Thermal models for planetary Science: the code

Marco Delbo delbo@oca.eu

June 6, 2016

1 Introduction

Thermophysical modelling of asteroid surfaces is about calculating the temperature of the surface and the subsurface of asteroids by means of computer models that take into account a number of physical processes such as the illumination from the Sun, the heat radiation and the heat conduction. These codes are the so-called Thermophysical models (TPM). Physical parameters such as the albedo (or reflectivity), thermal conductivity, heat capacity, emissivity, density and roughness, along with the shape (e.g. elevation model) of the body, its orientation in space, and its previous thermal history are taken into account. The main aim of thermophysical modelling is to derive physical characteristics of asteroids from observational data.

"Simple" thermal models, such as the near-Earth asteroid thermal model (NEATM, [Harris, 1998]), that is typically used where the data quality and/or the available knowledge about target shape and spin preclude the usage of TPMs. Typically, the NEATM allows a robust estimation of asteroid diameter and albedo, not providing any (direct) information on thermal inertia or surface roughness (see Harris and Lagerros, 2002, for a review). The recent large-scale thermal emission surveys of asteroids and trans-Neptunian Objects (see Mainzer et al., 2014; Lellouch et al., 2013) typically use the NEATM in their data analysis, thereby establishing it as the de-facto default among the simple thermal models. Other simple thermal models, such as the "Standard Thermal Model" (STM) or the "Isothermal Latitude Model" (ILM A.K.A. FRM), which were discussed, e.g., by [Harris and Lagerros, 2002, Delbo and Harris, 2002], have largely fallen out of use.

For more information see paper by [Delbo et al., 2015]: http://arxiv.org/abs/1508.05575 and the many references there in. In particular, we strongly advise to read [Rozitis et al., 2013, Rozitis et al., 2014, Rozitis and Green, 2011, Rozitis and Green, 2014]

2 In preparation for Install

I suggest to install gnuplot and wget and the bulid essentials (Make, gcc ecc)

On Ubuntu

```
sudo apt-get install gnuplot
sudo apt-get install wget
sudo apt-get install build-essential
```

On Mac OS using macports (brew can be used instead and I believe the commands are very similar) install macports from https://www.macports.org/install.php then, open a terminal and type:

```
sudo ports install gnuplot
sudo ports install wget
```

You need to install the buidling environment

On MacOS X the compiler is NOT installed by default.

FOR OLD SYSTEMS go get OS X original DVD and install the XCODE. See for more info `http://developer.apple.com/technologies/tools/xcode.html`. Or drop me en email.

For the OS X Mavericks, I suggest to Install Command Line Tools in OS X Mavericks Without Xcode:

Mac users who prefer to have a more traditional Unix toolkit accessible to them through the Terminal may wish to install the optional Command Line Tools subsection of the Xcode IDE. From OS X Mavericks onward, this is now easily possible directly and without installing the entire Xcode package first, no developer account is required either.

Follow these instructions:
`http://osxdaily.com/2014/02/12/install-command-line-tools-mac-os-x/`

or 1. Launch the Terminal, found in /Applications/Utilities/
2. Type the following command string:

```
xcode-select --install
```

3. Follow the instructions.
These actions require the administrator passwords.

On Windows: we do not support MS-Windows native mode of the code. Originally we developed the code using Visual Studio, but in the end of 2007, M. Delbo decided to migrate to UNIX style OS. It was a hard decision, as it costed him a girlfriend of that time. But a decision is a decision. At the beginning the code was run under Cygwin (https://www.cygwin.com) with good results. You might try to again, but watch your girl/boy friend.

3 Download

The code can be dowloaded from: `www.oca.eu/delbo/thermops.tar.gz`. This readme file is also on-line at: `www.oca.eu/delbo/readme.pdf`.

Expand the file in the current directory:

```
tar xvf thermops.tar.gz
```

4 Building the code

The code is distributed within the directory thermops.

Go to the directory thermops

```
cd thermops
```

and build the code by:

```
make
```

Note: you need to have the compiler(s) installed on your system. The executable files are created in the "thermops/bin" directory.

5 Installing the code

There are two option:

5.1 Clean installation

If you have administrator rights (I believe).

```
sudo make remove
sudo make install
```

Now you can even erase the downloaded file and the directory thermops.

5.2 Change the PATH

Modify your search path by adding the thermops/bin directory to it: you do

```
pwd
```

in order to know which directory are you in. e.g.: /Users/mdb/dev/thermops Then, you type:

```
PATH=$PATH:/Users/mdb/dev/thermops/bin
```

Note that you have to add the /bin directory!!. YOU CANNOT ERASE THE the thermops directory

5.3 Create symbolic links to the code

If you have administrator rights, AND YOU WISH TO MODIFY THE SOURCE CODE AND RECOMPILE IT ALL THE TIME you can use the commands:

```
sudo make remove
sudo make makelink
```

to create symbolic links to the code. You will be able to acess the programs from every directory. YOU CAN NOT ERASE THE the thermops directory.

bottom line: Now you can you access the code from whichever directory you wish to work in. Try it by typing:

```
runtpm -h
```

6 Description and usage of the package

TherMoPS is a sweet of the following programs:

- **asteroid2mpc** convert an asteroid number or previonsal designation into the MPC packed code. usage:

  ```
  echo 162500 | asteroid2mpc
  echo 1999_RQ36 | asteroid2mpc
  ```

- **caldat** convert a JD into a calendar date usage:

  ```
  echo 2456746.1268166667 | caldat
  ```

- **damit2obj**: Convert J. Durech's asteroids shapes (from the DAMIT) [Durech et al., 2015, Durech et al., 2009] to wavefront format (to be used by the TPM). usage:

  ```
  #get the shape of e.g. (32) Pomona from the DAMIT
   wget "http://astro.troja.mff.cuni.cz/projects/asteroids3D/ \
         data/archive/1-1000/A118.M127.shape.txt"
  # convert it to OBJ
  cat A118.M127.shape.txt | damit2obj >32.obj
  ```

- **fittm**: to determine diameter, albedo, eta-value of an air-less spherical (or supposed so) body from thermal IR measurments using simple thermal models: namely the NEATM [Harris, 1998], the STM and the FRM [Delbo and Harris, 2002, Harris and Lagerros, 2002, Delbo et al., 2015]. See section **??** for the usage.

- **tmflux**: to calculate the Spectra Energy Distribution emitted by an air-less spherical (or supposed so) body given a set of physical parameters: diameter, albedos, distances from the sun and the observer etc. See section **??** for the usage.

- **runtpm**: Thermophysical model. See section 8.3.

- **meshvolume**: Calculate the volume of a 3-d mesh. Usage:

  ```
  echo 32.obj | meshvolume
  ```

  where 32.obj is the filename of the mesh (32 Pomona).

