

2- \mathbb{D} _{UST} Users' Manual

Toshiya Ueta

Royal Observatory of Belgium, Ringlaan, 3, B-1180, Brussels, BELGIUM; ueta@oma.be

Margaret Meixner

*Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218, USA;
meixner@stsci.edu*

1. Introduction

The 2- \mathbb{D} _{UST} code¹ attempts to solve the radiative transfer equation within an axisymmetric dusty system following the principle of long characteristics. The iterative method to find a solution of radiative transfer is based on the scheme elucidated by Collison & Fix (1991), which is derived from the assumption that luminosity (energy) is conserved at each radial grid (i.e., shell) from the surface of the central source to the outermost shell of the axisymmetric system. The predecessor of the code, *DONUT*, was originally written in 1991 by Alan Collison in Fortran 77, and then was specifically tailored by Chris Skinner for the case of post-asymptotic giant branch (AGB) stars and very young planetary nebulae (PNs). The *donut* code had been maintained by Chris Skinner, Vince Mannings, Margaret Meixner, and Toshiya Ueta, during which a volume of results had been published (e.g., Meixner et al. 1997; Skinner et al. 1997; Skinner, Meixner, & Bobrowsky 1998; Ueta et al. 2001a,b).

In 2001, the code was completely rewritten in Fortran 90 by Toshiya Ueta in order to allow dynamic memory allocation to parameter arrays. Also considered was the possibility of future parallelization of the code, given the heavily looped structure of the code. Upon this rewrite, most of the algorithms were replaced with more flexible and efficient counterparts. Especially, the treatment of dust was widely expanded so that the code could handle a distribution of sizes, multiple grain species, and multiple composition layers within the given axisymmetric shell. As a result, the Fortran90 version of the code, now renamed 2- \mathbb{D} _{UST}, has become a generalized 2.5-dimensional radiative transfer code that can be used to investigate the intensity and temperature structures in any user-specified, axisymmetric dusty systems under the assumption of radiative equilibrium.

This manual explains how to use the code. We describe first how to compile the executables in §2, and then, how to run the code in §3, illustrating the input parameters and discussing some

¹The 2- \mathbb{D} _{UST} code is publicly available from the 2- \mathbb{D} _{UST} distribution page at <http://www003.upp.sonet.ne.jp/ueta/research/2dust/> or <http://homepage.oma.be/ueta/research/2dust/>.

subtleties concerning those parameters. Appendices are also provided to show an example run of the executables. A detailed discussion of the 2- \mathbb{D} _{UST} code itself has been given in Ueta & Meixner (2003). Many caveats may be found in published papers describing models constructed with the 2- \mathbb{D} _{UST} code (e.g., Meixner et al. 2002; O’Hara et al. 2003; Meixner et al. 2004; see also discussions in Ueta et al. 2001a).

2. Compiling the 2- \mathbb{D} _{UST} Code

The distribution of the code comes as a single gzipped tar file, 2dust.YYMMDD.tgz, which can be downloaded from the code distribution WWW page. The YYMMDD part indicates the distribution date of the particular package by two-digit year, month, and date. By unpacking the distribution, you will get a set of Fortran 90 program units accompanied by a simple Makefile. Fig. 1 shows an example of unpacking a distribution package.

```
prompt> tar zxvf 2dust.040406.tgz
-rw-r--r-- ueta/users      2430 2004-02-18 11:56:44 bbtabgen.f90
-rw-r--r-- ueta/users      5042 2004-02-18 11:58:26 bbtabgen2.f90
-rw-r--r-- ueta/users      3139 2004-02-06 13:16:04 const.f90
-rw-r--r-- ueta/users        313 2003-11-06 11:02:27 defvars.f90
-rw-r--r-- ueta/users        679 2004-02-18 18:37:07 dfunc.f90
-rw-r--r-- ueta/users     49605 2004-02-06 15:40:59 dustprep.f90
-rw-r--r-- ueta/users      2308 2004-02-18 11:55:08 extrap.f90
-rw-r--r-- ueta/users      2964 2003-11-26 16:38:53 extrap2.f90
-rw-r--r-- ueta/users    1030288 2004-04-07 14:41:01 f90fitsio.f90
-rw-r--r-- ueta/users      1569 2003-11-06 11:02:27 gauleg.f90
-rw-r--r-- ueta/users     15542 2004-02-17 15:12:31 geom.f90
-rw-r--r-- ueta/users       945 2003-11-06 11:02:27 getrho.f90
-rw-r--r-- ueta/users     14873 2003-12-10 19:51:55 gridgen.f90
-rw-r--r-- ueta/users      1393 2003-12-10 19:18:05 locate.f90
-rw-r--r-- ueta/users     26415 2004-04-07 16:01:59 mapgrid.f90
-rw-r--r-- ueta/users     67754 2004-02-19 16:00:26 mapspec.f90
-rw-r--r-- ueta/users      2921 2003-11-06 11:02:27 mie.f90
-rw-r--r-- ueta/users     3569 2003-11-06 11:02:27 mindens.f90
-rw-r--r-- ueta/users       594 2003-11-06 11:02:27 phase.f90
-rw-r--r-- ueta/users       518 2003-11-06 11:02:27 phase_iso.f90
-rw-r--r-- ueta/users       594 2003-11-06 11:02:27 phase_mHG.f90
-rw-r--r-- ueta/users      685 2003-12-12 16:25:00 prcsnd.f90
-rw-r--r-- ueta/users      686 2003-12-12 16:24:46 prcsns.f90
-rw-r--r-- ueta/users     11427 2003-12-10 19:58:38 shmass.f90
-rw-r--r-- ueta/users      1927 2003-11-06 11:02:27 spline.f90
-rw-r--r-- ueta/users     1270 2003-11-26 16:34:00 splint.f90
-rw-r--r-- ueta/users      1583 2003-12-17 18:54:59 temp.f90
-rw-r--r-- ueta/users    116664 2004-02-19 10:12:38 twodust.f90
-rw-r--r-- ueta/users      4719 2004-04-07 16:21:25 Makefile
```

Fig. 1.— Unpacking a 2- \mathbb{D} _{UST} distribution.

The distribution is a collection of Fortran90 program units, and the main units are 2- \mathbb{D} _{UST}

itself (twodust.f90) and two auxiliaries, MAPSPEC (mapspec.f90) and MAPGRID (mapgrid.f90)² The main program unit, twodust.f90, computes the temperature and intensity fields within a given axisymmetric system. The auxiliaries, ‘mapspec.f90’ and ‘mapgrid.f90’, respectively generate (1) the spectral energy distribution (SED) and intensity and optical depth maps (in polar grids) projected with respect to a given inclination angle (measured from the pole) and (2) rectified intensity and optical depth maps for a given pixel scale in FITS.

A simple ‘Makefile’ is provided to compile the code. After setting compiler options in Makefile to your liking, executing appropriate make commands will produce executables. The main code can be compiled in both single and double precision (single precision is usually enough and is faster to run), but the auxiliaries have to be compiled in double precision. In the previous distributions, precision had to be manually set in the ‘prcsn.f90’ subroutine. This is no longer the case: there are now two separate subroutines ‘prcsns.f90’ and ‘prcsnd.f90’ and the appropriate one is automatically selected. Since precision may be different for the particular program unit to be compiled, the ‘make clean’ command is also provided to delete old intermediary object files before compiling an executable. The following table summarizes the available make commands and resulting executables:

Program Unit	Program	Make Command	Executable	Note
2- \mathbb{D} _{UST} (single precision)	twodust.f90	make 2dust	2dust.exe	
2- \mathbb{D} _{UST} (double precision)	twodust.f90	make 2dustd	2dustd.exe	
MAPSPEC	mapspec.f90	make mapspec	mapspec.exe	
MAPGRID	mapgrid.f90	make mapgrid	mapgrid.exe	
		make clean		delete old obj. files
2- \mathbb{D} _{UST} (single precision)	twodust.f90	make 2d	2dust.exe	clean + 2dust
2- \mathbb{D} _{UST} (double precision)	twodust.f90	make 2dd	2dustd.exe	clean + 2dustd
MAPSPEC	mapspec.f90	make ms	mapspec.exe	clean + mapspec
MAPGRID	mapgrid.f90	make mg	mapgrid.exe	clean + mapgrid

Table 1: A summary of four program units in the 2- \mathbb{D} _{UST} distribution.

The ‘const.f90’ subroutine defines some constants that are used in calculations, and some of the constants are of particular importance for users. The lower part of the ‘const.f90’ subroutine lists the “user-specified parameters”, and they are described below. *CONDITION* sets the convergence condition. 2- \mathbb{D} _{UST} forces the luminosity constancy at each radial grid to derive a recursive relation to find the solution of the radiative transfer. When the fractional change between luminosities of the current and previous iterations at each radial grid is less than *CONDITION*, the current model is thought to have been converged and the iteration is terminated. *NSIZE* defines the number of

²Note: qedit.f90 and qmake.f90 are removed from the distribution since the 031212 release - these auxiliary codes are still in development.

grids in the size space. The temperature of dust grains will be followed for those *NSIZE* grain sizes. The exact sizes will be internally determined, and optical properties will be determined for each size. Therefore, the execution time is proportional to *NSIZE* when following dust temperature for each size in the expanded temperature/quantum heating mode. The last three parameters, *NGRID*, *TBOT*, and *TTOP*, controls the size and range of the temperature to κB look-up table. In 2- \mathbb{D}_{DUST} , radiative equilibrium is assumed and look-up tables are used to set dust temperatures and corresponding intensities (in the form of $\kappa_{\nu} B_{\nu}$). *NGRID* defines the size of the look-up table, while the range is set by (*TBOT* (the minimum temperature) and *TTOP* (the maximum temperature)). By default, *NGRID* = 100, *TBOT* = 2.7, and *TTOP* = 1000.0. If the dust temperature is expected to go beyond the range imposed by these parameters, they need to be modified accordingly. If *TTOP* is not properly set, for example, the local dust temperature remains to be *TTOP* even when the local intensity becomes large enough to warm up dust grains higher than *TTOP*.

Also note that the distribution may include more than one density functions (*dfunc_xxx.f90*) and phase functions (*phase_xxx.f90*). Chose one which would fit the best to your purposes, or create one to serve your purposes. These subroutines (i.e., your choice of density and scattering phase functions) have to be respectively named as *dfunc.f90* and *phase.f90*.

3. Using the 2- \mathbb{D}_{DUST} Code

3.1. Running 2DUST.EXE

“2dust.exe” (compiled from ‘twodust.f90’) performs the main radiative transfer calculations. In order to run the executable, a number of input parameter files have to be prepared. These parameter files will be discussed first in the following. Also discussed are some factors that can influence the choice of parameters. Some caveats based on our past experience with the code will also be offered at the end of the section.

3.1.1. DATAFILES.DAT

The source of all input/output file names for 2dust.exe is a file called DATAFILES.DAT that lists an integer and 6 file names. Shown below is an example.

The integer at the beginning of the file is the interactive mode switch (*IOFLAG*) that turns on or off the interactive mode during the execution of 2- \mathbb{D}_{DUST} . When *IOFLAG* is set to 0, 2- \mathbb{D}_{DUST} can be executed interactively. In this mode, users can check input values and some physical quantities derived from the given input values (such as the inner density, cross sections, and the stellar luminosity). This mode is useful, for example, when making a new cross sections file after changing dust size and composition. When *IOFLAG* is set to 1, on the other hand, 2dust.exe is executed non-interactively and all the input parameters are directly read from the files without

```

0                               ! Interactive mode switch  [IOFLAG]
shell                           ! Input parameter file name [FNAME]
specgrid                        ! Wavelength grid file name [SPEC]
quad45                          ! Quadrature file name     [QUAD]
shell_stuff                     ! Default output file name [STUFF]
dprop                           ! Dust properties file name [PROP]
xsec                            ! Cross sections file name  [XSEC]

```

Fig. 2.— DATAFILES.DAT

any messages for the users. The non-interactive mode is useful when running the code in a queue mode, for example.

Following *IOFLAG*, the next 6 lines refer to names of the input/output files, whose length must not exceed 21 characters for historical reasons. The 6 file names specified in DATAFILES.DAT are just “roots” of the real file names that are assumed to have the ‘.dat’ suffix. Therefore, following the above example, 2dust.exe will actually read input parameters from the file ‘shell.dat’ (from *FNAME*), define the wavelength grid as specified in the file ‘specgrid.dat’ (from *SPEC*), find its θ -directional quadrature data in the file ‘quad.dat’ (from *QUAD*), write the default output information to the file ‘shell_stuff.dat’ (from *STUFF*), retrieve dust property information from the file ‘dprop.dat’ (from *PROP*), and write/read cross sections for the model to/from the file ‘xsec.dat’ (from *XSEC*).

