GALFIDL: collection of IDL tools to aid in making GALFIT decompositions

Heikki Salo 10.01.10

1 What is GALFIDL

This short manual describes GALFIDL, a set of IDL-routines designed to help making 2D multi-component structural decompositions for S^4G galaxies using Chien Peng's GALFIT (version 3.0).

Normally, GALFIT is run from the operating system command line (see the extensive GALFIT documentation at galfit homepage; it is assumed that the user is familiar with running GALFIT decompositions)

`galfit` *`inputfile`*

which performs the decomposition iteration for the data, fitting models, and initial parameter values specified in *inputfile*. Final decomposition parameters are written to an output file *galfit.NN*, where *NN* is a running number. Optionally, GALFIT makes a fits data cube containing the observed image, total model, and the residuals. Another GALFIT option is to write a fits data cube containing the different model components.

The GALFIDL routines can be used to different extent, depending on how familiar the user is with IDL:

1) The simplest way is to use the procedure `galfit_display_new.pro` for visualizing the GALFIT decomposition results in any GALFIT output file (works equally well for visualizing the initial parameters). The user of GALFIDL can simply use this routine to 'see inside' the final decomposition models made by calling GALFIT directly. The options include showing the magnitude vs sky-distance plots and/or major/minor axis profiles (marking each component separately), model residuals etc.

2) There is also a wrapper routine for calling GALFIT from inside IDL (`galfit_iterate.pro`). The advantage is that the output files will then be automatically renamed in a systematic fashion described below. The iteration and display routines are also combined in `galfit_run_new.pro`, which is more handy to use than `galfit_iterate.pro` itself.

3) However, perhaps the most useful feature of GALFIDL are the additional tools for semi-automatic making of input parameter files for GALFIT (e.g. `datalist_to_ingal.pro`). This utilizes the output from S^4G pipelines 1 and 2.

Most of the GALFIDL procedures are subroutine-type and can be called interactively from IDL. The same procedures can also be used as building blocks for longer tasks. More detailed description of routines is given in Sections 4-6.

2 Installing the GALFIDL routines

We assume that IDL and GALFIT are installed on the system, and that GALFIT can be called simply as *galfit* (e.g. defined as an alias in *.cshrc* or equivalent). All procedures are stored in `galfidl.tar`, which is unpacked with the linux command `tar xvf galfidl.tar`. You can either load them to your current working directory together with the data, or what is strongly recommended, to a separate directory (e.g. `GALFIDL.dir`) included to your IDL search path defined via `$IDL_PATH` environment variable. In the latter case, any main program type driver-routines must still be on the same working directory. It is also assumed that the IDL *Astro* package is installed, including routines for handling fits-files. For writing of PNG-files, the *convert* utility is used. The `galfidl_example.tar` contains the ngc1097 example files used in Sections 5-6. To use the pipeline 3 specific procedures, S4G data directory tree must be present (Section 7).

3 Naming conventions

In GALFIDL the input-parameter files for GALFIT iteration are designated with the suffix "*ingal*". Similarly, final parameter files containing iteration files will have the suffix "*outgal*". The name of *ingal*-file consists of the galaxy id, decomposition type id, and the suffix. The example below shows input parameter file for *ngc7794*, for a decomposition using bulge and disk components (*_bd*), named as *ngc7793_bd.ingal*

This file (example file created automatically by pipeline 3 procedures, see below) specifies the input for GALFIT.
Input data related files (all file names start with *galaxy id*):
Data image
Sigma-image
PSF-image
Mask-image
Image region to fit
Magnitude zero point
Plate scale
+ initial guesses for model components, as described in GALFIT manual

```
# made by datalist_to_ingal 26-Sep-2009 16:32:03.00
================================================
# IMAGE and GALFIT CONTROL PARAMETERS
A) ngc7793.newcut_nonan.fits          # Input data image (FITS file)
B) ngc7793_bd.fits                    # Output data image block
C) ngc7793.newcut_sigma.fits          # Sigma image name (made from data if blan
k or "none")
D) psf.fits                           # PSF image name
E) 1                                  # PSF fine sampling factor relative to data
F) ngc7793.newmask_nonan.fits         # Bad pixel mask (FITS image or ASCII coord list)
G) none                               # File with parameter constraints (ASCII file)
H) 1 1161 1  851                      # Image region to fit (xmin xmax ymin ymax)
I) 40    40                           # Size of the convolution box (x y)
J) 18.32                              # Magnitude photometric zeropoint
K) 0.75     0.75                      # Plate scale (dx dy)   [arcsec per pixel]
O) regular                            # Display type (regular, curses, both)
P) 0                                  # Choose: 0=optimize, 1=model, 2=imgblock, 3=subcomps
#------------------------------------------------------------------
# STRUCTURE:  BULGE
# Sersic function
# Component number: 1
 0) sersic                          #  Component type
 1)   591.752  457.875   0 0       #  Position x, y
 3)     7.312  1                   #  Integrated magnitude
 4)    66.655  1                   #  R_e (effective radius)   [pix]
 5) 2.0          1                 #  Sersic index n (de Vaucouleurs n=4)
 9) 0.9          1                 #  Axis ratio (b/a)
10) 10.0         1                 #  Position angle (PA)  [deg: Up=0, Left=90]
 Z) 0                               #  leave in [1] or subtract [0] this comp from data?
#------------------------------------------------------------------
# STRUCTURE:  DISK
# Exponential function
# Component number: 2
 0) expdisk                         #  Component type
 1)  591.752  457.875   0 0        #  Position x, y
 3)    7.312  1                    #  Integrated magnitude
 4) 100.  1                         #  R_s (disk scale-length)   [pix]
 9)     0.640000 0                  #  Axis ratio (b/a)
10)      94.7000 0                 #  Position angle (PA) [deg: Up=0, Left=90]
 Z) 0                               #  leave in [1] or subtract [0] this comp from data?
```

Compared to normal GALFIT input file, there is only one additional feature:

#STRUCTURE -lines

These optional keywords (treated as comments by GALFIT but read by GALFIDL procedures) identify the type of the model component for `galfit_display_new.pro`, used for labeling the display plots.

Note that GALFIT does not name its decomposition output parameter files with the *outgal*-suffix, but rather gives it a running name *galfit.NN*. Nor are the #STRUCTURE -lines written to output files. This only takes place when GALFIDL routines are used.

4 List of GALFIDL-routines

• BASIC ROUTINES for display/making galfit iteration

```
galfit_display_new.pro  -  display ingal/outgal files  (subroutine)
galfit_iterate.pro      -  wrapper for making galfit iteration (sub)
galfit_run_new.pro      -  iterate + display initial/final models sub)
galfit_run_driver.pro   -  example of calling galfit_run for several galaxies (main)
```

• Collecting/displaying pipeline 1&2 data to datalist-file

```
s4g_p2_check.pro                  - plot P2 isophotes on top of P1 image
s4g_pae_plot.pro                  - reads P2 tables,
s4g_p2_check_driver_f.pro         - driver for making the isophote check
check_mask.pro                    - display masks
s4g_mask_check_driver_f.pro       - driver for calling this (galaxylist->datalist)
```

• Making Sigma images etc.

```
pipeline_2to3_interface_f_new.pro  - manipulate datalist files
make_sigma_image_new.pro           - make sigma-images
nan_to_median.pro                  - replace NaNs with median
```

• Making ingal-templates based on datalist-file

```
datalist_to_ingal.pro              - reads datalist-file --> ingal-templates
datalist_to_ingal_driver_f.pro     - driver for this
pipe3.pro                          - complete P3: galaxylist -> ingals
ingal_edit.pro                     - make a new ingal-file from an old one
```

• auxiliary routines called by the above

```
find_galfitnn.pro                - find latest galfit.NN file created by GALFIT
copy_galfitnn_outgal.pro         - rename it to proper outgal-name
```

• auxiliary general routines needed

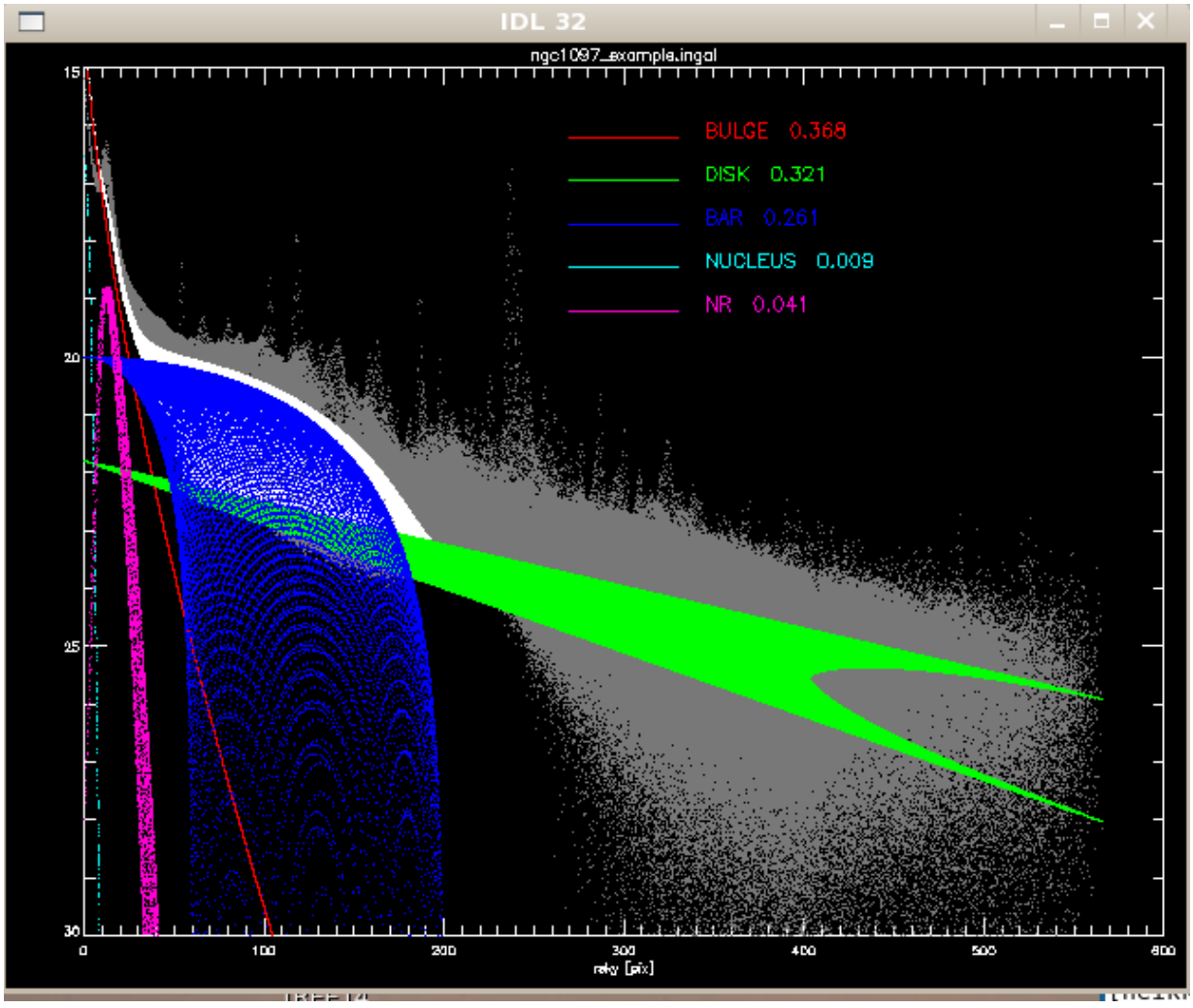
```
nwin_hs.pro                      - open  IDL window
psopen.pro                       - open output to PS file
psclose.pro                      - close output to PS file
psdirect.pro                     - handy tool for directing output between PS, EPS, X, PNG
label_data.pro                   - add labels to plots (modified from WINDT package)
remove_tag.pro                   - remove fits-extension from file name
shrink.pro                       - shrink the size of image
thick.pro                        - control line thickness
tabulate_f.pro                   - tabulation of values
tvplot2.pro                      - scalable plots of bitmaps (from IDL image_cont procedure)
wide.pro                         - delete IDL windows
image_bilin.pro                  - bilinear interpolation from image
job_edit2.pro                    - manipulation of files
sbox.pro                         - prehistoric relic
plot_stamp.pro                   - labels plot with time and user
hostname.pro                     - returns hostname
mywhoami.pro                     - returns username
ellipse_f.pro                    - plot ellipse
```

5 Displaying GALFIT parameter files

GALFIT input and output parameter files can both be displayed with the `galfit_display_new.pro`. This is a subroutine procedure, so it is called by its name, with parameters and keywords separated by commas. For example, to visualize the model in input parameter file *ngc1097_ example.ingal* (remember quotation marks on file names), enter the command

```
galfit_display_new,'ngc1097_example.ingal',/profile
```

The keyword `/profile` (same as writing `profile=1`) plots to the screen the observed and modeled magnitude of each pixel, against its sky distance from the galaxy center. Different components are separated by different colors. The numbers after labels indicate the components relative contribution to total model flux (see the example screen shot).



