

# Introducción a la Astrofísica

## Cuaderno de Prácticas

Curso 2004 – 2005

Actividad enmarcada dentro del proyecto de Innovación Docente “*Un observatorio astronómico en el aula*”.  
Financiado por el Vicerrectorado de Planificación, Calidad y Evaluación Docente.  
Con la colaboración del Instituto de Astrofísica de Andalucía.

**Asignatura:** *Introducción a la Astrofísica – Cuaderno de Prácticas, curso 2004-2005*

**Proyecto:** *Un observatorio astronómico en el aula*



# 1.- OBSERVACIÓN VISUAL DEL CIELO NOCTURNO (I)

## Objetivos

En esta práctica pretendemos visualizar sobre la esfera celeste y sin ayuda de instrumentos ópticos los elementos clave de la astronomía de posición. Previamente, el profesor habrá explicado en clase el manejo del planisferio celeste. Durante la salida, los alumnos tratarán de “dibujar” sobre el cielo las líneas y círculos máximos que definen los distintos sistemas de coordenadas celestes, entendiendo su razón de ser.

Se tratarán los siguientes conceptos con el objeto de que el alumno se familiarice con ellos:

- Orientación en el espacio y en el tiempo:
  - Identificación de la estrella polar y los puntos cardinales
  - Rotación de la Tierra
  - Tiempo civil y tiempo universal (UT)
- Puntos, líneas y planos principales sobre la esfera celeste:
  - Genit, nadir, polos.
  - Plano del horizonte
  - Línea del meridiano
  - Línea de la eclíptica
- Coordenadas horizontales: acimut y elevación
- Coordenadas ecuatoriales: ascensión recta y declinación
- Referencia local de ascensión recta; ángulo horario y tiempo sidéreo.
- Culminaciones, ortos y ocasos
- Reconocimiento de las principales constelaciones observables en el momento de la práctica.
- Brillo aparente: concepto de magnitud aparente de una estrella
- Efectos de la atmósfera en la observación astronómica y contaminación lumínica

## Material necesario

- Planisferio celeste: al menos uno por cada dos alumnos.
- Linterna de luz roja
- Puntero láser
- Ropa de abrigo

## Lugar y fecha de realización

Idealmente, la práctica se llevará a cabo en un lugar de atmósfera limpia y apartado de las luces de la ciudad. No obstante, por facilidad logística se puede considerar otro lugar de observación cercano a las aulas de docencia.

El momento óptimo para llevar a cabo esta práctica será inmediatamente después de la explicación en clase del tema de astronomía de posición.



## 2.- OBSERVACIONES CON TELESCOPIO EN MODO REMOTO

### Objetivos

En esta práctica realizaremos desde el aula observaciones astronómicas en modo remoto utilizando un telescopio de tipo Schmidt-Cassegrain de 26.5 cm de abertura situado en el Instituto de Astrofísica de Andalucía (IAA)<sup>1</sup>. El modo de realización de las observaciones que se plantean en esta práctica se aproxima bastante a cómo se hacen en los centros astronómicos profesionales, en los que el astrónomo no se sitúa a pie de telescopio, sino en una sala aparte, o incluso en su propio centro de investigación a cientos de kilómetros de distancia.

El empleo de un telescopio dotado de cámara CCD permite la observación de objetos que pasarían desapercibidos mediante observación ocular. Por otro lado, permite disfrutar de la observación a una amplia audiencia.

Esta práctica persigue los siguientes objetivos:

- Explicar y familiarizar al alumno con algunos de los elementos básicos de un observatorio astronómico real: telescopio, cámara CCD, filtros fotométricos, problemas de apuntado, problemas de la atmósfera, etc.
- Presentación del software de efemérides, software de control del telescopio y de control de la cámara CCD.
- Mostrar algunos de los objetos astronómicos más llamativos observables en el momento de realización de la práctica (estrellas binarias, planetas, nebulosas, cúmulos de estrellas, galaxias, etc.), explicando sus propiedades fundamentales y estableciendo la conexión con las clases de teoría.
- Proporcionar un primer contacto del alumno con el problema de tratamiento de imágenes por ordenador.

### Material necesario

- Ordenador portátil con conexión a internet
- Proyector de vídeo
- Telescopio controlable por ordenador
- Cámara CCD astronómica o cámara de vídeo de alta sensibilidad acoplada al telescopio.
- Paquete de software MatLab<sup>2</sup>

### Lugar y fecha de realización

La práctica se llevará a cabo en el aula de docencia, en coordinación con personal del IAA.

El periodo óptimo para llevar a cabo esta práctica es durante el horario de invierno, antes del cambio de hora civil, de modo que se alcance la oscuridad en horas de clase (19:00 a 21:00 horas).

---

<sup>1</sup> Para más información consultar la página web <http://www.iaa.csic.es/peti/peti.html>

<sup>2</sup> La Universidad de Granada tiene instalado el paquete MatLab en todas las aulas de informática.

## Modo de realización

El profesor expondrá brevemente las características principales de un observatorio astronómico real, mostrando fotografías del instrumental que se utilizará en esta práctica: telescopio, cámara CCD o cámara de vídeo de alta sensibilidad, filtros fotométricos.

Se describirán brevemente los programas informáticos utilizados para realizar las observaciones. En nuestro caso:

- Programa de efemérides “*The Sky*”
- Programa de control de la cámara CCD “*Maxim DL/CCD*”
- Conexión a escritorio remoto a través de internet.

El programa de efemérides hace uso de todos los contenidos expuestos en el tema de “Astronomía de posición” para conseguir la localización de objetos celestes a partir de las coordenadas ecuatoriales del astro, de la posición del observador sobre la superficie de la Tierra y de la fecha y hora de observación. Es capaz también de controlar el movimiento del telescopio, apuntándolo a la zona del cielo que se desea observar.

Finalmente, estudiando el programa de efemérides, seleccionaremos aquellos objetos astronómicos que ofrezcan buenas condiciones de observación a la hora de realización de la práctica. Tomaremos distintas imágenes de los mismos, comentando sus características principales.

Las imágenes obtenidas se harán accesibles al alumno a través de la página web del proyecto “*Un observatorio astronómico en el aula*” (<http://www.ugr.es/~lucas/Index.htm>). Las imágenes, en formato FITS, podrán ser cargadas con MatLab usando la rutina *rfits*. Posteriormente pueden ser convertidas al formato estándar *jpeg* utilizando la rutina *fit2jpg* (ver ejemplos). Esta rutina requiere como entrada el nombre que queremos dar al archivo con la imagen *jpeg*. Determina el nivel mínimo y máximo de cuentas por pixel, permitiendo cambiar el rango con que se mostrará la imagen. También permite jugar con un parámetro de compresión (puede tomar valores positivos y negativos; cero significa sin compresión) que controla como se representan en la imagen final las variaciones de brillo. La rutina *fit2jpg* proporciona una versión en falso color y otra en niveles de grises de las imágenes tratadas. Tanto *rfits* como *fit2jpg* están disponibles en la página web del proyecto<sup>3</sup>.

