Time Series Analysis in Astrophysics

Compulsory Exercise 3

PART 1

Nicholas Emborg Jannsen
(201205962)

March 28, 2019

1 Introduction

The aim of this compulsory exercise is to use Cross-Correlation (CC) and Auto-Correlation (AC) to search for transit signals from exoplanets and construct a phase folded transit curve. The software developed in the last 2 compulsory exercises will be used when needed. The software will be used on Kepler data and the CC and AC method will be compared. The developed routines will be explained along the way as they become relevant. All routines is again developed in the computational language Python.

2 Programs and tests on Kepler-1

Just as done in the last exercise, to more easily use the developed software in this compulsory exercise I have made a new routine named `test3` where all programs are executed through. I have developed a more handy software named `plot_tools` for plotting to make the code more readable. These routines will not be presented here but again the details/the code can be found in the appendix.

As an extra development all routines in this course have been collected into a single software called `Timeseries_Tools`. When needed small code blocks will be represented, however, the combine code should be seen in the appendix. To get a better understanding of what the code in this exercise does, I will show the test of each step on the first data sample of Kepler-1 which can be seen in Fig. 1. By visually inspecting the lightcurve for transiting signal and using

$$P_{\text{transit}} = \frac{t(N) - t(0)}{N}, \tag{1}$$

where N is the total number of transits, $t(0)$ is the time of the first transit, and $t(N)$ is time of the last observed transit. By these means $P_{\text{transit}} = 4.29$ days for Kepler-1b.

2.1 Correction filters

Before explaining how the CC and AC routine can be applied on the raw data to find transit signals, one need first to correct/post-process the data by the following steps.

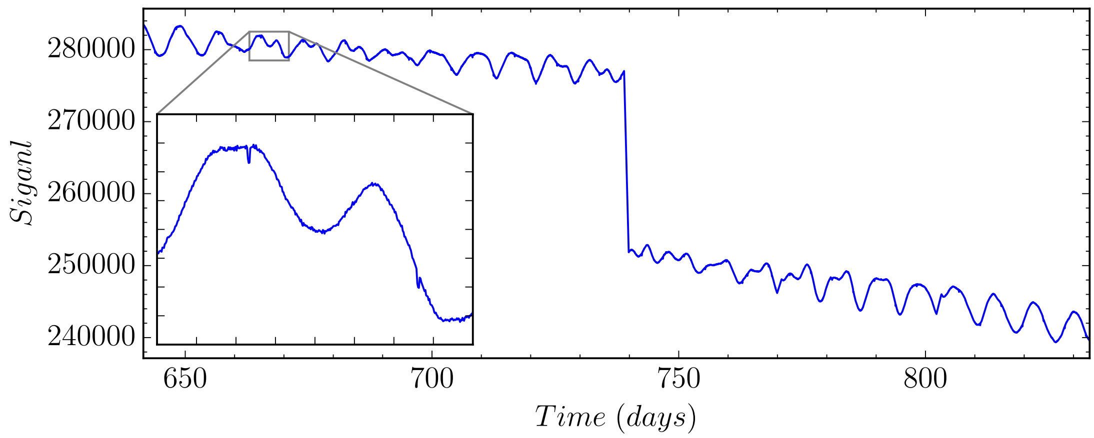


Figure 1: Time series for the 1. unknown exoplanet observed by Kepler. A zoom-in on a exoplanet transit is also visible.

2.1.1 Locate bad data

First bad data points needs to be located in the time series. This can be done by using a so-called *moving median* filter defined by

$$d_{\text{med}}(i,n) = \text{med}\left[d(i-n),\ldots,d(i+n)\right], \tag{2}$$

where d is a representation of the signal data. Usually $n = 1$ or $n = 2$ is a appropriate choice. This median filter is used to construct

$$\text{dif}(i) = \frac{d_{\text{med}}(i,n)}{d(i)} - 1. \tag{3}$$

If any data points are extreme (higher than chosen threshold), $\text{dif}(i)$ will locate these and replace them by d_{med} (in contrast to delete them). The new time series (corrected for bad data) is now called $D(i)$.

Software: `moving`

```
def moving(filtertype, S, n):
```

First I created the routine `moving` such it by construction can be used both as a moving median and a moving mean filter. The used filter is determined by the input parameter `filtertype` given by either `'median'` or `'mean'`. `S` is the data signal and `n` is the integer defined above. To avoid edge effect where the moving filters are not defined, the moving filters are calculated separately in the lower limit $d[-n,-(n-1),\ldots,n-1,n]$ and the upper limit $d[N-n,N-(n-1),\ldots,N+(n-1),N+n]$. The function returns the filtered signal.

Software: `locate`

```
def locate(data, n=None, cutoff=None, plot=None):
```

The function `locate` used the above given description of finding $\text{dif}(i)$ to locate bad data. Here `n` is the integer used in the median filter, `cutoff` is a user specified cut off value for which bad is considered, and `plot==1` calls the function `plot_locate` which visually display the outliers. For Kepler-1 this plot for $n = 1$ and $n = 2$ can be seen in Fig. 2. It is evident from these plots that $n = 2$ produce a significantly higher number of outliers for the same cut off limit (here 3×10^{-4}). As the outliers for $n = 2$ seems to be periodic/equidistant they may be due to effects of the transits, thus, $n = 1$ was used here.

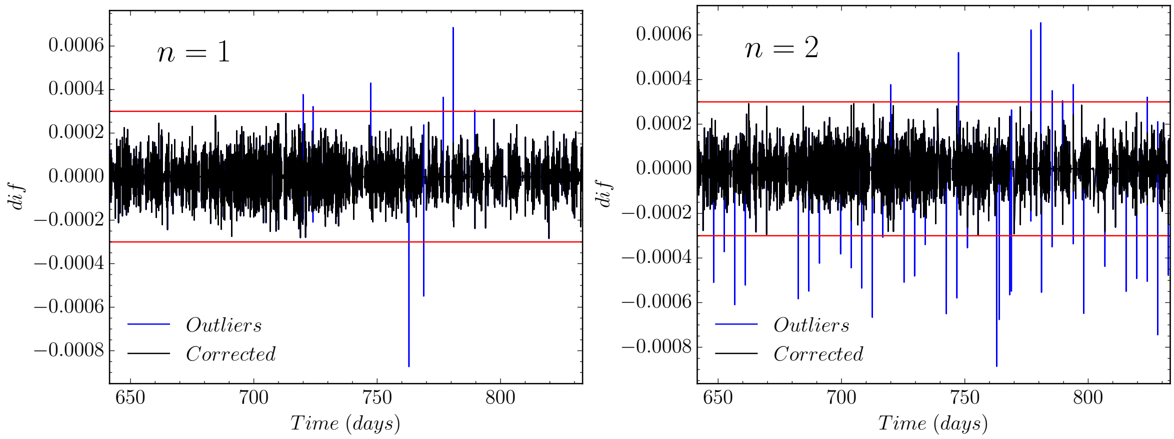


Figure 2: Bad data is found by a dif vs. time diagram as shown here for Kepler-1. Left and right plot is a dif calculated with $n = 1$ and $n = 2$, respectively. The red line (—) is the cut of value that distinguish between god (—) and bad (—) data.

2.1.2 Jumps

Software: `jumps`

```
def jumps(data, gapsize, plot=None):
```

If large jumps are present the data the next step is to correct for this. The computational way is first to find the difference (or more visually the distance) between each data point in the time series. A index of a jump is then registered for when this difference-value is higher than a lower user specified `gapsize` value. As a last step for each jump index, the data is set equal to the previous data point.

If `plot==1` two plots are made. The first one, seen in Fig. 3, is a histogram of the number of data point differences as a function of time. This plot helps to choose a good signal `gapsize` value, and in this case a value of 300 was used. Both the old data (•) and the jump-corrected (•) data can be seen in Fig. 4.

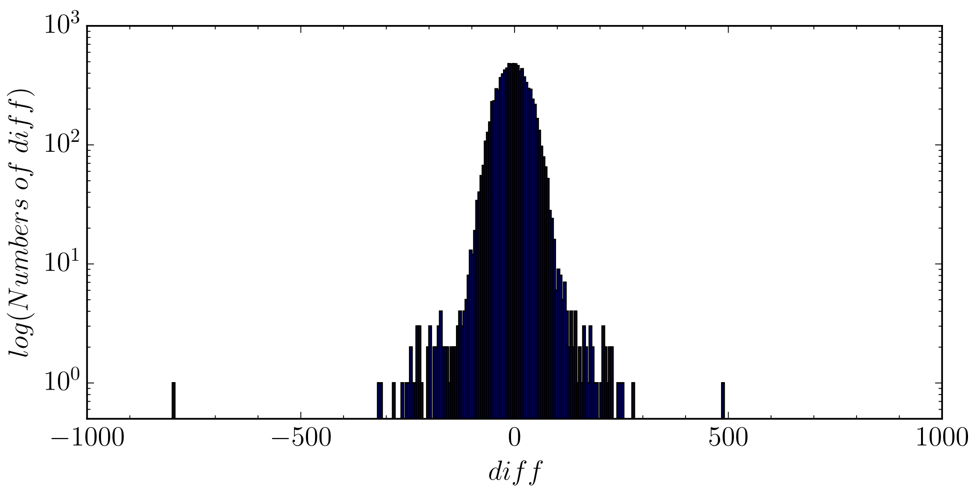


Figure 3: Histogram of the numbers of data points differences as a function of the same differences. The Gaussian shape display the normal distributed scatter of the data, however, outliers can be seen for $diff > |300|$ which are the large jumps in the data.

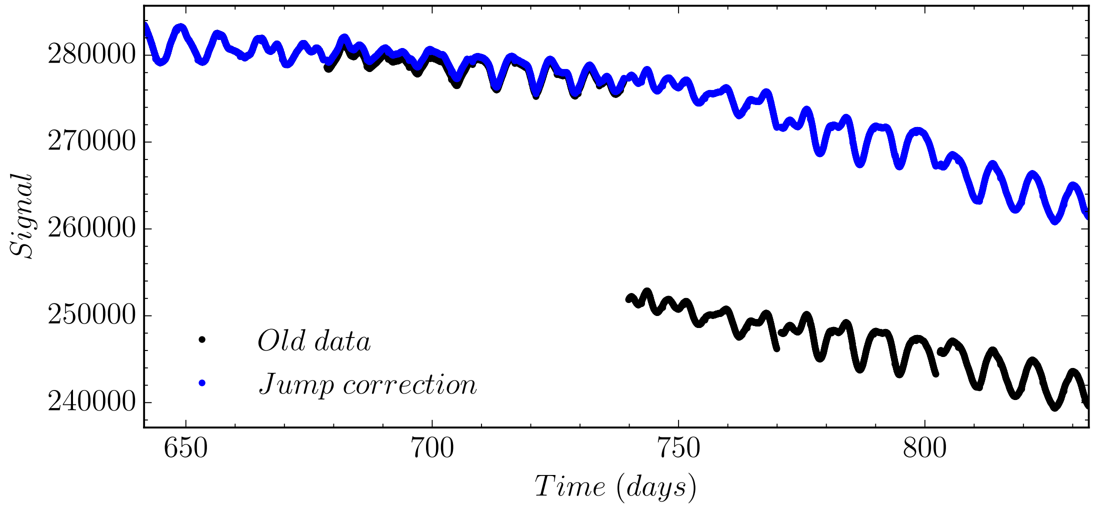


Figure 4: Time series before (•) and after (•) jumps have been corrected for.

2.1.3 Stellar noise

If large periodic signals are visible in the data, these are often produced by stellar pulsation and oscillations and in some cases it is convenient to remove these trends with `clean` before the later normalization. Because we want to find exoplanet signatures, these stellar signals are noise for us. As seen in the jump-corrected time series from Fig. 4 the combined time series are indeed effected by several sinusoidal modes.

Software: `stellar_noise`

```
def stellar_noise(data, f_int, sampling, N, plot=None):
```

The software `stellar_noise` use the already developed `clean` software from the 2. exercise to clean `N` number of peaks in the user specified frequency interval `f_int`. If `plot==1` a comparison of the cleaned data and the old data is both made in the time and frequency domain. For Kepler-1 is was possible to clean 10 peaks between 0.05-0.2 c/d (hence for time varying signals between 5-20 days). The power spectra can be seen in Fig. 5 and the time series can be seen in Fig. 6.

Seen from 5 the function `clean` seems to have a problem of finding peaks below 0.25 c/d which in amplitude are the must dominant ones. Although the stellar variability gets more suppressed as seen in Fig. 6, running the `stellar_noise` function does not much of an improvement. If this bug gets fixed I think that the long term trend above 20 days would significantly improve the time series.

2.1.4 Slow trends

Based on $D(i)$ the data can be corrected for slow trends (normalized) by using a new median filter

$$D_{\mathrm{med}}(i,m) = \mathrm{med}\left[D(i-m),\dots,D(i+m)\right]. \tag{4}$$

As standard $m = 25$, however, this value depends exclusively on the sampling of the data. For Kepler-1 a value of $m = 10$ was applied on the basis of an inspection of transit signatures.

The normalized data (D_{med}) is now smoothed using a so-called *moving mean* filter

$$D_{\mathrm{mean}}(i,m) = \mathrm{mean}\left[D_{\mathrm{med}}(i-m,m),\dots,D_{\mathrm{med}}(i+m,m)\right]. \tag{5}$$

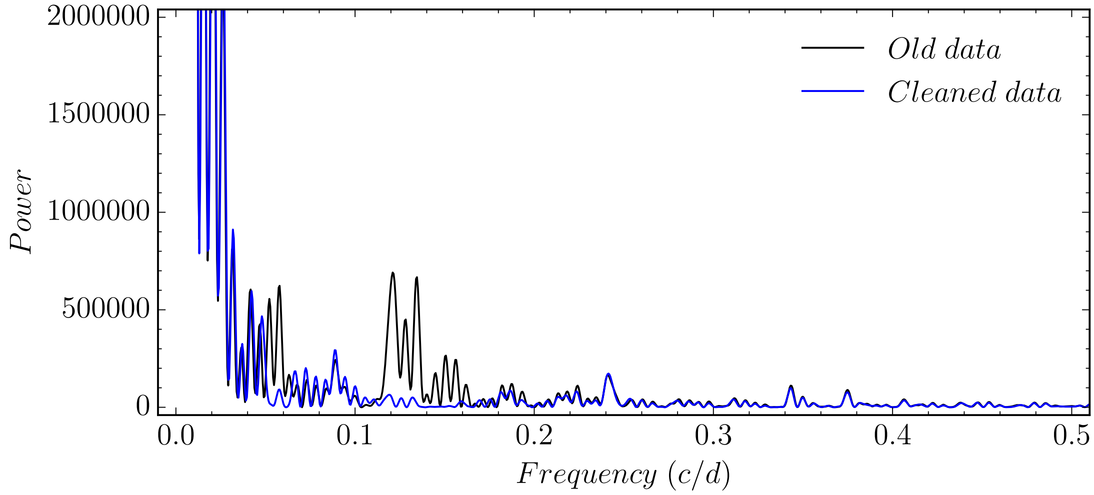


Figure 5: Power spectrum before (—) and after (—) 10 peaks in the range 0.05-0.2 c/d have been cleaned.