For example, such a plot can be used to get a rapid view of the model implied by ingal-file parameters, then one can edit the ingal-file and redisplay, until the initial guesses seem reasonable enough to start the GALFIT-iteration.

Besides such profile plots, `galfit_display_new` can also display obs/model/residual comparisons, major/minor axis profiles, profiles of each component along some specified PA.

Examples of various types of plots are given in the next pages. Note: if you enter the procedure name without any parameters → "brief" instructions are printed (same works with all GALFIDL-routines):

```
IDL> galfit_display_new
================================================================================
 galfit_display_new,INFILE
 visualize models corresponding to Chien Peng's GALFIT parameter file INFILE
================================================================================
 method: "galfit -o3" -> makes "subcomps.fits" containing model components
          store data to IDL-savefile "INFILE.subcomps.save"
          use the data to make the desired plots
--------------------------------------------------------------------------------
 PLOT KEYWORDS:
--------------
  PROFILE=  1     --> plot mag vs rsky profiles (plot every pixel)
         = -1     --> improved plot (resample images before plotting)
                          SAMPLE=sampling factor (bigger-less points)
         = -2     --> improved plot + header information on screen
                          SAMPLE=sampling factor (bigger-less points)
                          TITLE=string -> print this title to the plot
-------------
  /RESIDUAL       --> display data-cube: obs, model, residual
-------------
  /MAJORMINOR     --> plot x (PA=-90) and y (PA=0) total vs obs profiles
  = phi               plot major and minor axis profiles with major_PA=phi
  = 999               plot major and minor axis profiles with major_PA=disk_PA
-------------
  /PA             --> plot major axis profiles for each component
  = phi           --> plot major axis profiles with PA=phi
  = 999           --> plot major and minor axis profiles with major_PA=disk_PA
--------------------------------------------------------------------
 Keywords affecting profile plots:
  /FLUX makes  flux vs r_sky profiles (instead of mag vs r_sky)
  /XLOG -> use log r_sky rather than linear
   XR=[xrmin,xrmax] and YR=[yrmin,yrmax] -> define explicit x,y range
  /ARCSEC -> use arcsec rather than pix scale in plots
-----------------------------------------------------------
 Additional keywords:
  /RESTORE                  --> use previously saved INFILE.subcomps.save
                                FASTER THAN MAKING A NEW FITS-FILE
   STRUCTURE=[st1,st2,...]  --> override structure-component labels from INFILE
  /STOREFITS                --> store subcomp.fits to INFILE_subcomps.fits
  /SILENT                   --> less output
  /RUN                      --> call galfit_run_new,/new to make galfit decomposition
   CUT = npixels            --> cut image to +/- npixels around the center
                                NOTE: the INGAL-file is modified
  /PS    --> make ps and png-files instead of displaying on screen
             works with keywords:   profile=-2  --> INFILE_profile.png
                                    residual    --> INFILE_residual.png
                                    majorminor  --> INFILE_majorminor.png
                                    pa  --> INFILE_pa.png
  ADDPS=string  --> add this string to produced PS and PNG-filenames
                    (e.g. INFILE_ADDPS_profile.png)
-----------------------------------------------------------
 VARIABLES SAVED to "INFILE.subcomps.save"
      model,obs,comps,rad,ncomps,string_comps,frac_comps,zeropoint,$
      x1,x2,y1,y2,xc,yc,nx,ny,$
      data_image,out_image,sigma_image,psf_image,mask_image,xc_idl,yc_idl,$
      subcomp_header,obs_mask
 (see galfidl_manual.pdf)
-----------------------------------------------------------
 EXAMPLE:
   galfit_display_new,'ngc1097_example.ingal',profile=-2,/residual,/restore
-----------------------------------------------------------
 heikki.salo@oulu.fi 220909/070110
```

EXAMPLES: Below are examples of the use of `galfit_display_new.pro`. They assume that you have copied from the directory **GALFIDL_EXAMPLE.dir** (created by `tar xvf galfidl_example.tar`) the following files to your current working directory:

```
  ngc1097_example.ingal                - example input parameter file for GALFIT
  ngc1097_v7.phot.1_nonan.fits         - ngc1097 3.6 micron data image, NaNs removed, TEXP=1
  mask_ngc1097_nonan.fits              - corresponding mask image
  ngc1097_v7.phot.1_sigma.fits         - corresponding sigma image
  psf.fits                             - toy psf image (Gaussian with FWHM of 2.8 pixels)

These have been made from the pipeline 1 files

  ngc1097_v7.phot.1.fits                - data image from pipeline 1
  mask_ngc1097.fits                     - mask image from pipeline 1
  ngc1097_v7.phot.1_wt.fits             - weight image from pipeline 1

Using the driver-procedure ngc1097_preparations.pro, which calls make_sigma_image_new.pro
```

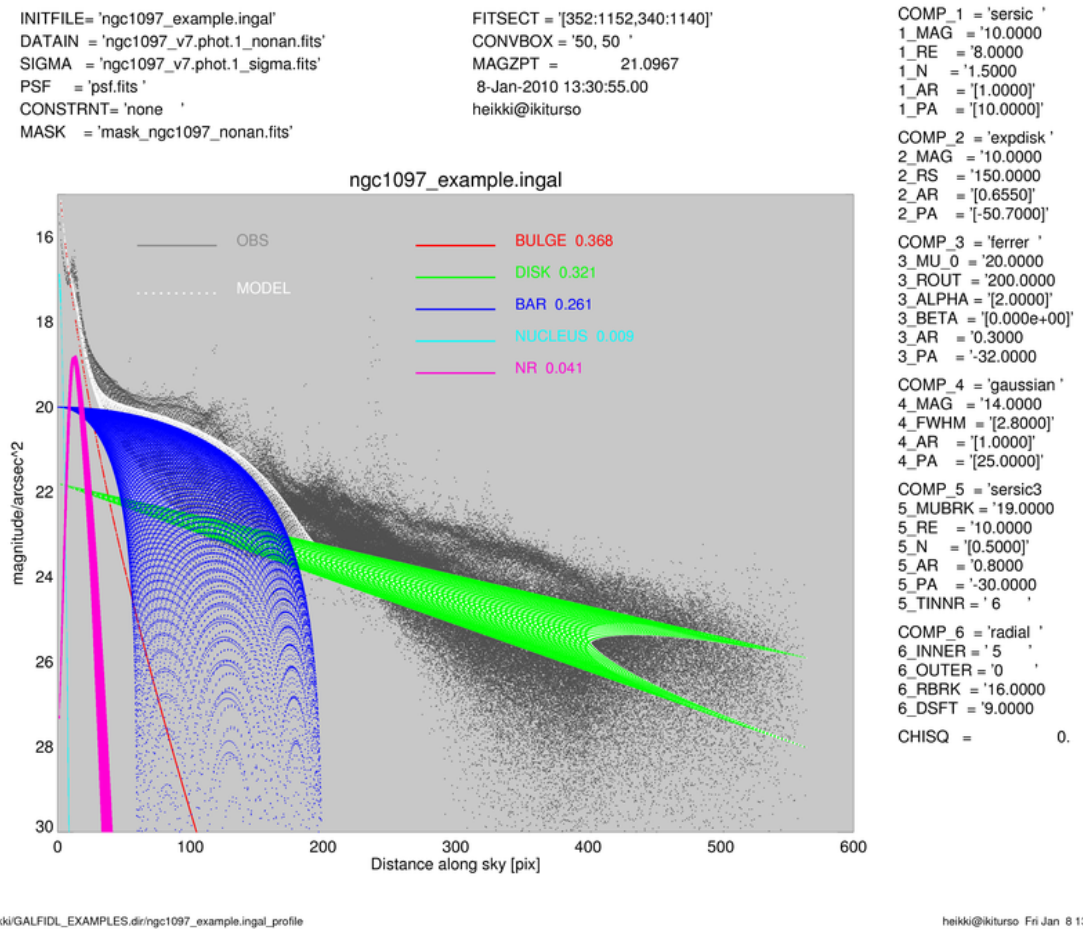
The file `ngc1097_example.ingal` is listed below: it contains a 5-component model (BULGE, DISK; BAR, NUCLEUS, NR)

```
#Example-file for using GALFIT with GALFIDL tools
#see galfidl070110.pdf  heikki.salo@oulu.fi

================================================================================
# IMAGE and GALFIT CONTROL PARAMETERS
A) ngc1097_v7.phot.1_nonan.fits      # Input data image (FITS file)
B) ngc1097_example.fits              # Output data image block
C) ngc1097_v7.phot.1_sigma.fits      # Sigma image name (made from data if blank or "none")
D) psf.fits                          # Input PSF image and (optional) diffusion kernel
E) 1                                 # PSF fine sampling factor relative to data
F) mask_ngc1097_nonan.fits           # Bad pixel mask (FITS image or ASCII coord list)
G) none                   # File with parameter constraints (ASCII file)
H) 352   1152   340  1140 # Image region to fit (xmin xmax ymin ymax)
I) 50     50              # Size of the convolution box (x y)
J) 21.0967                # Magnitude photometric zeropoint
K) 0.750  0.750           # Plate scale (dx dy)   [arcsec per pixel]
O) regular                # Display type (regular, curses, both)
P) 0                      # Choose: 0=optimize, 1=model, 2=imgblock, 3=subcomps

# INITIAL FITTING PARAMETERS
#
#   For component type, the allowed functions are:
#       sersic, expdisk, edgedisk, devauc, king, nuker, psf,
#       gaussian, moffat, ferrer, and sky.
#
#   Hidden parameters will only appear when they're specified:
#       Bn (n=integer, Bending Modes).
#       C0 (diskyness/boxyness),
#       Fn (n=integer, Azimuthal Fourier Modes).
#       R0-R10 (coordinate rotation, for creating spiral structures).
#       To, Ti, T0-T10 (truncation function).
#
# ------------------------------------------------------------------------------
#   par)    par value(s)    fit toggle(s)    # parameter description
# ------------------------------------------------------------------------------


# Component number: 1
# STRUCTURE: BULGE
 0) sersic                #  Component type
 1) 751.978   739.894 0 0  #  Position x, y
 3) 10.         1          #  Integrated magnitude
 4) 8.          1          #  R_e (effective radius)   [pix]
 5) 1.5         1          #  Sersic index n (de Vaucouleurs n=4)
 6) 0.0000      0          #     -----
 7) 0.0000      0          #     -----
 8) 0.0000      0          #     -----
 9) 1.0000      0          #  Axis ratio (b/a)
10) 10.0000     0          #  Position angle (PA) [deg: Up=0, Left=90]
 Z) 0                      #  Skip this model in output image?  (yes=1, no=0)
```

```
# Component number: 2
# STRUCTURE: DISK
 0) expdisk                #  Component type
 1) 751.978   739.894 0 0  #  Position x, y
 3) 10.         1          #  Integrated magnitude
 4) 150.        1          #  R_s (disk scale-length)   [pix]
 5) 0.0000      0          #     -----
 6) 0.0000      0          #     -----
 7) 0.0000      0          #     -----
 8) 0.0000      0          #     -----
 9) 0.6550      0          #  Axis ratio (b/a)
10) -50.70      0          #  Position angle (PA) [deg: Up=0, Left=90]
 Z) 0                      #  Skip this model in output image?  (yes=1, no=0)

# Component number: 3
# STRUCTURE: BAR
 0) ferrer                 #  Component type
 1) 751.978   739.894 0 0  #  Position x, y
 3) 20.         1          #  Central surface brghtness  [mag/arcsec^2]
 4) 200.        1          #  Outer truncation radius  [pix]
 5) 2.0000      0          #  Alpha (outer truncation sharpness)
 6) 0.0000      0          #  Beta (central slope)
 7) 0.0000      0          #     -----
 8) 0.0000      0          #     -----
 9) 0.3         1          #  Axis ratio (b/a)
10) -32.        1          #  Position angle (PA) [deg: Up=0, Left=90]
 Z) 0                      #  Skip this model in output image?  (yes=1, no=0)

# Component number: 4
# STRUCTURE: NUCLEUS
 0) gaussian               #  Component type
 1) 751.978   739.894 1 1  #  Position x, y
 3) 14          1          #  Integrated magnitude
 4) 2.8000      0          #     FWHM       [pix]
 5) 0.0000      0          #     -----
 6) 0.0000      0          #     -----
 7) 0.0000      0          #     -----
 8) 0.0000      0          #     -----
 9) 1.0000      0          #  Axis ratio (b/a)
10) 25.0000     0          #  Position angle (PA) [deg: Up=0, Left=90]
 Z) 0                      #  Skip this model in output image?  (yes=1, no=0)

# Component number: 5
# STRUCTURE: NR
 0) sersic3                #  Component type
 1) 751.978   739.894 1 1  #  Position x, y
 3) 19          1          #  Surface brghtnss @ outer R_break [mag/arcsec^2]
 4) 10          1          #  R_e (effective radius)   [pix]
 5) 0.5         0          #  Sersic index n (de Vaucouleurs n=4)
 6) 0.0000      0          #     -----
 7) 0.0000      0          #     -----
 8) 0.0000      0          #     -----
 9) 0.80        1          #  Axis ratio (b/a)
10) -30.        1          #  Position angle (PA) [deg: Up=0, Left=90]
Ti)  6                     #  Inner truncation by component number(s)
 Z) 0                      #  Skip this model in output image?  (yes=1, no=0)

# Component number: 6
T0) radial                 #  Truncation type (radial, length, height)
T4) 16.         1          #  Break radius (99% normal flux)    [pixels]
T5) 9.          1          #  Softening length (1% normal flux) [pixels]

================================================================================
```