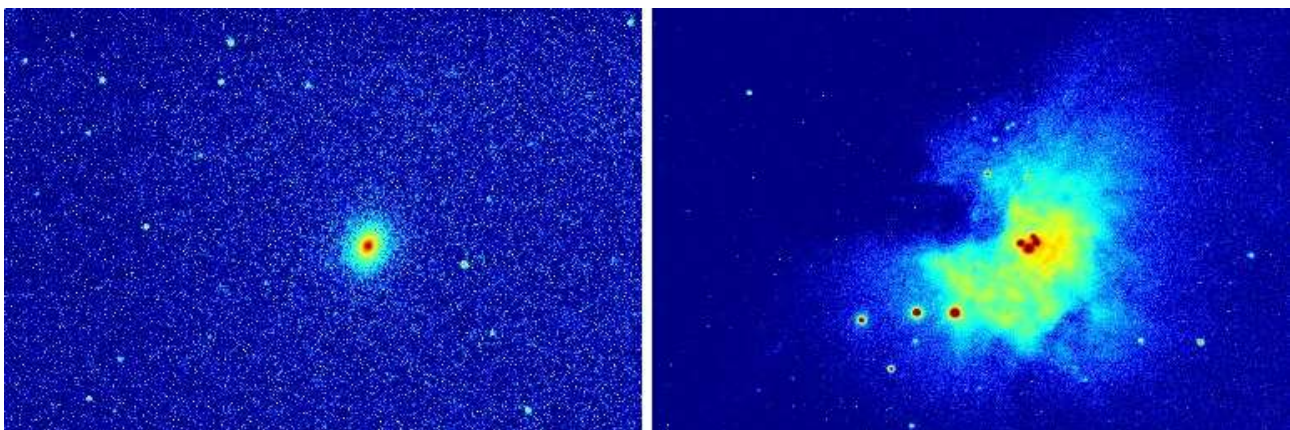


Figura 1: Imágenes en falso color de M32 (galaxia satélite de Andrómeda) y de la nebulosa de Orión

<sup>3</sup> Las rutinas de MatLab *rfits* y *fit2jpg* han sido realizadas por el Dr. Ángel de la Torre Vega.

### 3.- LOS SATÉLITES GALILEANOS DE JÚPITER

#### Introducción

Los cuatro satélites mayores de Júpiter fueron descubiertos por Galileo Galilei en 1610, por lo que son conocidos como satélites galileanos. La noche del 7 de Enero, Galileo observó tres pequeños objetos (Io, Europa, Ganímedes) próximos a Júpiter y cuando la noche del 8 de Enero apuntó su telescopio de nuevo a Júpiter, observó que las posiciones de éstos habían cambiado [1]. Los satélites galileanos son, por orden creciente de distancia a Júpiter, Io, Europa, Ganímedes y Calisto (este último menos brillante y con un periodo orbital mayor, por lo que no fue identificado como satélite de Júpiter en esta primera observación). Las principales características de estos satélites están recogidas en la tabla siguiente.

Satélite	Masa ( $10^{20}$ kg)	Radio (km)	Albedo visual geom.	Semieje mayor ( $10^3$ km)	Semieje mayor (Rad-Jup)	Periodo orbital (días)	Inclinación Orbital (grados)	Excentricidad órbita
Io	893.2	1821.6	0.62	421.6	5.91	1.769138	0.04	0.004
Europa	480.0	1560.8	0.68	670.9	9.40	3.551181	0.47	0.0101
Ganímedes	1481.9	2631.2	0.44	1070.4	14.97	7.154553	0.21	0.0015
Calisto	1075.9	2410.3	0.19	1882.7	26.33	16.689018	0.51	0.007

Tabla 1: Principales características de los satélites Galileanos [2].

El estudio de los satélites galileanos es muy interesante por varias razones. En primer lugar, su observación es posible con instrumental sencillo, siendo observables incluso con unos prismáticos de calidad. Por otra parte, los periodos orbitales son suficientemente cortos como para apreciar cambios en el sistema en una misma sesión de observación, por lo que constituyen un sistema ideal para el estudio de movimientos orbitales. A lo largo de la historia, además, han jugado un papel importante en la concepción del Universo. Constituyeron una prueba que desmontaba definitivamente la concepción geocéntrica del Universo. Permitieron la primera medición de la velocidad de la luz, llevada a cabo por el astrónomo danés Olaus Roemer en 1676, quien observó que los eclipses de los satélites galileanos presentaban desfases dependientes de la distancia entre la Tierra y Júpiter, que podían atribuirse a la velocidad de propagación de la luz [3]. La observación de los satélites galileanos y sus eclipses ha sido también utilizada por navegantes y cartógrafos de los siglos XVII, XVIII y XIX como cronómetro universal para la determinación de la longitud terrestre.

En esta práctica se pretende realizar observaciones telescópicas de los satélites galileanos de Júpiter. Tras la identificación de los satélites y determinación de sus posiciones, se propone comparar las posiciones observadas con las predicciones proporcionadas por un modelo orbital del sistema de satélites galileanos, con la ayuda del ordenador.

#### Objetivos

En esta práctica se persigue un doble objetivo. Por una parte la observación de los satélites de Júpiter y la introducción las características orbitales del sistema. Por otra parte se pretende acercar al alumno al uso del ordenador y del software matemático como herramienta para la utilización de modelos teóricos y la comparación de los resultados proporcionados por el modelo con las observaciones.

Los objetivos concretos de esta práctica son:

**Asignatura:** Introducción a la Astrofísica – Cuaderno de Prácticas, curso 2004-2005

**Proyecto:** Un observatorio astronómico en el aula

- Observar los satélites galileanos<sup>4</sup>.
- Observar el movimiento de los satélites en el intervalo de tiempo en el que se desarrolla la práctica.
- Si hay tiempo, se observará también el planeta Saturno, sus anillos y su sistema de satélites.
- Entender el concepto de rango dinámico de una imagen.
- Identificar los distintos satélites y determinar su posición con respecto al planeta Júpiter.
- Determinar la hora exacta de la observación a partir de las posiciones de los satélites mediante el uso de software específico.

## Material necesario

- Ordenador portátil y conexión a Internet.
- Proyector de vídeo.
- Telescopio PETI del IAA.
- Cámara CCD acoplada al telescopio.
- Paquete de software MatLab<sup>5</sup>.