At the end of the run, default output information will be written in ‘shell_stuff.dat’ and other calculated quantities will be written in three separate output files, whose names are derived by adding the following three suffixes, ‘_dat.dat’, ‘_datf.dat’, and ‘_dats.dat’, to *FNAME*. Therefore, users will get four output files, shell_stuff.dat, shell_dat.dat, shell_datf.dat, and shell_dats.dat in the above example. The cross sections file (*XSEC*) can be either an input or an output file depending on the availability of cross sections appropriate for the particular set of dust properties of the model (see §3.1.8).

3.1.2. Input Parameter File

The input parameter file (*FNAME*) contains the information concerning the computational grid, model geometry. An example input parameter file is listed below:

In this particular model, the computational polar grid has 45 grid points (*NRAD*) along the radial direction (*R*) and 8 grid grid points (*NQ*) along the latitudinal direction (Θ). The radial grids are log-spaced between the inner shell radius (R_{\min}) and the outer shell radius (R_{\max}). The latitudinal grid points are Gaussian quadrature points between $\Theta = 0^\circ$ (pole) and $\Theta = 90^\circ$ (equator). We assume symmetry with respect to the equator.

The next two parameters are related to line integrations along each characteristic in calculating

```

45      8                !  NRAD, NQ
1000    0.15            !  MXSTEP, VSPACE
0       0              !  DFLAG, SFLAG, BFLAG
417.0   6750.0         !  Rstar [Rsun], Tstar [K]
6.0     34.0           !  DISTANCE [kpc]
34.0    4.0  2.0  3.0  0.5  0.5  1.0 !  VELOCITY [km/sec]
4.0     2.0  3.0  0.5  0.5  1.0 !  A, B, C, D, E, F
0.20    12             !  Tau0, NO
1.50    50.0  5.0     !  Rmin [arcsec], Rmax/Rmin, Rsw/Rmin
4        4             !  Nlayer
5.0     10.0  20.0    !  Rlayer1, Rlayer2,... [Rmin]

```

Fig. 3.— An example for the input parameter file, ‘shell.dat’

angle-integrated intensity ($\int I_\nu d\omega$) available at a given integration point. The parameter *MXSTEP* sets the maximum number of steps allowed in line integration along each characteristic. The parameter *VSPACE* adjusts the step size with which the code actually does line integration: local dust emission and absorption is calculated at each line integration step. Ideally, the step size for line integrations should be sufficiently smaller than the local photon mean free path. In 2- \mathbb{D}_{DUST} , we set the smallest line integration step size based on the smallest local mean free path length ($l_{\text{mfp}} = (\kappa_{\text{max}}\rho_{\text{max}})^{-1}$) in the shell (actually, we use the value *much* smaller than the minimum l_{mfp} , which is derived by multiplying a model dependent factor, typically of the order of 10^{-3} , to l_{mfp}). With the parameter *VSPACE*, we can control this value. Since a *reciprocal* of *VSPACE* is multiplied to the minimum step size, a larger *VSPACE* value would result in a smaller minimum integration step size. In this particular example (Fig. 3), we can do as many as 1000 integrations along each characteristic using steps whose size is roughly 0.005 of the smallest photon mean free path at that radius. In practice, we must use the smallest *MXSTEP* possible in order to make the code runs as efficient as possible, since *MXSTEP* practically defines the array length for the long characteristics and too large a value would increase the memory access time upon execution of the code (see Appendix A for more details). Needless to say, the goodness of this “line integration template” is the key to produce good models with 2- \mathbb{D}_{DUST} .

The next parameter, the density flag (*DFLAG*), is used to set which of the units of density is assumed to be continuous in the shell. When *DFLAG* = 0, mass density (g cm^{-3}) is assumed to be continuous, while number density (cm^{-3}) is considered to be continuous when *DFLAG* = 1. This flag is useful to consider different physical conditions in models. For example, the mass density continuity is needed to create a single-composition model with multiple shells of different dust size distributions (i.e., number density is expected to be discontinuous at the size distribution boundary). On the other hand, the number density continuity is a necessary assumption if one follows the heterogeneous nucleation theory when trying multiple composition layer models because grains are thought to change composition on the existing grains and no new grains with new composition are created at the composition boundary (i.e., mass density is expected to be discontinuous at the composition boundary).

The *SFLAG* parameter sets the scattering mode: *SFLAG* = 0 assumes isotropic scattering while *SFLAG* = 1 assumes unisotropic scattering. Obviously, invoking the unisotropic scattering mode significantly increase the execution time, since it means there is one more loop to go through in each iteration. The *BFLAG* parameter can be used to allow the bicone cavity in the density distribution. Unless the density function has the cavity, it is ignored. Otherwise, if *BFLAG* = 0, there is no cavity, whereas if *BFLAG* = 1, there is a cavity. Note that unless you define a bicone cavity in the density function, `dfunc.f90`, enabling *BFLAG* would not make any sense at all. See the discussion on the geometric parameters below for more details.

The next four parameters set the characteristics of the central energy source. In this model, the central star, having radius (R_{star}) of $417.0 R_{\odot}$ and surface temperature (T_{star}) of 6750.0 K , is located 6 kpc (*DISTANCE*) away from the Sun. The shell is assumed to be expanding at 10 km s^{-1} (*VELOCITY*; assumed to be the same for the AGB wind and superwind). The remaining parameters are related to the dust shell characteristics. The density structure of the shell can be set by users by using users' own density distribution function *DFUNC* (see §3.1.9), and up to 6 parameters are allowed for the density function. (When defining a function with a bicone cavity, the parameter 'F' has to be used to define the opening angle of the cavity, and *BFLAG* must be set to 1.) Here, the six geometric parameters ($A=4$, $B=2$, $C=3$, $D=0.5$, $E=0.5$, and $F=1.0$) will be forwarded to the *DFUNC* function. The density at the inner radius at the equator will be determined by the user specified optical depth (τ_0) at a given wavelength (N_0). Users are to select the wavelength by specifying by the line number (N_0 -th line) and not by the actual wavelength. See §3.1.5 and §3.1.10 below for details.

The shell has the inner shell radius (R_{min}) of $1''5$. In *2-D_{DUST}*, we treat the inner radius of a dust shell to be a measurable quantity via high-resolution mid-IR imaging. By fixing R_{min} dust temperature at R_{min} is immediately determined, and the iterative calculations are performed solely to find intensity and temperature structure within the dust shell. Thus, we can reach convergence relatively quickly. The outer shell radius (R_{max}) and the superwind shell radius (R_{rmsw}) are defined by the ratios $R_{\text{max}}/R_{\text{min}} = 50$ and $R_{\text{sw}}/R_{\text{min}} = 5$, respectively. The superwind is expected to occur near the end of the AGB phase (Iben & Renzini 1983), and thus, is defined in the code by default. Users can define their own density function in `dfunc.f90`, and the use of R_{sw} is an option. To use the superwind radius parameter, it must be used in some way in `dfunc.f90`. The rest of the parameters sets up the composition structure of the shell. In this model, there are 4 layers (N_{layer}) in the shell and the 3 layer boundaries (R_{layer}) are located at 5, 10, and 20 R_{min} . Following N_{layer} , users need to list $N_{\text{layer}} - 1$ number of R_{layer} 's because for the shell with N_{layer} layers there should be $N_{\text{layer}} - 1$ boundaries. When $N_{\text{layer}} = 1$, however, there is no need to list R_{layer} . In all cases, $R(N_{\text{layer}})$ is set to R_{max} .

3.1.3. Optimum Number of Radial Grid Points

This is something of a black art. Using too many radial grid points will make the code run very slowly. The speed of the code decreases approximately as the square of the number of radial grid points. A good indication of whether the number of grid points is large enough comes from the unit 9 output file (fort.9). The second column of the fort.9 file lists the ratio of stellar luminosity to total (reddened star plus dust shell) luminosity for each radial grid point. This ratio should be within a few percent of unity. If the number of radial grid points is too small, then this ratio will differ significantly from unity. Situations where the number of radial grid points may need to be increased include models with a very high optical depth (though this can often be cured by use of the *VSPACE* parameter), and models with lots of scattered radiation. It seems that models with amorphous carbon dust typically need more radial grid points than models with silicate dust.

3.1.4. Optimum Number of Latitudinal Grid Points

For most typical PPN it seems that this parameter can be set to 6: in fact it was hardwired at 6 in the original version of the code, *DONUT*. For very highly flattened systems (looking more like disks than donuts) a larger number will be needed. Also, models where there is a low density bicone will generally need at least 8 quadrature points, because it is necessary to include a quadrature point just either side of the resulting density discontinuity. In models with a bicone, if luminosity conservation errors appear, it may be worth experimenting with this parameter.

3.1.5. Wavelength Grid File

The wavelength grid file defines the wavelength grid for the model. Wavelengths are used to calculate dust cross section and in the outputs. However, actual radiative transfer calculations are done in frequency units (i.e., I_ν). The conversion is done within the code. The λ grid file have to list specified wavelengths in microns in each line and any number of wavelength grid point can be listed. Typically, the wavelengths at and between any significant dust features are needed and about 30-40 wavelength grid points are recommended for practical reasons. An example wavelength grid file in Fig. 4.

In this example, 27 wavelength grid points are used in the model. As mentioned earlier, the *NO* parameter in the ‘shell.dat’ file specifies the line number (*NO*-th line) of the wavelength grid file. Therefore, in this example, we have set the optical depth along the equator at $9.7\mu\text{m}$ (the 12th wavelength) to be 0.2 in the input parameter file (see Fig. 3).

Important things to bear in mind when choosing a wavelength grid are;

1. The more wavelength points you choose, the slower the code will run. The computation speed

```
0.20          ! wavelength [microns]
0.41
0.55
0.79
1.25
1.65
2.20
3.00
4.80
7.00
8.80
9.70          ! 12th wavelength [microns] - see line 8 in shell.dat
10.30
11.0
11.8
12.5
15.0
17.4
18.0
20.6
24.0
30.0
40.0
60.0
80.0
100.0
250.0
```

Fig. 4.— An example for the wavelength grid file, ‘specgrid.dat’

is close to linearly proportional to the number of wavelength points in the grid.

2. If there are any gross features in the dust absorption data, such as the well known silicate resonance features at 9.7 and $18\mu m$ and/or polycyclic aromatic hydrocarbons (PAHs) at 6.8, 7.7, 8.6, 11.3, $\dots \mu m$, for initial runs just a few wavelength points in and around each of such feature are sufficient in order to characterize the effect of dust species on the radiation field. For final runs to generate publication quality model spectra, several wavelength points should be included inside each feature in order to properly characterize the shape of the predicted feature as well as its depth. However, it is important to remember that the way cross sections are calculated in 2-D_{DUST} is not optimized to reproduce very narrow dust features.
3. Think about the waveband you need to cover. Don’t include wavelengths “very” much shorter than that at which the central star’s radiation field peaks, because there will never be a significant contribution from radiation at such short wavelengths. On the other hand, if the shortest wavelength included is only “just” shorter than the peak of the stellar SED, some of the stellar flux will be lost and the model will be invalid. Similarly, the longest wavelength point included should be long enough to adequately characterize the emission from the coldest grains at the outer edge of the model. Include enough wavelength points that the code can adequately define the thermal emission as well as the scattering, by grains

at all the temperatures throughout the model. If the coldest grains are at 30K, then stopping the wavelength coverage at $100\mu\text{m}$ will yield incorrect temperatures and radiation fields for those grains.

3.1.6. Quadrature File

In order to determine the local radiation field at a given grid point, one has to consider incoming rays (characteristics) converging toward the grid point from all 4π directions. We discretize the 4π directions and specify the direction by the directional angles, (θ, ϕ) . The θ angle refers to the angle that defines the direction of the characteristics along the plane that goes through the grid point and the center of the geometry (i.e., the location of the central source). The θ angle is measured from the radially outward direction. The ϕ angle refers to azimuthal angle with respect to the radially outward vector (see the left in Fig. 5).