Example 1: Display the INGAL-file, using keyword `profile = -2` for plotting the magnitude of each pixel vs the pixel sky plane distance. In contrast to using `profile = 1`, the number of plotted points is reduced by resampling the images before plotting. Model parameters, read from the header of *subcomp.fits*, are also shown.

```
galfit_display_new,'ngc1097_example.ingal',profile=-2
```



Displaying *INFILE* will first read the contents of *INFILE*, and then call GALFIT to make the *subcomp.fits* file, which is also stored into IDL savefile *infile.subcomps.save*. Plotting is done using *infile.subcomps.save*.

In case you want to re-display a previous file, give `/restore` keyword in a call: this reads the IDL savefile (if this exists) which is much faster than using GALFIT to re-create the *subcomp.fits* file. Try

```
galfit_display_new,'ngc1097_example.ingal',profile=-2,/restore
```

which will produce the same plot as before, but noticeably faster.

Note that `galfit_display_new.pro` uses the STRUCTURE lines in *INFILE* to label the components. These labels can be replaced via keyword input.

• DIRECTING OUTPUT TO FILE

The output can be directed to PS-file using the keyword `/ps`. The created PS-file contains typically a lot of points and is fairly large and difficult to display/print. Therefore, the PS-file is also converted to a PNG-file: the above plot was made by the command

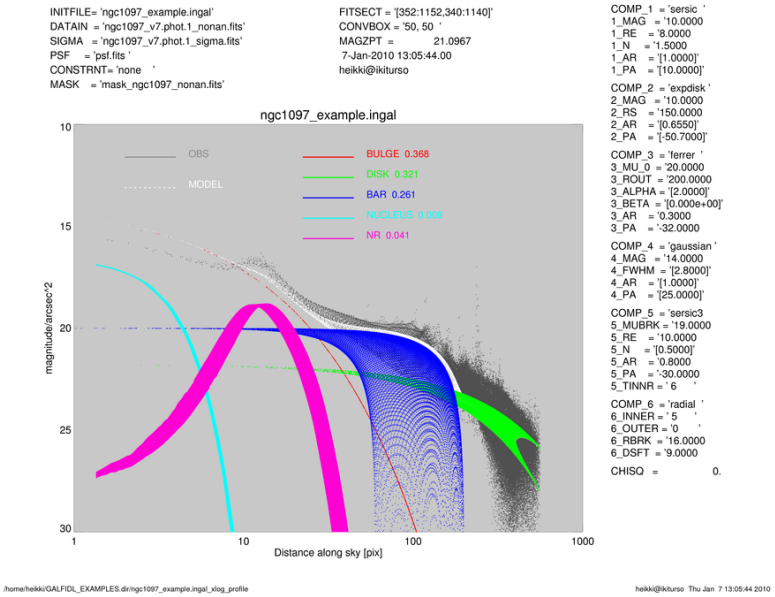
```
galfit_display_new,'ngc1097_example.ingal',profile=-2,/restore,/ps
```

which produced the shown PNG-file *ngc1097_example.ingal_profile.png*

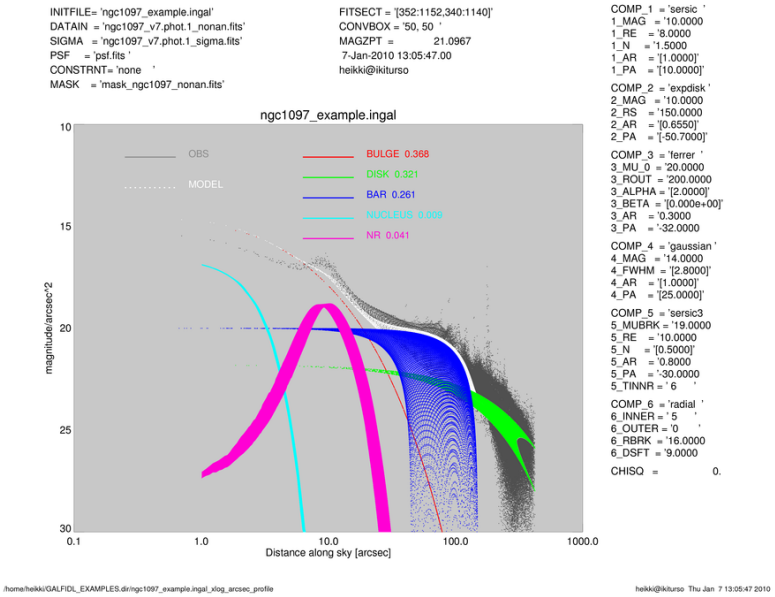
Example 1 continued

One can also use logarithmic radial scale and use arcsec-scale instead of pixels:

```
galfit_display_new,'ngc1097_example.ingal',profile=-2,/restore,/xlog,addps='xlog'
```



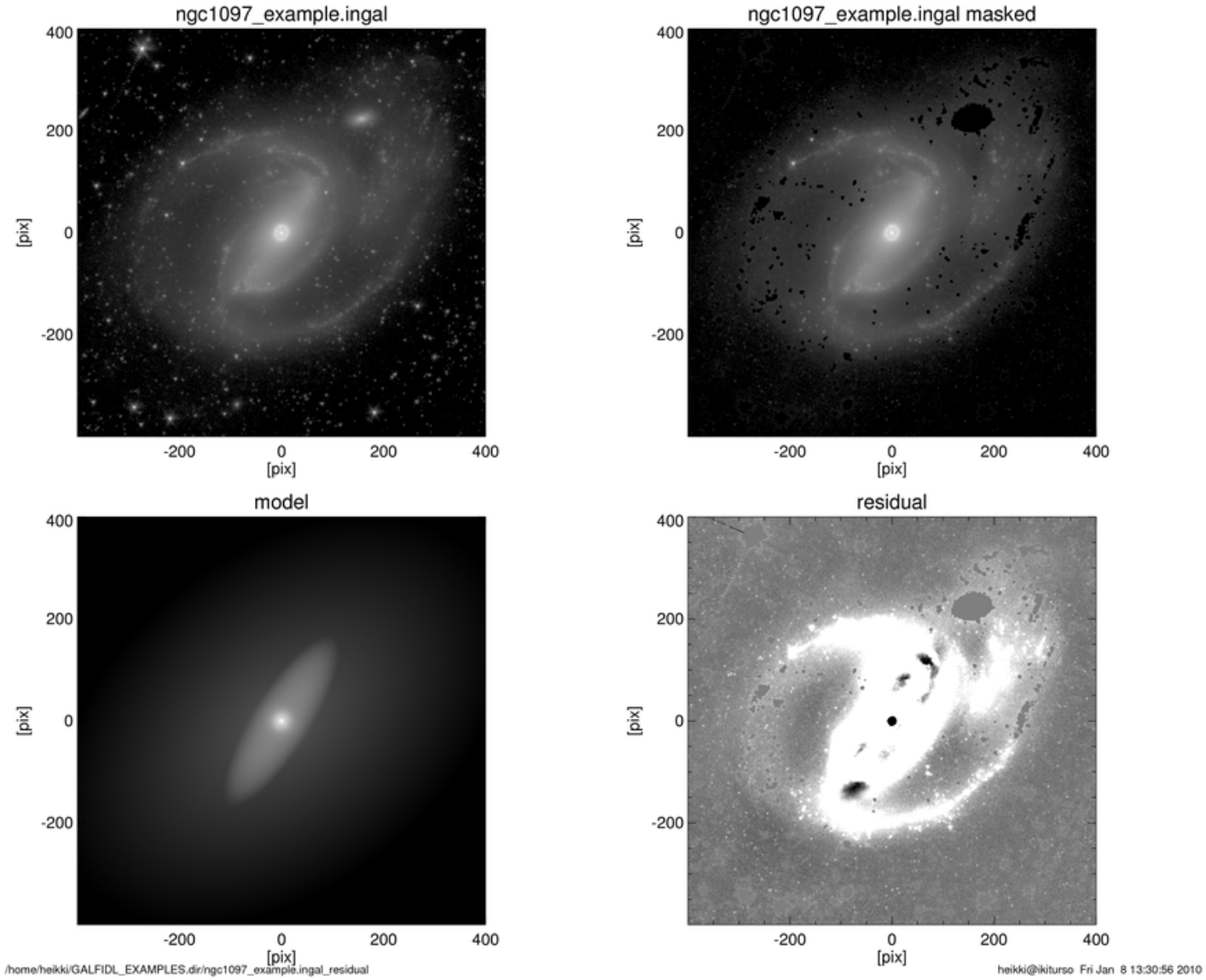

```
galfit_display_new,'ngc1097_example.ingal',profile=-2,/restore,/xlog,/arcsec,addps='xlog_arcsec'
```



The keyword `addps=string` was used for making different names for the output PNG-files.