## Lugar y fecha de realización

La práctica se realizará en el propio aula, en coordinación con el personal del IAA, antes del cambio de hora civil, de modo que se alcance la oscuridad en horas de clase (19:00 a 21:00 horas).

## Modo de realización

### 1.- Adquisición de las imágenes

Una vez finalizadas las observaciones con el telescopio, se proporcionará al alumno una o varias imágenes en formato FITS (Flexible Image Transport System), formato comúnmente utilizado en investigación astronómica. La imagen podrá ser descargada de la página web del proyecto "Un observatorio astronómico en el aula": <http://www.ugr.es/~lucas/observatorio/Index.htm>.

Esta imagen podrá ser cargada dentro del paquete MatLab (con la rutina "rfits") y transformada a formato jpeg (con la rutina "fit2jpg"<sup>6</sup>). La rutina "fit2jpg" transforma la imagen obtenida por la CCD del telescopio (número de cuentas en cada píxel de la CCD) en una imagen en niveles de gris (nivel de gris entre 0 y 255 para cada píxel). En principio, sería necesario transformar el rango dinámico de la imagen original (entre el número de cuentas del píxel más oscuro y el número de cuentas del píxel más brillante) al rango dinámico de la imagen de niveles de gris (entre 0 y 255). Sin embargo, para observar determinados detalles de la imagen, es conveniente modificar el rango dinámico de entrada y aplicar un factor de compresión en la transformación de "número-de-cuentas" a "nivel-de-gris". La rutina fit2jpg permite establecer tanto el rango dinámico de entrada como el factor de compresión para obtener una imagen jpg en la que se observen los detalles de interés, y genera un fichero jpg con una imagen en niveles de gris y un segundo fichero jpg con una imagen en falso color.

<sup>4</sup> La observación de Júpiter y sus satélites (así como la de Saturno) está condicionada a su posición en el cielo en el momento de realización de la práctica. Si no es posible su observación, se utilizarán para esta práctica imágenes de archivo adquiridas con la misma instrumentación.

<sup>5</sup> La Universidad de Granada tiene instalado el paquete MatLab en todas las aulas de docencia.

<sup>6</sup> Las rutinas de MatLab utilizadas en esta práctica se pueden descargar del proyecto "Un observatorio astronómico en el aula": <http://www.ugr.es/~lucas/observatorio/Index.htm>. Al final de este guión se incluye un apéndice donde se describen todas las rutinas utilizadas.



El alumno deberá cargar en MatLab el fichero en formato fits con la imagen de Júpiter, y deberá adaptar el rango dinámico de la imagen de entrada y el factor de compresión adecuadamente generando:

- Una imagen jpg en la que se observen las bandas de Júpiter.
- Una imagen jpg en la que se observen los satélites galileanos.

La imagen en niveles de gris en la que se aprecian los satélites es la que se utilizará para la siguiente parte de la práctica.

## 2.- Identificación de satélites y determinación de posiciones

La identificación de los satélites galileanos se puede realizar teniendo en cuenta el brillo (siempre y cuando los cuatro sean visibles). La secuencia ordenada, de más a menos brillante es: Ganímedes, Io, Europa, Calisto. La identificación de Calisto suele ser fácil por presentar un brillo mucho menor que el resto. La identificación del resto mediante inspección visual de la imagen es más complicada, y depende cómo se haya realizado la conversión de fits a jpg.

La determinación de la posición de cada satélite se suele expresar en radios de Júpiter, y vendrá dada por sus coordenadas en la dirección E-W y en la dirección S-N. La dirección E-W se puede establecer de forma aproximada ya que las órbitas de los cuatro satélites están aproximadamente en un mismo plano que sigue esta dirección.

Para facilitar la identificación de los satélites y la determinación de las posiciones se ha preparado una rutina en MatLab (denominada "galilean"), que permite introducir el fichero en formato jpg, y marcar cada uno de los objetos (tanto Júpiter como los cuatro satélites). Una vez marcados los satélites, los identifica teniendo en cuenta el brillo, determina el centroide asociado a cada objeto, calcula el radio de Júpiter, determina la dirección E-W, y finalmente proporciona las coordenadas de cada uno de los satélites (en las direcciones E-W y S-N) expresadas en radios de Júpiter.

El alumno deberá identificar los cuatro satélites y determinar sus posiciones con respecto a Júpiter, bien gráficamente o bien mediante la rutina de MatLab proporcionada.

Si se conoce la fecha y hora de observación, se puede comparar la imagen y las coordenadas obtenidas con las proporcionadas por un modelo del sistema de satélites galileanos. La rutina de MatLab "p\_sat2 (año,mes,día,hora,minuto,segundo)" calcula las posiciones de cada uno de los satélites y genera un diagrama donde se muestra cada uno de ellos, así como las coordenadas. Para representar las posiciones y obtener las coordenadas puede utilizarse también el programa JupSat95, realizado por Gary Nugent y accesible gratuitamente desde la dirección <http://indigo.ie/~gnugent/JupSat95> .

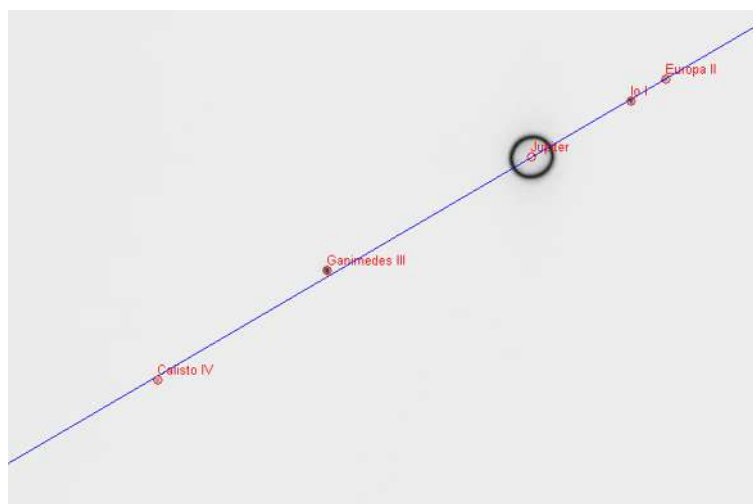


Figura 2: Resultado de la rutina galilean

**Asignatura:** Introducción a la Astrofísica – Cuaderno de Prácticas, curso 2004-2005

**Proyecto:** Un observatorio astronómico en el aula

### **3.- Determinación de la fecha y hora de observación (método gráfico)**

En el siguiente apartado de la práctica vamos a estudiar cómo se puede aplicar la observación del sistema de satélites galileanos para la determinación del instante de observación (aplicación como cronómetro universal).

