ADAM software: Some tips on usage

Matti Viikinkoski
matti.viikinkoski@gmail.com

February 7, 2017

1 Introduction

All Data Asteroid Modelling (ADAM) is a software package for shape reconstruction of asteroids from observations. Currently supported data modes are lightcurves, adaptive optics images, occultations and range-Doppler radar images. Code for thermal modelling is also available but not yet integrated into the main program. Asteroids are described by parametric models (octantoids and $\sqrt{3}$-subdivision surfaces) which facilitate representation of general, non star-shaped, closed surfaces. For theoretical background and description of various algorithms used, see [1] and references therein.

2 Shape representation

ADAM uses parametric shape representations. Currently implemented are octantoids and $\sqrt{3}$-subdivision surfaces. Octantoid parameterization expands each vertex coordinate using the spherical harmonics series. This is a global parameterization in the sense that a change in one parameter affects all vertices. In contrast, subdivision surfaces are more localized, since each parameter only affects a few vertices. Internally ADAM uses polyhedrons with triangular facets. Partial derivatives are calculated with respect to vertex coordinates, and only in last step derivatives wrt parameters are calculated using the chain rule. This approach makes it easy to add additional shape parameterizations into ADAM.

3 Data modes

3.1 Lightcurves

As an asteroid rotates, observed brightness varies. A lightcurve is obtained by observing brightness variations of an asteroid over a time period. As brightness depends on the asteroid's shape and surface albedo, we can find a best-fitting surface shape that minimizes difference between observed and measured brightness. Only convex features can be reliably reconstructed. While it is possible that in some rare cases (large phase angles are required) non-convex shape giving a better fit than convex one, non-convexities are not unique.

3.2 Occultations

Occultations are fitted using the procedure described in [2]. For optimal results, initial shape should intersect all the observed chords, and should not intersect any no-intersection chords. In some cases resulting shape is highly dependent on initially chosen offset.

3.3 Adaptive optics and other disk-resolved images

Reconstruction from adaptive optics images requires fitting the projected polyhedron to a 2D image degraded by atmospheric noise. ADAM solves this by minimizing difference between 2D Fourier transform of plane-projected polyhedron, multiplied with Fourier transformed, and Fourier transform of data image.

Image of $n \times n$ pixels with pixel size dp is sampled at $n^2/2 - 1$ frequency points (dc term is not fitted) with the maximum frequency $1/2dp - 1/(ndp)$ and minimum frequency $-1/2dp$, step size $1/(ndp)$. Note that the pixel size determines the maximum usable frequency.

4 Config options

ADAM reads configuration information from an ini file. All the currently available config options are listed in `adam.ini`.

4.1 Shape

The pole direction and rotation period of shape are determined by `Angles`, and the zero-time, JD0, is set by `MinTim`. It is possible to fix pole direction(and rotation period) or shape parameters by setting `FixAngles` and `FixShape`. Typically these options are used with `InitShapeFile` option, which loads a shape from file.

4.1.1 Octantoids

Initial shape is an ellipsoid which set by giving the lengths of semi-axes a,b,c using config option `InitEllipsoid`. The number of facets is $8\texttt{Nrows}^2$. The octantoid shape support is selected by setting `LMAX`. Typical values for `LMAX` are $5 - 9$ and for `nrows`, $10 - 12$. Octantoid shape is determined by $3(\texttt{LMAX} + 1)^2$ parameters.

4.1.2 Subdivision surfaces

If `LMAX` is not set, subdivision surfaces are used. In this case `Nrows` determines the number of shape parameters. Typical values for `nrows` are 4 or 5. `SDLevel` determines number of division steps. Possible values are $-1, \ldots, 2$:

- 2: Subdivision is performed twice, and vertices are mapped to the limit surface. Number of facets is 9 times the facets of the initial shape.
- 1: Subdivision is performed once, and vertices are mapped to the limit surface.
- 0: No subdivision, but vertices are mapped to the limit surface.
- -1: No subdivision, no mapping to the limit surface.

Instead of initial ellipsoid, it is also possible to load an arbitrary shape, e.g. a convex model, using `InitShapeFile`. Warning: There is no global size scaling with subdivision surfaces, so convergence could be affected if the size of initial ellipsoid is chosen badly. In case of elongated surfaces, it is a good idea to set `MinTim` such that shape is aligned with a coordinate axis.

4.2 Optimization

Number of LM optimization rounds is set by `NumberOfRounds`. `LambdaMax` and `LambdaInc` determine if optimization is to be aborted before the number of rounds is reached.

Various data and regularization weights can be set in this section. Weights must be tailored to data and to the shape representation. If data weights are too large compared to regularization weights, unrealistic shapes will result. On the other hand, too strict regularization weights will inhibit convergence.