Example 2: Display the INGAL-file, using keyword `residual`

```
galfit_display_new,'ngc1097_example.ingal',/restore,/residual
```



NOTE: the different plotting keywords can be used in combination: to make both the above plots

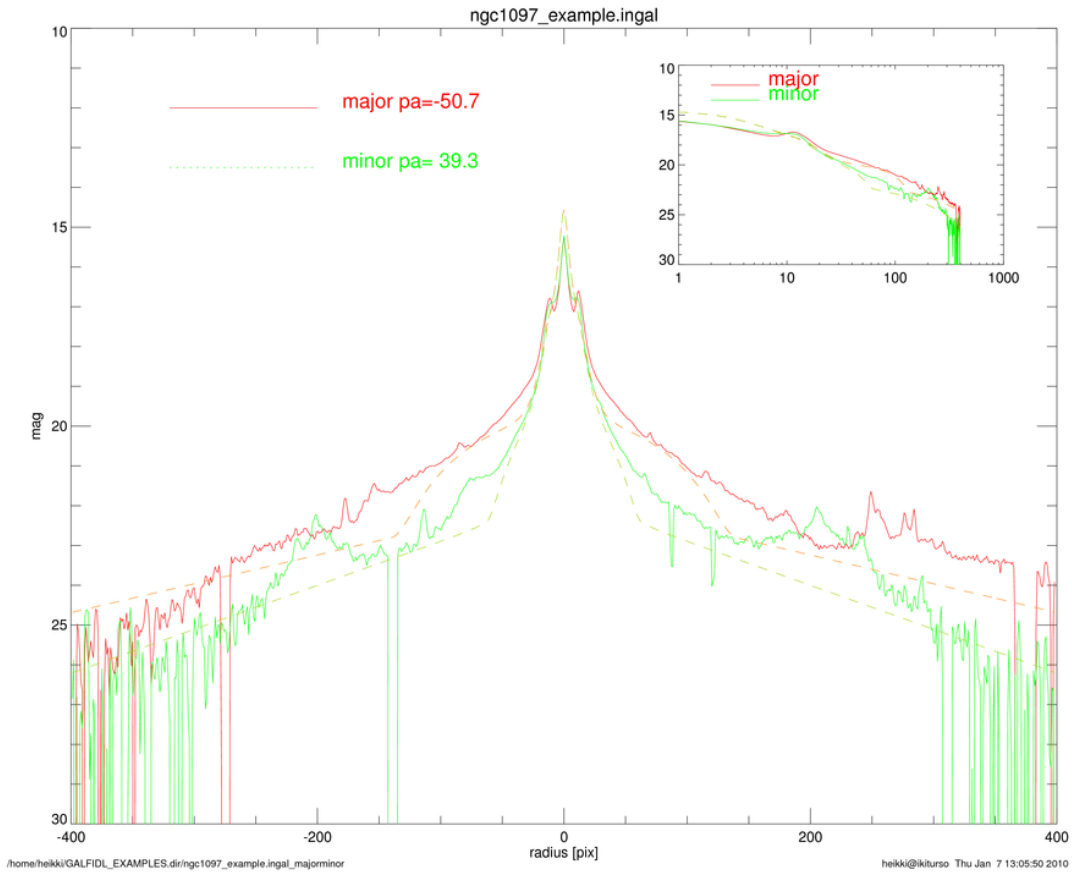
```
galfit_display_new,'ngc1097_example.ingal',/restore,/residual,profile = -2
```

Keyword `/ps` → makes the PNG-file *ngc1097_ example.ingal_ residual.png*

Example 3: Display the INGAL-file, using keyword `majorminor`

This displays the major and minor axis profiles, using the specified major axis position angle (`majorminor=PHI`). Giving `/majorminor` will plot profiles along x and y-axis. With `majorminor=999` the disk major axis specified in the INGAL-file is assumed

```
galfit_display_new,'ngc1097_example.ingal',/restore,majorminor = 999
```

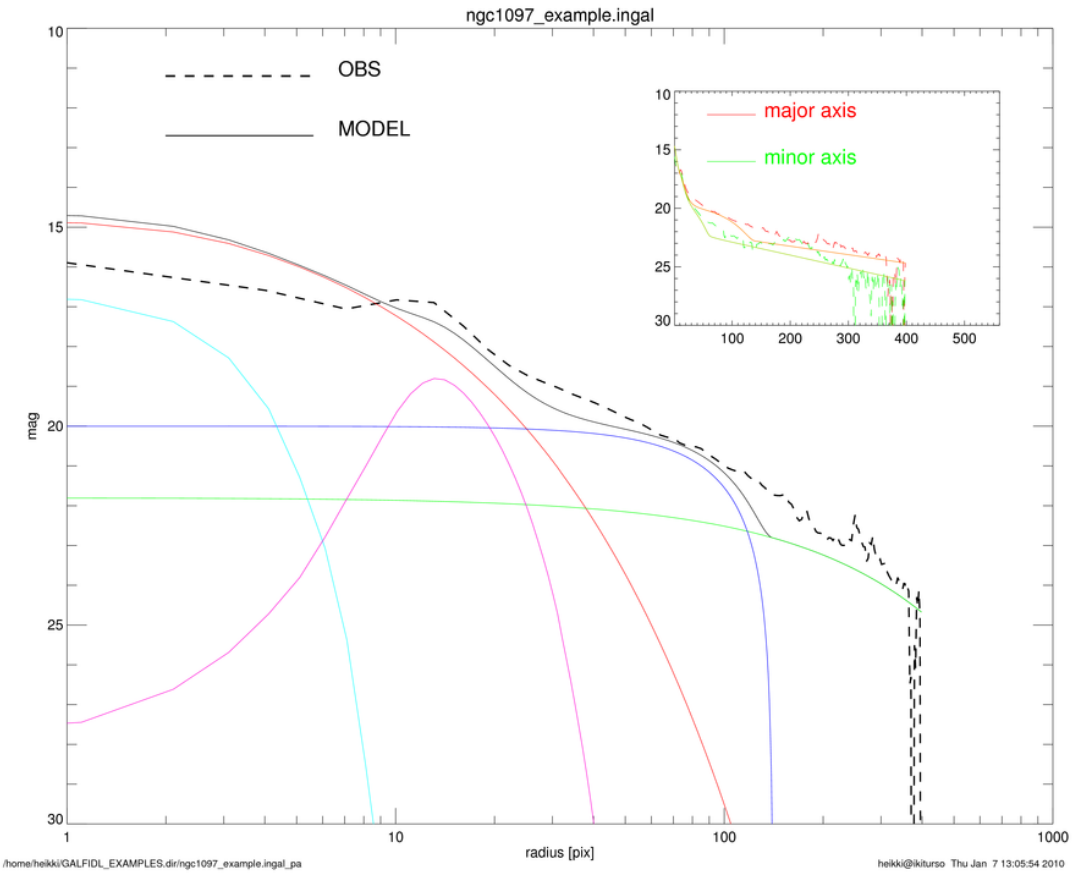


Keyword `/ps` → makes the PNG-file *ngc1097_ example.ingal_ majorminor.png*

Example 4: Display the INGAL-file, using keyword `pa`

This displays the profiles of each model component, along the position angle PHI specified by `pa=PHI`. Giving just `/PA` will plot the profile along x axis. With `PA=999` the disk major axis direction from the INGAL-file is assumed (the used phi is indicated in the plot)

```
galfit_display_new,'ngc1097_example.ingal',/restore,pa = 999
```



Keyword `/ps` → makes the PNG-file *ngc1097_ example.ingal_ pa.png*

NOTE: `galfit_display_new` is still under development: for example the keyword names and default values may be modified, based on the experience gained in making decompositions. Also, new features will be added as needed

6 MAKING GALFIT ITERATIONS with GALFIDL-tools

The previous plots have shown examples of how to visualize models corresponding to input parameter files. Once a good initial guess is found, one can make GALFIT iterations for the galaxies either directly outside IDL, or by calling GALFIT from IDL via `galfit_iterate`. For example

```
galfit_iterate','ngc1097_example.ingal'
```

This will start GALFIT iteration and display in the terminal the output from GALFIT. It also makes sure that the output parameter values are stored to proper *outgal*-file (`'ngc1097_example.outgal'` in this case). The GALFIT iteration results can then be displayed exactly in the same manner as the INGAL models

```
galfit_display_new,'ngc1097_example.outgal',prof=-2,pa=999,majorm=999,/resid
```

(Note that IDL keyword names can be abbreviated, as long as they are unique). It would also be useful to compare to the initial model, to check that the final model is indeed different from the original input.

• With the `galfit_run_new.pro` one can combine the above steps of iteration/display:

```
galfit_run_new,'ngc1097_example'
```

This will perform the iteration (*'ngc1097_ example.ingal'* → *'ngc1097_ example.outgal'*)
and display several plots:

```
1) MAG vs. SKY_DISTANCE:  2*2 frame plots comparing initial and final models,
                          both with linear and logarithmic radial scales
2) OBS/MASK/MODEL/RESIDUAL plot (/residual) for the final model
3) MAJOR/MINOR profiles of the final model (using majorminor=999 --> fitted disk orientation)
4) PA=disk major axis profiles for each final model  component (pa=999)
5) MAG vs. SKY-DISTANCE for the final model (profile=-2) in linear scale
6) MAG vs. SKY-DISTANCE for the final model (profile=-2) in log scale (xlog=1)
```

The resulting plots can be directed to PS-files and PNG-files by the keyword `ps=-2`

```
galfit_run_new','ngc1097_example.outgal',ps=-2
```

This produces PNG-files which can be printed or placed to web

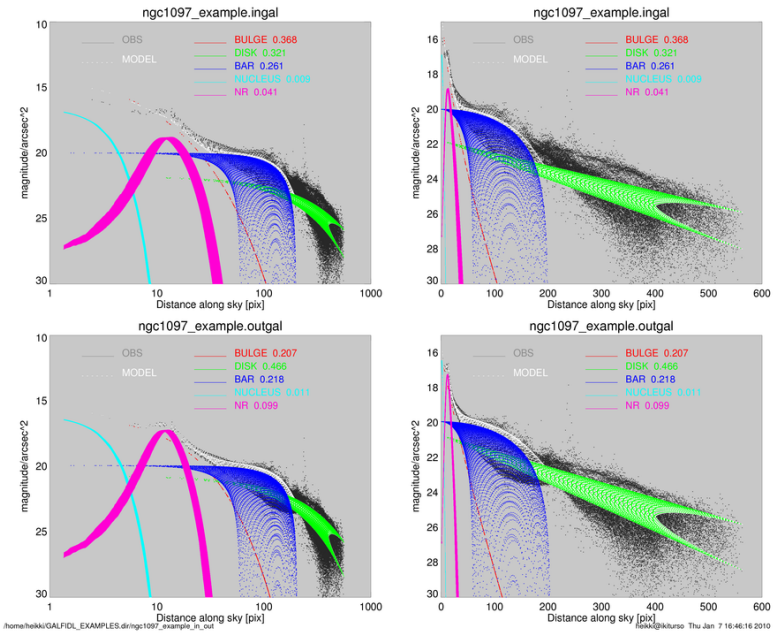
```
1) ngc1097_example_in_out.png
2) ngc1097_example_model.png
3) ngc1097_example_majorminor.png
4) ngc1097_example_pa.png
5) ngc1097_example_profile.png
6) ngc1097_example_logprofile.png
```

These files are made from PS files created by IDL, using the `convert` utility. Note: old versions of `convert` might cause problems!

