MCMC in DAOPHOT-II

Sean K. Terry
sean.terry@berkeley.edu

This is a guide for use of the modified version of the subroutine `NSTAR.F` which implements a Markov chain Monte Carlo (MCMC) routine for fitting blended stellar positions and fluxes. This manual assumes the user has properly installed the base version of `DAOPHOT-II`[1] (Stetson 1987). Descriptions of each major section of the MCMC routine are given with example outputs. This guide includes only specific instructions for running the MCMC version of NSTAR.F. For instructions using any other `DAOPHOT-II` module, I will refer the reader to the `DAOPHOT-II` User's Manual (DII-UM) when appropriate. Lastly, please cite Terry et al. (2021) [https://iopscience.iop.org/article/10.3847/1538-3881/abcc60] if you find this code useful in your research.

[1] http://www.star.bris.ac.uk/~mbt/daophot/

Contents

1 Installation

Adding this modified routine to the `DAOPHOT-II` workflow is quite simple:

1. Move the file `nstar-mcmc.f` into the `DAOPHOT-II` source code directory.
2. Open the Makefile and replace the two instances of "nstar.o" with "nstar-mcmc.o". This will allow the user to run the command **NSTAR-MCMC** from within `DAOPHOT-II`.
3. Run *make daophot* (or *sudo make daophot*) in the source code directory. This should only take a few seconds as only the `nstar-mcmc.o` executable will need to be built.

   Note: If the user simply adds "nstar-mcmc.o" to the Makefile instead of replacing "nstar.o" with "nstar-mcmc.o", the error "`duplicate symbol in:nstar.o nstar-mcmc.o`" will likely occur.

Once again, these installation steps assume that the user has previously (successfully) installed the entire `DAOPHOT-II` software package. When using the software, both the **NSTAR** and **NSTAR-MCMC** commands will run the modified version.

2 A Typical Run of NSTAR-MCMC

Before running **NSTAR-MCMC**, the user will need to have already produced a co-added master frame and run the stock DAOPHOT-II pipeline on the image through the following steps:

- ATTACH (DII-UM pg. 12)
- FIND (DII-UM pg. 20)
- PHOT (DII-UM pg. 25)
- PICK (DII-UM pg. 29)
- PSF (DII-UM pg. 29)
- GROUP (DII-UM pg. 36)

The user will also need to manually remove all irrelevant groups of stars from the .grp file, so that only the header information and the 1, 2, or 3-star group of interest remains. Below is an example .grp file with one remaining group (the group of interest):

NL	NX	NY	Lowbad	Highbad	Thresh	AP1	PH/ADU	Rdnoise	Watch	Progress
3	1463	1463	-71.4	9700.00	18.750	5.000	40.000	12.333		1
241	896.550	822.590	15.305	24.990						
450	890.490	823.160	15.995	25.550						
(1)	(2)	(3)	(4)	(5)						

(1) Star ID number.
(2) X coordinate of stellar centroid.
(3) Y coordinate of stellar centroid.
(4) Instrumental magnitude.
(5) Estimated modal sky value for the star (from PHOT).

From inside the working directory (and inside an instance of DAOPHOT-II with the image "attached"), follow the steps:

I Run **NSTAR-MCMC**. The user will first be prompted to give the .psf and .grp files (which were generated in the previous steps). The third prompt will ask the user for a filename for the output .nst file. This output is the default file that **NSTAR** produces using the Newton-Raphson method (see DII-UM pg. 38 and sample .nst output table given in DII-UM pg. 73).

II The next prompt asks for either a 1-star, 2-star, or 3-star fit. After entering "1", "2", or "3" the degrees of freedom ($d.o.f$) for the fitting will be printed to the console. The $d.o.f$ depends on the total number of pixels in the (pre-determined) fitting box, the number of fitting parameters, and fitting constraints (if any). The fitting parameters that contribute to the overall $d.o.f$ are:

(a) 1-star fit: x, y, total flux.

(b) 2-star fit: x1, y1, x2, y2, flux ratio, total flux.

(c) 3-star fit: x1, y1, x2, y2, x3, y3, lens-source (star 1-2) flux ratio, blend-source (star 1-3) flux ratio, total flux.