Vamos a suponer que, de la imagen observada únicamente, conocemos el mes en que fue obtenida. Trataremos de obtener, con la mayor precisión posible, el instante en que se obtuvo. Para ello se pide crear un mapa con las posiciones de los satélites a lo largo del mes considerado, lo que puede hacerse con la rutina "crea\_mapa\_sat". Esta rutina genera un diagrama en el que se marcan, a lo largo del mes, la evolución de las posiciones de los satélites galileanos (coordenada E-W). En el diagrama habrá que marcar las posiciones de los satélites (obtenidas de la imagen, gráficamente o con la ayuda de la rutina "galilean"). Esto permite verificar los instantes de tiempo compatibles con las posiciones observadas. Una vez localizado el día, se puede generar un nuevo diagrama para el día obtenido (con la rutina "crea\_mapa\_sat\_1d") para encontrar el instante de observación. Los diagramas con las posiciones de los satélites pueden también obtenerse con el programa JupSat95.

### **4.- Determinación de la fecha y hora de observación (método asistido por ordenador)**

La determinación de la fecha y hora de observación por el método anterior está sujeta a varias fuentes de error, que reducen la precisión del resultado. Las coordenadas de los satélites, medidas en radios de Júpiter, dependen de la estimación del radio del planeta, y ésta puede variar dependiendo del criterio seguido para transformar la imagen fits a jpg. Puede haber también ligeros errores en la determinación de la dirección E-W o en la estimación del centroide de Júpiter. Además hay que tener en cuenta que los modelos que describen el movimiento de los satélites (para generar los diagramas de posiciones) son complejos e incluyen numerosas correcciones para mejorar la precisión, pero no son absolutamente precisos.

Para mejorar la precisión en la determinación de la fecha y hora de observación, se ha desarrollado la rutina en MatLab "galilean\_ajuste", similar a la rutina "galilean" pero que incluye una búsqueda de la fecha de observación en un periodo de dos años, donde se hace una optimización conjunta del instante de observación, el radio de Júpiter, las coordenadas del centroide de Júpiter y el ángulo del eje de coordenadas E-W, minimizando la distancia entre las posiciones de los objetos observados y las predichas por el modelo teórico.

Se pide ejecutar la rutina buscando el instante de observación en el año que incluya la fecha de observación y analizar los resultados proporcionados. Repetir el procedimiento fijando el año de observación a un valor distinto y comparar los resultados.

## **Referencias**

[1] Gamow, George. "Biografía de la Física". Alianza Editorial, Madrid, 2001. (Traducción de "Biography of Physics", Harper & Brother Publishers, New York, 1961). Capítulo 2: "La Edad Media y el Renacimiento".

[2] Sitio web del "National Space Science Data Center" de la NASA, direcciones:

<http://nssdc.gsfc.nasa.gov/planetary/planetfact.html>

<http://nssdc.gsfc.nasa.gov/planetary/factsheet/joviansatfact.html>.

[3] Gamow, George. "Biografía de la Física". Alianza Editorial, Madrid, 2001. (Traducción de "Biography of Physics", Harper & Brother Publishers, New York, 1961). Capítulo 6: "La Revolución Relativista".

[4] Meeus, Jean. "Astronomical Algorithms". Ed. Willmann-Bell, 1991.

**Asignatura:** Introducción a la Astrofísica – Cuaderno de Prácticas, curso 2004-2005

**Proyecto:** Un observatorio astronómico en el aula

## Apéndice: Macros y funciones de MatLab desarrolladas para esta práctica

Se puede obtener información on-line sobre cada una de las rutinas con el comando "help rutina" ejecutado en la ventana de MatLab.

### Macros y funciones básicas:

#### **rfits.m**

Descripción:

Macro de MatLab para leer ficheros en formato fits. El nombre del fichero fits a leer se introduce interactivamente. La imagen se almacena en la matriz de datos B. La imagen se puede visualizar con `imagesc(B)` o con `imagesc(log10(B))`.

#### **fit2jpg.m**

Descripción:

Macro de MatLab para convertir ficheros fits (numero-de-cuentas) a ficheros jpg (nivel-de-gris). Lee la matriz de datos B (procedente de la rutina `rfits.m`). El nombre del fichero jpg de salida se introduce interactivamente. Genera un fichero jpg en nivel de gris y otro que representa los niveles de gris en falso color. Permite definir el rango dinámico de entrada (número de cuentas) que se mapea al rango dinámico de salida (nivel-de-gris entre 0 y 1). Para ello se introducen los valores mínimo y máximo del número de cuentas que definen el rango dinámico de entrada. También se introduce un factor de compresión para esta transformación. Cuando finaliza el proceso, pide confirmación del resultado. Si se acepta, escribe los ficheros jpg; si no se acepta, permite introducir de nuevo el rango dinámico de entrada y el factor de compresión.

Si se selecciona la variable `mediana_on=1` (hay que modificar la línea 33 del fichero `fit2jpg.m`), se aplica un filtro mediana de 3x3 para eliminar ruido impulsivo de tipo "salt and pepper" (pixels esporádicos con valores anómalos).

Para generar ficheros de salida con figuras (en general, válido para cualquier figura de MatLab):

- `figure(Nfigura)` selecciona la figura a generar.
- `print -depsc2 fichero.eps` genera fichero con la figura en formato eps color (postscript encapsulado).
- `print -dpsc2 fichero.ps` genera fichero con la figura en formato ps color (postscript).
- `print -dpdf fichero.pdf` genera fichero con la figura en formato pdf (a partir de versión 6 de MatLab).
- `print -djpeg fichero.jpg` genera fichero con la figura en formato jpg.
- `help print` información sobre otros formatos

#### **p\_sat.m**

Sintaxis:

`[x1,x2,x3,x4]=p_sat(y,m,d,h,min,s)`

Descripción:

Función. Obtiene la posición de los satélites de Júpiter (coordinada E-W para cada uno de los satélites galileanos, expresada en radios de Júpiter) para una fecha y hora determinada (y-year; m-month; d-day; h-hour; m-minute; s-second). Se ha desarrollado a partir de un programa en QBASIC, escrito por Keith Burnett ([kburnett@geocity.com](mailto:kburnett@geocity.com)), y bajado de la red desde la dirección <http://www.xylem.f2s.com/kepler/galileo.html>. El programa de Keith Burnett, a su vez, ha sido desarrollado a partir de los algoritmos presentados en el libro "Astronomical Algorithms" de Jean Meeus [4].