- **mesh2ellips**: Calculate the best fit ellipsoid to a 3-d mesh (wrote by Emilie Marchese). See also [Carbognani et al., 2012].

- **h2d**: Calculate asteroid diameter in (km) given the H magnitude and the geometric visible albedo. Usage:

  ```
  echo 15.5 0.15 | h2d
  ```

  where 15.5 is the H and 0.15 the albedo.

- **h2pv**: Calculate asteroid geometric visible albedo given the H magnitude and the diameter. Usage:

  ```
  echo 15.5 1.0 | h2pv
  ```

  where 15.5 is the H and 1.0 is the diameter of the asteroid in km.

- **pv2a**: Calculate bolometric Bond's albedo from geometric visible albedo. Usage:

  ```
  echo 0.23 0.15 | pv2a
  ```

  where, for instance, 0.23 is the geometric visible albedo and 0.15 is the G value of the H,G system of Bowell et al. in Asteroids II book .

- **tss**: Calculate the sub solar temperture assuming istantaneous solra re-irradation and correction of this using the *eta*-value (beaming parameter). Typical usage: is echo A eta epsilon r — tss e.g.

  ```
  echo 0.05 1.2 0.9 1.2 | tss
  ```

  where 0.05 = Bond Albedo, 1.2 = beaming parameter, 0.9 = emissivity, and r = heliocentric distance in AU.

- **skind**: Calculate the heat penetration wave depth for a periodic illumination.
  INPUT : k (W m-1 K-1) rho (kg m-3) c (J kg-1 K-1) period (h)
  OUTPUT: INPUT Gamma (J m-2 s-0.5 K-1) skindepth (m)
  USAGE e.g. :

  ```
  echo 0.1 2500 500 | skind
  ```

- **jdcnv**: Julian day calculation.

- **jy2w**: Convert Jy to $W\ m^{-2}\ \mu m^{-1}$ (bash).

- **fitplot**: Execute fittm and tmflux to preduce plot of thermal models (bash; requires gnuplot to be installed).

- **mirieph, mirivec, miridista**: Calculate ephemeris, heliocentric cartesian coordinates and distances, respectively of asteroids using the MIRIADE service of the IMCCE paris. Thanks to Jerome Brthier.

  other programs are :genobs refittpm tstat specpos f2stokes f2stokesBen p2pr thermalp tsurf tmul gamma2theta hor2vec miriflux getbestscalefactor tmapbin2ascii getmpcorb papers2bib

To get help on how to use each program keep reading this document or call the program with the "-h" switch: e.g.

```
fittm -h
```

7 License

The code is free. The STM, FRM, and NEATM algortihms are public. See [Harris, 1998, Delbo and Harris, 2002, Harris and Lagerros, 2002]. The TPM is described by [Delbo et al., 2015, Mueller, 2007, Delbo et al., 2007].

8 Using the code

8.1 How to fit simple thermal models

Simple thermal models are the STM, the FRM and the NEATM. These models are described by [Harris, 1998, Delbo and Harris, 2002, Harris and Lagerros, 2002].
Run

```
fittm -h
```

to get help about the usage of the code, input and output parameters.

Here we learn how to use the simple thermal model by performing a fit. We take as an example the Keck observations of the near-Earth asteroid (14402) 1991 DB in the thermal infrared from [Delbo et al., 2003].

I suggest to prepare an input file for fittm. The file should contain: model to be fitted, H, G, emissivity, initial eta value (adjusted by the NEATM), initial geometric albedo value (adjusted by the fit of the thermal model), heliocentric distance (AU), geocentric distance (AU), and phase angle (deg), followed by the measured infrared fuxes.

```
0 18.85 0.15 0.9 1.0 0.1 1.076 0.103 36.0
3.5 0.006 0.0008
4.8 0.046 0.003
8.0 0.368 0.016
8.9 0.445 0.014
10.7 0.622 0.011
11.7 0.682 0.012
12.5 0.640 0.012
20.0 0.513 0.021
20.0 0.507 0.024
```

Save the text above into a file e.g. 91db.dat. NOTE: the 3.5 micron flux is not real. I have added it just to show the use of the -V switch of fittm and tmflux.

To perform the fit, pipe the 91db.dat file into fittm:

```
fitm < 91db.dat
```

or

```
fitm -v < 91db.dat
```

if you want to use the verbose mode, which shows every step the code takes to perform the fit. This mode is fundamental to discover mistakes such as wrong input parameters (or code bugs).

This is the output of the code:

```
l>    D(km)   sigmaD     pV    sigma_pV  eta   sigma_eta   chi^2
o>    0.601   0.009  0.141 0.004  1.049 0.033   98.2
```

8.2 How to estimate fluxes from an asteroid using simple thermal models

The program *tmflux* can be used to calculate flux estimates of asteroids using simple thermal models. Run

```
tmflux -h
```

to get help about the usage of the code, input and output parameters.

The bash command:

```
echo 0.601 0.141 1.049  1.076 0.103 36.0 | tmflux -r
```

produces the model flux of 1991 DB using diameter, albedo and eta value derived from *fittm*. Input are: diameter, geometric visible albedo, eta value, heliocentric distance (AU), geocentric distance (AU), and phase angle (deg). The output is: 1st column: wavelength (um), second column: flux (Jy); namely:

```
  4.0 1.885912e-02
  4.5 4.036351e-02
  5.0 7.201254e-02
  5.5 1.128419e-01
  6.0 1.607446e-01
  6.5 2.131309e-01
  7.0 2.674420e-01
  7.5 3.214570e-01
  8.0 3.734255e-01
  8.5 4.220838e-01
  9.0 4.666064e-01
  9.5 5.065301e-01
 10.0 5.416733e-01
 10.5 5.720611e-01
 11.0 5.978628e-01
 11.5 6.193414e-01
 12.5 6.368161e-01
 12.5 6.506331e-01
 13.0 6.611464e-01
 13.5 6.687032e-01
 14.0 6.736356e-01
 14.5 6.762549e-01
 15.0 6.768484e-01
 15.5 6.756786e-01
 16.0 6.729831e-01
 16.5 6.689753e-01
 17.0 6.638460e-01
 17.5 6.577649e-01
 18.0 6.508820e-01
 18.5 6.433300e-01
 19.0 6.352255e-01
 19.5 6.266707e-01
```

```
 20.0 6.177550e-01
 20.5 6.085568e-01
 21.0 5.991438e-01
 21.5 5.895753e-01
 22.0 5.799024e-01
 22.5 5.701693e-01
 23.0 5.604141e-01
 23.5 5.506695e-01
 24.0 5.409636e-01
 24.5 5.313201e-01
 25.0 5.217593e-01
```