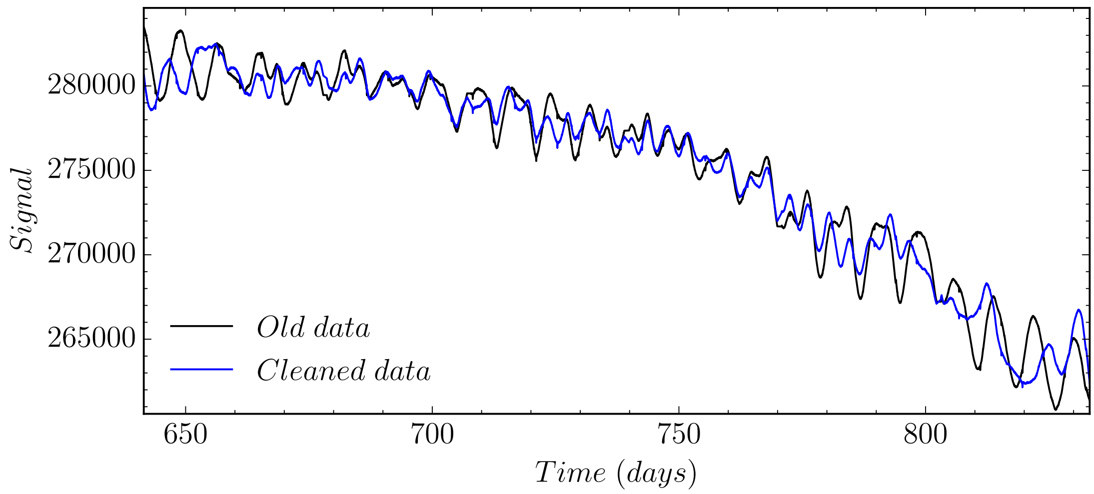


Figure 6: Time series before (—) and after (—) 11 peaks in the range 0.05-0.3 c/d have been cleaned.

Now the corrected data is given by

$$\text{data}(i) = \frac{D(i)}{D_{\text{mean}}(i,m)}. \tag{6}$$

Software: `slowtrend`

```
def slowtrend(data, gapsize, n=None, m=None, plot=None):
```

By using the above given description of applying a median and a mean filter to the data, I have allowed the two filters to use different integers: n for the median filter and m for the mean filter. Again `plot==1` plots the applied filters as can be seen in figure 7 and the normalized and final lightcurve as can be seen in Fig. 8.

In the final light curve several peaks have an signal amplitude of around 1.005 and these belongs to jumps in the data. Although jumps have been corrected for, cleaning peaks in the power spectrum with `stellar_noise` modifies the time series and introduces sometimes small jumps. Avoiding the moving filters getting effected by jumps the median filter with $n = 2$ is replaced for $\pm m$ data points around each jump. Here a jump is detected for when `gapsize` in time is larger than 0.2 days. This technique improves the final times series significantly (an example can be seen by the jump in the lower panel of Fig. 7).

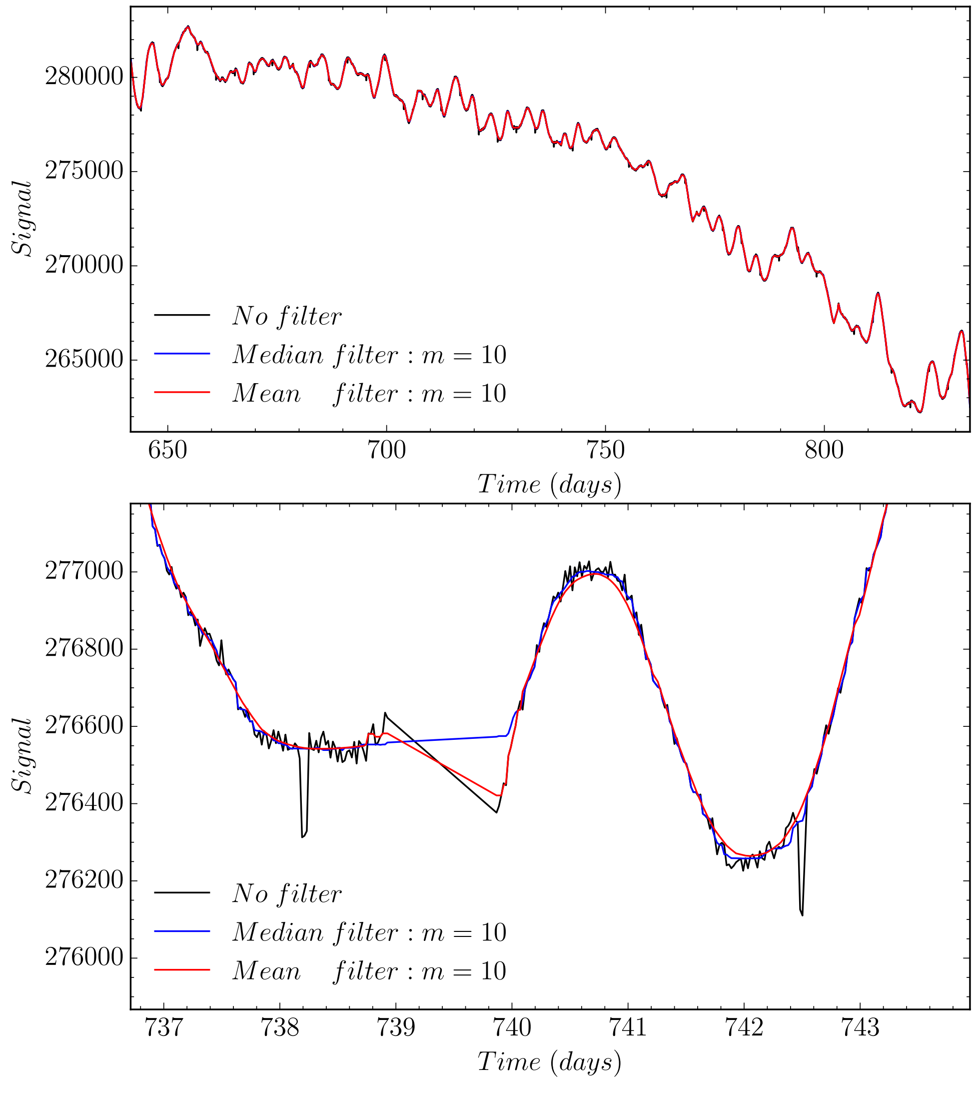


Figure 7: Time series before any filter have been applied (—), after median filter with $m = 10$ have been applied (—), and after mean filter with $m = 10$ have been applied (—). The upper plot shows the whole time series and the lower plot is a zoom-in on the upper plot.

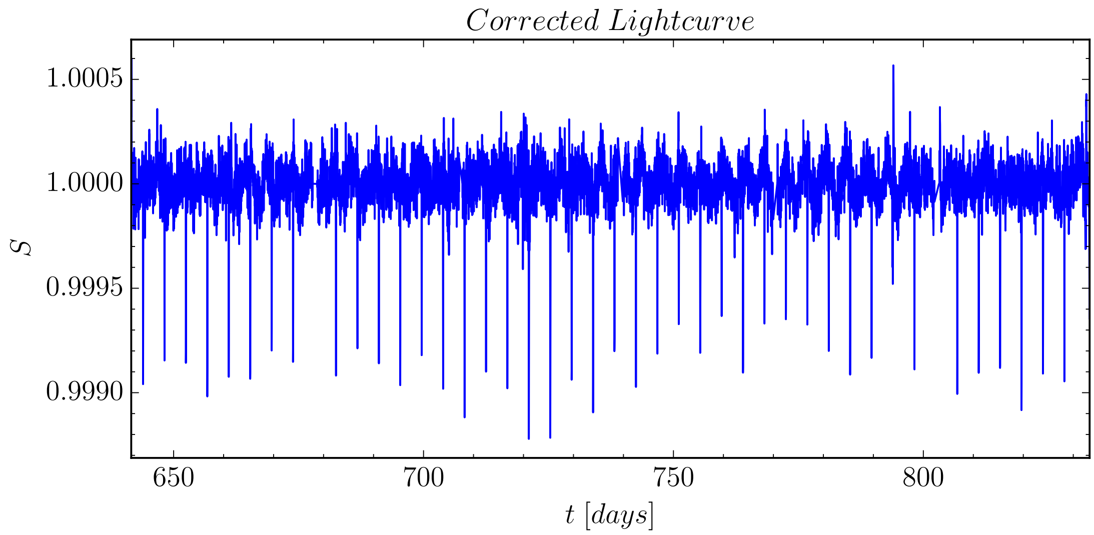


Figure 8: Final time series with clear transiting signals of Kepler-1b.

2.2 Cross-Correlation

Having corrected the data, I start by explaining how CC works. For the CC the data is represented as

$$\text{data}(i) = 1 - \frac{D(i)}{D_{\text{mean}}(i,m)}. \tag{7}$$

The idea behind CC is to find how well two data sets are correlated. This correlation is defined as the sum of the product of the two, where the mean value is subtracted from the individual points. If one add a normalization factor the following expression can be found

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}. \tag{8}$$

From (8) 3 cases can be realized for the so-called cross-correlation coefficient

- $r \sim +1 : x$ and y are *strongly* correlated
- $r \sim \ \ 0 : x$ and y are not correlated
- $r \sim -1 : x$ and y are *strongly* anti-correlated.

Software: `cc_coefficient`

```
def cc_coefficient(x, y):
```

Making my code look more simple I made a separate function for r called `cc_coefficients`. As can be seen from (8) it only needs the two data sets $\{x, y\}$ that needs to be correlated. The x data set is science data and the y data set is a time series model constructed by

$$\text{Model}(i, P, \phi) = \begin{cases} 1, & |t_i - mP - \phi| < \frac{\Delta T}{2} \\ 0, & \text{elsewhere} \end{cases} \tag{9}$$

where t_i is the i'th time, m is a integer, P is the period, ϕ is the phase, and ΔT is the transit duration. A value of a few hours for ΔT is recommended.

Software: `model`

```
def model(t, P, phi, dT, A):
```

To get the needed model, a function called `model` is created which returns the model data. As seen from (9) `model` needs to take the above mentioned parameters as input. To find the right model one needs to find the first point where (9) is satisfied, and then find the endpoint. By doing this for all possible start and end points, the complete set is then used as the model data.

Software: `crosscor`

```
def crosscor(data, P_int, phi_int=None, model_const=None, save=None):
```

A combined software `crosscor` is made and takes the `data`, a period interval (`P_int`), a possible phase interval (`Phi_int`), possible model constants (`model_const`), and an option for saving the data with a user specified name (if `save=='name'`). The input model constants are the transit duration, ΔT, a period resolution, ΔP, a phase resolution, $\Delta\phi$, and an scale CC amplitude, A. As can be seen in the following code block, CC is performed such that for every period (i), up to the total number of periods (N), the CC coefficient is calculated for every phases (j) up to the total number of phases (M). The result is a $N \times M$ matrix of CC amplitudes, here denoted `cc`

```
# Perform Cross-correlation:
for i in range(N):
    for j in range(M):
        y       = model(t, P[i], phi[j], dT, A)
        r_cc    = cc_coefficient(x, y)
        cc[i, j] = r_cc
```

One can tune ΔT from an inspection of the lightcurve, however, as long as the value is in the same ballpark of the actual transit duration (usually a couple of hours) the actual value do not matter much. First a initial calculation with a low sampling and wide range in parameter space was made. Then a finner grid was made around the crude initial solution. For all exoplanets I ended up using the parameter space of $N \times M = 400 \times 300 = 120,000$ with period and phase interval of $P \pm 10$ min and $\phi \pm 5$ min around the initial period and phase, respectively. For Kepler-1b the following grid was used $\{\Delta P, \Delta\phi\} = \{3, 17\}$ min. From the lightcurve the transit duration of Kepler-1b was estimated to $\Delta T = 130$ min.

Now a surface plot was made of the parameters space $\{P, \phi, A\}$ which can be seen in the upper plot of Fig. 9, where the maximum peak give $\{P, \phi\} = \{4.2875(21), 0.755(12)\}$ days. For clarity a projection of the maximum CC amplitude as a function of the period can be seen in the lower plot in the same figure.

2.3 Auto-Correlation

Software: `autocor`

```
def autocor(data, npeaks, cutoff, plot):
```

The implementation of the AC is done with the function `autocor`. Before an AC can take place the date needs first to be moved to a uniform time grid and, secondly, also linear interpolated in

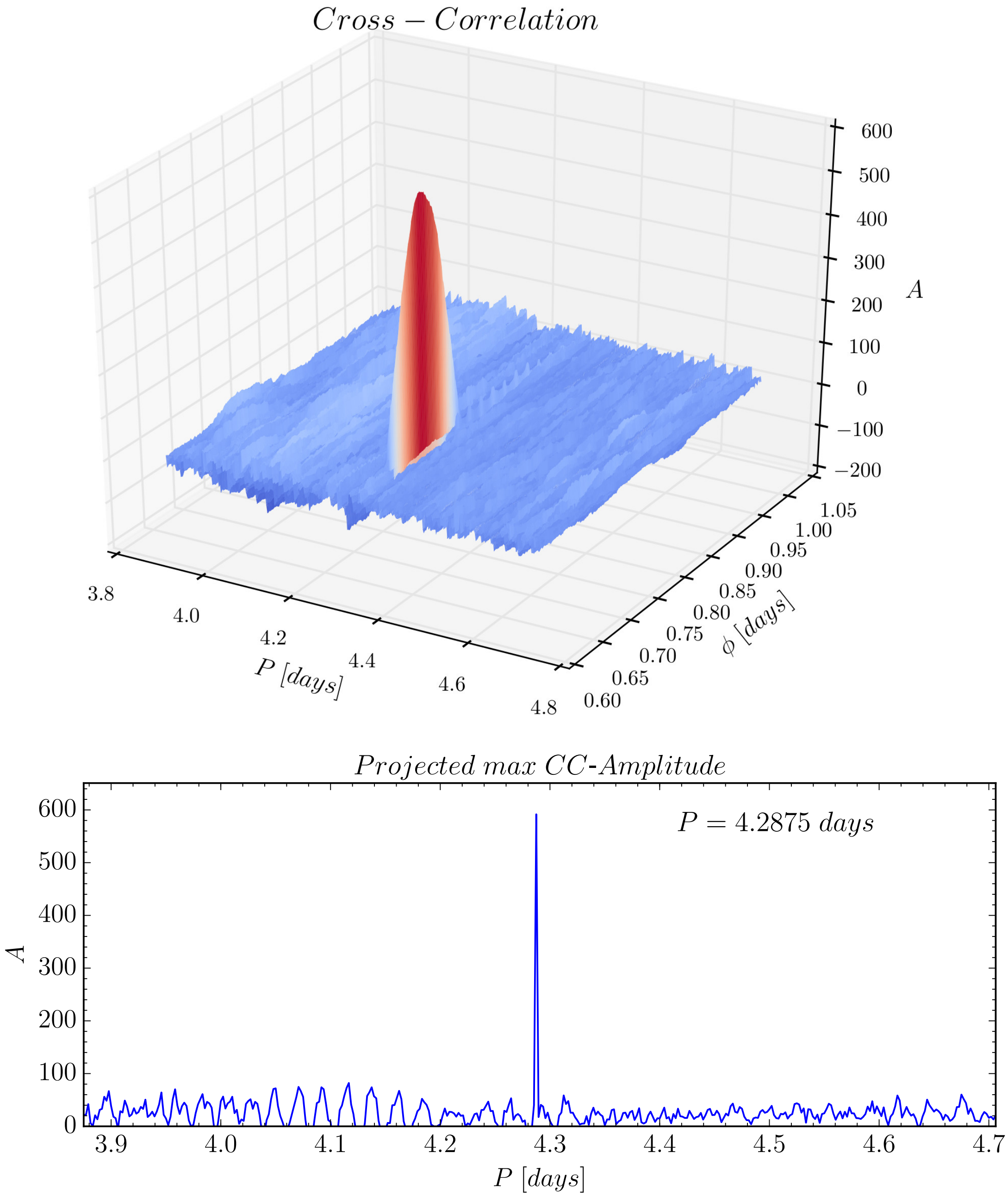


Figure 9: CC results for Kepler-1b: Upper figure shows a 3D plot the CC result. Lower plot shows projected CC amplitude of maximum power as a function of the period. From these plots $\{P, \phi\} = \{4.2875(21), 0.755(12)\}$ days.

gaps where no data is present. To do this I used the `griddata` routine from the Scipy library which does exactly this. Also all data points below $1 - S_{\text{grid}} < 0$ was equaled to zero after the interpolation. In Fig. 10 the corrected data (—) and the interpolated data ($\bullet$) is shown for Kepler-1.

