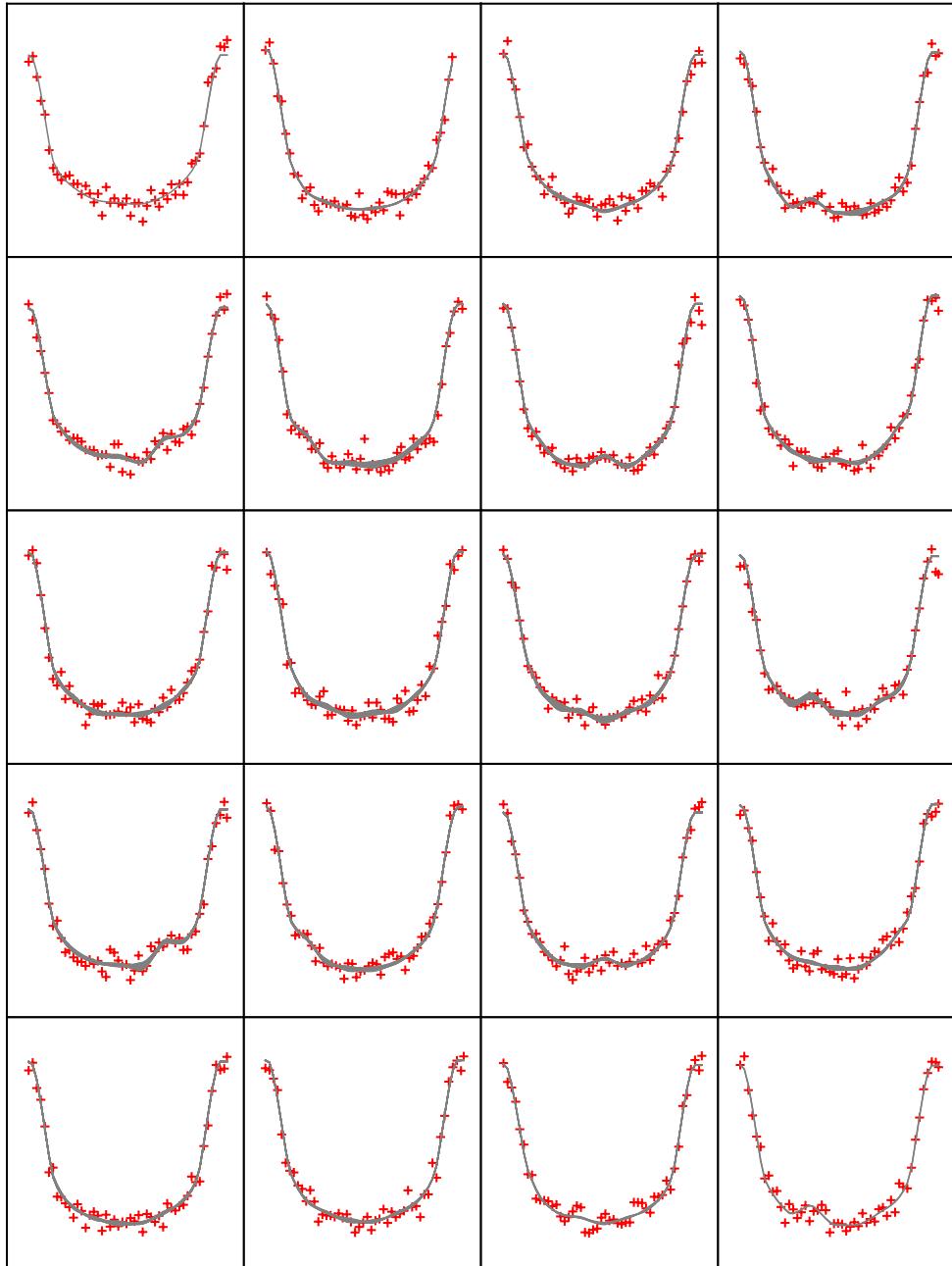


Fig. 18.— Each of the transits for the month of synthetic data. The transits are in order as English is read. Each transit plot shows the synthetic data in red points with every relevant model overlaid in grey lines.