The output of *tmflux* can be saved into a file by:

```
echo 0.601 0.141 1.049  1.076 0.103 36.0 | tmflux -r >91db.out
```

so that the model flux can be plotted against the data points (using e.g. gnuplot)

```
gnuplot> set logsc y
gnuplot> set xrange [4:21]
gnuplot> set xlabel 'Wavelength (um)'
gnuplot> set ylabel 'Flux (Jy)'
gnuplot> plot '91db.dat' every ::1 u 1:2:3 w error, '91db.out' w l

gnuplot> set term postscript enhanced "Helvetica 16"
gnuplot> set out '91db.fit.ps'
gnuplot> replot
gnuplot> set out
gnuplot> set term x11
```

8.3 Thermophysical Model

A thermophysical model of asteroids (TPM) is used to calculate model infrared fluxes taking into account a number of physical processes such as the heat conduction, the effect of roughness, albedo variegation, direction of the rotation spin, rotation period, and macroscopic shape instead of the empirical correction parameters used in the Standard Thermal Model and in the NEATM. see e.g.
`http://www.diss.fu-berlin.de/diss/receive/FUDISS_thesis_000000002596`
`http://www.astro.uu.se/planet/asteroid/thermal/`

I also strongly suggest to read Michael Mueller's PhD thesis "Surface Properties of Asteroids from Mid-Infrared Observations and Thermophysical Modeling" avaible on the ASTROPH at `http://lanl.arxiv.org/abs/1208.3993`

The code that performs TP modeling is **runtpm** which is called with the following MINIMUM syntax:

```
echo meshfile ephfile epsilon Gamma A gamma rho_c | runtpm
```

of course over one line, where:

shapefile	name of the file of the mesh	must have wavefront format
ephfile	name of the file of the ephemeris	ASCII
epsilon	infrared emissivity	
Gamma	Thermal inertia	$Jm^{-2}s^{-0.5}K^{-1}$
A	Bolometric Bond's Albdeo	
gamma	Semiaperture angle of craters	degree
rho_c	crater surface density	

runtpm has a number of switches to change its behavoir:

- -h print the help of the TPM
- -I ¡Nrings¿ :: change the number of rings by which craters are discretized
- -l ¡luminosity_factor¿ change Sun Luminosity. The new luminosity is 1373*luminosty_factor W m-2
- -n ¡Nslabs¿ Change number of slabs in the subsurface (default is 32)
- -x ¡dx¿ Change the dx (steps in depth) in unity of the heat penetration depth (default is 0.25)
- -t ¡delta_Temperature¿ Change the temperature threshold to check for stability (unit K)
- -i ¡Temperature¿ Force all facets and all crater facets to have the same initial ¡Temperature¿ (unit K)
- -L ¡mode¿ disable thermal modelling and enable lightcurve modeling (don't use it)
- -T ¡filename¿ Facet temperature are initialised to the values in the file
- -a ¡JD0¿ ¡JD1¿ Temperatures will be initialised using average insolation between ¡JD0¿ and ¡JD1¿
- -N interpret the shape file as a bunch of normals and not as an .obj (shape must be convex)
- -p ¡Npoints_per_rotation¿ Number of ministeps/rotation (default 360). Ephemeris are interpolated to have ¡Npoints_per_rotation¿
- -s ¡scale¿ Mesh vectors are scaled by the factor ¡scale¿ The facet area is multiplied by ¡scale¿^{2}

- -o ¡filename¿ Observations from ¡filename¿ are linked to the thermal model. Model fluxes are compared to observed ones
- -b ¡eta¿ The default beaming paramter of 1 is changed to the value of ¡eta¿ This changes the surface temperature.
- -C ¡filename¿ Model fluxes are color corrected using a color correction file
- -H ¡H¿ The ¡H¿ value of the H,G system is used to calculate model reflected light component
- -G ¡G¿ The ¡G¿ value of the H,G system is used to calculate model reflected light component (default 0.15)
- -R ¡R_p¿ Ratio between the IR and the visible albedo
- -S ¡filename¿ Uses spin data from filename. Format: see J. Duerch's DAMIT (lambda, beta, period, JD0, phi0)
- -f Performs full blown heat diffusion in crater tiles instead of using J. Lagerros approximation
- -j Do not use the optimised scale for chi2 and residuals output
- -e ¡scale¿ Multiply observed flux uncertainties by ¡scale¿
- -X ¡scale¿ Multiply observed fluxes by ¡scale¿

The mesh file is in wavefront format. See `https://en.wikipedia.org/wiki/Wavefront_.obj_file`.

The format of the ephemeris file is :
JD r_X r_Y r_Z
where JD is the julian date or an epoch in days, r_X, r_Y, r_Z are the X,Y,Z components of the position of the body in the heliocentric ecliptic reference frame as SEEN AT THE ASTEROID.
Comments may be inserted by a # on the first column. Each line of the file is an ephemeris point.

Format of the observation file :
The file must start with an integer value that is the <total number of observations in the file>

Each observation come in epoch-blocks. Then for each observation block:
JD $< No >$ i.e. the number of wavelengths r_X r_Y r_Z (as for the ephemeris file) d_X d_Y d_Z (heliocentric X,Y,Z components of the vector from the asteroid to the Earth) wavelength

flux sigma_flux ColorCorrectionIndex FLAGS

Wavelength is in μm, flux is the monochromatic measured infrared flux density in Jy, sigma_flux is its uncertainty.
FLAGS are optional and can be summed up: 1 generate the file tmap_temps.dat
2 generate the file tmap_temps.bin
4 generate the file tmap_flux.bin
8 generate the file tmap_flux.dat
A vaild FLAGS value is 1, 2 or 9, 11, etc.

Comments may be inserted by a # on the first column.

Example:

```
19

2455220.321482 2
2.12274000 0.22792300 0.94111100
1.57842000 1.04795000 0.94111400
11.0984 0.0154487073 0.0005122363 2
22.6405 0.0386639376 0.0030269182 3


2455220.453801 2
2.12271000 0.22908400 0.94086300
1.57645000 1.04784000 0.94086600
11.0984 0.0149586412 0.0004959870 2
22.6405 0.0441068539 0.0030061696 3

....
```

Let's learn how to use the code with examples rather than with a tedious description of the theory of its operations.

8.3.1 Exercise 1 Calculate the temperature of the asteroid 65803 Didymos

Semi-major axis 1.644 AU. Let's calculate the temperature at that distance from the Sun. The first step is to generate the ephemerides for this asteroid. The TPM will follow them and it will calc the temperature at each ephemeris point. If you give ephemeris with a step that is very much too big, don't worry, TPM will interpolate them (linearly) in order to guarantee that there are at leaf 360 point where the temperature is calculated. We need to do this in order to thermalise the asteroid. For thermal inertias typical of NEAs (50-700) 10-20 rotations are required for thermalisation. For higher value, more rotations are required. Let assume a contant position at r=1.644 AU for about 10 days.