So for a 2-star MCMC run over a fitting box of 700 total pixels, the total $d.o.f = 700 - 6 = 694$. Note that including constraints on any of the fitting parameters will additionally reduce the total $d.o.f$ by the number of parameters being constrained. The fitting box and degrees of freedom are discussed further in Section 4.1, and fitting constraints are discussed in Appendix A.

III The next prompt asks the user for a pixel scale in units of mas/pix. This is quite straight foward, and the routine will use this pixel scale to calculate the separation between the stars (in units of milliarcsecond). The NIRC2 and OSIRIS instruments onboard the Keck telescopes have pixel scales of 9.942 mas/pixel. The WFC3 instrument onboard Hubble Space Telescope has a pixel scale of ∼40 mas/pixel.

IV The next prompt asks the user for a "renormalization factor". This is used for re-normalizing the χ^2 per degree of freedom ($\chi^2 \simeq 1$) in order to obtain more reasonable error bars on the MCMC chains. If running the MCMC for the first time, this value should always equal '1'. After the first run, one can calculate the renormalization factor (i.e. best-fit χ^2 divided by degrees of freedom) and subsequently run the MCMC a second time while including this renormalization factor.

V Next, the user is prompted for "Number of MCMC iterations". This is straight forward. It is usually true that the larger the iteration size, the longer the routine will take to run.

VI The next prompt asks for initial guesses for the stellar positions of star 1, 2, and/or 3. The number of times this prompt repeats will of course depend on how many stars the user is attempting to fit. The initial guess needs to be within the fitting box or an error will be thrown. It is recommended that the user chooses star-1 to be the brightest, star-2 the second brightest, and star-3 the faintest. This is not a strict requirement, however the routine was originally written with the expectation that star-1 is always the brightest.

Note: If the user picks initial positions outside of the pre-determined fitting box, a "segmentation fault – invalid memory reference" error will result.

VII After the initial position guesses, the user is prompted to enter an initial guess for the source (star-1) flux contribution to the total flux. For the trivial case of 1-star fitting, a

good initial guess for this parameter is 1.0. For 2 and 3-star fitting, one can obtain an initial guess for this parameter by converting the instrumental magnitudes in the `.grp` file to fluxes and finding their individual contributions to the overall total flux. For example:

$$\frac{10^{(-0.4*m_1)}}{10^{(-0.4*m_1)} + 10^{(-0.4*m_2)}}$$

where m_1 and m_2 are the instrumental magnitudes of star-1 and star-2 respectively. If performing a 3-star fit, this prompt will be repeated for the 3rd star (also referred to as 'blend' star in the routine).

After the prompt in step VII, the routine begins. A counter will print the current step at every 50,000 iterations. Once completed, the best fit parameters are printed to the console, which includes the χ^2 value for the best fit. The following screenshot shows the terminal after a successful (2-star) run has completed:

```
Command: nstar-mcmc

           File with the psf (default image.psf):
     File with stellar groups (default image.grp):
            File for results (default image.nst):

 1, 2, or 3 star fit:
2
Degrees of Freedom = 774
 Pixel Scale (mas/pix):
9.942
 Renormalization Factor (type '1' if first time run):
1
 Number of MCMC Iterations:
100000
 Star 1 x,y:
896,822
 Star 2 x,y:
890,824
 Star 1 Flux Contribution (0.0 - 1.0):
0.88
 Fit box:         879         908         811         836
 Sky Avg:   25.2700005
 Zeropoint Mag:   14.1049995
 50000
100000
      X1              Y1              X2              Y2                SEP              F_RATIO       F_TOTAL          F1              F2              CHI2
   896.646484      822.604431      890.113708      824.308228      67.121451633453376      0.851279855      0.951911330      0.810342968      0.141568393      171.749146

    Done.
```

In addition to the best fit parameters that are printed to the terminal, two output files are generated titled "mcmc_fit.dat" and "chisq_pixel.dat". The first file includes all of the (accepted) MCMC steps in the chain. The header columns in this file are identical to the headers for the best-fit values printed to the terminal. One can use this raw output for various calculations like; using the MCMC chains as a probability distribution to estimate errors for the best-fit values, producing contour plots with confidence intervals for the best-fit parameters, etc. The second file, "chisq_pixel.dat", includes the best-fit χ^2 values for each pixel in the fitting grid. From this one could measure the χ^2/pixel value for various radii from the center of the brightest source, for example.