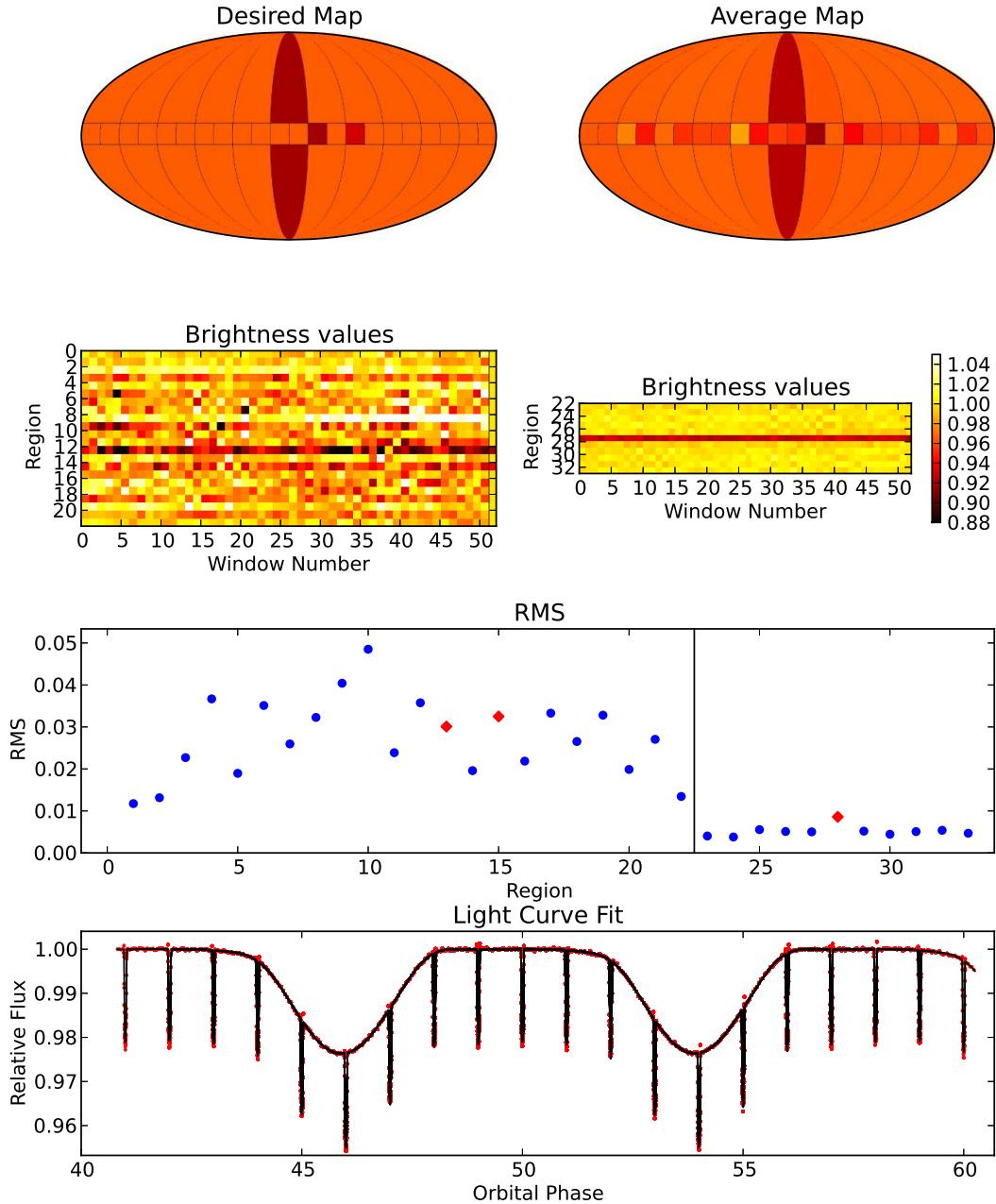


Fig. 19.— Diagnostic plots for synthetic starspot system recovery.