Open a terminal and type:

```
echo 0    1.644 0 0 >ephDidy.txt
echo 1    1.644 0 0 >>ephDidy.txt
echo 10   1.644 0 0  >>ephDidy.txt
echo 10.1 1.644 0 0 >>ephDidy.txt
```

This produces a file ephDidy.txt suitable for the code.

Get a suitable shape of the asteroid. As this is a top-shaped body. We get the radar shape of 1999 KW4.

```
wget "http://echo.jpl.nasa.gov/asteroids/shapes/kw4a.obj"
```

Finally we need to create a spin vector information file. go to: http://earn.dlr.de/nea/065803.htm Here are the info for the spin status: e.g.

```
echo  157 -90 2.2593 0 0 > spinDidy.txt
```

Which are longitude, latitude of the spin vector, rotation period in hours, initial phase (phi0) and epoch of the initial phase. These information are used to transform an heliocentric vector to vector centred on the asteroid and co-rotating with it. For more information, see the SPIN FILE FORMAT of the DAMIT. `http://astro.troja.mff.cuni.cz/projects/asteroids3D/web.php?page=db_description`

We are now ready to start the TPM:

```
echo kw4a.obj eph.txt 0.9 100 0.1 0 0 | runtpm -S spinDidy.txt
```

Notice that the program calculates the temperatures and then exits. Not very intersting. By inserting the FLAG "4" at JD=10 days, the TPM prodeuces a file called tmap_temps.dat that contains the temperatures of each facet calculated at that specific epoch.

Rebuild the eph.txt file by issuing the commands:

```
echo 0    1.644 0 0 >ephDidy.txt
echo 1    1.644 0 0 >>ephDidy.txt
echo 10   1.644 0 0 4 >>ephDidy.txt
echo 10.1 1.644 0 0 >>ephDidy.txt
```

then relaunch the TPM using the same command as above. You should notice the output of the TPM:

There is a line :

```
PrintTemp.. ... print temp done
```

The tmap_temps.dat file was created. If you type:

```
head tmap_temps.dat
```

you should be able to see the temperature of the surface elements of the shape models. E.g.

10.000000 36.000000 0.250000 0 30.2 88.3 32 120.0 -0.0 150.1 153.0 155.5 157.7 159.6 161.2 162.5 163.5 164.3 164.9 165.3 165.6 165.7 165.7 165.6 165.5 165.4 165.2 165.0 164.9 164.7 164.6 164.5 164.4 164.3 164.2 164.2 164.1 164.1 164.1 164.1 164.1

10.000000 36.000000 0.250000 1 90.1 88.3 32 120.0 -0.0 139.6 141.7 143.6 145.2 146.6 147.7 148.6 149.3 149.8 150.2 150.4 150.6 150.6 150.5 150.5 150.3 150.2 150.1 149.9 149.8 149.7 149.6 149.5 149.4 149.4 149.3 149.3 149.3 149.3 149.2 149.2 149.2

10.000000 36.000000 0.250000 2 150.0 88.2 32 120.0 -0.0 139.6 141.6 143.5 145.0 146.4 147.4 148.3 148.9 149.4 149.6 149.8 149.9 149.8 149.7 149.6 149.5 149.3 149.2 149.0 148.9 148.8 148.6 148.6 148.5 148.4 148.4 148.4 148.4 148.3 148.3 148.3 148.3

The surface temperature is the 10th column. A plot of the 10th column vs the 5th, the facet longitude, should show the typical diurnal temperature profile.

```
echo "plot 'tmap_temps.dat' u 5:10" | gnuplot -persist
```

To get a better view of the temperature distribution run gnuplot and then type (in gnuplot) the following commands:

```
set view 0, 00
set dgrid3d 60,60
set pm3d
set style data lines
set xlabel "Time of the day (deg)"
set ylabel "Latitude (deg)"
set cblabel "Temperature (K)"
set palette defined ( 0 "black", 1 "blue", 2 "green", 3 "red", 4 "yellow" )
splot [0:360][-90:90]'tmap_temps.dat' u 5:6:10 lc palette
```

8.3.2 runtpm command line switches

-h this help
-l *l*uminosity Scale the sun luminosity: 1=Sun 10=ten times the sun 0.5=half the sun
-n *n*umber of slabs: number of elements in the depth buffer [default is 32]
-x *d*x: length step: default is 0.25 x the heat lenght scale=sqrt(kappa/c/rho/omega)
-t *t*r: change the temperature threshold in the check fo stabiity: default is 1 deg
-i *T*i: initial temperature: force all facets and the depth buffer to the Ti at start
-L *m*odel : TPM calculates visibile lightcurves
model=1 Geometric scattering
model=2 Lambert scattering
model=3 Lommel-Seeliger Scattering
model=4 Kaasalainen Scattering (0.1 Lambert + 0.9 Lommel-Seeliger)

8.3.3 Exercise 2: TPM of IRAS data

See [Delbo and Tanga, 2009] for the science case. Here we want to perform the TPM of IRAS data of 184 Dejopeja. We will use shapes and pole solutions from the DAMIT database website: `http://astro.troja.mff.cuni.cz/projects/asteroids3D/web.php`.
There are 2 pole solutions for 184 Dejopeja, which implies that we will perform 2 separates TPMs.

- Create the directory structure where to work on:

```
$ mkdir 184Dejopeja
$ cd 184Dejopeja/
$ mkdir spin1
$ mkdir spin2
$ cd spin1
```

- Get the 3D shape and the spin solution from the DAMIT website:

```
wget "http://astro.troja.mff.cuni.cz/projects/asteroids3D/data/archive/1-1000/A149.
wget "http://astro.troja.mff.cuni.cz/projects/asteroids3D/php/spin.txt.php?model_id
```

- Convert them to the good format:

```
cat A149.M175.shape.txt | damit2obj > 184.obj
awk '{if(NF==3)printf("%s ",$0);if(NF==2)print $2,$1;}' spin.txt.php\?model_id\=175
```

- Extract IRAS infrared fluxes: The code to do this action is **simpsextract** usually called with the following syntax:

```
simpsextract -f <SIMPSdatabaseFile> <astnumber>
```

  However, its correct use requires a color correction of the caltalogue fluxes. The color correction is function of the effective temperature of the source. This is detailed in: `http://irsa.ipac.caltech.edu/IRASdocs/exp.sup/ch6/tabsupC6.html`.
  simpsextract has a very nice way to deal with the color correction. Simply give the expected albedo of the asteroid (e.g. the SIMPS albedo) and an η value. The effective temperature of the asteroid will be calculated for each sightings from the actual heliocentric distance of the asteroid at the time of the IRAS observation.