After the interpolation an AC is performed which can be seen in Fig. 11. Mathematically the AC is done by comparing the interpolated $1 - S_{\text{grid}}$ data with the same set of data shifted by $k\Delta t$ for different values of k using the following equation

$$r(k) = \frac{\sum_{i=1}^{N-k} (x_i - \bar{x}_{l \to N-k})(x_{i+k} - \bar{x}_{l+k \to N})}{N - k}. \tag{10}$$

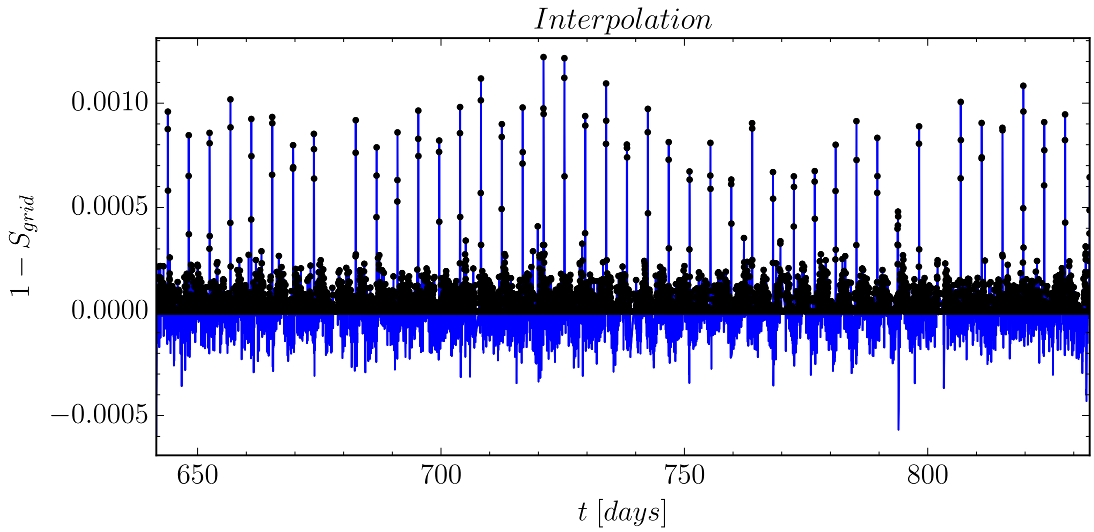


Figure 10: Corrected data (—) and interpolated data (•) for Kepler-1.

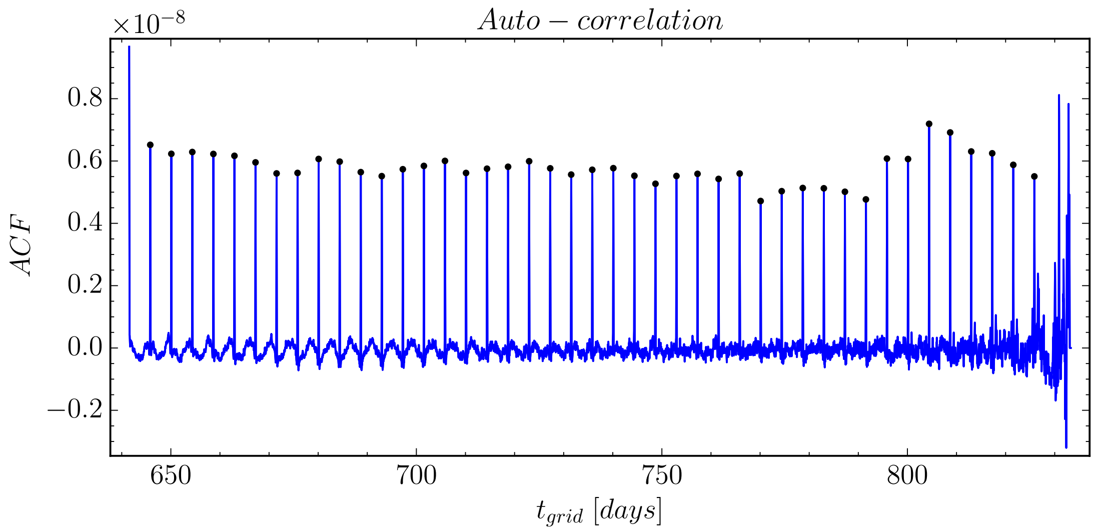


Figure 11: Auto-Correlation Function (ACF) vs. time for Kepler-1. The black dots are peak signals belonging to transits.

Here N is the total number of grid points and x is (as for the CC) represented by

$$x_i = 1 - S_{\text{grid}} = 1 - \frac{D(i)}{D_{\text{mean}}(i,m)}. \tag{11}$$

The AC defined by (10) returns a positive value every time the data is correlated to itself. Thus, if a periodic signal is present in the time series it will be clearly visible from the AC as peaks. The peaks in Fig. 11 are found using the program `argrelmax` also from the Scipy library. This function takes the signal and a integer number of peaks (`npeaks`) that should be search for as input. As k increases the AC compare fewer and fewer data, hence, the noise level increases toward the end which also is evident from Fig. 11.

Fom the ACF only peaks inside a user defined time `cuoff_x` and ACF `cutoff_y` value were selected. Due to the scatter toward the end of the correlation, one should take care with peaks here. In the case for Kepler-1b only peaks `cutoff_x`= 828 days and above `cutoff_y`= 4.4×10^{-9} were selected.

As can be seen in Fig. 12 the period can be determined from a linear fit between the measured periods ($N \times P$) from the ACF and the corresponding calculated period ($N \times P - P$) from the measured period. Before using ACF periods the initial time value was subtracted from all periods (this only give the right result if the very first transit peak is used in the fit!). Next the calculated periods $N \times P - P$ was found by the following code block

```
# Calculate periods:
P_cal = zeros(len(P_mea))
for i in range(1, len(P_mea)+1):
    P_cal[i-1] = i*P-P
```

Lastly `autocor` plots Fig. 12 (if `plot==1`) and print out the the best fit period, the mean period value, and a 1σ uncertainty limit. As also can be seen from the titling of Fig. 12 and in Tab. 2 the results for Kepler-1b are: $P_{\text{mean}} = 4.2868(84)$ days and $P_{\text{fit}} = 4.2874(84)$ days.

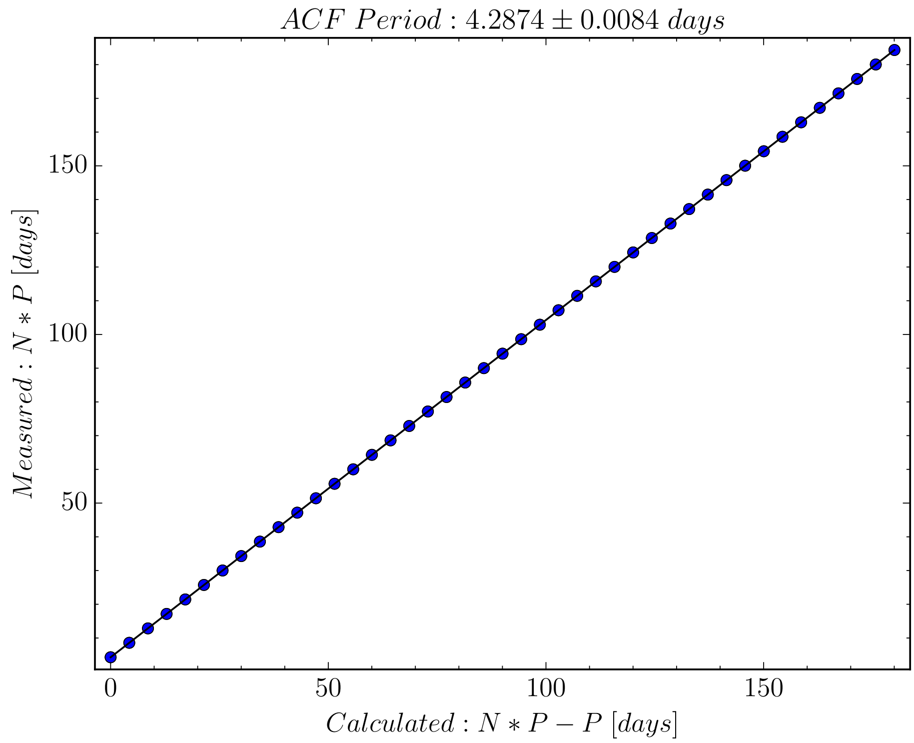


Figure 12: Linear fit of the measured and calculated values of $N \cdot P$ resulting from the AC of Kepler-1b. The b parameter from the linear regression $a \cdot x + b$ is the best fit period solution and can be seen in the title.

2.4 Phase folding

As a final thing it is first of all convenient to make a phase folded diagram as a sanity check of the period and, secondly, due to such a diagram can be used to find planetary parameters (however this is a project in it self and will not be done here). With the a period estimate both from CC and AC, I look at both solutions and tuned P to see if a better solution could be found. The best period for Kepler-1b is $P = 4.2869$ days, hence, very close to the period measured from the transit lightcurve alone $P = 4.29$ days. The best solution of Kepler-1b can be seen in Fig. 13 and in Tab. 2.

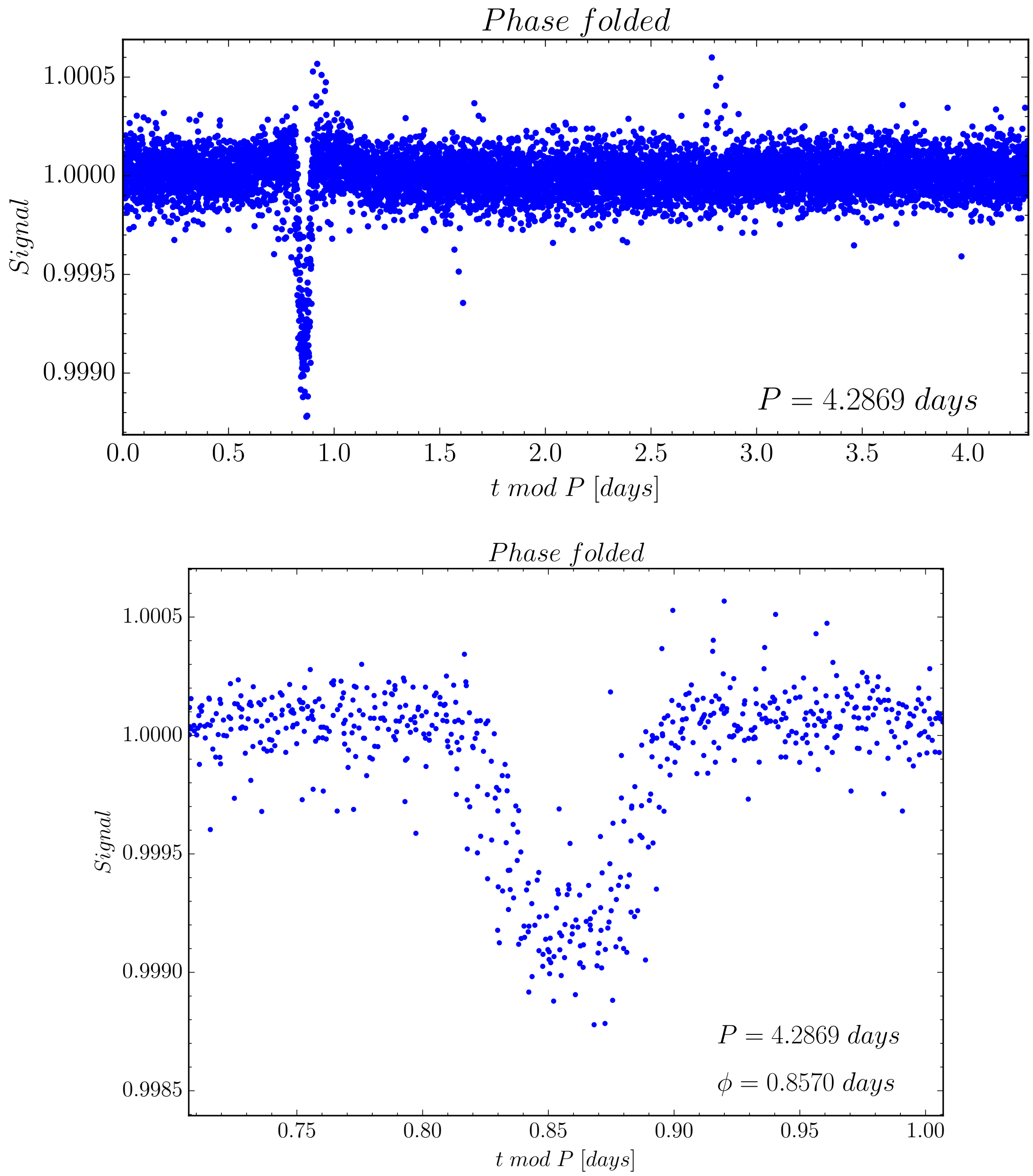


Figure 13: Phase folded diagram of with a best solution period of 4.2869 days and phase of for Kepler-1b. The upper plot shows the full phase folded diagram and the lower plot is a zoom-in on the transit.

3 Rest of Kepler Data

The analysis of the other Kepler host stars and their planets will be given in the following sections. To make is more clear I have made Tab. 2 which display the user defined values I have used in each program. The input parameters in the software `autocor` will however be specified for each sample later. Moreover, when performing the CC, for all samples I used a parameter space of $P_{\text{interval}} = P \pm 10$ hours and $\phi_{\text{interval}} = \pm 5$ hours with a resolution of $\{\Delta P, \Delta\phi\} = \{3, 2\}$ min. This resulted in a $N \times M = 400 \times 300$ grid, hence, 120,000 calculations needed to be done for the CC.

Table 1: User Defined Parameters

Software→		locate		jumps	stellar_noise		slowtrend				crosscor
Star	ID	n	cutoff	gapsize	f_int (c/d)	N	gapsize (days)	jump	n	m	dT (min)
Kepler-1	b	1	3.0×10^{-4}	300	0.05–0.2	10	0.2	1	10	10	130
Kepler-2	b	1	3.1×10^{-4}	300	—	—	0.2	1	10	10	230
Kepler-3	b	1	3.0×10^{-4}	200	—	—	0.5	0	25	25	170
	c										250
	d										490
Kepler-4	b	2	3.4×10^{-4}	300	—	—	0.2	1	25	10	256

3.1 Kepler 2

The Kepler-2 time series is plotted in Fig. 14 and one periodic transit signal is clearly visible. Using (1) the period of Kepler-2b is around $P_{\text{transit}} = 9.29$ days. Stellar noise was not visible from the lightcurve and thus not corrected for. Fig. 15, 16, and 17 shows the jump correction, the final corrected lightcurve, and the CC result, respectively. In `autocor` an upper ACF `cutoff_y` value of 2.5×10^{-9} was used. The ACF is shown in Fig. 18 and the best AC fit is shown in Fig. 19 and give $P_{\text{fit}} = 9.290(10)$ days. The mean value of the ACF result is $P_{\text{mean}} = 9.288(10)$ days. The best solution from the phase folded lightcurve is $P = 9.2875$ days as shown in Fig. 20.

