# Automatic Light Curve Fitting for MANOS

March 12, 2014

## 1 Introduction

This document is intended to be an introduction to the manosCurveFit.py program used for automatic light curve fitting, part of the automation software for the Mission-Accessible Near-Earth Object Survey (MANOS). The program is designed so that minimal user input is necessary to run, but additional inputs are available for customization of the processing procedures and output forms. The backbone of the fitting method is a nonlinear least squares minimization using the Fourier fitting method outlined in Harris et al. 1989.

## 2 Quick Start Guide

In order to process the light curve data, the data files must be set up appropriately. After this initial setup, the command line may then be used to set fitting and plotting options and initiate data processing.

### 2.1 File Setup

By default, manosCurveFit.py assumes that it is in a directory that is at the same level as a directory called 'Data,' which contains subdirectories named after each asteroid with data.

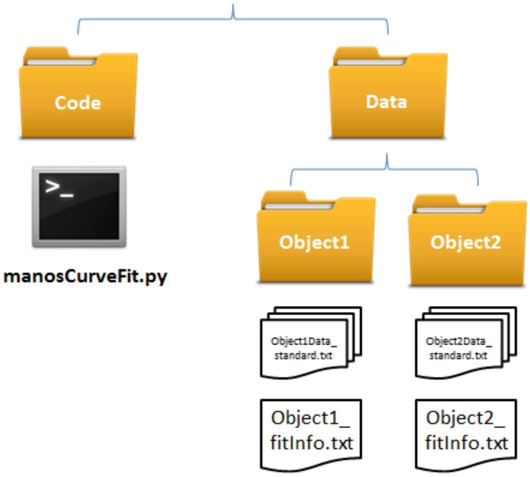

Figure 1: Default assumed file structure for manosCurveFit.py. Outputs will be saved in the objects' folders (Object1 and Object2 here).

Raw light curve data must be placed in space-delimited .txt files in each of these named directories, with the first column being Julian Date, the second being magnitude, the third being magnitude error, and an optional fourth column for the (numbered) night the data was collected. The nights column must be included if more than one night's data is used. These data do not need to be sorted, and multiple .txt files may be included, though all such files must have the word 'standard' in the file name (i.e. 'dataFileName_standard.txt').

A fitInfo file may also be optionally included in the named directory, according to the convention 'objectName_fitInfo.txt.' This file may use keywords to specify the number of data files to be processed (as a check against the actual number of data files in the directory), the period guess method and guess parameters, and any magnitude offsets associated with a particular night. Comments may be included with lines that begin with a '#' symbol. See the fitInfo specification section for the full list of keywords and keyword arguments.

### 2.2 Command Line

After starting manosCurveFit.py, the command line prompt will be displayed. In the simplest (and likely most common) case, the only command that needs to be used is `fitAll` (without any arguments), which will automatically scan all subdirectories in the Data directory and fit all objects that do not already have a light curve plot in their folders. If a only a specific object's data need to be fit, or if something that was already processed needed to be redone for any reason, the `fit` command can be used, with one or more object names as arguments. Other commands exist for access to more detailed options and other functions. Type `help` or `?` to get a list of commands, and `help <command>` or `? <command>` for a more detailed explanation of each command.

## 3 Software Details

### 3.1 System Dependencies

This software was developed and tested on Python 2.7.1, and imports from the following typically pre-installed packages: *operator, os, time, sys*, *string*, and *cmd* and the following typically non-pre-installed packages: *lmfit, matplotlb, numpy* and *uncertainties.*

The latter set of packages are commonly used for scientific applications and stable builds should be easily found.

## 3.2 System Overview

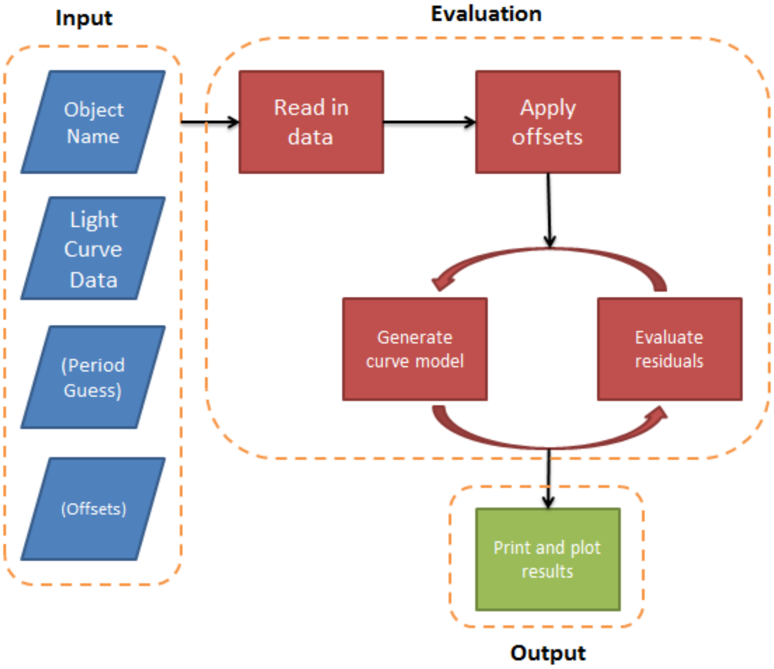

Figure 2: Flowchart showing the conversion from raw data to fitted plots. The dashed boxes represent the three main parts of the software.