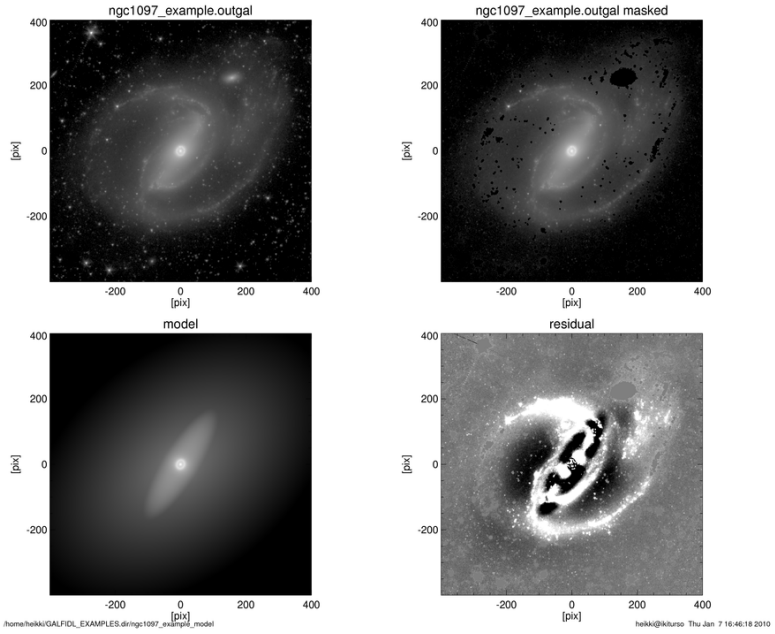
A handy feature of `galfit_run_new.pro` is that it can be called repeatedly, without re-doing the time consuming iteration. Thus for the example the above call including the making the PNG-files does not repeat the iterations. If one wants to redo the iteration (for example the *ingal*-file has been edited), then one needs to add the `/new` keyword.

EXAMPLE OUTPUT FROM `galfit_run_new','ngc1097_example.outgal',ps=-2`

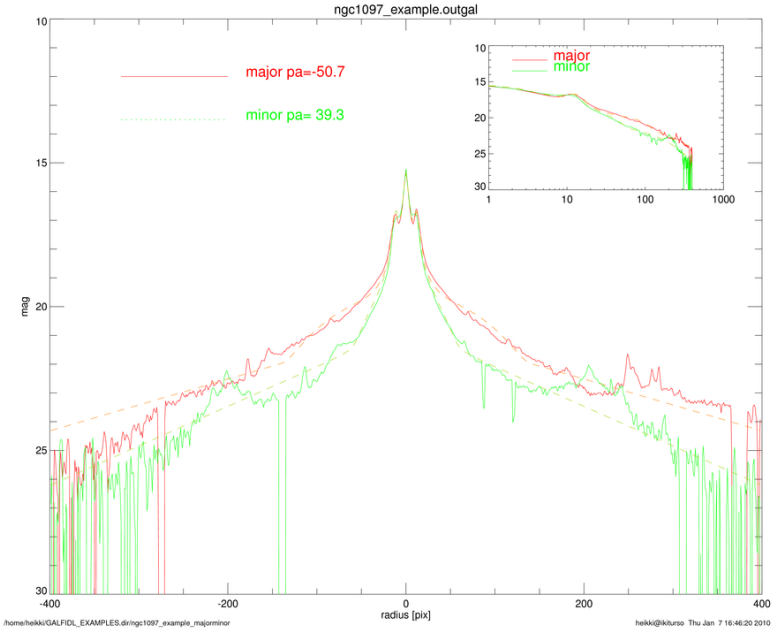
`ngc1097_example_in_out.png`



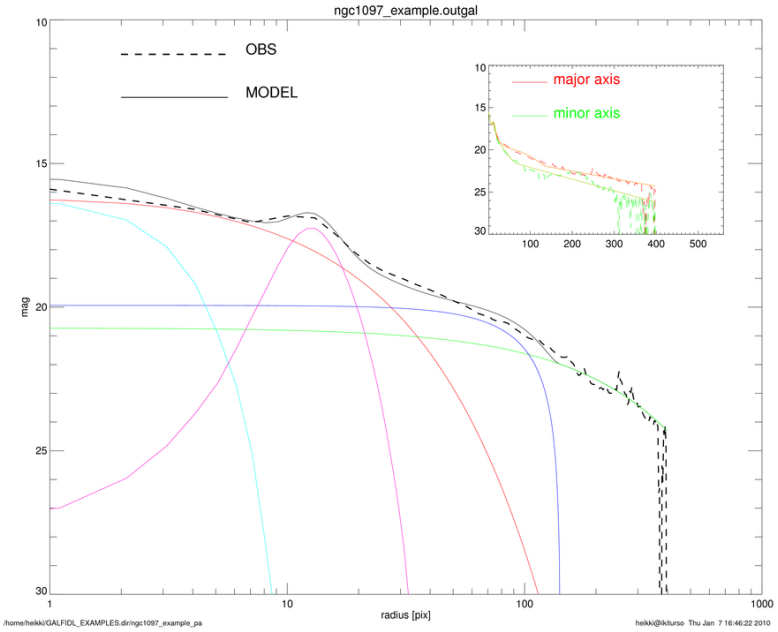
`ngc1097_example_model.png`



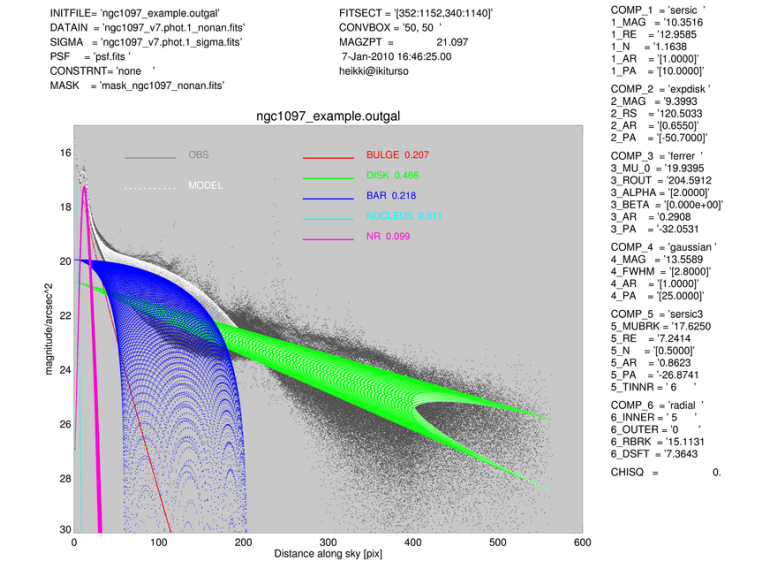
ngc1097_example_majorminor.png



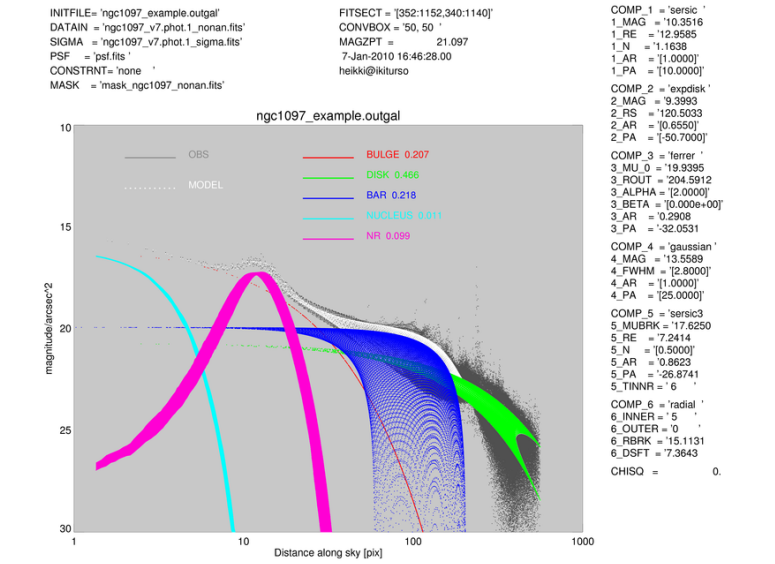
ngc1097_example_pa.png



ngc1097_example_profile.png



ngc1097_example_logprofile.png



The directory *GALFIDL_EXAMPLE.dir* contains the procedure used in making all the above plots:

```
IDL> $more ngc1097_examples.pro
;-------------------------------------------------------
;ngc1097_examples.pro
;-------------------------------------------------------
;the plots in the galfidl_manual.pdf are made with this procedure

;basic example - makes profile + residual plots
    galfit_display_new,'ngc1097_example.ingal',profile=-2,/residual,/restore,/ps

;profile with xlog
    galfit_display_new,'ngc1097_example.ingal',profile=-2,/restore,/ps,/xlog,addps='_xlog'

;profile with xlog  in arcsec scale + residual plot in arcsec scale
    galfit_display_new,'ngc1097_example.ingal',profile=-2,/restore,/resi,/ps,/xlog,/arcsec,addps='_xlog_arcsec'

;total major and minor axis profiles (major axis along the assumed disk nodal line)
    galfit_display_new,'ngc1097_example.ingal',majorminor=999,/restore,/ps

;major axis profiles for all component (along the assumed disk nodal line)
    galfit_display_new,'ngc1097_example.ingal',pa=999,/restore,/ps

;--------------
;make iteration, display results on screen
;may take time!  (22 iterations took 192 seconds in ikiturso)
;IF YOU HAVE ALREADY MADE THE ITERATION, THEN COMMENT OUT THE NEXT LINE TO SAVE TIME!

    galfit_display_new,'ngc1097_example.ingal',/run

;redisplay iteration results (note galfit_run_new does not repeat iteration unless /new)
;without ps -> screen
;ps=1  -> eps and pdf files (making pdf is slow and the files are large)
;ps=2  ->  ps and png   BETTER OPTION!  ps=-2 -> timestamp on the
;plots

    galfit_run_new,'ngc1097_example',ps=-2

;call browser to view the output png-files
    cmd='firefox *ngc1097_example_*png &'
    spwan,cmd

;results stored to
    restore,'ngc1097_example.outgal_subcomps.save',/v

end
```

• `galfit_display_new.pro` provides also a shortcut for making iterations: the `/run` keyword. In practice one edits the *ingal*-file, displays results, and then adds `/run` keyword, without having explicitly to call `galfit_run_new.pro`.

7 STORING RESULTS OF ITERATIONS

GALFIT stores a brief summary of all its iterations to *fit.log* , and the final fit parameters to *galfit.NN*, where NN is a running number. When GALFIDL is used for calling GALFIT, the *galfit.NN* file is automatically copied to the name *IDE.ingal* → *IDE.outgal*.