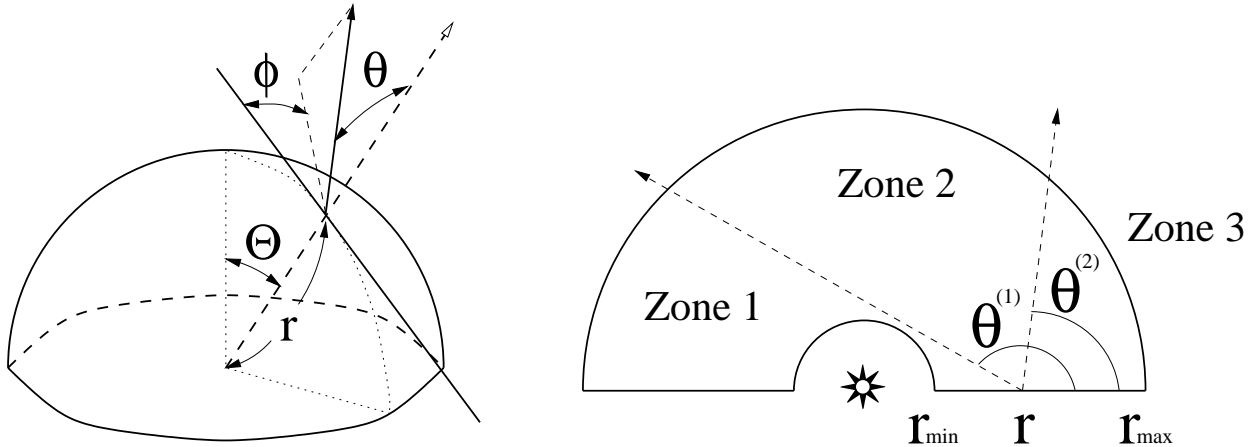


Fig. 5.— [Left] Schematic view of the geometry and the (θ, ϕ) direction of radiation field. [Right] A schematic of the three zones in the θ direction.

The ϕ direction is discretized simply into N_Q directions following the method of Gaussian quadrature. However, for the θ direction, we divide the 2π angles into three zones and perform a Gaussian integration of the radiation field in each of three distinct regions. These three regions are defined by two boundaries as follow: the first (zone 1) includes the region interior to the dust shell, the second (zone 2) the region over which the dust shell is brightest, and the third (zone 3) is ‘everything else’ (see the right in Fig. 5).

The first boundary is rather obvious (the direction between R_{\min} and the first radial grid point), and calculated automatically by the code (angle $\theta^{(1)}$). The second boundary (angle $\theta^{(2)}$) is not so obvious, and must be explicitly defined by the quadrature file. The number of characteristics in each of three θ integration zones and the direction of the second boundary (angle $\theta^{(2)}$) are defined by the quadrature file.

An example quadrature file, 'quad45.dat,' is listed below (Fig. 6): we define NK1, NK2, and NK3 characteristics in zone 1, 2, and 3, respectively, and the angle $\theta^{(2)}$ at each radial grid is also defined by their cosines. The exact angles between the adjacent characteristics will be calculated internally by the code by Gaussian quadrature.

					! line#	NK1	NK2	NK3	COS(Q2)
1	7	2	2	0.939693					
2	7	2	2	0.913447					
3	7	2	2	0.805291					
4	7	2	2	0.709328					
5	7	2	2	0.696230					
6	7	2	2	0.677288					
7	7	2	2	0.558998					
8	7	2	2	0.461716					
9	7	2	2	0.477121					
10	7	2	2	0.516240					
11	7	3	2	0.509069					
12	7	4	2	0.493524					
13	6	4	2	0.496584					
14	5	4	2	0.502482					
15	5	4	2	0.501136					
16	5	4	2	0.499548					
17	5	6	2	0.500380					
18	5	6	2	0.497175					
19	5	6	2	0.495280					
20	5	6	2	0.517866					
21	5	8	2	0.527203					
22	5	8	2	0.386562					
23	5	7	3	0.098667					
24	5	6	4	-0.106204					
25	5	6	4	-0.173117					
26	5	6	4	-0.302789					
27	5	6	4	-0.532174					
28	5	6	4	-0.675457					
29	5	6	4	-0.699935					
30	5	6	4	-0.717552					
31	5	6	4	-0.778478					
32	5	6	4	-0.837312					
33	5	6	4	-0.872789					
34	5	6	4	-0.890890					
35	5	6	4	-0.898425					
36	5	6	4	-0.905971					
37	5	6	4	-0.917230					
38	5	6	4	-0.923478					
39	5	6	4	-0.921403					
40	5	6	4	-0.917012					
41	5	6	4	-0.918013					
42	5	6	4	-0.931446					
43	4	6	5	-0.954422					
44	3	6	2	-0.971668					
45	2	5	1	-0.979069					

Fig. 6.— An example for the quadrature file, 'quad45.dat'

A new quadrature file is necessary if the radial structure of the model changes; in particular if the R_{\max}/R_{\min} parameter or the number of radial grid points is changed. There is no need to

generate a new quadrature file when more latitudinal points are added; this adjustment is performed by the code itself upon execution. The quadrature file included in the test models are generally good for optically thin, marginally equatorially enhanced dust shells. For very optically thick shells, the best course of action may be to increase the number of radial grid points. When modifying the quadrature file, the simplest thing to do may be to interpolate the NK and $\cos(Q2)$ values in the existing file. If the shell is not sampled well, the values in the second column of the fort.9 output deviate from unity. Thus, these values should be used as a guide to judge if further modification is needed on your quadrature file.³

We will then define steps of line integration along each characteristic defined by the quadrature file. The step size is defined based on the local dust density, and is computed (by geom.f90) prior to the main iteration process. Again, *MXSTEP* and *VSPACE* are the two important parameters in the input parameter file as described above.

3.1.7. Dust Properties File

The dust properties file lists information on the dust species to be included in the model such as the bulk density of the grains, the size distribution, the optical properties file names, and so forth. The cross sections for each wavelength is to be computed using MIE theory for spherical particles for the set of dust species specified with given size distributions (each species can have its own size distribution). We assume that dust grains are well mixed in the shell, and for the computational purposes we assume there is only one dust species that has the size and composition averaged property of all the dust species involved. The calculated cross sections are for such fiducial dust species.

2- \mathbb{D} _{UST} allows either one of the two following dust size distributions:

$$n(a) = \begin{cases} a^{-\gamma} & : a_{\min} < a < a_{\max} & \text{(MRN)} \\ a^{-\gamma} e^{-\frac{a}{a_{\max}}} & : a_{\min} < a & \text{(KMH)} \end{cases}$$

where a is the dust size. The former (MRN) is a simple power-law function discussed by Mathis, Rumpl, & Nordsieck (1977) and the latter (KMH) is a power-law plus an exponential fall-off function discussed by Kim, Martin, & Hendry (1994). The difference between the two is the notion of “maximum” dust size. In MRN, there is a clear-cut maximum dust size, while there is no real “maximum” size in KMH. In the KMH distribution, integration in the size space is done actually through infinity, and the input a_{\max} value acts as an “effective” maximum grain size.

When averaging a quantity over the size space, we can use either of the following two averaging methods. One way is to simply average over the size space. For a given quantity f which depends

³It is our hope that another auxiliary program QEDIT will be provided in the future 2- \mathbb{D} _{UST} distribution to help users modify the quadrature file as needed.

on a , the averaged quantity, $\langle f \rangle$ is

$$\langle f \rangle = \frac{\int_{a_{\min}}^{a_{\max}} f(a)n(a)da}{\int_{a_{\min}}^{a_{\max}} n(a)da}. \quad (1)$$

Another way is to average over the size space giving weights *according to the grain surface area*, in which the averaged quantity, $\langle f \rangle$ is

$$\langle f \rangle = \frac{\int_{a_{\min}}^{a_{\max}} f(a)\pi a^2 n(a)da}{\int_{a_{\min}}^{a_{\max}} \pi a^2 n(a)da}. \quad (2)$$

The latter case is based on the discussion given in Harrington, Monk, & Clegg (1988). This is especially a reasonable way of averaging Q values because absorption and scattering can be considered surface processes. Note that when the Harrington averaging is used, the quantities that will be averaged are *only* Q values and grain sizes. Therefore, other quantities (i.e., cross sections and grain masses) will be computed by for example

$$\langle C_{\text{abs}}(\lambda, a) \rangle = \langle Q_{\text{abs}}(\lambda, a) \rangle \pi \langle a \rangle^2, \quad (3)$$

and

$$\langle m_{\text{grain}} \rangle = \frac{4\pi}{3} \langle a \rangle^3 \rho_{\text{grain}}. \quad (4)$$

In general, the former averaging method yields smaller cross sections and larger number of grains and vice versa.

An example dust properties file looks as follows:

```

1 0                                     ! XFLAG, NFLAG
2                                     ! NGTYPE for layer 1
1 3.2 0.50 3.5 0.001 1.0 nk_dorschner95_olmg40.dat ! SD RHO MASSWT GAMMA Amin Amax NKFILE
1 3.2 0.50 3.5 0.001 1.0 nk_dwsuvSil.optical_rev ! SD RHO MASSWT GAMMA Amin Amax NKFILE
2                                     ! NGTYPE for layer 2
2 3.2 0.50 3.5 0.001 1.0 nk_dorschner95_olmg40.dat ! SD RHO MASSWT GAMMA Amin Amax NKFILE
2 3.2 0.50 3.5 0.001 1.0 nk_dwsuvSil.optical_rev ! SD RHO MASSWT GAMMA Amin Amax NKFILE

```

Fig. 7.— An example for the dust properties file, ‘dprop.dat’

The *XFLAG* determines if the cross sections are computed in the next run. When *XFLAG* = 1, MIE computation is performed (dustprep.f90) and results will be written in the file specified by the *XSEC* parameter in DATAFILES.DAT. When *XFLAG* = 0, a previously calculated cross sections will be read from the file specified by the *XSEC* parameter in DATAFILES.DAT.

The *NFLAG* determines which of the two averaging schemes for dust properties will be used and if an expanded temperature/quantum heating option is invoked. When *NFLAG* = 0, dust property averaging is done over the size space (eq. 1), while the Harrington averaging is done when *NFLAG* = 1 (eq. 2). Furthermore, when *NFLAG* = 2 and 3, we invoke an expanded temperature/quantum heating mode. When *NFLAG* = 2, radiative transfer is done as in *NFLAG* = 1 (with the Harrington

averaging), but when a convergence is achieved, we go into the second radiative transfer with expanded dust temperatures for each size for each species involved in the model. In other words, this option allows to follow emission contribution from each dust species. When $NFLAG = 3$, we go into the second radiative transfer as in $NFLAG = 2$, but we also activate the quantum heating option - we examine non-linear heating of small particles and include contribution from such *superheated* dust grains. When this option is invoked, two files, SizeQ.dat and XsecEXP.dat, are created. SizeQ.dat bookkeeps dust sizes and corresponding Q values in and XsecEXP.dat records cross sections for each size and wavelength. (Note: in the present distribution, the quantum heating option is not available yet - just the exnapded temperature mode only.)

When the expanded dust temperature option is invoked ($NFLAG = 2, 3$), the Harrington averaging method is always used. Besides this, the exact algorithms used in averaging the dust properties without and with this option ($NFLAG = 1$ vs. 3) are slightly different due to expansion of the algorithm in the size space. Thus, resulting cross sections are accordingly, however slightly, different. The algorithm was not standardized to the new scheme in order to keep a backward compatibility.

Following these flags, the file lists the number of grain species in the first layer ($NGTYPE$) followed by specific info on each species in that layer. This set of parameters needs to be repeated for each composition layer in the shell. In the present example, there are two layers in the shell, and therefore, compositional information is given twice. Even though the present example has the same grain bulk density and size distribution for each species in each layer, each species can have its own grain density and size distribution.

By making $SD = 1$, $2-DUST$ uses the MRN function and by $SD = 2$ the KMH function. When the MRN function is used, a_{min} and a_{max} define the range of the size distribution and γ sets the power coefficient (the negative sign will be added in the code). When the KMH function is used, a_{min} sets the lower limit of the size distribution and a_{max} sets the exponential factor (the denominator) in the exponential fall-off part of the function. In this case, the size space integration is done from a_{min} to infinity. γ is the power coefficient of the function. Specific info of each species includes type of the size distribution function (SD), bulk density (g cm^{-3}) of the grain species (ρ_{grain}), mass weight of the species ($MASSWT$), the exponential factor in the size distribution ($GAMMA$), the minimum grain radius (a_{min}), the “maximum” grain radius (a_{max} ; see below for details), and the optical properties file name for the species.

ρ_{grain} is the bulk density of the species and $MASSWT$ is the mass weight of the species. The sum of $MASSWT$'s in a layer must be 1. Using these parameters along with the number weights computed from given mass weights, the size and composition averaged single particle mass will be computed for each layer. In the main radiative transfer algorithm, number density at a given location is multiplied by the cross section to get the optical depth. This size and composition averaged single particle mass is needed to obtain the number density from mass density from DFUNC (when $DFLAG = 0$ in the input parameter file).

NKFILE is the file name for each “n & k” file, which should consists of three columns of values, first being wavelength in microns, second being the “n” value at that wavelength, and third being the “k” value at that wavelength. These “n & k” files **bf** MUST cover the entire wavelength range you specified in the “specgrid.file” because the “n & k” values at wavelengths specified in the “specgrid.file” will be interpolated from the “n & k” files you supplied. The warning message will alert you if your n & k files are not covering the appropriate wavelength range.