Most important regularization terms are convex regularization (`ConvexWeight`), dihedral angle regularization (`DiAWeight`), and for octantoids, octantoid regularization (`OctWeight`). Convex regularization penalizes divergence from convexity, dihedral angle regularization prefers planar surfaces, and octantoid regularization calculates the norm of octantoid parameterization.

`UseAOScaling` will fit an additional scaling term for each AO image. This is usually a good idea, but in some cases better results can be obtained without this option.

4.3 Data

Lightcurves can be combined with adaptive optics, occultations, and range-Doppler images. Currently, lightcurves are mandatory. It is possible to use only lightcurves, but only subdivision surfaces support reconstruction from relative lightcurves.

4.4 LC

`LCFile` points to the lightcurve file. `AllLCRelative` takes all lightcurves as relative, even if they are calibrated. This is usually a good idea, since convergence is better for relative lightcurves.

If calibrated lightcurves are used, then the phase parameters must be set in `PhaseParams`. If phase parameters are not to be optimized, then `FixedParams` should be set. In most cases, phase parameters should be fixed, since optimization is often unstable.

Fitting an albedo term for each facet is possible by setting `FitAlbedo`. This should not be used without a good reason.

4.5 AO

Fits file containing AO data is read with `AOFile`. if `AORect` and `AOSize` is used, then only a subimage of full AO image is used. Similarly, `PSFFile` points to a fits file containing PSF image. If the AO image is already deconvolved, then `PSFFile` should not be set.

Observation time (JD) without LT correction is contained in `Date`. If this is not set, then the value of the keyword MJD-OBS from the fits file is used.

`PixScale` sets the pixel scale of the AO image.

It is possible to set the observation geometry of the AO image with `SetCamera` or `SetCamUp` options. If neither are set, then CD keywords from fits file are used. If CDs are not found, then it is assumed that image in aligned north up in the equatorial coordinate system. Additional weights for individual images can be set with `Weight` keyword. It is assumed that `Weight` is equal to 1 unless explicitly set otherwise.

`LowFreq` keyword means that only low frequency part of image is used for fitting. This is useful for low detail images. Also the optimization is consirably faster since only quarter of points are used.

4.6 RD

Config options of range-Doppler images are similar to AO images. Vertical is the delay, horizontal axis is the frequency. It is assumed that horizontal axis is reversed. `RadarFreq` sets the radar frequency. For the best performance, zero frequency should be close to the center of horizontal axis.

Option `PixScale` determines the pixel size. First term is the frequency, second term is the delay in microseconds.

4.7 Ephm

Ephemeris information for all radar and adaptive optics images are contained in the file pointed by `EphFile`. Each line consists of

- observation time (JD, without LT correction)
- Asteroid-Sun vector
- Asteroid-Earth vector

There are python scripts in the `Utils` directory to query Horizons web service for ephemeris information, given the asteroid and the observation time.

4.8 OC

ADAM supports both occultation and non-detection chords. However, if non-detections chords are used, then the initial shape must be chosen so that it does not intersect non-detection chords. Moreover, convergence is usually better without non-detection chords. If only occultations and lightcurves are used, then the initial shape should be chosen so that it intersects almost all the occultation chords. Occultation file is set with `OCFile` config option. The format of the occultation file is described in `README_occultations.txt`. Keyword `SetCamera` describes image orientation. If north is up in equatorial coordinate system, set this to `Equ`. Keyword `OCCOffset` allows presetting shape projection origin to a nonzero value. This is useful for nudging the optimization routine to the right direction. Individual chords can be moved from their initial position using `ChordOffset` keyword. Additionally, position of chords can be optimized: Set movable chords using `FreeChords` keyword. Keyword `ChordWeight` in `Optimization` section controls how much chords are allowed to move from their default position. Larger values will inhibit movement. Warning: Do not set non-detection chords free.

4.9 Output

Keyword `ShapeFile` defines the output file for the shape representation. Shape parameterization is written to the file pointed by `ShapeParamFile`.

5 Frequently asked questions

5.1 How do I select the shape representation?

Octantoids:

- Set `InitEllipsoid`
- Set `Nrows`=10. If facets look to large, increase this value.
- Set `LMAX`=6. Typically values 6-9 work well.

Subdivision surfaces:

- Set `InitEllipsoid`
- Comment out `LMAX`
- Set `Nrows`=4. Usually 4-5 are good choices.
- Set `SDLevel`=2. This means that ADAM does two subdivisions and maps vertices to the limit surface. Number of facets is $9 \cdot 8 \cdot \texttt{Nrows}^2$.

5.2 How do I load an arbitrary shape file as an initial shape?

Arbitrary initial shapes are only supported with subdivision surfaces.

- Set `InitShapeFile` to a file name containing the shape.
- Set `SDLevel`. If the initial shape is used as-is without subdivision, set this value to -1.