Besides the plots described in the previous sections, GALFIDL stores the GALFIT iteration results to the IDL savefile *IDE.outgal_subcomps.save*

```
IDL> restore,'ngc1097_example.outgal_subcomps.save',/v
    % RESTORE: Portable (XDR) SAVE/RESTORE file.
    % RESTORE: Save file written by heikki@ikiturso, Thu Jan  7 13:04:49 2010.
    % RESTORE: IDL version 7.0 (linux, x86).
    % RESTORE: Restored variable: MODEL.
    % RESTORE: Restored variable: OBS.
    % RESTORE: Restored variable: COMPS.
    % RESTORE: Restored variable: RAD.
    % RESTORE: Restored variable: NCOMPS.
    % RESTORE: Restored variable: STRING_COMPS.
    % RESTORE: Restored variable: FRAC_COMPS.
    % RESTORE: Restored variable: ZEROPOINT.
    % RESTORE: Restored variable: X1.
    % RESTORE: Restored variable: X2.
    % RESTORE: Restored variable: Y1.
    % RESTORE: Restored variable: Y2.
    % RESTORE: Restored variable: XC.
    % RESTORE: Restored variable: YC.
    % RESTORE: Restored variable: NX.
    % RESTORE: Restored variable: NY.
    % RESTORE: Restored variable: DATA_IMAGE.
    % RESTORE: Restored variable: OUT_IMAGE.
    % RESTORE: Restored variable: SIGMA_IMAGE.
    % RESTORE: Restored variable: PSF_IMAGE.
    % RESTORE: Restored variable: MASK_IMAGE.
    % RESTORE: Restored variable: XC_IDL.
    % RESTORE: Restored variable: YC_IDL.
    % RESTORE: Restored variable: SUBCOMP_HEADER.
    % RESTORE: Restored variable: OBS_MASK.
```

Content of variables:

```
   MODEL             nx*xy          model image
   OBS               nx*ny          obs  image
   COMPS             nx*ny*ncomps   model component images
   RAD               nx*ny          distance from center
   NCOMPS            number of components
   STRING_COMPS      component names
   FRAC_COMPS        component fractional fluxes
   ZEROPOINT         magnitude zeropoint
   X1,X2,Y1,Y2       fitted region [x1:x2,y1:y2]
   XC,YC             center [XC,YC] iraf coords
   NX,NY             image size nx*ny
   DATA_IMAGE        image fits-file
   OUT_IMAGE         output datacube fits-file
   SIGMA_IMAGE       sigma image fits-file
   PSF_IMAGE         psf image fits-file
   MASK_IMAGE        mask image fits-file
   XC_IDL,YC_IDL     center [XC,YC] idl coords
   SUBCOMP_HEADER    header containing all model parameters
   OBS_MASK          nx*ny  obs image, bad pixels masked
```

Total magnitude $= -2.5\log_{10}(\sum_i MODEL_i) + zeropoint$

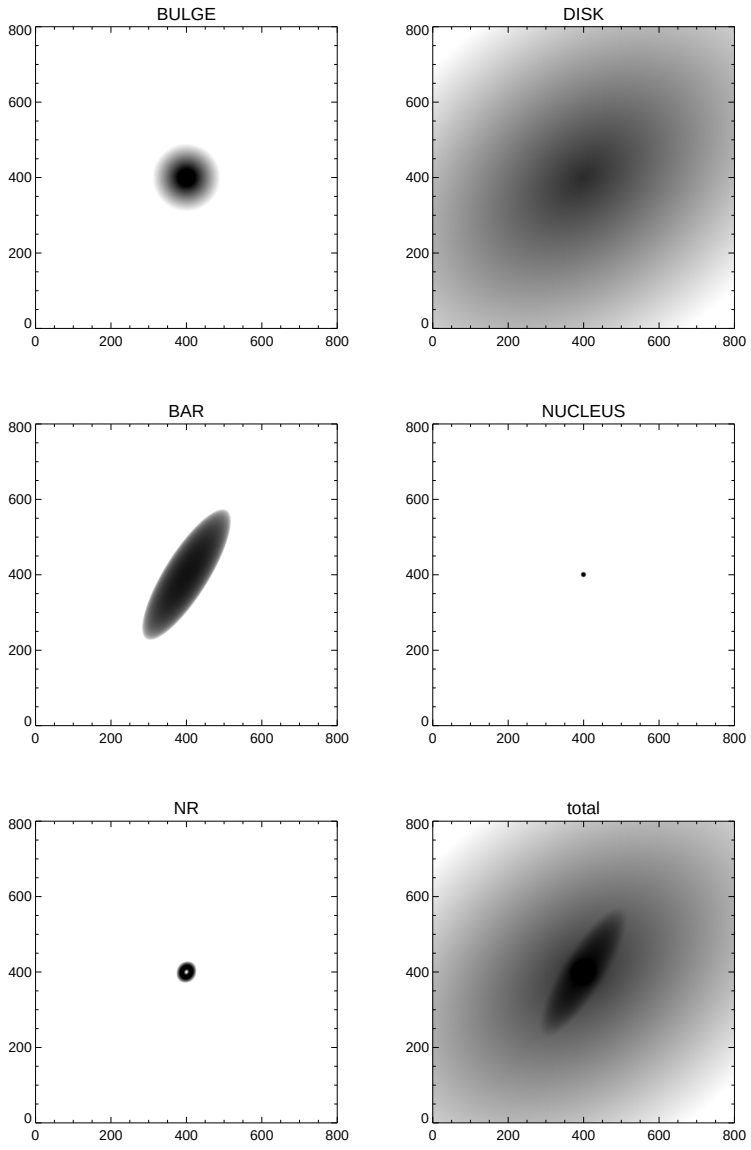
Surface brightness $-2.5\log_{10}(MODEL_i/pix^2) + zeropoint$

SUBCOMP_HEADER contains:

```
IDL> hprint,subcomp_header
XTENSION= 'IMAGE   '           / IMAGE extension
BITPIX  =                   -32 / Bits per pixel
NAXIS   =                     2 / Number of axes
NAXIS1  =                   801 / Length of data axis 1
NAXIS2  =                   801 / Length of data axis 2
OBJECT  = 'sersic3 /'          / Component type
COMMENT ========== GALFIT Input Parameters ==========
INITFILE= 'ngc1097_example.outgal' / GALFIT input file
DATAIN  = 'ngc1097_v7.phot.1_nonan.fits' / Input data image
SIGMA   = 'ngc1097_v7.phot.1_sigma.fits' / Input sigma image
PSF     = 'psf.fits '          / Convolution PSF and kernel
CONSTRNT= 'none    '           / Parameter constraint file
MASK    = 'mask_ngc1097_nonan.fits' / Input mask image
FITSECT = '[352:1152,340:1140]' / Image section fitted
CONVBOX = '50, 50  '           / Convolution box size
MAGZPT  =                21.097 / Magnitude zeropoint
COMMENT ========== GALFIT Final Parameters ==========
COMP_1  = 'sersic  '           / Component type
1_XC    = '[751.9780]'         / X center [pixel]
1_YC    = '[739.8940]'         / Y center [pixel]
1_MAG   = '10.3516 +/- 0.0000' / Integrated magnitude [mag]
1_RE    = '12.9585 +/- 0.0000' / Effective radius Re [pixels]
1_N     = '1.1638 +/- 0.0000'  / Sersic index
1_AR    = '[1.0000]'           / Axis ratio (b/a)
1_PA    = '[10.0000]'          / Position Angle (PA) [degrees: Up=0, Left=90]
COMMENT ------------------------
COMP_2  = 'expdisk '           / Component type
2_XC    = '[751.9780]'         / X center [pixel]
2_YC    = '[739.8940]'         / Y center [pixel]
2_MAG   = '9.3993 +/- 0.0000'  / Integrated magnitude
2_RS    = '120.5033 +/- 0.0000' / Scalelength [pixels]
2_AR    = '[0.6550]'           / Axis ratio (b/a)
2_PA    = '[-50.7000]'         / Position Angle (PA) [degrees: Up=0, Left=90]
COMMENT ------------------------
COMP_3  = 'ferrer  '           / Component type
3_XC    = '[751.9780]'         / X center [pixel]
3_YC    = '[739.8940]'         / Y center [pixel]
3_MU_0  = '19.9395 +/- 0.0000' / Central surface brightness
3_ROUT  = '204.5912 +/- 0.0000' / Outer radius [pixels]
3_ALPHA = '[2.0000]'           / Alpha (outer truncation sharpness)
3_BETA  = '[0.000e+00]'        / Beta (central slope)
3_AR    = '0.2908 +/- 0.0000'  / Axis ratio (b/a)
3_PA    = '-32.0531 +/- 0.0000' / Position Angle (PA) [degrees: Up=0, Left=90]
COMMENT ------------------------
COMP_4  = 'gaussian '          / Component type
4_XC    = '751.9261 +/- 0.0000' / X center [pixel]
4_YC    = '739.8774 +/- 0.0000' / Y center [pixel]
4_MAG   = '13.5589 +/- 0.0000' / Integrated magnitude
4_FWHM  = '[2.8000]'           / FWHM [pixels]
4_AR    = '[1.0000]'           / Axis ratio (b/a)
4_PA    = '[25.0000]'          / Position Angle (PA) [degrees: Up=0, Left=90]
COMMENT ------------------------
COMP_5  = 'sersic3 /'          / Component type
5_XC    = '751.9189 +/- 0.0000' / X center [pixel]
5_YC    = '740.0079 +/- 0.0000' / Y center [pixel]
5_MUBRK = '17.6250 +/- 0.0000' / Surf. brghtnss @ break radius [mag/arcsec^2]
5_RE    = '7.2414 +/- 0.0000'  / Effective radius Re [pixels]
5_N     = '[0.5000]'           / Sersic index
5_AR    = '0.8623 +/- 0.0000'  / Axis ratio (b/a)
5_PA    = '-26.8741 +/- 0.0000' / Position Angle (PA) [degrees: Up=0, Left=90]
5_TINNR = ' 6      '           / Inner truncation by component(s)
COMMENT ------------------------
COMP_6  = 'radial  '           / Component type
6_INNER = ' 5      '           / Component(s) with inner truncation
6_OUTER = '0       '           / Component(s) with outer truncation
6_RBRK  = '15.1131 +/- 0.0000' / Break radius (99% normal flux) [pixels]
6_DSFT  = '7.3643 +/- 0.0000'  / Softening length (1% normal flux) [pixels]
COMMENT ------------------------
CHISQ   =                   0. / Chi^2 of fit
NDOF    =                    0 / Degrees of freedom
NFREE   =                   20 / Number of free parameters
NFIX    =                   16 / Number of fixed parameters
CHI2NU  =                   0. / Reduced Chi^2
LOGFILE = 'galfit.02'          / Output logfile
COMMENT ============================================
COMMENT
END
```

Example: use the data stored in IDL savefile to display separately each component

```
restore,'ngc1097_example.outgal_subcomps.save'

magsi=-2.5*alog10(comps/0.75^2>1d-16)+zeropoint
magtot=-2.5*alog10(model/0.75^2>1d-16)+zeropoint
mag1=20.
mag2=28.
magsi(0,0,*)=mag1 & magsi(0,1,*)=mag2
magtot(0,0)=mag1 & magtot(0,1)=mag2

nwin_hs,xs=400,ys=600
!p.multi=[0,2,3]
!p.charsize=1
tvplot2,magsi(*,*,0)>mag1<mag2,/asp,title=string_comps(0)
tvplot2,magsi(*,*,1)>mag1<mag2,/asp,title=string_comps(1)
tvplot2,magsi(*,*,2)>mag1<mag2,/asp,title=string_comps(2)
tvplot2,magsi(*,*,3)>mag1<mag2,/asp,title=string_comps(3)
tvplot2,magsi(*,*,4)>mag1<mag2,/asp,title=string_comps(4)
tvplot2,magtot>mag1<mag2,/asp,title='total'
```



8 PIPELINE 3 PROCEDURES: making *ingal*-templates

• REQUIREMENTS:

The pipeline 3 -specific procedures assume that the Pipeline 1 and 2 data are stored to the `S4G` directory, located under the user's home-directory (or pointed by a symbolic link). The directory has the structure

```
 S4G/PRODUCTS/ARCHIVAL/GALAXY/P1
                             /P2
                             /P3

     P1 contains DATA-images, MASK-images and WEIGHT-images
        (weight images indicate how many individual frames were used for
         each pixel in the final mosaic)
     P2 contains surface photometry and isophotal ellipse fits,
        including ellipse parameters at 26.5 mag.
     P3 are directores to which store final pipeline 3 products

It is assumed that each directory S4G/PRODUCTS/ARCHIVAL/GALAXY/P1 contains the files

    GALAXY.phot.1_wt.fits
    GALAXY.1mask.fits
    GALAXY.phot.1_wt.fits

From S4G/PRODUCTS/ARCHIVAL/GALAXY/P2 the following files are used

    GALAXY.26p5.dat
    GALAXY.1fr6a_obs.dat
    GALAXY.1fr2a_obs.dat

At Dec 4th, 2009, the S4G archive contained 38 galaxies
for which both P1 and P2 data were available.
```

NOTE: The new S4G data will be stored to `S4G/PRODUCTS/WARM`. The current procedures also assume that 3.6 micron data is used. Accessing WARM data and 4.5 micron band will require small changes to some of the procedures.