  The IRAS albedo can be obtained from the file **SIMPS.txt**, fourth column. For 184 Dejopeja the geometric visible albedo is of 0.19. We need to convert this albedo into a bolometric Bond's albedo:

```
echo 0.19 0.15 | pv2a
```

where we assumed G=0.15, resulting in A=0.0754. Let's assume $\eta = 1.2$ and this is the complete call to the code to retrieve the correct fluxes. First we do it in verbose mode to see that everything went ok:

```
simpsextract -v -a 0.1 1.2 -f ~/Documents/DATA/iras/data/sightings.tab 184
Color correction using the bolometric Bond albedo 0.100000 and eta of 1.200000 will be performed
sightings file /Users/mdb/Documents/DATA/iras/data/sightings.tab will is used
Color correction for 202.9K black body will be performed: r=3.451000 AU
K@12 25 60 100 um
0.832882 1.045765 1.160000 1.060000
 1983-07-14T15:50:13   2445530.1598726851  12.0   0.902   0.130     7.000  0.882
 1983-07-14T15:50:13   2445530.1598726851  25.0   1.950   0.385    17.600  1.000
 1983-07-14T15:50:13   2445530.1598726851  60.0   1.356   0.360    12.000  1.000
 1983-07-14T15:50:13   2445530.1598726851 100.0   1.319   0.305     6.300  0.855
Color correction for 202.9K black body will be performed: r=3.451000 AU
K@12 25 60 100 um
0.832882 1.045765 1.160000 1.060000
 1983-07-14T17:33:24   2445530.2315277779  12.0   0.690   0.093     7.000  0.882
 1983-07-14T17:33:24   2445530.2315277779  25.0   1.921   0.395    19.500  1.000
 1983-07-14T17:33:24   2445530.2315277779  60.0   1.078   0.244    12.900  1.000
Color correction for 202.9K black body will be performed: r=3.451000 AU
K@12 25 60 100 um
0.832882 1.045765 1.160000 1.060000
 1983-07-22T02:24:47   2445537.6005439814  12.0   0.917   0.124     8.900  0.957
 1983-07-22T02:24:47   2445537.6005439814  25.0   2.523   0.411    31.000  1.000
 1983-07-22T02:24:47   2445537.6005439814  60.0   1.274   0.300    13.200  1.000
 1983-07-22T02:24:47   2445537.6005439814 100.0   3.157   0.833    10.300  1.000
Color correction for 202.9K black body will be performed: r=3.451000 AU
K@12 25 60 100 um
0.832882 1.045765 1.160000 1.060000
 1983-07-22T12:54:02   2445538.0375231481  12.0   0.780   0.107     7.000  0.882
 1983-07-22T12:54:02   2445538.0375231481  25.0   2.183   0.385    26.200  1.000
 1983-07-22T12:54:02   2445538.0375231481  60.0   1.159   0.262     9.400  0.976
```

We finally produce fluxes and we store them in the file 184.iras

```
simpsextract -a 0.0746 1.2 -f ~/Documents/DATA/iras/data/sightings.tab 184 > 184.ir
```

- Generate position of the observer: wWe need the observer position at the time of the IRAS observations. We do a small error assuming that IRAS was at the center of the Earth. We use ephemeris service of the IMCCE MIRIADE (Thanks to J. Berthier) to calculate the vector position observer-asteroid for each IRAS observation.

```
jds=‘awk ’{print $2}’ 184.iras‘; for jd in $jds; do
  mirivec 184 $jd 1 1 500 |
  awk ’{print $5,$6,$7}’;
  done > 184.obs.vect
```

- Generate the final observation file

```
awk ’{print $2"\t"$3"\t"$4"\t"$5"\t"}’ 184.iras > .t
paste .t 184.obs.vect > 184.obs
rm .t
```

- Create the asteroid position file of ephemeris. Note: we have to start the TPM about 1/2 month before the first observing epoch (2445530.1598726851)

```
mirivec 184 2445515.1598726851 24 1 500 |
 awk ’{print $1"\t"$2"\t"$3"\t"$4}’ > 184.eph
```

- Start the TPM!!!

  ```
  spin=‘cat 184.spin‘
  echo 184.obj 184.eph 184.obs $spin 180 1.0 100 0.0746 45.0 1.0 80.0 1 | runtpm
  ```

  and look at its awesome aoutput!

```
mesh 184.obj loaded OK fn=2040; vn=1022
Ephemerides 184.eph loaded OK N=24; elapsed time 22.999988 days
Observations 184.obs loaded OK N=14; elapsed time 7.877650 days
3.490659 0.907571 0.268380 0.000000 2443395.000000
TPM will use 180 points per rotation
Physical Parameters: emissivity=1.000000(SI); Gamma=100.000000(SI); A=0.074600; CRATERS angle=45.000000; density=1.000000
mesh scaled by 80.000000
Model Mode = 1
2445515.159873 2445538.159861
jd extraction OK:  N=15425; dt=128.822220 seconds
Sun pos interpolated OK N=15425; dt=128.822220 seconds
2445515.159873 2445515.162855
lambda,beta, P, phi0, t0: 3.490659 0.907571 0.268380 0.000000 2443395.000000
matrix elments:
-0.740488 -0.269516 -0.615661
0.342020 -0.939693 0.000000
-0.578533 -0.210569 0.788011

Rotation of Sun vector to asteroid frame done
lambda,beta, P, phi0, t0: 3.490659 0.907571 0.268380 0.000000 2443395.000000
matrix elments:
-0.740488 -0.269516 -0.615661
0.342020 -0.939693 0.000000
-0.578533 -0.210569 0.788011

Rotation to asteroid frame done
Mesh temperature buffers allocated OK
Craters added: Hapke thetabar=20.04
Starting TPM with 15425 points and 14 observations
Setting up obs points: jds=0.000000(min)
initial temp -1.000000
resetting temperatures for 2040 facets
runTPM started with Thermal Inertia, Craters, and N=15425
Using Lagerros' approx heat diffusion into craters
(T_theta/T_0)crat = (T_theta/T_0)smooth
f> 00 2445530.1598726851   -2.6  12.00    0.900000   0.130000    1.342145      0.6706   69.4   35.7  93.1   37.4
f> 01 2445530.1598726851   -2.6  25.00    1.945000   0.385000    4.107788      0.4735   69.4   35.7  93.1   37.4
f> 02 2445530.1598726851   -2.6  60.00    1.356000   0.360000    2.370380      0.5721   69.4   35.7  93.1   37.4
f> 03 2445530.1598726851   -2.6 100.00    1.319000   0.305000    1.157386      1.1396   69.4   35.7  93.1   37.4
f> 04 2445530.2315277779   -2.5  12.00    0.689000   0.093000    1.193701      0.5772  333.4   35.7 356.9   37.4
f> 05 2445530.2315277779   -2.5  25.00    1.916000   0.395000    3.695473      0.5185  333.4   35.7 356.9   37.4
f> 06 2445530.2315277779   -2.5  60.00    1.078000   0.244000    2.149160      0.5016  333.4   35.7 356.9   37.4
f> 07 2445537.6005439814   -2.4  12.00    0.916000   0.124000    1.301156      0.7040  170.1   35.9 193.5   37.3
f> 08 2445537.6005439814   -2.4  25.00    2.516000   0.411000    3.748461      0.6712  170.1   35.9 193.5   37.3
f> 09 2445537.6005439814   -2.4  60.00    1.274000   0.300000    2.169815      0.5871  170.1   35.9 193.5   37.3
f> 10 2445537.6005439814   -2.4 100.00    3.157000   0.833000    1.067344      2.9578  170.1   35.9 193.5   37.3
f> 11 2445538.0375231481   -2.2  12.00    0.779000   0.107000    1.412709      0.5514  304.2   35.9 327.4   37.3
f> 12 2445538.0375231481   -2.2  25.00    2.177000   0.385000    4.199134      0.5184  304.2   35.9 327.4   37.3
f> 13 2445538.0375231481   -2.2  60.00    1.159000   0.262000    2.406229      0.4817  304.2   35.9 327.4   37.3

runTPM completed

Gamma  tbar  scale  eff_diam  chi^2  NvalidObs  Nobs
o>    100.00  20.0 0.883366 87.679462 5.054124 14 14
Mesh temperature buffers freed OK
Sun vectors freed OK
Observer vectors freed OK
JD vectors freed OK
TPM run in 9 seconds
```