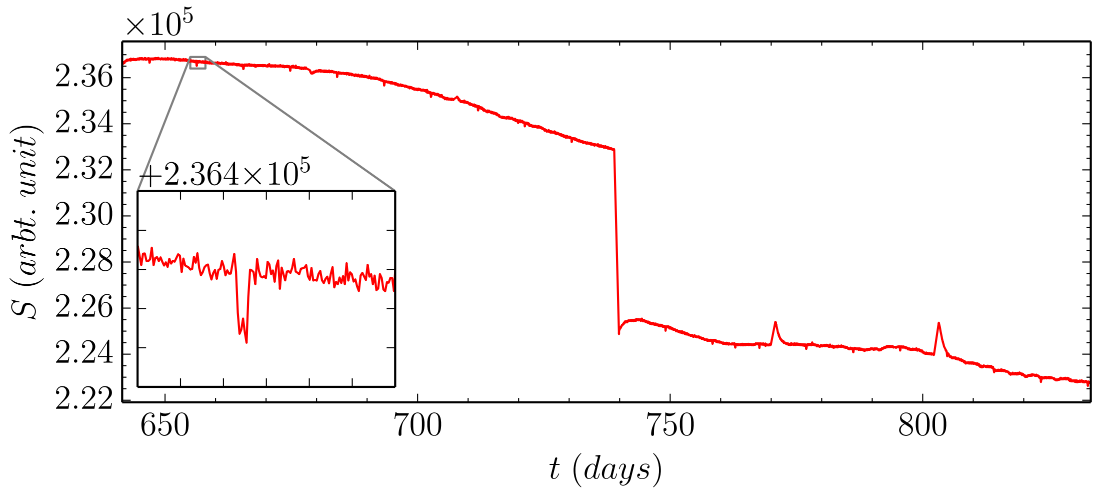


Figure 14: Time series for the 2. unknown exoplanet host star observed by Kepler. A zoom-in on a exoplanet transit is also visible.

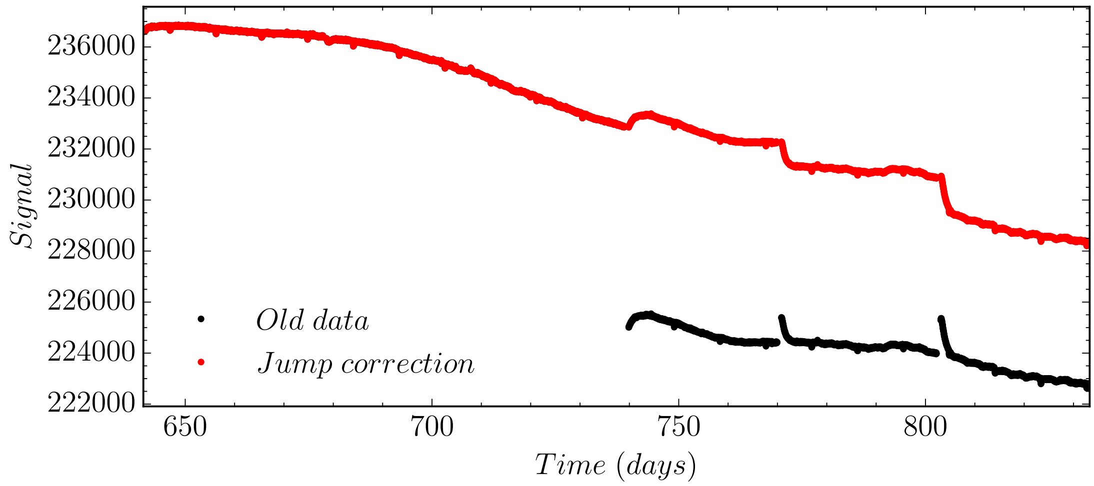


Figure 15: Time series before (•) and after (•) jumps have been corrected for Kepler-2.

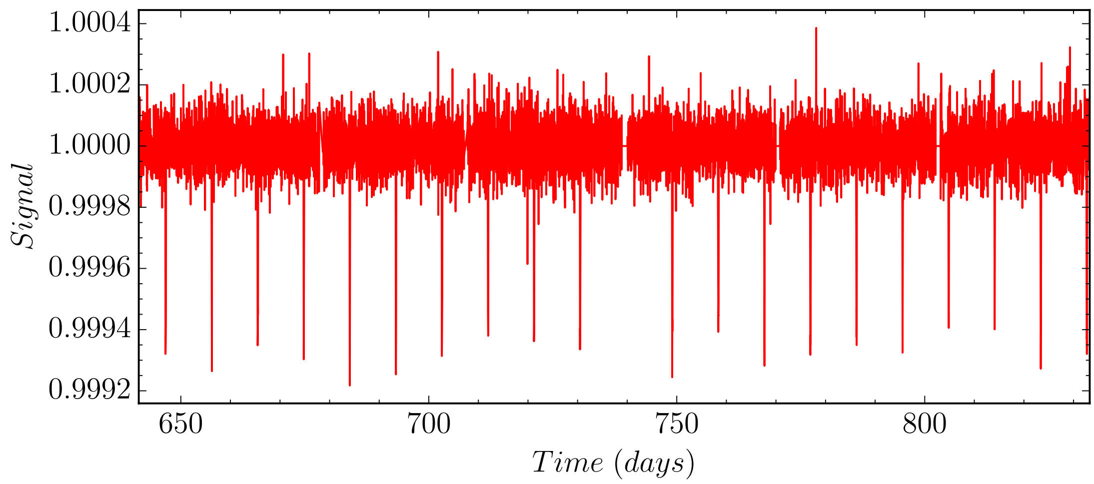


Figure 16: Final time series with clear transiting signals of Kepler-2b.

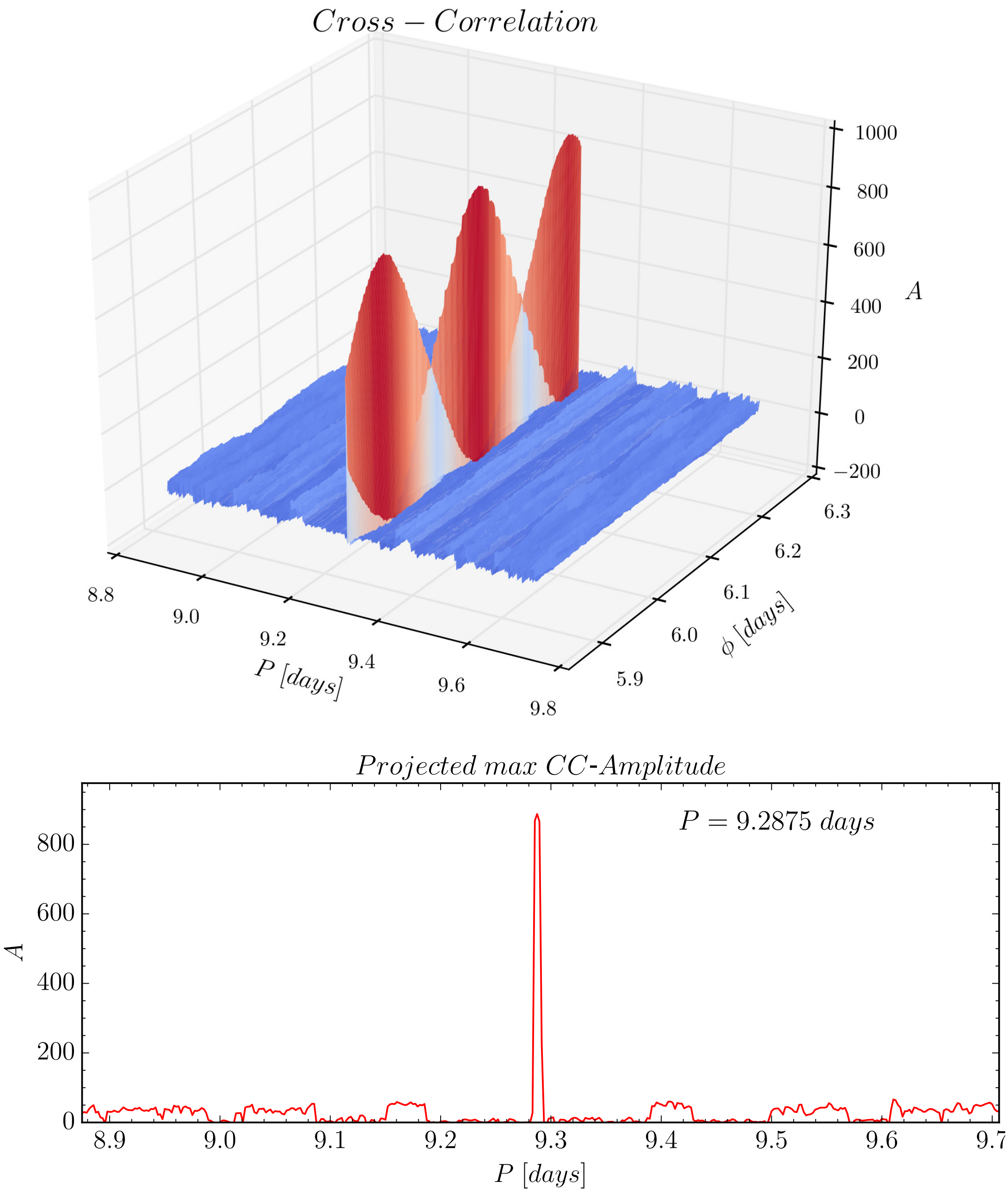


Figure 17: CC results for Kepler-2b: Upper figure shows a 3D plot of the CC result. Lower plot shows projected CC amplitude of maximum power as a function of the period. From these plots $\{P, \phi\}$ = {9.2875(21), 6.10(2)} days.

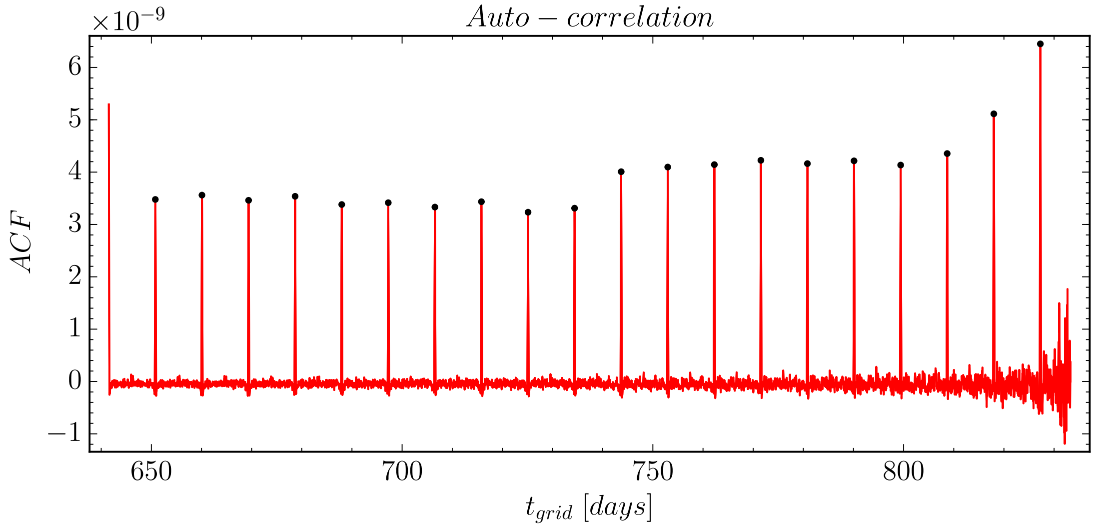


Figure 18: Auto-Correlation Function (ACF) vs. time for Kepler-2b. The black dots are peaks signals belonging to transits.

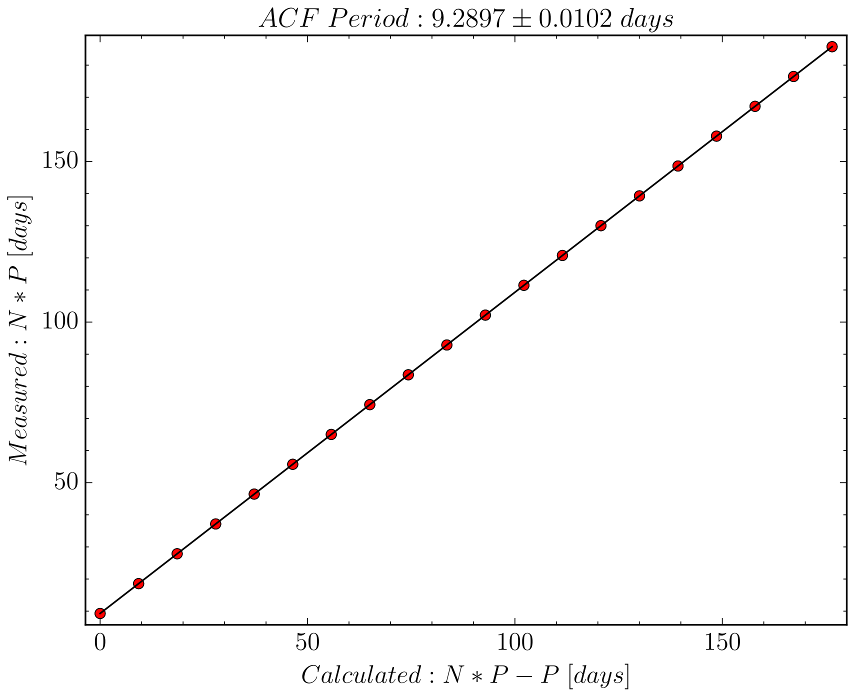
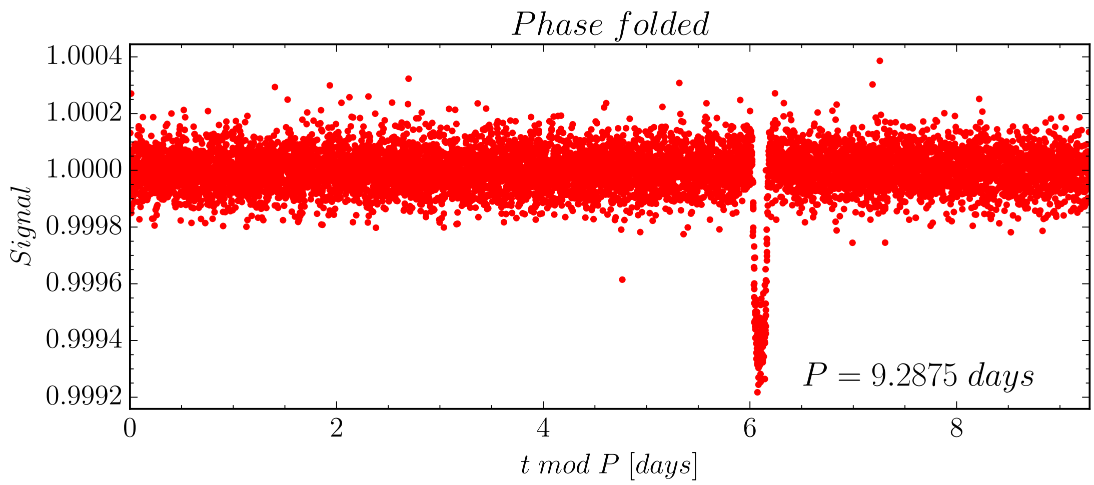


Figure 19: Linear fit of the measured and calculated values of $N \cdot P$ resulting from the AC of Kepler-2b. The b parameter from the linear regression $a \cdot x + b$ is the best fit period solution and can be seen in the title.