5.3 How do I fit a shape using only relative lightcurves?

- Select subdivision surfaces
- Set `MinTim` to JD0
- Set `Angles`
- Set `UseLC`=1.
- Set `LCWeight`,`ConvexWeight` and `DiAWeight`
- Point `LCFile` to the lightcurve data file.
- If there are calibrated lightcurves set `AllLCRelative` to 1 to convert them to relative.
- Set outputfile `ShapeFile`

5.4 How about using calibrated lightcurves?

Relative lightcurves converge much better and the use of calibrated lightcurves is strongly discouraged. Calibrated lightcurves are used as follows:

- Select shape representation
- Comment out `AllLCRelative` if set
- Set phase parameters with `PhaseParams` keyword.
- Set `FixedParams`=1,1,1,1.
- Phase parameters can be optimized along with shape, but this is usually a very bad idea. By setting some (or all) values `FixedParams` to 0, corresponding phase parameters are optimized.
- Additional weight for calibrated lightcurves only can be set with `CalibLCWeight`.

5.5 How do I fit albedo variagation to the existing shape?

- Set `InitShapeFile` to a file name containing the shape
- Set `SDLevel`=-1
- Set `FixShape`=1 and `FixAngles`=1
- Set `FitAlbedo`=1
- Set `AlbedoMax` and `AlbedoMin`
- Set `AlbRegWeight`. Larger values produce smoother results
- Set `AlbedoFile` for the albedo output

5.6 How do I use adaptive optics images?

- Select shape representation
- Select lightcurves
- Set `AOWeight` to some value
- Set `UseAO` to the number of AO images
- For each AO image:
- `AOFile` points to the AO image file
- `AORect` and `AOSize` determine subimage within the fits file. If unset whole image will be used. Warning: Image size must be even.
- Set `Date` to the observation date (JD) without LT correction. If unset, MJD-OBS from the fits file will be used.
- PSF of the AO image is set with `PSFFile` keyword. If unset, delta function is assumed.
- `Pixscale` sets pixel size (in arcsec) for both horizontal and vertical.
- Usually `setCamera` and `SetCamUp` keywords are not needed, since coordinate information is read from the fits file.
- `Weight` keyword allows setting additional weight for individual images, e.g. for images of bad quality it is useful to set `Weight` ¡1.
- `UseAOScaling`=1 is usually useful, but not always.
- Point `EphFile` to the file containing the ephemeris information corresponding to the AO images

5.7 How do I use occultation data?

Only occultation and non-detection chords are supported. Points are not supported and must be removed from the data file. Warning: Non-detection chords could inhibit convergence and better results are usually obtained without them.

- Set `UseOC`=1
- Point OCFile to the occultation data file (for formatting see `README_occultations.txt`)
- (Optional) Keyword `OCCOffset` allows moving the initial projection center from origin. This can affect convergence and shape. For example, `OCCOffset`$= 0, 0, 1.2, -3$ means that projection center of first occultation is not moved, and the origin of projection of second event is moved to $(1.2, -3)$ km. The initial shape and `OCCOffset` should chosen so that lines determined by the occultations intersect the shape. Warning: Non-detection chords must not intersect the initial shape.
- Additional weights for individual chords can be set with `ChordWeight`
- Position of individual chords can be adjusted with `ChordOffset` keyword. For example, `ChordOffset`$= 0, 1.2, -1$ means that second and third chord are moved respectively 1.2 and -1 seconds from their default position.
- Chord positions can be adjusted during the optimization. Keyword `FreeChords` list chords that are allowed to move along the velocity vector. For instance `FreeChords`$= 3, 6$ means that the third and sixth chord are allowed to move. Movement of chords is regularized with `ChordWeight` keyword. Larger the value, less the chords will be allowed to move. Warning: Non-detection chords must not be free.

6 Notes

- Subdivision surfaces have no global scaling term, so it is important to choose the initial ellipsoid carefully.
- Because the image size determines the density of frequency sample points and the 2D Fourier transform of data images is calculated with FFT, images should be not cropped too much.
- Since shape reconstruction is an inverse problem, reconstructed shape is seldom unique, and depends on regularization and data weights.
- If the reconstructed shape seems unrealistic or has self-intersections, increase regularization weights and/or decrease data weights.
- A rule of thumb for choosing data weights is to keep the initial chisq≤ 1000 and set DiaWeight to $0.5 - 2$.

- For elongated shapes, it is beneficial to choose JD0 such that the longest axis is aligned with a coordinate axis.
- If shape looks too nonconvex, increase `ConvexWeight` and/or `DiAWeight`.
- In Utils directory, there are python scripts to query Horizons for ephemeris information.

References

[1] Matti Viikinkoski *Shape reconstruction from generalized projections* Tampere University of Technology, 2016 `http://urn.fi/URN:ISBN:978-952-15-3673-1`

[2] Durech J. et al. *Combining asteroids derived by lightcurve inversion with asteroid occultation silhouettes* Icarus, vol 2014, 2011