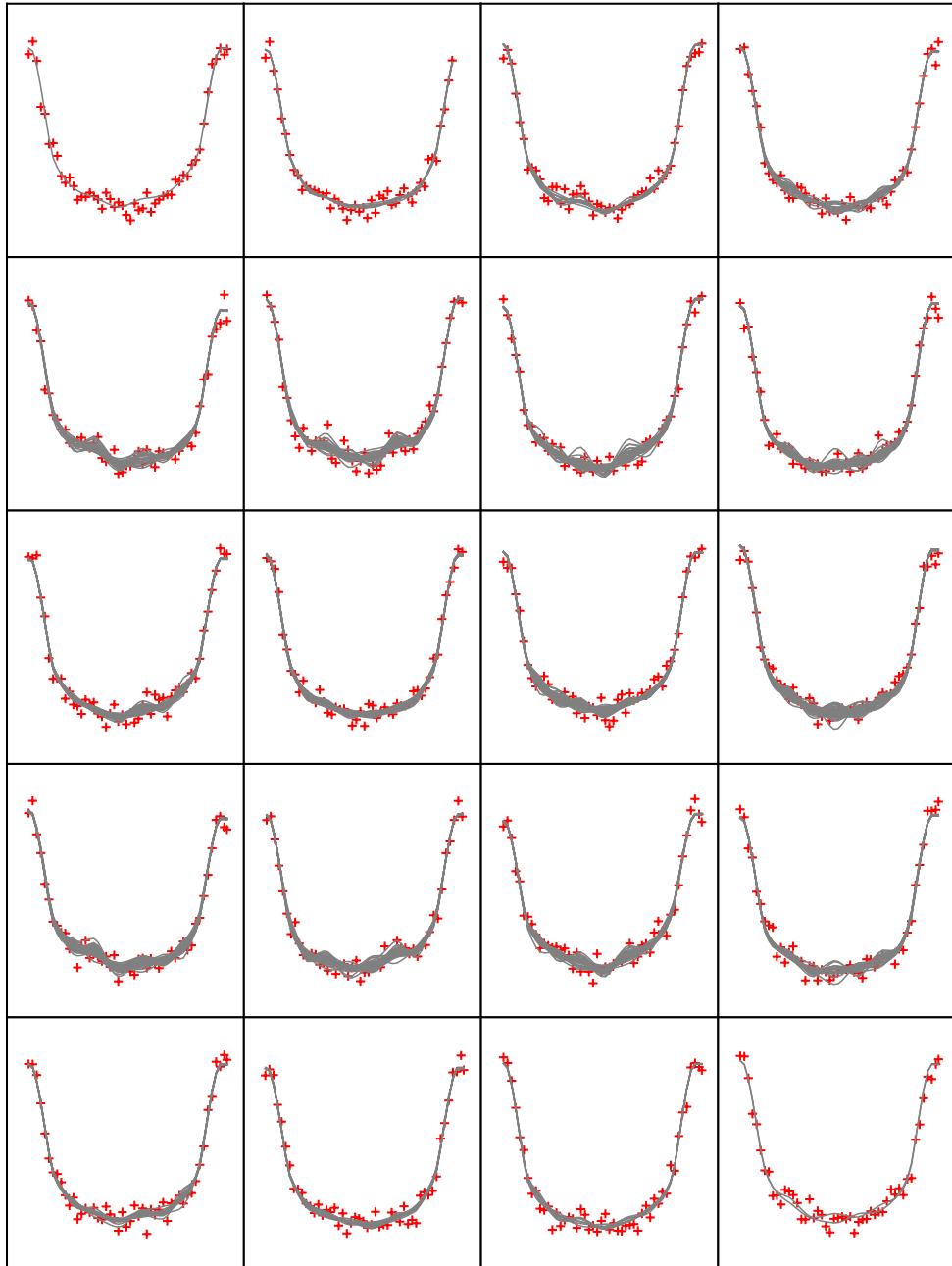


Fig. 20.— Each of the transits for the month of synthetic data. The transits are in order as English is read. Each transit plot shows the synthetic data in red points with every relevant model overlaid in grey lines.