3.1.8. Cross Sections File

The cross sections file is either generated or required as an input file depending on the *XFLAG* in the dust properties file. The cross sections file consists of a single value *AVGMASS* (the size and composition averaged mass) in g cm^{-3} followed by 4 columns of values being wavelength (μm), absorption cross section, κ (cm^2), scattering cross section, σ (cm^2), and asymmetry parameter, *G* from left to right. This repeats the number of wavelengths, and then, the whole set is repeated for the number of layers. An example of the cross sections file is given in Fig. 11.

1.814385E-18				! AVGMASS [g/cc]
0.20000	2.523282E-14	3.499326E-14	9.959032E-04	! Lambda [um], Kappa [cm^2], Sigma [cm^2], G
0.41000	1.155153E-14	2.246079E-14	2.418522E-04	
0.55000	7.337785E-15	1.874462E-14	1.341939E-04	
0.79000	4.823870E-15	1.396419E-14	6.495807E-05	
1.25000	3.811106E-15	8.729764E-15	2.627764E-05	
1.65000	2.649999E-15	6.765256E-15	1.530687E-05	
2.20000	1.524399E-15	5.080195E-15	8.678982E-06	
3.00000	8.432322E-16	3.359887E-15	4.639289E-06	
4.80000	4.515525E-16	1.413462E-15	1.775660E-06	
7.00000	3.518410E-16	4.152609E-16	7.718468E-07	
8.80000	3.011204E-15	1.082279E-16	4.021249E-07	
9.70000	4.757411E-15	2.291892E-16	3.207295E-07	
10.30000	4.318182E-15	2.694153E-16	3.064319E-07	
11.00000	3.275520E-15	2.758048E-16	3.163554E-07	
11.80000	2.231847E-15	2.509882E-16	3.201516E-07	
12.50000	1.599750E-15	2.106660E-16	2.956266E-07	
15.00000	1.425707E-15	6.798773E-17	1.719883E-07	
17.40000	2.450370E-15	7.438567E-17	1.096953E-07	
18.00000	2.378872E-15	7.617097E-17	1.037853E-07	
20.60000	1.851594E-15	6.821630E-17	8.802672E-08	
24.00000	1.300156E-15	4.950762E-17	7.591690E-08	
30.00000	7.094443E-16	2.611190E-17	6.666441E-08	
40.00000	3.543156E-16	9.161995E-18	4.300617E-08	
60.00000	1.478707E-16	1.787746E-18	2.261001E-08	
80.00000	8.034534E-17	5.371773E-19	-1.977802E-09	
100.00000	5.052310E-17	2.127111E-19	-3.827003E-08	
250.00000	7.883001E-18	5.168662E-21	-1.028005E-07	

Fig. 8.— An example for the cross sections file, ‘xsec.dat’

3.1.9. Density Distribution in the Shell

The density distribution in the axisymmetric shell can be defined by the subroutine `dfunc.f90` as a function of r and Θ with six geometrical parameters (A to F). The distribution includes several density functions, but by default, it is set to the following density function that describe a layered-shell model of post-AGB shell (Meixner et al. 2002; Ueta & Meixner 2003).

$$\rho(r, \Theta) = \rho_{\min} \left(\frac{r}{r_{\min}} \right)^{-B} \left(1 + C \sin^F \Theta \left(\frac{e^{-(r/r_{\text{sw}})^D}}{e^{-(r_{\min}/r_{\text{sw}})^D}} \right) \right) \left(1 + A(1 - \cos \Theta)^F \left(\frac{e^{-(\frac{r}{r_{\text{sw}}})^E}}{e^{-(\frac{r_{\min}}{r_{\text{sw}}})^E}} \right) \right). \quad (5)$$

in which ρ_{\min} is mass/number density (g cm^{-3} or cm^{-3}) at the inner radius ($r = R_{r_{\text{min}}}$) along the equator ($\Theta = 0$). The way ρ_{\min} is determined is discussed in the next section.

This function defines a generally power-law dust density distribution ($B = 2$ corresponding to a constant velocity mass loss), in which an equatorial enhancement is introduced by the sine functions ($\Theta = 90^\circ$ being the equatorial direction) appearing at two places in the function. The degree of the equatorial enhancement is controlled by the parameters A (the equator-to-pole density ratio at r_{\min} is $1 + A$) and C (the power-law density fall-off along the equatorial direction at r_{\min} can be as steep as $B(1 + C)$). The exponential parts following the sine functions are intended to gradually reduce the effects of the equatorial enhancements as radius increases beyond the superwind radius, r_{sw} , in order to make the density distribution spherically symmetric at large radii. The density distribution in the inner shell ($< r_{\text{sw}}$) can be made oblate or prolate by changing the exponential fall-off parameters, D and E . The equatorial enhancement can be made torus-like or disk-like by changing the factor F (larger F makes the distribution more disk-like). We assume azimuthal symmetry and symmetry with respect to the equatorial plane. A schematic representation of the density distribution is given in Fig. 9. A detailed discussion on the effect of each geometrical parameter is given in (Ueta & Meixner 2003).

If users wish to use their own density function, a new `dfunc.f90` subroutine must be made. This subroutine has to be a function of ‘X’ (r) and ‘TH’ (Θ) plus up to 6 geometrical parameters (A to E , and $THCRIT$). Three major radii, R_{\min} , R_{sw} , and R_{\max} , can also be used. For historical reasons, the sixth geometrical parameter is defined as $THCRIT$ in the code but not by F as in the function shown above. If users wish to have a density distribution in which there is a bicone opening along the polar axis, such bicone opening angle should be defined by $THCRIT$, and $BFLAG$ in the input parameter file must be set to 1. $BFLAG = 1$ ensures that the latitudinal sampling near “the bicone cavity wall” is treated properly. Examine the default ‘`dfunc.f90`’ file and modify the function as you see fit. However, be aware that a new quadrature file is likely needed when a new density function is used.

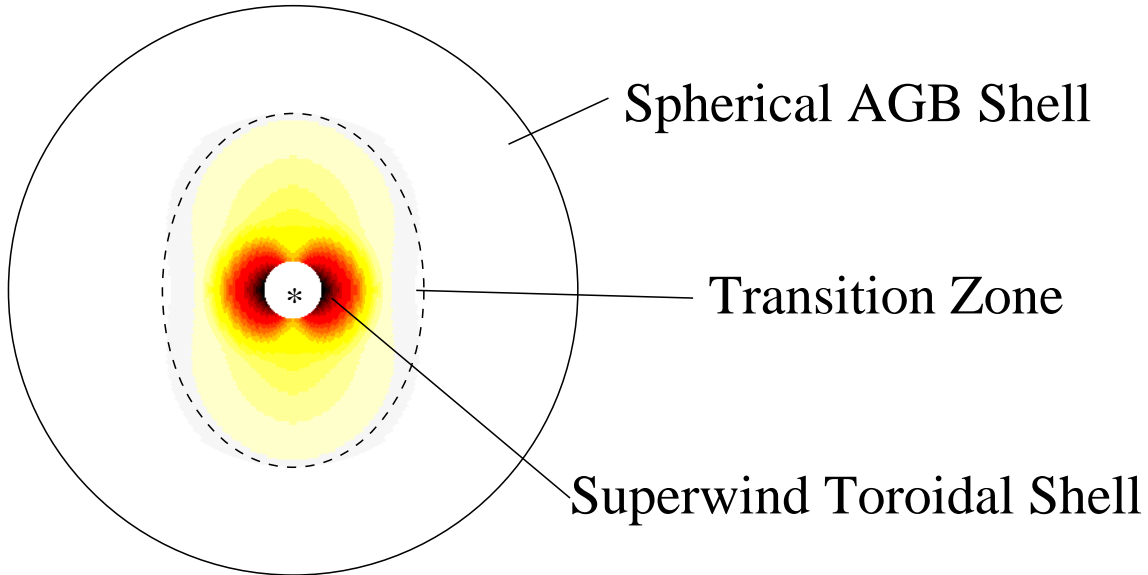


Fig. 9.— Example schematic representations of the default density distribution function.

3.1.10. Density at the Inner Radius

The density at R_{\min} is not an input parameter and has to be computed for a given optical depth at a given wavelength. We need to calculate only the density along the equator, and so we can express the density in terms of the user specified optical depth τ_λ , by

$$\tau_\lambda = \kappa_\lambda \rho_{\min} \int_{R_{\min}}^{R_{\max}} \rho(r, 90^\circ) dr \quad (6)$$

where τ_λ and λ are TAU and the WTAU'th wavelength specified in the geometrical parameters file. κ_λ is the absorption coefficient of the WTAU'th wavelength just calculated by the MIE theory.

3.1.11. Outputs

2dust.exe generates two very large data files, plus a few of smaller files. In the example parameter file given above, the root filename is 'shell'. In this case 2dust.exe will write various miscellaneous accounting information into a file called 'shell_stuff.dat'. For each of its grid of radial and latitudinal points, it will output the reddened stellar intensity into the file 'shell_datf.dat', and the dust shell intensity into 'shell_dats.dat'. The 'shell_datf.dat' file lists the radial distance (in R_{\min}), latitudinal angle (in radians), luminosity at that radial distance (in L_\odot), and dust temperature (in K) for each grid point. For each grid point, also listed are the wavelength (in μm), local mean specific intensity (in $\text{erg s}^{-1} \text{cm}^{-2} \text{Hz}^{-1} \text{sr}^{-1}$), and total specific flux (in $\text{erg s}^{-1} \text{cm}^{-2}$

Hz^{-1}). The ‘shell_dats.dat’ file lists the local mean specific intensity (in $\text{erg s}^{-1} \text{cm}^{-2} \text{Hz}^{-1} \text{sr}^{-1}$), and dust temperature (in K) for each grid point (and for each wavelength). These outputs (only ‘shell_dats.dat’, to be precise) are fed into the auxiliary program `mapspec.exe` (see §3.2).

A variety of useful information on how well the code converged and the quality of the solution are written into the Fortran unit 9 file (typically, `fort.9`). This file lists the ratio of the stellar luminosity to the calculated total (reddened stellar plus dust shell) luminosity (second column) and grain temperatures at the minimum and maximum latitudes (fourth and fifth columns, respectively) at each radial grid (first column). The ratio in the second column in the `fort.9` file is of particular importance because these values basically characterize the ‘goodness’ of the model convergence. The model is considered to have been converged if all these ratios did not change more than a certain value (the `CONDITION` parameter defined in `const.f90`) from the previous iteration. When this happens, actual values of these ratios should be close to unity. If these ratios are far from unity energy may not be conserved properly in the model. Such cases usually imply that models would need more sampling.

3.2. Running MAPSPEC.EXE

“`mapspec.f90`” takes output files generated by `2dust.exe` and produces a spectral energy distribution (SED) and 2-D projected surface brightness and optical depth maps (on a polar grid) for a given inclination angle using the `2dust.exe` output (‘shell_dats.dat’ file in the above example). Upon execution you need to input three values. The first value is the inclination angle of the system measured from the pole ($0 \leq \theta_{\text{incl}} \leq 90^\circ$). The second and third values are the number of radial and angular grids for 2-D projected mapping on to a polar grid. Then, users are prompted to include the central star in the map or not. If you say yes, the star will be included in the map. If you say no, the star will not be included in the map - the central star is simply ignored (the resulting maps will be for the dust shell only). The SED output has the ‘_spec.dat’ suffix and the map output has the ‘_map.dat’ suffix. The root file name is taken from the ‘DATAFILES.DAT’ file and the default 2-D_{DUST} dust shell intensity output file (‘_dats.dat’) is automatically read as input. So, do not change any output file names after running `2dust.exe`. The map output (the ‘_map.dat’ file) can be fed into ‘`mapgrid.exe`’ to generate a rectified ASCII text file for given array size and pixel scale. This ASCII file can then be directly fed into IRAF to create a FITS image.

The SED output file consists of two columns of values, the first being wavelength in microns and the second being flux in Janskys ($1\text{Jy} = 10^{23} \text{erg s}^{-1} \text{cm}^{-2} \text{sr}^{-1} \text{Hz}^{-1} = 1 \text{Jy}$) or in power ($\text{erg s}^{-1} \text{cm}^{-2} \text{sr}^{-1}$). In addition, a file called ‘source_spec.dat’ will be created. This file contains a SED of the source (without extinction by the shell) and users can compare it with the model SED.

When running `mapspec.exe` non-interactively, one has to set the interactive mode flag to 1 in the `datafiles.dat` file and prepare the `getspec.dat` file. This file contains a single line of 5 integers, `ROT`, `NLL`, `NPHI`, `STARFLAG`, and `UNITFLAG`, which are respectively, inclination angle in

degree (ROT), number of radial grid point (of the output map; NLL), number of latitudinal grid point (of the output map; NPHI), a flag whether or not to include the central source in the SED (1 or 0, respectively; STARFLAG), and a flag whether to express the SED in Jy or power (0 or 1, respectively; UNITFLAG). In the example below, we will have an SED of the shell (with the central star) of 80° inclination angle in Jy.

```
80.0    64    31    1    0
```

Fig. 10.— An example for the getspec.dat file.

