

1. ¿Qué es GALFIT?

GALFIT¹ (Peng et al., 2002) es un *software* de análisis de datos que permite ajustar funciones bidimensionales a galaxias y fuentes puntuales directamente en imágenes digitales. La versión actual, al momento de escribir éste artículo, es la 3.0 y puede consultarse en Peng et al. (2010).

La principal diferencia que se puede encontrar entre GALFIT y otros paquetes de ajuste es que el primero utiliza funciones de dos variables

$$f_{model}(x, y) = \sum_{\nu=1}^m f(x, y; \alpha_1, \dots, \alpha_n), \quad (1)$$

para crear un modelo del objeto de estudio. En donde el modelo está dado por una suma de m funciones f_{ν} que dependen de la posición (x, y) y de n parámetros libres $(\alpha_1, \dots, \alpha_n)$. Además, permite ajustar un número arbitrario de funciones, entre las cuales se pueden encontrar las más comunes de la literatura: Sérsic, exponencial, de Vaucoulers, Nuker, Gaussiana, King, Moffat y PSF.

El *software* continúa desarrollandose y promete nuevas facilidades para la siguiente versión. En la fase actual permite reproducir formas mucho más complejas que el clásico modelo elíptico, incluyendo modos de Fourier, rotaciones, traslaciones, truncamientos, etc (Peng et al., 2010). Permite tener en cuenta condiciones de contorno independientes para acoplar o desacoplar parámetros en cada uno de los modelos utilizados.

Desde la página oficial puede descargarse el programa compilado para los sistemas operativos más utilizados y ejecutarse sencillamente con

```
$ galfit input_file
```

En las secciones siguientes describiremos los aspectos más importantes del contenido necesario de `input_file` para que el programa pueda funcionar correctamente.

1.1. Diferencias con otros algoritmos de ajuste

La diferencia fundamental entre GALFIT y métodos de ajuste de perfiles de brillo unidimensionales (i.e. *ellipse* en IRAF) es el uso de funciones de dos variables (Ecuación 1). Por otro lado, GALFIT permite crear modelos más complejos: dos o más componentes; composición de varias fuentes desplazadas entre sí; permite formas más generales que las clásica elíptica; etc.

La salida de GALFIT consiste es una imagen que corresponde al modelo generado, éste puede analizarse mediante IRAF para cuantificar la evolución de los parámetros geométricos $(\epsilon, \theta, a_4, b_4)$ a lo largo del semieje mayor a . Pero ésta información no es proporcionada directamente por GALFIT, sólo brinda los coeficientes ajustados de cada componente.

2. ¿Cómo funciona GALFIT?

GALFIT está basado en el algoritmo de minimización Levenberg-Marquardt (Levenberg 1944), que permite encontrar la optima solución para el ajuste. Para determinar la bondad del mismo se utiliza el estadístico χ^2 normalizado

¹<http://users.obs.carnegiescience.edu/peng/work/galfit/galfit.html>

$$\chi^2_\nu = \frac{1}{N_{DOF}} \sum_{x=1}^{nx} \sum_{y=1}^{ny} \frac{(f_{data}(x, y) - f_{model}(x, y))^2}{\sigma(x, y)^2}, \quad (2)$$

en donde $\sigma(x, y)$ es la imagen de peso de cada pixel. Si ésta imagen no se da de entrada, GALFIT la genera en base a ciertos *keywords* en el *header* de la imagen: EXPTIME, GAIN, RDNOISE², NCOMBINE².

Como vemos en las ecuaciones 1 y 2, para completar el `input_file` se debe tener información del tamaño de la imagen de entrada, se debe comprobar que los *keywords* de la imagen sean correctos y se debe completar cada uno de los items (A a P) de la primer parte del archivo de configuración: IMAGE and GALFIT CONTROL PARAMETERS.

En la segunda parte de la configuración del GALFIT, INITIAL FITTING PARAMETERS, se puede agregar un párrafo por cada función que querramos ajustar a la distribución de brillo de la imagen. Como dijimos al comienzo, se pueden agregar un número arbitrario de funciones. Cada función posee parámetros específicos y cada uno de ellos debe tener un valor inicial, como se muestra en la Figura 1.

Este valor inicial puede, por ejemplo, obtenerse de una salida de la tarea *ellipse* de IRAF sobre el objeto y puede/debe mejorarse a medida que se realizan nuevas corridas de GALFIT. Para conocer cuales son las funciones disponibles, consulte la documentación provista por el desarrollador. Nótese que existe una función llamada “sky” que permite proponer un valor del nivel de cielo y un gradiente. Eventualmente puede fijarse éste nivel en un valor nulo si se poseen imagenes ya corregidas por este efecto.