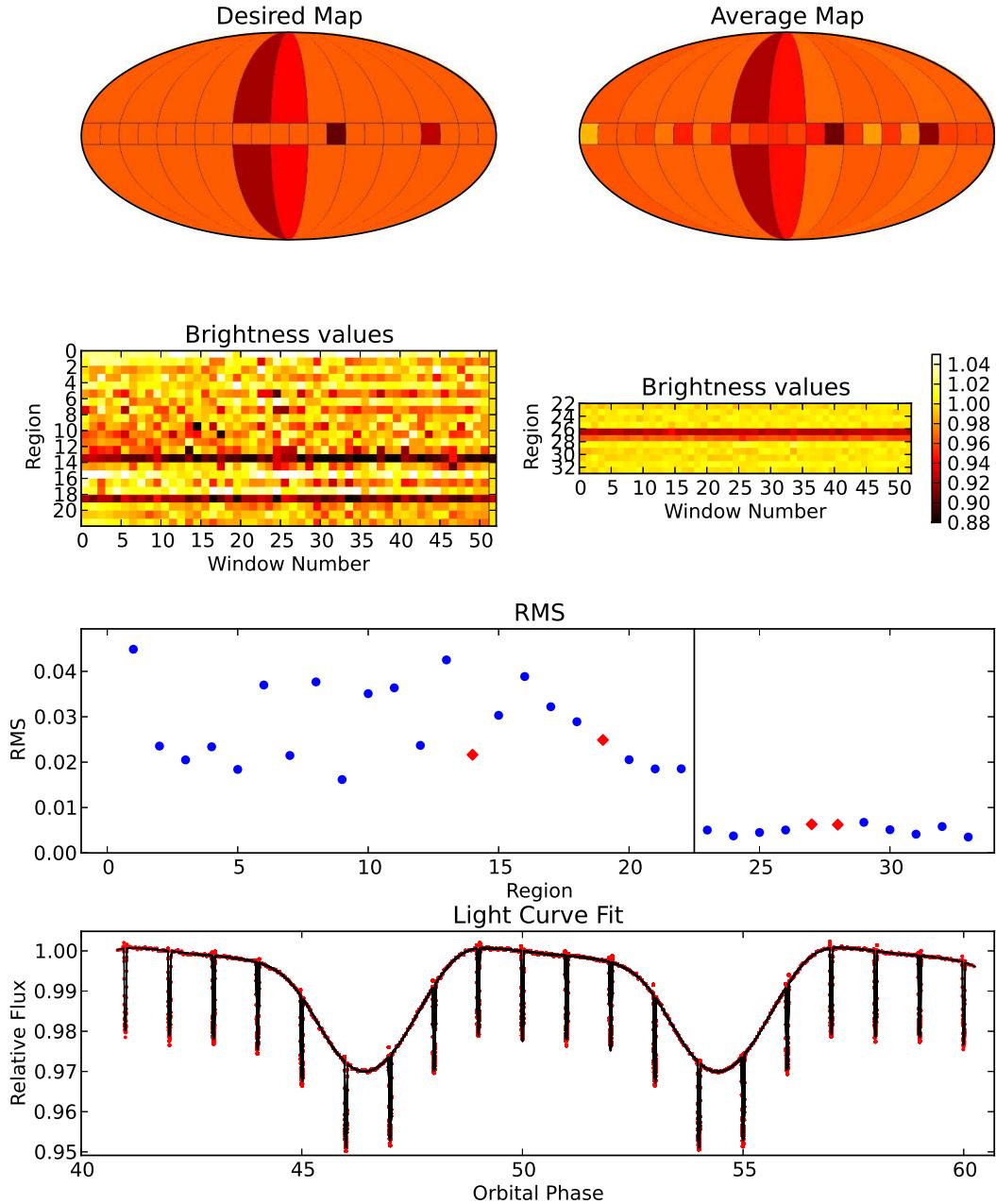


Fig. 21.— Diagnostic plots for synthetic starspot system recovery..