• MAKING INGAL-FILES:

STEP 1: Choose the set of galaxies to be processed by pipeline 3 procedures.

This will be done by writing a LISTFILE = list of galaxy identifications ("GALAXY"in the above) to a file. Below is an example *pipe3_ example.galaxylist* (contained in directory **GALFIDL _EXAMPLE.dir**, which lists 37 galaxies for which both P1 and P2 data is available

NGC5194_95 is omitted since its file names do not obey the above naming convention; Also for ngc1097 the mask file is 'NGC1097mask.fits' which needs to be renamed to 'NGC1097.1mask.fits'

```
#------------------------------------------------------------
#pipe3_example.galaxylist
#Contains all galaxies with P1+P2 data available
#before dec 2009 IAC workshop
#Commented out all except for 5 test cases
#------------------------------------------------------------
#ESO418-008
#NGC0300
 NGC0337
 NGC0428
#NGC0628
 NGC0986
#NGC1097
#NGC1291
 NGC1433,
#NGC1512
 NGC1566
...
#NGC7479
#NGC7793
#------------------------------------------------------------
```

Lines starting by '#' are treated as comments: in the following examples 5 galaxies are used (NGC0337, NGC0428, NGC0986, NGC1433, NGC1566)

STEP 2: COLLECTING DATA TO DATALIST-TABLE

For the galaxies in LISTFILE, we store to DATALIST-table the names of the `DATA-image, MASK-image and WEIGHT-image` (without the P1 directory path). A PSF-image for each galaxy is also be needed, but for the time being we use a toy PSF *psf.fits* contained in directory **GALFIDL_EXAMPLE.dir** (This is a Gaussian with FWHM=2.8 pixels).

Simultaneously, we fetch the galaxy center location `xc, yc and the 26.5 mag orientation parameters PA, ELLIP,A26p5` derived at Pipeline 2, and store them to the same DATALIST-table.

To perform the above operations for all the galaxies in the LISTFILE (=step 2), enter the IDL command

```
s4g_p2_check_driver_f,'pipe3_example.galaxylist','pipe3_example.datalist0'
```

This writes the DATALIST file `pipe3_example.datalist0`

```
IDL> $more pipe3_example.datalist0
version=1
# ============================================================
# Interface between pipelines 2 and 3
# datalist0-file made from infile= pipe3_example.galaxylist
# heikki@ikiturso     8-Jan-2010 16:16:18.00
# ============================================================
# IDE       image                   XC(iraf)   YC(iraf)   PA     ELLIP   A26p5  AMULT  mask              weight                  psf
#
NGC0337_36  NGC0337.phot.1.fits     289.000     291.848  -60.19  0.2699  147.2  1.3  NGC0337.1mask.fits  NGC0337.phot.1_wt.fits  psf.fits
NGC0428_36  NGC0428.phot.1.fits     545.440     873.576  -64.76  0.1076  171.3  1.3  NGC0428.1mask.fits  NGC0428.phot.1_wt.fits  psf.fits
NGC0986_36  NGC0986.phot.1.fits     970.009     760.810   73.59  0.0570  211.9  1.3  NGC0986.1mask.fits  NGC0986.phot.1_wt.fits  psf.fits
NGC1433_36  NGC1433.phot.1.fits     336.808     359.475   13.93  0.1426  333.6  1.3  NGC1433.1mask.fits  NGC1433.phot.1_wt.fits  psf.fits
NGC1566_36  NGC1566.phot.1.fits     706.519     716.103  -64.18  0.2518  416.1  1.3  NGC1566.1mask.fits  NGC1566.phot.1_wt.fits  psf.fits
```

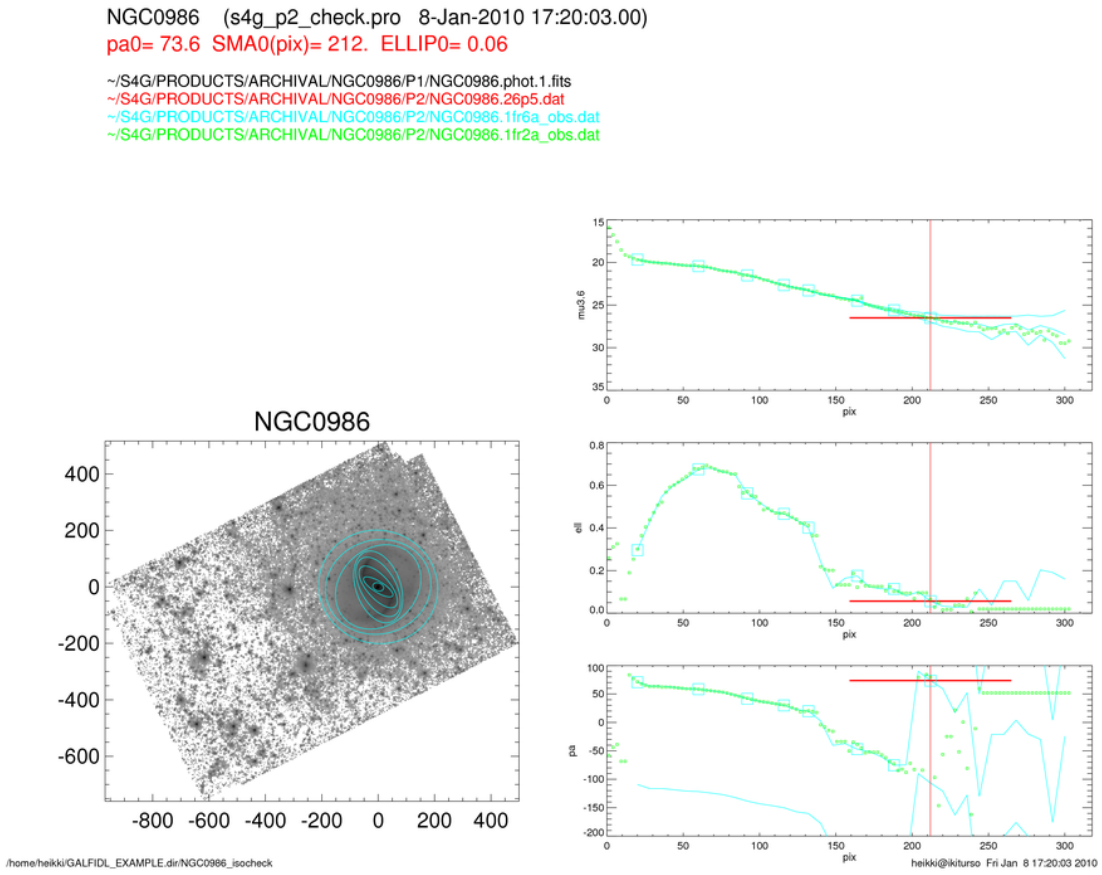
The first column is the galaxy ID, to be used in all subsequent decompositions. Center coordinates are in iraf-coordinates system starting from [1,1]: XC and YC will be used in giving the center locations of model components PA and ELLIP are orientation of 26.5 mag isophote, with major axis radius A26p5 pixels. AMULT is explained below.

Since this procedure reads the P1 and P2 data, in can used for displaying the isophotes on top of the images. Such plots are useful for example in deciding whether one could cut the fitting region in the decomposition processing. The procedure is also useful in providing an easy visual check that the IRAF ellipse fitting in P2 has worked as desired.

This check is by adding the keyword `/check`. Default is to plot on the screen: with `/ps` the output is directed to PS and PNG files (eg. *NGC0986_ isocheck.PNG*)

```
s4g_p2_check_driver_f,'pipe3_example.galaxylist','pipe3_example.datalist0',/check,/ps
```

In the plot blue and green indicate isophotes with 6arcsec and 2arcsec bins, red vertical line marks the 26.5 isophotal radius, ell and PA from *"GALAXY.26p5.dat"*. Blue boxes correspond to isophotes plotted on top of the log-stretched image.



Below are the brief instructions

```
IDL> s4g_p2_check_driver_f
-----------------------------------------------------------
 s4g_p2_check_driver_f,infile,datalist0
-----------------------------------------------------------
 FIRST PROCEDURE TO CALL IN PIPELINE3:
   read a list of galaxy identifications from infile
   fetch P1 image and P2 params --> write to datalist0-file
 CAN ALSO BE USED FOR CHECKING ISOPHOTES - keyword check
   check=1 -> also plot the isophotes on top of images
   check=2 -> just plot the isophotes on top of images
              do not write datalist0
   wait=1   -> wait between plots
   /ps      -> write  to ps and png-files instead of screen
-----------------------------------------------------------
 FORMAT OF INFILE:
   1 entry per line (eg. NGC0300), line starting with # ignored
   list of files can also be given via keyword (files=strarr)
-----------------------------------------------------------
 Currently works only for 3.6 micron ARCHIVAL DATA:
 assumes that data is at
   S4G_DIR='~/S4G/PRODUCTS/ARCHIVAL/'
   datadir=S4G_DIR+ide+'/P1/'
   image_file=datadir+ide+'.phot.1.fits'
   profdir=S4G_DIR+ide+'/P2/'
-----------------------------------------------------------
```

STEP 3: EDITING THE DATALIST-TABLE

Besides telling where the S4G data is located, the table created in STEP 2 has another purpose: it helps to provide initial guesses for GALFIT input parameters.

One reasonable option is to make bulge-disk decompositions and bulge-disk-bar decompositions, where the disk ellipticity and position angle are fixed to isophotal values of the outer disk. As a proxy of the outer disk orientation 26.5 mag isophotes are used. However, since this isophote is not necessarily optimal (for example, the galaxy may terminate before this magnitude level, the image may be too small etc.), one can edit the orientation parameters in the DATALIST-table (for example, utilizing the isophote-plots obtained in STEP 2).

Also, in this table one marks the edge-on galaxies, for which different fitting functions should be used by GALFIT. This is done by writing 'zzzzz' in place of disk ellipticity.

The isophotal radius A26p5 is utilized in choosing where to cut the images: the cutting distance relative to A26p5 is set by AMULT, which can be edited in this phase.

In the example provided, there is no need to edit the table.

STEP 4: FINAL DATALIST-TABLE: calculating sigma images

The zero '0' in the end of the previous DATALIST-file indicated that this is not yet the final data compilation we need for the decompositions.

For the decompositions we actually need the Sigma images (SIGMA), specifying the statistical uncertainty associated with each image pixel, in the same units as the image data. Additionally, the input images should not contain NaN pixel values as GALFIT does not handle them. Also, the EXPTIME header keyword in the DATA image files should be set to unity, so that GALFIT does not try to normalize the input data values.