3.3. Running MAPGRID.EXE

“mapgrid.exe” generates a rectified, 2-D projected surface brightness and optical depth maps for given wavelengths in the FITS format (addition thanks to Ben Sugerman using the FITSIO routines). This is done by converting the mapspec.exe output maps in the polar grid. The final surface brightness maps are in mJy arcsec^{-2} . Upon execution of the code, you will need to supply (1) the input file name (‘_map.dat’), (2) the size of the output image in pixels, (3) the pixel scale of the output image, and (4) the size of the search box to interpolate values in pixels. The first three things are straightforward, however, the last one deserves some more insights.

The way mapgrid.exe works is that first all the existing values of the map in the polar coordinates will be put into pixels in the Cartesian coordinates and then the values in the “empty” pixels will be interpolated using nearby non-empty pixel values. The last parameter defines how big an area you want to search for non-empty pixels in the vicinity of an empty pixel. So, if you make it 1 the entire field size of the map will be searched. If it is 0.5 half the entire field size is searched. Thus, the larger the value the more time it takes to generate a final map. The actual number of non-pixel values used for interpolation will be decided by the code itself and it is typically about 5-10 pixels. Each non-empty pixel value is weighted according to the distance to that pixel and averaged.

The resulting maps are named ‘_mapf_xxxxx.fits’ and ‘_mapt_xxxxx.fits’, respectively for the surface brightness and optical depth maps. The ‘xxxxx’ part stands for 5-digit wavelength in microns. The output maps are oriented in such a way that the near side of the shell is always on the +y side of the field, i.e., the near side of the pole is pointed towards the -y direction.

When running mapgrid.exe non-interactively, one has to set the interactive mode flag to 1 in the datafiles.dat file and prepare the getmap.dat file. This file has to contain the input parameters in the order as you type them in in the interactive mode. The first line holds the name of the mapspec.exe output file. The second line has the number of pixels in x and y direction, separated by a space. The third line holds the pixel scale in arcseconds. The fourth line has the fraction that define the size of the search box to interpolate the pixel value. The fifth line holds the number of

wavelengths for which maps are created. In the following lines, the actual wavelengths have to be specified by the line number in the specgrid file by one wavelength per line. In the example below, we will make maps of surface brightness and optical depth (100×100 pixels at $0''.0975$ pixel scale) at 3rd, 5th, 12th, and 16th wavelengths (i.e., 0.55, 1.25, 9.7, and $12.5 \mu\text{m}$) from ‘shell_map.dat’ file (an output from mapspec.exe) using the default size search box.

```
shell_map.dat
100      100
0.0975  0.0975
0.3
4
3
5
12
16
```

Fig. 11.— An example for the getmap.dat file.

A. An Example Run of 2dust.exe

Below is an example run of 2dust.exe in the interactive mode, using the example model available from the 2-D_{UST} web site. We need to compile the executables.

```
prompt> make 2d
prompt> make ms
prompt> make mg
prompt> ls *.exe
2dust.exe 2dustd.exe mapspec.exe mapgrid.exe
```

Use 'make 2dd' if you want a double precision version of 2-D_{UST}. Now we are ready to start making dust models.

```
prompt> 2dust.exe

*** 2-Dust ***

      AXISYMMETRIC DUST SHELLS
Computation of Grid Point Intensities
in Donut Density Distribution
-----
Based on Collison & Fix (1991)
-----
Modified by CJS, VGM 1993-1995
by TU      1997-
-----

Resolving General Output Filename...
Input file: 07134_stuff.dat
Correct? [Y/N] >> y

Retrieving Geometric Parameters...
Input file: 07134.dat

Correct? [Y/N] >>
      48      8
      322  2.00000E-02
      0      0      0
49.57000      7250.000
2.400000
11.00000
3.000000      2.000000      2.500000      3.000000      3.000000
1.500000
2.500000E-02      18
1.200000      6.950000      3.500000
      1

Correct? [Y/N] >> y

Retrieving Wavelength Grid...
Input file: specgrid.dat
Correct? [Y/N] >> y

Number of wavelength grid : 50
Wavelength # 1 : 1.0000E-01 um
Wavelength # 2 : 2.0000E-01 um
Wavelength # 3 : 3.6500E-01 um
Wavelength # 4 : 4.5000E-01 um
Wavelength # 5 : 5.5600E-01 um
Wavelength # 6 : 6.4000E-01 um
Wavelength # 7 : 7.9000E-01 um
Wavelength # 8 : 1.2500E+00 um
Wavelength # 9 : 1.6500E+00 um
Wavelength # 10 : 2.2000E+00 um
Wavelength # 11 : 3.4500E+00 um
Wavelength # 12 : 3.7500E+00 um
Wavelength # 13 : 4.8000E+00 um
Wavelength # 14 : 7.0000E+00 um
Wavelength # 15 : 8.0000E+00 um
Wavelength # 16 : 8.5000E+00 um
```

```
Wavelength # 17 : 9.7000E+00 um
Wavelength # 18 : 1.0300E+01 um
Wavelength # 19 : 1.1000E+01 um
Wavelength # 20 : 1.1700E+01 um
Wavelength # 21 : 1.2000E+01 um
Wavelength # 22 : 1.2500E+01 um
Wavelength # 23 : 1.2890E+01 um
Wavelength # 24 : 1.5000E+01 um
Wavelength # 25 : 1.7400E+01 um
Wavelength # 26 : 1.7720E+01 um
Wavelength # 27 : 1.8000E+01 um
Wavelength # 28 : 1.9500E+01 um
Wavelength # 29 : 2.0800E+01 um
Wavelength # 30 : 2.1000E+01 um
Wavelength # 31 : 2.1500E+01 um
Wavelength # 32 : 2.3000E+01 um
Wavelength # 33 : 2.4500E+01 um
Wavelength # 34 : 2.5000E+01 um
Wavelength # 35 : 2.7000E+01 um
Wavelength # 36 : 3.0000E+01 um
Wavelength # 37 : 3.1500E+01 um
Wavelength # 38 : 3.3000E+01 um
Wavelength # 39 : 3.6000E+01 um
Wavelength # 40 : 4.2000E+01 um
Wavelength # 41 : 4.8000E+01 um
Wavelength # 42 : 5.5000E+01 um
Wavelength # 43 : 6.0000E+01 um
Wavelength # 44 : 6.5000E+01 um
Wavelength # 45 : 8.0000E+01 um
Wavelength # 46 : 1.0000E+02 um
Wavelength # 47 : 2.5000E+02 um
Wavelength # 48 : 4.5000E+02 um
Wavelength # 49 : 5.0000E+02 um
Wavelength # 50 : 9.0000E+02 um
```

```
Correct? [Y/N] >> y
Retrieving Dust Property Parameters...
Input file: dprop.dat
Correct? [Y/N] >>
```

Here, if *XFLAG* = 1 the cross sections will be computed and you see the following messages associated with dust properties (when *NFLAG* = 0):

```
Retrieving Dust Property Parameters...
Input file: dprop.dat

Correct? [Y/N] >> y
Grain Properties Chart (Units in CGS)
Layer # 1
Dust Grains
Species # 1: zubko96_ac_acar.optc
Species # 2: zubko96_ac_ach2.optc
Species # 3: zubko96_ac_be.optc
Physical Parameters
SD  MassWT  NumWT  GrainMass  Density  a_min  a_max/a_0
MRN 3.400E-01 3.400E-01 8.177E-20 1.800E+00 1.000E-07 1.000E-06
MRN 3.300E-01 3.300E-01 8.177E-20 1.800E+00 1.000E-07 1.000E-06
MRN 3.300E-01 3.300E-01 8.177E-20 1.800E+00 1.000E-07 1.000E-06
Size & composition averaged grain mass: 8.177E-20 g
```

```
Computing cross sections for each layer...
      8.177455E-20
0.10000      2.707664E-14      3.057327E-16      2.115221E-03
0.20000      1.377076E-14      2.639655E-17      5.575866E-04
0.36500      8.196734E-15      6.332703E-18      1.896184E-04
0.45000      5.601627E-15      2.897246E-18      1.346282E-04
0.55600      3.723244E-15      1.266974E-18      9.493406E-05
0.64000      2.866942E-15      7.222770E-19      7.498766E-05
0.79000      1.982158E-15      3.125195E-19      5.261989E-05
1.25000      9.848624E-16      5.112225E-20      2.390654E-05
1.65000      6.852269E-16      1.730244E-20      1.459738E-05
2.20000      4.870694E-16      5.686729E-21      8.728191E-06
3.45000      3.132810E-16      1.006280E-21      3.840522E-06
```

3.75000	2.882515E-16	7.306023E-22	3.301753E-06	4.80000	6.761402E-16	8.287041E-22	9.177845E-06
4.80000	2.318962E-16	2.842212E-22	2.108569E-06	7.00000	5.298738E-16	1.989491E-22	4.419862E-06
7.00000	1.817312E-16	6.823369E-23	1.015456E-06	8.00000	4.813921E-16	1.232033E-22	3.491550E-06
8.00000	1.651033E-16	4.225513E-23	8.021857E-07	8.50000	4.504969E-16	9.859454E-23	3.172400E-06
8.50000	1.545072E-16	3.381503E-23	7.287558E-07	9.70000	3.806075E-16	6.015282E-23	2.611881E-06
9.70000	1.305372E-16	2.063065E-23	6.000796E-07	10.30000	3.579684E-16	4.782124E-23	2.360306E-06
10.30000	1.227727E-16	1.640128E-23	5.419621E-07	11.00000	3.352739E-16	3.700454E-23	2.082519E-06
11.00000	1.149891E-16	1.269147E-23	4.783899E-07	11.70000	3.203390E-16	2.955713E-23	1.863609E-06
11.70000	1.098669E-16	1.013723E-23	4.282696E-07	12.00000	3.138719E-16	2.692660E-23	1.779615E-06
12.00000	1.076489E-16	9.235034E-24	4.088494E-07	12.50000	3.015498E-16	2.305372E-23	1.660970E-06
12.50000	1.034227E-16	7.906750E-24	3.819292E-07	12.89000	2.943843E-16	2.052643E-23	1.555274E-06
12.89000	1.009652E-16	7.039963E-24	3.574435E-07	15.00000	2.518451E-16	1.166593E-23	1.211343E-06
15.00000	8.637546E-17	4.001072E-24	2.782347E-07	17.40000	2.155243E-16	6.678238E-24	9.291635E-07
17.40000	7.391851E-17	2.290440E-24	2.138876E-07	17.72000	2.115172E-16	6.234951E-24	9.009811E-07
17.72000	7.254420E-17	2.138405E-24	2.061348E-07	18.00000	2.085109E-16	5.881195E-24	8.772854E-07
18.00000	7.151312E-17	2.017077E-24	2.016814E-07	19.50000	1.922695E-16	4.342516E-24	7.625794E-07
19.50000	6.594279E-17	1.489356E-24	1.749161E-07	20.80000	1.815292E-16	3.421481E-24	6.746884E-07
20.80000	6.225920E-17	1.173467E-24	1.552277E-07	21.00000	1.797050E-16	3.303642E-24	6.641621E-07
21.00000	6.163353E-17	1.133052E-24	1.531591E-07	21.50000	1.752410E-16	3.029271E-24	6.395779E-07
21.50000	6.010252E-17	1.038951E-24	1.476479E-07	23.00000	1.623602E-16	2.360296E-24	5.679714E-07
23.00000	5.568478E-17	8.095122E-25	1.305081E-07	24.50000	1.507943E-16	1.859934E-24	5.102414E-07
24.50000	5.171803E-17	6.379029E-25	1.176969E-07	25.00000	1.474150E-16	1.723458E-24	4.923470E-07
25.00000	5.055903E-17	5.910954E-25	1.133326E-07	27.00000	1.336473E-16	1.302411E-24	4.307447E-07
27.00000	4.583710E-17	4.466887E-25	1.000994E-07	30.00000	1.205356E-16	8.608575E-25	3.613434E-07
30.00000	4.134017E-17	2.952488E-25	8.329835E-08	31.50000	9.992212E-17	7.518824E-25	3.274718E-07
31.50000	3.427036E-17	2.578736E-25	7.559343E-08	33.00000	9.313148E-17	6.253710E-25	3.049917E-07
33.00000	3.194136E-17	2.144839E-25	6.883273E-08	36.00000	8.884291E-17	4.350906E-25	2.666541E-07
36.00000	3.047051E-17	1.492233E-25	6.053489E-08	42.00000	6.934126E-17	2.412930E-25	2.010453E-07
42.00000	2.378202E-17	8.275641E-26	4.461535E-08	48.00000	5.385110E-17	1.449934E-25	1.586599E-07
48.00000	1.846935E-17	4.972847E-26	3.602650E-08	55.00000	4.232869E-17	8.524175E-26	1.238537E-07
55.00000	1.451750E-17	2.923542E-26	3.009809E-08	60.00000	3.597621E-17	6.068425E-26	1.069498E-07
60.00000	1.233878E-17	2.081292E-26	2.732011E-08	65.00000	3.091710E-17	4.427507E-26	9.104839E-08
65.00000	1.060366E-17	1.518505E-26	1.928547E-08	80.00000	2.047242E-17	1.939180E-26	6.222435E-08
80.00000	7.021441E-18	6.650817E-27	9.367239E-09	100.00000	1.342717E-17	8.009928E-27	3.535174E-08
100.00000	4.605125E-18	2.747171E-27	-1.028204E-08	250.00000	2.702204E-18	2.079394E-28	-2.728051E-08
250.00000	9.267768E-19	7.131711E-29	-8.904333E-08	450.00000	1.044088E-18	1.997913E-29	-7.813779E-08
450.00000	3.580914E-19	6.852255E-30	-1.449574E-07	500.00000	8.750793E-19	1.312193E-29	3.199219E-07
500.00000	3.001266E-19	4.500438E-30	7.026558E-07	900.00000	3.550605E-19	1.258577E-30	-3.048665E-07
900.00000	1.217753E-19	4.316551E-31	-4.855820E-07				

Proceed? [Y/N] >>

Proceed? [Y/N] >>

However, when $NFLAG = 1$, the averaging is done following the Harrington method (eq. 2) and you will get a larger average grain mass (i.e., small number of particles and larger cross sections).