3 Description of Major Routines

3.1 Single Star MCMC

DAOPHOT-MCMC has functionality for fitting the position and flux for the trivial case of a single star. The star, of course, could actually be a binary or have a very close companion such that the separation is significantly smaller than some fraction of the FWHM, which would show no significant signal detected in the residual. For the purposes of this section, we will assume the star is indeed a single stellar object.

The MCMC routine begins at a pixel location chosen by the user. The size of the fitting box is ∼1.5x the FWHM of the PSF in both directions and is pre-determined by `DAOPHOT-II` before the iteration begins. The iteration proceeds by taking random steps in any direction and measuring the total flux (f_T) distribution:

$$f_T = f_1\psi(i - x_1, j - y_1),$$

where f_1 is the fractional flux contribution from the single star (1.0 for a single-star fit), ψ is the two-dimensional empirical PSF, x_1 and y_1 are the initial pixel coordinates. A χ^2 value is also computed at each step in the chain. Once the routine is finished, the best-fit parameters are printed to the terminal screen (i.e. X1, Y1, Total Flux, χ^2). For the ideal case of 100% of the flux in the fitting box coming from the star (minus background), the best-fit total flux value would be 1.00. However, due to photon noise, imperfect PSF shape, and other sources of flux contamination (like an unresolved blend as mentioned earlier), this value may deviate from unity. The total flux calculation is rather simple; it is just the ratio of the total sum of each raw pixel value (minus background) and the total sum of the measured flux distribution at each pixel across the grid:

$$\frac{\sum_{i,j}^{n} P_{i,j} - s_{i,j}}{\sum_{i,j}^{n} f_1\psi(i - x_1, j - y_1)},$$

where P is the raw pixel value and s is the background. Finally, the χ^2 minimization routine used for the fitting is given in section 4.5.

3.2 Dual Star MCMC

DAOPHOT-MCMC can simultaneously fit the positions, separations, and fluxes for a two-star blend. Similar to the single-star routine, the iteration begins at a user-defined pixel location (for both stars) inside the fitting box of ∼1.5x FWHM, as well as an initial guess for the flux ratio of the stars. The iteration proceeds by taking random steps in any direction and any flux ratio. This flux ratio is an additional parameter to consider for the two stellar objects. The routine measures the total flux distribution at each step in the chain:

$$f_T = f_1\psi(i - x_1, j - y_1) + (1 - f_1)\psi(i - x_2, j - y_2),$$

with the additional parameters as follows; $(1 - f_1)$ is the flux contribution from the second (fainter) star, x_2 and y_2 are the initial pixel positions for the second (fainter) star. Similar to the single-star fit, a χ^2 value is calculated at each step, and the best-fit parameters are printed to the screen once the routine has finished (i.e. X1, Y1, X2, Y2, Separation, Flux Ratio, Total Flux, χ^2). As mentioned in the previous section, the best-fit total flux may deviate from unity. The total flux is calculated similarly to the previous section:

$$\frac{\sum_{i,j}^{n} P_{i,j} - s_{i,j}}{\sum_{i,j}^{n} f_1\psi(i - x_1, j - y_1) + (1 - f_1)\psi(i - x_2, j - y_2)},$$

where the additional parameters from the second (fainter) star are now included.

The separation between star-1 and star-2 is measured at each step in the chain, the calculation is very straightforward:

$$\Omega\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2},$$

where Ω is the pixel scale (mas/pixel). For the NIRC2 or OSIRIS instruments using adaptive optics (AO) on-board the Keck 10m telescopes, $\Omega = 9.942$ mas/pixel.

3.3 Triple Star MCMC

The third major function in DAOPHOT-MCMC is simultaneous fitting of three stars. The fitting will work on three blended stars, two blended stars and a nearby neighbor, or three nearby (non-blended) stars. Of course, factors like the star separations and flux ratios will influence the properties of the MCMC runs. Similar to the dual-star fitting scenario, if the stars have very small separations and similar fluxes (or large separation and very different fluxes), some difficulty may arise when attempting to accurately fit their positions.