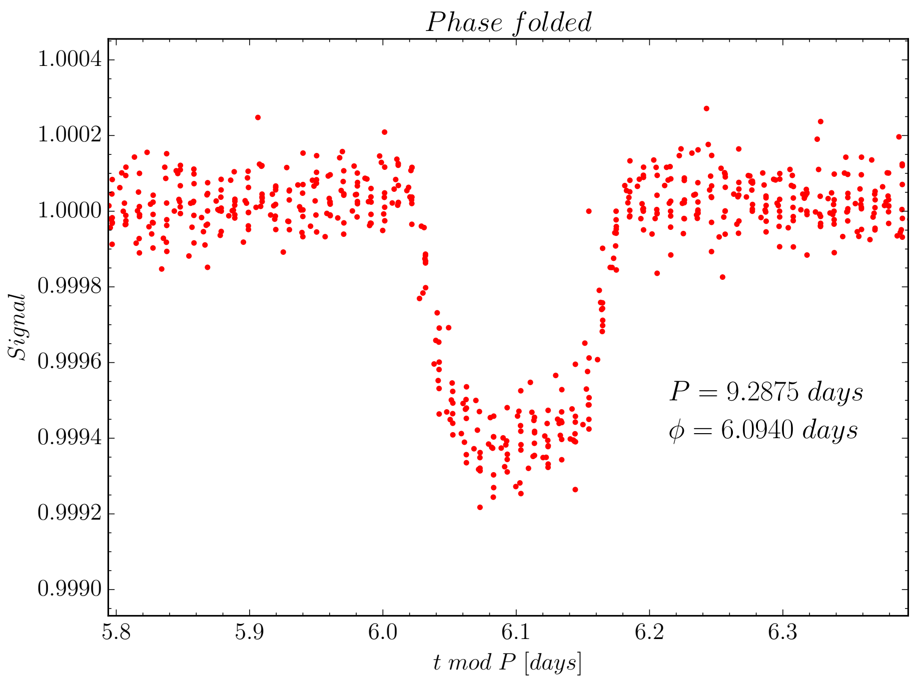


Figure 20: Phase folded diagram of with a best solution period of 9.2875 days and phase of 6.0940 days for Kepler-2b. Upper plot is the full diagram and the lower plot is a zoom-in on the transit using ϕ.

3.2 Kepler 3

Kepler-3 time series is plotted in figure 21 and two periodic transit signal is clearly visible. Using (1) the period of Kepler-3b is around $P_{\text{transit}} = 3.23$ days and for Kepler-3c $P_{\text{transit}} = 10.26$ days. Stellar noise was not corrected for. In `slowtrends` because the jump correction in this filter actually removed some of the lower period transits, no low-median-filtered jump correction was used here. Fig. 22 and 23 shows the jump correction and the final corrected lightcurve, respectively. From the final lightcurve another exoplanet with only 2 transits was visible. These transits can be seen in Fig. 24 and this indicates a period of around $P_{\text{transit}} = 77.6$ days.

Fig. 25, 26, and 27 shows the CC result for Kepler-3b, Kepler-3c, and Kepler-3d, respectively. The best solution of Kepler-3b is actually a alias peak from the main peak of Kepler-3c. This is evident due to the fact that the period is exactly one third of that of Kepler-3c and the phase is exactly that of Kepler-3c. Thus, the period and phase of Kepler-3b was found by adjusting the period grid range, hence, avoiding the peak around 3.61 days as seen in the Fig. 25. Due the fact that only two transits for Kepler-3d were visible in data, from the upper plot of Fig. 27 the phase information is more or less lost.

Although a third planet may be present in the data, it was not possible to use AC to estimate its period. For Kepler-3b a ACF `cutoff_y` region of $[1.6\times10^{-9}, 7.8\times10^{-9}]$ was used and a `cutoff_x` region of [666, 790] were used. For Kepler-3c only a lower limit of `cutoff_y`$= 7.8\times10^{-9}$ was used. The ACF function together with the found peaks for Kepler-3b and Kepler-3c are shown in Fig. 28 and the best AC fit is shown in Fig. 29, respectively.

For Kepler-3b, Kepler-3c, Kepler-3d a best solution of P = 3.6962, 10.8541, and 77.62 days and ϕ = 0.7480, 4.1680, and 75.25 days was found, and these are shown in Fig. 30, 31, and 32, respectively.

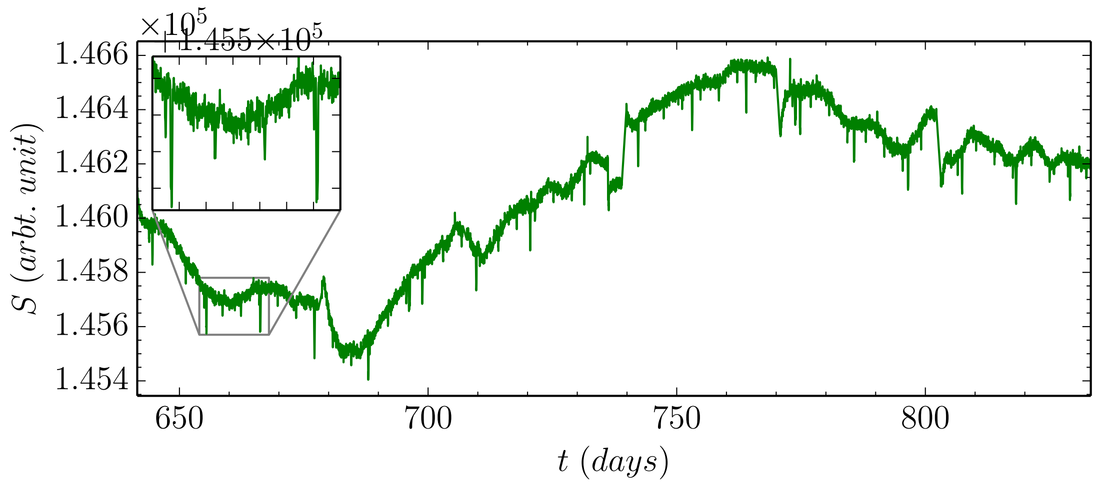


Figure 21: Time series for the 3. unknown exoplanet host star observed by Kepler. A zoom-in on a exoplanet transit is also visible.

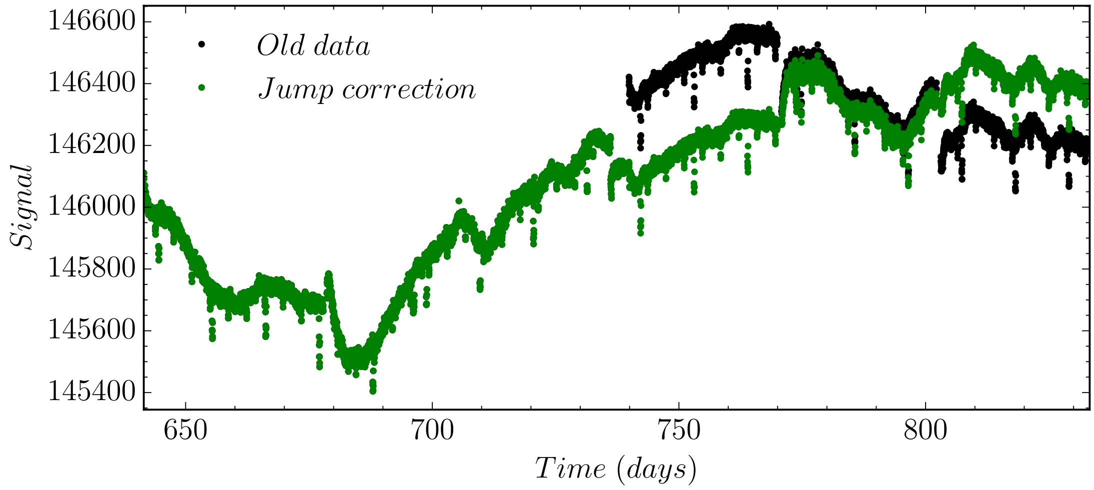


Figure 22: Time series before (•) and after (•) jumps have been corrected for Kepler-3.

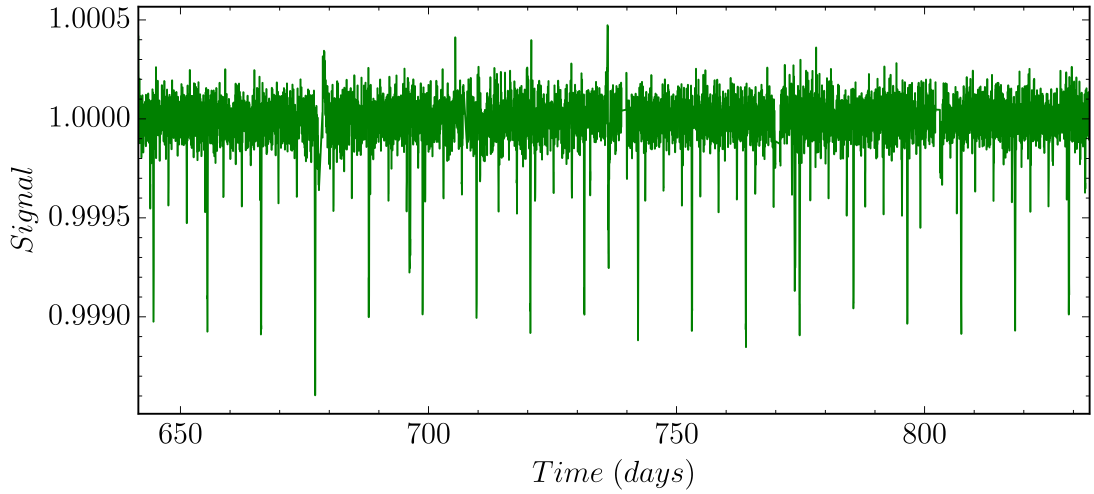


Figure 23: Final time series with clear transiting signals of Kepler-3b and Kepler-3c.

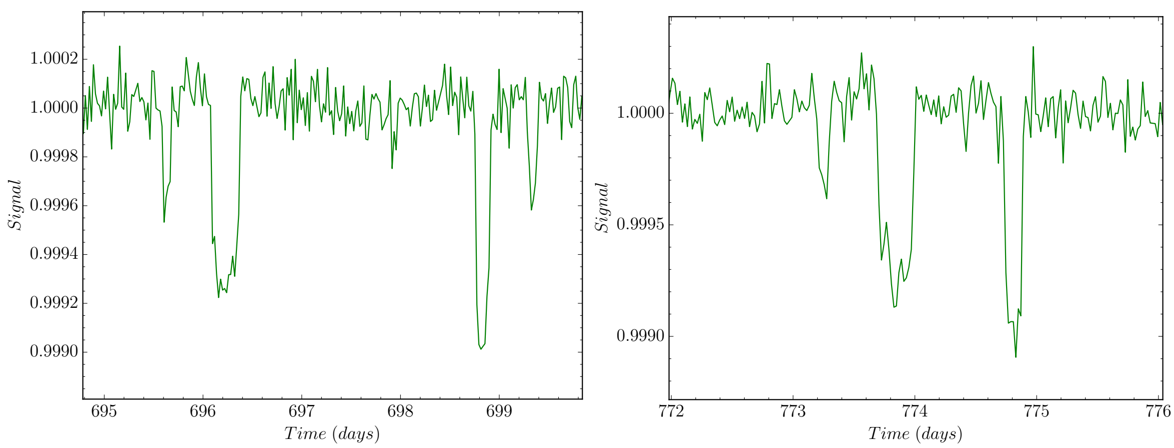


Figure 24: Final time series showing a new possible exoplanet candidate Kepler-3d. Looking at the right hand plot the first, second, and third transit belongs to Kepler-3b, Kepler-3d, and Kepler-3b, respectively. From the transits alone the period of Kepler-3d is around $P = 72.6$ days.

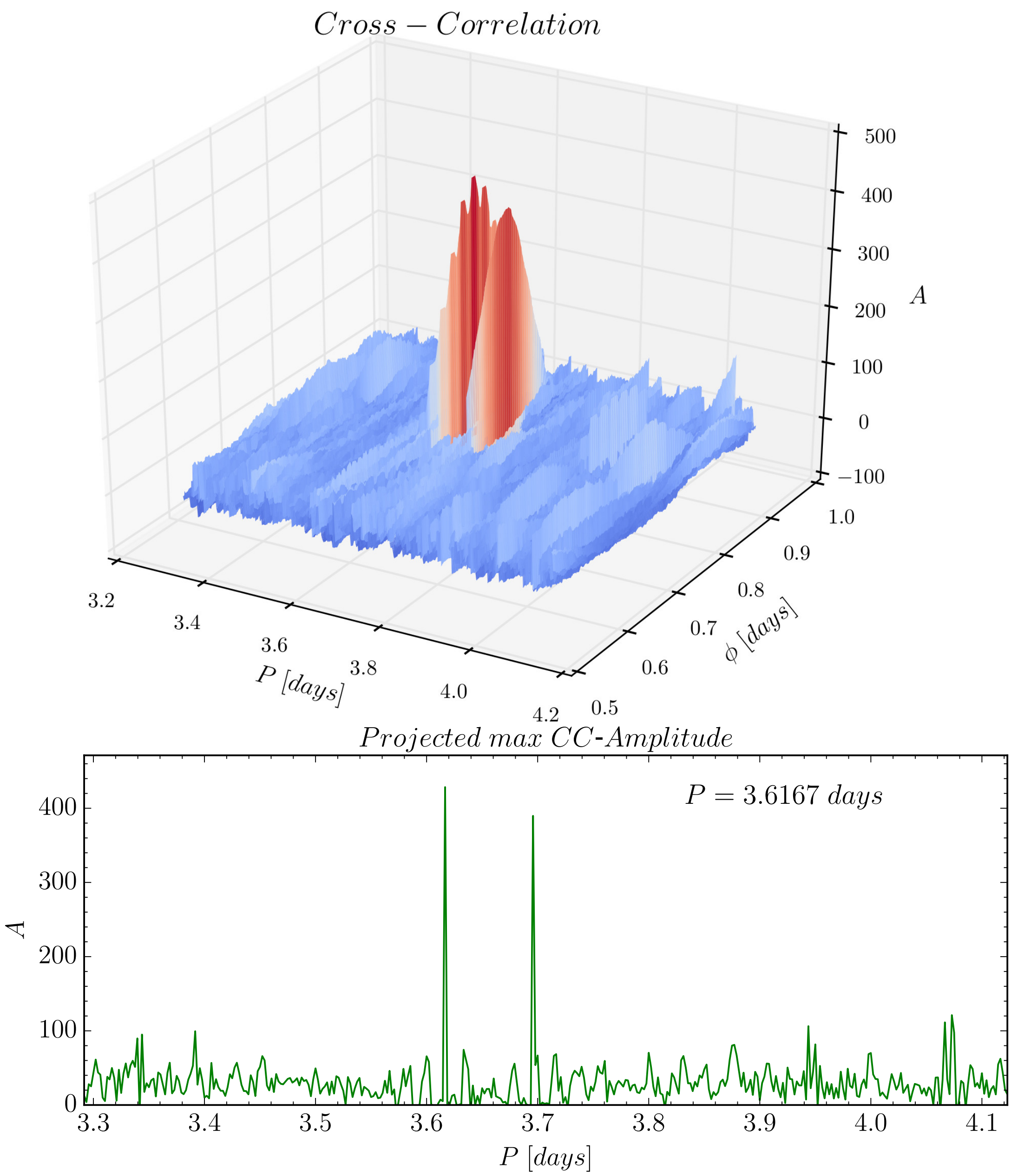
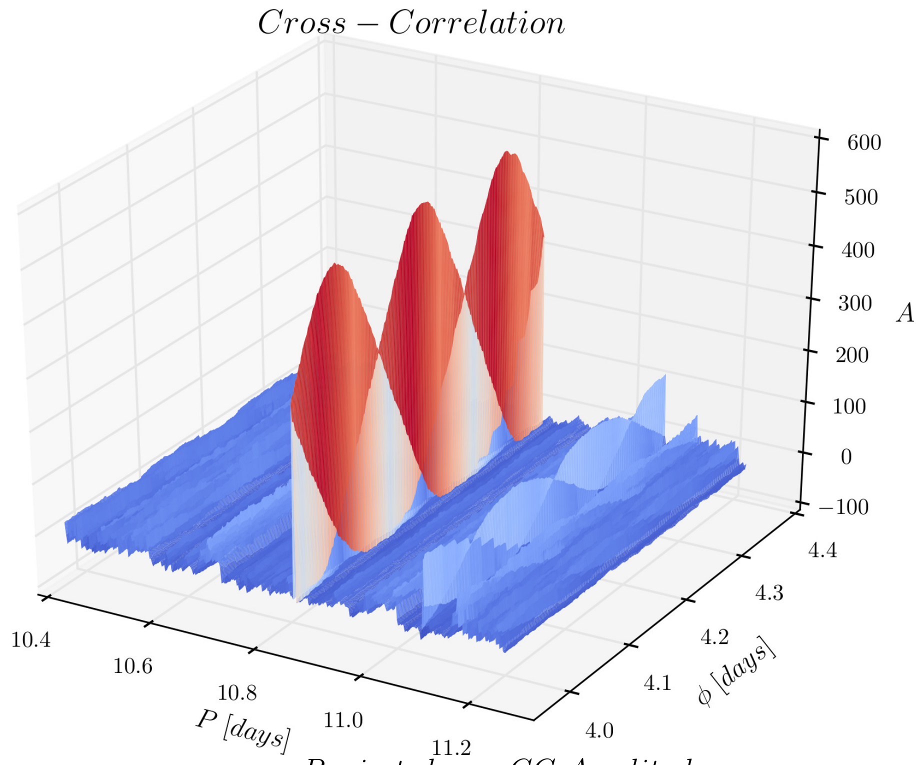


Figure 25: CC results for Kepler-3b: Upper figure shows a 3D plot the CC result. Lower plot shows projected CC amplitude of maximum power as a function of the period. The highest peak is an alias from the main peak of Kepler-3c, however, the second largest peak is belonging to Kepler-3b. This peaks was found to have $\{P, \phi\}$ = {3.6958(21), 0.7475(14)} days.