Moreover, when $NFLAG = 2$ or 3 , the averaging algorithm is slightly different.

Grain Properties Chart (Units in CGS)

```

Layer # 1
Dust Grains
Species # 1: zuko96_ac_acar.optc
Species # 2: zuko96_ac_ach2.optc
Species # 3: zuko96_ac_be.optc
Physical Parameters
SD   MassWT   NumWT   GrainMass   Density   a_min   a_max/a_0
MRN  3.400E-01  3.400E-01  2.384E-19  1.800E+00  1.000E-07  1.000E-06
MRN  3.300E-01  3.300E-01  2.384E-19  1.800E+00  1.000E-07  1.000E-06
MRN  3.300E-01  3.300E-01  2.384E-19  1.800E+00  1.000E-07  1.000E-06
Size & composition averaged grain mass: 2.384E-19 g

```

Computing cross sections for each layer...

```

2.384302E-19
0.10000  7.894740E-14  8.914251E-16  9.351298E-03
0.20000  4.015143E-14  7.696447E-17  2.429273E-03
0.36500  2.389923E-14  1.846427E-17  8.241929E-04
0.45000  1.633267E-14  8.447504E-18  5.852896E-04
0.55600  1.085587E-14  3.694119E-18  4.128308E-04
0.64000  8.359145E-15  2.105944E-18  3.261495E-04
0.79000  5.779382E-15  9.112135E-19  2.289169E-04
1.25000  2.871565E-15  1.490572E-19  1.040377E-04
1.65000  1.997917E-15  5.044873E-20  6.353144E-05
2.20000  1.420149E-15  1.658080E-20  3.798909E-05
3.45000  9.134340E-16  2.934011E-21  1.671616E-05
3.75000  8.404553E-16  2.130218E-21  1.437111E-05

```

Grain Properties Chart (Units in CGS)

```

Layer # 1
Dust Grains
Species # 1: zuko96_ac_acar.optc
Species # 2: zuko96_ac_ach2.optc
Species # 3: zuko96_ac_be.optc
Physical Parameters
SD   MassWT   NumWT   GrainMass   Density   a_min   a_max/a_0
MRN  3.400E-01  3.400E-01  2.384E-19  1.800E+00  1.000E-07  1.000E-06
MRN  3.300E-01  3.300E-01  2.384E-19  1.800E+00  1.000E-07  1.000E-06
MRN  3.300E-01  3.300E-01  2.384E-19  1.800E+00  1.000E-07  1.000E-06
Size & composition averaged grain mass: 2.384E-19 g

```

Computing cross sections for each layer...

```

2.384298E-19
0.10000  7.707107E-14  7.897728E-16  9.351298E-03
0.20000  3.979149E-14  7.666456E-17  2.429273E-03
0.36500  2.179057E-14  1.729339E-17  8.241929E-04
0.45000  1.455732E-14  7.799469E-18  5.852896E-04
0.55600  9.524555E-15  3.367111E-18  4.128308E-04
0.64000  7.327877E-15  1.905308E-18  3.261495E-04
0.79000  5.138573E-15  8.178064E-19  2.289169E-04
1.25000  2.692517E-15  1.332761E-19  1.040377E-04
1.65000  1.928381E-15  4.504062E-20  6.353144E-05
2.20000  1.403571E-15  1.483593E-20  3.798909E-05
3.45000  9.123211E-16  2.655711E-21  1.671616E-05
3.75000  8.374735E-16  1.932503E-21  1.437111E-05
4.80000  6.633441E-16  7.606578E-22  9.177845E-06
7.00000  5.048308E-16  1.875145E-22  3.491550E-06
8.00000  4.511439E-16  1.172103E-22  3.172400E-06
8.50000  4.187255E-16  9.415015E-23  2.611881E-06

```

9.70000	3.447693E-16	5.791566E-23	2.611881E-06	11.00000	3.352739E-16	3.700454E-23	2.082519E-06
10.30000	3.196022E-16	4.624695E-23	2.360306E-06	11.70000	3.203390E-16	2.955713E-23	1.863609E-06
11.00000	2.979011E-16	3.584358E-23	2.082519E-06	12.00000	3.138719E-16	2.692660E-23	1.779615E-06
11.70000	2.813403E-16	2.879192E-23	1.863609E-06	12.50000	3.015498E-16	2.305372E-23	1.660970E-06
12.00000	2.743276E-16	2.628425E-23	1.779615E-06	12.89000	2.943843E-16	2.052643E-23	1.555274E-06
12.50000	2.617964E-16	2.254698E-23	1.660970E-06	15.00000	2.518451E-16	1.166593E-23	1.211343E-06
12.89000	2.560659E-16	2.009090E-23	1.555274E-06	17.40000	2.155243E-16	6.678238E-24	9.291635E-07
15.00000	2.125955E-16	1.150145E-23	1.211343E-06	17.72000	2.115172E-16	6.234951E-24	9.009811E-07
17.40000	1.796055E-16	6.614832E-24	9.291635E-07	18.00000	2.085109E-16	5.881195E-24	8.772854E-07
17.72000	1.755332E-16	6.179107E-24	9.009811E-07	19.50000	1.922695E-16	4.342516E-24	7.625794E-07
18.00000	1.726530E-16	5.831078E-24	8.772854E-07	20.80000	1.815292E-16	3.421481E-24	6.746884E-07
19.50000	1.573775E-16	4.313979E-24	7.625794E-07	21.00000	1.797050E-16	3.303642E-24	6.641621E-07
20.80000	1.479744E-16	3.404298E-24	6.746884E-07	21.50000	1.752410E-16	3.029271E-24	6.395779E-07
21.00000	1.461622E-16	3.287687E-24	6.641621E-07	23.00000	1.623602E-16	2.360296E-24	5.679714E-07
21.50000	1.417559E-16	3.016288E-24	6.395779E-07	24.50000	1.507943E-16	1.859934E-24	5.102414E-07
23.00000	1.303544E-16	2.353222E-24	5.679714E-07	25.00000	1.474150E-16	1.723458E-24	4.924370E-07
24.50000	1.202878E-16	1.855417E-24	5.102414E-07	27.00000	1.336473E-16	1.302411E-24	4.307447E-07
25.00000	1.173665E-16	1.719555E-24	4.924369E-07	30.00000	1.205356E-16	8.608575E-25	3.613434E-07
27.00000	1.060627E-16	1.300304E-24	4.307447E-07	31.50000	9.992212E-17	7.518824E-25	3.274718E-07
30.00000	9.377264E-17	8.595468E-25	3.613434E-07	33.00000	9.313148E-17	6.253710E-25	3.049917E-07
31.50000	8.215114E-17	7.493694E-25	3.274718E-07	36.00000	8.884291E-17	4.350906E-25	2.666541E-07
33.00000	7.615326E-17	6.233921E-25	3.049917E-07	42.00000	6.934126E-17	2.412930E-25	2.010453E-07
36.00000	6.987935E-17	4.342482E-25	2.666541E-07	48.00000	5.385110E-17	1.449934E-25	1.586599E-07
42.00000	5.588292E-17	2.405863E-25	2.010453E-07	55.00000	4.232869E-17	8.524175E-26	1.238537E-07
48.00000	4.465108E-17	1.443784E-25	1.586599E-07	60.00000	3.597621E-17	6.068425E-26	1.069498E-07
55.00000	3.609923E-17	8.484108E-26	1.238537E-07	65.00000	3.091710E-17	4.427507E-26	9.104839E-08
60.00000	3.134062E-17	6.038159E-26	1.069498E-07	80.00000	2.047242E-17	1.939180E-26	6.222435E-08
65.00000	2.742893E-17	4.405620E-26	9.104839E-08	100.00000	1.342717E-17	8.009928E-27	3.535174E-08
80.00000	1.911916E-17	1.931143E-26	6.222435E-08	250.00000	2.702204E-18	2.079394E-28	-2.728051E-08
100.00000	1.296308E-17	7.983877E-27	3.535174E-08	450.00000	1.044088E-18	1.997913E-29	-7.813779E-08
250.00000	2.614827E-18	2.076866E-28	-2.728051E-08	500.00000	8.750793E-19	1.312193E-29	3.199219E-07
450.00000	9.786080E-19	1.996454E-29	-7.813780E-08	900.00000	3.550605E-19	1.258577E-30	-3.048665E-07
500.00000	8.156145E-19	1.311284E-29	3.199219E-07				
900.00000	3.216225E-19	1.257950E-30	-3.048665E-07				

Proceed? [Y/N] >>

If you say NO here ('n' or 'N'), the execution terminates and you would obtain the cross sections file (and SizeQ.dat and XsecEXP.dat, when *NFLAG* = 2, 3). IF YES ('y' or 'Y') the 2dust.exe run continues on and the cross sections file will be available at the end of the run.

When *DFLAG* = 0, cross sections will be read from the cross sections file and the run would look like the following:

```
Retrieving Dust Property Parameters...
Input file: dprop.dat
Correct? [Y/N] >> y

Retrieving Cross Sections ...
Input file: xsec.dat

Correct? [Y/N] >> y
2.384302E-19
0.10000 7.894740E-14 8.914251E-16 9.351298E-03
0.20000 4.015143E-14 7.696447E-17 2.429273E-03
0.36500 2.389923E-14 1.846427E-17 8.241929E-04
0.45000 1.633267E-14 8.447504E-18 5.852896E-04
0.55600 1.085587E-14 3.694119E-18 4.128308E-04
0.64000 8.359145E-15 2.105944E-18 3.261495E-04
0.79000 5.779382E-15 9.112135E-19 2.289169E-04
1.25000 2.871565E-15 1.490572E-19 1.040377E-04
1.65000 1.997917E-15 5.044873E-20 6.353144E-05
2.20000 1.420149E-15 1.658080E-20 3.798909E-05
3.45000 9.134340E-16 2.934011E-21 1.671616E-05
3.75000 8.404553E-16 2.130218E-21 1.437111E-05
4.80000 6.761402E-16 8.287041E-22 9.177845E-06
7.00000 5.298738E-16 1.989491E-22 4.419862E-06
8.00000 4.813921E-16 1.232033E-22 3.491550E-06
8.50000 4.504969E-16 9.859454E-23 3.172400E-06
9.70000 3.806075E-16 6.015282E-23 2.611881E-06
10.30000 3.579684E-16 4.782124E-23 2.360306E-06
```

Correct? [Y/N] >>

Then, after the proper quadrature file is read, you are presented with some information regarding the computation grid and some of the physical quantities that are immediately available such as densities at the inner shell radius, shell mass, and mass loss rates. After that, the temperature-Planck function look-up tables and source luminosity and shell parameters are displayed for a final check.

```
Retrieving Quadrature Parameters...
Input file: quad48.dat

Correct? [Y/N] >>
Radial Grid Info
Number of composition layers: 1
Number of radial zones : 48
Layer # 1: zone 1 to 48 ( 12487.91 to 86790.98 R*)

Densities at the inner radius of the shell (Rmin)
on the equator : 1.9690E-21 g cm^-3
along the pole : 4.9224E-22 g cm^-3
Based on Tau = 0.0250 at 10.3000 micron

Physical Quantities of the Shells (dust only)
Mass (Msun) Duration (yrs) Rate (Msun/yrs)
Superwind 2.58714E-04 3.10262E+03 8.33858E-08
AGB 6.46971E-04 4.28161E+03 1.51105E-07

Generating the T & kappa*B(T) tables...

Calculating the luminosity and radii of the object...

Source Luminosity: 6.708505E+03 L_sun.
Source Radius : 3.450072E+12 cm.
Inner Shell Size : 4.308420E+16 cm.
Outer Shell Size : 2.994352E+17 cm.

Proceed? [Y/N] >>
```