Just like before, the iteration begins at a user-defined pixel location (for all three stars) and initial guess for the flux ratios. The routine takes random steps in any direction and flux ratios (star1-2 and star1-3 flux ratios). The measurement of the total flux distribution at each step is:

$$f_T = f_1\psi(i - x_1, j - y_1) + (1 - f_1 - f_3)\psi(i - x_2, j - y_2) + f_3\psi(i - x_3, j - y_3),$$

where f_1 is the flux contribution from star 1, $(1 - f_1 - f_3)$ is the flux contribution from star 2, f_3 is the flux contribution from star 3, and x_3, y_3 are initial pixel values for star 3. Similar to the previous major routines, a χ^2 minimization routine is performed (i.e. section **??**) and the best-fit parameters are printed to the screen when the process has finished.

The total flux for the three-star fitting is calculated as follows:

$$\frac{\sum_{i,j}^{n} P_{i,j} - s_{i,j}}{\sum_{i,j}^{n} f_1\psi(i - x_1, j - y_1) + (1 - f_1 - f_3)\psi(i - x_2, j - y_2) + f_3\psi(i - x_3, j - y_3)},$$

where the new factor in the denominator comes from the flux distribution of the third star. The new parameters that are calculated for three star fitting are: star 3's x,y position and flux contribution, and star 1-3 separation.

4 Description of Internal Functions

4.1 Degrees of Freedom

The degree of freedom ($d.o.f$) is calculated based on the following parameters:

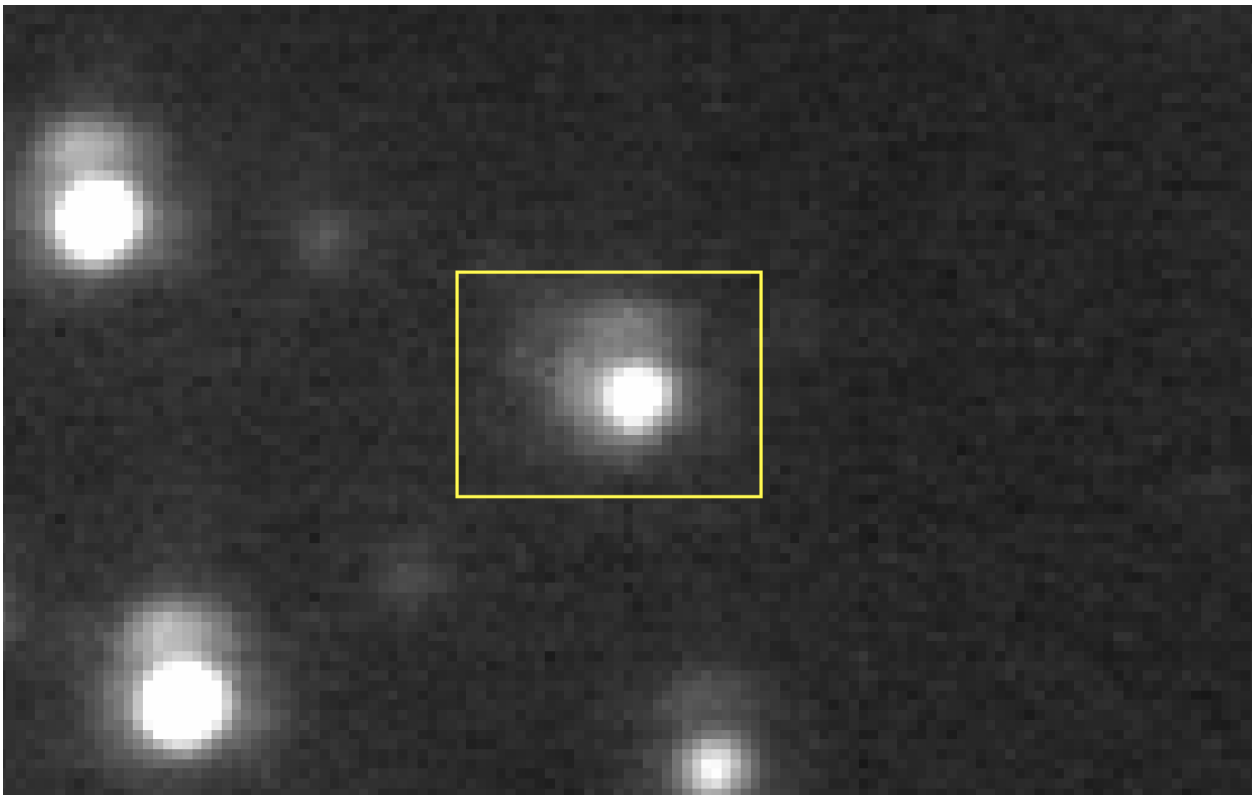
- Number of pixels
- Number of fitting parameters
- Number of fitting constraints