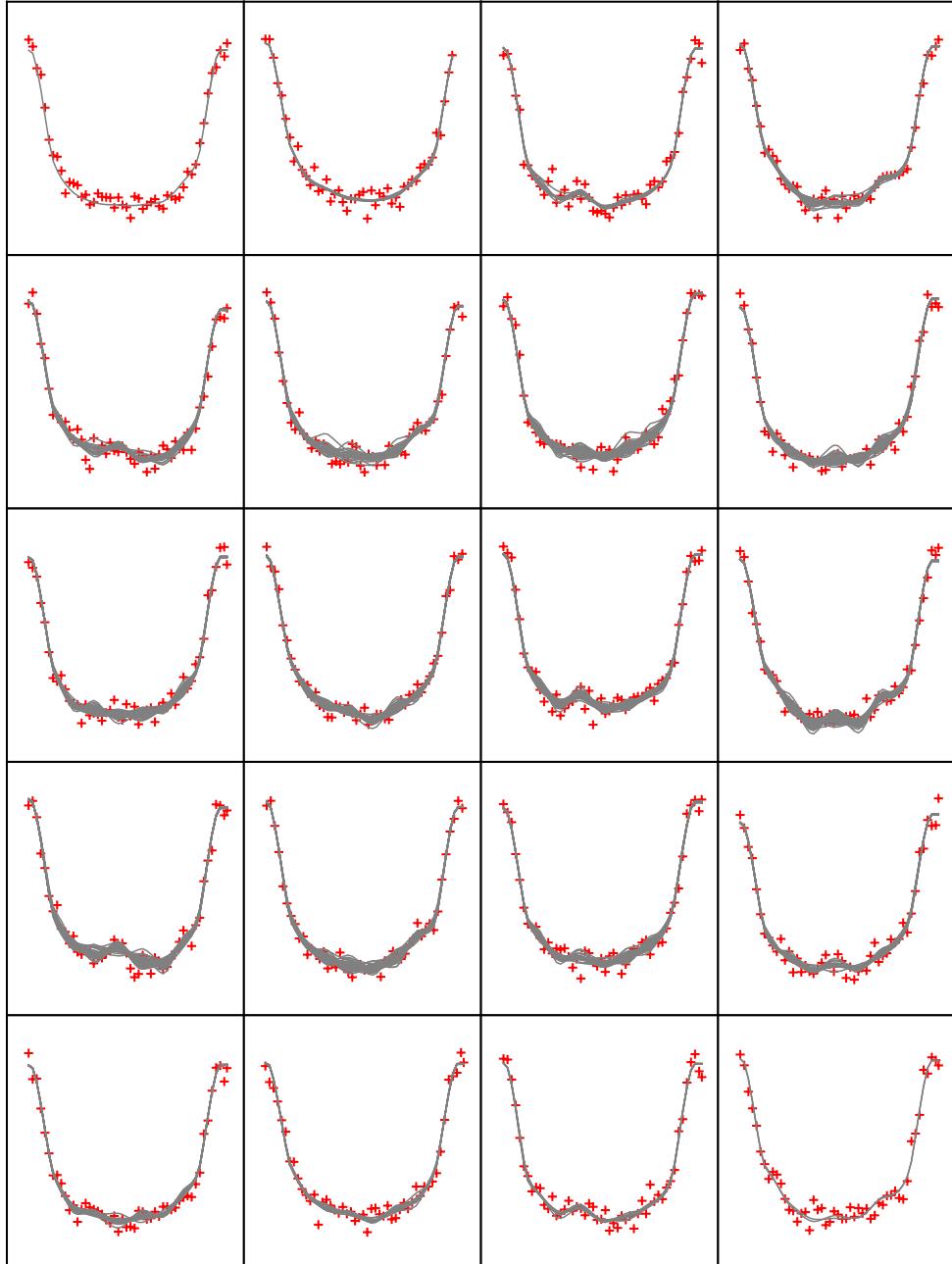


Fig. 22.— Each of the transits for the month of synthetic data. The transits are in order as English is read. Each transit plot shows the synthetic data in red points with every relevant model overlaid in grey lines.