- Multiple TPM runs as function of thermal inertia

  ```
  for g in 0 10 50 100 200 500 1000 2000; do echo 184.obj 184.eph 184.obs $spin 180 1
  ```

  and watch the output

  ```
  grep 'o>' tpm.*.out | sort -n -k 2
  ```

8.3.4 TPM Modeling of WISE data

For information about WISE, see Wright et al. (2010), Mainzer et al. (2011), Masiero et al. (2011).

The procedure to run TPM is the same as for IRAS data but there are two additional requirements:

- A color-correction table (file `T_color_correction_table.dat`) is required. Each WISE flux reported in the IRSA/IPAC archive requires a color correction (see Togunaga & Vacca 2005) related to the width of each band, W1, W2, W3 and W4. Instead of color correcting the fluxes (the source's SED is not known a priori), we integrate our model SED over the WISE bands total spectral response profiles to be able to fit the data without color correction (we effectively "color contaminate" our model).
- In the observations file (see above for details about the format), each flux entry must contain an additional column: after the wavelength value, the WISE flux, and its errorbar, there must be a filter tag (0 for W1, 1 for W2, ... 3 for W4). This will select the appropriate color correction factor for each band according to each facet's temperature.

For example, for (1980) Tezcatlipoca:

```
echo SHAPE.obj eph.txt 0.9 220 0.09 0 0 | runtpm -S spin.txt -o WISE.obs -C T_color_correction_table.dat
```

Typically, you want to save the output by adding `> TPMoutputFilename.txt` to this line. In more up-to-date versions of the code, the chi2 of the fit and other parameters, such as the scale that minimizes it, are output in a line starting with the characters "r>". It can be printed on the console with grep:

```
grep -B 1 'r>' TPMoutputFilename.txt
```

```
h>     Gamma t.Bar gamma  rhoC     A      scale         Dv         Ds    eta      chi^2        RMS
r>     220.0   0.0   0.0  0.00  0.09 5.23835850      6.499      6.857  1.000    564.294    0.00688
```

Running this with a script for different values of thermal inertia and surface roughness and storing the output in different files allows us to easily search for a minimum in the chi2 versus gamma space.

Note: WISE took data in up to four filters W1, W2, ... W4 (for more information, see Mainzer et al. 2011, Masiero et al. 2011). The fact that band W1 is mostly dominated by reflected sunlight and that W2 fluxes have a non-negligible reflected light component (usually the case for main belt asteroids) means that TPM will mostly be based on W3 and W4 data.

9 code Updates

8 Feb 2009

I noticed that the Blender (`www.blender.org`) can generate super good spheres: the function is Add-¿mesh-¿icoShpere and it is very cool. Then you can pinch, sculpt it etc etc....

9 Feb 2009

runtpm and tpm now can use an arbitray Sun Brighntess. To change the value of the value of the brightness of the sun use the argument -l e.g. runtpm -l 0.5 forces the TPM to use half the Sun Brightness ..... updated the dependencies within the makefile

Feb 27, 2009

The heat diffusion eq. is numerically stable provided the time resolution is not too coarse, specically (see Press et al., 1992, sect. 19.2) the parameter dTdZ2 (dimensionless time resolution divided by the square of the dimensionless depth resolution) must not exceed 0.5. dimensionless time resolution: $\tau = 2\pi/Tt$ dimensionless depth resolution: $0.25 * ls = \sqrt{\kappa/\rho}$

March 10, 2009

binary temperature output intx2 fn, NTslabs; double JD; floatx2 dt, dx; for each facet int facetindex; floatx3 lambda, beta, area; floatxNTslabs Temps;

March 15, 2009

Finally I made it! Rigorous adaptive time step in the differential equation of heat diffusion. It is built in within the DiffuseHeat function of the tpm.c file. The time step is varied during the heat diffusion such that the max surface temperature variation between the current and the next time step is kept below a threshold ΔT: namely:

$$dt < \frac{\Delta T}{2} \left| \frac{\omega}{dX^2}(T_{X=1} - T_{X=0}) - \frac{\sqrt{\omega}}{\Gamma dX}(\epsilon\sigma T_{X=0}^4 - c_3 F) \right|^{-1} \quad (1)$$

The value of the threshold is by default = 1K, but it can be adjusted through the switch -t call runtpm -h for further datails.

May 29, 2009

The initial value of the temperature is set at position given by the first ephemeris point: a full rotation of the body with $\Gamma = 0$ is performed and the average temperature of each facet is calculated ($\overline{T_i}$). The latter value is then assigned to teh facet and to all slabs in the sub-soil.

The "-i" switch can be used to force all facets to have an initial temperture: e.g.

```
runtpm -i 100.0
```

sets the initial temperature to 100 K for all facets.

This option is very handy to check the model convergence. I suggest to perform at least two runs, one with automatic initialization of the temperature and a second one with facets

temperature forced to a user value. Then a generation of a tempertaure map is required at be beginning of the observation set, and you test that the temperature maps of the two runs are very close.