As described in Section 2, for a single star fit with zero constraints, the $d.o.f$ = total pixels + 3. For a two-star fit this increases by 3 (total of 6 fitting parameters), and the $d.o.f$ increases by 3 again for a three-star fit (total of 9 fitting parameters). The two main fitting constraints that are written into the program currently are constraints on the total flux and separation between the stars. These constraints are discussed in Appendix A.

NSTAR will automatically compute the size of the fitting box based on 1.5x the FWHM of the PSF for each star in the group. The fitting box will usually be a square grid of pixels, but can sometimes be rectangular depending on the separation amplitude and direction for two or three fitted stars in the box. The shape of the fitting box (square or rectangle) will not significantly impact any part of the MCMC or analysis thereafter. The total number of pixels in the fitting box can vary quite significantly (between a few hundred and few thousand) depending on the size of the PSF, the number of stars in the group and their separations. Finally, before the MCMC routine begins, the size of the fitting box is printed to the screen by: min_X, max_X, min_Y, max_Y. See screenshot below:

```
 1, 2, or 3 star fit:
2
Degrees of Freedom = 774
 Pixel Scale (mas/pix):
9.942
 Renormalization Factor (type '1' if first time run):
1
 Number of MCMC Iterations:
10000
 Star 1 x,y:
896,822
 Star 2 x,y:
890,824
 Star 1 Flux Contribution (0.0 - 1.0):
0.88
 Fit box:         879         908         811         836
 Sky Avg:   25.2700005
 Zeropoint Mag:   14.1049995
```

The following screenshot shows the target and an outline of the fitting box. The target consists of a blended source (brighter) star and lens (fainter) star for the mircolensing target MOA-2009-BLG-319.



There are a total of 780 pixels inside the fitting box.

4.2 Separation

The separation between the stars (in two-star or three-star fit) is *not* considered a fitting parameter. It is simply the calculation that was presented earlier in Section 3.2:

$$\Omega\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2},$$

where Ω is the user-defined pixel scale (in mas/pix) described earlier. For each step in the MCMC chain, the separation is calculated using the central positions of each star in the group. The positions are denoted $x_1, y_1, x_2, y_2, x_3, y_3$, and the calculated separations are denoted "S1-2_SEP" for the separation between star '1' and star '2', and "S1-3_SEP" for the separation between star '1' and star '3'. With this information, it is trivial to calculate the separation between star '2' and star '3'. As mentioned earlier, the general guideline is to consider star '1' as the brightest, star '2' as the intermediate, and star '3' as the faintest in the group (for a three-star fitting).

4.3 Flux Ratio

The flux ratio fitting parameter describes the ratio between the best-fit flux value for each star and the best-fit total blended flux). The user will need to give an initial guess for the flux ratio for 'star 1' (i.e. brighter), and for the three-star fit will also need to supply an initial guess for 'star 3' flux ratio (i.e. faintest). As described earlier in Section 3.2, a first guess for the flux ratio's can be determined from the `.grp` file that provides the aperture photometry for the stars in the group. There is also a *Python* script in the `/Example` directory that will quickly calculate the flux ratio's based on the magnitudes from the `.grp` file.

The best-fit flux ratio will be printed to the screen when the routine finishes. For a one-star fit, this value should be very close to 1.0. For a three-star fit, the best-fit flux ratio between star '1' and star '2' will is denoted as 'S1-2_FRATIO' and the best-fit flux ratio between star '1' and star '3' will be denoted as 'S1-3_FRATIO'.

4.4 Total Flux

The best-fit total flux is the final major fitting parameter. The calculation was presented in Section's 3.1, 3.2, and 3.3 and is used in all three fitting runs. Theoretically, the total flux for the single-star fit should always equal 1.0 so long as the target is truly a single stellar source. As stated earlier, the total flux may deviate from unity for several reasons related to several sources of error. Because of variations in the PSF from image to image, drawing the uncertainty in total flux from the MCMC distribution will likely result in underestimated error bars. The Jackknife method (presented in Bhattacharya et al. 2020 and Terry et al. 2020) can account for these PSF variations and is likely a more reasonable method for estimating the errors on the total flux.