The fitting routine may be generally divided into input, evaluation, and output portions. The inputs in parentheses are not required for the fitting to work, but a guess at the rotational period may significantly constrain the search space, and if data is taken over multiple nights and/or by different instruments, offsets will be required to normalize the magnitudes to some baseline. A number of other inputs may also be supplied to constrain or expand the search space. These inputs may be supplied by the fitInfo file for each object. The light curve data is then read in from one or more text files with standardized columns. Two kinds of magnitude offsets are then applied: offsets by night/instrument, if provided, and a normalizing offset which subtracts the weighted average magnitude of the set from the entire set. The latter offset is necessary in order to center the data points on the y-axis for fitting purposes, and may be used in this context because only differential magnitudes are needed. This acts as a coarse adjustment for y-axis centering. A finer adjustment is made by a y-axis offset for the model, which is taken to be part of the least squares minimization. For each period to be checked, a least squares minimization is performed for each order of Fourier coefficients from two to six, unless specified otherwise in the fitInfo file. The parameters that generate the best fit to the data are kept and printed at the end, along with the light curve and residual plots.

## 3.3 Fitting Rationale

This fitting routine is based on equations 1, 2, and 3 from Harris et al. 1989 (equations 1, 3, and 4, below), where the model is based on the Fourier Series

$$H(\alpha,t) = \bar{H}(\alpha) + \sum_{L=1}^{m} A_L sin\frac{2\pi L}{P}(t-t_0) + B_L cos\frac{2\pi L}{P}(t-t_0), \qquad (1)$$

where $\bar{H} = 0$ because absolute magnitudes are not necessary for MANOS, and $m$ (the series order), $P$ (the period), and $A_L$ and $B_L$ (the Fourier coefficients) are free parameters. Since a fine-adjustment y-shift is also added, the actual fitting equation becomes

$$H(\alpha,t) = y + \sum_{L=1}^{m} A_L sin\frac{2\pi L}{P}(t-t_0) + B_L cos\frac{2\pi L}{P}(t-t_0), \tag{2}$$

where $y$ is an additional free parameter, which generally takes on a small value.The residual of a particular observation $i$ may be obtained by

$$\frac{\delta_i}{\epsilon_i} = \frac{V_i(\alpha_j) - H(\alpha_j, t_i)}{\epsilon_i}, \tag{3}$$

where $\alpha_j$ is the reference phase angle on the $j^{th}$ night, $t_i$ is the time of the $i^{th}$ observation, and $\epsilon_i$ is the error of the measurement. In the context of NEOs, the phase angle may very well change, particularly as targets pass very close to Earth. However, the majority of MANOS targets will be observed for a short enough period of time that $\alpha$ will be assumed to be a constant. As such, manosCurveFit does not take phase angles into account. The least squares minimization is then performed on the bias-corrected variance, given by

$$s^2 = \frac{1}{n-k}\sum_{i=1}^{n}\left(\frac{\delta_i}{\epsilon_i}\right)^2 = minimum, \tag{4}$$

where $n$ is the total number of observations, $k = 2m + 1$, where $m$ is defined in 1. The total number of nights of data is also added into $k$ in the form that Harris et al. uses, but here, this again needs not be considered because we are concerned with differential photometry, and offsets for different nights will be provided as necessary.

By default, the program will run the fit from $m = 2$ to $m = 6$. The minimum of order two is due to the fact that asteroid light curves are expected to be double-peaked (except for some cases of unstable rotation or degenerate cases of near-spherical bodies), and the maximum of order six is used to prevent over-fitting. The curve is centered around zero magnitude by a weighed average of the data, but since there is often still a slight magnitude offset due to the nonuniform nature of the sampling, a magnitude offset parameter was added to allow for a better fit. The user does not normally interact with the optimization of this y-shift parameter.

Additionally, any models which produce amplitudes greater than 2 will be rejected. This is necessary to prevent the fit from assuming a model in which the data are a small portion of a much longer period with one or more large spikes where data is not present. As an additional precaution, if the fitted period is more than 25% of the range of the phase-folded data points, a warning is given to notify the user of a potentially unsubstantiated model.