Here it is:
first run:

```
runtpm -f 100 < tpm.ini
mv tmap_temps.dat tmap_temps.1.dat
```

Second run:

```
runtpm < tpm.ini
mv tmap_temps.dat tmap_temps.2.dat
```

Then you check that the temperatures in the two tmaps file are pretty close.

September 12, 2009

Uses only >0 observed fluxes for the calculation of the χ^2. Interferometric visibilities can now be entered as (-1*visibility) in the .obs file at the place of fluxes. The TPM will calculated the model at the visibility epochs but will not use them for χ^2 calculation.

Comments can now be entered in the observation and ephemerides files. Any line starting with a # in the first column is not red.

10 Applications

10.1 How hot can a NEO get?

We use the program Tss to calculate the subsolar temperature of an asteroid at different heliocentric distances. The program requires: Bond Albedo, beaming parmeter, emissivity, and heliocentric distance to be entered from the stdio.

From [**?**] the average geometric albedo of NEOs S- and C-types are 0.239 and 0.101 respectively. Assuming G=0.15 the corresponing Bolometric Bond's albedos are 0.094 and 0.040, respectively.

11 Using the code from IDL

11.1 example project

```
pro rtpm
  P=6.0/24; rotation period in hour
  a=10^(findgen(55)/20-2)  ; semimajor axes
;  plot, a, a^1.5*365.25, psym=4, /ylog, /xlog
```

```
  window, 0, xsize=600, ysize=600
  plot, [1,0], xra=[-5,5],yra=[-5,5], /nodata, /xstyle, /ystyle

  Gamma=2500.0 ; Thermal inertia SI units

  for i=50,N_ELEMENTS(a)-1 do begin
    if a[i] le 0.5 then N=20
    if a[i] gt 0.5 and a[i] le 1.0 then N=50
    if a[i] gt 1.0 and a[i] le 3.0 then N=250
    if a[i] gt 3.0 then N=500
  ; set up times
    tei=0
    tee=(N+1)*P
    toi=(N)*P
    toe=(N+1)*P
    ts=a[0]^1.5*365.25/360 ; time step of the ephemeris: it makes sure to have 360 point
    tos=P/360

    T=a[i]^1.5*365.25
    M=-360*tee/T MOD 360

; generate ephems
    cmd='echo 0.0 '+string(M, format='(f9.3)')+' 0.0 00.0 0.0 0.0'+$
         string(a[i],format='(f9.4)')+' 0.0 0.0'+ string(tee,format='(f15.8)')+ $
         string(ts,format='(f8.3)')+' | elem2v | awk ''{print $1,-$2,-$3,-$4}'' > eph.ep
    print, cmd
    spawn, cmd
; get and plots ephems
    readcol, 'eph.eph', jd, x,y,z
    oplot, -x,-y,psym=4,symsize=0.3
;    print, [rotate(jd,1),rotate(sqrt(x^2+y^2+z^2),1)]

; generate obs
    cmd='echo '+string(toi,format='(f15.8)')+' '+string(toe,format='(f15.8)')+$
        ' '+string(tos,format='(f15.8)')+' | genobs | awk ''{print $0,1,0,0,1}'' >obs.ob
    print, cmd
    spawn, cmd

; run the TPM
    cmd='echo sphere40.obj eph.eph obs.obs 0.0 90.0 '+$
        string(P*24, format='(f7.3)')+$
        ' 0.0 0.0 360 1.0 '+$
```

```
        string(Gamma,format='(f7.1)')+$
        ' 0.1 45.0 0.0 1.0 0 | runtpm -t 5.0 > tpm.out'
    print, cmd
    spawn, cmd

    cmd='cp tpm.out tpm.'+strcompress(string(a[i],format='(f6.3)'),/remove_all)+'.out'
    spawn, cmd
    cmd='cp tmap_temps.bin tmap.'+strcompress(string(a[i],format='(f6.3)'),/remove_all)+
    spawn, cmd

    wait, 3.0
  endfor
end
```

11.2 Temperature of Marco Polo Targets

The potential targets for the Marco Polo space mission are:

Number	Desigantion	Rotation Period (hrs)
(65679)	1989 UQ	7.7
(162173)	1999 JU3	7.5
-	2001 SG286	7.0
(162998)	2001 SK162	68.0

These objects are primitive types. Thus a low albedo is expected. I assumed p_V=0.101: the average abedo of C-type NEAs [**?**], and G=0.15, implying a bolometric Bond albedo A=0.04.

11.2.1 Equatorial tempertaure of 1999 JU3

First of all we generate a ring shape made of a mesh 36 triangles. All facets have a normal perpendicular to the Z-axis. The normals form angles from 0 to 2π rad with respect to the X-axis. We do this using a specific idl script that we can issue using the bash command (sorry for those using Windows):

```
echo makering,36 | idl > ring.obj
```

The ring mesh can be visualized using programs such as the Blender: `http://www.blender.org/`.

We need to generate the orbit of the body. Beacuse we want the equatorial temperture we build an orbit on the ecliptic plane with the actual semimajor axis (a=1.1896488 AU) and eccentricity (e=0.1903617) Of 99JU3. We generate asteroid positions with respect to the Sun for (500 days) $\sim$ one full orbital period:

```
wcalc "1.1896488^1.5*365.25"
 = 473.935
```

We use the command "elem2v" which calculate the position of a planet given the orbital elements.

```
#    epoch
#        M   Omega  omega i   e         a         JD  JD0 JD1   JDstep
echo 0.0 0.0 0.0    0.0   0.0 0.1903617 1.1896488 0.0 0.0 500.0 1 | elem2v
| awk '{print $1,"\t",-$2,"\t",-$3,"\t",-$4}' > eph.eph
```

The perihelion is at JD=0.0 and at JD=473.935. The aphelion is at JD=236.968. The observations file requied by the TPM will be very simple. It is composed of two lines. One to get the temperature at the aphelion and one at perihelion.

```
echo 236.968 10.0 0.0 0.0 1.0 0.0 0.0 2 > obs.obs
echo 437.935 10.0 0.0 0.0 1.0 0.0 0.0 2 >> obs.obs
```

We are now ready to run the TPM:

```
echo ring.obj eph.eph obs.obs 0.0 90.0 7.7 0.0 0.0
  360 1.0 700.0 0.03 45 0.0 1.0 0 | runtpm
```

We can do a loop to run the TPM with different thermal inertia values:

```
for i in 10 50 200 700 2500; do
  echo ring.obj eph.eph obs.obs 0.0 90.0 7.7 0.0 0.0
  360 1.0 $i 0.03 45 0.0 1.0 0 | runtpm;
  mv tmap_temps.dat tmap_temps.$i.dat;
done
```