4.5 χ^2 Minimization

DAOPHOT-MCMC employs the Metropolis-Hastings (MH) algorithm on the 3 , 6 , or 9-dimensional parameter space for the one-star, two-star, or three-star scenarios respectively. For each step in the chain, the following is computed:

$$\chi^2 = \sum_{i,j} [\frac{1}{\sigma}\{P_{i,j} - s_* - f_1\psi(i - x_1, j - y_1) - (1 - f_1)\psi(i - x_2, j - y_2)\}]^2,$$

where $P_{i,j}$ is the intensity at pixel i, j, σ is the uncertainty in pixel intensity, and s_* is the background flux. The procedure for minimizing this value follows the MH method, and the best-fit minimum χ^2 is then printed to screen when the run has finished.

4.5.1 Renormalization

The renormalization factor is implemented in order to obtain more reasonable error bars on the MCMC chains. This is achieved by dividing the 'un-renormalized' best-fit minimum χ^2 by the $d.o.f$ from the initial run of NSTAR-MCMC, which results in $\chi^2/d.o.f \sim 1.0$. I'll note that this is a standard technique in the microlensing field; I am not specifically using a reduce χ^2 to distinguish models. I use the difference between the χ^2 values, and only use the renormalization to scale that difference.

While it has been argued that including the errorbar scaling factor as a model parameter while fitting the model to the data may be a better approach, there are some potentially significant drawbacks. First, the idea of scaling the error bars while fitting could erroneously increase the chances of overfitting. The scaling factor for one data set can be driven very large and hide the fact that a given data set doesn't fit the model. Second, including the errorbar scaling as a model parameter while fitting may require one to define a prior on the proposal distribution $P(x)$ in order to allow the MCMC to converge. In this case it would likely be more beneficial to build a new likelihood, and use an uninformative prior. A future update to the code may include a different approach to recaling the errorbars, however the author does not expect any change to significantly impact the MCMC results.

A Fitting Constraints

There are currently two main parameters that can be constrained; the total flux and the separation. At present, the only way to implement these constraints is by manually editing the source code. There are several lines inside `NSTAR-MCMC.F` that are commented to show where the expression is for implementing the constraint(s). These constraints having the following forms.

For constraining the flux:

$$\chi^2 = \sum_{i,j} \{(f_1\psi(i-x_1, j-y_1) - (1-f_1)\psi(i-x_2, j-y_2) - F_C)/\sigma_{F_C}\}^2$$

where F_C and σ_{F_C} are the flux constraint and uncertainty on the flux constraint, respectively.

Similarly, for constraining the separation:

$$\chi^2 = \sum_{i,j} \{(\Omega\sqrt{(x_1-x_2)^2 + (y_1-y_2)^2} - S_C)/\sigma_{S_C}\}^2$$

where S_C and σ_{S_C} are the separation constraint and uncertainty on the separation constraint, respectively. As mentioned earlier, Ω is the user-defined pixel scale (in mas).

A future update to the code will include user-defined inputs on the terminal screen along with the other initial parameters for the MCMC. This will mitigate hard-coding the constraints inside the source code itself.