#### **p\_sat1.m**

Sintaxis:

`[x1,x2,x3,x4,y1,y2,y3,y4]=p_sat1(y,m,d,h,min,s)`

Descripción:

Función. Versión mejorada de la función `p_sat.m`. Proporciona las coordenadas de los satélites en las direcciones E-W y S-N (en radios de Júpiter). Incluye modificaciones para tener en cuenta las correcciones por el paso del calendario juliano al gregoriano (fechas anteriores al 15-oct-1582). Utiliza ángulos corregidos para mejorar ligeramente la precisión. Se ha desarrollado a partir de un programa escrito para MathCad por Martín Zombeck (Cambridge Center for Astrophysics, Cambridge MA), que está también

**Asignatura:** *Introducción a la Astrofísica – Cuaderno de Prácticas, curso 2004-2005*

**Proyecto:** *Un observatorio astronómico en el aula*

basado en los algoritmos de Jean Meeus [4].

### **p\_sat2.m**

Sintaxis:

[x1,x2,x3,x4,y1,y2,y3,y4]=p\_sat2(y,m,d,h,min,s)

Descripción:

Función. Versión similar a p\_sat1.m, que incluye una presentación por pantalla (un diagrama con las posiciones de los satélites en la fecha solicitada). Tanto p\_sat.m como p\_sat1.m admiten que uno de los parámetros que especifican la fecha u hora sea un vector, en cuyo caso proporcionan las coordenadas para un conjunto de fechas y horas. Sin embargo, p\_sat2.m exige que todos los parámetros que especifican fecha y hora sean escalares y sólo es posible la representación para una fecha y hora.

### **crea\_mapa\_sat.m**

Descripción:

Macro de MatLab que genera un diagrama de posiciones de los satélites para el mes especificado. Los datos de entrada se introducen interactivamente. Genera el diagrama en formato postscript. El diagrama es creado en el fichero "mapa.ps".

### **crea\_mapa\_sat\_1d.m**

Descripción:

Macro de MatLab que genera un diagrama de posiciones de los satélites para el día especificado. Los datos de entrada se introducen interactivamente. Genera el diagrama en formato postscript. El diagrama es creado en el fichero "mapa\_1d.ps".

### **galilean.m**

Descripción:

Macro de MatLab para analizar imágenes de los satélites galileanos. Se introduce interactivamente el nombre del fichero con la imagen de Júpiter. La macro prepara la imagen y calcula un histograma de niveles de gris. Pide el nivel de gris a partir de cual se va a considerar iluminación de fondo (para la localización de objetos). La macro permite marcar objetos (Júpiter y los 4 satélites). Para ello hay que marcar con el ratón un polígono alrededor del objeto y pulsar ENTER, repitiéndolo para cada objeto. Una vez marcados los objetos la macro identifica los satélites (teniendo en cuenta el brillo), calcula el radio de Júpiter y determina la dirección E-W (haciendo una regresión con las posiciones de los satélites). Finalmente proporciona las coordenadas en las direcciones S-N y E-W expresadas en radios de Júpiter.

### **galilean\_ajuste.m**

Descripción:

Macro de MatLab para analizar imágenes de los satélites galileanos. Realiza una estimación de la fecha que mejor ajusta las posiciones de los satélites. Para ello usa la función "optimiza\_fecha\_varios.m". (Consultar esta función para más detalles). Se introduce interactivamente el nombre del fichero con la imagen de Júpiter. Se introduce también el año en el que se va a buscar la fecha y hora que mejor ajusta las posiciones de los satélites. La macro prepara la imagen y calcula un histograma de niveles de gris. Pide el nivel de gris a partir de cual se va a considerar iluminación de fondo (para la localización de objetos). La macro permite marcar objetos (Júpiter y los 4 satélites). Para ello hay que marcar con el ratón un polígono alrededor del objeto y pulsar ENTER, repitiéndolo para cada objeto. Una vez marcados los objetos la macro identifica los satélites (teniendo en cuenta el brillo), calcula el radio de Júpiter y determina la dirección E-W (haciendo una regresión con las posiciones de los satélites).

Con las posiciones de los objetos identificados, realiza una búsqueda de las posibles fechas y horas en las que la configuración del sistema era parecida a la observada, optimizando distintos parámetros y presentando en las distintas figuras diagramas que permiten el seguimiento del proceso de optimización (consultar la función "optimiza\_fecha\_varios.m" para más detalles). Finalmente proporciona las coordenadas en las direcciones S-N y E-W expresadas en radios de Júpiter (valores optimizados) y la fecha y hora más probable en la que se obtuvo la imagen, a lo largo del periodo analizado.

**Asignatura:** Introducción a la Astrofísica – Cuaderno de Prácticas, curso 2004-2005

**Proyecto:** Un observatorio astronómico en el aula

## Otras funciones:

### **transform\_escalado.m**

Sintaxis:

[x1,y1,x2,y2,x3,y3,x4,y4]=transform\_escalado(px0,py0,px1,py1,px2,py2,px3,py3,px4,py4,R,b)

Descripción:

Función. Devuelve las posiciones de los satélites en radios de Júpiter a partir de las posiciones en pixels, el radio (R), y la pendiente de la línea E-W (b); px0 py0 es la posición de Júpiter en pixels. Esta función es usada por varias de las rutinas anteriores.

### **distancia\_cuad.m**

Sintaxis:

D=distancia\_cuad(x1,x1r,x2,x2r,x3,x3r,x4,x4r,y1,y1r,y2,y2r,y3,y3r,y4,y4r)

Descripción:

Función. Proporciona la distancia (error cuadrático) entre las posiciones observadas de los satélites y las de referencia (predichas por el modelo para una fecha determinada). Esta función es usada por las rutinas de optimización de fecha.

### **f\_range.m**

Sintaxis:

a = frange(x)

Descripción:

Función. Transforma el ángulo x (en radianes) en el mismo ángulo en el intervalo [0 2pi] si  $x > 0$ , o bien en el intervalo [-2pi 0] si  $x < 0$ . Esta función es usada por el modelo orbital, en las rutinas p\_sat.m, p\_sat1.m y p\_sat2.m.

### **circulo.m**

Sintaxis:

circulo(R,x0,y0,color,ancho)

Descripción:

Función. En la figura activa, pinta un círculo, de radio R, centrado en (x0,y0) y de color ' color'. La anchura de la línea viene determinada por el valor ' ancho'. Función utilizada por p\_sat2.m y por optimización\_fecha\_dia.m.

### **optimizacion\_fecha\_fast.m**

Sintaxis:

[D0,r0,b0,d0]=  
optimizacion\_fecha\_fast(px0,px1,px2,px3,px4,py0,py1,py2,py3,py4,R\_ini,B\_ini,year0,dia0)

Descripción:

Función. Algoritmo rápido para determinar la fecha y hora correspondiente a las posiciones de los satélites. Entrada: px0 py0: coordenadas de Júpiter (pixels); px1, py1, px2, py2, px3, py3, px4, py4: coordenadas de los satélites (en pixels); R\_ini, B\_ini: radio de Júpiter inicial y pendiente de la dirección E-W inicial (estos valores son optimizados); year0, día0: fecha donde se busca la hora que mejor ajusta las posiciones de los satélites.