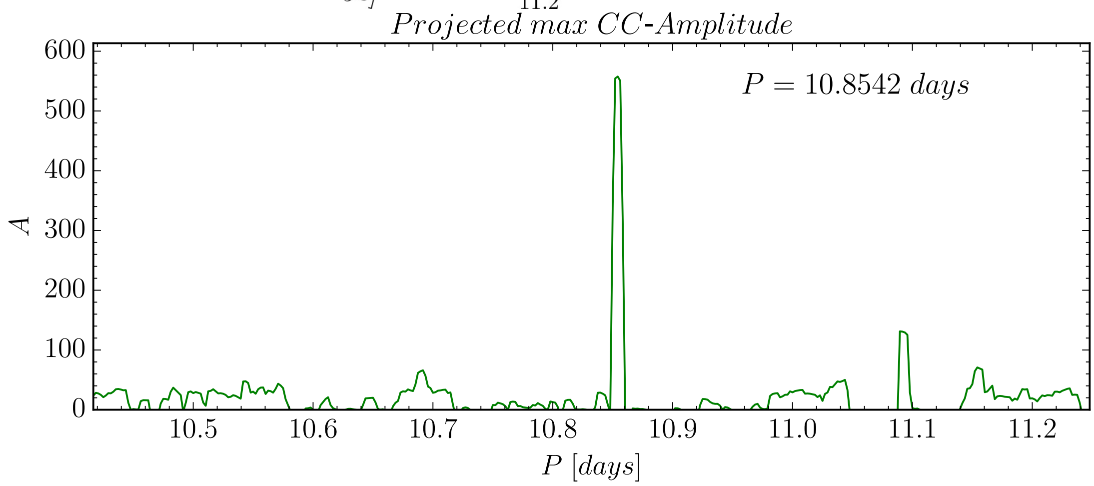
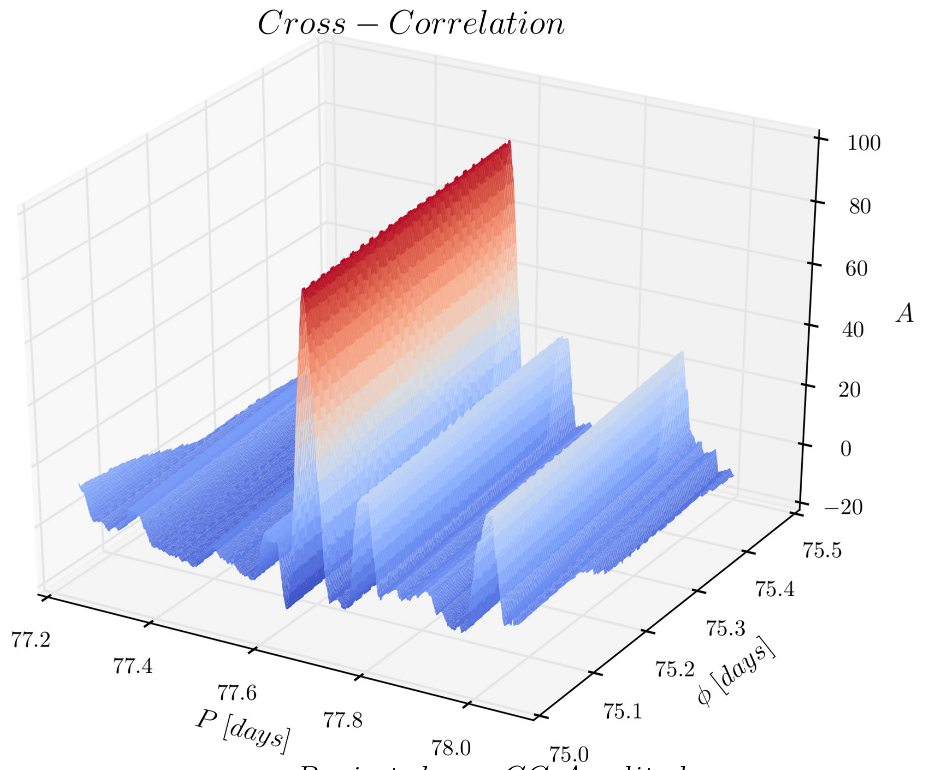


Figure 26: CC results for Kepler-3c: Upper figure shows a 3D plot the CC result. Lower plot shows projected CC amplitude of maximum power as a function of the period. From these plots $\{P, \phi\}$ = {10.8542(21), 4.1653} days.



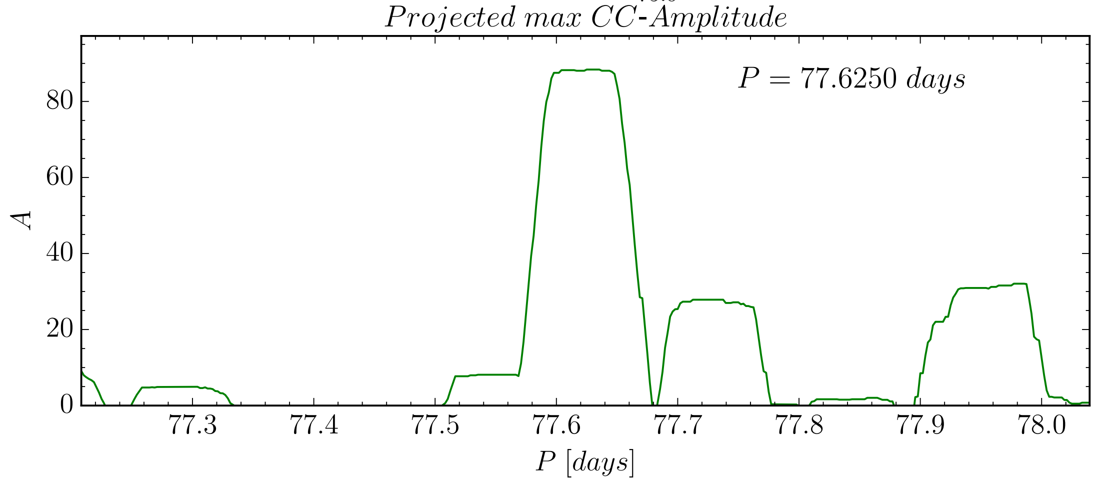


Figure 27: CC results for Kepler-3d: Upper figure shows a 3D plot the CC result. Lower plot shows projected CC amplitude of maximum power as a function of the period. From these plots $\{P, \phi\}$ = {77.6250(21), 75.2097(14)} days.

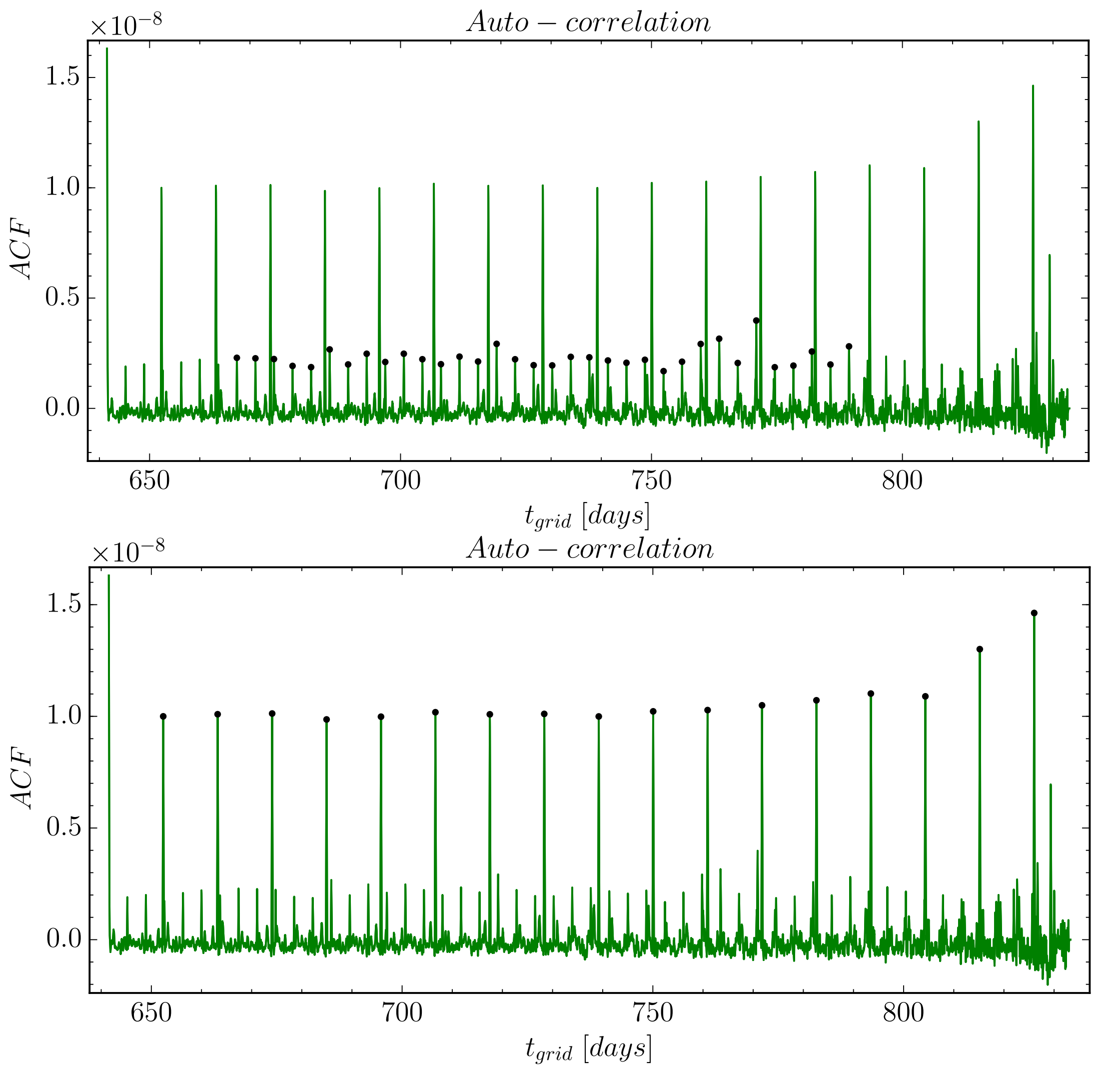


Figure 28: Auto-Correlation Function (ACF) vs. time for: upper plot for Kepler-3b and lower plot for Kepler-3c. The black dots are peaks signals belonging to transits and thus used.

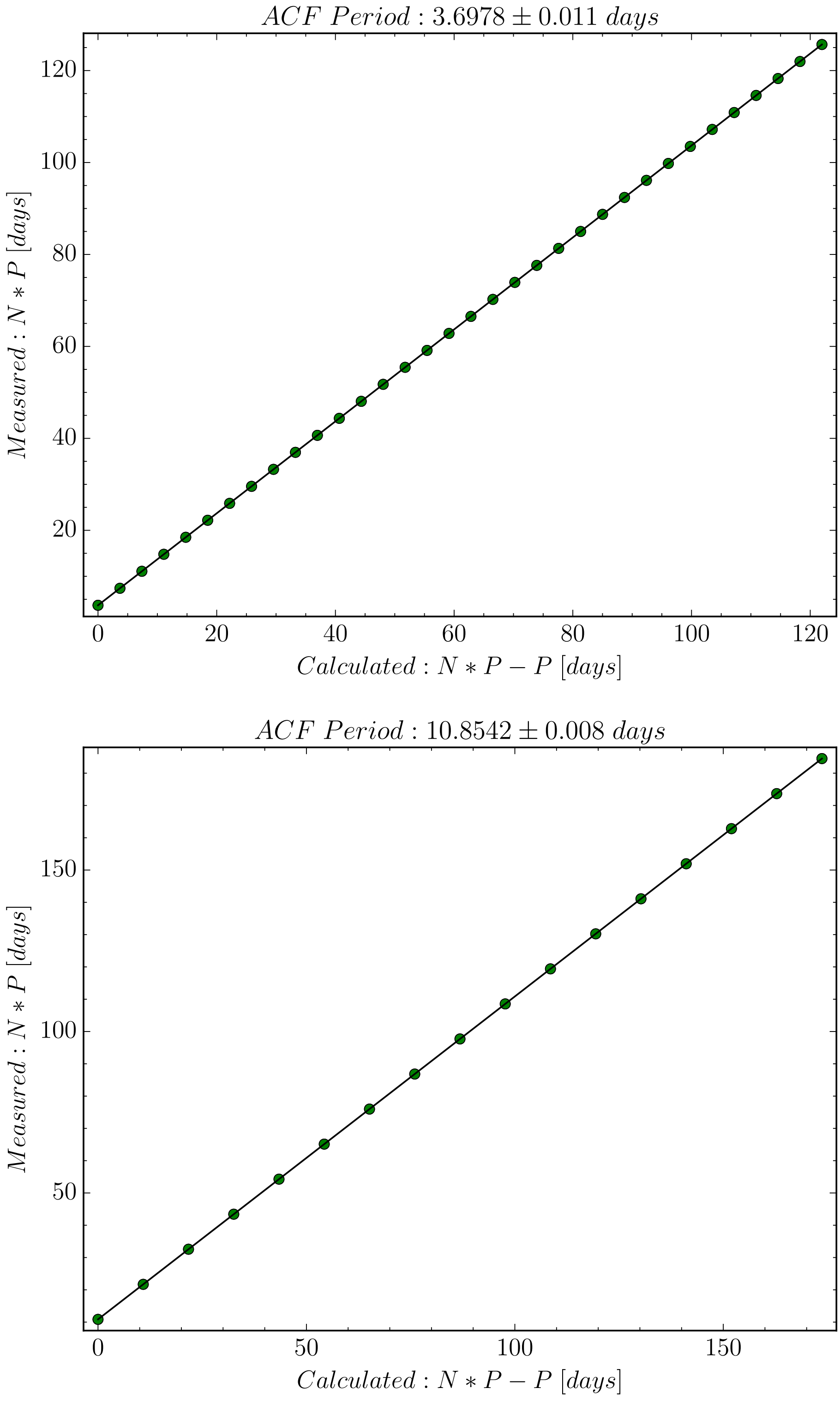


Figure 29: Linear fit of the measured and calculated values of $N \cdot P$ resulting from the auto-correlation of Kepler-3b (upper plot) and Kepler-3c (lower plot). The b parameter from the linear regression $a \cdot x + b$ is the best fit period solution and can be seen in the titles.