B Example Outputs / Figures / Tables

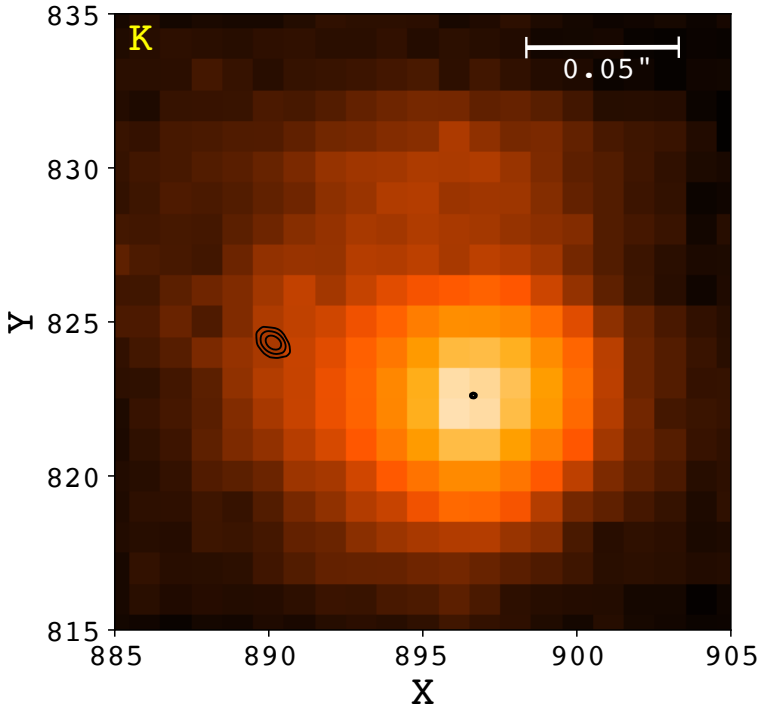
The following outputs are from the DAOPHOT-MCMC analysis of the microlensing target MOA-2009-BLG-319 (Terry et al. 2020).

The following table shows the first several lines from the "mcmc_fit.dat" output file:

X1	Y1	X2	Y2	SEP	F_RATIO	F_TOTAL	F1	F2	CHISQ
896.000000	822.000000	890.000000	824.000000	62.8787308	0.880000055	0.949538827	0.835594237	0.113944605	476.541962
896.178772	821.930115	889.810486	823.843506	66.1095428	0.886963248	0.949477732	0.842151880	0.107325882	453.851593
896.145813	821.969910	889.947937	824.038330	64.9601669	0.893375695	0.949382246	0.848155022	0.101227224	440.780151
896.152771	821.977234	890.033081	824.092590	64.3742294	0.895395815	0.949335396	0.850030959	9.93044525E-02	435.635895
896.248108	822.114014	890.158264	824.058105	63.5555038	0.901857197	0.949421048	0.856242180	9.31788459E-02	351.118225
896.308533	822.161804	890.163208	824.139771	64.1835709	0.902112067	0.949595094	0.856641173	9.29538980E-02	317.609924

.
.
.

Using the "2star-contour.py" Python script (https://github.com/skterry/DAOPHOT-MCMC/tree/master/Example/Output/2star-contour.py) to plot the contour intervals for both stars on top of the stellar image gives the follow figure:

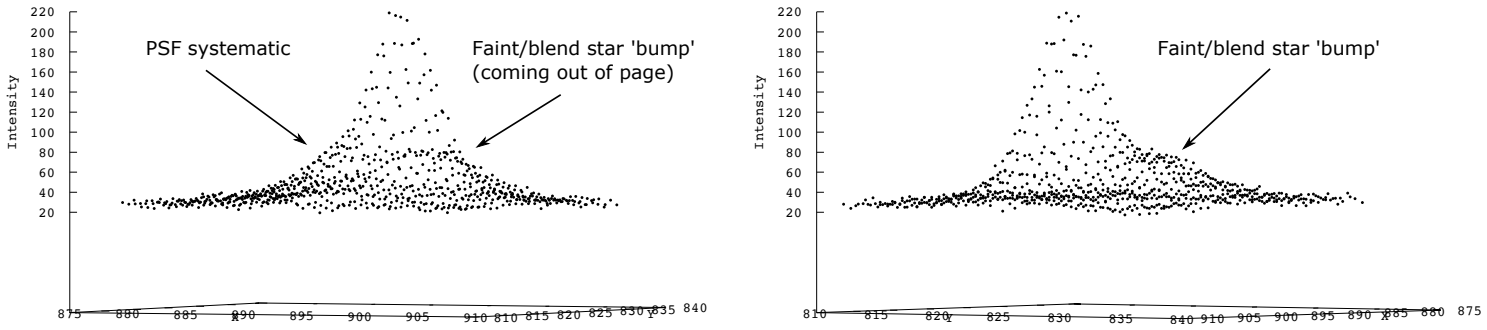


Caption: Best fit MCMC contours (68.3%, 99.5%, 99.7%) for the two blended stellar positions, over-plotted on the K band image of the target. The fainter star contributes ∼15% of flux to the total blend. Figure from Terry et al. 2020 (https://arxiv.org/abs/2009.08461).