Salida: D0: error cuadrático tras optimización; r0: radio de Júpiter tras optimización; b0: pendiente de la dirección E-W tras optimización; d0: fecha y hora tras optimización.

Procedimiento de optimización: primero se optimiza el tiempo (barrido en un día de minuto en minuto); para ese minuto se optimiza el radio; se recalcula el tiempo (con precisión de 15 segs); se optimiza la pendiente; se recalcula el tiempo (con precisión de 5 segundos). Para cada barrido se selecciona el valor que da menor error cuadrático entre los valores proporcionados por el modelo y

**Asignatura:** Introducción a la Astrofísica – Cuaderno de Prácticas, curso 2004-2005

**Proyecto:** Un observatorio astronómico en el aula

los observados en la imagen.

### **optimizacion\_fecha\_dia.m**

Sintaxis:

```
[r0,b0,px0new,py0new,d0,D0]=  
optimizacion_fecha_dia(px0,px1,px2,px3,px4,py0,py1,py2,py3,py4,R_ini,B_ini,year0,dia0)
```

Descripción:

Función. Algoritmo para determinar la fecha y hora correspondiente a las posiciones de los satélites. Optimiza: la fecha y hora; las coordenadas de Júpiter; el radio de Júpiter; la pendiente de la dirección E-W.

Entrada: px0 py0: coordenadas de Júpiter (pixels) (estos valores son optimizados); px1, py1, px2, py2, px3, py3, px4, py4: coordenadas de los satélites (en pixels); R\_ini, B\_ini: radio de Júpiter inicial y pendiente de la dirección E-W inicial (estos valores son optimizados); year0, dia0: fecha donde se busca la hora que mejor ajusta las posiciones de los satélites.

Salida: r0: radio de Júpiter tras optimización; b0: pendiente de la dirección E-W tras optimización; px0new, py0new coordenadas de Júpiter (pixels) tras optimización; fecha: fecha y hora tras optimización; D0: error cuadrático tras optimización.

Procedimiento de optimización: Realiza de forma iterativa varios ciclos de optimización. En cada ciclo se optimiza: r0, b0, px0, py0 y la fecha y hora. Para cada barrido se selecciona el valor que da menor error cuadrático entre los valores proporcionados por el modelo ajustado y los observados en la imagen.

Proporciona por pantalla un seguimiento del proceso de optimización. Dibuja en la figura 10 un diagrama con Júpiter y sus satélites, marcando las posiciones proporcionadas por el modelo ajustado (en colores: blanco, amarillo, rojo y verde para Io, Europa, Ganímedes y Calisto, respectivamente) y las observadas en la imagen (en magenta).

### **optimizacion\_fecha\_varios.m**

Sintaxis:

```
[r0,b0,px0new,py0new,fecha,D0]=  
optimizacion_fecha_varios(px0,px1,px2,px3,px4,py0,py1,py2,py3,py4,R_ini,B_ini,year0)
```

Descripción:

Función. Algoritmo para determinar la fecha y hora correspondiente a las posiciones de los satélites. Realiza una búsqueda de posibles coincidencias alrededor del año solicitado (desde Jul del año anterior hasta Jun del año siguiente). Encuentra el mejor candidato y después optimiza esta coincidencia. Optimiza: la fecha y hora; las coordenadas de Júpiter; el radio de Júpiter; la pendiente de la dirección E-W.

Entrada: px0 py0: coordenadas de Júpiter (pixels); px1, py1, px2, py2, px3, py3, px4, py4: coordenadas de los satélites (en pixels); R\_ini, B\_ini: radio de Júpiter inicial y pendiente de la dirección E-W inicial (estos valores son optimizados); year0: año donde se busca la fecha y hora que mejor ajusta las posiciones de los satélites.

Salida: r0: radio de Júpiter tras optimización; b0: pendiente de la dirección E-W tras optimización; px0new, py0new coordenadas de Júpiter (pixels) tras optimización; fecha: fecha y hora tras optimización; D0: error cuadrático tras optimización.

Procedimiento de búsqueda: hace un barrido en un periodo de dos años que contiene al año solicitado con intervalos de una hora, para buscar, con un criterio de mínimo error cuadrático, las posibles coincidencias entre las posiciones observadas y las obtenidas mediante el modelo. Busca los mejores candidatos. Optimiza los mejores candidatos con la rutina "optimiza\_fecha\_fast.m". Finalmente, optimiza el mejor candidato con la rutina "optimiza\_fecha\_dia.m".

Procedimiento de optimización: Véanse rutinas "optimiza\_fecha\_fast.m" y "optimiza\_fecha\_dia.m".

Proporciona por pantalla un seguimiento del proceso de optimización. Dibuja en la figura 7 el error cuadrático para el barrido por horas sin optimizar. Se representa el inverso del error, y está expresado en  $1/(\text{radios de Júpiter})^2$ . Cuanto mayor sea el valor, mejor es la coincidencia. Dibuja en la figura 8 el error cuadrático optimizado para los candidatos seleccionados para optimización. Se representa el logaritmo en base 10 del cociente entre el error del mejor y el error de cada uno de los candidatos (para el mejor, el valor obtenido es cero; para el resto, el valor es menor, más bajo cuanto peor es la coincidencia). Dibuja en la figura 9 el error cuadrático (en radios de Júpiter al cuadrado) para los 60 minutos anteriores y posteriores a la fecha y hora obtenida como resultado. Dibuja en la figura 10 un diagrama con Júpiter y sus satélites, marcando las posiciones proporcionadas por el modelo ajustado (en colores: blanco, amarillo, rojo y verde para Io, Europa, Ganímedes y Calisto, respectivamente) y las observadas en la imagen (en magenta).

**Asignatura:** *Introducción a la Astrofísica – Cuaderno de Prácticas, curso 2004-2005*

**Proyecto:** *Un observatorio astronómico en el aula*

## 4.- CORRECCIÓN DE UNA IMÁGEN ASTRONÓMICA

### Objetivos

Esta práctica enseñará al alumno el primer paso básico previo al análisis de una imagen astronómica obtenida con CCD. Se trata de correcciones de defectos debidos al propio detector CCD o a una iluminación no uniforme del mismo. Existe una amplia variedad de software específico para el tratamiento de imágenes astronómicas. No obstante, hemos preferido usar de nuevo el paquete MatLab. Por un lado, por tener licencia en la Universidad de Granada, y por otro, porque la utilización de un software no específico permitirá entender todos los pasos que se siguen hasta la obtención de la imagen final adecuada para obtener información científica.