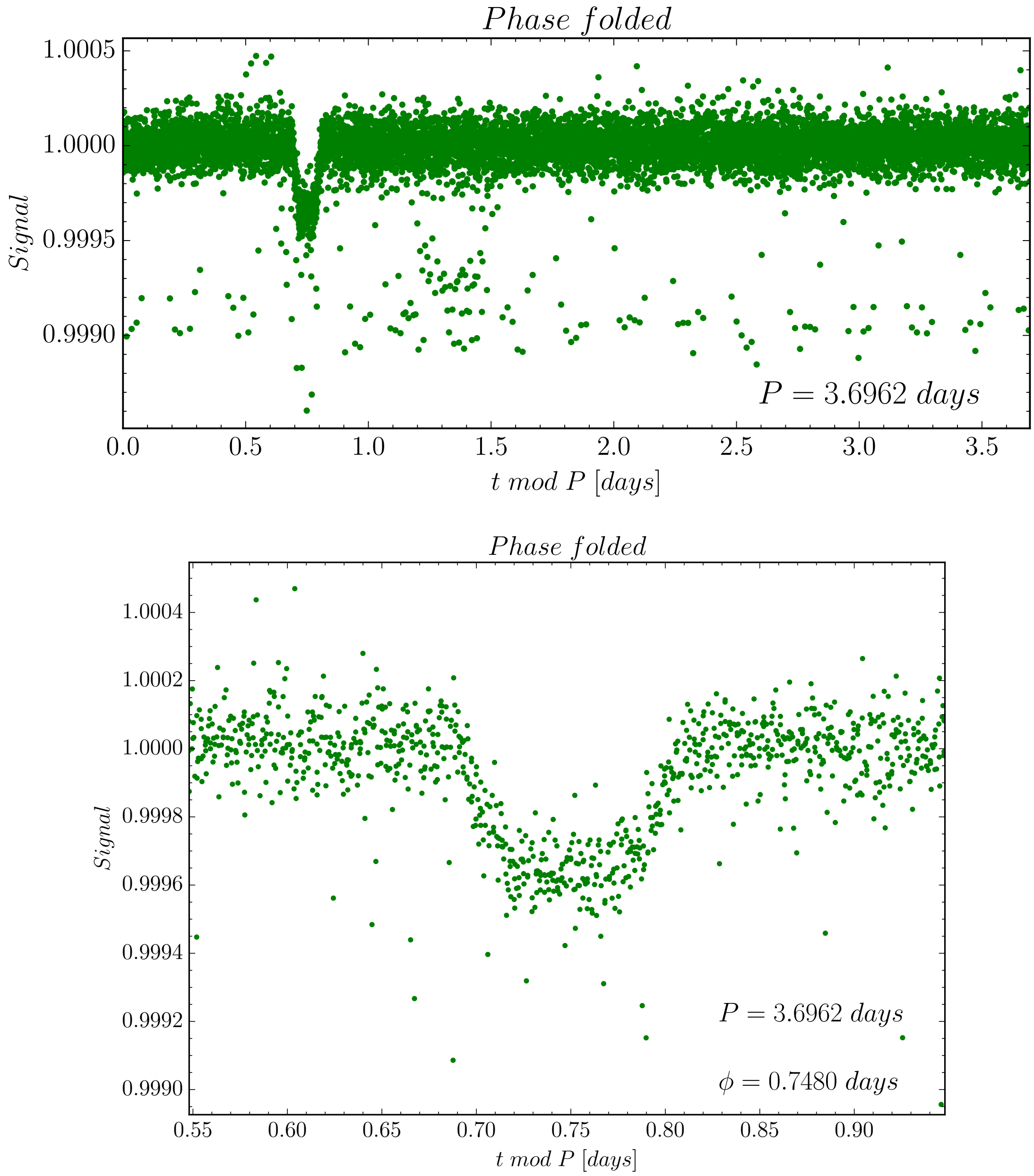
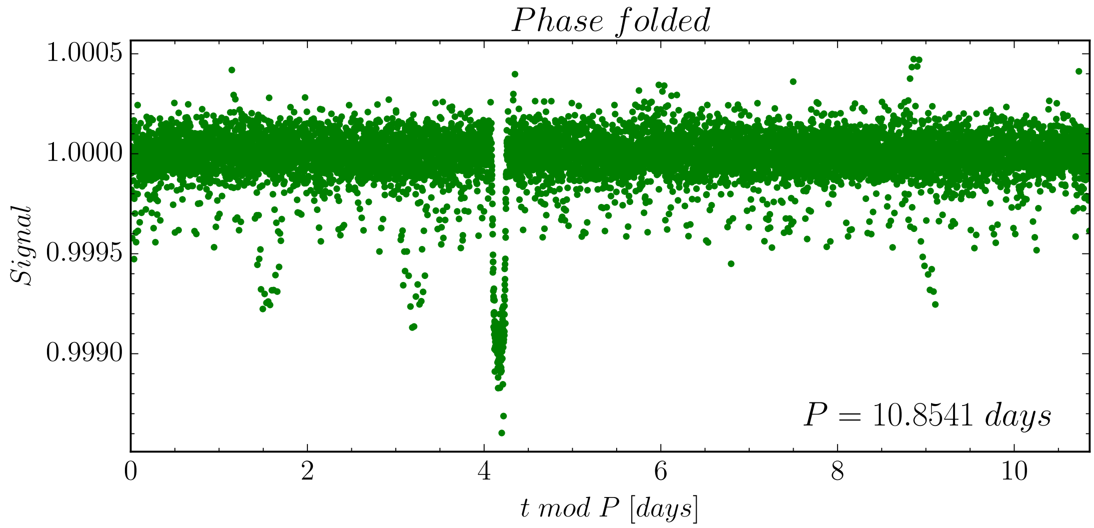


Figure 30: Phase folded diagram of with a best solution period of 3.2869 days and phase of 0.7480 days for Kepler-3b. Upper plot is the full phase folded lightcurve and the lower plot is a zoom-in on the transit using ϕ.



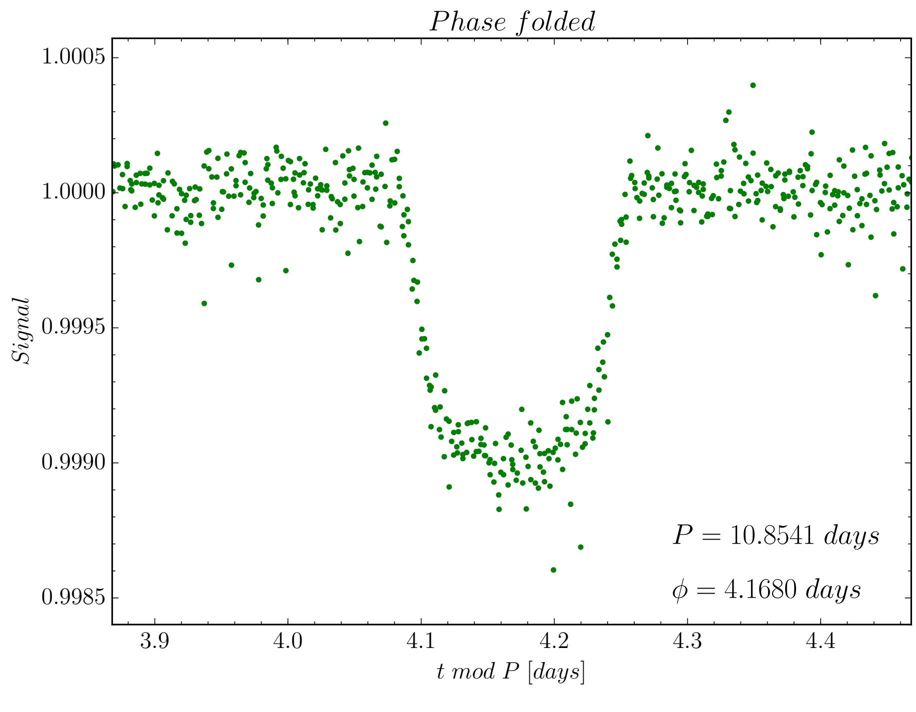


Figure 31: Phase folded diagram of with a best solution period of 10.8541 days and phase of 4.1680 days for Kepler-3c. Upper plot is the full phase folded lightcurve and the lower plot is a zoom-in on the transit using ϕ.

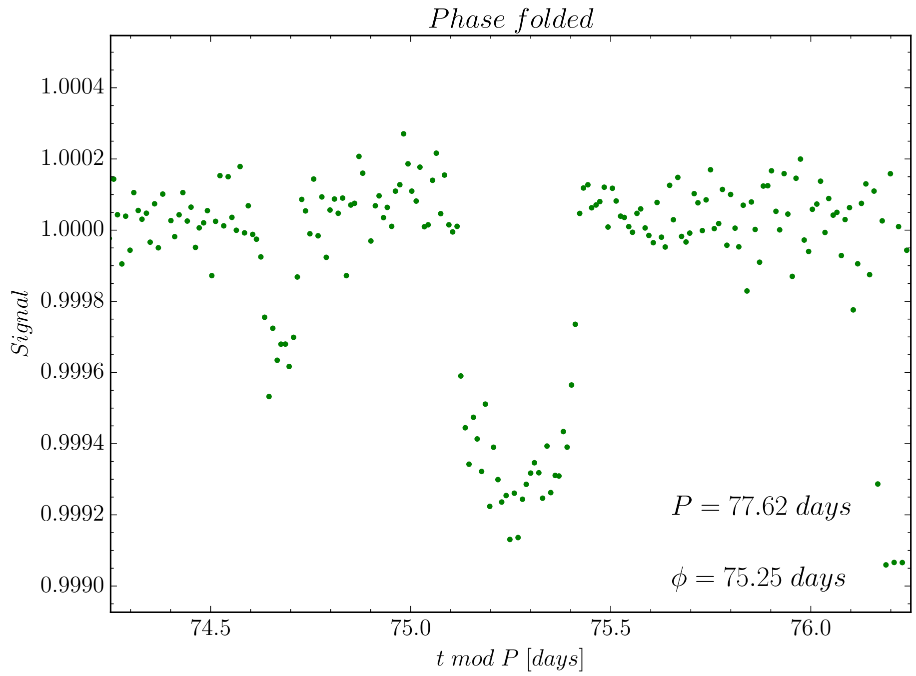


Figure 32: Phase folded diagram of with a best solution period of 72.62 days and phase of 75.25 days for Kepler-3d. The full phase folded lightcurve is messy, hence, only a zoom-in on the transit using ϕ is shown here.

3.3 Kepler 4

The Kepler-4 time series is plotted in Fig. 33 and one periodic transit signal is clearly visible. Using (1) the period of Kepler-4b is around $P = 3.16$ days. Stellar noise was not corrected for. Jump corrections can be seen in Fig. 34, and time-jump corrections were also made here in `slowtrend`. The final corrected lightcurve can be seen in Fig. 35. The CC result can be seen in Fig. 36. In `autocor` an upper ACF `cutoff_y` value of 2.8×10^{-9} was used. The ACF is shown in Fig. 37 and the best AC fit is shown in Fig. 38 and give $P_{\text{fit}} = 3.2136(98)$ days. The mean value of the ACF result is $P_{\text{mean}} = 3.2135(98)$ days. From the phase folded diagram a best solution of $P = 3.2136$ days and $\phi = 1.618$ days was found for Kepler-4b as shown in Fig. 39.

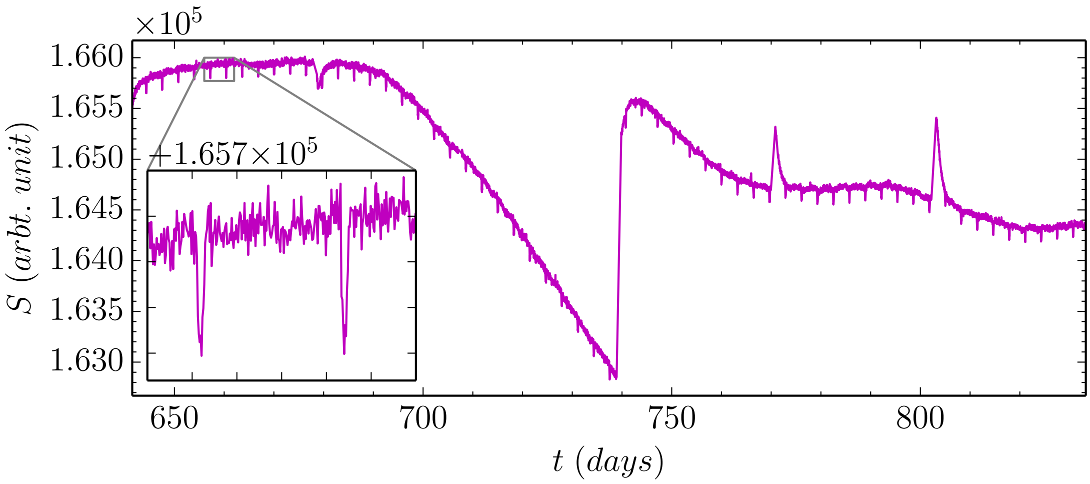


Figure 33: Time series for the 4. unknown exoplanet host star observed by Kepler. A zoom-in on a exoplanet transit is also visible.

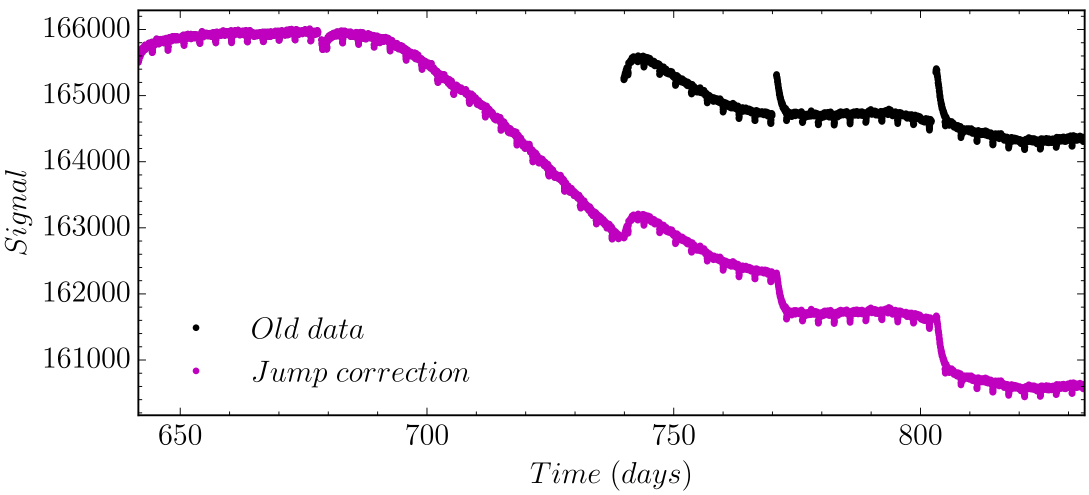


Figure 34: Time series before (•) and after (•) jumps have been corrected for.