The following idl script is then used to plot the tempreatures:

```
pro ploteqtemp
 myprefs
 bgw, /ps
 Gamma=[10,50,200,700,2500]
 fn=['tmap_temps.10.dat', 'tmap_temps.50.dat', 'tmap_temps.200.dat', $
     'tmap_temps.700.dat', 'tmap_temps.2500.dat']

 for i=0, N_ELEMENTS(Gamma)-1 do begin
   readcol, fn[i], JD, d,d,d,l,b,d,t
   i0=where(Jd eq min(JD))
   i1=where(Jd eq max(JD))
   psmode, /start, /color, $
           filename=strcompress(string(Gamma[i]),$
           /remove_all)+'eqtemp.ps'
   plot,  l[i1], t[i1], xrange=[0,360], /xstyle, /nodata, $
          xtitle='Longitude (deg)', $
          ytitle='Temperature (K)',$
          title='Equatorial Temperature: 1999 JU3; TI='+$
          string(Gamma[i], format='(I4)')
   oplot,  l[i1], t[i1],color=1, psym=-4
   oplot, l[i0], t[i0],color=3, psym=-5
   legend,['Perilehion','Aphelion'],psym=[4,5], color=[1,3]
   psmode,/stop
 endfor
end
```

12 Calculation of interferometric visibilities from TPM

Calculation of a map of the thermal infrared emission at several wavelengths of a body spatially resolved. We use the TPM. Parameters are A, Γ, and surface roughness: i.e: γ_c, ρ_c.

13 Calculation of the apparent V magnitude of the asteroid

In the file /tpm/runtpm.c was added a further flag -p, eg the asteroid geometric albedo (the name of the new variables is palbedo). If the user put a numerical value after the -p, the software consider the "scale factor" given in the standard input to be the object average diameter expressed in km; in this case the mesh is rescaled to the volume of the equivalent sphere and the given albedo is used. The rescale of the mesh is made according to the formula (see line 124 of runtpm.c):

$$l = R\sqrt[3]{\frac{4\pi}{3V_{mesh}}} \tag{2}$$

Here, R is the effective radius of the asteroid. The albedo is an input value of the function runlc (see line 311 of runtpm.c), whose explicit expression is inside the file /tpm/lc.c. In this file, for each scattering model (eg Geometric, Lambert, Lommel-Seeliger and Kaasalainen), the apparent asteroid V magnitude is calculated in accordance with the formula:

$$m - m_{sun} = -2.5 \log_{10}\left(\frac{p_v S\phi(\alpha) a^2}{\pi \Delta^2 r^2}\right) \tag{3}$$

In this equation $m_{sun} = 26.74$ is the Sun mean apparent magnitude from Earth in the V band, whose value is defined in the constant "SUNMA" inside the file /tms/constants.h, while $S\phi(\alpha)$ is the flux value from the surface of the asteroid calculated by the scattering model adopted (α is the phase angle).

References

[Carbognani et al., 2012] Carbognani, A., Tanga, P., Cellino, A., Delbo, M., Mottola, S., and Marchese, E. (2012). The representation of asteroid shapes: A test for the inversion of Gaia photometry. *Planetary and Space Science*, 73(1):80–85.

[Delbo et al., 2007] Delbo, M., dell'Oro, A., Harris, A. W., Mottola, S., and Mueller, M. (2007). Thermal inertia of near-Earth asteroids and implications for the magnitude of the Yarkovsky effect. *Icarus*, 190(1):236–249.

[Delbo and Harris, 2002] Delbo, M. and Harris, A. W. (2002). Physical properties of near-Earth asteroids from thermal infrared observations and thermal modeling. *Meteoritics & Planetary Science*, 37(12):1929–1936.

[Delbo et al., 2003] Delbo, M., Harris, A. W., Binzel, R. P., Pravec, P., and Davies, J. K. (2003). Keck observations of near-Earth asteroids in the thermal infrared. *Icarus*, 166(1):116–130.

[Delbo et al., 2015] Delbo, M., Mueller, M., Emery, J. P., Rozitis, B., and Capria, M. T. (2015). Thermophysical modeling of asteroid surfaces. In *Asteroids IV*, pages 1–21.

[Delbo and Tanga, 2009] Delbo, M. and Tanga, P. (2009). Thermal inertia of main belt asteroids smaller than 100 km from IRAS data. *Planetary and Space Science*, 57(2):259–265.

[Durech et al., 2015] Durech, J., Carry, B., Delbo, M., Kaasalainen, M., and Viikinkoski, M. (2015). Asteroid Models from Multiple Data Sources. In *Asteroids IV*. University of Arizona Press, Tucson, AZ.

[Durech et al., 2009] Durech, J., KAASALAINEN, M., Warner, B. D., Fauerbach, M., Marks, S. A., Fauvaud, S., Fauvaud, M., Vugnon, J. M., Pilcher, F., Bernasconi, L., and Behrend, R. (2009). Asteroid models from combined sparse and dense photometric data. *Astronomy and Astrophysics*, 493(1):291–297.

[Harris, 1998] Harris, A. W. (1998). A Thermal Model for Near-Earth Asteroids. *Icarus*, 131(2):291–301.

[Harris and Lagerros, 2002] Harris, A. W. and Lagerros, J. S. V. (2002). Asteroids in the Thermal Infrared. *Asteroids III, W. F. Bottke Jr., A. Cellino, P. Paolicchi, and R. P. Binzel (eds), University of Arizona Press, Tucson.*, pages 205–218.

[Mueller, 2007] Mueller, M. (2007). *Surface Properties of Asteroids from Mid-Infrared Observations and Thermophysical Modeling*. PhD thesis, PhD thesis on arXiv.org.

[Rozitis et al., 2013] Rozitis, B., Duddy, S. R., Green, S. F., and Lowry, S. C. (2013). A thermophysical analysis of the (1862) Apollo Yarkovsky and YORP effects. *Astronomy and Astrophysics*, 555:A20.

[Rozitis and Green, 2011] Rozitis, B. and Green, S. F. (2011). Directional characteristics of thermal-infrared beaming from atmosphereless planetary surfaces - a new thermophysical model. *Monthly Notices of the Royal Astronomical Society*, 415(3):2042–2062.

[Rozitis and Green, 2014] Rozitis, B. and Green, S. F. (2014). Physical characterisation of near-Earth asteroid (1620) Geographos. Reconciling radar and thermal-infrared observations. *Astronomy & Astrophysics*, 568:A43.

[Rozitis et al., 2014] Rozitis, B., MacLennan, E., and Emery, J. P. (2014). Cohesive forces prevent the rotational breakup of rubble-pile asteroid (29075) 1950 DA. *Nature*, 512(7):174–176.