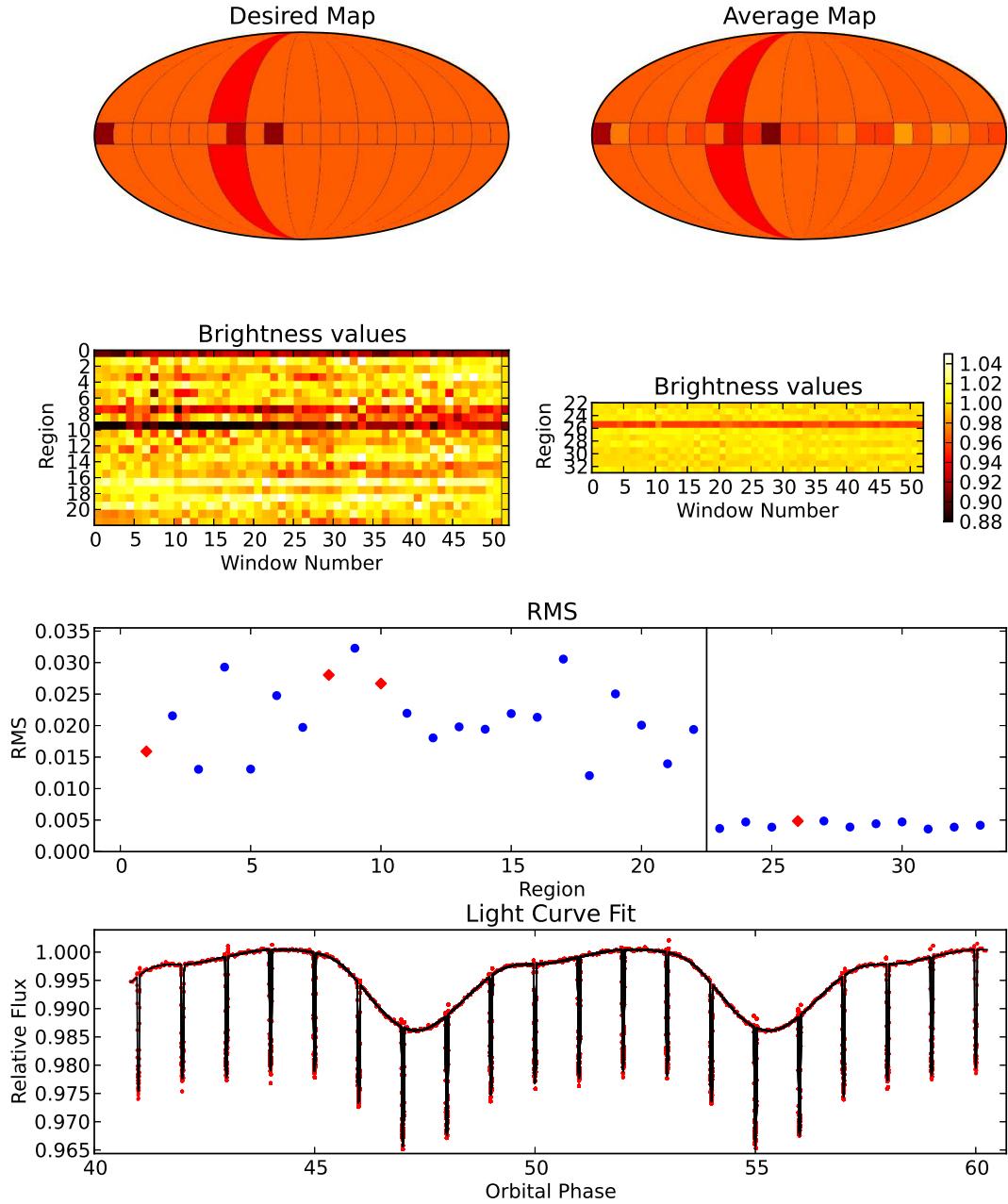


Fig. 23.— Diagnostic plots for synthetic starspot system recovery.