Existen una serie de parámetros (disponibles a partir de la versión 3.0) que permiten romper la simetria elipsoidal del ajuste. Los mismos son: **C0** para el grado de *Diskyness/Boxyness*, **FN** para la corrección por modos de Fourier de orden N, **BN** para los modos de flexión, **RN** para la rotación de coordenadas y **T[ioN]** para el truncamiento del perfil. Es recomendable que estas perturbaciones se apliquen una vez que se halla obtenido una convergencia satisfactoria del modelo propuesto. De esta forma, es posible llevar el modelo clásico del perfil de brillo a uno más realista.

2.1. Un ejemplo

Como ya se dijo, GALFIT modela el objeto siguiendo las directivas del `input_file`. El producto final es una imagen (cuyo nombre debe especificarse, ver parámetro **B** en la Figura 1) que consta de tres capas. La primer capa es la imagen original recortada al tamaño de ajuste (**H**), la segunda capa es el modelo producido por GALFIT, y la tercera es la resta de las capas anteriores.

A modo de ejemplo, en La Figura 2 se muestra las tres capas de la imagen de salida producto de aplicar GALFIT sobre imagenes tomadas con la cámara WFPC2 del HST (Peng et al. 2002).

La serie superior (a) se produjo modelando tres componentes: bulbo, disco y barra, mientras que en la serie inferior (b) se modelaron además los brazos espirales. En la primer columna se observa la imagen original. En la segunda el modelo producido por GALFIT y en la tercer columna se ven los residuos del ajuste. Se debe notar que el ajuste con cuatro componentes sigue mejor los brazos espirales y los residuos en la parte central son mejores.

²No se utilizan a partir de la versión 3.0.

```

=====
# IMAGE and GALFIT CONTROL PARAMETERS
A) gal.fits          # Input data image (FITS file)
B) imgblock.fits    # Output data image block
C) none             # 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) none            # Bad pixel mask (FITS image or ASCII coord list)
G) none            # File with parameter constraints (ASCII file)
H) 1 93 1 93       # Image region to fit (xmin xmax ymin ymax)
I) 100 100         # Size of the convolution box (x y)
J) 26.563          # Magnitude photometric zeropoint
K) 0.038 0.038    # Plate scale (dx dy) [arcsec per pixel]
O) regular         # Display type (regular, curses, both)
P) 0               # Options: 0=normal run; 1,2=make model/imgblock & quit

# THE OBJECT LIST BELOW can be however long or short as the complexity
# requires. The user has complete freedom to mix and match the components
# by duplicating each object block.

# INITIAL FITTING PARAMETERS
#
# column 1: Parameter number
# column 2:
#
# -- Parameter 0: the allowed functions are: sersic, nuker, expdisk
# edgedisk, devauc, king, moffat, gaussian, ferrer, psf, sky
# -- Parameter 1-10: value of the initial parameters
# -- Parameter C0: For diskiness/boxiness
# <0 = disky
# >0 = boxy
# -- Parameter Z: Outputting image options, the options are:
# 0 = normal, i.e. subtract final model from the data to create
# the residual image
# 1 = Leave in the model -- do not subtract from the data
#
# column 3: allow parameter to vary (yes = 1, no = 0)
# column 4: comment

# Sersic function

0) sersic          # Object type
1) 300. 350. 1 1  # position x, y [pixel]
3) 20.00 1        # total magnitude
4) 4.30 1        # R_e [Pixels]
5) 5.20 1        # Sersic exponent (deVauc=4, expdisk=1)
9) 0.30 1        # axis ratio (b/a)
10) 10.0 1       # position angle (PA) [Degrees: Up=0, Left=90]
Z) 0             # Skip this model in output image? (yes=1, no=0)

# sky

0) sky
1) 0.77 0        # sky background [ADU counts]
2) 0.000 0      # dsky/dx (sky gradient in x)
3) 0.000 0      # dsky/dy (sky gradient in y)
Z) 0            # Skip this model in output image? (yes=1, no=0)

```

Figura 1: Ejemplo de archivo de configuración para ajustar un modelo de Sérsic y un valor representativo del nivel de cielo.