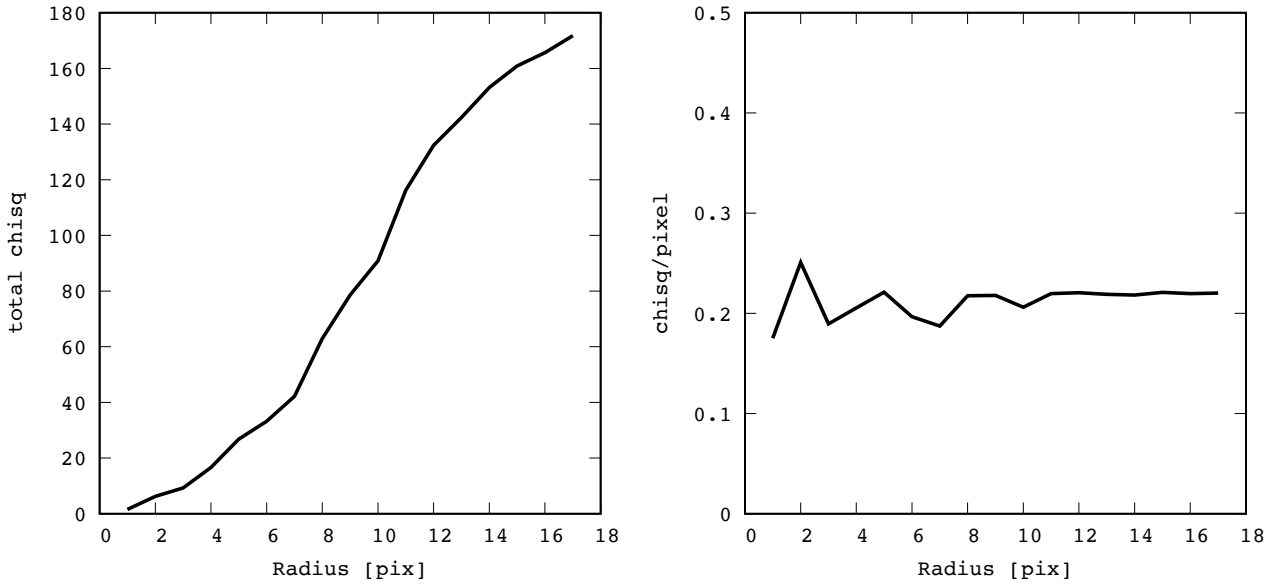
The second file that is printed out, "chisq_pixel.dat":

X	Y	CHISQ	PIX_INTENSITY
879	811	0.343843967	29.9540348
879	812	4.09128983E-03	27.9113522
879	813	2.59341728E-02	27.8329487
879	814	0.226609126	24.6133747
879	815	0.639254093	23.6851978
879	816	1.21737766	22.3830376
.			
.			
.			

This table is very straightforward, it can be used to plot the variation in pixel intensities across the grid, or a diagnostic test for the stability of the PSF model as fitted to the stellar position(s).



Caption: 3-D plots of pixel intensities across the grid. Panels are rotated (45° about z-axis) to show a systematic feature from the PSF model, and a real feature from the blended faint star.



Caption: Diagnostic plots for chisq as a function of pixel radius (i.e. pixels from peak centroid).

5 Change Log

- **2020 November 9**: Add appendix B to user manual, add headers to "chisq_pixel.dat" output.
- **2020 October 22**: `v1.3` release.
- **2020 October 16**: Bug fixes from ref. report (c.f. Terry et al. 2021).
- **2020 July 16**: Added separation and flux constraint (commented out by default).
- **2020 July 7**: χ^2 renormalization implemented (`v1.2` release).
- **2020 July 2**: Fix dimension size to match pre-determined gridsize via `ALLOCATABLE`.
- **2020 June 30**: Added column headers to raw MCMC output file.
- **2020 June 26**: Fix bug printing star(1-2) separation and star(1-3) separation for triple-star MCMC.
- **2020 June 10**: Cleaner printing of best-fit parameters to the terminal, minor bug fix in degree of freedom ($d.o.f$) calculation (`v1.1` release).
- **2020 May 30**: Added functionality for 3-star fitting, minor bug fixes.
- **2020 May 15**: `v1.0` release.