# 4 Input, Evaluation, and Output Methods

The software handles data input by reading text files and storing user-defined data columns as numpy arrays in a *lightCurveData* object. Evaluation is handled by the *fitData* function, which utilizes *lmfit*'s minimization routine with free parameters given as Parameter object inputs. Output is handled by the *outputResults* function, which has options to display results in various ways.

## 4.1 fitInfo Specification

| Keyword | Arguments | Meaning |
|---|---|---|
| FILES | integer | number of data files for this object (used as a check) |
| GUESS | string, then 1 or 3 integers | see Guess Specifications section, below |
| HARDMAXPERIOD | float or int | hard maximum period to not search above |
| HARDMINPERIOD | float or int | hard minimum period to not search below |
| OFFSET | string | starts a series of night/offset pairs used by the string specifying the dataset |
| ENDOFFSETS | N/A | ends the series of offsets (required if OFFSETS are used) |

Table 1: fitInfo keywords (all keywords are optional)

**Example fitInfo File:**

```
FILES 2
# method min max step
GUESS range 14 18 0.25
HARDMAXPERIOD 13
HARDMINPERIOD 20
OFFSET Elisa\elisa_mine_standard.txt
1 0.0
2 -0.04
3 0.464
ENDOFFSETS
OFFSET Elisa\elisa_his_standard.txt
1 -0.324
2 -0.257
3 -0.237
4 -0.194
5 -0.223
6 -0.321
7 -0.246
8 -0.372
9 -0.15
ENDOFFSETS
```

## 4.2 Guess Specification

Three different ways to specify initial guesses at the period value (in hours) exist.

- range (3 floats or ints) - a range of guesses will be used, following the convention min, max, and step size
  - Example: `GUESS RANGE 0.1 5.5 0.25`
- single (1 float or int) - one initial guess will be used
  - Example: `GUESS SINGLE 2`
- None - if the GUESS line is excluded, an interval from 15 minutes to 5 times the observing window will be used (see section on fitData)

## 4.3 Command Line Interface

Descriptions of all command line options

- `exit` (no arguments) - exits the program
- `fit` (names of objects to be fit, separated by spaces- must match folder names in the Data directory) - runs a fit on the objects specified, regardless of whether or not they have already been processed
  - Example: `fit Martes Elisa`
- `fitAll` (no arguments, or `'redo'`) - runs fits on all objects in the Data directory that do not have an existing light curve plot; if the `'redo'` argument is provided, all plots are fitted, regardless of any existing fits
- `setFitOptions` (option and value arguments) - sets options used in the fitting routine; multiple options may be set at once
  - `minOrder` (non-negative integer argument) - sets the minimum order to be used in the Fourier fit; default is 2

  - `maxOrder` (non-negative integer argument greater than minOrder) - sets the maximum order to be used in the Fourier fit; default is 6
  - `timer` (boolean) - turns a fitting timer on or off, which measures the amount of time required for each fit, generally for diagnostic purposes
    - Example: `setFitOptions minOrder 3 maxOrder 5 timer true`

- `setOutputOptions` (option and boolean setting arguments) - sets options used in the program output; multiple options may be set at once; all options take booleans
  - `printReport` - whether or not to print the fitting report on the console (default is True)
  - `saveReport` - whether or not to save the fitting report to the object's directory (default is True)
  - `plotFullPeriod` - whether or not to plot the full period as determined by the model; if not, the model will only plot up to the available data (default is True)
  - `plotErrorBars` - whether or not to plot the error bars on the data (default is True)
  - `phaseFoldData` - whether or not to phase fold the data and model (default is True)
  - `plotResiduals` - whether or not to plot the residuals of the data as a subplot of the light curve (default is True)
  - `plotPeriodErrors` - whether or not to plot the mean RMS values the errors as a function of the period attempted (default is True)
  - `showPlots` - whether or not the show the plots (default is False); the plots will always be saved to the object's directory

- `showObjects` (no arguments) - lists the object subdirectories found under Data

# 5 Class and Function Specifications

## 5.1 The `lightCurveData` Class

**class** `lightCurveData`(objectName, fileNamesAndFormat[, offsetsList = None])
Creates a lightCurveData object which is used to read in and manipulate the dataset.
**Parameters**

- **objectName** (string) - name of the object associated with the dataset
  - Example: 'Spartacus20090130'
  - Stored in lightCurveData.name

- **fileNamesAndFormat** (dictionary of dictionaries) - names of text files to be read in, along with the associated column definitions in the data (format specification)
  - Example:

    ```
    fileName = 'Spartacus20090130_MANOS.txt'
    # list of lists specifying ['property',column] in the text file
    formatSpec = [['night',0],['jd',3],['diffMag',6],['magErr',7]]
    fileNamesAndFormat = {fileName:formatSpec}
    ```

  - Multiple key/value pairs may be used when multiple text files are to be used
  - `'jd'` (Julian date), `'diffMag'` (differential magnitude), and `'magErr'` (magnitude error) must be specified to run the program, additional properties may also be stored in the `lightCurveData` object
  - Remember that Python indexes from zero, so the left-most column in the text file is column 0

- Any white space in the text file is considered a delimiter (leading and trailing white space is ignored)
- Stored in lightCurveData.data

- **offsetsList** (list of dictionaries, `None` = no offsets) - offsets associated with nights in each text file

  - Example: [{1:0.0,2:-0.04,3:0.464}]
  - Key/value pairs must be int/float pairs, where the key is the night number, and the value is the offset
  - Multiple dictionaries may be used when multiple text files are to be used- when this is done, the order of these dictionaries must correspond to the order of the files names and specifications used in `fileNamesAndFormat`
  - Keys may be repeated as long as they are in different dictionaries
  - `'night'` property must be specified in the format to use offsetsList
  - **If more than one night is used for any target, all data must have associated night and offset values**
    - The only case where offsets are not necessary is if the entire dataset came from a single night

## 5.2 The `fitData()` Function

`fitData`(lightCurveData, fitOptions, method = None[, periodGuess = None[, hardMinPeriod = None[, hardMaxPeriod = None]]])

- **lightCurveData** (lightCurveData object)

- **fitOptions** (dictionary) - the options used in calculating the fit, as specified in `setFitOptions` (see Command Line Interface)

  - `orderMin` - minimum $m$ value to be attempted in the Fourier model, as outlined in the Fitting Rationale section (default is 2)
  - `orderMax` - maximum $m$ value to be attempted in the Fourier model, as outlined in the Fitting Rationale section (default is 6)
  - `timer` - whether or not to measure the amount of time it takes to fit the model (default is False)

- **method** (string) - method to be used for traversing the search space of periods: `None`, `'single'` or `'range'`; supplied by the fitInfo file, when available

  - when method is `None`, a maximum recoverable period is estimated for up to 5 times the observing window; periods are checked at 15 minute (0.25 hour) intervals; periodGuess is ignored in this case
  - when method is `single`, periodGuess must be provided as an int or a float, which serves as the only initial period used in the minimization
  - when method is `range`, periodGuess must be provided as a three-element list of `[start, stop, step]` integers or floats, which is then automatically converted into a list of initial periods for minimization

- **periodGuess** (int or float or three-element list of ints or floats) - the initial period used for minimization, given in hours; this provides a starting point for the the period parameter, which does not remain fixed during the minimization; supplied by the fitInfo file, when available

- **hardMinPeriod** (int or float) - the hard lower limit for the period fitting, no period below this value will be attempted in the evaluation; supplied by the fitInfo file, when available

- **hardMaxPeriod** (int or float) - the hard upper limit for the period fitting, no period above this value will be attempted in the evaluation; supplied by the fitInfo file, when available

**Returns** (bestFit, bestOrder, periodsTested, periodErrors), where bestFit is a Minimizer object, bestOrder is an int, and periodsTested and periodErrors are corresponding lists of floats.

### 5.3 The `outputResults()` Function

`outputResults`(fit, m, lightCurveData, outputOptions[, periodErrors = None])

- **fit** (Minimizer object) - the bestFit object returned by `fitData()`
- **m** (int) - the bestOrder returned by `fitData()`
- **lightCurveData** (lightCurveData object) - the lightCurveData object used for this run
- **outputOptions** (dictionary) - the options used in displaying and saving the results of the run, as specified in `setOutputOptions` (see Command Line Interface)
- **periodErrors** (n by 2 list of lists) - when provided, a second figure will be plotted showing the mean RMS of the residuals as a function of period

## 6 Future Improvements

- phase angle considerations, pending upgrades to the Lowell orbital databases
- polling from the Lowell ephemeris database to specify changes due to orbit
- method to limit the Nyquist maximum recoverable period result, which may be unreasonably large due to uneven sampling, possibly using a periodogram
- generalize the inputs for `lightCurveData.getData()`, which currently can only take in floats, so things such as filters (which may be characters) can't be read in

## References

[1] Harris, A. W., et al. "Photoelectric observations of asteroids 3, 24, 60, 261, and 863." Icarus 77.1 (1989): 171-186.

[2] Harris, A. W. & Lupishko, D. F. 1989, “Photometric Lightcurve Observations and Reduction Techniques” in “Asteroids II,” ed. R. P. Binzel, T. Gehrels, & M. S. Matthews.