Los objetivos perseguidos se pueden resumir en los siguientes puntos:

- Aprender los procesos básicos a los que hay que someter las imágenes astronómicas obtenidas con CCD como paso previo al posterior análisis de las mismas.
- Concepto de las correcciones “*dark*”, “*bias*” y “*flat field*”
- Aplicar estas correcciones sobre imágenes reales obtenida con un telescopio

### Material necesario

- Ordenador portátil
- Proyector de vídeo y conexión a internet
- Telescopio PETI del IAA
- Cámara CCD acoplada al telescopio
- Paquete de software MatLab<sup>7</sup>

### Lugar y fecha de realización

Las observaciones y la obtención de la imagen se realizará en el propio aula, en coordinación con personal del IAA. Conviene realizar las observaciones antes del cambio de hora civil, de modo que se alcance la oscuridad en horas de clase (19:00 a 21:00 horas).

La parte de corrección de imagen la realizará cada alumno con su propio ordenador o con los ordenadores disponibles en las aulas de informática de la facultad de Ciencias.

### Modo de realización

#### Conceptos básicos

Una imagen de un objeto astronómico obtenida mediante una cámara CCD consiste en un mapa de bits que almacena la información de cada celdilla de la cámara. La imagen en bruto está afectada por una serie de circunstancias que han de ser tenidas en cuenta y corregidas a la hora de obtener información de valor astrofísico. Estas circunstancias están asociadas a la propia electrónica de la CCD y a posibles

<sup>7</sup> La Universidad de Granada tiene instalado el paquete MatLab en todas las aulas de informática.

irregularidades en la iluminación del *chip*. Los principales efectos de debemos corregir son los siguientes (usamos la comúnmente utilizada nomenclatura en lengua inglesa):

**Bias:** se entiende por *bias* o sesgo, el ruido de base de la CCD. Por el mero hecho de estar conectada, existe una corriente electrónica que se traduce en un cierto número de cuentas en la CCD, es decir, de electrones atrapados en las celdillas de la CCD. Ese ruido de fondo hay que descontarlo. Para ello se toman varias exposiciones a tiempo cero y sin entrada de luz, realizando la mediana de todas ellas. Así construimos lo que se llaman *bias*. A la imagen en bruto hay que restarle el *bias*.

**Dark:** cuando realizamos una exposición larga sobre una CCD, aparte del número de cuentas inducidas por los fotones incidentes se producen cuentas adicionales debidas al propio ruido electrónico (ruido térmico). Para descontar este efecto, cuando tomemos una exposición de un objeto astronómico de tiempo T, deberemos tomar varias exposiciones sin entrada de luz (de ahí el nombre de *dark*), también durante un tiempo T y en las mismas condiciones. Conviene tomar al menos 3 *darks* para luego hacer la mediana de todos ellos. A la imagen astronómica hay que restarle el *dark*.

**Flat field:** la respuesta de la CCD a una fuente de iluminación no es uniforme, sino que hay variaciones según la celdilla que estemos tratando. Estas variaciones pueden ser debidas a partículas de polvo, temperatura no homogénea del chip, irregularidades en la CCD, viñeteo del telescopio, etc. Para corregir este problema se obtiene una imagen de un campo uniformemente iluminado, que puede ser una pantalla, o un fondo de cielo con pocas o ninguna estrella. En este último caso moveremos el telescopio en cada imagen para luego realizar la mediana de todas las imágenes y eliminar las estrellas que aparezcan. Las imágenes "*flat*" han de ser obtenidas en condiciones similares a las del objeto astronómico (mismo tiempo de exposición, misma posición de la cámara). A cada imagen *flat* le quitaremos el *dark*. Posteriormente, normalizamos todos los *flats* dividiéndolos por su media. Finalmente obtenemos nuestro *flat* definitivo realizando la mediana de todos los *flats*. Cuando tengamos el *flat* definitivo, dividiremos nuestra imagen astronómica por el *flat*.

Normalmente habremos realizado al menos tres exposiciones idénticas de nuestro objeto astronómico. A cada una de ellas, imagen en bruto, le restaremos el *bias* y el *dark*, y posteriormente la dividiremos por el *flat* bueno. Es decir, la imagen corregida se puede expresar de la siguiente forma:

$$\text{Imagen corregida} = (\text{Imagen en bruto} - \text{Bias} - \text{Dark}) / \text{flat}$$

Finalmente, hallaremos la mediana de todas las imágenes astronómicas corregidas, que será nuestra imagen final. Haciendo la mediana de las imágenes, de los *darks*, *flats* y *bias* conseguimos eliminar otra fuente de error que afecta a las imágenes astronómicas: el error aleatorio.

## **Metodología**

Tomaremos, mediante conexión remota al telescopio PETI del Instituto de Astrofísica de Andalucía, las imágenes astronómicas, así como las distintas imágenes necesarias para la posterior corrección de los efectos asociados a la CCD: *bias*, *darks* y *flats*. Las imágenes se obtendrán en formato FITS (Flexible Image Transport System), que podrán ser cargadas dentro del paquete MatLab mediante la rutina *rfits*. Usando rutinas propias de MatLab efectuaremos las operaciones necesarias hasta obtener la imagen final.

Existen varios paquetes de software a nivel de astrónomo aficionado que son específicos para la reducción de datos astronómicos. Por ejemplo, el programa Maxim-DL que usamos para controlar la CCD del PETI permite también realizar la corrección y calibración de las imágenes.



Tanto las imágenes como las rutinas de MatLab necesarias para la realización de esta práctica podrán obtenerse de la página web del proyecto “Un observatorio astronómico en el aula”:  
<http://www.ugr.es/~lucas/observatorio/Index.htm>

## **Procedimiento**

A continuación, presentamos las rutinas que deben ejecutarse con MatLab en cada uno de los casos.

- **Bias:** A la imagen en bruto queremos restarle el *bias*. Para ello, previamente habremos obtenido las imágenes del objeto astronómico y la imagen “*bias*”, ambas en formato FITS. En MatLab, dentro de nuestro directorio de trabajo donde copiaremos los procedimientos *rfits.m* y *fit2jpg.m*, escribiremos:
  - **rfits** Ejecuta la rutina *rfits.m* que permite “leer” una imagen FITS y transformarla en una matriz (A) que son los objetos con los que trabaja MatLab. Habrá que introducir el nombre del fichero de la imagen de nuestro objeto astronómico, sin la extensión “.fits”. Por ejemplo, **orion** para el caso del fichero *orion.fits*
  - **NAME=A;** Donde *NAME* será el nombre con el que queramos denominar a la variable imagen en bruto que MatLab ha convertido en una matriz columna (A).
  - **rfits** Volvemos a ejecutar esta rutina para “leer” ahora la imagen “*bias*”.
  - **NAME\_BIAS=A;** El programa ha obtenido una nueva matriz, A, a la que llamaremos *NAME\_BIAS*, por ejemplo.
  - **R = NAME – NAME\_BIAS;** Restamos las dos imágenes y a la matriz resultante la denotamos con la letra R.
  - **R\_BIAS = R;** Realizaremos esta copia de la matriz R, que denominaremos *R\_BIAS*, para poder utilizarla posteriormente.
  - **fit2jpg** Ejecutamos este procedimiento para obtener la imagen resultante en el formato estándar JPG.