Saying YES here will start generating the line integration grid.

```

Generating the T & kappa*B(T) tables...

Calculating the luminosity and radii of the object...

Source Luminosity: 6.708505E+03 L_sun.
Source Radius    : 3.450072E+12 cm.
Inner Shell Size : 4.308420E+16 cm.
Outer Shell Size : 2.994352E+17 cm.

Proceed? [Y/N] >> y

Computing stellar flux at each grid point...

Generating the line integration grid ...
Minimum stepsize      :      2.627930
Maximum stepsize     :    9957.061523
Minimum radial grid size :    21.037109
Max steps ( 322) exhausted 185/ 656 times

Change MXSTEP & VSPACE? [Y/N] >>

```

Here, the message warns you that the *MXSTEP* in the line integration template has been exhausted 60 times out of 606 characteristics. In other words, the present line integration template has failed to sample the entire shell along 60 characteristics. Therefore, we need to either increase *MXSTEP* or decrease *VSPACE* in order to sample the entire shell. Here, we sat YES and try to reduce *VSPACE*.

```

Change MXSTEP & VSPACE? [Y/N] >> y
Enter new MXSTEP ( 322) >> 1000
Enter new VSPACE ( 0.02) >> 0.02

Generating the line integration grid ...
Minimum stepsize      :      2.627930
Maximum stepsize     :    9957.061523
Minimum radial grid size :    21.037109
Max number of steps   :      756

Change MXSTEP & VSPACE? [Y/N] >>

```

This time we have successfully sampled the entire shell with the line integration template. Thus, we don't have to change *MXSTEP* and/or *VSPACE* any more and can proceed. However, if you want to minimize the memory usage during the execution (and save some extra execution time), you can quit out of the run and manually modify *MXSTEP* in the input file and re-start the run. Simply saying NO here would start the main iteration process without the hustle, of course.

```

Change MXSTEP & VSPACE? [Y/N] >> n

Starting the main iteration ...
Iter # 1: Latitudinal Grid 1/ 8

(... iteration continues until the model is converged...)

```

In the interactive mode, messages will keep you informed on the progress of the run until the model converges. The last two numbers indicate how the line integrations are really proceeding in the run. The last number is the "total" number

of characteristics (for all wavelength and θ grids) from the given grid point, (r, Θ) . The second number from the last is the number of times the line integrations really reached the outer radius (R_{\max}). So, in this example run, the line integrations were done along 2904 characteristics from the grid (1,1), and the integration continued to R_{\max} 1685 times. For the rest of 1219 times, the integration did not go to R_{\max} because the optical depth along the characteristics became too high and the integration was terminated there as soon as no more contribution to the local radiation is expected from the grains located in the integration step position.

Alternatively, you can run the model non-interactively. Still, you have to run the model interactively in order to define the optimum *MXSTEP* and *VSPACE* for your model first. After the optimum *MXSTEP* and *VSPACE* are determined, you can quit out of the 2dust.exe run and manually modify the input parameter file. Once this is done, you can set *IOFLAG* = 1 and run the model non-interactively.

B. An Example Run of mapspec.exe

Basically an mapspec.exe run goes similarly as a 2dust.exe run. After the computation of the inner radius density is completed, you need to define some parameters, the inclination angle measured from the pole in degrees and numbers of the radial and angular grid points in the polar-grid surface brightness and optical depth maps. Also, you can choose if you want the central star included in the map or not.

```

prompt> mapspec.exe

*** MAPSPEC ***

AXISYMMETRIC DUST SHELLS
Generate 2-D Projection Maps and SED
from 2-Dust outputs
-----
Created by CJS, VGM 1993-1995
Modified by TU 1995-
-----

Retrieving Geometric Parameters...
Input file: 07134.dat

Correct? [Y/N] >>
      48      8
      756  2.00000000000000D-002
      0      0
49.57000000000000      7250.000000000000
2.4000000000000000
11.0000000000000000
3.0000000000000000      2.0000000000000000      2.5000000000000000
3.0000000000000000      3.0000000000000000      1.5000000000000000
2.50000000000000D-002      18
1.2000000000000000      6.9500000000000000      3.5000000000000000
      1

Correct? [Y/N] >> y

Retrieving Wavelength Grid...
Input file: specgrid.dat
Correct? [Y/N] >> y

Number of wavelength grid : 50

```

```
Wavelength # 1 : 1.000000E-01 um      1.250000E+01  3.015498E-16  2.305372E-23  1.660970E-06
Wavelength # 2 : 2.000000E-01 um      1.289000E+01  2.943843E-16  2.052643E-23  1.555274E-06
Wavelength # 3 : 3.650000E-01 um      1.500000E+01  2.518451E-16  1.166593E-23  1.211343E-06
Wavelength # 4 : 4.500000E-01 um      1.740000E+01  2.155243E-16  6.678238E-24  9.291635E-07
Wavelength # 5 : 5.560000E-01 um      1.772000E+01  2.115172E-16  6.234951E-24  9.009811E-07
Wavelength # 6 : 6.400000E-01 um      1.800000E+01  2.085109E-16  5.881195E-24  8.772854E-07
Wavelength # 7 : 7.900000E-01 um      1.950000E+01  1.922695E-16  4.342516E-24  7.625794E-07
Wavelength # 8 : 1.250000E+00 um      2.080000E+01  1.815292E-16  3.421481E-24  6.746884E-07
Wavelength # 9 : 1.650000E+00 um      2.100000E+01  1.797050E-16  3.303642E-24  6.641621E-07
Wavelength # 10 : 2.200000E+00 um      2.150000E+01  1.752410E-16  3.029271E-24  6.395779E-07
Wavelength # 11 : 3.450000E+00 um      2.300000E+01  1.623602E-16  2.360296E-24  5.679714E-07
Wavelength # 12 : 3.750000E+00 um      2.450000E+01  1.507943E-16  1.859934E-24  5.102414E-07
Wavelength # 13 : 4.800000E+00 um      2.500000E+01  1.474150E-16  1.723458E-24  4.924370E-07
Wavelength # 14 : 7.000000E+00 um      2.700000E+01  1.336473E-16  1.302411E-24  4.307447E-07
Wavelength # 15 : 8.000000E+00 um      3.000000E+01  1.205356E-16  8.608575E-25  3.613434E-07
Wavelength # 16 : 8.500000E+00 um      3.150000E+01  9.992212E-17  7.518824E-25  3.274718E-07
Wavelength # 17 : 9.700000E+00 um      3.300000E+01  9.313148E-17  6.253710E-25  3.049917E-07
Wavelength # 18 : 1.030000E+01 um      3.600000E+01  8.884291E-17  4.350906E-25  2.666541E-07
Wavelength # 19 : 1.100000E+01 um      4.200000E+01  6.934126E-17  2.412930E-25  2.010453E-07
Wavelength # 20 : 1.170000E+01 um      4.800000E+01  5.385110E-17  1.449934E-25  1.586599E-07
Wavelength # 21 : 1.200000E+01 um      5.500000E+01  4.232869E-17  8.524175E-26  1.238537E-07
Wavelength # 22 : 1.250000E+01 um      6.000000E+01  3.597621E-17  6.068425E-26  1.069498E-07
Wavelength # 23 : 1.289000E+01 um      6.500000E+01  3.091710E-17  4.427507E-26  9.104839E-08
Wavelength # 24 : 1.500000E+01 um      8.000000E+01  2.047242E-17  1.939180E-26  6.222435E-08
Wavelength # 25 : 1.740000E+01 um      1.000000E+02  1.342717E-17  8.009928E-27  3.535174E-08
Wavelength # 26 : 1.772000E+01 um      2.500000E+02  2.702204E-18  2.079394E-28  -2.728051E-08
Wavelength # 27 : 1.800000E+01 um      4.500000E+02  1.044088E-18  1.997913E-29  -7.813779E-08
Wavelength # 28 : 1.950000E+01 um      5.000000E+02  8.750793E-19  1.312193E-29  3.199219E-07
Wavelength # 29 : 2.080000E+01 um      9.000000E+02  3.550605E-19  1.258577E-30  -3.048665E-07
Wavelength # 30 : 2.100000E+01 um
Wavelength # 31 : 2.150000E+01 um
Wavelength # 32 : 2.300000E+01 um
Wavelength # 33 : 2.450000E+01 um
Wavelength # 34 : 2.500000E+01 um
Wavelength # 35 : 2.700000E+01 um
Wavelength # 36 : 3.000000E+01 um
Wavelength # 37 : 3.150000E+01 um
Wavelength # 38 : 3.300000E+01 um
Wavelength # 39 : 3.600000E+01 um
Wavelength # 40 : 4.200000E+01 um
Wavelength # 41 : 4.800000E+01 um
Wavelength # 42 : 5.500000E+01 um
Wavelength # 43 : 6.000000E+01 um
Wavelength # 44 : 6.500000E+01 um
Wavelength # 45 : 8.000000E+01 um
Wavelength # 46 : 1.000000E+02 um
Wavelength # 47 : 2.500000E+02 um
Wavelength # 48 : 4.500000E+02 um
Wavelength # 49 : 5.000000E+02 um
Wavelength # 50 : 9.000000E+02 um
Correct? [Y/N] >> y
```

Retrieving Cross Sections ...

```
Input file: xsec.dat
Correct? [Y/N] >> y

2.384302E-19
1.000000E-01  7.894740E-14  8.914251E-16  9.351298E-03
2.000000E-01  4.015143E-14  7.696447E-17  2.429273E-03
3.650000E-01  2.389923E-14  1.846427E-17  8.241929E-04
4.500000E-01  1.633267E-14  8.447504E-18  5.852896E-04
5.560000E-01  1.085587E-14  3.694119E-18  4.128308E-04
6.400000E-01  8.359145E-15  2.105944E-18  3.261495E-04
7.900000E-01  5.779382E-15  9.112135E-19  2.289169E-04
1.250000E+00  2.871565E-15  1.490572E-19  1.040377E-04
1.650000E+00  1.997917E-15  5.044873E-20  6.353144E-05
2.200000E+00  1.420149E-15  1.658080E-20  3.798909E-05
3.450000E+00  9.134340E-16  2.934011E-21  1.671616E-05
3.750000E+00  8.404553E-16  2.130218E-21  1.437111E-05
4.800000E+00  6.761402E-16  8.287041E-22  9.177845E-06
7.000000E+00  5.298738E-16  1.989491E-22  4.419862E-06
8.000000E+00  4.813921E-16  1.232033E-22  3.491550E-06
8.500000E+00  4.504969E-16  9.859454E-23  3.172400E-06
9.700000E+00  3.806075E-16  6.015282E-23  2.611881E-06
1.030000E+01  3.579684E-16  4.782124E-23  2.360306E-06
1.100000E+01  3.352739E-16  3.700454E-23  2.082519E-06
1.170000E+01  3.203390E-16  2.955713E-23  1.863609E-06
1.200000E+01  3.138719E-16  2.692660E-23  1.779615E-06
```

Retrieving Quadrature Parameters...

```
Input file: quad48.dat

Correct? [Y/N] >>
Radial Grid Info
Number of composition layers: 1
Number of radial zones : 48
Layer # 1: zone 1 to 48 ( 12487.91 to 86790.99 R*)

Densities at the inner radius of the shell (Rmin)
on the equator : 1.9690E-21 g cm-3
along the pole : 4.9224E-22 g cm-3
Based on Tau = 0.0250 at 10.3000 micron

Enter the viewing angle (deg) from pole [ 90.00 ] >> 90

Enter the number of radial grids for map [ 64 ] >>

Enter the number of angular grids for map [ 31 ] >>

Include the central star in the map? [y/n] >> y
Including the central star in the map...

Working on the main loop ...
Angular Grid # 1/31 at -81.2903 degrees

(...lines truncated...)

Angular Grid # 31/31 at 92.9032 degrees

Output MAP w/o QH is 07134_map.dat

Output SED w/o QH is 07134_spec.dat

Source SED is source_spec.dat

Saving MAP with the central source.
Saving SED.

Done!
```

Three output files will be created after an mapspec.exe run as shown above. The '07134_map.dat' is the map file which is an input for mapgrid.exe. The other two files, '07134_spec.dat'

and 'source_spec.dat', are spectral SED files (first being model's and second being of only the central source) having two columns of values.

Saving SED.