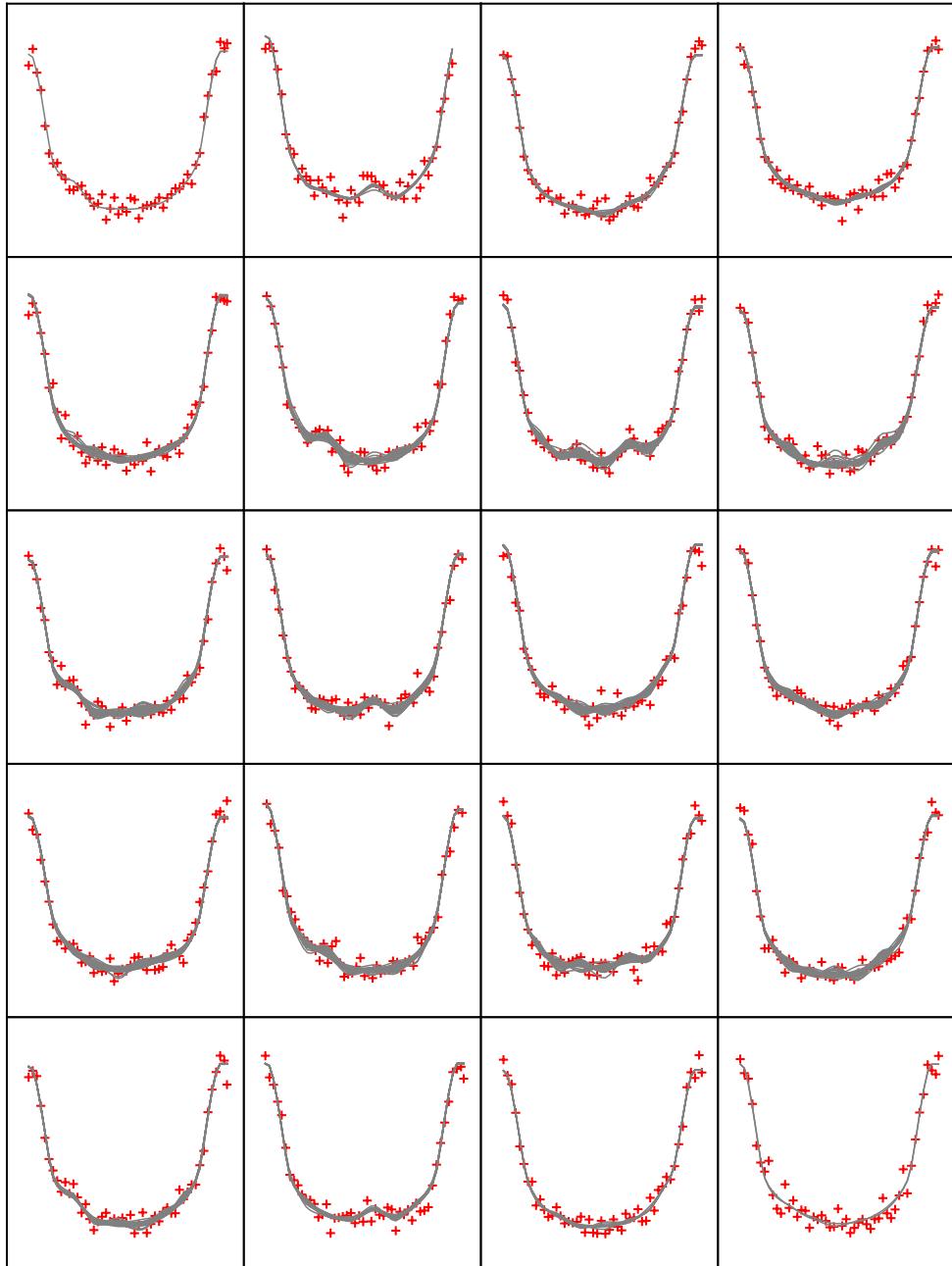


Fig. 24.— Each of the transits for the month of synthetic data. The transits are in order as English is read. Each transit plot shows the synthetic data in red points with every relevant model overlaid in grey lines.

REFERENCES

- W. J. Borucki, D. G. Koch, G. Basri, N. Batalha, T. M. Brown, S. T. Bryson, D. Caldwell, J. Christensen-Dalsgaard, W. D. Cochran, E. DeVore, E. W. Dunham, T. N. Gautier, III, J. C. Geary, R. Gilliland, A. Gould, S. B. Howell, J. M. Jenkins, D. W. Latham, J. J. Lissauer, G. W. Marcy, J. Rowe, D. Sasselov, A. Boss, D. Charbonneau, D. Ciardi, L. Doyle, A. K. Dupree, E. B. Ford, J. Fortney, M. J. Holman, S. Seager, J. H. Steffen, J. Tarter, W. F. Welsh, C. Allen, L. A. Buchhave, J. L. Christiansen, B. D. Clarke, S. Das, J.-M. Désert, M. Endl, D. Fabrycky, F. Fressin, M. Haas, E. Horch, A. Howard, H. Isaacson, H. Kjeldsen, J. Kolodziejczak, C. Kulesa, J. Li, P. W. Lucas, P. Machalek, D. McCarthy, P. MacQueen, S. Meibom, T. Miquel, A. Prsa, S. N. Quinn, E. V. Quintana, D. Ragozzine, W. Sherry, A. Shporer, P. Tenenbaum, G. Torres, J. D. Twicken, J. Van Cleve, L. Walkowicz, F. C. Witteborn, and M. Still. Characteristics of Planetary Candidates Observed by Kepler. II. Analysis of the First Four Months of Data. *ApJ*, 736:19, July 2011. doi: 10.1088/0004-637X/736/1/19.