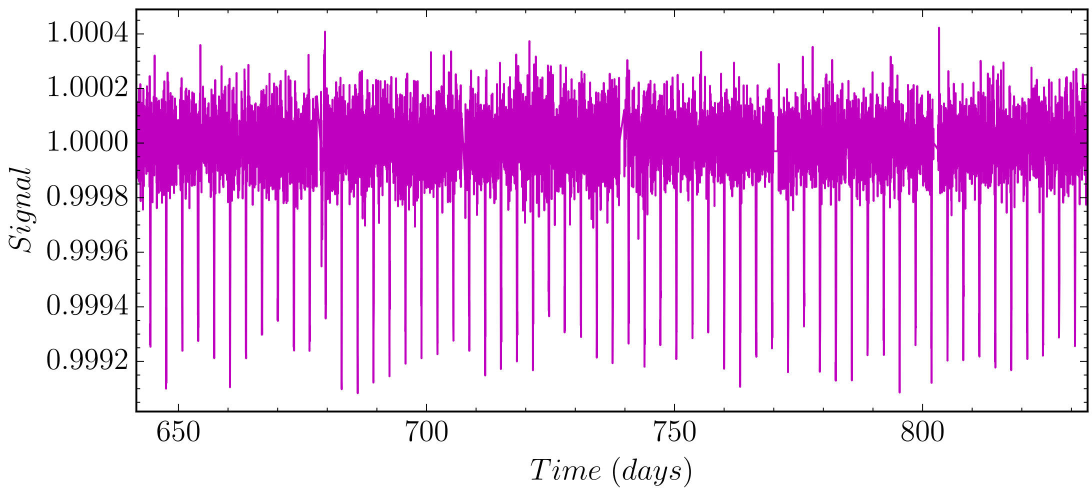


Figure 35: Final time series with clear transiting signals for Kepler-4b.

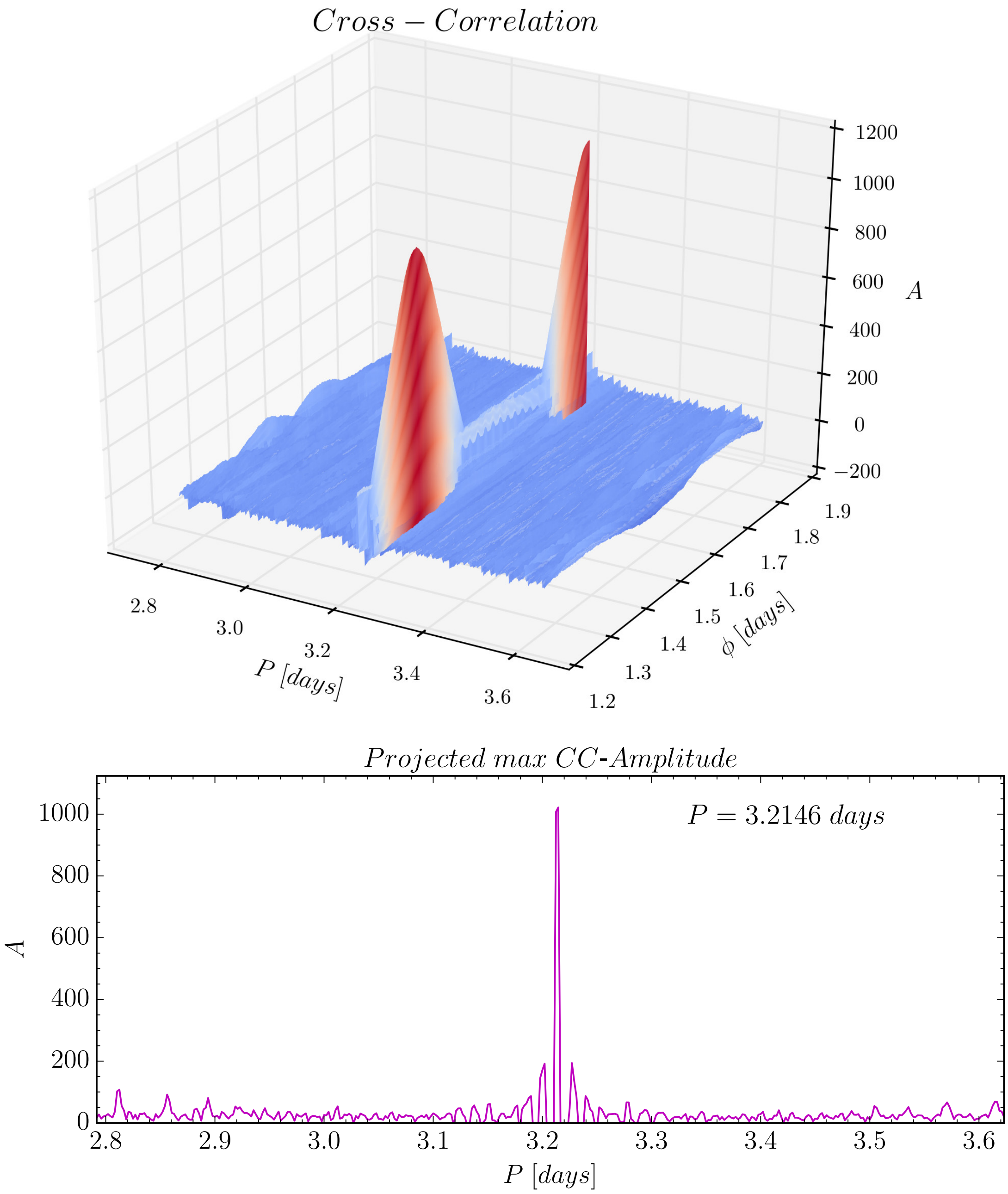


Figure 36: CC results for Kepler-4b: Upper figure shows a 3D plot the CC result. Lower plot shows projected CC amplitude of maximum power as a function of the period. From these plots $\{P, \phi\}$ = {3.2146(21), 1.3958(14)} days.

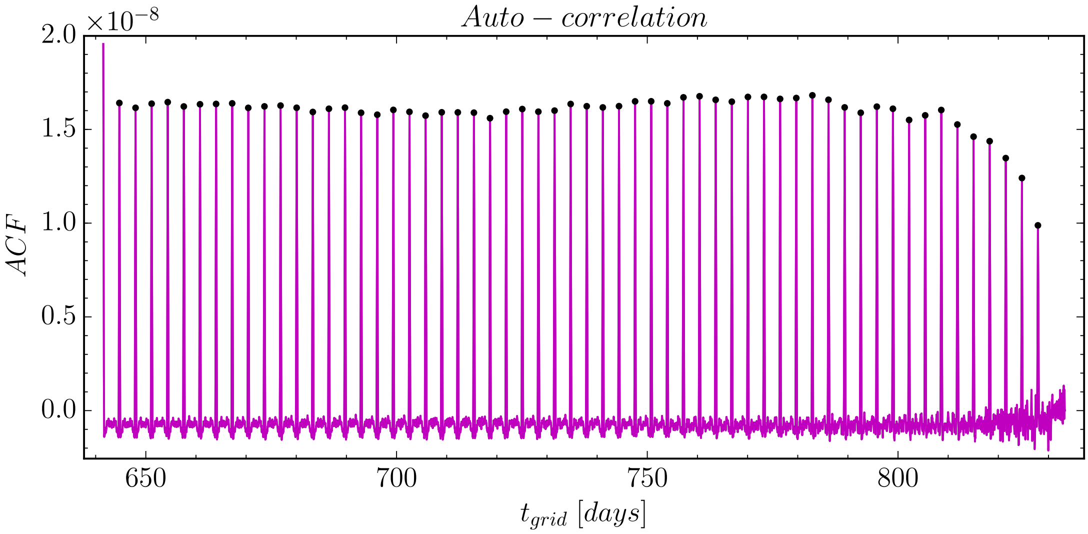


Figure 37: Auto-Correlation Function (ACF) vs. time for Kepler-4b. The black dots are peaks signals belonging to transits.

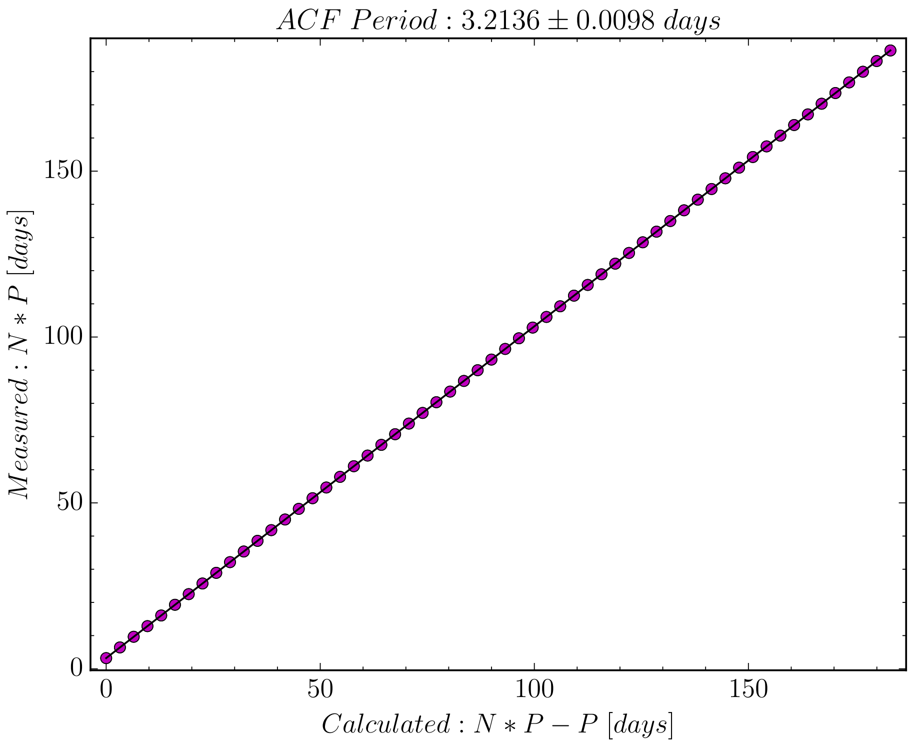
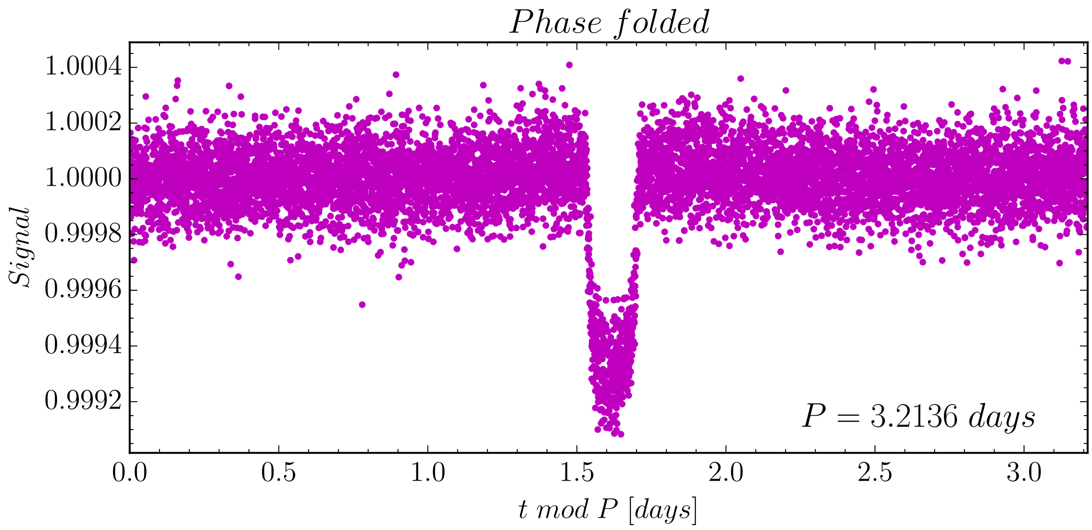


Figure 38: Linear fit of the measured and calculated values of $N \cdot P$ resulting from the auto-correlation of Kepler-4b. The b parameter from the linear regression $a \cdot x + b$ is the best fit period solution and can be seen in the title.



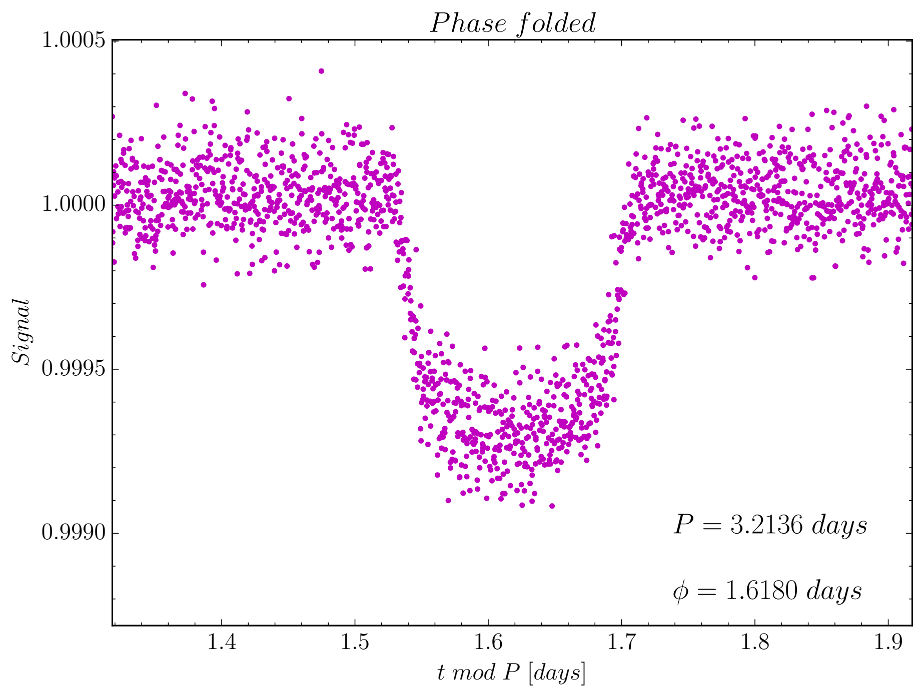


Figure 39: Phase folded diagram of with a best solution period of 3.2136 days and phase of 1.6180 days for Kepler-4b. Upper plot is the full phase folded lightcurve and the lower plot is a zoom-in on the transit using ϕ.

4 Results

A summary of all the results found in this computational exercise can be seen in Tab. 2. The results from the CC and AC seems all in all to agree well with each other. Things to improve is first of all the bug with cleaning peaks in the low frequency domain. Secondly, one can always perform the CC with a finner period and phase grid. It may be noticed in this exercise that there is a hight phase degeneracy – I do not know if this is a bug or not. Third, the AC routine do not work (yet) if (1) the found peaks are not equidistant and (2) the very first visible transit are not detected as a peak. These are thing to improve.

Table 2: Result Overview

		Lightcurve (♃)	Cross-Correlation (♄)		Auto-Correlation (♅)		Best Phase (♆)	
Star	ID	P (days)	P (days)	ϕ (days)	$P_{\rm fit}$ (days)	$P_{\rm mean}$ (days)	P (days)	ϕ (days)
Kepler-1	b	4.29	4.2875(21)	0.755(12)	4.2874(84)	4.2868(84)	4.2869	0.857
Kepler-2	b	9.29	9.2875(21)	6.1023(14)	9.290(10)	9.288(10)	9.2875	6.094
Kepler-3	b	3.63	3.6958(21)	0.7475(14)	3.6978(110)	3.6961(110)	3.6962	0.748
	c	10.26	10.8542(21)	4.1653(14)	10.8542(80)	10.8544(80)	10.8541	4.168
	d	77.6	77.6250(21)	75.2097(14)	—	—	77.62	75.26
Kepler-4	b	3.16	3.2146(21)	1.3958(14)	3.2136(98)	3.2135(98)	3.2136	1.618

♃: P is estimated from the lightcurve alone and (1).
♄: Best solutions from CC. The uncertainty is due to the used parameter grid.
♅: $P_{\rm fit}$ is best fit to linear regression between measured and calculated periods. $P_{\rm mean}$ is the mean value from the ACF. The uncertainty is a 1σ scatter of based on the $P_{\rm mean}$.
♆: P and ϕ are the best solutions when looking at a phase folded diagram.