Done!

prompt> cat 07134_spec.dat

1.000000E-01 1.547402E-07
2.000000E-01 7.230747E-03
3.650000E-01 4.391553E-01
4.500000E-01 1.170803E+00
5.560000E-01 2.169402E+00
6.400000E-01 2.746692E+00
7.900000E-01 3.233818E+00
1.250000E+00 2.885568E+00
1.650000E+00 2.225542E+00
2.200000E+00 1.558026E+00
3.450000E+00 7.901226E-01
3.750000E+00 6.905547E-01
4.800000E+00 4.561518E-01
7.000000E+00 5.354646E-01
8.000000E+00 1.333775E+00
8.500000E+00 2.079825E+00
9.700000E+00 5.020058E+00
1.030000E+01 7.257558E+00
1.100000E+01 1.063023E+01
1.170000E+01 1.485211E+01
1.200000E+01 1.685642E+01
1.250000E+01 2.033932E+01
1.289000E+01 2.353325E+01
1.500000E+01 4.063545E+01
1.740000E+01 6.105407E+01
1.772000E+01 6.349728E+01
1.800000E+01 6.579393E+01
1.950000E+01 7.677030E+01
2.080000E+01 8.616707E+01
2.100000E+01 8.724643E+01
2.150000E+01 8.978206E+01
2.300000E+01 9.662800E+01
2.450000E+01 1.017409E+02
2.500000E+01 1.032362E+02
2.700000E+01 1.069843E+02
3.000000E+01 1.107525E+02
3.150000E+01 1.033229E+02
3.300000E+01 1.010639E+02
3.600000E+01 1.010079E+02
4.200000E+01 8.997038E+01
4.800000E+01 7.567573E+01
5.500000E+01 6.211109E+01
6.000000E+01 5.345470E+01
6.500000E+01 4.586163E+01
8.000000E+01 2.892004E+01
1.000000E+02 1.644773E+01
2.500000E+02 9.898706E-01
4.500000E+02 1.360135E-01
5.000000E+02 9.379640E-02
9.000000E+02 1.242228E-02

The first column lists wavelengths in μm and the second column lists flux in Jy.

With the expanded dust temperature mode on, you will get extra outputs.

Output MAP w/o QH is 07134_map.dat
Output MAP with QH is 07134_map2.dat

Output SED w/o QH is 07134_spec.dat
Output SED with QH (total) is 07134_spec2all.dat
Output SED with QH (each dust) is 07134_spec2each.dat

Source SED is source_spec.dat

Saving MAP with the central source.

The extra outputs are SEDs and maps WITH the expanded dust temperature mode. You would still get SEDs and maps WITHOUT the expanded dust temperature mode. Each column in '07134_spec2each.dat' corresponds to emission from a particular dust species in the order of appearance in the PROP file.

When the expanded dust temperature mode is on, you will get extra files as shown below. In this case, you will need 'fort.21', 'fort.22', 'fort.23', 'SizeQ.dat', and 'XsecEXP.dat' files from your 2-DUST run. Since NSIZE matters here, you will get error messages if you attempt to run mapspec.exe having inconsistent NSIZE values in 2dust.exe and mapspec.exe.

prompt> cat 07134_spec2each.dat

1.000000E-01 0.000000E+00 0.000000E+00 0.000000E+00
2.000000E-01 0.000000E+00 0.000000E+00 0.000000E+00
3.650000E-01 0.000000E+00 0.000000E+00 0.000000E+00
4.500000E-01 0.000000E+00 0.000000E+00 0.000000E+00
5.560000E-01 0.000000E+00 0.000000E+00 0.000000E+00
6.400000E-01 0.000000E+00 0.000000E+00 0.000000E+00
7.900000E-01 0.000000E+00 0.000000E+00 0.000000E+00
1.250000E+00 0.000000E+00 0.000000E+00 0.000000E+00
1.650000E+00 2.424437E-21 7.146880E-31 4.694535E-19
2.200000E+00 9.064048E-15 6.321844E-22 4.743967E-13
3.450000E+00 8.643544E-08 2.748793E-12 1.101318E-06
3.750000E+00 7.623098E-07 5.896039E-11 8.111237E-06
4.800000E+00 1.715877E-04 1.267465E-07 1.095518E-03
7.000000E+00 5.623327E-02 5.274773E-04 2.092960E-01
8.000000E+00 2.470345E-01 4.775668E-03 7.803764E-01
8.500000E+00 4.419598E-01 1.143584E-02 1.279573E+00
9.700000E+00 1.271701E+00 6.153360E-02 3.165193E+00
1.030000E+01 1.926252E+00 1.229428E-01 4.563994E+00
1.100000E+01 2.903321E+00 2.380737E-01 6.674015E+00
1.170000E+01 4.220475E+00 4.379317E-01 9.185157E+00
1.200000E+01 4.838739E+00 5.527444E-01 1.037216E+01
1.250000E+01 6.076023E+00 7.875908E-01 1.224789E+01
1.289000E+01 6.981486E+00 9.944310E-01 1.420958E+01
1.500000E+01 1.287734E+01 2.944721E+00 2.298845E+01
1.740000E+01 1.975437E+01 6.503011E+00 3.261591E+01
1.772000E+01 2.066501E+01 7.113942E+00 3.351232E+01
1.800000E+01 2.152732E+01 7.661041E+00 3.437384E+01
1.950000E+01 2.558336E+01 1.074127E+01 3.813551E+01
2.080000E+01 2.842067E+01 1.356336E+01 4.182275E+01
2.100000E+01 2.868462E+01 1.402075E+01 4.218391E+01
2.150000E+01 2.961108E+01 1.517558E+01 4.264978E+01
2.300000E+01 3.147936E+01 1.830127E+01 4.454724E+01
2.450000E+01 3.311287E+01 2.121172E+01 4.518670E+01
2.500000E+01 3.358172E+01 2.214419E+01 4.530511E+01
2.700000E+01 3.459364E+01 2.514394E+01 4.516401E+01
3.000000E+01 3.521898E+01 2.998764E+01 4.362223E+01
3.150000E+01 3.417227E+01 2.580802E+01 4.163152E+01
3.300000E+01 3.273425E+01 2.655557E+01 4.017208E+01
3.600000E+01 3.094807E+01 3.074967E+01 3.782887E+01
4.200000E+01 2.807909E+01 2.876582E+01 3.197100E+01
4.800000E+01 2.362850E+01 2.459759E+01 2.659377E+01
5.500000E+01 1.972268E+01 2.033574E+01 2.145069E+01
6.000000E+01 1.727387E+01 1.732752E+01 1.839313E+01
6.500000E+01 1.504623E+01 1.469586E+01 1.577150E+01
8.000000E+01 1.030826E+01 8.510377E+00 9.959229E+00
1.000000E+02 6.282092E+00 4.428874E+00 5.700537E+00
2.500000E+02 4.798664E-01 1.767889E-01 3.379194E-01
4.500000E+02 6.992553E-02 2.004766E-02 4.690039E-02
5.000000E+02 4.865600E-02 1.348868E-02 3.225647E-02
9.000000E+02 6.725252E-03 1.597896E-03 4.179386E-03

The first column lists wavelengths in μm and the second, third, and fourth columns list flux in Jy. From the second column on, species correspond to zubko96_ac_acar.optc, zubko96_ac_ach2.optc, and zubko96_ac_be.optc.

C. An Example Run of mapgrid.exe

Finally, “mapgrid.exe” needs to be run to create rectified maps of surface brightness and optical depth. You need to give the input file name (the ‘*_spec.dat’ file created by the previous mapspec.exe run), a wavelength of your choice, the map size, the pixel scale, and the interpolating factor.

```
prompt> mapgrid.exe
```

```
*** MAPGRID ***
```

```
Enter the input map file name >> 07134_map.dat
```

```
READING INPUT FILE...
```

```
1: 1.000000E-01 microns
2: 2.000000E-01 microns
3: 3.650000E-01 microns
4: 4.500000E-01 microns
5: 5.560000E-01 microns
6: 6.400000E-01 microns
7: 7.900000E-01 microns
8: 1.250000E+00 microns
9: 1.650000E+00 microns
10: 2.200000E+00 microns
11: 3.450000E+00 microns
12: 3.750000E+00 microns
13: 4.800000E+00 microns
14: 7.000000E+00 microns
15: 8.000000E+00 microns
16: 8.500000E+00 microns
17: 9.700000E+00 microns
18: 1.030000E+01 microns
19: 1.100000E+01 microns
20: 1.170000E+01 microns
21: 1.200000E+01 microns
22: 1.250000E+01 microns
23: 1.289000E+01 microns
24: 1.500000E+01 microns
25: 1.740000E+01 microns
26: 1.772000E+01 microns
27: 1.800000E+01 microns
28: 1.950000E+01 microns
29: 2.080000E+01 microns
30: 2.100000E+01 microns
31: 2.150000E+01 microns
32: 2.300000E+01 microns
33: 2.450000E+01 microns
34: 2.500000E+01 microns
35: 2.700000E+01 microns
36: 3.000000E+01 microns
37: 3.150000E+01 microns
38: 3.300000E+01 microns
39: 3.600000E+01 microns
40: 4.200000E+01 microns
41: 4.800000E+01 microns
42: 5.500000E+01 microns
43: 6.000000E+01 microns
44: 6.500000E+01 microns
45: 8.000000E+01 microns
46: 1.000000E+02 microns
47: 2.500000E+02 microns
48: 4.500000E+02 microns
49: 5.000000E+02 microns
50: 9.000000E+02 microns
```

```
Select multiple wavelengths?
```

```
(Y / n) >> n
```

```
Select a wavelength by the number
(number) >> 27
```

```
Enter output image SIZE in even integer pixels
(xsize ysize) >> 100 100
Output images will have 100 x 100 pixels.
```

```
Enter output image PIXEL SCALE in arcsec/pixel
(xpixscale ypixscale) >> 0.0975 0.0975
Output images will cover 9.7500E+00 x 9.7500E+00 arcsec^2.
```

```
Enter SIZE of the search box relative to final image
(factor = 0.3 -> box is 30% of the image, max = 1) [0.3000] >> 0.5
Searchbox of [ 50, 50] will be used.
```

```
Interpolation summary
```

```
Number of pixels: 10000
Number of zero value pixels before: 9158
Number of zero value pixels after: 0
```

```
Output flux map is 07134_mapf_9.700.fits
Output tau map is 07134_mapt_9.700.fits
DONE!
```

In this example, maps at $18.0 \mu\text{m}$ are made for a grid (100 pixels by 100 pixels) having $0''.0975 \text{ pixel}^{-1}$. When interpolating the pixel values, a grid of size 50 pixels by 50 pixels is used to search for non-empty pixels. The outputs are ‘_mapf.fits’ and ‘_mapt.fits’ which list respectively surface brightness and optical depth maps.

REFERENCES

- Collison, A. J., & Fix, J. D. 1991, *ApJ*, 368, 545
- Harrington, J. P., Monk, D. J., & Clegg, R. E. S. 1988, *MNRAS*, 231, 577
- Iben, I., & Renzini, A. 1983, *ARA&A*, 21, 271
- Kim, S.-H., Martin, P. G., & Hendry, P. D. 1994, *ApJ*, 422, 164
- Mathis, J. S., Rumpl, W., & Nordsieck, K. H. 1977, *ApJ*, 217, 425
- Meixner, M., Skinner, C. J., Graham, J. R., Keto, E., Jernigan, J. G., & Arens, J. F. 1997, *ApJ*, 482, 897
- Meixner, M., Ueta, T., Bobrowsky, M., & Speck, A. K. 2002, *ApJ*, 571, 936
- Meixner, M., Zalucha, A., Ueta, T., Fong, D., & Justtanont, K. 2004, *ApJ*, submitted
- O’Hara, T. B., Meixner, M., Speck, A. K., Ueta, T., & Bobrowsky, M. 2003, *ApJ*, submitted
- Skinner, C. J., Meixner, M., Barlow, M. J., Collison, A. J., Justtanont, K., Blanco, P., Pina, R., Ball, J. R., Keto, E., Arens, J. F., & Jernigan, J. G. 1997, *A&A*, 328, 290
- Skinner, C. J., Meixner, M., & Bobrowsky, M. 1998, *MNRAS*, 300, L29
- Ueta, T., Meixner, M., & Bobrowsky, M. 2000, *ApJ*, 561, 861
- Ueta, T., Meixner, M., Dayal, A., Deutsch, L. K., Fazio, G. G., Hora, J. L., & Hoffmann, W. F. 2001, *ApJ*, 548, 1020
- Ueta, T., Meixner, M., Hinz, P. M., Hoffmann, W. F., Brandner, W., Dayal, A., Deutsch, L. K., Fazio, G. G. & Hora, J. L. 2001, *ApJ*, 557, 831
- Ueta, T., & Meixner, M. 2003, *ApJ*, in press