2.2. Estrategias de uso

En forma resumida, se enumeraran ciertas cuestiones a tener en cuenta antes de proceder a utilizar GALFIT:

1. Se debe proponer una región de convolución que contenga toda la luz de la galaxia, aunque en las primeras pruebas pueden utilizarse regiones que solo abarquen la zona central. De ésta forma, se pueden encontrar los valores iniciales más rápidamente.
2. Se debe corroborar que el valor del nivel de cielo es el adecuado.

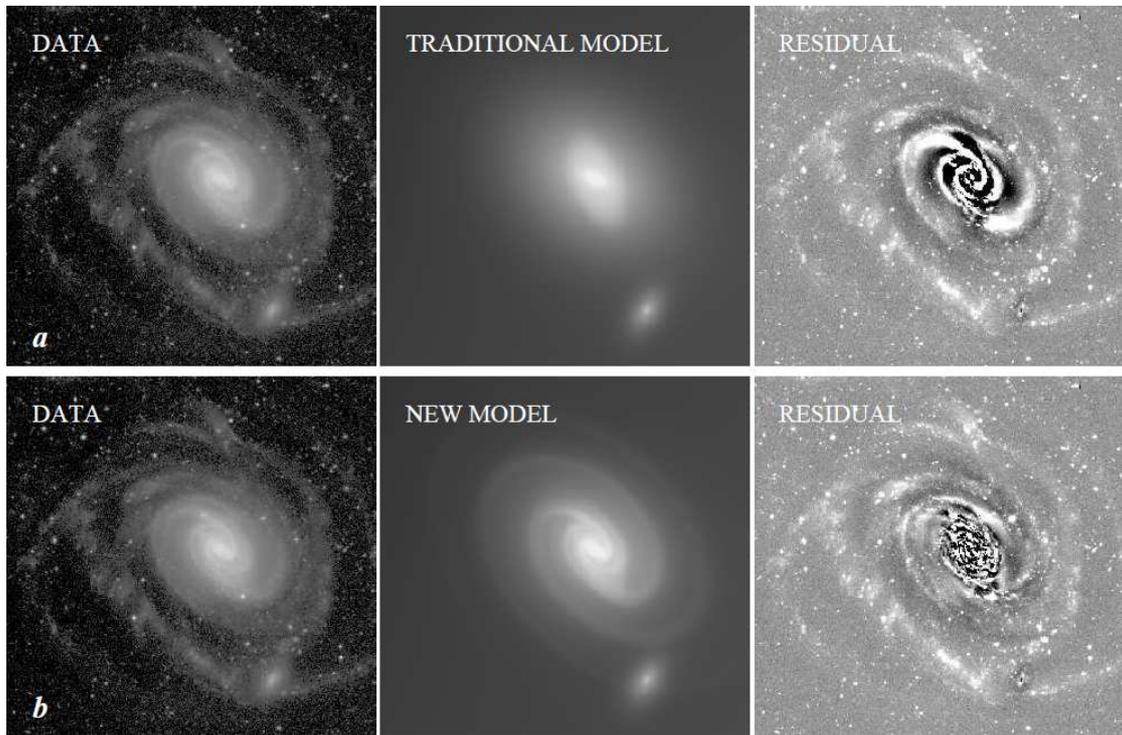


Figura 2: Ejemplo de un ajuste para la galaxia NGC 289.

3. Es recomendable comprobar la imagen de peso que produce GALFIT en cada iteración.
4. Es recomendable proporcionar una PSF (parámetro **D** en la Figura 1). Si GALFIT no la encuentra, no se realizara la convolución.
5. Si se encuentran objetos que son de interés, es necesario proporcionar una máscara para que GALFIT no los tenga en cuenta. La opción *segmentation* de SExtractor puede ser útil para esto.
6. Al finalizar cada iteración, se debe comprobar que la tercer capa de la imagen de salida, presente residuos consistentes. Es decir, si se encuentra en la imagen de residuos algun objeto, es posible que se deba comprobar si el número de componentes es el adecuado para el ajuste o si el nivel de cielo es el correcto.
7. El valor esperado de χ^2_ν para el ajuste por mínimos cuadrados es 1. Por lo que se espera que el mejor modelo sea aquel en el que χ^2_ν sea muy próximo.

Bibliografía

Levenberg, K.: 1944, The Quarterly of Applied Mathematics **2**, 164

Peng, C. Y., Ho, L. C., Impey, C. D., and Rix, H.-W.: 2002, AJ **124**, 266

Peng, C. Y., Ho, L. C., Impey, C. D., and Rix, H.-W.: 2010, AJ **139**, 2097