Explicaremos con más detalle el procedimiento **fit2jpg.m**. Ésta es una rutina que nos proporcionará dos imágenes finales de nuestro objeto de interés, una en blanco y negro y otra en falso color, con la extensión “.jpg”. Para ello, inicialmente introduciremos un nombre para esa imagen resultante, por ejemplo **orion**, de manera que en nuestro directorio de trabajo, una vez que finalice la rutina, deberíamos tener dos nuevos ficheros, uno llamado **orion.jpg** y otro **orion\_color.jpg**.

El procedimiento lee la matriz resultante (R) obtenida anteriormente cuyos valores no son sino el número de cuentas en cada píxel, y nos proporciona el máximo y el mínimo número de cuentas por píxel. La imagen se representaría inicialmente en función de este número de cuentas, pero nos interesa más representarla en una escala de grises, por lo que realizaremos una transformación no lineal que nos relacionará el número de cuentas y los niveles en la escala de grises. Precisamente, debido a esa no linealidad, podrán modificarse los valores máximo y mínimo y un factor de compresión,  $\mu$ , que nos magnificará diferencias en un intervalo determinado dentro del rango total elegido. En el estándar europeo  $\mu$  varía entre 0 (sin compresión) y 255. Luego, iremos modificando estos tres parámetros hasta que estemos satisfechos con la imagen final.

- **Dark:** A la imagen astronómica hay que restarle el *dark*. Conviene tomar al menos 3 *darks* para luego hacer la mediana de todos ellos que nos dará el dark definitivo. En MatLab, siempre dentro de nuestro directorio de trabajo donde copiaremos los procedimientos *rfits.m* y *fit2jpg.m*, escribiremos:
  - **rfits** Permitirá a MatLab “leer” la primera imagen dark en formato FITS y transformarla en una matriz (A)
  - **DARK\_1=A;** *DARK\_1* será el nombre, por ejemplo, con el que denominaremos a esa primera variable imagen dark que MatLab ha convertido en una matriz columna (A).
  - **rfits** Volvemos a ejecutar esta rutina para “leer” ahora la segunda imagen dark que está en formato FITS.
  - **DARK\_2=A;** Llamaremos, por ejemplo, *DARK\_2* a la segunda variable imagen dark, que es una matriz con una sola columna.
  - **rfit** Cargamos finalmente la tercera imagen dark que está en formato FITS.
  - **DARK\_3=A;** *DARK\_3* será, por ejemplo, el nombre de la tercera variable imagen dark, que es una matriz columna.
  - **H = [DARK\_1 DARK\_2 DARK\_3];** Construyo una matriz de tres columnas. En cada fila se tiene el valor del número de cuentas para un mismo píxel en cada una de las imágenes dark tomadas. De esos valores para un mismo píxel elegiremos, a continuación, su mediana y a partir de ese valor construiremos la matriz *DARK mediana*.
  - **DARK = (median(H'))';**
  - **rfit** Cargamos ahora la imagen en bruto en formato FITS.
  - **NAME=A;** Le damos un nombre a la variable imagen en bruto.
  - **R = NAME - DARK;** Obtenemos así la matriz resultante (R)
  - **R\_DARK = R;** Realizaremos esta copia de la matriz R, que denominaremos *R\_DARK*, para utilizarla después.
  - **fit2jpg** Ejecutamos este procedimiento para obtener la imagen resultante en el formato estándar JPG.
- **Flat:** En este último caso realizaremos la mediana de todas las imágenes de forma similar a la anterior. A cada imagen *flat* le quitaremos el *dark*. Posteriormente, normalizamos todos los *flats* dividiéndolos por su media. Finalmente obtenemos nuestro *flat* definitivo realizando la mediana de todos los *flats*. Cuando tengamos el *flat* definitivo, dividiremos nuestra imagen astronómica por el *flat*.
  - **rfits**
  - **FLAT\_1 = A;**

- **F\_1 = FLAT\_1 - DARK;** Después de haber “leído” la primera imagen flat en formato FITS, la renombramos como FLAT\_1, por ejemplo, y le restamos el DARK. Repetiremos este proceso con todas las imágenes que tengamos.
- **MED\_1 = mean(F\_1);** Cada uno de los flats deberá dividirse por su media, que obtenemos de esta manera.
- **FN\_1 = F\_1 / MED\_1;** Llamamos FN a la imagen flat sin el dark, ya normalizada.
- **H = [FN\_1 FN\_2 FN\_3...FN\_n];** Con todas las imágenes normalizadas que tenemos construimos una matriz para poder calcular la mediana de todas ellas.
- **FLAT = (median(H'))';** Ésta será la imagen flat definitiva

Finalmente obtendremos la imagen corregida de la siguiente manera:

$$\text{Imagen corregida} = (\text{R\_BIAS} - \text{DARK}) / \text{FLAT}$$



## **5.- OBSERVACIÓN VISUAL DEL CIELO NOCTURNO (II)**

### **Objetivos**

Esta última práctica se ha concebido como el acto de clausura del curso de *"Introducción a la Astrofísica"*. Los alumnos habrán preparado de antemano una lista con los principales hitos astronómicos observables en el día y hora de realización de la práctica: planetas, constelaciones, nebulosas, cúmulos abiertos y globulares, galaxias, etc., tomando nota de las magnitudes aparentes y principales propiedades. Para algún objeto y hora concreta determinarán el acimut y elevación (suponiendo el observador en Granada).

Se intentarán localizar estos objetos con la ayuda del planisferio y se observarán a través de unos prismáticos de calidad astronómica de gran campo. Se pondrán a prueba la mayoría de los conocimientos adquiridos durante el curso: astronomía de posición, Sistema Solar, física estelar y física galáctica.

### **Material necesario**

- Planisferio celeste: uno por cada dos alumnos
- Prismáticos astronómicos
- Linterna roja
- Puntero láser
- Ropa de abrigo acorde con la fecha de observación
- Medio de transporte a un lugar libre de contaminación lumínica

### **Lugar y fecha de realización**

La práctica se llevará a cabo en un lugar de atmósfera limpia y apartado de las luces de la ciudad. Para ello se requiere un autobús para el transporte de los alumnos y el material necesario. Esta práctica se realizarán a finales del mes de Mayo o principios de Junio.