GALFIDL can take care of all that:

```
1) Construct SIGMA images from the DATA and WEIGHT images:
2) Make cleaned versions of the DATA image, where NaN pixel values have been removed,
   and EXPTIME is set to unity.
3) make new MASK images, where the pixels which had NaN's either in DATA or in WEIGHT
   (and thus also in SIGMA) have been included to the pixels to be masked away.
4) Write a new datalist-file where file names reflect the above changes
```

To do this type

```
.run pipeline_2to3_interface_f_new,'pipe3_example'
```

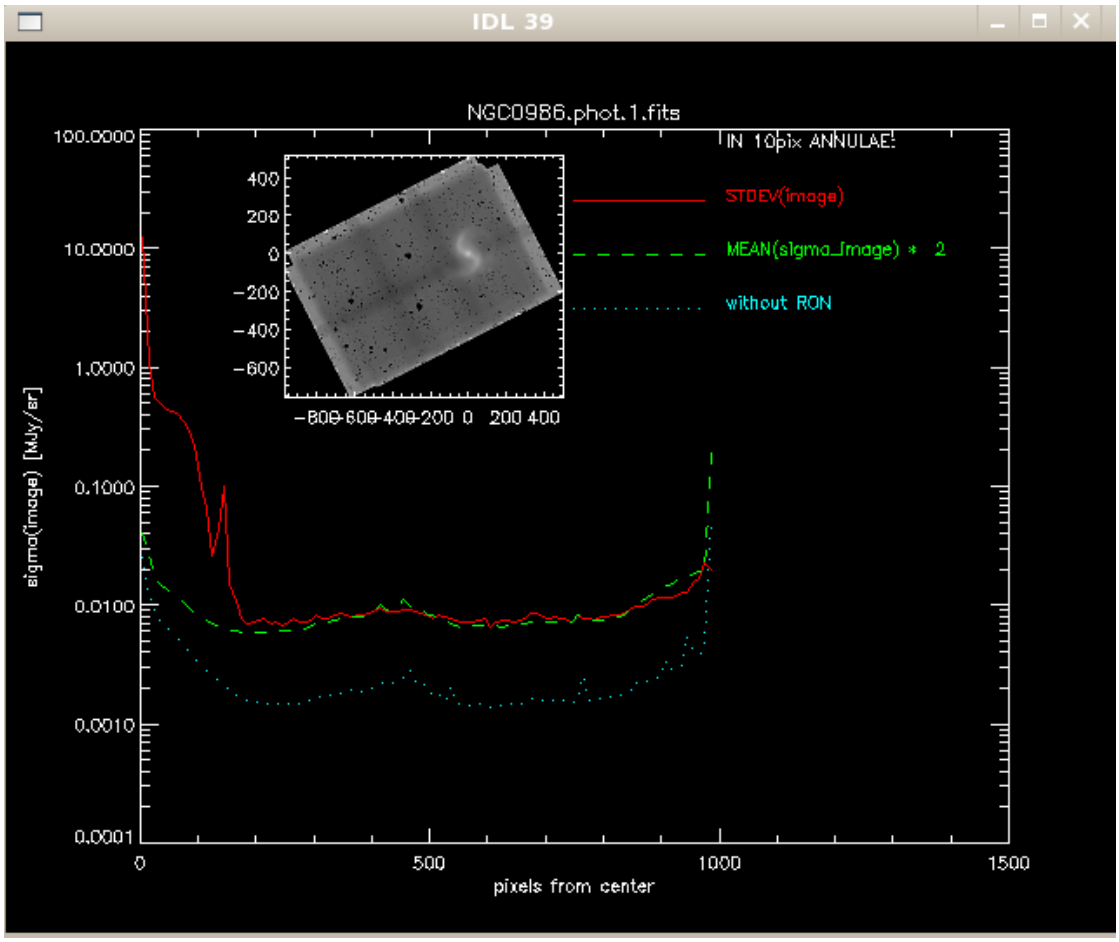
This creates a new datalist file *pipe3_ example'.datalist*, where the new DATA and MASK image names are inserted, and the WEIGHT images have been replaced by SIGMA images.

```
IDL> $more  pipe3_example.datalist
version=1
# ============================================================
# Interface between pipelines 2 and 3
# datalist-file made from infile= pipe3_example.datalist0
# heikki@ikiturso     8-Jan-2010 17:06:05.00
# ============================================================
# IDE       image                      XC(iraf) YC(iraf)   PA    ELLIP  A26p5  AMULT  mask                    igma                     psf
#
NGC0337_36  NGC0337.phot.1_nonan.fits  289.000  291.848  -60.19  0.2699  147.2  1.3 NGC0337.1mask_nonan.fits NGC0337.phot.1_sigma.fits psf.fit
NGC0428_36  NGC0428.phot.1_nonan.fits  545.440  873.576  -64.76  0.1076  171.3  1.3 NGC0428.1mask_nonan.fits NGC0428.phot.1_sigma.fits psf.fit
NGC0986_36  NGC0986.phot.1_nonan.fits  970.009  760.810   73.59  0.0570  211.9  1.3 NGC0986.1mask_nonan.fits NGC0986.phot.1_sigma.fits psf.fit
NGC1433_36  NGC1433.phot.1_nonan.fits  336.808  359.475   13.93  0.1426  333.6  1.3 NGC1433.1mask_nonan.fits NGC1433.phot.1_sigma.fits psf.fit
NGC1566_36  NGC1566.phot.1_nonan.fits  706.519  716.103  -64.18  0.2518  416.1  1.3 NGC1566.1mask_nonan.fits NGC1566.phot.1_sigma.fits psf.fit
```

Additionally, the above procedure wrote a new procedure for checking the just-written SIGMA-images. Run this by the command

```
.run pipe3_example_make_sigma_driver.pro
```

which makes a lot of plots, comparing the calculated sigma with the actual standard deviation in the image (in circular zones). At sky region these should correspond each other (and they do, but only if the fudge-factor 2 is used).



To get rid of excess plots, delete the IDL windows with the command

```
wide
```

STEP 5: SEMI-AUTOMATIC MAKING *ingal* -FILES

Making template files for GALFIT iterations using the data in *"pipe3_ example.datalist"*.

At the moment, the template files created by the procedure `datalist_to_ingal.pro` include the following ("ID stands for galaxy id")

```
If the disk is not too inclined (disk ellipticity given in datalist-file)

ID_b.ingal              - 1 component fit initial values (Sersic bulge)
ID_bd.ingal             - 2 component fit initial values (Sersic bulge + exponential disk - fixed orientation)
ID_bdf.ingal            - 2 component fit initial values (Sersic bulge + exponential disk - free orientation)
```

```
ID_bdbarmodel.ingal    - 4 component template (Sersic bulge, expo disk, Ferrers bar, nucleus)

For an edge-on disk (disk ellipticity indicated as 'zzzzz')

ID_b.ingal             - 1 component fit initial values (Sersic bulge)
ID_bz.ingal            - 2 component fit initial values (Sersic bulge + edge-on disk)
ID_bzbarmodel.ingal    - 4 component template (Sersic bulge, edge-on disk, Ferrers bar, nucleus)
```

The 1 and 2-component templates should usually provide reasonably good starting values (= GALFIT iteration will converge to sensible final models).

The 4-component templates are likely to lead 'to 'crash" if used as they are.
Instead, utilize them as follows. First obtain a good 2-component fit, copy the *outgal*-file to a new name reflecting the used 3-component model, then copy/paste/edit the structure component from *ID_ bdbarmodel.ingal*, then run `galfit_run_new`, perhaps after first inspecting the initial values with `galfit_display_new`.

To make these ingal-files enter the command

```
.run datalist_to_ingal_driver_f,'pipe3_example'
```

The names of printed templates (without ingal suffix) are printed. Altogether the following files have been created

```
1-component models

   NGC0337_36_b.ingal
   NGC0428_36_b.ingal
   NGC0986_36_b.ingal
   NGC1433_36_b.ingal
   NGC1566_36_b.ingal

2-component models with exponential disk:

   NGC0337_36_bd.ingal
   NGC0428_36_bd.ingal
   NGC0986_36_bd.ingal
   NGC1433_36_bd.ingal
   NGC1566_36_bd.ingal

And copy/paste/edit templates:

   NGC0337_36_bdbarnmodel.ingal
   NGC0428_36_bdbarnmodel.ingal
   NGC0986_36_bdbarnmodel.ingal
   NGC1433_36_bdbarnmodel.ingal
   NGC1566_36_bdbarnmodel.ingal
```

The set of 5 example galaxies did not contain any edge-on galaxies, so no '_bz.ingal' files were created.

• **ALL IN ONE PROCEDURE:** `pipe3.pro`

The above STEPS 2 - 5 can be easily collected to a single IDL-routine `pipe3.pro`.

Given the LISTFILE (without the assumed *.galaxylist* suffix), the procedure will write the initial *datalist0*-file, stop while the user edits the file if necessary, then proceed with making of Sigma-images (and *.datalist*-file), and finally the ingal-files.

```
pipe3,'pipe3-example'
```

HOWEVER: be very careful when running things automatically!

```
[heikki@ikiturso ~/GALFIDL.dir]$ more pipe3.pro
;-------------------------------------------------------
pro pipe3,file,check=check
;-------------------------------------------------------
if(n_params() le 0) then begin
   print,'pipe3,file'
   print,'makes ingal-files for galaxies listed in'
   print,'      file.galaxylist'
   return
endif
;-------------------------------------------------------
listfile=file+'.galaxylist'
datalist0=file+'.datalist0'

s4g_p2_check_driver_f,listfile,datalist0,check=check,wait=1

print,'PIPE3: New datalist-file created:'
print,listfile+'  -->  '+datalist0
print,'Edit '+datalist0+' and hit return when ready'
print,'or give "q"-> quit'
vast=''
read,vast
if(vast eq 'q') then goto,endi

print,'PIPE3: make SIGMA-files'
pipeline_2to3_interface_f_new,file

print,' '
print,'PIPE3: make INGAL-files'
datalist_to_ingal_driver_f,file

print,' '
print,'to run the just-created b,bd,bdf ingals?'
print,'enter the command:'
print,'.run '+listfile+'_driver'


endi:
end
;-------------------------------------------------------
```

STEP 6 MAKING GALFIT ITERATIONS IN 'BATCH' MODE

The `galfit_run_new` calls can be collected to procedures, in order to make several iterations.

Here is an example procedure for making the 1- and 2-component decompositions for the above five galaxies. (it also contains the 4-component templates, but these are commented out).

The procedure was created automatically by `datalist_to_ingal_driver_f.pro`, and is run by the command

```
.run pipe3_example_driver.pro
```

```
IDL> $more pipe3_example_driver.pro
;Automatically generated driver-procedure
;10-Jan-2010 21:02:01.00
new=1
ps=0

galfit_run_new,"NGC0337_36_b",ps=ps,new=new
galfit_run_new,"NGC0337_36_bd",ps=ps,new=new
galfit_run_new,"NGC0337_36_bdf",ps=ps,new=new
;galfit_run_new,"NGC0337_36_bdbarnmodel",ps=ps,new=new
galfit_run_new,"NGC0428_36_b",ps=ps,new=new
galfit_run_new,"NGC0428_36_bd",ps=ps,new=new
galfit_run_new,"NGC0428_36_bdf",ps=ps,new=new
;galfit_run_new,"NGC0428_36_bdbarnmodel",ps=ps,new=new
galfit_run_new,"NGC0986_36_b",ps=ps,new=new
galfit_run_new,"NGC0986_36_bd",ps=ps,new=new
galfit_run_new,"NGC0986_36_bdf",ps=ps,new=new
;galfit_run_new,"NGC0986_36_bdbarnmodel",ps=ps,new=new
galfit_run_new,"NGC1433_36_b",ps=ps,new=new
galfit_run_new,"NGC1433_36_bd",ps=ps,new=new
galfit_run_new,"NGC1433_36_bdf",ps=ps,new=new
;galfit_run_new,"NGC1433_36_bdbarnmodel",ps=ps,new=new
galfit_run_new,"NGC1566_36_b",ps=ps,new=new
galfit_run_new,"NGC1566_36_bd",ps=ps,new=new
galfit_run_new,"NGC1566_36_bdf",ps=ps,new=new
;galfit_run_new,"NGC1566_36_bdbarnmodel",ps=ps,new=new
```

By changing the keyword value to new=0, the same procedure can be run again, to display the iteration results. Adding ps=2 would make ps and png files etc.

This procedure will produce a lot of plots: 6 plots/model *3 models/galaxy* 5 galaxies = 90 plots. So it might be handier to run in pieces. Remember the `wide` command for deleting IDL windows.

The GALFIDL procedures are under development: for any problems connect heikki.salo@oulu.fi