- A. Claret. Non-linear limb-darkening law for LTE models. III. (Claret, 2004). *VizieR Online Data Catalog*, 342:81001, October 2004.
- K. F. Huber, S. Czesla, U. Wolter, and J. H. M. M. Schmitt. A planetary eclipse map of CoRoT-2a. Comprehensive lightcurve modeling combining rotational-modulation and transits. *A&A*, 508:901–907, December 2009. doi: 10.1051/0004-6361/200912867.
- K. F. Huber, S. Czesla, U. Wolter, and J. H. M. M. Schmitt. Planetary eclipse mapping of CoRoT-2a. Evolution, differential rotation, and spot migration. *A&A*, 514:A39, May 2010. doi: 10.1051/0004-6361/200913914.
- A. F. Lanza, M. Rodonò, I. Pagano, P. Barge, and A. Llebaria. Modelling the rotational modulation of the Sun as a star. *A&A*, 403:1135–1149, June 2003. doi: 10.1051/0004-6361:20030401.
- K. Mandel and E. Agol. Analytic Light Curves for Planetary Transit Searches. *ApJ*, 580: L171–L175, December 2002. doi: 10.1086/345520.
- William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992. ISBN 0-521-43108-5.
- L. M. Walkowicz, G. Basri, and J. A. Valenti. The Information Content in Analytic Spot Models of Broadband Precision Light Curves. *ApJS*, 205:17, April 2013. doi: 10.1088/0067-0049/205/2/17.